Übersicht

Endliche Automaten

Äquivalenz der Automatenmodelle

Reguläre Ausdrücke
- Definition
- NEA aus regulärem Ausdruck
- Regulärer Ausdruck aus NEA

Pumping Lemma

Kontextfreie Sprachen

Pushdown-Automaten

Theorie für MI

Endliche Automaten
Äquivalenz der Automatenmodelle
Reguläre Ausdrücke
Definition
NEA aus regulärem Ausdruck
Regulärer Ausdruck aus NEA
Pumping Lemma
Kontextfreie Sprachen
Pushdown-Automaten

http://xkcd.com

Operationen auf Sprachen

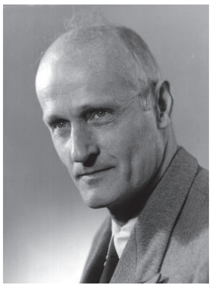
Endliche Automaten
Äquivalenz der Automatenmodelle
Reguläre Ausdrücke
Definition
NEA aus regulärem Ausdruck
Regulärer Ausdruck aus NEA
Pumping Lemma
Kontextfreie Sprachen
Pushdown-Automaten

Für Sprachen $L, L' \subseteq \Sigma^*$ definiere:

$$L \cdot L' := \{\, u \cdot v \,;\, u \in L \text{ und } v \in L' \,\}$$

Die Kleenesche Hülle L^* einer Sprache $L \subseteq \Sigma^*$ ist definiert:

$$L^0 := \{\epsilon\}$$
$$L^{i+1} := L^i \cdot L$$
$$L^* := \bigcup_{i \geq 0} L^i$$



L^* ist die Menge aller Wörter der Form $w = v_1 \ldots v_n$ mit $v_i \in L$ für alle i.

Die Konkatenation zweier Sprachen ist, ähnlich wie z.B. in der Analysis die Summe zweier Funktionen, *punktweise* definiert.

Die Kleenesche Hülle ist benannt nach Stephen Cole Kleene, einem Pionier der Mathematischen Logik und Theoretischen Informatik, der in den 50er Jahren die Entwicklung der Theorie von Automaten und formalen Sprachen entscheidend mitgeprägt hat.

Kleene'sche Hülle

Beispiel:

$$L = \{\, 01 \,,\, 110 \,\}$$

$$L^* = \big\{\, \epsilon \,,\, 01 \,,\, 110 \,,\, 0101 \,,\, 01110 \,,\, 11001 \,,\, 010101 \,,\, 110110 \,,\, 0101110 \,,\, 0111001 \,,\, 1100101 \,,\, 01010101 \,,\, \ldots \big\}$$

Eigenschaften:

- $\left(L^*\right)^* = L^*$
- $\emptyset^* = \{\epsilon\}$
- $\{\epsilon\}^* = \{\epsilon\}$
- L^* ist unendlich, außer in diesen 2 Fällen

Weiterhin gilt $L \subseteq L^*$ und die Monotonie-Eigenschaft: wenn $A \subseteq B$ ist, dann auch $A^* \subseteq B^*$. Zusammen mit der ersten Eigenschaft auf der Folie bedeutet dies, dass die Kleenesche Hülle ein Hülloperator im mathematischen Sinne ist.

Reguläre Ausdrücke

Endliche Automaten
Äquivalenz der Automatenmodelle
Reguläre Ausdrücke
Definition
NEA aus regulärem Ausdruck
Regulärer Ausdruck aus NEA
Pumping Lemma
Kontextfreie Sprachen
Pushdown-Automaten

Reguläre Ausdrücke und ihre Sprachen sind induktiv definiert:

- Konstanten ϵ und $\emptyset$ sind reguläre Ausdrücke, mit $L(\epsilon) = \{\epsilon\}$ und $L(\emptyset) = \emptyset$
- Für $a \in \Sigma$ ist a ein regulärer Ausdruck, mit $L(a) = \{a\}$
- Sind E, F reguläre Ausdrücke, dann auch $(E + F)$, mit $L(E + F) = L(E) \cup L(F)$
- Sind E, F reguläre Ausdrücke, dann auch $(E \cdot F)$, mit $L(E \cdot F) = L(E) \cdot L(F)$
- Ist E regulärer Ausdruck, dann auch E^*, mit $L(E^*) = L(E)^*$

Reguläre Ausdrücke sind Terme, die Sprachen, also Mengen von Wörtern beschreiben, ähnlich wie in der Mathematik beispielsweise Polynome Punktmengen wie Kurven oder Flächen beschreiben.

Abkürzende Schreibweisen

Endliche Automaten
Äquivalenz der Automatenmodelle
Reguläre Ausdrücke
Definition
NEA aus regulärem Ausdruck
Regulärer Ausdruck aus NEA
Pumping Lemma
Kontextfreie Sprachen
Pushdown-Automaten

- Bei mehreren $+$ werden Klammern weggelassen:

$$(a + b + c) \text{ statt } ((a + b) + c)$$

- Der $\cdot$ wird auch weggelassen:

$$rs \text{ statt } r \cdot s$$

- Auch bei $\cdot$ können Klammern weggelassen werden:

$$(abc) \text{ statt } ((ab)c)$$

- Mit Vorrangregeln: $*$ vor $\cdot$ vor $+$
können Klammern eingespart werden:

$$01 + 10^* \text{ statt } ((01) + (1(0^*)))$$

Beispiele

Endliche Automaten
Äquivalenz der Automatenmodelle
Reguläre Ausdrücke
Definition
NEA aus regulärem Ausdruck
Regulärer Ausdruck aus NEA
Pumping Lemma
Kontextfreie Sprachen
Pushdown-Automaten

- $(0 + 1)^*01(0 + 1)^*$
- $\left(b + c + ab + ac + aab\right)^*$
- $(b + c)^*a(a + b + c)^*a \; + \; (a + c)^*b(a + b + c)^*b$
$+ (a + b)^*c(a + b + c)^*c$

Regeln zum Vereinfachen

Endliche Automaten
Äquivalenz der Automatenmodelle
Reguläre Ausdrücke
Definition
NEA aus regulärem Ausdruck
Regulärer Ausdruck aus NEA
Pumping Lemma
Kontextfreie Sprachen
Pushdown-Automaten

- Kommutativgesetz

  $$(R + S) = (S + R)$$

- Neutrale Elemente

  $$\emptyset + R = R + \emptyset = R \qquad \epsilon R = R\epsilon = R$$

- Absorption

  $$\emptyset R = R\emptyset = \emptyset$$

- Distributivgesetze

  $$R(S + T) = RS + RT \qquad (S + T)R = SR + TR$$

- Gesetze über Kleene-Stern

  $$(R^*)^* = R^* \qquad \emptyset^* = \epsilon \qquad \epsilon^* = \epsilon$$

Reguläre Ausdrücke in der Praxis

Endliche Automaten
Äquivalenz der Automatenmodelle
Reguläre Ausdrücke
Definition
NEA aus regulärem Ausdruck
Regulärer Ausdruck aus NEA
Pumping Lemma
Kontextfreie Sprachen
Pushdown-Automaten

Reguläre Ausdrücke kommen z.B. in Skriptsprachen vor.

Dort wird eine reichhaltigere Syntax verwendet:

Notation:	steht für:
`[abcd]`	$(a + b + c + d)$
`[0-9]`	$(0 + 1 + \ldots + 9)$
`.`	beliebiges Symbol
`R|S`	$R + S$
`R*`	R^*
`R?`	$\epsilon + R$
`R+`	RR^*
`R[5]`	$RRRRR$

Endliche Automaten
Äquivalenz der Automatenmodelle
Reguläre Ausdrücke
Definition
NEA aus regulärem Ausdruck
Regulärer Ausdruck aus NEA
Pumping Lemma
Kontextfreie Sprachen
Pushdown-Automaten

Ausdrucksstärke

Wir werden im Folgenden zeigen:

Theorem

Reguläre Ausdrücke beschreiben genau die Sprachen, die von DEA (oder NEA, ϵ-NEA) erkannt werden.

Dazu zeigen wir zwei Teile:

Lemma

Für jeden regulären Ausdruck R gibt es einen ϵ-NEA A_R mit $L(A_R) = L(R)$.

Lemma

Für jeden NEA A gibt es einen regulären Ausdruck R_A mit $L(R_A) = L(A)$.

Die Klasse der Sprachen, die durch reguläre Ausdrücke beschreibbar und durch endliche Automaten erkennbar sind, nennt man die *regulären Sprachen*.

Verfahren zur Übersetzung regulärer Ausdrücke in endliche Automaten sind nicht nur von theoretischem Interesse, sondern finden Anwendungen z.B. bei Generatoren für lexikalische Scanner wie `flex`.

Vom regulären Ausdruck zum Automaten

Endliche Automaten
Äquivalenz der Automatenmodelle
Reguläre Ausdrücke
Definition
NEA aus regulärem Ausdruck
Regulärer Ausdruck aus NEA
Pumping Lemma
Kontextfreie Sprachen
Pushdown-Automaten

Für jeden regulären Ausdruck R definiere ϵ-NEA A_R mit $L(A_R) = L(R)$ und

- genau einem Endzustand,

$$F = \{q_f\}$$

- kein Übergang in den Startzustand,

$$q_0 \notin \delta(q, a) \quad \text{für alle } q \text{ und } a$$

- kein Übergang aus dem Endzustand,

$$\delta(q_f, a) = \emptyset \quad \text{für alle } a$$

Zum Beweis des ersten Lemmas müssen wir für jeden regulären Ausdruck R einen ϵ-NEA A_R konstruieren, der die Sprache von R erkennt. Die Konstruktion erfolgt induktiv gemäß der induktiven Definition von regulären Ausdrücken.

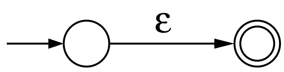
Wie häufig bei induktiven Argumenten zeigt man etwas stärkeres, damit der Induktionsschritt gelingen kann. Hier werden bei der Konstruktion von A_R zusätzlich die auf der Folie angegeben Invarianten gefordert.

Konstruktion des ϵ-NEA

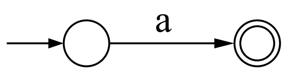
Konstruktion eines ϵ-NEA für regulären Ausdruck R:

$R = \emptyset$

$R = \epsilon$



$R = a$



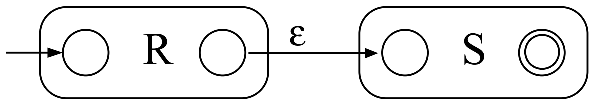
Zum Induktionsanfang werden explizit Automaten für die 3 Basisfälle der induktiven Definition von regulären Ausdrücken angegeben.

Die auf der Folie abgebildeten Automaten erfüllen offensichtlich die geforderten Invarianten, und erkennen die durch die entsprechenden Ausdrücke definierten Sprachen.

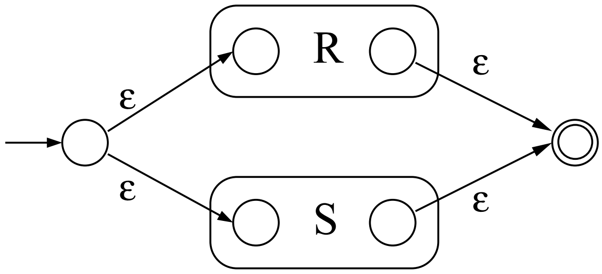
Konstruktion des ϵ-NEA 2

Konstruktion eines ϵ-NEA für regulären Ausdruck:

$R \cdot S$



$R + S$



Für den Induktionsschritt gibt es drei Fälle zu betrachten. Die Fälle für reguläre Ausdrücke $R \cdot S$ und $R + S$ sind auf dieser Folie, der für Ausdrücke der Form R^* auf der Nächsten.

Bei der Konstruktion der Automaten A_{RS} und A_{R+S} gehen wir davon aus, das wir nach Induktionshypothese bereits Automaten A_R und A_S haben, wie oben im Bild schematisch dargestellt.

Wir konstruieren daraus den Automaten A_{RS}, indem vom Endzustand von A_R ein ϵ-Übergang zum Anfangszustand von A_S eingefügt wird. Anfangszustand ist der von A_R, Endzustand der von A_S.

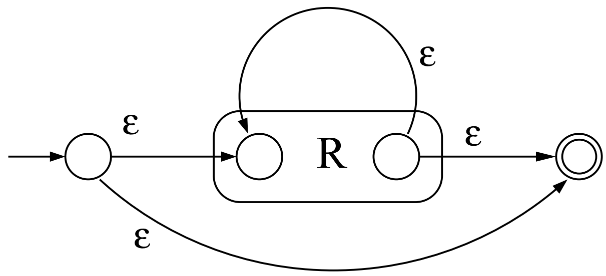
Um ein Wort zu akzeptieren, wird erst der Automat A_R zum Endzustand durchlaufen, dann wird in den Anfangszustand von A_S übergegangen, und von dort zum Endzustand von A_S gelaufen. Das Wort besteht also aus einem ersten Teil aus R, gefolgt von einem zweiten Teil aus S, ist also in RS.

Beim Automaten A_{R+S} werden ein neuer Anfangs- und Endzustand hinzugefügt und mit den Anfangs- bzw. Endzuständen von A_R und A_S durch ϵ-Übergänge verbunden. Vom Anfangszustand wird nichtdeterministisch in einen der Anfangszustände gesprungen, von dort wird der jeweilige Automat bis zu seinem Endzustand durchlaufen, von dem dann in den neuen Endzustand gesprungen wird. Ein akzeptiertes Wort ist also entweder in R oder in S, also in $R + S$.

Konstruktion des ϵ-NEA 3

Konstruktion eines ϵ-NEA für regulären Ausdruck:

R^*



Bei der Konstruktion von A_{R^*} wird zunächst der Endzustand von A_R mit dem Anfangszustand verbunden, so dass der Automat beliebig oft durchlaufen werden kann. Dadurch werden aber die Invarianten verletzt, so dass ein neuer Anfangs- und Endzustand hinzugefügt werden müssen. Schliesslich werden der neue Anfangs- und Endzustand mit einem ϵ-Übergang verbunden, so dass in jedem Fall das leere Wort akzeptiert wird.

Ein akzeptiertes Wort ist also leer, oder es besteht aus beliebig vielen Teilen aus R hintereinander, ist also in R^*.

Das Arden'sche Lemma

Seien $U, V \subseteq \Sigma^*$ Sprachen mit $\epsilon \notin U$.

Erfüllt $L \subseteq \Sigma^*$ die Gleichung $\quad L = UL + V$

dann gilt: $\quad L = U^*V$

Erfüllt $L \subseteq \Sigma^*$ die Gleichung $\quad L = UL$

dann gilt: $\quad L = \emptyset$

Die zweite Behauptung ist ein Spezialfall der ersten, mit $V = \emptyset$.
Dann ist nämlich $L = U^*\emptyset = \emptyset$.

Zum Beweis der Behauptung sei L gegeben mit $L = UL + V$.
Wir zeigen zuerst $U^*V \subseteq L$. Durch Induktion nach $|w|$ zeigen wir für alle $w \in U^*V$, dass $w \in L$ ist.
Induktionsanfang: Ist $w \in U^*V$ mit $|w|$ minimal, so ist $w \in V$, also $w \in L$ da $V \subseteq L$.
Induktionsschritt: Es ist entweder $w \in V$, also ebenfalls $w \in L$, oder sonst $w = uv$ für ein $u \in U$ und $v \in U^*V$. Da nach Voraussetzung $\epsilon \notin U$ gilt, ist $|v| < |w|$, und somit nach Induktionshypothese $v \in L$, also ist $w = uv \in UL \subseteq L$, also $w \in L$.

Die Umkehrung $L \subseteq U^*V$ wird analog gezeigt:
Induktionsanfang: Ist $w \in L$ mit $|w|$ minimal, so ist $w \in V$, also $w \in U^*V$.
Induktionsschritt: Es ist entweder $w \in V$, also ebenfalls $w \in L$, oder sonst $w = uv$ für ein $u \in U$ und $v \in L$. Da $|v| < |w|$, ist nach Induktionshypothese $v \in U^*V$, also auch $w = uv \in U^*V$.

Regulärer Ausdruck aus NEA

Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein NEA.

Für $q \in Q$ definiere: $L_q := \{\, w \,;\, \hat{\delta}(q, w) \cap F \neq \emptyset \,\}$

Also ist $L(A) = L_{q_0}$.

L_q erfüllt die Gleichung:

$$L_q = \sum_{a \in \Sigma} \sum_{p \in \delta(q,a)} a\, L_p$$

$$L_q = \sum_{a \in \Sigma} \sum_{p \in \delta(q,a)} a\, L_p + \epsilon \qquad \text{für } q \in F$$

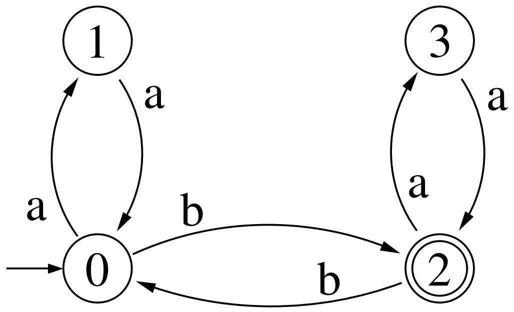
$\rightsquigarrow$ Gleichungssystem für die L_q,

kann mit Arden's Lemma gelöst werden.

Ein (rekursives) Gleichungssystem, das für jeden Zustand eine Gleichung enthält, kann direkt aus der Beschreibung des Automaten abgelesen und mit dem Lemma von Arden gelöst werden.

Die Korrektheit des Gleichungssystems ist offensichtlich. Das Beispiel auf der nächsten Folie verdeutlicht die Konstruktion und zeigt, wie so ein Gleichungssystem gelöst wird. Auf eine formale Beschreibung des Lösungsverfahrens verzichten wir.

Beispiel



Gleichungssystem zum NEA A:

$$\begin{aligned} L_0 &= a\,L_1 + b\,L_2 \\ L_1 &= a\,L_0 \\ L_2 &= a\,L_3 + b\,L_0 + \epsilon \\ L_3 &= a\,L_2 \end{aligned}$$

Lösung: $L(A) = L_0 = \big(aa + b(aa)^* b\big)^* b(aa)^*$

Einsetzen der vierten in die dritte Gleichung liefert:

$$L_2 = aa\,L_2 + b\,L_0 + \epsilon$$

also liefert das Lemma von Arden:

$$L_2 = (aa)^*(b\,L_0 + \epsilon) = (aa)^* b\,L_0 + (aa)^*$$

Einsetzen der zweiten in die erste Gleichung liefert:

$$L_0 = aa\,L_0 + b\,L_2$$

und Einsetzen der obigen Lösung für L_2 ergibt:

$$\begin{aligned} L_0 &= aa\,L_0 + b\big((aa)^* b\,L_0 + (aa)^*\big) \\ &= aa\,L_0 + b(aa)^* b\,L_0 + b(aa)^* \\ &= \big(aa + b(aa)^* b\big) L_0 + b(aa)^* \end{aligned}$$

also liefert das Arden'sche Lemma die Lösung:

$$L_0 = \big(aa + b(aa)^*\, b\big)^* b(aa)^*$$