Einführung in die Theoretische Informatik

Johannes Köbler

Institut für Informatik
Humboldt-Universität zu Berlin

WS 2022/23

Inhalt der Vorlesung

Themen dieser VL:

- **Welche Rechenmodelle eignen sich zur Lösung welcher algorithmischen Problemstellungen?** **Automatentheorie**
- **Welche algorithmischen Probleme sind überhaupt lösbar?** **Berechenbarkeitstheorie**
- **Welcher Aufwand ist zur Lösung eines geg. algorithmischen Problems nötig?** **Komplexitätstheorie**

Themen der VL Algorithmen und Datenstrukturen:

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

Themen der VL Logik in der Informatik:

- Mathem. Grundlagen der Informatik, Beweise führen, Modellierung Aussagenlogik, Prädikatenlogik

Lernziele

- 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

Maschinenmodelle

- Rechenmaschinen spielen in der Informatik eine zentrale Rolle
- Es gibt viele unterschiedliche math. 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 Algorithmenbegriff

- Der Begriff Algorithmus geht auf den persischen Gelehrten Muhammed Al Chwarizmi (8./9. Jhd.) zurück
- Ältester bekannter nicht-trivialer Algorithmus: Euklidischer Algorithmus zur Berechnung des ggT (300 v. Chr.)
- Von einem Algorithmus wird erwartet, dass er bei jeder zulässigen 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

Alphabet, Wort, Sprache

Definition

- Ein Alphabet ist eine endliche linear geordnete Menge

  $$\Sigma = \{a_1, \dots, a_m\}$$

  von $m \geq 1$ Zeichen $a_1 < \cdots < a_m$
- Eine Folge $x = x_1 \dots x_n$ von $n \geq 0$ Zeichen $x_i \in \Sigma$ heißt Wort der Länge n über Σ
- Die Länge von x wird mit $|x|$ und die Menge aller Wörter der Länge n über Σ wird mit Σ^n bezeichnet
- Die Menge aller Wörter über Σ ist

  $$\Sigma^* = \bigcup_{n \geq 0} \Sigma^n = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \cdots$$

- Das (einzige) Wort der Länge $n = 0$ ist das leere Wort, welches wir mit ε bezeichnen, d.h. $\Sigma^0 = \{\varepsilon\}$
- Jede Teilmenge $L \subseteq \Sigma^*$ heißt Sprache über dem Alphabet Σ

Beispiele für Sprachen

Beispiel

- Sprachen über Σ sind beispielsweise $\varnothing, \Sigma^*, \Sigma$ und $\{\varepsilon\}$
- $\varnothing$ enthält keine Wörter und heißt leere Sprache
- Σ^* enthält dagegen alle Wörter über Σ
- Σ enthält alle Wörter über Σ der Länge 1
- $\{\varepsilon\}$ enthält nur das leere Wort, ist also einelementig
- Sprachen, die genau ein Wort enthalten, werden auch als Singletonsprachen bezeichnet
- In der Informatik spielen Programmiersprachen eine wichtige Rolle

Operationen auf Sprachen

- 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
  - $A \cap B = \{x \in \Sigma^* \mid x \in A \wedge x \in B\}$ 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

Konkatenation von Wörtern

Definition

Die Konkatenation von zwei Wörtern $x = x_1 \dots x_n$ und $y = y_1 \dots y_m$ ist das Wort $x \circ y = x_1 \dots x_n y_1 \dots y_m$, das wir auch einfach mit xy bezeichnen

Beispiel

- Für $x = aba$ und $y = abab$ erhalten wir $xy = abaabab$ und $yx = abababa$
- Die Konkatenation ist also nicht kommutativ
- Allerdings ist $\circ$ assoziativ, d.h. es gilt $x(yz) = (xy)z$
  Daher können wir hierfür auch einfach xyz schreiben
- Es gibt auch ein neutrales Element, da $x\varepsilon = \varepsilon x = x$ ist
- Eine algebraische Struktur $(M, \square, e)$ mit einer assoziativen Operation $\square : M \times M \to M$ und einem neutralen Element e heißt Monoid
- $(\Sigma^*, \circ, \varepsilon)$ ist also ein Monoid

Spezielle Sprachoperationen

Neben den Mengenoperationen Schnitt, Vereinigung und Komplement gibt es auch spezielle Sprachoperationen

Definition

- Das Produkt (Verkettung, Konkatenation) von zwei Sprachen A und B ist

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

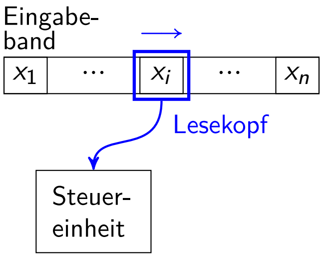
- Ist $A = \{x\}$ eine Singletonsprache, so schreiben wir für $\{x\}B$ auch einfach xB
- Die n-fache Potenz A^n einer Sprache A ist induktiv definiert durch

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

- Die Sternhülle einer Sprache A ist $A^* = \bigcup_{n \geq 0} A^n$
- Die Plushülle einer Sprache A ist $A^+ = \bigcup_{n \geq 1} A^n = AA^*$

Algorithmische Erkennung von Sprachen

- Ein einfaches Rechenmodell zum Erkennen von Sprachen ist der endliche Automat:



- Ein endlicher Automat
  - nimmt zu jedem Zeitpunkt genau einen von endlich vielen Zuständen an
  - macht bei Eingaben der Länge n genau n Rechenschritte und
  - liest in jedem Schritt genau ein Eingabezeichen

Formale Definition eines endlichen Automaten

Definition

- 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 einem DFA $M = (Z, \Sigma, \delta, q_0, E)$ 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

Die Klasse der regulären Sprachen

Frage

Welche Sprachen lassen sich durch endliche Automaten erkennen und welche nicht?

Definition

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

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

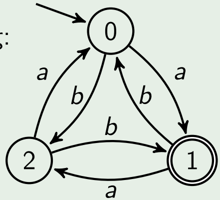
DFAs beherrschen Modulare Arithmetik

Beispiel

Sei $M_3 = (Z, \Sigma, \delta, 0, E)$ ein DFA 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:



Endzustände werden durch einen doppelten Kreis und der Startzustand wird durch einen Pfeil gekennzeichnet ◁

Frage: Welche Wörter akzeptiert M_3?

- Ist $w_1 = aba \in L(M_3)$? Ja (akzeptierende Rechnung: $0, 1, 0, 1$)
- Ist $w_2 = abba \in L(M_3)$? Nein (verwerfende Rechnung: $0, 1, 0, 2, 0$)

DFAs beherrschen Modulare Arithmetik

Behauptung

Die von M_3 erkannte Sprache ist

$$L(M_3) = \{x \in \{a,b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 1\}, \quad \text{wobei}$$

- $\#_a(x)$ die Anzahl der Vorkommen von 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 der Behauptung durch Induktion über die Länge von x

Wir betrachten zunächst das Erreichbarkeitsproblem für DFAs

Das Erreichbarkeitsproblem für DFAs

Frage

Sei $M = (Z, \Sigma, \delta, q_0, E)$ ein DFA und sei $x = x_1 \dots x_n \in \Sigma^*$. Welchen Zustand erreicht M nach Lesen der ersten i Zeichen der Eingabe x?

Antwort

- $i = 0$: den Startzustand q_0
- $i = 1$: den Zustand $\delta(q_0, x_1)$
- $i = 2$: den Zustand $\delta(\delta(q_0, x_1), x_2)$

  $\vdots$
- $i = n$: den Zustand $\delta(\dots\delta(\delta(q_0, x_1), x_2), \dots x_n)$

Das Erreichbarkeitsproblem für DFAs

Definition

- Für $x \in \Sigma^*$ 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 über die Länge n von x wie folgt definieren:
  Für $q \in Z$, $x \in \Sigma^n$ und $a \in \Sigma$ sei

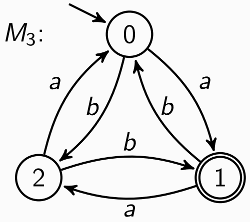
  $$\begin{aligned} n = 0: & \quad \hat{\delta}(q, \varepsilon) = q, \\ n \rightsquigarrow n + 1: & \quad \hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a) \end{aligned}$$

- Die von M erkannte Sprache lässt sich nun elegant durch

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

  beschreiben

DFAs beherrschen Modulare Arithmetik



Behauptung

Für alle $x \in \{a, b\}^*$ gilt die Äquivalenz

$$x \in L(M_3) \Leftrightarrow \#_a(x) - \#_b(x) \equiv_3 1$$

Beweis

- 1 ist der einzige Endzustand von M
- Daher ist $L(M_3) = \{x \in \{a, b\}^* \mid \hat{\delta}(0, x) = 1\}$
- Obige Behauptung ist also äquivalent zu folgender Behauptung

$$\text{Für alle } x \in \{a, b\}^* \text{gilt}: \ \hat{\delta}(0, x) = 1 \Leftrightarrow \#_a(x) - \#_b(x) \equiv_3 1$$

- Folglich reicht es, für alle $x \in \{a, b\}^*$ folgende Kongruenz zu zeigen:

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

DFAs beherrschen Modulare Arithmetik

Induktionsbehauptung: Für alle $x \in \{a,b\}^n$ gilt $\hat{\delta}(0,x) \equiv_3 \#_a(x) - \#_b(x)$

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

Induktionsschritt ($n \rightsquigarrow n+1$): Sei $x = x_1 \ldots x_{n+1} \in \{a,b\}^{n+1}$ gegeben

- Nach Induktionsvoraussetzung (IV) gilt für $x' = x_1 \ldots x_n$:

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

- Zudem gilt für alle $q \in Z = \{0,1,2\}$:

$$\begin{aligned}\delta(q,x_{n+1}) &\equiv_3 \begin{cases} q+1, & x_{n+1} = a \\ q-1, & x_{n+1} = b \end{cases} \\ &= q + \#_a(x_{n+1}) - \#_b(x_{n+1}) && (*)\end{aligned}$$

- Somit folgt

$$\begin{aligned}\hat{\delta}(0,x) &= \delta(\hat{\delta}(0,x'),x_{n+1}) \\ &\equiv_3 \hat{\delta}(0,x') + \#_a(x_{n+1}) - \#_b(x_{n+1}) && (*) \\ &\equiv_3 \#_a(x') - \#_b(x') + \#_a(x_{n+1}) - \#_b(x_{n+1}) && (IV) \\ &= \#_a(x) - \#_b(x)\end{aligned}$$

□

Singletonsprachen sind regulär

Vereinbarung

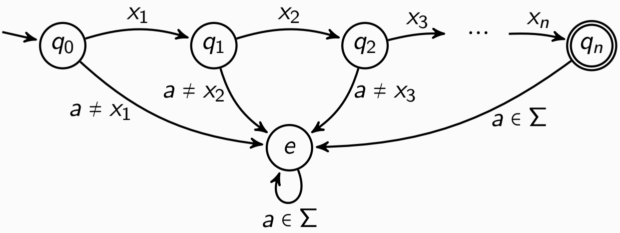
Für das Folgende sei $\Sigma = \{a_1, \ldots, a_m\}$ ein fest gewähltes Alphabet

Beobachtung 1

Alle Sprachen, die nur ein Wort $x = x_1 \ldots x_n \in \Sigma^*$ enthalten, sind regulär

Beweis

Folgender DFA M erkennt die Sprache $L(M) = \{x\}$:



□

REG ist unter Komplement abgeschlossen

Beobachtung 2

Ist $L \in \mathrm{REG}$, so ist auch die Sprache $\overline{L} = \Sigma^* \setminus L$ regulär

Beweis

- Sei $M = (Z, \Sigma, \delta, q_0, E)$ ein DFA mit $L(M) = L$
- Dann wird das Komplement $\overline{L}$ von L von dem DFA $\overline{M} = (Z, \Sigma, \delta, q_0, Z \setminus E)$ akzeptiert □

Definition

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

Korollar

co-REG = REG

REG ist unter Schnitt abgeschlossen

Beobachtung 3

Sind $L_1, L_2 \in \mathrm{REG}$, so ist auch die Sprache $L_1 \cap L_2$ regulär

Beweis

- Seien $M_i = (Z_i, \Sigma, \delta_i, q_i, E_i)$, $i = 1, 2$, DFAs mit $L(M_i) = L_i$.
- Dann wird der Schnitt $L_1 \cap L_2$ von dem DFA

  $$M = (Z_1 \times Z_2, \Sigma, \delta, (q_1, q_2), E_1 \times E_2)$$

  mit

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

  erkannt
- M wird auch als Kreuzproduktautomat bezeichnet □

REG ist unter Vereinigung abgeschlossen

Beobachtung 4

Die Vereinigung $L_1 \cup L_2$ von regulären Sprachen L_1 und L_2 ist regulär

Beweis

Es gilt $L_1 \cup L_2 = \overline{(\overline{L_1} \cap \overline{L_2})}$ $\square$

Frage

Wie sieht der zugehörige DFA M' aus?

Antwort

- Seien $M_i = (Z_i, \Sigma, \delta_i, q_i, E_i)$, $i = 1, 2$, DFAs mit $L(M_i) = L_i$
- Dann gilt $L(M') = L_1 \cup L_2$ für den DFA

$$M' = (Z_1 \times Z_2, \Sigma, \delta, (q_1, q_2), (E_1 \times Z_2) \cup (Z_1 \times E_2))$$

mit $\delta((p,q), a) = (\delta_1(p,a), \delta_2(q,a))$ für alle $(p,q) \in Z_1 \times Z_2$ und $a \in \Sigma$

Abschlusseigenschaften von Sprachklassen

Definition

- 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
- 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 (bzgl. Inklusion) kleinste Sprachklasse $\mathcal{K}'$, die $\mathcal{K}$ enthält und unter op abgeschlossen ist

Beispiel

- Der 2-stellige Schnittoperator $\cap$ bildet L_1 und L_2 auf $L_1 \cap L_2$ ab
- 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 $\triangleleft$

REG ist unter Mengenoperationen abgeschlossen

Korollar

Die Klasse REG der regulären Sprachen ist unter folgenden Operationen abgeschlossen:

- Komplement
- Schnitt
- Vereinigung

Wie umfangreich ist REG?

Folgerung

- Aus den Beobachtungen folgt, dass alle endlichen und alle co-endlichen Sprachen regulär sind
- Da die reguläre Sprache

  $$L(M_3) = \{x \in \{a,b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 1\}$$

  weder endlich noch co-endlich ist, haben wir damit allerdings noch nicht alle regulären Sprachen erfasst

Wie umfangreich ist REG?

Nächstes Ziel

Zeige, dass REG unter Produktbildung und Sternhülle abgeschlossen ist

Problem

Bei der Konstruktion eines DFA M für das Produkt $L(M_1)L(M_2)$ bereitet es Schwierigkeiten, den richtigen Zeitpunkt für das Ende der Simulation des DFA M_1 und den Start der Simulation des DFA M_2 zu finden

Lösungsidee

Ein nichtdeterministischer endlicher Automat (NFA) kann den richtigen Zeitpunkt „raten“

Verbleibendes Problem

Zeige, dass auch NFAs nur reguläre Sprachen erkennen

Nichtdeterministische endliche Automaten

Definition

- Ein nichtdeterministischer endlicher Automat $N = (Z, \Sigma, \Delta, Q_0, E)$ (kurz: NFA; *Nondeterministic Finite Automaton*) ist genau so aufgebaut wie ein DFA, nur dass er
  - eine Menge $Q_0 \subseteq Z$ von Startzuständen hat und
  - die 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 oft auch mit 2^Z bezeichnet
- Die von einem NFA $N = (Z, \Sigma, \Delta, Q_0, E)$ akzeptierte (oder erkannte) Sprache ist

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

- Eine Zustandsfolge $q_0, \dots, q_n$ heißt Rechnung von $N(x_1 \dots x_n)$, falls $q_0 \in Q_0$ und $q_{i+1} \in \Delta(q_i, x_{i+1})$ für $i = 0, \dots, n-1$ gilt

Eigenschaften von NFAs

- 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, der jede Eingabe zu Ende liest, kann ein NFA N „stecken bleiben“
- Dieser Fall tritt ein, wenn N in einen Zustand q gelangt, in dem er das nächste Eingabezeichen x_i wegen

  $$\Delta(q, x_i) = \varnothing$$

  nicht lesen kann

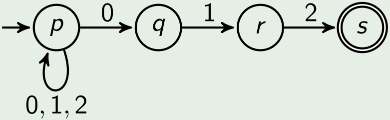
Eigenschaften von NFAs

Beispiel

- Betrachte den NFA $N = (Z, \Sigma, \Delta, Q_0, E)$ mit $Z = \{p, q, r, s\}$, $\Sigma = \{0, 1, 2\}$, $Q_0 = \{p\}$, $E = \{s\}$ und 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:



- Ist $w_1 = 012 \in L(N)$? Ja: (akzeptierende Rechnung: p, q, r, s)
  Es gibt aber auch verwerfende Rechnungen bei Eingabe w_1: p, p, p, p
- Ist $w_2 = 021 \in L(N)$? Nein, da es keine akzeptierende Rechnung gibt
- Es gilt $L(N) = \Sigma^* 012 = \{x012 \mid x \in \Sigma^*\}$ ◁

Ein NFA für das Produkt von regulären Sprachen

Beobachtung 5

Seien $N_i = (Z_i, \Sigma, \Delta_i, Q_i, E_i)$ NFAs mit $L(N_i) = L_i$ für $i = 1, 2$. Dann wird auch das Produkt $L_1 L_2$ von einem NFA erkannt

Beweis

- Wir können $Z_1 \cap Z_2 = \varnothing$ annehmen
- Dann gilt $L(N) = L_1 L_2$ für den NFA $N = (Z_1 \cup Z_2, \Sigma, \Delta, Q_1, E)$ mit

$$\Delta(p, a) = \begin{cases} \Delta_1(p, a), & p \in Z_1 \setminus E_1, \\ \Delta_1(p, a) \cup \bigcup_{q \in Q_2} \Delta_2(q, a), & p \in E_1, \\ \Delta_2(p, a), & p \in Z_2 \end{cases}$$

und

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

Ein NFA für das Produkt von regulären Sprachen

- Dann gilt $L(N) = L_1 L_2$ für den NFA $N = (Z_1 \cup Z_2, \Sigma, \Delta, Q_1, E)$ mit

$$\Delta(p,a) = \begin{cases} \Delta_1(p,a), & p \in Z_1 \setminus E_1, \\ \Delta_1(p,a) \cup \bigcup_{q \in Q_2} \Delta_2(q,a), & p \in E_1, \\ \Delta_2(p,a), & p \in Z_2 \end{cases}$$

  und $E = E_2$, falls $Q_2 \cap E_2 = \varnothing$, bzw. $E = E_1 \cup E_2$ sonst

Beweis von $L_1 L_2 \subseteq L(N)$: Seien $x = x_1 \dots x_k \in L_1$, $y = y_1 \dots y_l \in L_2$ und seien $q_0, \dots, q_k$ und $p_0, \dots, p_l$ akz. 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

- $q_i \in \Delta(q_{i-1}, x_i)$ für $i = 1, \dots, k$ ist (wg. $q_i \in \Delta_1(q_{i-1}, x_i)$) und
- im Fall $l = 0$ zudem $q_k \in E$ ist (wg. $q_k \in E_1$ und $p_l \in Q_2 \cap E_2$), sowie
- im Fall $l \geq 1$ zudem $p_1 \in \Delta(q_k, y_1)$ (wg. $q_k \in E_1$, $p_0 \in Q_2$, $p_1 \in \Delta_2(p_0, y_1)$) und $p_j \in \Delta(p_{j-1}, y_j)$ für $j = 2, \dots, l$ (wegen $p_j \in \Delta_2(p_{j-1}, y_j)$) und $p_l \in E_2 \subseteq E$ ist

Ein NFA für das Produkt von regulären Sprachen

- Dann gilt $L(N) = L_1 L_2$ für den NFA $N = (Z_1 \cup Z_2, \Sigma, \Delta, Q_1, E)$ mit
$$\Delta(p,a) = \begin{cases} \Delta_1(p,a), & p \in Z_1 \setminus E_1, \\ \Delta_1(p,a) \cup \bigcup_{q \in Q_2} \Delta_2(q,a), & p \in E_1, \\ \Delta_2(p,a), & p \in Z_2 \end{cases}$$
und $E = E_2$, falls $Q_2 \cap E_2 = \varnothing$, bzw. $E = E_1 \cup E_2$ sonst

Beweis von $L(N) \subseteq L_1 L_2$: Sei $x = x_1 \dots x_n \in L(N)$ 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 $i \le n$.

- 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. Da zudem $q_n \in E \cap Z_2 = E_2$ ist, ist $q_0, \dots, q_i$ eine akz. Rechnung von $N_1(x_1 \dots x_i)$ und $q, q_{i+1}, \dots, q_n$ eine akz. Rechnung von $N_2(x_{i+1} \dots x_n)$
- Im Fall $i = n$ ist $q_n \in E \cap Z_1$, was $q_n \in E_1$ und $Q_2 \cap E_2 \neq \varnothing$ impliziert. Also ist $q_0, \dots, q_n$ eine akz. Rechnung von $N_1(x_1 \dots x_n)$ und $\varepsilon \in L_2$ $\square$

Ein NFA für die Sternhülle einer regulären Sprache

Beobachtung 6

Ist $N = (Z, \Sigma, \Delta, Q_0, E)$ ein NFA, so wird auch die Sprache $L(N)^*$ von einem NFA erkannt

Beweis

Die Sprache $L(N)^*$ wird von dem NFA

$$N^* = (Z \cup \{q_{neu}\}, \Sigma, \Delta^*, Q_0 \cup \{q_{neu}\}, E \cup \{q_{neu}\})$$

mit

$$\Delta^*(p,a) = \begin{cases} \Delta(p,a), & p \in Z \setminus E, \\ \Delta(p,a) \cup \bigcup_{q \in Q_0} \Delta(q,a), & p \in E, \\ \varnothing, & p = q_{neu} \end{cases}$$

erkannt □

Überblick

Ziel

Zeige, dass REG unter Produktbildung und Sternhülle abgeschlossen ist

Problem

Bei der Konstruktion eines DFA für das Produkt L_1L_2 bereitet es Schwierigkeiten, den richtigen Zeitpunkt für den Übergang von (der Simulation von) M_1 zu M_2 zu finden

Lösungsidee (bereits umgesetzt)

Ein nichtdeterministischer Automat (NFA) kann den richtigen Zeitpunkt für den Übergang „raten“

Noch zu zeigen

NFAs erkennen genau die regulären Sprachen

NFAs erkennen genau die regulären Sprachen

Satz (Rabin und Scott)

$\mathrm{REG} = \{L(N) \mid N \text{ ist ein NFA}\}$

Beweis von $\mathrm{REG} \subseteq \{L(N) \mid N \text{ ist ein NFA}\}$

Diese Inklusion ist klar, da jeder DFA $M = (Z, \Sigma, \delta, q_0, E)$ in einen äquivalenten NFA

$$N = (Z, \Sigma, \Delta, Q_0, E)$$

transformiert werden kann, indem wir $\Delta(q, a) = \{\delta(q, a)\}$ und $Q_0 = \{q_0\}$ setzen. $\square$

Für die umgekehrte Inklusion ist das Erreichbarkeitsproblem für NFAs von zentraler Bedeutung

Das Erreichbarkeitsproblem für NFAs

Frage

Sei $N = (Z, \Sigma, \Delta, Q_0, E)$ ein NFA und sei $x = x_1 \dots x_n$ eine Eingabe. Welche Zustände sind in i Schritten erreichbar?

Antwort

- in 0 Schritten: alle Zustände in Q_0
- nach einem Schritt: alle Zustände in

$$Q_1 = \bigcup_{q \in Q_0} \Delta(q, x_1)$$

- nach i Schritten: alle Zustände in

$$Q_i = \bigcup_{q \in Q_{i-1}} \Delta(q, x_i)$$

Simulation von NFAs durch DFAs

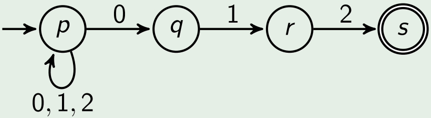
Idee

- Wir können einen NFA $N = (Z, \Sigma, \Delta, Q_0, E)$ durch einen DFA $M = (Z', \Sigma, \delta, q_0', E')$ simulieren, der in seinem Zustand die Information speichert, in welchen Zuständen sich N momentan befinden könnte
- Die Zustände von M sind also Teilmengen Q von Z (d.h. $Z' = \mathcal{P}(Z)$) mit Q_0 als Startzustand (d.h. $q_0' = Q_0$) und der Endzustandsmenge
$$E' = \{Q \subseteq Z \mid Q \cap E \neq \varnothing\}$$
- Die Überführungsfunktion $\delta : \mathcal{P}(Z) \times \Sigma \to \mathcal{P}(Z)$ von M berechnet dann für einen Zustand $Q \subseteq Z$ und ein Zeichen $a \in \Sigma$ die Menge
$$\delta(Q, a) = \bigcup_{q \in Q} \Delta(q, a)$$
aller Zustände, die N ausgehend von einem beliebigen Zustand $q \in Q$ bei Lesen des Zeichens a erreichen kann
- M wird auch als der zu N gehörige Potenzmengenautomat bezeichnet

Simulation von NFAs durch DFAs

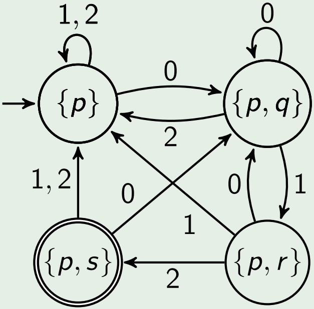
Beispiel

- Betrachte den NFA N



- Ausgehend von $Q_0 = \{p\}$ liefert δ dann die folgenden Werte:

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



Simulation von NFAs durch DFAs

- Im obigen Beispiel werden für die Konstruktion des Potenzmengenautomaten nur 4 der insgesamt

  $$|\mathcal{P}(Z)| = 2^{|Z|} = 2^4 = 16$$

  Zustände benötigt, da die übrigen 12 Zustände nicht erreichbar sind (hierbei bezeichnet $|A|$ die Mächtigkeit einer Menge A)
- Es gibt jedoch Beispiele, bei denen alle $2^{|Z|}$ Zustände benötigt werden (siehe Übungen)

NFAs erkennen genau die regulären Sprachen

Beweis von $\{L(N) \mid N \text{ ist ein NFA}\} \subseteq \mathrm{REG}$

- Sei $N = (Z, \Sigma, \Delta, Q_0, E)$ ein NFA und sei $M = (\mathcal{P}(Z), \Sigma, \delta, Q_0, E')$ der zugehörige Potenzmengenautomat mit $\delta(Q, a) = \bigcup_{q \in Q} \Delta(q, a)$ und $E' = \{Q \subseteq Z \mid Q \cap E \neq \varnothing\}$
- Dann folgt die Korrektheit von M mittels folgender Behauptung, die wir auf der nächsten Folie beweisen

Behauptung

$\hat{\delta}(Q_0, x)$ enthält genau die von N nach Lesen von x erreichbaren Zustände

- Für alle Wörter $x \in \Sigma^*$ gilt

$$\begin{aligned} x \in L(N) &\Leftrightarrow N \text{ kann nach Lesen von } x \text{ einen Endzustand erreichen} \\ &\overset{\textit{Beh.}}{\Leftrightarrow} \hat{\delta}(Q_0, x) \cap E \neq \varnothing \\ &\Leftrightarrow \hat{\delta}(Q_0, x) \in E' \\ &\Leftrightarrow x \in L(M) \end{aligned}$$

□

Beweis der Behauptung

Behauptung

$\hat{\delta}(Q_0, x)$ enthält genau die von N nach Lesen von x erreichbaren Zustände

Beweis durch Induktion über die Länge n von x

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

$n \rightsquigarrow n+1$: Sei $x = x_1 \dots x_{n+1}$ gegeben und sei $x' = x_1 \dots x_n$.

Nach IV enthält dann $\hat{\delta}(Q_0, x')$ genau die von N nach Lesen von x' erreichbaren Zustände. Wegen

$$\begin{aligned} \hat{\delta}(Q_0, x) &= \delta(\hat{\delta}(Q_0, x'), x_{n+1}) && \text{(nach Def. von } \hat{\delta}) \\ &= \bigcup_{q \in \hat{\delta}(Q_0, x')} \Delta(q, x_{n+1}) && \text{(nach Def. von } \delta) \end{aligned}$$

enthält dann $\hat{\delta}(Q_0, x)$ genau die Zustände, die N nach Lesen von x erreichen kann $\square$

Abschlusseigenschaften der Klasse REG

Satz (Rabin und Scott)

REG = $\{L(N) \mid N \text{ ist ein NFA}\}$

Korollar

Die Klasse REG der regulären Sprachen ist unter folgenden Operationen abgeschlossen:

- Komplement
- Schnitt
- Vereinigung
- Produkt
- Sternhülle

Überblick

Nächstes Ziel

Zeige, dass REG als Abschluss der endlichen Sprachen unter Vereinigung, Produkt und Sternhülle charakterisierbar ist

Bereits gezeigt:

Jede Sprache, die mittels der Operationen Vereinigung, Produkt und Sternhülle (sowie Schnitt und Komplement) angewandt auf endliche Sprachen darstellbar ist, ist regulär

Noch zu zeigen:

Jede reguläre Sprache lässt sich aus endlichen Sprachen mittels Vereinigung, Produkt und Sternhülle erzeugen

Konstruktive Charakterisierung von REG

Induktive Definition der Menge RA_Σ aller regulären Ausdrücke über Σ

Die Symbole $\emptyset$, ϵ und a ($a \in \Sigma$) sind reguläre Ausdrücke über Σ, die

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

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

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

Bemerkung

RA_Σ ist eine Sprache über dem Alphabet $\Gamma = \Sigma \cup \{\emptyset, \epsilon, |, ^*, (,)\}$

Reguläre Ausdrücke

Beispiel

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

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

◁

Vereinbarungen

- Um Klammern zu sparen, definieren wir folgende Präzedenzordnung: Der Sternoperator * bindet stärker als der Produktoperator und dieser wiederum stärker als der Vereinigungsoperator $|$
- Zudem nutzen wir die Assoziativität von $|$ zur Klammerersparnis
- Für $(0|(\epsilon 0|\varnothing(1)^*))$ können wir also kurz $0|\epsilon 0|\varnothing 1^*$ schreiben
- Da der reguläre Ausdruck $\gamma\gamma^*$ die Sprache $L(\gamma)^+$ beschreibt, verwenden wir γ^+ als Abkürzung für den Ausdruck $\gamma\gamma^*$

Charakterisierung von REG durch reguläre Ausdrücke

Satz

REG $= \{L(\gamma) \mid \gamma$ ist ein regulärer Ausdruck$\}$

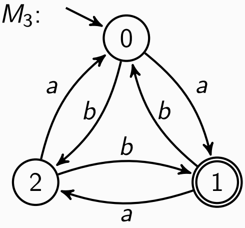
Beweis der Inklusion von rechts nach links.

Klar, da

- die Basisausdrücke $\varnothing$, ϵ und a, $a \in \Sigma^*$, reguläre Sprachen beschreiben und
- die Sprachklasse REG unter Produkt, Vereinigung und Sternhülle abgeschlossen ist $\square$

Charakterisierung von REG durch reguläre Ausdrücke

Für die umgekehrte Inklusion betrachten wir zunächst den DFA M_3.



Frage

Wie lässt sich die Sprache

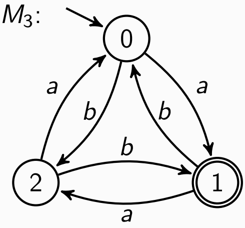
$$L(M_3) = \{x \in \{a,b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 1\}$$

durch einen regulären Ausdruck beschreiben?

Antwort

- Sei $L_{p,q}$ die Sprache aller Wörter x, die M_3 vom Zustand p in den Zustand q überführen (d.h. $L_{p,q} = \{x \in \{a,b\}^* \mid \hat{\delta}(p,x) = q\}$)
- Weiter sei $L_{p,q}^{\neq r}$ die Sprache aller Wörter $x = x_1 \ldots x_n \in L_{p,q}$, die hierzu nur Zustände ungleich r benutzen (d.h. $\hat{\delta}(p, x_1 \ldots x_i) \neq r$ für $i = 1, \ldots, n-1$)
- Dann gilt $L(M_3) = L_{0,1} = L_{0,0} L_{0,1}^{\neq 0}$, wobei $L_{0,0} = (L_{0,0}^{\neq 0})^*$ ist

Antwort (Fortsetzung)



- Dann ist $L(M_3) = L_{0,0} L_{0,1}^{\neq 0} = (L_{0,0}^{\neq 0})^* L_{0,1}^{\neq 0}$
- $L_{0,1}^{\neq 0}$ und $L_{0,0}^{\neq 0}$ lassen sich durch folgende reguläre Ausdrücke beschreiben:

$$\gamma_{0,1}^{\neq 0} = (a|bb)(ab)^*$$

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

- Also ist $L(M_3)$ durch folgenden regulären Ausdruck beschreibbar:

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

Charakterisierung von REG durch reguläre Ausdrücke

Satz

REG $= \{L(\gamma) \mid \gamma$ ist ein regulärer Ausdruck$\}$

Beweis der Inklusion von links nach rechts.

- Wir konstruieren zu einem DFA $M = (Z, \Sigma, \delta, q_0, E)$ einen regulären Ausdruck γ mit $L(\gamma) = L(M)$
- Wir nehmen an, dass $Z = \{1, \dots, m\}$ 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
- Es reicht also, reguläre Ausdrücke für die Sprachen $L_{p,q}$ mit $1 \le p, q \le m$ anzugeben

Charakterisierung von REG durch reguläre Ausdrücke

Satz

REG $\subseteq \{L(\gamma) \mid \gamma$ ist ein regulärer Ausdruck$\}$

Beweis (Fortsetzung)

- Es reicht also, reguläre Ausdrücke für die Sprachen $L_{p,q}$ mit $1 \leq p, q \leq m$ anzugeben
- Hierzu betrachten wir für $r = 0, \ldots, m$ die Sprachen

  $$L_{p,q}^{\leq r} = \left\{ x_1 \ldots x_n \in L_{p,q} \,\middle|\, \text{für } i = 1, \ldots, n-1 \text{ ist } \hat{\delta}(p, x_1 \ldots x_i) \leq r \right\},$$

  die wir auch einfach mit $L_{p,q}^r$ bezeichnen
- Wegen $L_{p,q} = L_{p,q}^m$ reicht es, reguläre Ausdrücke für die Sprachen $L_{p,q}^r$ mit $1 \leq p, q \leq m$ und $0 \leq r \leq m$ anzugeben
- Wir zeigen induktiv über r, dass die Sprachen $L_{p,q}^r$ durch reguläre Ausdrücke beschreibbar sind

Charakterisierung von REG durch reguläre Ausdrücke

Satz

REG $\subseteq \{L(\gamma) \mid \gamma$ ist ein regulärer Ausdruck$\}$

Beweis (Schluss)

$r = 0$: In diesem Fall sind die Sprachen

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

endlich, also durch reg. Ausdrücke $\gamma^0_{p,q}$ beschreibbar

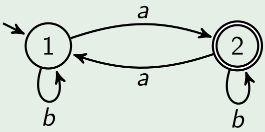
$r \rightsquigarrow r+1$: Nach IV existieren reguläre Ausdrücke $\gamma^r_{p,q}$ für die Sprachen $L^r_{p,q}$. Wegen

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

sind dann $\gamma^{r+1}_{p,q} = \gamma^r_{p,q} | \gamma^r_{p,r+1}(\gamma^r_{r+1,r+1})^* \gamma^r_{r+1,q}$ reguläre Ausdrücke für die Sprachen $L^{r+1}_{p,q}$ □

Beispiel

- Betrachte den DFA M



- Da M nur einen Endzustand $q = 2$ und insgesamt $m = 2$ Zustände besitzt, folgt

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

Beispiel (Fortsetzung)

- Um reguläre Ausdrücke $\gamma^r_{p,q}$ für die Sprachen $L^r_{p,q}$ zu bestimmen, benutzen wir für $r \geq 0$ die Rekursionsformel

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

- Damit erhalten wir

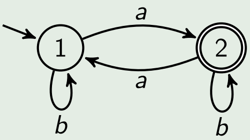
$$\begin{aligned} \gamma^2_{1,2} &= \gamma^1_{1,2} | \gamma^1_{1,2} (\gamma^1_{2,2})^* \gamma^1_{2,2} \\ \gamma^1_{1,2} &= \gamma^0_{1,2} | \gamma^0_{1,1} (\gamma^0_{1,1})^* \gamma^0_{1,2} \\ \gamma^1_{2,2} &= \gamma^0_{2,2} | \gamma^0_{2,1} (\gamma^0_{1,1})^* \gamma^0_{1,2} \end{aligned}$$

- Für die Berechnung von $\gamma^2_{1,2}$ werden also nur die regulären Ausdrücke $\gamma^0_{1,1}$, $\gamma^0_{1,2}$, $\gamma^0_{2,1}$, $\gamma^0_{2,2}$, $\gamma^1_{2,2}$ und $\gamma^1_{1,2}$ benötigt

Charakterisierung von REG durch reguläre Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformeln

$$L^0_{p,p} = \{c \in \Sigma \mid \delta(p,c) = p\} \cup \{\varepsilon\}$$

$$L^0_{p,q} = \{c \in \Sigma \mid \delta(p,c) = q\} \text{ für } p \neq q$$

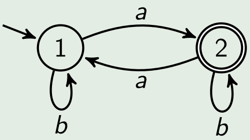
$$\gamma^{r+1}_{p,q} = \gamma^r_{p,q} | \gamma^r_{p,r+1} (\gamma^r_{r+1,r+1})^* \gamma^r_{r+1,q}$$

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\gamma^0_{1,1}$	$\gamma^0_{1,2}$	$\gamma^0_{2,1}$	$\gamma^0_{2,2}$
1	-	$\gamma^1_{1,2}$	-	$\gamma^1_{2,2}$
2	-	$\gamma^2_{1,2}$	-	-

Charakterisierung von REG durch reguläre Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

$$L^0_{1,1} = \{c \in \Sigma \mid \delta(1, c) = 1\} \cup \{\varepsilon\} = \{\varepsilon, b\}$$

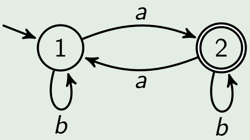
$$\rightsquigarrow \quad \gamma^0_{1,1} = \epsilon|b$$

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\epsilon\|b$	$\gamma^0_{1,2}$	$\gamma^0_{2,1}$	$\gamma^0_{2,2}$
1	-	$\gamma^1_{1,2}$	-	$\gamma^1_{2,2}$
2	-	$\gamma^2_{1,2}$	-	-

Charakterisierung von REG durch reguläre Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

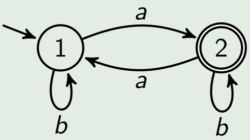
$$L^0_{1,2} = \{c \in \Sigma \mid \delta(1,c) = 2\} = \{a\}$$

$$\rightsquigarrow \quad \gamma^0_{1,2} = a$$

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\epsilon \mid b$	a	$\gamma^0_{2,1}$	$\gamma^0_{2,2}$
1	-	$\gamma^1_{1,2}$	-	$\gamma^1_{2,2}$
2	-	$\gamma^2_{1,2}$	-	-

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

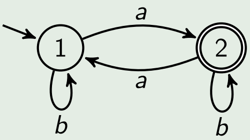
$$L^0_{2,1} = \{c \in \Sigma \mid \delta(2, c) = 1\} = \{a\}$$

$$\rightsquigarrow \quad \gamma^0_{2,1} = a$$

r	p, q			
	$1, 1$	$1, 2$	$2, 1$	$2, 2$
0	$\epsilon \vert b$	a	a	$\gamma^0_{2,2}$
1	-	$\gamma^1_{1,2}$	-	$\gamma^1_{2,2}$
2	-	$\gamma^2_{1,2}$	-	-

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

$$L^0_{2,2} = \{c \in \Sigma \mid \delta(2, c) = 2\} \cup \{\varepsilon\} = \{\varepsilon, b\}$$

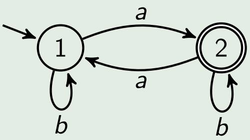
$$\rightsquigarrow \quad \gamma^0_{2,2} = \epsilon|b$$

r	p, q			
	$1, 1$	$1, 2$	$2, 1$	$2, 2$
0	$\epsilon\|b$	a	a	$\epsilon\|b$
1	-	$\gamma^1_{1,2}$	-	$\gamma^1_{2,2}$
2	-	$\gamma^2_{1,2}$	-	-

Charakterisierung von REG durch reguläre Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

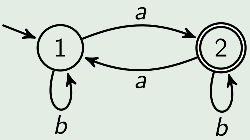
$$\gamma_{1,2}^{1} = \gamma_{1,2}^{0} | \gamma_{1,1}^{0} (\gamma_{1,1}^{0})^* \gamma_{1,2}^{0}$$
$$= a | (\epsilon | b)(\epsilon | b)^* a$$
$$\equiv b^* a$$

r	p, q			
	$1, 1$	$1, 2$	$2, 1$	$2, 2$
0	$\epsilon \| b$	a	a	$\epsilon \| b$
1	-	$b^* a$	-	$\gamma_{2,2}^{1}$
2	-	$\gamma_{1,2}^{2}$	-	-

Charakterisierung von REG durch reguläre Ausdrücke

Beispiel (Fortsetzung)

DFA M



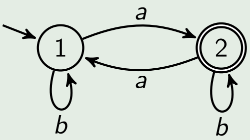
Rekursionsformel

$$\gamma_{2,2}^{1} = \gamma_{2,2}^{0} | \gamma_{2,1}^{0} (\gamma_{1,1}^{0})^* \gamma_{1,2}^{0}$$
$$= (\epsilon|b)|a(\epsilon|b)^* a$$
$$\equiv \epsilon|b|ab^* a$$

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\epsilon\|b$	a	a	$\epsilon\|b$
1	-	$b^* a$	-	$\epsilon\|b\|ab^* a$
2	-	$\gamma_{1,2}^{2}$	-	-

Beispiel (Fortsetzung)

DFA M



Rekursionsformel

$$\begin{aligned}\gamma_{1,2}^2 &= \gamma_{1,2}^1 | \gamma_{1,2}^1 (\gamma_{2,2}^1)^* \gamma_{2,2}^1 \\ &= b^*a | b^*a(\epsilon|b|ab^*a)^*(\epsilon|b|ab^*a) \\ &\equiv b^*a(b|ab^*a)^*\end{aligned}$$

r	p,q: 1, 1	1, 2	2, 1	2, 2
0	$\epsilon\|b$	a	a	$\epsilon\|b$
1	-	b^*a	-	$\epsilon\|b\|ab^*a$
2	-	$b^*a(b\|ab^*a)^*$	-	-

◁

Charakterisierungen der Klasse REG

Korollar

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 lässt sich mit den Operationen Vereinigung, Produkt und Sternhülle aus endlichen Sprachen gewinnen
- L lässt sich mit den Operationen Vereinigung, Schnitt, Komplement, Produkt und Sternhülle aus endlichen Sprachen gewinnen

Ausblick

- Als nächstes wenden wir uns der Frage zu, wie sich die Anzahl der Zustände eines DFA minimieren lässt
- Da hierbei Äquivalenzrelationen eine wichtige Rolle spielen, befassen wir uns zunächst mit Relationalstrukturen

Relationalstrukturen

Definition

- Sei A eine nichtleere Menge, R ist eine k-stellige Relation auf A, wenn $R \subseteq A^k = \underbrace{A \times \cdots \times A}_{k\text{-mal}} = \{(a_1, \ldots, a_k) \mid a_i \in A \text{ für } i = 1, \ldots, k\}$ ist
- Für $i = 1, \ldots, n$ sei R_i eine k_i-stellige Relation auf A. Dann heißt $(A; R_1, \ldots, R_n)$ Relationalstruktur
- Die Menge A heißt der Individuenbereich, die Trägermenge oder die Grundmenge der Relationalstruktur

- Wir werden hier nur 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

Relationalstrukturen

Beispiel

- (F, M) mit $F = \{f \mid f \text{ ist Fluss in Europa}\}$ und
  $$M = \{(f, g) \in F \times F \mid f \text{ mündet in } g\}$$
- (U, B) mit $U = \{x \mid x \text{ ist Berliner}\}$ und
  $$B = \{(x, y) \in U \times U \mid x \text{ ist Bruder von } y\}$$
- $(\mathcal{P}(M), \subseteq)$, wobei M eine beliebige Menge und $\subseteq$ die Inklusionsrelation auf den Teilmengen von M ist
- (A, Id_A) mit $Id_A = \{(x, x) \mid x \in A\}$ (die Identität auf A)
- $(\mathbb{R}, \leq)$
- $(\mathbb{Z}, |)$, wobei $|$ die "teilt"-Relation bezeichnet (d.h. $a|b$, falls ein $c \in \mathbb{Z}$ mit $b = ac$ existiert)

◁

Mengentheoretische Operationen auf Relationen

- Da Relationen Mengen sind, können wir den Schnitt, die Vereinigung, die Differenz und das Komplement von Relationen bilden:

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

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

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

Weitere Operationen auf Relationen

Definition

- 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 $(\mathbb{R}, \leq^T) = (\mathbb{R}, \geq)$
- Das Produkt (oder die Komposition) zweier Relationen R und S ist

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

Beispiel

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 $\triangleleft$

Das Relationenprodukt

Notation

- Für $R \circ S$ wird auch $R\,;S$, $R \cdot S$ oder einfach RS geschrieben
- Für $\underbrace{R \circ \cdots \circ R}_{\text{n-mal}}$ schreiben wir auch R^n, wobei $R^0 = Id$ ist

Vorsicht!

Das Relationenprodukt R^n darf nicht mit dem kartesischen Produkt

$$\underbrace{R \times \cdots \times R}_{\text{n-mal}}$$

verwechselt werden

Vereinbarung

Wir vereinbaren, dass R^n das n-fache Relationenprodukt bezeichnen soll, falls R eine Relation ist

Eigenschaften von Relationen

Definition

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

Überblick über Relationalstrukturen

Äquivalenz- und Ordnungsrelationen

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

- In der Tabelle sind nur die definierenden Eigenschaften durch ein "$\checkmark$" gekennzeichnet
- Das schließt nicht aus, dass noch weitere Eigenschaften vorliegen

Eigenschaften von Relationen

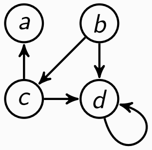
Beispiel

- 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 $(\mathcal{P}(M), \subseteq)$ sind reflexiv, antisymmetrisch und transitiv und somit Ordnungen
- $(\mathbb{R}, \leq)$ ist auch konnex und somit eine lineare Ordnung
- $(\mathcal{P}(M), \subseteq)$ ist zwar im Fall $|M| \leq 1$ konnex, aber im Fall $|M| \geq 2$ weder semikonnex noch konnex $\triangleleft$

Darstellung von endlichen Relationen

Graphische Darstellung

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



- Eine Relation R auf einer (endlichen) Menge A kann durch einen gerichteten Graphen (kurz Digraphen) $G = (A, R)$ mit Knotenmenge A und Kantenmenge 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 adjazent oder benachbart

Darstellung von endlichen Relationen

Definition

Sei R eine binäre Relation auf A

- Die Menge der Nachfolger bzw. Vorgänger von x ist

  $$R[x] = \{y \in A \mid xRy\} \text{ bzw. } R^{-1}[x] = \{y \in A \mid yRx\}$$

- Der Ausgangsgrad eines Knotens x ist $\deg^+(x) = |R[x]|$
- Der Eingangsgrad von x ist $\deg^-(x) = |R^{-1}[x]|$
- Ist R symmetrisch, so können wir die Pfeilspitzen auch weglassen
- In diesem Fall heißt $\deg(x) = \deg^-(x) = \deg^+(x)$ der Grad von x und $R[x] = R^{-1}[x]$ die Nachbarschaft von x in G
- G ist schleifenfrei, falls R irreflexiv ist
- Ist R irreflexiv und symmetrisch, so nennen wir $G = (A, R)$ 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 von endlichen Relationen

Matrixdarstellung (Adjazenzmatrix)

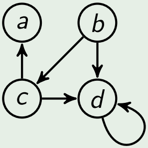
Eine Relation R auf $A = \{a_1, \ldots, a_n\}$ lässt sich auch durch die boolesche $(n \times n)$-Matrix $M_R = (m_{ij})$ darstellen mit

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

Beispiel

Die Relation $R = \{(b,c),(b,d),(c,a),(c,d),(d,d)\}$ auf $A = \{a,b,c,d\}$ hat beispielsweise 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 von endlichen Relationen

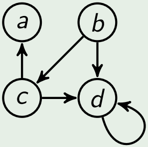
Listendarstellung (Adjazenzlisten)

R lässt sich auch durch eine Tabelle darstellen, die jedem Element $x \in A$ seine Nachfolger in Form einer Liste zuordnet

Beispiel

Die Relation $R = \{(b,c),(b,d),(c,a),(c,d),(d,d)\}$ auf $A = \{a,b,c,d\}$ lässt sich beispielsweise durch folgende Adjazenzlisten darstellen:

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



◁

Berechnung des Relationenprodukts

Berechnung von $R \circ S$

- 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

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

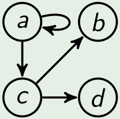
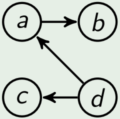
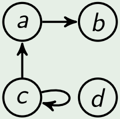
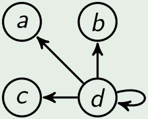
- Die Nachfolgermenge $T[x]$ von x bzgl. der Relation $T = R \circ S$ berechnet sich zu

$$T[x] = \bigcup_{y \in R[x]} S[y]$$

Das Relationenprodukt

Beispiel

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	$\begin{matrix}1&0&1&0\\0&0&0&0\\0&1&0&1\\0&0&0&0\end{matrix}$	$\begin{matrix}0&1&0&0\\0&0&0&0\\0&0&0&0\\1&0&1&0\end{matrix}$	$\begin{matrix}0&1&0&0\\0&0&0&0\\1&0&1&0\\0&0&0&0\end{matrix}$	$\begin{matrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\1&1&1&1\end{matrix}$
Adjazenz-listen	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

◁

Hüllenoperatoren

Frage

Welche Paare muss man zu einer Relation R mindestens hinzufügen, damit R transitiv wird?

Antwort

- Es ist klar, dass der Schnitt von transitiven Relationen wieder transitiv ist
- Die transitive Hülle von R ist

$$R^+ = \bigcap \{S \subseteq A \times A \mid S \text{ ist transitiv und } R \subseteq S\}$$

- R^+ ist also eine transitive Relation, die R enthält
- Da R^+ zudem in jeder Relation mit diesen Eigenschaften enthalten ist, gibt es keine transitive Relation mit weniger Paaren, die R enthält
- Da auch die Reflexivität und die Symmetrie bei der Schnittbildung erhalten bleiben, lassen sich nach demselben Muster weitere Hüllenoperatoren definieren

Weitere Hüllenoperatoren

Definition

Sei R eine Relation auf A

- Die reflexive Hülle von R ist

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

- Die symmetrische Hülle von R ist

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

- 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

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

Transitive und reflexive Hülle

Satz

$h_{\mathrm{refl}}(R) = R \cup Id_A, \quad h_{\mathrm{sym}}(R) = R \cup R^T, \quad R^+ = \bigcup_{n \geq 1} R^n, \quad R^* = \bigcup_{n \geq 0} R^n$

Beweis

Siehe Übungen. $\square$

Transitive und reflexive Hülle

Bemerkung. Sei $G = (V, E)$ ein Digraph.

- Ein Paar (a, b) ist genau dann in der reflexiv-transitiven Hülle E^* von E enthalten, wenn es ein $n \geq 0$ gibt mit $aE^n b$
- Dies bedeutet, dass es Elemente $x_0, \ldots, x_n \in V$ gibt mit
$$x_0 = a,\ x_n = b \text{ und } (x_{i-1}, x_i) \in E \text{ für } i = 1, \ldots, n$$
- $x_0, \ldots, x_n$ heißt Weg der Länge n von a nach b in G
- G heißt zusammenhängend, wenn es in G 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 in G von jedem Knoten a einen Weg zu jedem Knoten b gibt

Äquivalenzrelationen

Definition

(A, R) heißt Äquivalenzrelation, wenn R eine reflexive, symmetrische und transitive Relation auf A ist

Beispiel

- 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" $\triangleleft$

Äquivalenzklassen

Definition

- Ist E eine Äquivalenzrelation, so nennt man die Nachbarschaft $E[x]$ die von x repräsentierte Äquivalenzklasse und bezeichnet sie auch mit $[x]_E$ (oder einfach mit $[x]$, falls E aus dem Kontext ersichtlich ist):

$$[x]_E = [x] = E[x] = \{y \mid xEy\}$$

- Eine Menge $S \subseteq A$ heißt Repräsentantensystem, falls sie genau ein Element aus jeder Äquivalenzklasse enthält
- Die Menge aller Äquivalenzklassen von E wird Quotienten- oder Faktormenge von A bzgl. E genannt und mit A/E bezeichnet:

$$A/E = \{[x]_E \mid x \in A\}$$

- Die Anzahl $|A/E|$ der Äquivalenzklassen von E wird auch als der Index von E (kurz: $index(E)$) bezeichnet

Äquivalenzrelationen

Beispiel (Fortsetzung)

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
- Ein Repräsentantensystem wird beispielsweise durch die Menge aller Ursprungsgeraden gebildet
- 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
- Repräsentantensystem: $\{0, 1, \ldots, m-1\}$. $\triangleleft$

Äquivalenzrelationen

- Die kleinste Äquivalenzrelation auf A ist die Identität Id_A, die größte ist die Allrelation $A \times A$
- Die Äquivalenzklassen der Identität enthalten jeweils nur ein Element, d.h. $[x]_{Id_A} = \{x\}$ für alle $x \in A$
- Daher hat die Identität Id_A nur ein Repräsentantensystem, nämlich A
- Die Allrelation erzeugt dagegen nur eine Äquivalenzklasse, nämlich $[x]_{A \times A} = A$ für alle $x \in A$
- Daher kann als Repräsentantensystem für die Allrelation $A \times A$ jede Singletonmenge $\{x\} \subseteq A$ fungieren

Partition einer Menge

Wie wir sehen werden, bilden die Äquivalenzklassen eine Zerlegung von A

Definition

Eine Familie $\{B_i \mid i \in I\}$ von nichtleeren Teilmengen $B_i \subseteq A$ heißt Partition (oder Zerlegung) der Menge A, falls gilt:

- die Mengen B_i überdecken A, d.h. $A = \bigcup_{i \in I} B_i$ und
- die Mengen B_i sind paarweise disjunkt, d.h. für je zwei verschiedene Mengen $B_i \neq B_j$ gilt $B_i \cap B_j = \varnothing$

Verfeinerung und Vergröberung von Äquivalenzrelationen

- Seien E und E' zwei Äquivalenzrelationen auf einer Menge A
- Gilt $E \subseteq E'$, so 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
- Im Fall $E \subseteq E'$ sagt man auch, E bewirkt eine feinere Zerlegung von A als E'
- Demnach ist die Identität die feinste und die Allrelation die gröbste Äquivalenzrelation

Äquivalenzrelationen und Partitionen

Satz

Für eine Relation E auf A sind folgende Aussagen äquivalent:

1. E ist eine Äquivalenzrelation auf A
2. 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.

1 $\Rightarrow$ 2: Falls E eine Äquivalenzrelation auf A ist, bildet die Faktormenge $\{E[x] \mid x \in A\}$ eine Partition von A mit der gewünschten Eigenschaft:

- Da E reflexiv ist, gilt $x \in E[x]$ für alle $x \in A$, 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

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

Satz

Sei E eine Relation auf A. Dann sind folgende Aussagen äquivalent:

1. E ist eine Äquivalenzrelation auf A
2. 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.

2 $\Rightarrow$ 1: Hat A umgekehrt eine Partition $\{B_i \mid i \in I\}$ mit der Eigenschaft $xEy \Leftrightarrow \exists i \in I : x, y \in B_i$, so ist E

- reflexiv, da zu jedem $x \in A$ eine Menge B_i mit $x \in B_i$ existiert, was xEx impliziert
- symmetrisch, da aus $x, y \in B_i$ auch $y, x \in B_i$ folgt, was $xEy \Leftrightarrow yEx$ impliziert, 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, was $xEy \wedge yEz \Rightarrow xEz$ impliziert $\square$

Ordnungsrelationen

Definition

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

Beispiel

- $(\mathcal{P}(M), \subseteq)$, $(\mathbb{Z}, \leq)$, $(\mathbb{R}, \leq)$, $(\mathbb{N}, |)$ sind Ordnungen
- $(\mathbb{Z}, |)$ ist keine Ordnung, aber eine Quasiordnung
- 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 $\mathbb{N}$ $\triangleleft$

Darstellung einer Ordnung durch ein Hasse-Diagramm

- Sei $\leq$ eine Ordnung auf A und sei $<$ die Relation $\leq \setminus Id_A$, d.h.

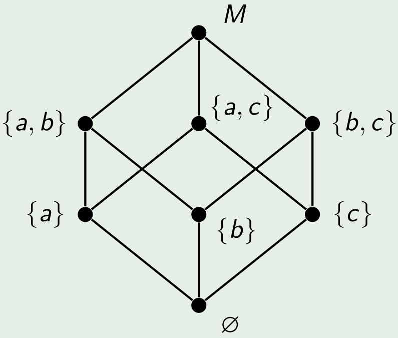
  $$x < y \;\Leftrightarrow\; x \leq y \wedge x \neq y$$

- Ein Element $x \in A$ heißt unterer Nachbar von y (kurz: $x \lessdot y$), falls $x < y$ gilt und kein $z \in A$ existiert mit $x < z < y$
- $\lessdot$ ist also die Relation $< \setminus <^2$
- Um die Ordnung $(A, \leq)$ in einem Hasse-Diagramm darzustellen, wird nur der Digraph der Relation $(A, \lessdot)$ gezeichnet
- Weiterhin wird im Fall $x \lessdot y$ der Knoten y oberhalb des Knotens x gezeichnet, so dass auf die Pfeilspitzen verzichtet werden kann

Das Hasse-Diagramm für $(\mathcal{P}(M); \subseteq)$

Beispiel

Die Inklusion $\subseteq$ auf $\mathcal{P}(M)$ mit $M = \{a, b, c\}$ lässt sich durch folgendes Hasse-Diagramm darstellen:

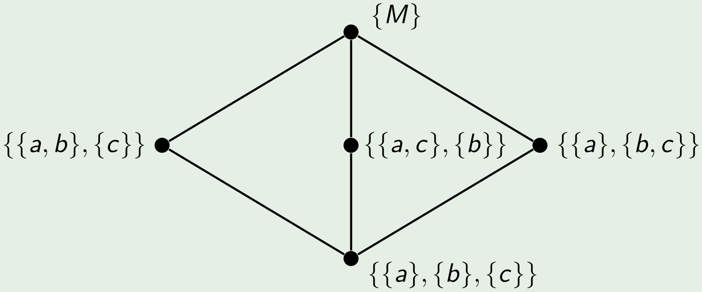


◁

Das Hasse-Diagramm der Feiner-Relation

Beispiel

Die "feiner als" Relation auf der Menge aller Partitionen von $M = \{a, b, c\}$ ist durch folgendes Hasse-Diagramm darstellbar:

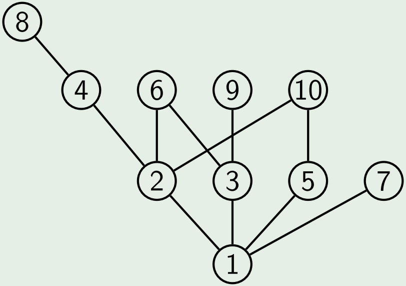


◁

Das Hasse-Diagramm der "teilt"-Relation

Beispiel

Die Einschränkung der "teilt"-Relation auf die Menge $\{1, 2, \ldots, 10\}$ ist durch folgendes Hasse-Diagramm darstellbar:



$\triangleleft$

Maximale, minimale, größte und kleinste Elemente

Definition

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 ($b = \min B$), falls gilt:

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

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

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

Wegen der Antisymmetrie kann es in B höchstens ein kleinstes und höchstens ein größtes Element geben