Einführung in die Theoretische Informatik

Johannes Köbler

Institut für Informatik
Humboldt-Universität zu Berlin

WS 2015/16

Vorlesung

Mo,Mi	15-17	RUD 26	0'115	Johannes Köbler

Übungen

Mo	09-11	RUD 26	0'313	Frank Fuhlbrück
Mo	09-11	RUD 26	1'303	Robert Prüfer
Mo	13-15	RUD 26	0'313	Frank Fuhlbrück
Di	11-13	RUD 26	1'306	Robert Prüfer
Mi	09-11	RUD 26	0'313	Robert Prüfer
Mi	09-11	RUD 26	1'303	Berit Grußien
Do	09-11	RUD 26	0'313	Wolfgang Kössler
Do	11-13	RUD 26	0'313	Wolfgang Kössler
Fr	09-11	RUD 26	1'305	Berit Grußien

Tutorien

Mo	13-15	RUD 26	0'310	Michael Jung
Mi	13-15	RUD 26	0'310	Michael Jung
Mi	17-19	RUD 26	0'307	Michael Jung
Fr	15-17	RUD 26	1'303	Michael Jung

Weitere Infos unter

- **u.hu-berlin.de/ethi15**

bzw.

- **www.informatik.hu-berlin.de/forschung/gebiete/algorithmenII/Lehre/ws15/einftheo**

Übungen (Anmeldung über GOYA erforderlich)

Ausgabe der Aufgabenblätter

- in der VL, auf GOYA und der VL-Webseite

Bearbeitung

- in Gruppen von zwei bis drei Teilnehmern
- Teilnehmer müssen nicht in der gleichen Übungsgruppe sein
- bitte Übungsgruppe, Namen und Matrikelnr. angeben
- bitte jede Aufgabe auf einem separaten Blatt bearbeiten

Abgabe (bitte nur in Papierform)

- bis 15:10 Uhr hier im Hörsaal (Abgabetermin s. Aufgabenblatt)

Rückgabe

- in den Übungsgruppen

Schein, Klausur, Skript

Scheinkriterien

- Lösen von $\geq 50\%$ der schriftlichen Aufgaben,
- Erfolgreiches Vorrechnen von ≥ 2 mündl. Aufgaben.

Klausur

- **Termin: 19.02.2016**
- Zulassung nur mit Übungsschein
- Nachklausur: 22.03.2016

Skript

- wird wöchentlich ins Netz gestellt.

Gibt es zum organisatorischen Ablauf noch Fragen?

Inhalt der Vorlesung

Themen dieser VL:

- **Welche Rechenmodelle sind adäquat?** **Automatentheorie**
- **Welche Probleme sind lösbar?** **Berechenbarkeitstheorie**
- **Welcher Aufwand ist 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

Maschinenmodelle

- Rechenmaschinen spielen in der Informatik eine zentrale Rolle.
- Es gibt viele unterschiedliche math. Modelle.
- Diese können sich in der 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 größten gemeinsamen Teilers zweier natürlicher Zahlen (300 v. Chr.).
- Von einem Algorithmus wird erwartet, dass er bei jeder zulässigen Problemeingabe nach endlich vielen Rechenschritten eine korrekte Ausgabe liefert.
- Problemeingaben können Zahlen, Formeln, Graphen etc. sein.
- Diese werden über einem Eingabealphabet Σ kodiert.

Alphabet, Wort, Sprache

Definition

- Ein Alphabet ist eine geordnete endliche Menge

  $$\Sigma = \{a_1, \ldots, a_m\}, \ m \geq 1$$

  von Zeichen a_i.
- Eine Folge $x = x_1 \ldots x_n \in \Sigma^n$ heißt Wort (der Länge n).
- Die Menge aller Wörter über Σ ist

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

- 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

Sei Σ ein Alphabet.

- Dann sind $\varnothing, \Sigma^*, \Sigma$ und $\{\varepsilon\}$ Sprachen über Σ.
- $\varnothing$ 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.
  Solche Sprachen werden auch als Singleton-Sprachen bezeichnet.

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

Seien $x = x_1 \dots x_n$ und $y = y_1 \dots y_m$ Wörter. Dann wird das Wort $x \circ y = x_1 \dots x_n y_1 \dots y_m$ als Konkatenation von x und y bezeichnet.
Für $x \circ y$ schreiben wir auch einfach xy.

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

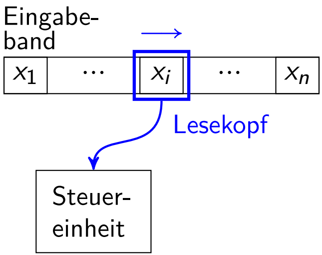
Neben den Mengenoperationen Schnitt, Vereinigung und Komplement gibt es auch spezielle Sprachoperationen.

Definition

- Das Produkt (Verkettung, Konkatenation) 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 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 von A ist $A^* = \bigcup_{n \geq 0} A^n$.
- Die Plushülle von 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 M akzeptierte oder erkannte Sprache ist

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

- Eine Zustandsfolge $q_0, q_1, \ldots, q_n$ heißt Rechnung von $M(x_1 \ldots x_n)$, falls $\delta(q_i, x_{i+1}) = q_{i+1}$ für $i = 0, \ldots, n-1$ gilt.
- Sie heißt akzeptierend, falls $q_n \in E$ ist.

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

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

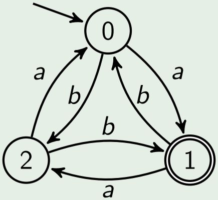
DFAs beherrschen Modulare Arithmetik

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?

- $w_1 = aba$? Ja (Rechnung: $0, 1, 0, 1$).
- $w_2 = abba$? Nein (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 $x = x_1 \ldots x_n \in \Sigma^*$. Welchen Zustand erreicht M bei Eingabe x nach i Schritten?

Antwort

- nach 0 Schritten: q_0,
- nach 1 Schritt: $\delta(q_0, x_1)$,
- nach 2 Schritten: $\delta(\delta(q_0, x_1), x_2)$,
- nach i Schritten: $\delta(\ldots\delta(\delta(q_0, x_1), x_2), \ldots x_i)$.

Definition

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

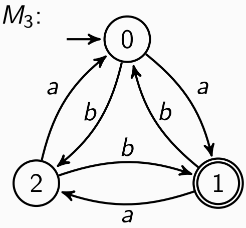
$$\begin{aligned} \hat{\delta}(q, \varepsilon) &= q, \\ \hat{\delta}(q, xa) &= \delta(\hat{\delta}(q, x), a). \end{aligned}$$

- Die von M erkannte Sprache lässt sich nun auch in der Form

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

  schreiben.

DFAs beherrschen Modulare Arithmetik



Behauptung

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

Beweis

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

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

- Folglich reicht es, folgende Kongruenzgleichung zu zeigen:

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

DFAs beherrschen Modulare Arithmetik

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

Wir führen Induktion über die Länge n von x.

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

Induktionsschritt $n \rightsquigarrow n+1$:

- Sei $x = x_1 \dots x_{n+1}$ gegeben und sei $i = \hat{\delta}(0, x_1 \dots x_n)$.
- Nach IV gilt $i \equiv_3 \#_a(x_1 \dots x_n) - \#_b(x_1 \dots x_n)$.
- 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

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

$\square$

Singletons sind regulär

Vereinbarung

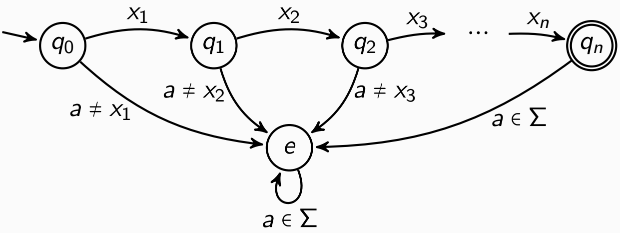
Für das Folgende sei $\Sigma = \{a_1, \ldots, a_m\}$ ein fest gewähltes Alphabet.

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 $L(M) = \{x\}$:



$\square$

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.

$\square$

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 aus?

Antwort

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

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. $\triangleleft$

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 für das Produkt $L_1 L_2$ bereitet es Schwierigkeiten, den richtigen Zeitpunkt für den Übergang von (der Simulation von) M_1 zu M_2 zu finden.

Lösungsidee

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

Verbleibendes Problem

Zeige, dass auch NFAs nur reguläre Sprachen erkennen.

Nichtdeterministische endliche Automaten

Definition

- Ein nichtdet. endl. Automat (kurz: NFA; *Nondet. Finite Automaton*)

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

  ist genau so aufgebaut wie ein DFA, nur dass er
  - eine Menge $Q_0 \subseteq Z$ von Startzuständen hat und
  - die Überführungsfunktion folgende Form hat:

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

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

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

- Eine Zustandsfolge $q_0, q_1, \dots, q_n$ heißt Rechnung von $N(x_1 \dots x_n)$, falls $q_0 \in Q_0$ 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

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

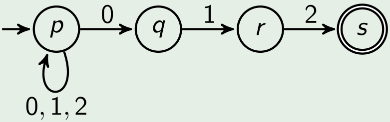
  nicht verarbeiten kann.

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:



- Dann ist $L(M) = \{x012 \mid x \in \Sigma^*\}$ die Sprache aller Wörter, die mit dem Suffix 012 enden. $\triangleleft$

Eigenschaften von NFAs

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

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

und

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

Eigenschaften von NFAs

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

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

  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 $x = x_1 \cdots x_k \in L_1$, $y = y_1 \cdots y_l \in L_2$ und seien $q_0, \ldots, q_k$ und $p_0, \ldots, p_l$ akzeptierende Rechnungen von $N_1(x)$ und $N_2(y)$.
- Dann gilt $q_0 \in Q_1$, $q_k \in E_1$ und $p_0 \in Q_2$, $p_l \in E_2$.
- Im Fall $l \geq 1$ ist zudem $p_1 \in \Delta_2(p_0, y_1)$ und somit $p_1 \in \Delta(q_k, y_1)$.
- Im Fall $l = 0$ ist zudem $p_l \in Q_2 \cap E_2$ und somit $q_k \in E$.
- Also ist $q_0, \ldots, q_k, p_1, \ldots, p_l$ eine akzeptierende Rechnung von $N(xy)$.

Eigenschaften von NFAs

- Dann gilt $L(N) = L_1L_2$ für den NFA $N = (Z_1 \cup Z_2, \Sigma, \Delta, Q_1, E)$ mit

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

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

Beweis von $L(N) \subseteq L_1L_2$:

- Sei $x = x_1 \cdots x_n \in L(N)$ und sei $q_0, \ldots, q_n$ eine akz. Rechnung von $N(x)$.
- Dann gilt $q_0 \in Q_1$, $q_n \in E$, $q_0, \ldots, q_i \in Z_1$ und $q_{i+1}, \ldots, q_n \in Z_2$ für ein i.
- Im Fall $i = n$ ist $q_n \in E_1$ (d.h. $x \in L_1$) und $Q_2 \cap E_2 \neq \varnothing$ (d.h. $\varepsilon \in L_2$).
- 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.
- Also ist $q_0, \ldots, q_i$ eine akz. Rechnung von $N_1(x_1 \cdots x_i)$ und $q, q_{i+1}, \ldots, q_n$ eine akz. Rechnung von $N_2(x_{i+1} \cdots x_n)$, d.h. $x \in L_1L_2$. $\square$

Eigenschaften von NFAs

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 $L_1 L_2$ 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 $M = (Z, \Sigma, \delta, q_0, E)$ leicht 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. □

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.
- in einem Schritt: alle Zustände in

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

- in 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

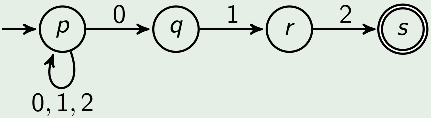
  $$\delta(Q, a) = \bigcup_{q \in Q} \Delta(q, a)$$

  aller Zustände, in die N gelangen kann, wenn N ausgehend von einem beliebigen Zustand $q \in Q$ das Zeichen a liest.
- M wird auch als der zu N gehörige Potenzmengenautomat bezeichnet.

Simulation von NFAs durch DFAs

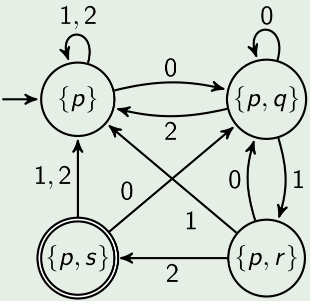
Beispiel

- Betrachte den NFA N



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

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



◁

Bemerkung

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

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

  Zustände benötigt, da die übrigen 12 Zustände nicht erreichbar sind.
- 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 leicht 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

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

$\square$

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.

$n - 1 \rightsquigarrow n$: Sei $x = x_1 \ldots x_n$ gegeben. Nach IV enthält

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

die Zustände, die N nach Lesen von $x_1 \ldots x_{n-1}$ 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 $\hat{\delta}(Q_0, x)$ die Zustände, die N nach Lesen von x erreichen kann. $\square$

Abschlusseigenschaften der Klasse REG

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 endl. 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 $\varnothing$, ϵ und a $(a \in \Sigma)$ sind reguläre Ausdrücke über Σ, die

- die leere Sprache $L(\varnothing) = \varnothing$,
- 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 $\Gamma = \Sigma \cup \{\varnothing, \epsilon, |, ^*, (,)\}$.

Reguläre Ausdrücke

Beispiel

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

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

$\triangleleft$

Vereinbarungen

- Um Klammern zu sparen, definieren wir folgende Präzedenzordnung: Der Sternoperator * bindet stärker als der Produktoperator und dieser wiederum stärker als der Vereinigungsoperator.
- 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 $\gamma\gamma^*$.

Satz

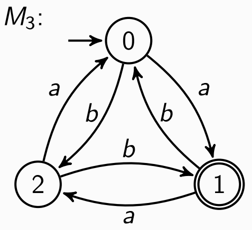
$\{L(\gamma) \mid \gamma$ ist ein regulärer Ausdruck über $\Sigma\} \subseteq \mathrm{REG}$.

Beweis.

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



Frage

Wie lässt sich die Sprache

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

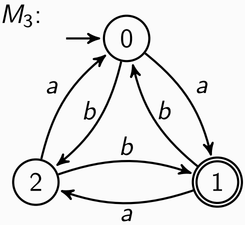
durch einen regulären Ausdruck beschreiben?

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 $L^{\neq r}_{p,q}$ die Sprache aller Wörter $x = x_1 \cdots x_n \in L_{p,q}$, die hierzu nur Zustände ungleich r benutzen (d.h. $\hat{\delta}(p, x_1 \cdots x_i) \neq r$ für $i = 1, \dots, n-1$).
- Dann gilt $L(M_3) = L_{0,1} = L_{0,0} L^{\neq 0}_{0,1}$ und $L_{0,0} = (L^{\neq 0}_{0,0})^*$, also $L(M_3) = (L^{\neq 0}_{0,0})^* L^{\neq 0}_{0,1}$.

Charakterisierung von REG durch reguläre Ausdrücke

Antwort (Fortsetzung)



- Dann gilt $L(M_3) = (L_{0,0}^{\neq 0})^* L_{0,1}^{\neq 0}$.
- $L_{0,1}^{\neq 0}$ 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 $\subseteq \{L(\gamma) \mid \gamma$ ist ein regulärer Ausdruck$\}$.

Beweis

- 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 $Z = \{1, \ldots, m\}$ 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.
- Es reicht also, reguläre Ausdrücke für die Sprachen $L_{p,q}$ mit $1 \leq p, q \leq m$ anzugeben.

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

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

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

Satz

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

Beweis (Schluss)

- Wir zeigen induktiv über r, dass die Sprachen $L^r_{p,q}$ durch reguläre Ausdrücke beschreibbar sind.

  $r = 0$: 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 und somit durch reguläre Ausdrücke beschreibbar.

  $r \rightsquigarrow r+1$: Wegen

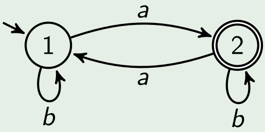
  $$L^{r+1}_{p,q} = L^r_{p,q} \cup L^r_{p,r+1}(L^r_{r+1,r+1})^* L^r_{r+1,q}$$

  sind mit $L^r_{p,q}$, $1 \leq p,q \leq m$, auch die Sprachen $L^{r+1}_{p,q}$, $1 \leq p,q \leq m$, durch reguläre Ausdrücke beschreibbar. □

Charakterisierung von REG durch reguläre Ausdrücke

Beispiel

- Betrachte den DFA M



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

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

Charakterisierung von REG durch reguläre Ausdrücke

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

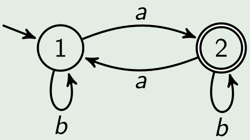
$$\begin{aligned}
\gamma^2_{1,2} &= \gamma^1_{1,2} | \gamma^1_{1,2} (\gamma^1_{2,2})^* \gamma^1_{2,2}, \\
\gamma^1_{1,2} &= \gamma^0_{1,2} | \gamma^0_{1,1} (\gamma^0_{1,1})^* \gamma^0_{1,2}, \\
\gamma^1_{2,2} &= \gamma^0_{2,2} | \gamma^0_{2,1} (\gamma^0_{1,1})^* \gamma^0_{1,2}.
\end{aligned}$$

- Es genügt also, die regulären Ausdrücke $\gamma^0_{1,1}$, $\gamma^0_{1,2}$, $\gamma^0_{2,1}$, $\gamma^0_{2,2}$, $\gamma^1_{1,2}$, $\gamma^1_{2,2}$ und $\gamma^2_{1,2}$ zu berechnen.

Charakterisierung von REG durch reguläre Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformeln

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

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

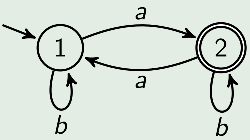
$$L^{r+1}_{p,q} = L^r_{p,q} \cup L^r_{p,r+1}(L^r_{r+1,r+1})^* L^r_{r+1,q}.$$

r	p,q			
	1, 1	1, 2	2, 1	2, 2
0				
1				
2				

Charakterisierung von REG durch reguläre Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformeln

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

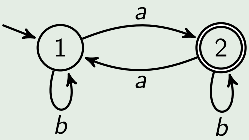
$$\rightsquigarrow \gamma^0_{1,1} = \epsilon|b$$

r	p,q			
	1, 1	1, 2	2, 1	2, 2
0	$\epsilon\|b$			
1				
2				

Charakterisierung von REG durch reguläre Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformeln

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

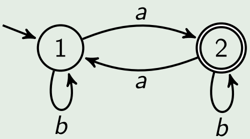
$$\rightsquigarrow \quad \gamma^0_{1,2} = a$$

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\epsilon \vert b$	a		
1				
2				

◁

Beispiel (Fortsetzung)

DFA M



Rekursionsformeln

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

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

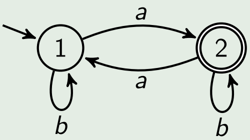
r	p, q: 1, 1	1, 2	2, 1	2, 2
0	$\epsilon \mid b$	a	a	
1				
2				

◁

Charakterisierung von REG durch reguläre Ausdrücke

Beispiel (Fortsetzung)

DFA M



Rekursionsformeln

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

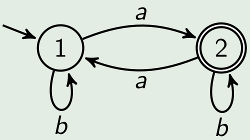
$$\rightsquigarrow \quad \gamma^0_{2,2} = \epsilon|b$$

r	p,q			
	1, 1	1, 2	2, 1	2, 2
0	$\epsilon\|b$	a	a	$\epsilon\|b$
1	-			
2				

Charakterisierung von REG durch reguläre Ausdrücke

Beispiel (Fortsetzung)

DFA M



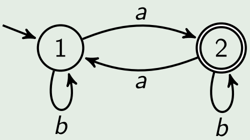
Rekursionsformeln

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

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

Beispiel (Fortsetzung)

DFA M



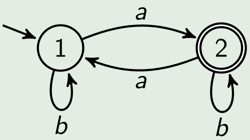
Rekursionsformeln

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

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	-			

Beispiel (Fortsetzung)

DFA M



Rekursionsformeln

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

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

Sei L eine Sprache. Dann sind folgende Aussagen äquivalent:

- L ist regulär,
- 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.

Bemerkung

- 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

- (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. Dabei ist $R^0 = Id$.

Vorsicht!

Das Relationenprodukt R^n sollte 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	✓	✓	✓				
(Halb-)Ordnung	✓		✓	✓			
Striktordnung			✓		✓		
lineare Ordnung			✓	✓		✓	
lin. Striktord.			✓		✓		✓
Quasiordnung	✓		✓				

Bemerkung

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

Eigenschaften von Relationen

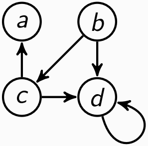
Beispiel

- 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

$\triangleleft$

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

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

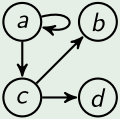
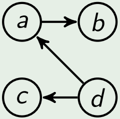
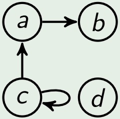
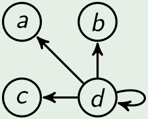
- Die Nachfolgermenge $T[x]$ von x bzgl. der Relation $T = R \circ S$ berechnet sich zu

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

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 leicht zu sehen, 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

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

Beweis

Siehe Übungen. □

Bemerkung

- Ein Paar (a, b) ist also genau dann in der reflexiv-transitiven Hülle R^* von R enthalten, wenn es ein $n \geq 0$ gibt mit $aR^n b$.
- Dies bedeutet, dass es Elemente $x_0, \ldots, x_n \in A$ gibt mit

$$x_0 = a,\ x_n = b \ \text{ und } \ x_0 R x_1 R x_2 \ldots x_{n-1} R x_n.$$

- $x_0, \ldots, x_n$ heißt Weg der Länge n von a nach b.

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". $\lhd$

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 bezeichnet.

Äquivalenzrelationen

Beispiel

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: $\{0, 1, \ldots, m-1\}$. ◁

Bemerkungen

- 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 $x \in A$.
- Die Allrelation erzeugt dagegen nur eine Äquivalenzklasse, nämlich $[x]_{A \times A} = A$ für alle $x \in A$.
- 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.

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 $B_i \cap B_j = \emptyset$.

Verfeinerung und Vergröberung von Äquivalenzrelationen

Bemerkungen

- 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.
- 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

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

1. E ist eine Äquivalenzrelation auf A,
2. Für alle $x, y \in A$ gilt $xEy \Leftrightarrow E[x] = E[y]$,
3. 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 impliziert 2: Sei E eine Äquivalenzrelation auf A.
Da E transitiv ist, impliziert xEy die Inklusion $E[y] \subseteq E[x]$:

$$z \in E[y] \Rightarrow yEz \overset{xEy}{\Rightarrow} xEz \Rightarrow z \in E[x].$$

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

Umgekehrt folgt aus $E[x] = E[y]$ wegen der Reflexivität von E, dass $y \in E[y] = E[x]$ enthalten ist, und somit xEy.

Äquivalenzrelationen und Partitionen

Satz

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

1. E ist eine Äquivalenzrelation auf A,
2. Für alle $x, y \in A$ gilt $xEy \Leftrightarrow E[x] = E[y]$,
3. 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 impliziert 3: Gelte $xEy \Leftrightarrow E[x] = E[y]$ für alle $x, y \in A$.

- Wegen $E[x] = E[x]$ folgt xEx und somit $x \in E[x]$, d.h. $A = \bigcup_{x \in A} E[x]$.
- Ist $E[x] \cap E[y] \neq \emptyset$ und z ein Element in $E[x] \cap E[y]$, so folgt xEz und yEz und somit $E[x] = E[z] = E[y]$.
- Folglich bildet $\{E[x] \mid x \in A\}$ eine Partition von A.
- Zudem gilt

$$xEy \Leftrightarrow E[x] = E[y] \Leftrightarrow \exists z : E[x] = E[z] = E[y] \Leftrightarrow \exists z : x, y \in E[z].$$

Äquivalenzrelationen und Partitionen

Satz

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

1. E ist eine Äquivalenzrelation auf A,
2. Für alle $x, y \in A$ gilt $xEy \Leftrightarrow E[x] = E[y]$,
3. 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.

3 impliziert 1: Existiert schließlich 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. $\square$

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.

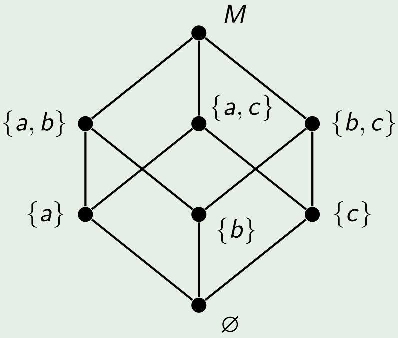
  $$x < y \;\Leftrightarrow\; x \leq y \wedge x \neq y$$

- 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 vom Knoten 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:

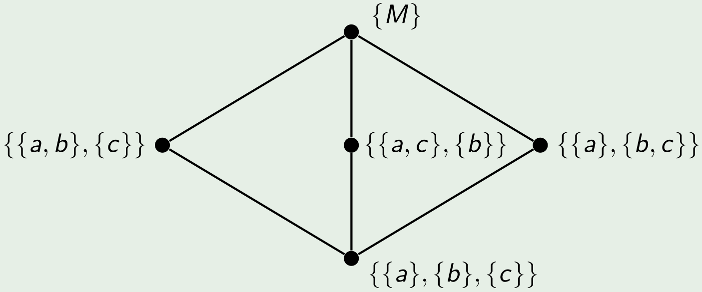


◁

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:

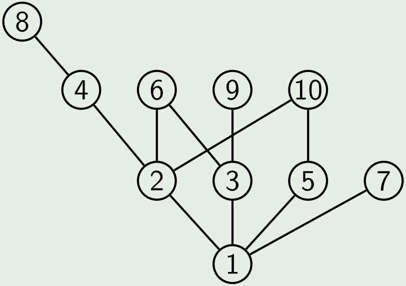


$\triangleleft$

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:



◁

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, falls gilt:

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

- b heißt größtes Element oder Maximum von 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:

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

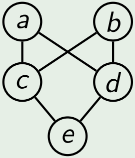
Bemerkung

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

Maximale, minimale, größte und kleinste Elemente

Beispiel

Betrachte folgende Ordnung.



B	minimal in B	maximal in B	Minimum von B	Maximum von B
$\{a, b\}$	a, b	a, b	-	-
$\{c, d\}$	c, d	c, d	-	-
$\{a, b, c\}$	c	a, b	c	-
$\{a, b, c, e\}$	e	a, b	e	-
$\{a, c, d, e\}$	e	a	e	a

Obere und untere Schranken

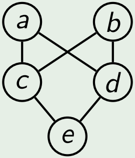
Definition

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

- Ein Element $u \in A$ mit $u \leq b$ für alle $b \in B$ heißt untere Schranke von B.
- Ein Element $o \in A$ 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.
- B heißt nach unten beschränkt, wenn B eine untere Schranke hat.
- B heißt beschränkt, wenn B nach oben und nach unten beschränkt ist.

Obere und untere Schranken

Beispiel (Fortsetzung)



B	minimal	maximal	min	max	untere Schranken	obere Schranken
$\{a, b\}$	a, b	a, b	-	-	c, d, e	-
$\{c, d\}$	c, d	c, d	-	-	e	a, b
$\{a, b, c\}$	c	a, b	c	-	c, e	-
$\{a, b, c, e\}$	e	a, b	e	-	e	-
$\{a, c, d, e\}$	e	a	e	a	e	a

Infima und Suprema

Definition

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

- 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 ($i = \inf B$):

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

- 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 ($s = \sup B$):

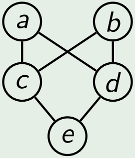
$$(\forall b \in B : b \leq s) \wedge [\forall o \in A : (\forall b \in B : b \leq o) \Rightarrow s \leq o]$$

Bemerkung

B kann nicht mehr als ein Supremum und ein Infimum haben.

Infima und Suprema

Beispiel (Schluss)



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

◁

Existenz von Infima und Suprema in linearen Ordnungen

Bemerkung

- 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

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

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

Definition

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

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

Abbildungen

Abbildungen ordnen jedem Element ihres Definitionsbereichs genau ein Element zu.

Definition

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

- 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

Sei $f : A \to B$ eine Abbildung.

- 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.

Bemerkung

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.

Homomorphismen

Definition

Seien (A_1, R_1) und (A_2, R_2) Relationalstrukturen.

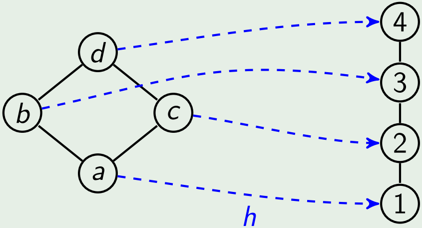
- Eine Abbildung $h : A_1 \to A_2$ heißt Homomorphismus, falls für alle $a, b \in A_1$ gilt:

$$aR_1 b \Rightarrow h(a) R_2 h(b).$$

- Sind (A_1, R_1) und (A_2, R_2) Ordnungen, so spricht man auch von Ordnungshomomorphismen oder einfach von monotonen Abbildungen.
- Injektive Ordnungshomomorphismen werden auch streng monotone Abbildungen genannt.

Homomorphismen

Beispiel



$(A, \leq)$ $(B, \sqsubseteq)$

- Die Abbildung $h : A \to B$ ist ein bijektiver Ordnungshomomorphismus.
- Die Umkehrabbildung h^{-1} ist jedoch kein Homomorphismus, da h^{-1} nicht monoton ist.
- Es gilt nämlich $2 \sqsubseteq 3$, aber $h^{-1}(2) = b \nleq c = h^{-1}(3)$. $\triangleleft$

Isomorphismen

Definition

- Seien (A_1, R_1) und (A_2, R_2) Relationalstrukturen.
- Ein bijektiver Homomorphismus $h : A_1 \to A_2$, bei dem auch h^{-1} ein Homomorphismus ist, d.h. es gilt für alle $a, b \in A_1$,

  $$aR_1 b \Leftrightarrow h(a) R_2 h(b).$$

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

Sind (A_1, R_1) und (A_2, R_2) isomorph, so bedeutet dies, dass sich die beiden Strukturen nur in der Benennung ihrer Elemente unterscheiden.

Beispiel

- Die Bijektion $h: x \mapsto e^x$ ist ein Ordnungsisomorphismus zwischen $(\mathbb{R}, \leq)$ und $(\mathbb{R}^+, \leq)$.
- Für $n \in \mathbb{N}$ sei

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

  und

  $$P_n = \{p \in T_n \mid p \text{ ist prim}\}.$$

  Dann ist die Abbildung

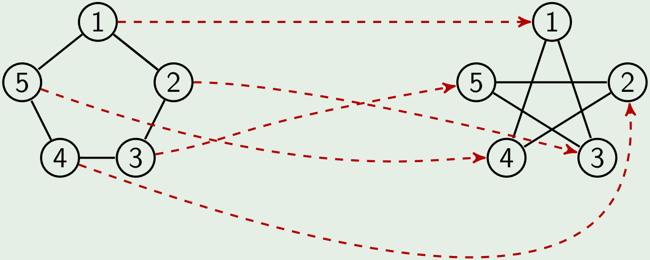
  $$h: k \mapsto P_k$$

  ein Ordnungshomomorphismus von $(T_n, |)$ auf $(\mathcal{P}(P_n), \subseteq)$.

  h ist sogar ein Isomorphismus, falls n quadratfrei ist (d.h. es gibt keine Primzahl p, so dass p^2 die Zahl n teilt). $\triangleleft$

Isomorphismen

Beispiel



$G = (V, E)$

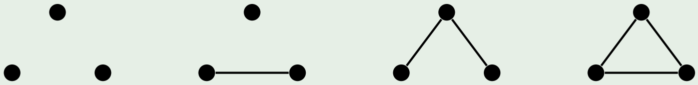
v	1 2 3 4 5
$h_1(v)$	1 3 5 2 4
$h_2(v)$	1 4 2 5 3

$G' = (V, E')$

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

Beispiel

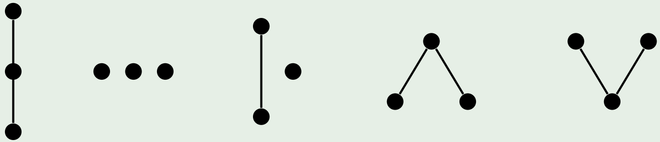
- Während auf der Knotenmenge $V = \{1, 2, 3\}$ insgesamt $2^{\binom{3}{2}} = 2^3 = 8$ verschiedene Graphen existieren, gibt es auf dieser Menge nur 4 verschiedene nichtisomorphe Graphen:



$\triangleleft$

Beispiel

- Es existieren genau 5 nichtisomorphe Ordnungen mit 3 Elementen:



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

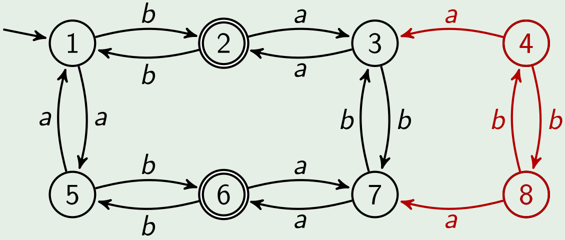
Minimierung von DFAs

Frage

Wie können wir feststellen, ob ein DFA $M = (Z, \Sigma, \delta, q_0, E)$ eine minimale Anzahl von Zuständen besitzt (und Z evtl. verkleinern)?

Beispiel

- Betrachte den DFA M



- Zunächst können alle vom Startzustand aus unerreichbaren Zustände entfernt werden. ◁

Minimierung von DFAs

Frage

Wie können wir feststellen, ob ein DFA $M = (Z, \Sigma, \delta, q_0, E)$ eine minimale Anzahl von Zuständen besitzt (und Z evtl. verkleinern)?

Antwort

- Zunächst können alle vom Startzustand aus unerreichbaren Zustände entfernt werden.
- Zudem lassen sich zwei Zustände p und q verschmelzen, wenn M von p und q aus jeweils dieselben Wörter akzeptiert.
- Für $z \in Z$ sei

$$M_z = (Z, \Sigma, \delta, z, E).$$

- Dann können wir p und q verschmelzen (in Zeichen: $p \sim q$), wenn $L(M_p) = L(M_q)$ ist.
- Offensichtlich ist $\sim$ eine Äquivalenzrelation auf Z.

Minimierung von DFAs

Idee

Verschmelze jeden Zustand z mit allen äquivalenten Zuständen $z' \sim z$ zu einem neuen Zustand.

Notation

- Für die durch z repräsentierte Äquivalenzklasse

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

  schreiben wir auch einfach $[z]$ oder $\tilde{z}$.
- Für eine Teilmenge $Q \subseteq Z$ bezeichne

  $$\tilde{Q} = \{\tilde{q} \mid q \in Q\}$$

  die Menge aller Äquivalenzklassen $\tilde{q}$, die mind. ein $q \in Q$ enthalten.
- Dann führt obige Idee auf folgenden DFA:

  $$M' = (\tilde{Z}, \Sigma, \delta', \tilde{q}_0, \tilde{E}) \quad \text{mit} \quad \delta'(\tilde{q}, a) = \widetilde{\delta(q, a)}.$$

Wie können wir M' aus M berechnen?

- Es genügt, alle Paare $p \not\sim q$ von inäquivalenten Zuständen zu finden.
- Sei $A \Delta B = (A \setminus B) \cup (B \setminus A)$ die symmetrische Differenz von A und B.
- Dann gilt $p \not\sim q \Leftrightarrow L(M_p) \Delta L(M_q) \neq \varnothing$.
- Wörter $x \in L(M_p) \Delta L(M_q)$ heißen 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 enthält $D = \bigcup_{i \geq 0} D_i$ alle inäquivalenten Paare $\{p, q\}$, d.h.

  $$p \sim q \Leftrightarrow \{p, q\} \notin D.$$

.

Algorithmische Konstruktion von M'

- Offenbar unterscheidet ε Endzustände und Nichtendzustände, d.h.

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

- Zudem haben p und q genau dann einen Unterscheider ax der Länge $i+1$, wenn x die 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

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

Algorithmus zur Berechnung eines minimalen DFA

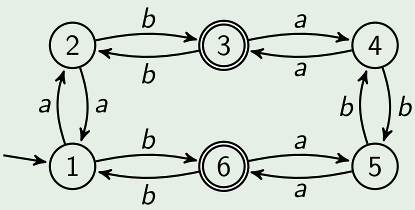
Algorithmus `min-DFA`(M)

Input: DFA $M = (Z, \Sigma, \delta, q_0, E)$
    entferne alle nicht erreichbaren Zustände
    $D' := \{\{z, z'\} \subseteq Z \mid z \in E, z' \notin E\}$
    repeat
        $D := D'$
        $D' := D \cup \{\{p, q\} \mid \exists a \in \Sigma : \{\delta(p, a), \delta(q, a)\} \in D\}$
    until $D' = D$
Output: $M' = (\tilde{Z}, \Sigma, \delta', \tilde{q}_0, \tilde{E})$, wobei für jeden Zustand $z \in Z$ gilt: $\tilde{z} = \{z' \in Z \mid \{z, z'\} \notin D\}$

Algorithmus für die Konstruktion von M'

Beispiel

Betrachte den DFA M



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

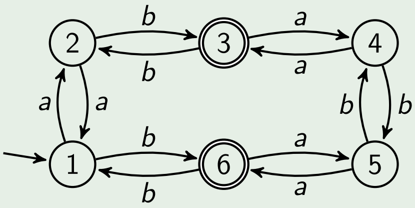
Dann enthält D_0 die Paare

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

Algorithmus für die Konstruktion von M'

Beispiel

Betrachte den DFA M



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

Wegen

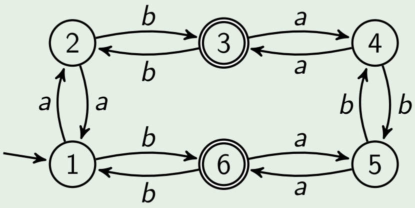
$\{p, q\}$	$\{1, 4\}$	$\{1, 5\}$	$\{2, 4\}$	$\{2, 5\}$
$\{\delta(q, a), \delta(p, a)\}$	$\{2, 3\}$	$\{2, 6\}$	$\{1, 3\}$	$\{1, 6\}$

enthält D_1 zusätzlich die Paare $\{1, 4\}$, $\{1, 5\}$, $\{2, 4\}$, $\{2, 5\}$.

Algorithmus für die Konstruktion von M'

Beispiel

Betrachte den DFA M



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

Da nun jedoch keines der verbliebenen Paare $\{1,2\}$, $\{3,6\}$, $\{4,5\}$ wegen

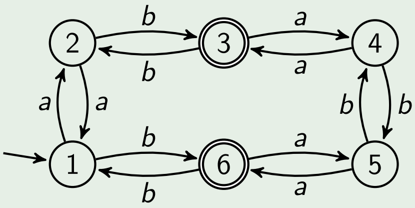
$\{p,q\}$	$\{1,2\}$	$\{3,6\}$	$\{4,5\}$
$\{\delta(p,a),\delta(q,a)\}$	$\{1,2\}$	$\{4,5\}$	$\{3,6\}$
$\{\delta(p,b),\delta(q,b)\}$	$\{3,6\}$	$\{1,2\}$	$\{4,5\}$

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

Algorithmus für die Konstruktion von M'

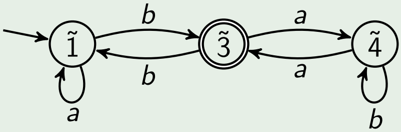
Beispiel

Betrachte den DFA M



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

Da die Paare $\{1,2\}$, $\{3,6\}$ und $\{4,5\}$ nicht in D enthalten sind, können die Zustände 1 und 2, 3 und 6, sowie 4 und 5 verschmolzen werden. Demnach hat M' die Zustände $\tilde{1} = \{1,2\}$, $\tilde{3} = \{3,6\}$ und $\tilde{4} = \{4,5\}$:



◁

Minimierung von DFAs

Satz

Sei $M = (Z, \Sigma, \delta, q_0, E)$ ein DFA ohne unerreichbare Zustände. Dann ist

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

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

Beweis

- Zuerst müssen wir zeigen, dass δ' wohldefiniert ist, also $\delta'(\tilde{q}, a)$ nicht von der Wahl des Repräsentanten q für die Äquivalenzklasse $\tilde{q}$ abhängt.
- Hierzu ist die Implikation $p \sim q \Rightarrow \delta(p, a) \sim \delta(q, a)$ zu zeigen.
- Diese ist wiederum äquivalent zur Kontraposition $\delta(p, a) \not\sim \delta(q, a) \Rightarrow p \not\sim q$, die wir bereits gezeigt haben.

Minimierung von DFAs

Satz

Sei $M = (Z, \Sigma, \delta, q_0, E)$ ein DFA ohne unerreichbare Zustände. Dann ist

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

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

Beweis (Fortsetzung)

- Als nächstes zeigen wir, dass $L(M') = L(M)$ ist.
- Sei $x = x_1 \dots x_n \in \Sigma^*$ eine Eingabe und seien $q_0, q_1, \dots, q_n$ die von $M(x)$ durchlaufenen Zustände, d.h. es gilt $\delta(q_{i-1}, x_i) = q_i$ für $i = 1, \dots, n$.
- Nach Definition von δ' folgt daher $\delta'(\tilde{q}_{i-1}, x_i) = \tilde{q}_i$ für $i = 1, \dots, n$, d.h. M' durchläuft bei Eingabe x die Zustände $\tilde{q}_0, \tilde{q}_1, \dots, \tilde{q}_n$.
- Da aber $\tilde{q}_n$ entweder nur End- oder nur Nicht-Endzustände enthält, gehört q_n genau dann zu E, wenn $\tilde{q}_n \in \tilde{E}$, d.h. es gilt

$$x \in L(M) \Leftrightarrow x \in L(M').$$

Minimierung von DFAs

Satz

Sei $M = (Z, \Sigma, \delta, q_0, E)$ ein DFA ohne unerreichbare Zustände. Dann ist

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

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

Beweis (Schluss)

- Noch z.z.: die Anzahl $\|\tilde{Z}\|$ der Zustände von M' ist minimal.
- Wegen $\|\tilde{Z}\| \le \|Z\|$ ist M' sicher dann minimal, wenn M minimal ist.
- Es reicht also zu zeigen, dass $\|\tilde{Z}\|$ nur von der Sprache $L(M)$ abhängt.
- Wegen $p \sim q \Leftrightarrow L(M_p) = L(M_q)$ gilt $\|\tilde{Z}\| = \|\{L(M_z) \mid z \in Z\}\|$.
- Sei $L = L(M)$ und für $x \in \Sigma^*$ sei L_x die Sprache $\{y \in \Sigma^* \mid xy \in L\}$.
- Dann gilt $\{L_x \mid x \in \Sigma^*\} = \{L(M_z) \mid z \in Z\}$:

  $\subseteq$: Klar, da $L_x = L(M_z)$ für $z = \hat{\delta}(q_0, x)$ ist.

  $\supseteq$: Auch klar, da jedes $z \in Z$ über ein $x \in \Sigma^*$ erreichbar ist.
- Also hängt $\|\tilde{Z}\| = \|\{L_x \mid x \in \Sigma^*\}\|$ nur von L ab. $\square$

Direkte Konstruktion eines Minimal-DFA aus L

Satz

Sei $M = (Z, \Sigma, \delta, q_0, E)$ ein DFA ohne unerreichbare Zustände. Dann ist

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

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

Bemerkung

- M' erreicht nach Lesen von x den Zustand $\widetilde{\hat{\delta}(q_0, x)}$. Wegen

$$\begin{aligned}\widetilde{\hat{\delta}(q_0, x)} = \widetilde{\hat{\delta}(q_0, y)} &\Leftrightarrow \hat{\delta}(q_0, x) \sim \hat{\delta}(q_0, y) \\ &\Leftrightarrow L(M_{\hat{\delta}(q_0,x)}) = L(M_{\hat{\delta}(q_0,y)}) \Leftrightarrow L_x = L_y\end{aligned}$$

  können wir den Zustand $\widetilde{\hat{\delta}(q_0, x)}$ von M' auch mit L_x bezeichnen.
- Dies führt auf den zu M' isomorphen DFA $M_L = (Z_L, \Sigma, \delta_L, L_\varepsilon, E_L)$ mit

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

  der sich direkt aus der Sprache L gewinnen lässt.

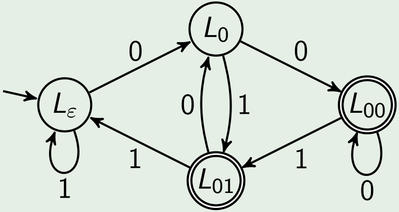
Direkte Konstruktion eines Minimal-DFA aus L

Beispiel

- Betrachte die Sprache $L = \{x_1 \ldots x_n \in \{0,1\}^* \mid x_{n-1} = 0\}$.
- Dann hat M_L die folgenden Sprachen als Zustände:

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

- Graphische Darstellung von M_L:



◁

Der Satz von Myhill und Nerode

- Notwendig und hinreichend für die Existenz von M_L ist, dass die Menge $Z_L = \{L_x \mid x \in \Sigma^*\}$ endlich ist.
- L ist also genau dann regulär, wenn folgende Äquivalenzrelation R_L einen endlichen Index hat:

  $$x\, R_L\, y \;:\Leftrightarrow\; L_x = L_y$$

- Ist M ein DFA mit einer minimalen Anzahl von Zuständen, so haben die Zustände von M' die Form $\tilde{q} = \{q\}$, d.h. M ist isomorph zu M'.
- Da M' wiederum isomorph zu M_L ist, ist jeder minimale DFA M mit $L(M) = L$ isomorph zu M_L, d.h. für jede reguläre Sprache L gibt es bis auf Isomorphie nur einen Minimal-DFA.

Satz (Myhill und Nerode)

1. $\mathrm{REG} = \{L \mid index(R_L) < \infty\}$.
2. Sei L regulär und sei $index(R_L)$ der Index von R_L. Dann gibt es für L bis auf Isomorphie genau einen Minimal-DFA. Dieser hat $index(R_L)$ Zustände.

Der Äquivalenzklassen-DFA M_{R_L} für L

- Zwei Eingaben x und y überführen den DFA M_L genau dann in denselben Zustand, wenn $L_x = L_y$ ist (also xR_Ly gilt).
- Die Zustände von M_L können daher anstelle von L_x auch mit den Äquivalenzklassen $[x]$ von R_L (bzw. mit geeigneten Repräsentanten) benannt werden.
- Der resultierende Minimal-DFA M_{R_L} wird auch als Äquivalenzklassen-automat bezeichnet:

$$M_{R_L} = (Z, \Sigma, \delta, [\varepsilon], E) \text{ mit } Z = \{[x] \mid x \in \Sigma^*\} \text{ und } E = \{[x] \mid x \in L\}.$$

- Für die Konstruktion von δ genügt es, ausgehend von $r_1 = \varepsilon$ eine Folge von Wörtern $r_1, \dots, r_k$ mit $[r_i] \neq [r_j]$ zu bestimmen, so dass zu jedem r_i und jedem Zeichen $a \in \Sigma$ ein r_j existiert mit $r_i a \in [r_j]$.
- In diesem Fall ist dann $\delta([r_i], a) = [r_i a] = [r_j]$.
- Die Konstruktion von M_{R_L} erfordert meist weniger Aufwand als die von M_L, da die Bestimmung der Sprachen L_x entfällt.

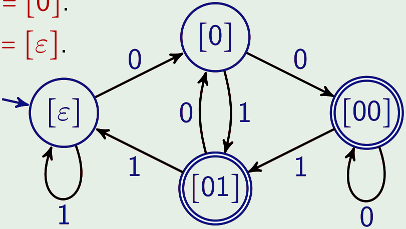
Direkte Konstruktion von M_{R_L} aus L

Beispiel

Für die Sprache $L = \{x_1 \ldots x_n \in \{0,1\}^* \mid x_{n-1} = 0\}$ lässt sich der Äquivalenzklassen-DFA M_{R_L} ausgehend von $r_1 = \varepsilon$ wie folgt konstruieren:

1. Wegen $r_1 0 = 0 \notin [\varepsilon]$ ist $r_2 = 0$ und $\delta([\varepsilon], 0) = [0]$.
2. Wegen $r_1 1 = 1 \in [\varepsilon]$ ist $\delta([\varepsilon], 1) = [\varepsilon]$.
3. Wegen $r_2 0 = 00 \notin [\varepsilon] \cup [0]$ ist $r_3 = 00$ und $\delta([0], 0) = [00]$.
4. Wegen $r_2 1 = 01 \notin [\varepsilon] \cup [0] \cup [00]$ ist $r_4 = 01$ und $\delta([0], 1) = [01]$.
5. Wegen $r_3 0 = 000 \in [00]$ ist $\delta([00], 0) = [00]$.
6. Wegen $r_3 1 = 001 \in [01]$ ist $\delta([00], 1) = [01]$.
7. Wegen $r_4 0 = 010 \in [0]$ ist $\delta([01], 0) = [0]$.
8. Wegen $r_4 1 = 011 \in [\varepsilon]$ ist $\delta([01], 1) = [\varepsilon]$.

r	ε	0	00	01
$[r0]$	$[0]$	$[00]$	$[00]$	$[0]$
$[r1]$	$[\varepsilon]$	$[01]$	$[01]$	$[\varepsilon]$



◁

Charakterisierungen der Klasse REG

Korollar

Sei L eine Sprache. Dann sind folgende Aussagen äquivalent:

- L ist regulär,
- 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 $L = L(\gamma)$,
- die Äquivalenzrelation R_L hat endlichen Index.

Wir können also beweisen, dass eine Sprache L **nicht** regulär ist, indem wir unendlich viele Wörter finden, die paarweise inäquivalent bzgl. R_L sind.

Nachweis von $L \notin \mathrm{REG}$ mittels Myhill und Nerode

Satz

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

Beweis

Die Wörter a^i, $i \geq 0$, sind bzgl. R_L paarweise inäquivalent.
Für $i \neq j$ gilt nämlich $\neg a^i R_L a^j$, da

$$b^i \in L_{a^i} \Delta L_{a^j}$$

enthalten ist. $\square$

Das Pumping-Lemma

Frage

Wie lässt sich möglichst einfach zeigen, dass eine Sprache nicht regulär ist?

Antwort

Oft führt die Kontraposition folgender Aussage zum Ziel.

Satz (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 $i \geq 0$.

Das kleinste solche l wird auch die Pumping-Zahl von L genannt.

Beispiel

- Die Sprache

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

  lässt sich „pumpen“ (mit Pumping-Zahl $l = 3$).

- Sei $x \in L$ beliebig mit $|x| \geq 3$.
  - 1. Fall: x hat das Präfix $\underline{ab}$.
    Zerlege $x = uvw$ mit $u = \varepsilon$ und $v = ab$.
  - 2. Fall: x hat das Präfix $a\underline{ab}$.
    Zerlege $x = uvw$ mit $u = a$ und $v = ab$.
  - 3. Fall: x hat das Präfix $\underline{aaa}$.
    Zerlege $x = uvw$ mit $u = \varepsilon$ und $v = aaa$.
  - Restliche Fälle (Präfixe $\underline{ba}$, $b\underline{ba}$ und $\underline{bbb}$): analog.

Beispiel

- Sei L eine endliche Sprache.
- Offenbar lässt sich kein Wort $x \in L$ „pumpen".
- Sei

$$l = \begin{cases} 1 + \max_{x \in L} |x|, & L \neq \emptyset, \\ 0, & \text{sonst.} \end{cases}$$

- Dann lässt sich jedes Wort $x \in L$ der Länge $\geq l$ „pumpen", da solche Wörter gar nicht existieren. Also hat L eine Pumping-Zahl $\leq l$.
- Zudem gibt es im Fall $l > 0$ ein Wort $x \in L$ der Länge $l - 1$, das sich nicht „pumpen" lässt.
- Somit ist die Pumping-Zahl von L gleich l.

Das Pumping-Lemma

Satz (Pumping-Lemma für reguläre Sprachen)

Zu jeder regulären Sprache L gibt es eine Zahl l, 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 $i \geq 0$.

Das kleinste solche l wird auch die Pumping-Zahl von L genannt.

Das Pumping-Lemma

Beweis

- Sei $M = (Z, \Sigma, \delta, q_0, E)$ ein DFA mit l Zuständen und sei $x = x_1 \dots x_n \in L$ mit $n = |x| \geq l$.
- Dann muss $M(x)$ nach spätestens l Schritten einen Zustand zum zweiten Mal annehmen, d.h. es ex. $0 \leq j < k \leq l$ und $z \in Z$ mit

$$\hat{\delta}(q_0, x_1 \dots x_j) = z \text{ und}$$

$$\hat{\delta}(q_0, x_1 \dots x_j x_{j+1} \dots x_k) = z.$$

- Setze $u = x_1 \dots x_j$, $v = x_{j+1} \dots x_k$ und $w = x_{k+1} \dots x_n$.
- Dann gilt $|v| = k - j \geq 1$ (d.h. $v \neq \varepsilon$), $k = |uv| \leq l$.
- Zudem gehört für alle $i \geq 0$ das Wort $uv^i w$ zu L, da wegen $\hat{\delta}(z, v^i) = z$

$$\hat{\delta}(q_0, uv^i w) = \hat{\delta}(\underbrace{\hat{\delta}(\underbrace{\hat{\delta}(q_0, u)}_{z}, v^i)}_{z}, w) = \hat{\delta}(\underbrace{\hat{\delta}(\hat{\delta}(q_0, u), v)}_{z}, w) = \hat{\delta}(q_0, x)$$

in E ist. ∎

Kontraposition des Pumping-Lemmas

Um also $L \notin \mathrm{REG}$ zu zeigen, genügt es,

- für jede Zahl $l \geq 0$ ein Wort $x \in L$ der Länge $|x| \geq l$ zu finden, so dass
- für jede Zerlegung $x = uvw$ mindestens eine der folgenden drei Bedingungen verletzt ist:
  1. $v \neq \varepsilon$,
  2. $|uv| \leq l$ oder
  3. $uv^i w \in L$ für alle $i \geq 0$.

Beispiel: $L = \{a^n b^n \mid n \geq 0\} \notin \mathrm{REG}$

- Für jede Zahl $l \geq 0$ enthält L das Wort $x = a^l b^l$ mit $|x| = 2l \geq l$.
- Für jede Zerlegung $x = uvw$ von $x = a^l b^l$ mit

  1. $v \neq \varepsilon$

  ist die Bedingung

  3. $uv^i w \in L$

  für alle $i \geq 2$ verletzt.