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 DEA 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 endiche Automaten erkennbar sind, nennt man die *regulären Sprachen*.

Vom regulären Ausdruck zum 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.

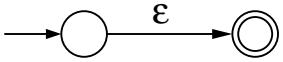
Wie häufig bei induktiven Argumenten zeigt man etwas stärkeres, damit der Induktionsschritt gelingen kann. Hier werden bei der Konstruktion von A_R zusätzlich die auf der Folie angegeben Invarianten gefordert.

Konstruktion des ϵ-NEA

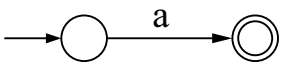
Konstruktion eines ϵ-NEA für regulären Ausdruck R:

$R = \emptyset$

$R = \epsilon$



$R = \mathbf{a}$



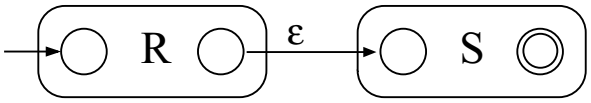
Zum Induktionsanfang werden explizit Automaten für die 3 Basisfälle der induktiven Definition von regulären Ausdrücken angegeben.

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

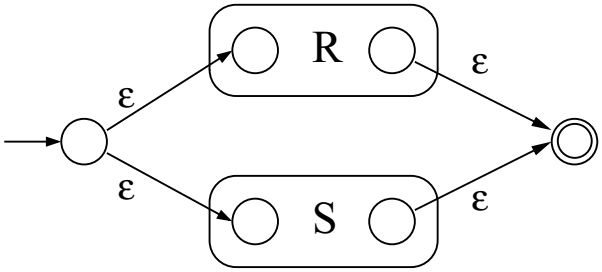
Konstruktion des ϵ-NEA 2

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.

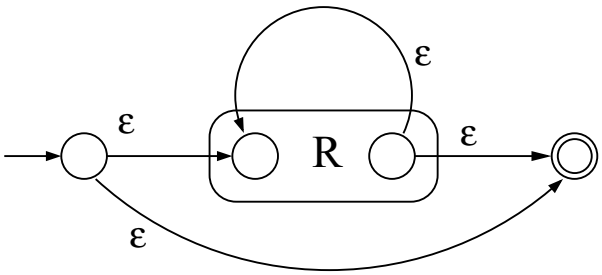
Um ein Wort zu akzeptieren, wird erst der Automat A_R zum Endzustand durchlaufen, dann wird in den Anfangszustand von A_S Übergegangen, und von dort zum Endzustand von A_S gelaufen. Das Wort besteht aslso aus einem ersten Teil aus R, gefolgt von einem zweiten Teil aus S, ist also in RS.

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^*.

Regulärer Ausdruck aus DEA

Sei $A = (Q, \Sigma, \delta, q_1, F)$ ein DEA mit $Q = \{q_1, \ldots, q_n\}$.

Für $i, j, k \leq n$ definiere reguläre Ausdrücke $R_{i,j}^{(k)}$
mit $w \in L(R_{i,j}^{(k)})$ gdw.

- $\hat{\delta}(q_i, w) = q_j$,
- für alle $\epsilon \prec v \prec w$ ist $\hat{\delta}(q_i, v) \in \{q_1, \ldots, q_k\}$.

Dann ist für $F = \{q_{j_1}, \ldots, q_{j_m}\}$

$$L(A) = R_{1,j_1}^{(n)} + \ldots + R_{1,j_m}^{(n)}$$

Es sollen induktiv reguläre Ausdrücke konstruiert werden, die mehr und mehr der Abläufe von A beschreiben.

Dazu betrachtet man die Mengen der Wörter, die von einem Zustand q_i zu einem anderen q_j führen, und dabei auf dem Weg immer mehr verschiedene Zwischenzustände benutzen können.

Der Ausdruck $R_{i,j}^{(k)}$ beschreibt dementsprechend diejenigen Wörter, für die der Automat in q_j endet, wenn er in q_i startet (also $\hat{\delta}(q_i, w) = j$), und wo auf dem Weg dazwischen nur die Zustände q_1 bis q_k besucht werden. Diese werden durch Induktion über k definiert.

Die Konstruktion ähnelt dem Algorithmus von Floyd und Warshall für das *All-pairs-shortest-path*-Problem.

Induktive Konstruktion: Anfang

Definiere $R_{i,j}^{(k)}$ induktiv nach k.

Induktionsanfang: $k = 0$

Seien $\{a_1, \ldots a_m\}$ diejenigen $a_\ell \in \Sigma$ mit $\delta(q_i, a_\ell) = q_j$.

Fall 1: $i = j$

$$R_{i,i}^{(0)} = (\epsilon + a_1 + \ldots + a_m)$$

Fall 2: $i \neq j$

$$R_{i,j}^{(0)} = (a_1 + \ldots + a_m)$$

Beim Induktionsanfang $k = 0$ sind diejenigen Wörter in $R_{i,j}^{(0)}$, mit denen ohne Zwischenzustand direkt von q_i in q_j übergegangen wird.

Für $i \neq j$ sind dies die Wörter der Länge 1, also Symbole, für die es einen Übergang von q_i nach q_j gibt. Für $i = j$ kommt zusätzlich noch das leere Wort hinzu.

Induktive Konstruktion: Schritt

Für ein $w \in R_{i,j}^{(k+1)}$ gibt es zwei Möglichkeiten:

- der Zustand q_{k+1} kommt auf dem Weg nicht vor.

  $\rightsquigarrow w \in R_{i,j}^{(k)}$
- der Zustand q_{k+1} kommt mindestens einmal vor.

  $\rightsquigarrow$ w kann zerlegt werden $w = w_1 w_2 \dots w_{m-1} w_m$ mit
  - $w_1 \in R_{i,k+1}^{(k)}$
  - $w_m \in R_{k+1,j}^{(k)}$
  - $w_2, \dots, w_{m-1} \in R_{k+1,k+1}^{(k)}$

Regulärer Ausdruck:

$$R_{i,j}^{(k+1)} = R_{i,j}^{(k)} + R_{i,k+1}^{(k)} \cdot \left(R_{k+1,k+1}^{(k)}\right)^* \cdot R_{k+1,j}^{(k)}$$

Beim Induktionsschritt betrachten wir Wörter, die von q_i zu q_j führen, und dabei nur Zwischenzustände bis q_{k+1} verwenden.

Kommt auf dem Weg der Zustand nicht vor, dann ist das Wort schon in $R_{i,j}^{(k)}$.

Andernfalls zerfällt es in einen vorderen Teil, der von q_i zum ersten Vorkommen von q_{k+1} führt – der ist in $R_{i,k+1}^{(k)}$ – und einen hinteren Teil, der vom letzten Vorkommen von q_{k+1} zu q_j führt – der ist in $R_{k+1,j}^{(k)}$. Dazwischen liegen beliebig viele Teile, die von einem Vorkommen von q_{k+1} zum nächsten führen – diese liegen in $R_{k+1,k+1}^{(k)}$.

Damit erhält man insgesamt den angegebenen Ausdruck.

Reguläre Ausdrücke in der Praxis

Reguläre Ausdrücke kommen z.B. in Skriptsprachen vor.

Dort wird eine reichhaltigere Syntax verwendet:

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

Regeln zum Vereinfachen

- Kommutativgesetz

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

- Neutrale Elemente

  $\emptyset+R=R+\emptyset=R$ $\quad$ $\epsilon R=R\epsilon=R$

- Absorption

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

- Distributivgesetze

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

- Gesetze über Kleene-Stern

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