

Diskrete Mathematik

Martin Avanzini Arne Dür Christoph Kollreider Georg Moser

Fakultät für Mathematik, Informatik und Physik @ UIBK
Sommersemester 2010



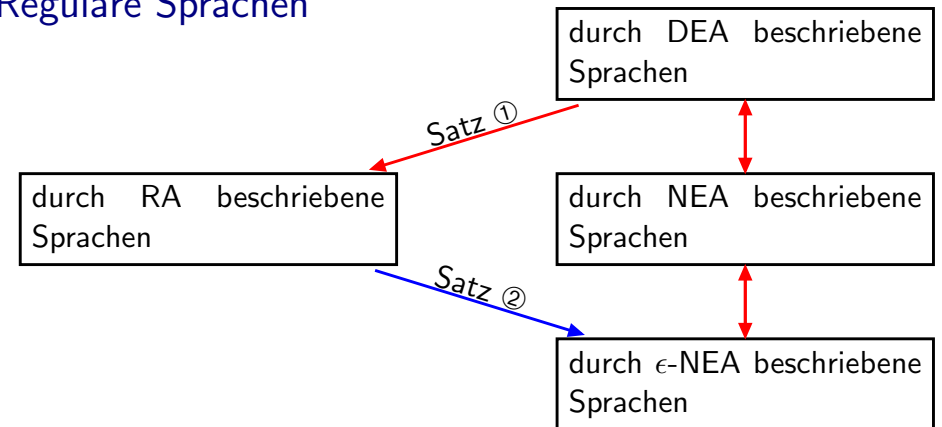
Satz ① DEA \rightarrow RA

Wenn für einen beliebigen DEA A : $L = L(A)$, dann
 \exists regulären Ausdruck R , sodass $L = L(R)$

Satz ② RA \rightarrow ϵ -NEA

Wenn für einen beliebigen RA R : $L = L(R)$, dann
 \exists ϵ -NEA E , sodass $L = L(E)$

Reguläre Sprachen



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.

Übersicht

Automaten, reguläre Sprachen und Grammatiken, (nicht)-deterministische endliche Automaten, Teilmengenkonstruktion, Automaten mit ϵ -Übergängen, **Umwandlung endlicher Automaten in reguläre Ausdrücke**, **Algebraische Gesetze für reguläre Ausdrücke**, Pumpinglemma, Minimierung

Einführung in die Berechenbarkeitstheorie, Turing Maschinen, Äquivalente Formulierungen, Entscheidungsprobleme, Universelle Maschinen und Diagonalisierung,

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

Reguläre Ausdrücke in Endliche Automaten verwandeln

Satz ②

 $RA \rightarrow \epsilon\text{-NEA}$

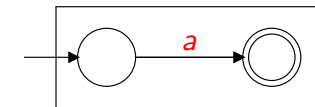
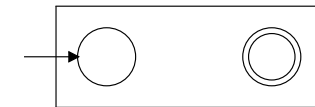
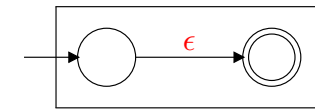
Wenn für einen beliebigen RA $R: L = L(R)$, dann
 $\exists \epsilon\text{-NEA } E$, sodass $L = L(E)$

Beweis

- angenommen $L = L(R)$, für einen RA R
- wir zeigen $\exists \epsilon\text{-NEA } E$ 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**

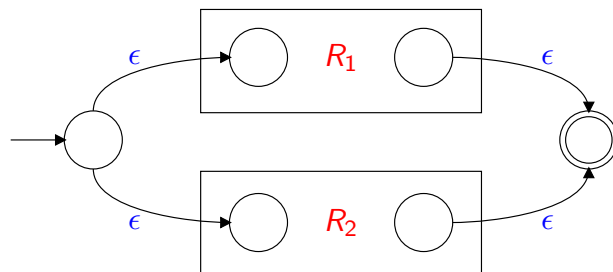
Basis

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



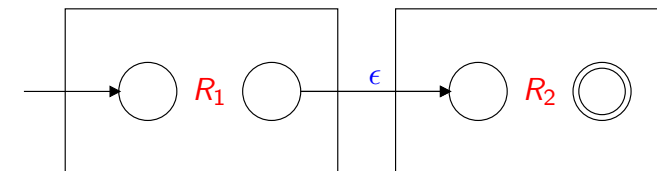
Induktionsschritt: $R \equiv R_1 + R_2$

- nach Induktionshypothese existieren Automaten für R_1 und R_2
- diese kombinieren wir:



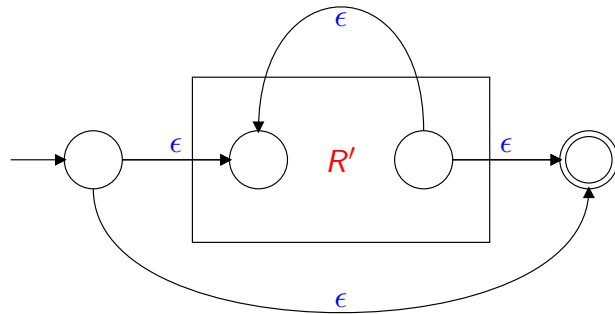
Induktionsschritt: $R \equiv R_1 R_2$

- nach Induktionshypothese existieren Automaten für R_1 und R_2
- diese kombinieren wir:



Induktionsschritt: $R \equiv (R')^*$

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



■

Algebraische Gesetze für reguläre Ausdrücke

Beispiel

$$0^*1 + (0^*1)(0^*1 + \epsilon)^*(0^*1) \equiv (0^*1)^+$$

seien L, M, N beliebige reguläre Ausdrücke

Assoziativität und Kommutativität

$$1 \quad L(L + M) = L(M + L)$$

Kommutativität von $+$

$$2 \quad L((L + M) + N) = L(L + (M + N))$$

Assoziativität von $+$

$$3 \quad L((LM)N) = L(L(MN))$$

Assoziativität der Verkettung

Erinnerung:

Kommutativität der Verkettung gilt nicht

Neutrales Element und Löscher

Definition

ein **neutrales Element** für einen Operator ist ein Element das die Operation nicht beeinflusst

Satz

$$1 \quad L(\emptyset + L) = L(L + \emptyset) = L(L) \quad \emptyset \text{ ist das neutrale Element für } +$$

$$2 \quad L(\epsilon L) = L(L\epsilon) = L(L) \quad \epsilon \text{ is das neutrale Element von } \cdot$$

$$\emptyset + 0^* \equiv 0^* \quad 0^* \equiv 0^* \epsilon$$

Definition

ein **Löscher** (Annihilator) für einen Operator ist ein Element das die Operation zunichte macht

Satz

$$1 \quad L(\emptyset L) = L(L\emptyset) = \emptyset \quad \emptyset \text{ ist ein Löscher der Verkettung}$$

$$\emptyset 0^* \equiv \emptyset \quad 0^* \emptyset \equiv \emptyset$$

Distributivgesetze und Idempotenzgesetz

Satz

Distributivgesetze von \cdot bezüglich $+$

$$1 \quad L(L(M + N)) = L(LM + LN)$$

$$2 \quad L((M + N)L) = L(ML + NL)$$

$$0 + 01^* \equiv 0\epsilon + 01^* \equiv 0(\epsilon + 1^*) \equiv 01^*$$

Satz

$$1 \quad L(L + L) = L(L)$$

Idempotenzgesetz von $+$

$$0^* + 0^* \equiv 0^*$$

Gesetze für den Kleene-Stern

Satz

$$1 \quad L(L^*) = L(L^* L^*) = L((L^*)^*)$$

$$2 \quad L(\emptyset^*) = L(\epsilon)$$

$$3 \quad L(L^+) = L(LL^*) = L(L^*L)$$

$$4 \quad L(L^*) = L(L^+ + \epsilon)$$

$$5 \quad L(L?) = L(\epsilon + L)$$

$$6 \quad L((E + F)^*) = L((E^* + F^*)^*) = \\ = L((E^*F^*)^*) = L((E^*F)^*E^*) = L(E^*(FE^*)^*)$$

$$(0^* + 1?)^* \equiv ((0^*)^*(1?)^*)^* \equiv (0^*(\epsilon + 1)^*)^* \equiv (0^*1^*)^*$$

Definition

Definition

Gesetze für den Kleene-Stern (2)

Lemma

$$L(L^*) = L(L^*L^*) = L(L^*)L(L^*) = L(L)^*L(L)^*$$

Beweis

zunächst zeigen wir: $L(L)^* \subseteq L(L)^*L(L)^*$:

- sei $x \in L(L)^*$, dann $x = x\epsilon \in L(L)^*L(L)^*$
- also folgt die Behauptung

zunächst zeigen wir: $L(L)^* \supseteq L(L)^*L(L)^*$:

- sei $x \in L(L)^*L(L)^*$
- $\exists y, z$ aus $L(L)^*$, sodass $x = yz$
- $\exists k, l$ sodass $y \in L(L)^k$, $z \in L(L)^l$ und $x \in L(L)^{k+l}$
- somit $x \in L(L)^*$ und die Behauptung folgt



Ein Algorithmus für Algebraische Gesetze

seien G, H seinen Schemata von regulären Ausdrücken

Algorithmus

- 1 \forall 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 auch $L(G) = L(H)$
 - Wenn **Nein**, dann ist $L(G) \neq L(H)$ falsch

Satz

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