Grundlagen Theoretischer Informatik I

SoSe 2011 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

Grundlagen Theoretischer Informatik I Gesamtübersicht

- Organisatorisches; Einführung
- Logik & Beweisverfahren
- Mengenlehre
- reguläre Sprachen

Ein **nichtdeterministischer endlicher Automat** oder NEA wird beschrieben durch ein Quintupel

$$A = (Q, \Sigma, \delta, Q_0, F)$$

wobei gilt:

Q: endliche Menge von *Zuständen* (Zustandsalphabet)

Σ: endliche Menge von *Eingabezeichen* (Eingabealphabet)

$\delta \subseteq Q \times \Sigma \times Q$: *Überführungsrelation*

$Q_0 \subseteq Q$: *Anfangszustände*

$F \subseteq Q$: *Endzustände*

$\rightsquigarrow$ **NEA**-Sprachen

λ-Übergänge

Manchmal: NEAs mit λ-Übergängen (kurz: λ-NEA), d.h., für die Überführungsrelation δ gilt: $\delta \subseteq Q \times (\Sigma \cup \{\lambda\}) \times Q$.
Betrachte λ-NEA $A = (Q, \Sigma, \delta, Q_0, F)$ und die Relation

$$R_\lambda = \{(q_1, q_2) \mid (q_1, \lambda) \vdash_A^* (q_2, \lambda)\} \subseteq Q \times Q.$$

Für $A' = (Q, \Sigma, \delta', Q_0', F)$ gilt $L(A) = L(A')$, wobei
$Q_0' = \{q \in Q \mid \exists q_0 \in Q_0 : (q_0, q) \in R_\lambda\}$
$\delta' = \{(p, a, q') \in Q \times \Sigma \times Q \mid \exists q \in Q : (p, a, q) \in \delta, (q, q') \in R_\lambda\}$.
Es gilt für $w = a_0 \dots a_{n-1} \in \Sigma^n$: $(q_0, w) \vdash_A^* (q_f, \lambda)$, $q_f \in F$, gdw.
$\exists q_0', q_1, q_1', \dots, q_n \in Q : \forall i \in \mathbb{Z}_n (q_i, q_i') \in R_\lambda, (q_i', a_i, q_{i+1}) \in \delta$ mit $q_n' = q_f$
gdw. $\forall i \in \mathbb{Z}_n (q_i', a_i, q_{i+1}') \in \delta'$.
Die Konstruktion zeigt:
Satz: Zu jedem λ-NEA gibt es einen *äquivalenten* NEA.

Zur Berechnung der transitiven Hülle

Motivation: R_λ ist die transitive reflexive Hülle der Relation $\delta \cap Q \times \{\lambda\} \times Q$.

Betrachte allgemein $X = \{1, \ldots, n\}$ und eine binäre Relation R auf X.
Frage: Wie berechnet man die transitive Hülle R^+?
(Beachte: $R^* = (R \cup R^0)^+$.)

$\rightsquigarrow$ Erinnerung: Algorithmus von Warshall/Floyd liefert kubischen Algorithmus.

Warum NEAs mit λ-Übergängen ?

Lemma: Zu jedem NEA (mit λ-Übergängen) gibt es einen äquivalenten NEA mit λ-Übergängen, der nur einen Anfangs- und nur einen Endzustand besitzt; der Anfangszustand hat nur ausgehende Kanten und der Endzustand nur eingehende Kanten.

Beweis: Führe neuen Anfangszustand q_0 und neuen Endzustand q_f ein.

Verbinde q_0 zu allen "alten" Anfangszuständen mit λ-Übergängen.

Führe λ-beschriftete Kanten ein von allen "alten" Endzuständen zu q_f.

Beispiel: Man veranschauliche sich die Konstruktion beim Skelettautomaten!

Monoide aus Monoiden

Ist $(M, \circ, e)$ ein Monoid, so kann der Menge 2^M durch das *Komplexprodukt* zu einem Monoid gemacht werden. Dazu definieren wir:

$$A \circ B := \{a \circ b \mid a \in A \wedge b \in B\}$$

Das zugehörige neutrale Element ist $\{e\}$.

Beispiel: $(\Sigma^*, \cdot, \lambda)$ ist ein Monoid, und so kann man auch die Konkatenation $\cdot$ als Sprachoperation auffassen.

Satz: **REG** ist gegen Konkatenation abgeschlossen.

Beweis: Es seien $L_1, L_2 \in$ **REG**.
Wir können davon ausgehen, dass ein λ-NEA

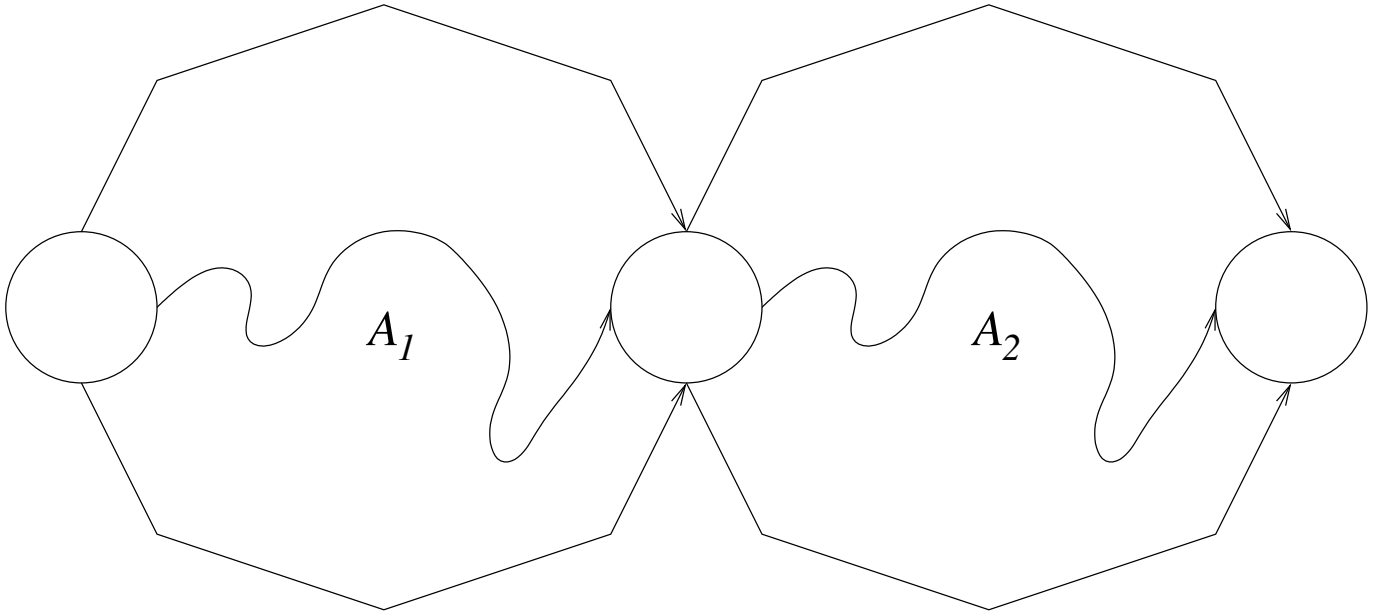
$$A_i = (Q_i, \Sigma, \delta_i, Q_{0,i}, F_i)$$

L_i akzeptiert, der nur einen Anfangs- und einen Endzustand besitzt; der Anfangszustand hat nur ausgehende Kanten und der Endzustand nur eingehende.

Wir gehen ferner davon aus, dass $Q_1 \cap Q_2 = F_1 = Q_{0,2}$ gilt.

Setze $Q = Q_1 \cup Q_2$ und $\delta = \delta_1 \cup \delta_2$. $A = (Q, \Sigma, \delta, Q_{0,1}, F_2)$ akzeptiert $L_1 \cdot L_2$.

REG ist gegen Konkatenation abgeschlossen: Skizze



Potenzen in Monoiden: Der Weg zum Kleene-Stern.

Ist $(M, \circ, e)$ ein Monoid, so können wir rekursiv die n-te *Potenz* eines Elementes $x \in M$ rekursiv festlegen durch:
$x^0 = e$
$x^{n+1} = x^n \circ x$ für $n \in \mathbb{N}$.

Wie wir gesehen haben, bildet auch $(2^M, \circ, \{e\})$ ein Monoid.
Somit ist auch A^n für $A \subseteq M$ und $n \in \mathbb{N}$ definiert.
(Leider kollidiert diese Schreibweise mit dem von dem kartesischen Mengenprodukt induzierten Potenz, aber nicht arg...)

Dann kann man $A^+ = \bigcup_{n \geq 1} A^n$ definieren und $A^* = \bigcup_{n \geq 0} A^n$.

Somit ist auch L^+ und L^* (*Kleene-Stern*) für $L \subseteq \Sigma^*$ festgelegt.

Satz: **REG** ist gegen Kleene-Stern abgeschlossen.

Beweis: Sei $L \in$ **REG** akzeptiert durch einen λ-NEA A, der nur einen Anfangs- und einen Endzustand q_0 und q_f besitzt; der Anfangszustand hat nur ausgehende Kanten und der Endzustand nur eingehende.
Durch Verschmelzen von q_0 und q_f als neuer Anfangs- und Endzustand erhalten wir einen NEA A' mit $L(A') = L^*$.

Reguläre Ausdrücke (ähnlich `grep`)

Definition durch *strukturelle Induktion*:

- $\emptyset$ und $\mathtt{a}$ sind RA für jedes $a \in \Sigma$.
- Ist R ein RA, so auch $(R)*$.
- Sind R_1 und R_2 RAs, so auch R_1R_2 und $(R_1 \cup R_2)$.

Beispiel: $((\mathtt{b} \cup \mathtt{a}))*\mathtt{aaa}(\mathtt{bb})*$ ist ein RA.

Klammern können weggelassen werden: $*$ bindet stärker als Konkatenation, und jenes wieder stärker als Vereinigung.

Die **durch einen RA beschriebene Sprache** ist ebenfalls rekursiv gegeben:

- $L(\emptyset) = \emptyset$; $L(a) = \{a\}$.
- Ist R aus RE, setze $L((R)*) = (L(R))^*$.
- Sind R_1 und R_2 RA, setze
  $L(R_1R_2) = (L(R_1)L(R_2))$ und $L((R_1 \cup R_2)) = (L(R_1) \cup L(R_2))$.

Beispiel: $L((b \cup a)*) = \{a, b\}^*$

Beispiele

(1) Beschreibe die Sprache zum Ausdruck $(\mathtt{ab}*)\mathtt{a}$ in Mengennotation.

$$\begin{aligned} L((\mathtt{ab}*)\mathtt{a}) &= L((\mathtt{ab}*)) \cdot L(\mathtt{a}) \\ &= L(\mathtt{a}) \cdot L(\mathtt{b}*) \cdot \{\mathtt{a}\} \\ &= \{\mathtt{a}\} \cdot (L(\mathtt{b}))^* \cdot \{\mathtt{a}\} \\ &= \{\mathtt{a}\} \cdot \{\mathtt{b}\}^* \cdot \{\mathtt{a}\} \\ &= \{\mathtt{a}\mathtt{b}^n\mathtt{a} \mid n \in \mathbb{N}\} \end{aligned}$$

(2) Beschreibe die Sprache zum Ausdruck $(\mathtt{a} * \mathtt{b})*$ in Worten.

Die Menge aller Wörter über $\{\mathtt{a}, \mathtt{b}\}$, die nicht mit $\mathtt{a}$ enden.

Satz: Jede RA-Sprache ist regulär.

Beweis: (durch strukturelle Induktion)

- Endliche Sprachen sind regulär.
- Reguläre Sprachen sind gegen Kleene-Stern abgeschlossen.
- Reguläre Sprachen sind gegen Vereinigung und Konkatenation abgeschlossen.

Beispiel: $(\mathtt{a} \cup \mathtt{ab})*$ (siehe Tafel)

Satz: Jede reguläre Sprache ist durch einen RA beschreibbar.

Beweis: Betrachte DEA $A = (Q, \Sigma, \delta, q_0, F)$ mit $Q = \{1, \ldots, n\}$ und $q_0 = 1$.

$R[i, j, k]$ RA für die Sprache, die von A akzeptiert wird, indem (1) A in Zustand i anfängt, (2) in Zustand j aufhört, und (3) zwischendurch nur Zustände aus $\{1, \ldots, k\}$ erreicht.

Hinweis: Warshall/Floyd

Offenbar gilt: $L(A) = \bigcup_{j \in F} L(R[1, j, n]) = L(\bigcup_{j \in F} R[1, j, n])$.

$R[i, j, 0] = x_1 \cup \cdots \cup x_\ell$, wobei x_l alle Beschriftungen von Kanten zwischen i und j auflisten (zusätzlich $\emptyset*$ falls $i = j$)

Für $k > 0$ setze rekursiv $R[i, j, k] = R[i, j, k-1] \cup R[i, k, k-1] R[k, k, k-1]* R[k, j, k-1]$.

Daraus ergibt sich ein Algorithmus durch **dynamisches Programmieren**.

$R[1..n, 1..n, 0..n]$ ist 3-dimensionales Array, dessen Einträge reguläre Ausdrücke sind.

Für $i := 1$ bis n tue:
    Für $j := 1$ bis n tue:
        $R[i, j, 0] := \bigcup_{a \in \Sigma, (i,a,j) \in \delta} a$
        Falls $i = j$, so setze $R[i, j, 0] := R[i, j, 0] \cup \emptyset*$.

Für $k := 1$ bis n tue:
    Für $i := 1$ bis n tue:
        Für $j := 1$ bis n tue:
            $R[i, j, k] := R[i, j, k-1] \cup R[i, k, k-1] R[k, k, k-1] * R[k, j, k-1]$.

Damit klar: kubische Komplexität, i.Z.: $O(n^3)$.
Geschicktere Vorgehensweise: Lazy Evaluation.

Die Spiegeloperation

Informell: w^R ergibt sich aus w durch “Rückwärtslesen” (Spiegeln).

Rekursiv: $\lambda^R = \lambda$;
für $w = va$ mit $v \in \Sigma^*$, $a \in \Sigma$, definiere: $w^R := a(v^R)$.

Beispiel: $(abcd)^R = d(abc)^R = dc(ab)^R = dcb(a)^R = dcba(\lambda^R) = dcba\lambda = dcba$.

Erweiterung auf Wortmengen: $L^R = \{w^R \mid w \in L\}$.

Satz: Die regulären Sprachen sind unter Spiegelung abgeschlossen.

Beweis: Wichtig: Wahl des richtigen Modells für **REG**.

Das Pumping-Lemma

Satz: Zu jeder regulären Sprache L gibt es eine Zahl $n > 0$, sodass jedes Wort $w \in L$ mit $\ell(w) \geq n$ als Konkatenation $w = xyz$ dargestellt werden kann mit geeigneten x, y, z mit folgenden Eigenschaften:

1. $\ell(y) > 0$;
2. $\ell(xy) \leq n$;
3. $\forall i \geq 0 : xy^i z \in L$.

Hinweis: Die Umkehrung gilt nicht !

Zur Anwendung des Pumping-Lemmas (schematisch)

1. Wir vermuten, eine vorgegebene Sprache L ist nicht regulär.

2. Im Widerspruch zu unserer Annahme nehmen wir an, L wäre doch regulär. Dann gibt es die im Pumping-Lemma genannte Pumping-Konstante $\mathfrak{n}$.

3. Wir wählen ein geeignetes, hinreichend langes Wort $w \in \mathrm{L}$ (d.h., $\ell(w) \geq \mathfrak{n}$). Dies ist der Schritt, wo man leicht "gut" oder "schlecht" wählt ! Bemerkung: Da wir ja vermuten, L ist nicht regulär, ist L insbesondere unendlich, d.h., zu jedem $\mathfrak{n}$ finden wir ein $w \in \mathrm{L}$ mit $\ell(w) \geq \mathfrak{n}$.

4. Wir diskutieren alle möglichen Zerlegungen $w = xyz$ mit $\ell(y) > 0$ und $\ell(xy) \leq n$ und zeigen für jede solche Zerlegung, dass es ein $i \geq 0$ gibt, sodass $xy^iz \notin L$ gilt.

   Die Komplexität dieser Diskussion hängt im Wesentlichen von der "geschickten" Wahl von w ab; das Anfangsstück von w der Länge n sollte "schön" sein.

Zur Anwendung des Pumping-Lemmas (ein geschickter Einsatz)

Betrachte $L = \{a^k b^k \mid k \in \mathbb{N}\}$.

Wäre L regulär, so gäbe es Pumping-Konstante n.

Betrachte $a^n b^n \in L$; denn: $\ell(a^n b^n) = 2n \geq n$ und Präfix der Länge n ist a^n (sehr schön).

Diskutiere $a^n b^n = xyz$ mit $\ell(xy) \leq n$ und $\ell(y) > 0$:
Offenbar ist $xy \in \{a\}^+$ und damit $y = a^m$ für ein $m > 0$.
$\rightsquigarrow$ Nullpumpen liefert $a^{n-m} b^n \notin L$, Widerspruch zur Annahme, L wäre regulär.

Zur Anwendung des Pumping-Lemmas (ein ungeschickter Einsatz)

Betrachte $L = \{a^k b^k \mid k \in \mathbb{N}\}$.

Wäre L regulär, so gäbe es Pumping-Konstante n. Da L nur Wörter gerader Länge enthält, können wir annehmen, n ist gerade.

Betrachte $w = a^{n/2} b^{n/2} \in L$ mit $\ell(w) = n$.

Diskutiere $w = xyz$ mit $\ell(xy) \leq n$ und $\ell(y) > 0$:
Fall 1: $xy \in \{a\}^+$ (Nullpumpen ähnlich wie letzte Folie)
Fall 2: $xy = a^{n/2} b^m$ mit $m > 0$.
Fall 2a: $y \in \{b\}^+$ (Nullpumpen ähnlich wie letzte Folie)
Fall 2b: $y = a^r b^m$ mit $r, m > 0$. Dann liegt auch $xy^2z = a^{n/2} b^m a^r b^{n/2} \in L$ im Widerspruch zur Struktur von L.

Noch eine Anwendung des Pumping-Lemmas

Betrachte $L = \{w \in \{a, b\}^* \mid w = w^R\}$ (*Palindrome*)

Wäre L regulär, so gäbe es Pumping-Konstante n für L.

Betrachte $w = a^n b a^n \in L$ mit $\ell(w) \geq n$.

Widerspruch ergibt sich durch Nullpumpen.

. . . und noch eine . . .

Betrachte $L = \{a^{k^2} \mid k \in \mathbb{N}\}$.

Wäre L regulär, so gäbe es Pumping-Konstante n für L.

Betrachte $w = a^{(n+1)^2} \in L$ mit $\ell(w) \geq n$.

Widerspruch ergibt sich durch Nullpumpen:
Genauer haben wir, dass für ein $0 < i \leq n$ stimmen muss, dass $a^{(n+1)^2-i} \in L$ gilt, im Widerspruch zu folgender Abschätzung, die $a^{(n+1)^2-i} = a^{r^2}$ annimmt:

$$r^2 \leq n^2 < n^2 + n + (n - i) + 1 = n^2 + 2n + 1 - i = (n + 1)^2 - i.$$

Der Beweis des Pumping-Lemmas

Ist L endlich, so ist die Aussage trivial mit $n := \max\{\ell(w) \mid w \in L\}$.

Ist L unendlich aber regulär, so wird L von einem DEA A mit n Zuständen akzeptiert mit Anfangszustand q_0.

Betrachte ein Wort $w = a_1 \dots a_m \in L$, $a_i \in \Sigma$, $m \geq n$.
Sei $(q_k, a_{k+1} \dots a_m)$ für $0 \leq k \leq n$ die Konfiguration nach k Schritten von A.
Da hierbei $n + 1$ Zustände durchlaufen werden, gibt es nach dem Schubfachprinzip einen Zustand, der zweimal erreicht wird, d.h., $\exists 0 \leq r < s \leq n : q_r = q_s$.
$\rightsquigarrow y = a_{r+1} \dots a_s$ erfüllt $(q_r, y) \vdash_A^* (q_r, \lambda)$.
$\rightsquigarrow \forall i \geq 0 : (q_r, y^i) \vdash_A^* (q_r, \lambda)$.
Mit $x = a_1 \dots a_r$ und $z = a_{s+1} \dots a_m$ folgt die Behauptung.