

Einführung in die Theoretische Informatik

Johannes Köbler



Institut für Informatik
Humboldt-Universität zu Berlin

WS 2016/17

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

- 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 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
- Die (maximale) Anzahl der Rechenschritte bei allen möglichen Eingaben ist nicht beschränkt, d.h. mit wachsender Eingabelänge kann auch die Rechenzeit beliebig anwachsen
- Die Beschreibung eines Algorithmus muss jedoch endlich sein
- Problemeingaben können Zahlen, Formeln, Graphen etc. sein
- Diese werden über einem Eingabealphabet Σ kodiert

Definition

- Ein **Alphabet** ist eine geordnete endliche Menge

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

von **Zeichen** a_i

- Eine Folge $x = x_1 \dots x_n \in \Sigma^n$ von Zeichen heißt **Wort**
- Die **Länge** von $x = x_1 \dots x_n \in \Sigma^n$ ist n und wird mit $|x|$ bezeichnet
- 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

Beispiel

- Sprachen über Σ sind beispielsweise $\emptyset$, Σ^* , Σ und $\{\varepsilon\}$
- $\emptyset$ 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

- Da Sprachen Mengen sind, können wir sie bzgl. Inklusion vergleichen
- Zum Beispiel gilt $\emptyset \subseteq \{\varepsilon\} \subseteq \Sigma^*$
- 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

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 $\square : M \times M \rightarrow M$ und einem neutralen Element e heißt **Monoid**
- $(\Sigma^*, \circ, \varepsilon)$ ist also ein Monoid

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

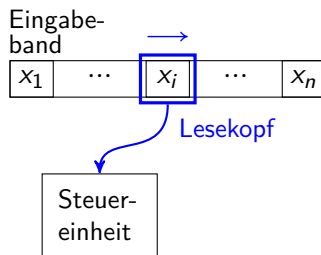
Definition

- 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 $A^+ = \bigcup_{n \geq 1} A^n = AA^*$

- 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

Definition

- Ein **endlicher Automat** (kurz: **DFA**; *Deterministic Finite Automaton*) wird durch ein 5-Tupel $M = (Z, \Sigma, \delta, q_0, E)$ beschrieben, wobei
 - $Z \neq \emptyset$ eine **endliche** Menge von **Zuständen**
 - Σ das **Eingabealphabet**,
 - $\delta : Z \times \Sigma \rightarrow Z$ die **Überföhrungsfunktion**
 - $q_0 \in Z$ der **Startzustand** und
 - $E \subseteq Z$ die Menge der **Endzustände** ist
- Die von M **akzeptierte** oder **erkannte Sprache** ist

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

- 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**

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

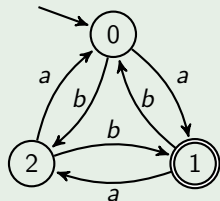
$$\text{REG} = \{L(M) \mid M \text{ ist ein DFA}\}$$

Beispiel

Sei $M_3 = (Z, \Sigma, \delta, 0, E)$ ein DFA mit $Z = \{0, 1, 2\}$, $\Sigma = \{a, b\}$, $E = \{1\}$ und der Überföhrungsfunktion

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 (akzeptierende Rechnung: 0, 1, 0, 1)
- $w_2 = abba$? Nein (verwerfende Rechnung: 0, 1, 0, 2, 0)

Behauptung

Die von M_3 erkannte Sprache ist

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

- $\#_a(x)$ 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

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 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: $\delta(\dots \delta(\delta(q_0, x_1), x_2), \dots x_i)$

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

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

induktiv über die Länge von x wie folgt definieren

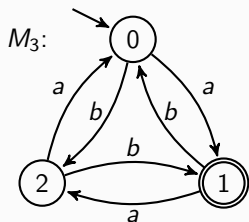
- Für $q \in Z$, $x \in \Sigma^*$ und $a \in \Sigma$ sei

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

- 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

**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, für alle $x \in \Sigma^*$ folgende Kongruenz zu zeigen:

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

Induktionsbehauptung: Für alle $x \in \Sigma^n$ 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 $x = x_1 \dots x_{n+1}$ gegeben.

- Nach IV gilt

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

- Zudem gilt

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

- Somit folgt

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

Vereinbarung

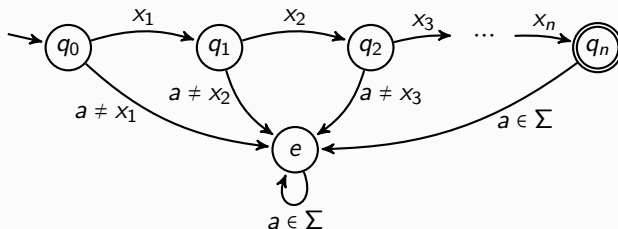
Für das Folgende sei $\Sigma = \{a_1, \dots, a_m\}$ ein fest gewähltes Alphabet.

Beobachtung 1

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

Beweis

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



Beobachtung 2

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

Beweis

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

□

Definition

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

Korollar

$\text{co-REG} = \text{REG}$.

Beobachtung 3

Sind $L_1, L_2 \in \text{REG}$, 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.

Beobachtung 4

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

Beweis

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

□

Frage

Wie sieht der zugehörige DFA aus?

Antwort

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

Definition

- Ein (*k*-stelliger) **Sprachoperator** ist eine Abbildung op , die k Sprachen $L_1, \dots, L_k$ auf eine Sprache $op(L_1, \dots, L_k)$ abbildet.
- Eine Sprachklasse $\mathcal{K}$ heißt unter op **abgeschlossen**, wenn gilt:

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

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

Beispiel

- 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.

Korollar

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

- Komplement,
- Schnitt,
- Vereinigung.

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.

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 das Ende der Simulation von M_1 und den Start der Simulation von M_2 zu finden.

Lösungsidee

Ein nichtdeterministischer Automat (NFA) kann den richtigen Zeitpunkt „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 Überföhrungsfunktion folgende Form hat

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

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 **akzeptierte** oder **erkannte Sprache** ist

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

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

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

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

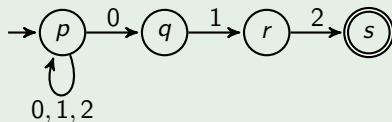
nicht verarbeiten kann.

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 Überföhrungsfunktion

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

Graphische Darstellung:



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

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 $Z_1 \cap Z_2 = \emptyset$ 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

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

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

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

und $E = E_2$, falls $Q_2 \cap E_2 = \emptyset$, 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 $q_0, \dots, q_k$ und $p_0, \dots, p_l$ akzeptierende Rechnungen von $N_1(x)$ und $N_2(y)$.

Dann ist $q_0, \dots, q_k, p_1, \dots, p_l$ eine akz. Rechnung von $N(xy)$, da

- $q_0 \in Q_1$ und $p_l \in E_2$ ist, und
- im Fall $l \geq 1$ wegen $q_k \in E_1, p_0 \in Q_2$ und $p_1 \in \Delta_2(p_0, y_1)$ zudem $p_1 \in \Delta(q_k, y_1)$ und
- im Fall $l = 0$ wegen $q_k \in E_1$ und $p_l \in Q_2 \cap E_2$ zudem $q_k \in E$ ist.

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

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

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

Beweis von $L(N) \subseteq L_1 L_2$:

Sei $x = x_1 \cdots x_n \in L(N)$ und sei $q_0, \dots, q_n$ eine akz. Rechnung von $N(x)$.

Dann gilt $q_0 \in Q_1$, $q_n \in E$, $q_0, \dots, q_i \in Z_1$ und $q_{i+1}, \dots, q_n \in Z_2$ für ein i .

Wir zeigen, dass $q_0, \dots, q_i$ eine akz. Rechnung von $N_1(x_1 \cdots x_i)$ und $q, q_{i+1}, \dots, q_n$ für ein $q \in Q_2$ eine akz. Rechnung von $N_2(x_{i+1} \cdots x_n)$ ist:

- Im Fall $i < n$ impliziert der Übergang $q_{i+1} \in \Delta(q_i, x_{i+1})$, dass $q_i \in E_1$ und $q_{i+1} \in \Delta_2(q, x_{i+1})$ für ein $q \in Q_2$ ist
- Im Fall $i = n$ ist $q_n \in E_1$ (d.h. $x \in L_1$) und $Q_2 \cap E_2 \neq \emptyset$ (d.h. $\varepsilon \in L_2$)

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

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

erkannt.

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)

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

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

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.

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

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

- Die Überföhrungsfunktion $\delta : \mathcal{P}(Z) \times \Sigma \rightarrow \mathcal{P}(Z)$ von M berechnet dann für einen Zustand $Q \subseteq Z$ und ein Zeichen $a \in \Sigma$ die Menge

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

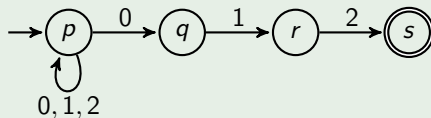
aller Zustände, in die N gelangen kann, wenn N ausgehend von einem beliebigen Zustand $q \in Q$ das Zeichen a liest.

- M wird auch als der zu N gehörige **Potenzmengenautomat** bezeichnet.

Simulation von NFAs durch DFAs

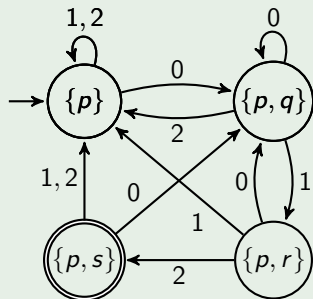
Beispiel

- 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 Potenzmengenautomaten 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 $\{L(N) \mid N \text{ ist ein NFA}\} \subseteq \text{REG}$

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

$x \in L(N)$	$\Leftrightarrow$	N kann nach Lesen von x einen Endzustand erreichen
	$\stackrel{\text{Beh.}}{\Leftrightarrow}$	$\hat{\delta}(Q_0, x) \cap E \neq \emptyset$
	$\Leftrightarrow$	$\hat{\delta}(Q_0, x) \in E'$
	$\Leftrightarrow$	$x \in L(M)$.

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. Nach IV enthält

$$Q_n = \hat{\delta}(Q_0, x_1 \dots x_n)$$

die Zustände, die N nach Lesen von $x_1 \dots x_n$ erreichen kann.
Wegen

$$\hat{\delta}(Q_0, x) = \delta(Q_n, x_{n+1}) = \bigcup_{q \in Q_n} \Delta(q, x_{n+1})$$

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

Korollar

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

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

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.

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 $L(\epsilon) = \{\epsilon\}$ 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:

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

Bemerkung

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

Reguläre Ausdrücke

Beispiel

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

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



Vereinbarungen

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

Satz

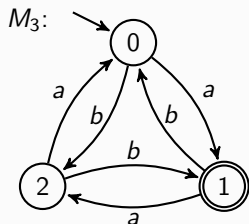
$\{L(\gamma) \mid \gamma \text{ ist ein regulärer Ausdruck über } \Sigma\} \subseteq \text{REG}.$

Beweis.

Klar, da

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





Frage

Wie lässt sich die Sprache

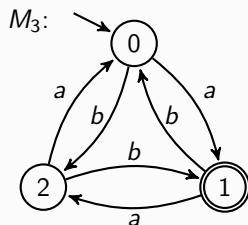
$$L(M_3) = \{x \in \{a, b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 1\}$$

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 $x = x_1 \cdots x_n \in L_{p,q}$, 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 $L(M_3) = L_{0,1} = L_{0,0} L_{0,1}^{\neq 0}$ und $L_{0,0} = (L_{0,0}^{\neq 0})^*$, also $L(M_3) = (L_{0,0}^{\neq 0})^* L_{0,1}^{\neq 0}$.

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:

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

Satz

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

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 $Z = \{1, \dots, m\}$ 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 $1 \leq p, q \leq m$ anzugeben.

Satz

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

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 $r = 0, \dots, m$ die Sprachen

$$L_{p,q}^{\leq r} = \{x_1 \dots x_n \in L_{p,q} \mid \text{für } i = 1, \dots, n-1 \text{ ist } \hat{\delta}(p, x_1 \dots x_i) \leq r\},$$

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.

Satz

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

Beweis (Schluss)

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

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

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

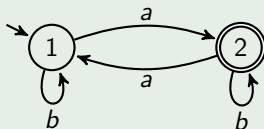
$r \rightsquigarrow r+1$: Nach IV existieren reguläre Ausdrücke $\gamma_{p,q}^r$ für die Sprachen $L_{p,q}^r$. Wegen

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

sind dann $\gamma_{p,q}^{r+1} = \gamma_{p,q}^r \mid \gamma_{p,r+1}^r (\gamma_{r+1,r+1}^r)^* \gamma_{r+1,q}^r$ reguläre Ausdrücke für die Sprachen $L_{p,q}^{r+1}$. □

Beispiel

- Betrachte den DFA M



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

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

Beispiel (Fortsetzung)

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

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

- Damit erhalten wir

$$\gamma_{1,2}^2 = \gamma_{1,2}^1 | \gamma_{1,2}^1 (\gamma_{2,2}^1)^* \gamma_{2,2}^1,$$

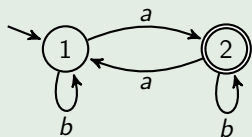
$$\gamma_{1,2}^1 = \gamma_{1,2}^0 | \gamma_{1,1}^0 (\gamma_{1,1}^0)^* \gamma_{1,2}^0,$$

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

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

Beispiel (Fortsetzung)

DFA M



Rekursionsformeln

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

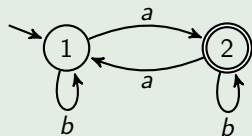
$$L_{p,q}^0 = \{a \mid \delta(p, a) = q\} \text{ für } p \neq q$$

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

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

Beispiel (Fortsetzung)

DFA M



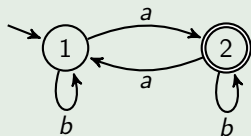
Rekursionsformel

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

$$\leadsto \gamma_{1,1}^0 = \varepsilon|b$$

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	εb	$\gamma_{1,2}^0$	$\gamma_{2,1}^0$	$\gamma_{2,2}^0$
1	-	$\gamma_{1,2}^1$	-	$\gamma_{2,2}^1$
2	-	$\gamma_{1,2}^2$	-	-

Beispiel (Fortsetzung)

DFA M 

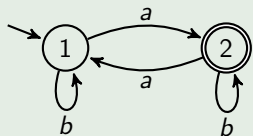
Rekursionsformel

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

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

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	ϵb	a	$\gamma_{2,1}^0$	$\gamma_{2,2}^0$
1	-	$\gamma_{1,2}^1$	-	$\gamma_{2,2}^1$
2	-	$\gamma_{1,2}^2$	-	-

Beispiel (Fortsetzung)

DFA M 

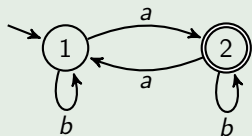
Rekursionsformel

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

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

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	ϵb	a	a	$\gamma_{2,2}^0$
1	-	$\gamma_{1,2}^1$	-	$\gamma_{2,2}^1$
2	-	$\gamma_{1,2}^2$	-	-

Beispiel (Fortsetzung)

DFA M 

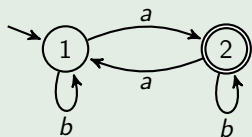
Rekursionsformel

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

$$\leadsto \gamma_{2,2}^0 = \varepsilon|b$$

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	εb	a	a	εb
1	-	$\gamma_{1,2}^1$	-	$\gamma_{2,2}^1$
2	-	$\gamma_{1,2}^2$	-	-

Beispiel (Fortsetzung)

DFA M 

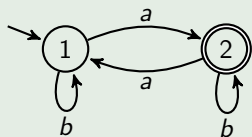
Rekursionsformel

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

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

Beispiel (Fortsetzung)

DFA M



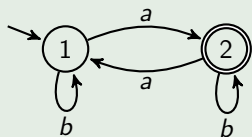
Rekursionsformel

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

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

Korollar

Sei L eine Sprache. Dann sind folgende Aussagen äquivalent:

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

Definition

- Sei A eine nichtleere Menge, R ist eine **k -stellige Relation auf A** , wenn $R \subseteq A^k = \underbrace{A \times \dots \times A}_{k\text{-mal}} = \{(a_1, \dots, a_k) \mid a_i \in A \text{ für } i = 1, \dots, k\}$ ist.
- Für $i = 1, \dots, n$ sei R_i eine k_i -stellige Relation auf A . Dann heißt $(A; R_1, \dots, R_n)$ **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).

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

$$R \cap S = \{(x, y) \in A \times A \mid xRy \wedge xSy\},$$

$$R \cup S = \{(x, y) \in A \times A \mid xRy \vee xSy\},$$

$$R - S = \{(x, y) \in A \times A \mid xRy \wedge \neg xSy\},$$

$$\overline{R} = (A \times A) - R.$$

- 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:

$$\bigcap \mathcal{M} = \bigcap_{R \in \mathcal{M}} R = \{(x, y) \mid \forall R \in \mathcal{M} : xRy\},$$

$$\bigcup \mathcal{M} = \bigcup_{R \in \mathcal{M}} R = \{(x, y) \mid \exists R \in \mathcal{M} : xRy\}.$$

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. $\triangleleft$

Notation

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

Vorsicht!

Das Relationenprodukt R^n sollte nicht mit dem kartesischen Produkt

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

verwechselt werden.

Vereinbarung

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

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 $Id_A \subseteq \bar{R}$)

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 $R \subseteq \bar{R}^T$)

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 $\bar{Id} \subseteq R \cup R^T$)

transitiv, falls $\forall x, y, z \in A : xRy \wedge yRz \Rightarrow xRz$ (also $R^2 \subseteq R$)

gilt.

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.

Beispiel

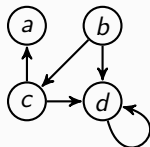
- Die Relation "ist Schwester von" ist zwar in einer reinen Damengesellschaft symmetrisch, i.a. jedoch weder symmetrisch noch asymmetrisch noch antisymmetrisch.
- Die Relation "ist Geschwister von" ist zwar symmetrisch, aber weder reflexiv noch transitiv und somit keine Äquivalenzrelation.
- $(\mathbb{R}, <)$ 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.



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**.

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 $E = \{\{a, b\} \mid aRb\}$ notiert.

Matrixdarstellung (Adjazenzmatrix)

Eine Relation R auf $A = \{a_1, \dots, a_n\}$ 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

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



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 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

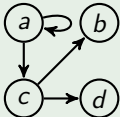
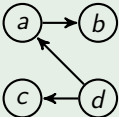
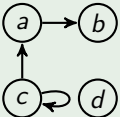
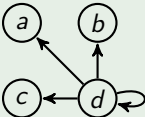
$$t_{ij} = \bigvee_{k=1, \dots, n} (r_{ik} \wedge s_{kj}).$$

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

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

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				
Adjazenzmatrix	$\begin{matrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{matrix}$	$\begin{matrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{matrix}$	$\begin{matrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$	$\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{matrix}$
Adjazenzlisten	$\begin{array}{l} a: a, c \\ b: - \\ c: b, d \\ d: - \end{array}$	$\begin{array}{l} a: b \\ b: - \\ c: - \\ d: a, c \end{array}$	$\begin{array}{l} a: b \\ b: - \\ c: a, c \\ d: - \end{array}$	$\begin{array}{l} a: - \\ b: - \\ c: - \\ d: a, b, c, d \end{array}$

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.

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

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

Satz

$$h_{\text{refl}}(R) = R \cup \text{Id}_A, \quad h_{\text{sym}}(R) = R \cup R^T, \quad R^+ = \bigcup_{n \geq 1} R^n, \quad R^* = \bigcup_{n \geq 0} R^n.$$

Beweis

Siehe Übungen.



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 $x_0, \dots, x_n \in V$ gibt mit
$$x_0 = a, x_n = b \text{ und } (x_{i-1}, x_i) \in E \text{ für } i = 1, \dots, n.$$
- $x_0, \dots, x_n$ heißt **Weg** der Länge n von a nach b .
- G heißt **zusammenhängend**, wenn es für je zwei Knoten a und b einen Weg von a nach b oder einen Weg von b nach a gibt.
- G heißt **stark zusammenhängend**, wenn es von jedem Knoten a einen Weg zu jedem Knoten b in G gibt.

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 ".



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.

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 $[0], [1], \dots, [m-1]$ mit

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

bildet eine Äquivalenzklasse.

- Repräsentantensystem: $\{0, 1, \dots, m-1\}$.

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.

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 $B_i \cap B_j = \emptyset$.

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 **größte** Äquivalenzrelation.

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.

1 impliziert 2: Sei E eine Äquivalenzrelation auf A . Dann bildet $\{E[x] \mid x \in A\}$ eine Partition von A mit der gewünschten Eigenschaft:

- Da E reflexiv ist, gilt xEx und somit $x \in E[x]$, d.h. $A = \bigcup_{x \in A} E[x]$.
- Ist $E[x] \cap E[y] \neq \emptyset$ und $u \in E[x] \cap E[y]$, so folgt $E[x] = E[y]$:

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

- Zudem gilt

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

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 impliziert 1: Existiert umgekehrt eine Partition $\{B_i \mid i \in I\}$ von A mit $xEy \Leftrightarrow \exists i \in I : x, y \in B_i$, so ist E

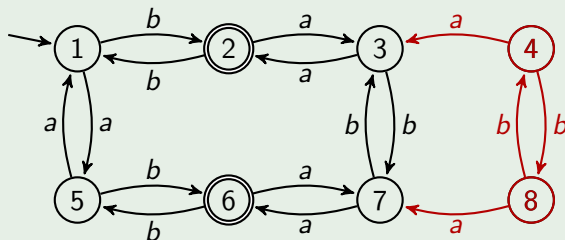
- reflexiv, da zu jedem $x \in A$ eine Menge B_i mit $x \in B_i$ existiert,
- symmetrisch, da aus $x, y \in B_i$ auch $y, x \in B_i$ folgt, und
- transitiv, da aus $x, y \in B_i$ und $y, z \in B_j$ wegen $y \in B_i \cap B_j$ die Gleichheit $B_i = B_j$ und somit $x, z \in B_i$ folgt. $\square$

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.

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: $p \sim_M q$), wenn $L(M_p) = L(M_q)$ ist.
- Offensichtlich ist $\sim_M$ eine Äquivalenzrelation auf Z .

Idee

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

Notation

- Für die durch z repräsentierte Äquivalenzklasse

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

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 $p \not\sim_M q$ von inäquivalenten Zuständen zu finden.
- Sei $A \triangle B = (A \setminus B) \cup (B \setminus A)$ die **symmetrische Differenz** von A und B .
- Dann gilt

$$\begin{aligned} p \not\sim_M q &\Leftrightarrow L(M_p) \neq L(M_q) \\ &\Leftrightarrow L(M_p) \triangle L(M_q) \neq \emptyset. \end{aligned}$$

- Wörter $x \in L(M_p) \triangle L(M_q)$ 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.

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

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

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

- 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.

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

- Da es nur endlich viele Zustandspaare gibt, muss es ein k geben mit

$$D = D_k.$$

- Offensichtlich gilt

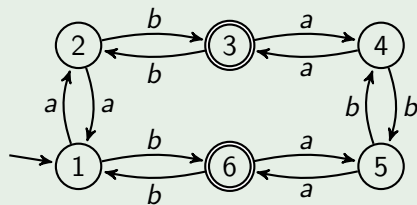
$$D = D_k \Leftrightarrow D_{k+1} = D_k.$$

Algorithmus min-DFA(M)

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

Beispiel

Betrachte den DFA M



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

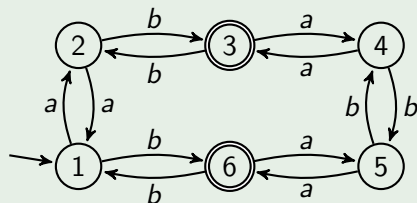
Dann enthält D_0 die Paare

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

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

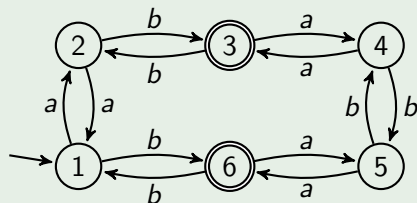
$\{p, q\}$	$\{1, 4\}$	$\{1, 5\}$	$\{2, 4\}$	$\{2, 5\}$
$\{\delta(q, a), \delta(p, a)\}$	$\{2, 3\}$	$\{2, 6\}$	$\{1, 3\}$	$\{1, 6\}$

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



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

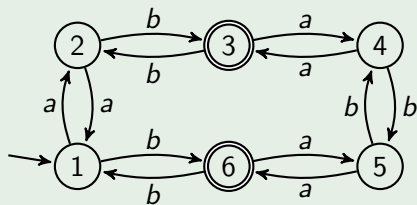
Da nun jedoch keines der verbliebenen Paare $\{1, 2\}$, $\{3, 6\}$, $\{4, 5\}$ wegen

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

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

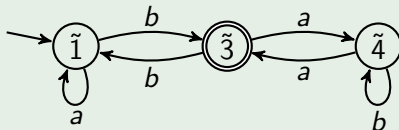
Beispiel

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

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 $\tilde{4} = \{4,5\}$:



Satz

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

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

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 $p \sim_M q \Rightarrow \delta(p, a) \sim_M \delta(q, a)$ zu zeigen.
- Diese ist äquivalent zur Kontraposition $\delta(p, a) \not\sim_M \delta(q, a) \Rightarrow p \not\sim_M q$, die wir bereits gezeigt haben:

$$\begin{aligned} \delta(p, a) \not\sim_M \delta(q, a) &\Rightarrow \text{es gibt einen Unterscheider } x \\ &\quad \text{zwischen } \delta(p, a) \text{ und } \delta(q, a) \\ &\Rightarrow \text{es gibt einen Unterscheider } ax \\ &\quad \text{zwischen } p \text{ und } q \\ &\Rightarrow p \not\sim_M q \end{aligned}$$

Satz

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

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

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 $M(x)$ besuchten 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' besucht 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

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

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.: M' hat eine minimale Anzahl $\|\tilde{Z}\|$ von Zuständen.
- Wegen $\|\tilde{Z}\| \leq \|Z\|$ 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 $p \sim_M q \Leftrightarrow L(M_p) = L(M_q)$ 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.

Direkte Konstruktion eines Minimal-DFA aus L

Satz

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

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

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

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

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 $M_L = (Z_L, \Sigma, \delta_L, L_\epsilon, E_L)$ mit

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

der sich direkt aus der Sprache L gewinnen lässt.

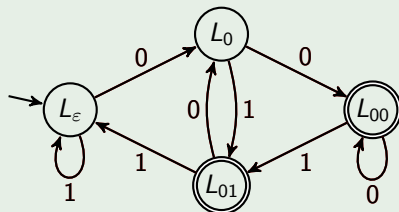
Beispiel

- Betrachte die Sprache $L = \{x_1 \dots x_n \in \{0,1\}^* \mid n \geq 2 \text{ und } x_{n-1} = 0\}$.
- 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 :

L_x	L_ε	L_0	L_{00}	L_{01}
L_{x0}	L_0	L_{00}	L_{00}	L_0
L_{x1}	L_ε	L_{01}	L_{01}	L_ε



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 auf Σ^* einen endlichen Index hat:

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

- Die Relation $\sim_L$ wird als **Nerode-Relation** der Sprache L bezeichnet.
- 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 Sei L regulär und sei $index(\sim_L)$ der Index von $\sim_L$. Dann gibt es für L bis auf Isomorphie genau einen Minimal-DFA. Dieser hat $index(\sim_L)$ Zustände.
- 2 $REG = \{L \mid index(\sim_L) < \infty\}$.

Der Äquivalenzklassen-DFA $M_{\sim_L}$ für L

- Zwei Eingaben x und y überführen den DFA M_L genau dann in denselben Zustand, wenn $L_x = L_y$ ist (also $x \sim_L y$ gilt).
- Die Zustände von M_L können daher anstelle von L_x auch mit den Äquivalenzklassen $[x]_{\sim_L}$ von $\sim_L$ benannt werden, die wir auch einfach mit $\tilde{x}$ bezeichnen.
- Der resultierende Minimal-DFA $M_{\sim_L}$ wird auch als **Äquivalenzklassen-automat** bezeichnet:

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

- Für die Konstruktion von δ genügt es, ausgehend von $r_1 = \varepsilon$ eine Folge von Wörtern $r_1, \dots, r_k$ mit $r_i \not\sim_L r_j$ für $j < i$ zu bestimmen, so dass zu jedem r_i und jedem Zeichen $a \in \Sigma$ ein r_j existiert mit $r_i a \sim_L r_j$.
- In diesem Fall ist dann $\delta(\tilde{r}_i, a) = \widetilde{r_i a} = \tilde{r}_j$.
- Die Konstruktion von $M_{\sim_L}$ erfordert meist weniger Aufwand als die von M_L , da die Bestimmung der Sprachen L_x entfällt.

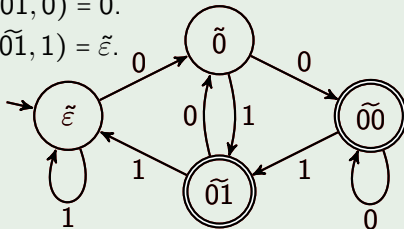
Direkte Konstruktion von $M_{\sim_L}$ aus L

Beispiel

Für die Sprache $L = \{x_1 \dots x_n \in \{0,1\}^* \mid n \geq 2 \text{ und } x_{n-1} = 0\}$ lässt sich der Äquivalenzklassen-DFA $M_{\sim_L}$ ausgehend von $r_1 = \varepsilon$ wie folgt konstruieren:

- ➊ Wegen $r_1 0 = 0 \not\sim_L r_1$ ist $r_2 = 0$ und $\delta(\tilde{\varepsilon}, 0) = \tilde{0}$.
- ➋ Wegen $r_1 1 = 1 \sim_L \varepsilon$ ist $\delta(\tilde{\varepsilon}, 1) = \tilde{\varepsilon}$.
- ➌ Wegen $r_2 0 = 00 \not\sim_L r_i$ für $i = 1, 2$ ist $r_3 = 00$ und $\delta(\tilde{0}, 0) = \tilde{00}$.
- ➍ Wegen $r_2 1 = 01 \not\sim_L r_i$ für $i = 1, 2, 3$ ist $r_4 = 01$ und $\delta(\tilde{0}, 1) = \tilde{01}$.
- ➎ Wegen $r_3 0 = 000 \sim_L 00$ ist $\delta(\tilde{00}, 0) = \tilde{00}$.
- ➏ Wegen $r_3 1 = 001 \sim_L 01$ ist $\delta(\tilde{00}, 1) = \tilde{01}$.
- ➐ Wegen $r_4 0 = 010 \sim_L 0$ ist $\delta(\tilde{01}, 0) = \tilde{0}$.
- ➑ Wegen $r_4 1 = 011 \sim_L \varepsilon$ ist $\delta(\tilde{01}, 1) = \tilde{\varepsilon}$.

r	ε	0	00	01
$\tilde{r0}$	$\tilde{0}$	$\tilde{00}$	$\tilde{00}$	$\tilde{0}$
$\tilde{r1}$	$\tilde{\varepsilon}$	$\tilde{01}$	$\tilde{01}$	$\tilde{\varepsilon}$



Korollar

Sei L eine Sprache. Dann sind folgende Aussagen äquivalent:

- L ist regulär (d.h. es gibt einen DFA M mit $L = L(M)$),
- es gibt einen NFA N mit $L = L(N)$,
- es gibt einen regulären Ausdruck γ mit $L = L(\gamma)$,
- die Nerode-Relation $\sim_L$ von L hat endlichen Index.

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

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. $\sim_L$ paarweise inäquivalent.

Für $i \neq j$ gilt nämlich $a^i \not\sim_L a^j$, da

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

enthalten ist.



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

- ❶ $v \neq \varepsilon$,
- ❷ $|uv| \leq l$ und
- ❸ $uv^i w \in L$ 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 ab.

Zerlege $x = uvw$ mit $u = \varepsilon$ und $v = ab$.

- 2. Fall: x hat das Präfix aab.

Zerlege $x = uvw$ mit $u = a$ und $v = ab$.

- 3. Fall: x hat das Präfix aaa.

Zerlege $x = uvw$ mit $u = \varepsilon$ und $v = aaa$.

- Restliche Fälle (Präfixe ba, bba und bbb): analog.