

Diskrete Mathematik

Arne Dür Kurt Girstmair Simon Legner
 Georg Moser Harald Zankl

Fakultät für Mathematik, Informatik und Physik © UIBK
 Sommersemester 2011



Zusammenfassung der letzten LV

Definition

sei Σ ein endliches Alphabet. Wir definieren reguläre Ausdrücke induktiv

Basis

- | | | |
|---|---|-------------------------------|
| 1 | \emptyset ist ein regulärer Ausdruck (kurz: RA) | $L(\emptyset) := \emptyset$ |
| 2 | ϵ ist ein RA | $L(\epsilon) := \{\epsilon\}$ |
| 3 | für jedes Symbol a ist a ein RA | $L(a) := \{a\}$ |

Schritt

- | | | |
|---|--|------------------------------|
| 1 | für jeden RA E ist E^* ein RA | $L(E^*) := L(E)^*$ |
| 2 | für RAs E und F ist EF ein RA | $L(EF) := L(E) \cdot L(F)$ |
| 3 | für RAs E und F ist $E + F$ ein RA | $L(E + F) := L(E) \cup L(F)$ |
| 4 | wenn E ein RA ist, dann ist (E) ein RA | $L((E)) := L(E)$ |

Ein Algorithmus für Algebraische Gesetze

seien G, H beliebige reguläre Ausdrücke

Algorithmus

- 1 für alle Variablen in G und H : durch konkrete Symbole ersetzen (verschiedene Variablen durch verschiedene Symbole ersetzen)
- 2 Gleichung $L(G) = L(H)$ in eine Gleichung $L(C) = L(D)$ umwandeln
- 3 teste $L(C) = L(D)$
 - wenn **ja**, dann gilt $L(G) = L(H)$
 - wenn **nein**, dann gilt $L(G) \neq L(H)$

Lemma

der angegebene Test ist korrekt:

für jedes so gefundene Gesetz gilt $L(G) = L(H)$ für alle beliebigen regulären Ausdrücke an der Stelle von G bzw. H

Exkurs: Kleene Algebren

Definition

- 1 betrachte den idempotenten Halbring $(A, +, \cdot, 0, 1)$
- 2 sei auf A eine partielle Ordnung $a \leq b$ definiert:

$$a \leq b \Leftrightarrow a + x = b \text{ für } x \in A$$

- 3 definiere auf A die Operation $*$:

$$1 + aa^* \leq a^* \qquad b + ax \leq x \Rightarrow a^*b \leq x$$

$$1 + a^*a \leq a^* \qquad b + xa \leq x \Rightarrow ba^* \leq x$$

dann heißt die Struktur $\mathcal{A} = (A, +, \cdot, *, 0, 1)$ **Kleene Algebra**

Beispiel

- RAs formen die Kleene Algebra $(\Sigma^*, +, \cdot, *, \emptyset, \epsilon)$
- as heißt, wenn zwei RAs E, F die gleiche Sprache beschreiben, dann gilt in der entsprechenden Kleene Algebra die Gleichung $E = F$

Übersicht

Endliche Automaten

Automaten, reguläre Sprachen und Grammatiken, (nicht)-deterministische endliche Automaten, Teilmengenkonstruktion, ϵ -NEAs, **Umwandlung endlicher Automaten in reguläre Ausdrücke**, Pumpinglemma, Minimierung

Berechenbarkeitstheorie

Einführung in die Berechenbarkeitstheorie, Turingmaschinen, Entscheidungsprobleme, Universelle Maschinen und Diagonalisierung,

Komplexitätstheorie

Einführung in die Komplexitätstheorie, die Klassen P und NP, logarithmisch platzbeschränkte Reduktionen, Speicherplatzkomplexität

Reguläre Ausdrücke und Endliche Automaten

Satz

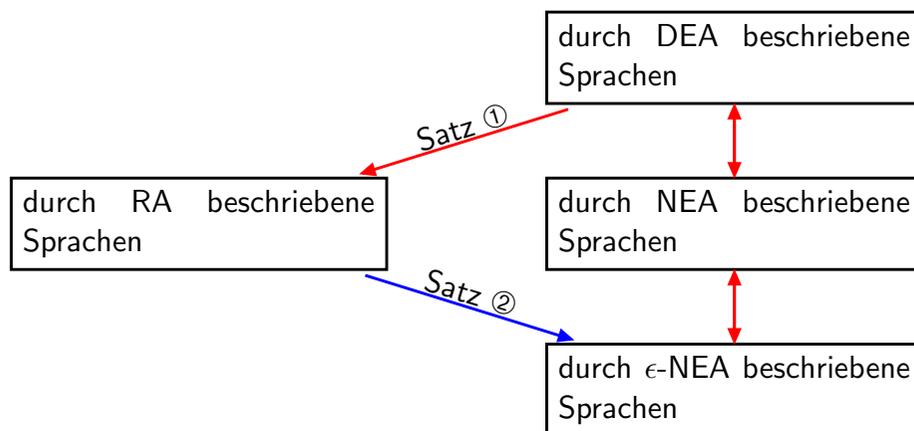
die folgenden Mengen sind gleich:

- die Menge der durch RA beschriebenen Sprachen
- die Menge der von DEA akzeptierten Sprachen
- die Menge der von NEA akzeptierten Sprachen
- die Menge der von ϵ -NEA akzeptierten Sprachen

Definition

diese Sprachen nennt man **reguläre Sprachen**

Reguläre Sprachen



EAs in reguläre Ausdrücke transformieren

Satz ①

sei A ein beliebiger DEA, dann

\exists regulären Ausdruck R , sodass $L(A) = L(R)$

Beweisansatz.

- Zustände von A in natürliche Zahlen umbenennen
- sei n die Anzahl der Zustände
- definiere regulären Ausdrücke:

$$R_{ij}^{(k)}$$

für alle $1 \leq i, j \leq n, 0 \leq k \leq n$

Sprache von $R_{ij}^{(k)}$ sind die Wörter, die A von Zustand i bis Zustand j lesen kann wobei nur Zustände $\leq k$ besucht werden

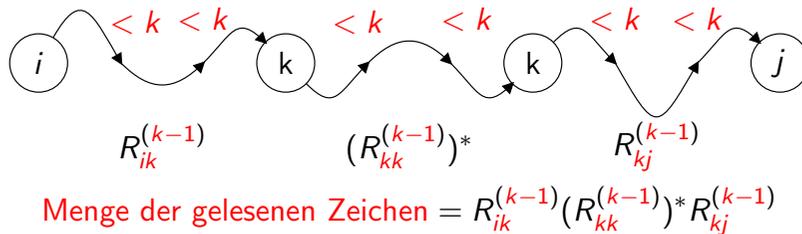
Beweisansatz (Fortsetzung).

- angenommen $R_{ij}^{(k)}$ ist schon definiert, für alle k
- angenommen der Startknoten des DEA A ist 1
- definiere RA R , sodass $L(A) = L(R)$:

$$R = R_{1m_1}^{(n)} + \dots + R_{1m_\ell}^{(n)}$$

wobei m_1, \dots, m_ℓ alle akzeptierenden Zustände in A

Beobachtung



Schritt

 $k > 0$

- 1 angenommen \exists Weg von i nach j , der nur durch Zustände $\leq k$ führt
- 2 zwei Fälle: ① k liegt auf dem Weg oder nicht ②

Fall ①

 k liegt auf dem Weg

- spalte den Pfad in **Teilstücke**, die k nicht passieren
- das Anfangsstück führt von i nach k
- die Mittelstücke führen von k nach k
- das Endstück von k nach j

Fall ②

 k liegt nicht auf dem Pfad

- verwende $R_{ij}^{(k-1)}$

in Summe erhalten wir:

$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)}(R_{kk}^{(k-1)})^*R_{kj}^{(k-1)}$$



definiere $R_{ij}^{(k)}$ induktiv

Basis

 $k = 0$

- 1 betrachte die **Wege der Länge ≤ 1**
- 2 unterscheide die Fälle $i \neq j$ und $i = j$

Fall $i \neq j$

betrachte die Kanten zwischen i und j

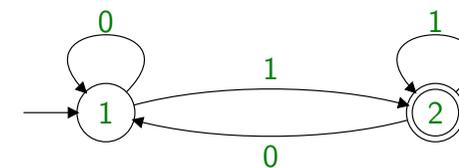
- wenn es **keine** solche **Kante**
setze $R_{ij}^{(0)} = \emptyset$
- wenn es **genau eine** solche **Kante** (i, a, j) gibt
setze $R_{ij}^{(0)} = a$
- wenn es **mehrere Kanten** $(i, a_1, j), \dots, (i, a_l, j)$ gibt
setze $R_{ij}^{(0)} = a_1 + \dots + a_l$

Fall $i = j$

- zusätzlich repräsentiere die **Wege der Länge 0** durch das Leerwort ϵ

Beispiel

gegeben der folgende DEA A



mit Hilfe des Verfahrens folgt $R = (\mathbf{0}^* \mathbf{1})(\mathbf{0}^* \mathbf{1})^* = L(A)$:

$$\begin{aligned} R_{12}^{(2)} &= R_{12}^{(1)} + R_{12}^{(1)}(R_{22}^{(1)})^*R_{22}^{(1)} \\ R_{12}^{(1)} &= R_{12}^{(0)} + R_{11}^{(0)}(R_{11}^{(0)})^*R_{12}^{(0)} \\ &= \mathbf{1} + (\mathbf{0} + \epsilon)(\mathbf{0} + \epsilon)^*\mathbf{1} = \mathbf{0}^*\mathbf{1} \\ R_{22}^{(1)} &= R_{22}^{(0)} + R_{21}^{(0)}(R_{11}^{(0)})^*R_{12}^{(0)} \\ &= (\mathbf{1} + \epsilon) + \mathbf{0}(\mathbf{0} + \epsilon)^*\mathbf{1} = \mathbf{0}^*\mathbf{1} + \epsilon \end{aligned}$$

Reguläre Ausdrücke in Endliche Automaten verwandeln

Satz ②

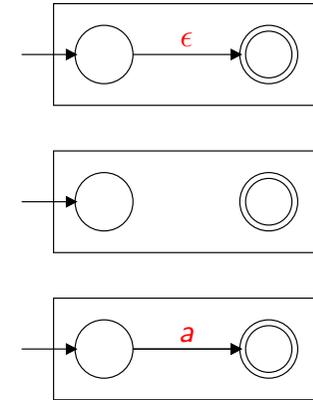
sei R ein beliebiger RA, dann \exists ϵ -NEA E , sodass $L(R) = L(E)$

Beweisansatz.

- angenommen $L = L(R)$, für einen RA R
- wir zeigen \exists ϵ -NEA E mit
 - 1 genau einem akzeptierenden Zustand
 - 2 keinen Kanten zum Startzustand
 - 3 keinen Kanten vom akzeptierenden Zustand
- $L = L(E)$
- wir verwenden Induktion **über reguläre Ausdrücke**

Basis

für die RA ϵ , \emptyset und a betrachten wir die folgenden 3 Automaten.



Induktionsschritt wir betrachten nur den Fall: $R \equiv (R')^*$

- nach Induktionshypothese existiert ein Automat für R'
- wir transformieren:

