

Algorithmen mit konstantem Platzbedarf: Die Klasse REG

Sommerakademie Rot an der Rot — AG 1
Wieviel Platz brauchen Algorithmen wirklich?

Daniel Alm

Institut für Numerische Simulation
Universität Bonn

August 2010

Gliederung

1 Endliche Automaten

Motivation

Wie können wir eine Sprache beschreiben?

Arbeitsweise eines endlichen Automaten

Formale Definition

Das Pumping-Lemma

Nichtdeterministische Automaten

2 Reguläre Ausdrücke

Reguläre Sprachen

Reguläre Ausdrücke

Beispiele

Endliche Automaten

Motivation

- Erfüllt ein Wort bestimmte Kriterien, enthält es z.B. genau drei a's?
- Enthält ein bestimmter Text eine Email-Adresse?
- Wie können wir einen einfachen Computer simulieren?
 - Eingabe
 - Verarbeitung
 - Ausgabe
- Ausgabe in diesem Fall: Ja/Nein, d.h. wir betrachten Entscheidungsprobleme.

Was ist eine Sprache?

Definition (Wiederholung)

- *Wort über Σ* : endliche Folge von Symbolen aus einem endlichen, nichtleeren Alphabet Σ . Das leere Wort bezeichnen wir mit λ .
- *Sprache über Σ* : Menge von Worten über Σ
- Σ^* : Sprache aller Worte über Σ

Was ist eine Sprache?

Definition (Wiederholung)

- *Wort über Σ* : endliche Folge von Symbolen aus einem endlichen, nichtleeren Alphabet Σ . Das leere Wort bezeichnen wir mit λ .
- *Sprache über Σ* : Menge von Worten über Σ
- Σ^* : Sprache aller Worte über Σ

Wie schreiben wir eine Sprache auf?

- Alle Worte in der Sprache: $\{\lambda, a, b, aa, ab, bb, \dots\}$
- Beschreibung der Menge: $\{a^n b^m \mid n, m \in \mathbb{N}\}$
- **Test jedes Wortes mit Hilfe eines endlichen Automaten**
- Erzeugen aus einer *Grammatik*
- Beschreibung mittels *regulärer Ausdrücke*

Arbeitsweise eines endlichen Automaten

- Beginne in einem Anfangszustand q_0 und erhalte ein Wort als Eingabe.
- Lese jetzt jeden einzelnen Buchstaben und bestimme jedes Mal einen neuen Zustand.
- Wenn das Wort gelesen ist, entscheide an Hand des aktuellen Zustands, ob das Wort *akzeptiert* oder *verworfen* wird.

Definition (deterministischer endlicher Automat, DFA)

Tupel $M = (\Sigma, Q, q_0, Q_a, \delta)$, bestehend aus

- Eingabealphabet Σ
- endliche Zustandsmenge Q
- Startzustand $q_0 \in Q$
- Menge $Q_a \subseteq Q$ von akzeptierenden Zuständen
- Zustandsübergangsfunktion

$$\delta : Q \times \Sigma \rightarrow Q.$$

Deterministisch, weil sein Ergebnis durch ein bestimmtes Eingabewort vollständig bestimmt ist.

Algorithmus zur Berechnung

Daten : endlicher Automat M wie oben, Wort w über dem Alphabet Σ

Ergebnis : **true** falls der Automat w akzeptiert, sonst **false**

begin

setze $q := q_0$;

for $i := 1$ **to** $|w|$ **do**

└ setze $q := \delta(q, w[i])$;

if $q \in Q_a$ **then return true**;

else return false;

Definition

- *erweiterte Zustandsübergangsfunktion* $\delta^* : Q \times \Sigma^*$:

$$\delta^*(q, \lambda) = q$$

$$\delta^*(q, xw) = \delta^*(\delta(q, x), w)$$

$\delta^*(q, w)$ entspricht dem Zustand des Automaten nach dem Lesen des Wortes w , wenn er vorher im Zustand q war.

- *akzeptiertes Wort* w : $\delta^*(q_0, w) \in Q_a$
- *akzeptierte Sprache* $L(M)$: $\{w \in \Sigma^* \mid \delta^*(q_0, w) \in Q_a\}$

Beispiel

Ein Automat, der die Sprache $\{a^n b^m \mid n, m \in \mathbb{N}\}$ akzeptiert:

$$\delta(q_0, a) = q_0$$

$$\delta(q_0, b) = q_1$$

$$\delta(q_1, a) = q_2$$

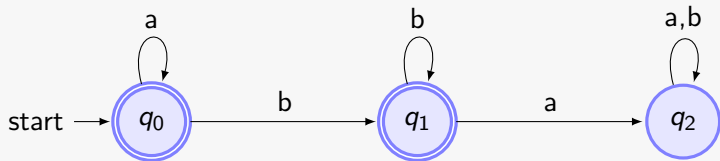
$$\delta(q_1, b) = q_1$$

$$\delta(q_2, a) = q_2$$

$$\delta(q_2, b) = q_2$$

Die akzeptierenden Zustände sind q_0 und q_1 .

In Graphdarstellung:



Einschränkungen von DFAs

- DFAs sind vergesslich.
- Sie können nur endlich viele Zustände annehmen, aber beliebig lange Eingangsworte lesen.
- \Rightarrow Zustände werden mehrfach angenommen.
- Wiederholt man das Teilwort zwischen zwei identischen Zuständen (“Aufpumpen”), so ändert dies nichts am Ergebnis.

Lemma (Pumping-Lemma)

- Sei L eine von einem DFA akzeptierte Sprache.
- Dann existiert eine Wortlänge n , sodass:
- Für alle Wörter w der Länge mindestens n existiert eine Zerlegung $w = x \circ y \circ z$ mit $|y| \geq 1$ und $|x \circ y| \leq n$, sodass:
- Für alle $i \in \mathbb{N}$ gilt

$$x \circ y^i \circ z \in L.$$

Beweis

- Wähle $n := |Q|$ und sei $w \in L$ beliebig mit $|w| \geq n$.
- Sei $q^{(0)}, q^{(1)}, \dots, q^{(|w|)}$ die Folge der vom L akzeptierenden Automaten durchlaufenen Zustände.
- Mindestens einer der ersten n durchlaufenen Zustände muss zweimal in der Folge auftauchen, z.B. bei $q^{(i)}$ und $q^{(j)}$, $0 \leq i < j \leq n$.
- Wähle $x := w[1]w[2] \cdots w[i]$, $y := w[i+1] \cdots w[j]$ und $z := w[j+1] \cdots w[|w|]$.
- Nach Wahl von x und y ist

$$q^{(i)} = \delta^*(q^{(0)}, x) = \delta^*(q^{(0)}, x \circ y) = \delta^*(q^{(i)}, y) = \delta^*(q^{(i)}, y^k)$$

für beliebige $k \in \mathbb{N}$.

Anwendung

Hinweise

- Nur ein notwendiges Kriterium, nicht hinreichend:
 $\{0^n 1^m 0^m \mid n, m \in \mathbb{N}\} \cup \{1^n 0^m \mid n, m \in \mathbb{N}\}$ hat die Pump-Eigenschaft, kann aber nicht durch einen DFA akzeptiert werden.
- Umkehrung: wenn für alle n ein n langes Wort existiert, das nicht “aufpumpbar” ist, dann ist L nicht durch einen DFA akzeptiertbar.

Anwendung

Hinweise

- Nur ein notwendiges Kriterium, nicht hinreichend:
 $\{0^n 1^m 0^m \mid n, m \in \mathbb{N}\} \cup \{1^n 0^m \mid n, m \in \mathbb{N}\}$ hat die Pump-Eigenschaft, kann aber nicht durch einen DFA akzeptiert werden.
- Umkehrung: wenn für alle n ein n langes Wort existiert, das nicht “aufpumpbar” ist, dann ist L nicht durch einen DFA akzeptiertbar.

Beispiel

$L := \{a^k b^k \mid k \in \mathbb{N}\}$ lässt sich nicht durch einen DFA akzeptieren.

- Sei $1 \leq n \in \mathbb{N}$.
- Betrachte $w := a^n b^n$.
- Mögliche Zerlegungen: $y = a^p$, $1 \leq p \leq n$.
- $x \circ y^2 \circ z = a^{n+p} b^n \notin L$.

Übung

Welche der folgenden Sprachen sind durch einen DFA akzeptierbar?

Übung

Welche der folgenden Sprachen sind durch einen DFA akzeptierbar?

- $\{w \in \{a, b\}^* \mid w \text{ enthält mindestens drei mehr a's als b's}\}$

Übung

Welche der folgenden Sprachen sind durch einen DFA akzeptierbar?

- $\{w \in \{a, b\}^* \mid w \text{ enthält mindestens drei mehr a's als b's}\}$
nicht akzeptierbar

Übung

Welche der folgenden Sprachen sind durch einen DFA akzeptierbar?

- $\{w \in \{a, b\}^* \mid w \text{ enthält mindestens drei mehr a's als b's}\}$
nicht akzeptierbar
- $\{w \in \{a, \dots, z, A, \dots, Z\}^* \mid w \text{ steht im Duden}\}$

Übung

Welche der folgenden Sprachen sind durch einen DFA akzeptierbar?

- $\{w \in \{a, b\}^* \mid w \text{ enthält mindestens drei mehr a's als b's}\}$
nicht akzeptierbar
- $\{w \in \{a, \dots, z, A, \dots, Z\}^* \mid w \text{ steht im Duden}\}$
akzeptierbar: diese Sprache ist endlich

Übung

Welche der folgenden Sprachen sind durch einen DFA akzeptierbar?

- $\{w \in \{a, b\}^* \mid w \text{ enthält mindestens drei mehr a's als b's}\}$
nicht akzeptierbar
- $\{w \in \{a, \dots, z, A, \dots, Z\}^* \mid w \text{ steht im Duden}\}$
akzeptierbar: diese Sprache ist endlich
- $\{w \in \{a, b, c, d\}^* \mid w \text{ hat endliche Länge}\}$

Übung

Welche der folgenden Sprachen sind durch einen DFA akzeptierbar?

- $\{w \in \{a, b\}^* \mid w \text{ enthält mindestens drei mehr a's als b's}\}$
nicht akzeptierbar
- $\{w \in \{a, \dots, z, A, \dots, Z\}^* \mid w \text{ steht im Duden}\}$
akzeptierbar: diese Sprache ist endlich
- $\{w \in \{a, b, c, d\}^* \mid w \text{ hat endliche Länge}\}$
akzeptierbar: ist einfach $\{a, b, c, d\}^*$

Übung

Welche der folgenden Sprachen sind durch einen DFA akzeptierbar?

- $\{w \in \{a, b\}^* \mid w \text{ enthält mindestens drei mehr a's als b's}\}$
nicht akzeptierbar
- $\{w \in \{a, \dots, z, A, \dots, Z\}^* \mid w \text{ steht im Duden}\}$
akzeptierbar: diese Sprache ist endlich
- $\{w \in \{a, b, c, d\}^* \mid w \text{ hat endliche Länge}\}$
akzeptierbar: ist einfach $\{a, b, c, d\}^*$
- $\{w \in \{0, 1\}^* \mid |w| \text{ ist quadratisch}\}$

Übung

Welche der folgenden Sprachen sind durch einen DFA akzeptierbar?

- $\{w \in \{a, b\}^* \mid w \text{ enthält mindestens drei mehr a's als b's}\}$
nicht akzeptierbar
- $\{w \in \{a, \dots, z, A, \dots, Z\}^* \mid w \text{ steht im Duden}\}$
akzeptierbar: diese Sprache ist endlich
- $\{w \in \{a, b, c, d\}^* \mid w \text{ hat endliche Länge}\}$
akzeptierbar: ist einfach $\{a, b, c, d\}^*$
- $\{w \in \{0, 1\}^* \mid |w| \text{ ist quadratisch}\}$
nicht akzeptierbar

Übung

Welche der folgenden Sprachen sind durch einen DFA akzeptierbar?

- $\{w \in \{a, b\}^* \mid w \text{ enthält mindestens drei mehr a's als b's}\}$
nicht akzeptierbar
- $\{w \in \{a, \dots, z, A, \dots, Z\}^* \mid w \text{ steht im Duden}\}$
akzeptierbar: diese Sprache ist endlich
- $\{w \in \{a, b, c, d\}^* \mid w \text{ hat endliche Länge}\}$
akzeptierbar: ist einfach $\{a, b, c, d\}^*$
- $\{w \in \{0, 1\}^* \mid |w| \text{ ist quadratisch}\}$
nicht akzeptierbar
- $\{w \in \{0, 1, \dots, 9\}^* \mid |w| \text{ ist durch 3 teilbar}\}$

Übung

Welche der folgenden Sprachen sind durch einen DFA akzeptierbar?

- $\{w \in \{a, b\}^* \mid w \text{ enthält mindestens drei mehr a's als b's}\}$
nicht akzeptierbar
- $\{w \in \{a, \dots, z, A, \dots, Z\}^* \mid w \text{ steht im Duden}\}$
akzeptierbar: diese Sprache ist endlich
- $\{w \in \{a, b, c, d\}^* \mid w \text{ hat endliche Länge}\}$
akzeptierbar: ist einfach $\{a, b, c, d\}^*$
- $\{w \in \{0, 1\}^* \mid |w| \text{ ist quadratisch}\}$
nicht akzeptierbar
- $\{w \in \{0, 1, \dots, 9\}^* \mid |w| \text{ ist durch 3 teilbar}\}$
akzeptierbar

Übung

Welche der folgenden Sprachen sind durch einen DFA akzeptierbar?

- $\{w \in \{a, b\}^* \mid w \text{ enthält mindestens drei mehr a's als b's}\}$
nicht akzeptierbar
- $\{w \in \{a, \dots, z, A, \dots, Z\}^* \mid w \text{ steht im Duden}\}$
akzeptierbar: diese Sprache ist endlich
- $\{w \in \{a, b, c, d\}^* \mid w \text{ hat endliche Länge}\}$
akzeptierbar: ist einfach $\{a, b, c, d\}^*$
- $\{w \in \{0, 1\}^* \mid |w| \text{ ist quadratisch}\}$
nicht akzeptierbar
- $\{w \in \{0, 1, \dots, 9\}^* \mid |w| \text{ ist durch 3 teilbar}\}$
akzeptierbar
- $\{w \in \{0, 1\}^* \mid w \text{ ist eine durch 7 teilbare Binärzahl}\}$

Übung

Welche der folgenden Sprachen sind durch einen DFA akzeptierbar?

- $\{w \in \{a, b\}^* \mid w \text{ enthält mindestens drei mehr a's als b's}\}$
nicht akzeptierbar
- $\{w \in \{a, \dots, z, A, \dots, Z\}^* \mid w \text{ steht im Duden}\}$
akzeptierbar: diese Sprache ist endlich
- $\{w \in \{a, b, c, d\}^* \mid w \text{ hat endliche Länge}\}$
akzeptierbar: ist einfach $\{a, b, c, d\}^*$
- $\{w \in \{0, 1\}^* \mid |w| \text{ ist quadratisch}\}$
nicht akzeptierbar
- $\{w \in \{0, 1, \dots, 9\}^* \mid |w| \text{ ist durch 3 teilbar}\}$
akzeptierbar
- $\{w \in \{0, 1\}^* \mid w \text{ ist eine durch 7 teilbare Binärzahl}\}$
akzeptierbar: addiere Dreiergruppen modulo 7 ($x \cdot 8 \equiv x \pmod{7}$)

Was passiert, wenn wir “Zufall” zulassen?

- Nach dem Lesen eines Symbols kann der Automat in einen von mehreren Zuständen wechseln.
- Die Berechnung ist nicht wirklich *zufällig*, zu einem Eingabewort sind verschiedene Berechnungen möglich – aber wir wissen schon vorher, welche möglich sind!
- In jeder *Berechnung* wird der Automat nur einmal ausgeführt und das Wort nur einmal gelesen.
- Dieses Konstrukt erscheint zunächst mächtiger als ein deterministischer Automat. Ist es aber nicht!

Definition (nichtdeterministischer endlicher Automat, NFA)

Tupel $M = (\Sigma, Q, q_0, Q_a, \Delta)$, bestehend aus

- Eingabealphabet Σ
- endliche Zustandsmenge Q
- Startzustand $q_0 \in Q$
- Menge $Q_a \subseteq Q$ von akzeptierenden Zuständen
- Zustandsübergangsrelation

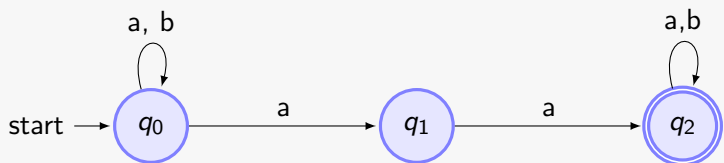
$$\Delta \subseteq (Q \times \Sigma) \times Q.$$

Wir können auch ein Tupel aus $Q \times \Sigma$ auf einen *Powerzustand* $\subseteq Q$ abbilden, der die möglichen neuen Zustände enthält.

Wir sagen, der Automat *akzeptiert* ein Wort w , wenn er nach dem Lesen von w in einem akzeptierenden Zustand sein *kann*.

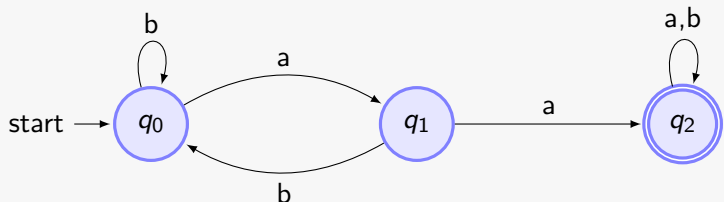
Beispiel

NFA, der die Sprache $\{u \circ aa \circ v \mid u, v \in \{a, b\}^*\}$ akzeptiert:



Beachte, dass ein Übergang weggelassen wurde!

Als DFA:



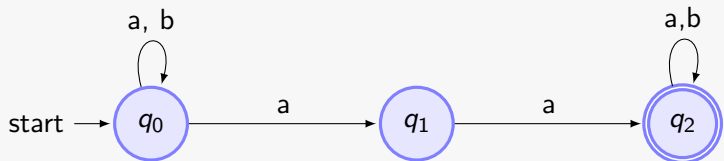
NFAs sind nicht mächtiger als DFAs!

Konstruiere zu einem NFA $M = (\Sigma, Q, q_0, Q_a, \Delta)$ einen DFA, den *Potenzmengenautomat* $M' = (\Sigma, Q', q'_0, Q'_a, \delta')$:

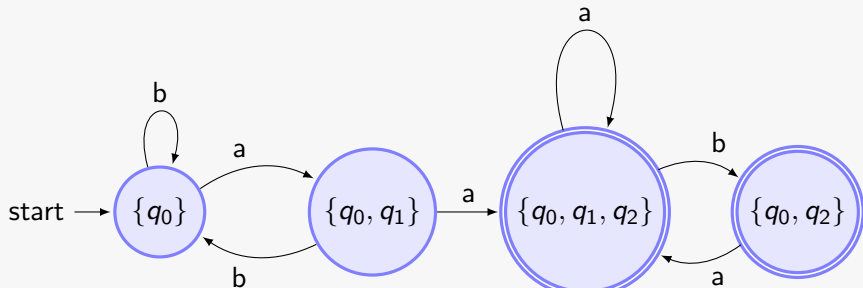
- $Q' := \{P \mid P \subseteq Q\}$
- $q'_0 := \{q_0\}$
- $Q'_a := \{P \in Q' \mid P \cap Q_a \neq \emptyset\}$
- $\delta'(P, x) := \{q \in Q \mid \exists p \in P : (p, x, q) \in \Delta\}$

Dann ist der nach dem Lesen eines Wortes w von M' angenommene Powerzustand genau die Menge aller Zustände, die M nach Lesen von w angenommen haben kann.

Beispiel



Ist als Potenzmengenautomat:



Reguläre Ausdrücke

Motivation

- Für jede Sprache einen kompletten endlichen Automaten aufzuschreiben ist ziemlich umständlich.
- Verwende eine andere Schreibweise: setze eine Sprache aus vielen einfacheren Sprachen rekursiv zusammen.
- Für diese einfachen Sprachen definiert man sich bestimmte Ausdrücke, die man dann zu komplizierten Ausdrücken zusammensetzt.
- Dafür brauchen wir zunächst eine andere Klassifikation der von DFAs akzeptierbaren Sprachen: die regulären Sprachen.

Was ist eine Sprache?

Definition (Wiederholung)

- *Wort über Σ* : endliche Folge von Symbolen aus einem endlichen, nichtleeren Alphabet Σ . Das leere Wort bezeichnen wir mit λ .
- *Sprache über Σ* : Menge von Worten über Σ
- Σ^* : Sprache aller Worte über Σ

Wie schreiben wir eine Sprache auf?

- Alle Worte in der Sprache: $\{\lambda, a, b, aa, ab, bb, \dots\}$
- Beschreibung der Menge: $\{a^n b^m \mid n, m \in \mathbb{N}\}$
- Test jedes Wortes mit Hilfe eines endlichen Automaten
- Erzeugen aus einer *Grammatik*
- Beschreibung mittels *regulärer Ausdrücke*

Definition (Grammatik)

Tupel (N, T, S, P) , bestehend aus

- *Nonterminalalphabet* N
- *Terminalalphabet* T , $N \cap T = \emptyset$
- *Startsymbol* $S \in N$
- Endliche Menge P von *Ersetzungsregeln*

$$l \rightarrow r, \quad l, r \in (N \cup T)^*.$$

l enthält mindestens ein Nonterminal.

Definition (Grammatik)

Tupel (N, T, S, P) , bestehend aus

- *Nonterminalalphabet* N
- *Terminalalphabet* T , $N \cap T = \emptyset$
- *Startsymbol* $S \in N$
- Endliche Menge P von *Ersetzungsregeln*

$$l \rightarrow r, \quad l, r \in (N \cup T)^*.$$

l enthält mindestens ein Nonterminal.

Reguläre Grammatik

- Alle Regeln bilden ein einzelnes Nonterminal auf eine Folge von Terminalen (mit höchstens einem Nonterminal am Ende) ab.

Beispiel

Reguläre Grammatik, die die Sprache $L := \{a^n b^m \mid n, m \in \mathbb{N}\}$ erzeugt, d.h. L enthält genau die Worte aus T^* , die mit Hilfe von Ersetzungsregeln in P aus S erzeugt werden:

- $N := \{S, B\}$
- $T := \{a, b\}$
- Ersetzungsregeln in P :
 - $S \rightarrow aS$
 - $S \rightarrow B$
 - $B \rightarrow bB$
 - $B \rightarrow \lambda$

Eine von einer regulären Grammatik erzeugte Sprache heißt *regulär*. Alle von DFAs erzeugten Sprachen sind regulär.

Beweis (1)

- Sei M ein endlicher Automat mit akzeptierter Sprache $L(M)$. Wir wählen $N := Q$, $T := \Sigma$ und $S := q_0$.
- Die Ableitungsregeln konstruieren wir aus der Zustandsfunktion: Ist $\delta(q, x) = q'$, so fügen wir die Regel $q \rightarrow xq'$ hinzu.
- Zusätzlich dürfen akzeptierende Zustände das Wort beenden: $q \rightarrow \lambda$ für $q \in Q_a$.

Beweis (2)

- Sei w ein von diesem Automaten akzeptiertes Wort mit durchlaufenen Zuständen $q^{(0)}, q^{(1)}, \dots, q^{(|w|)}$. Dann ist $\delta(q^{(i)}, w[i+1]) = q^{(i+1)}$ für $0 \leq i \leq |w| - 1$, d.h. es gibt die Regeln $q^{(i)} \rightarrow w[i+1]q^{(i+1)}$ und zusätzlich $q^{(|w|)} \rightarrow \lambda$.
- Sei umgekehrt w von der oben angegebenen Grammatik erzeugt, und zwar mit der Ableitungskette

$$q^{(0)} \Rightarrow w_1 \Rightarrow \dots \Rightarrow w_n = w.$$

Dann enden alle w_i außer dem letzten auf ein Nonterminal (d.h. einen Zustand), und diese Zustandsfolge entspricht den Zuständen des Automaten beim Lesen von w .

Beweis (2)

- Sei w ein von diesem Automaten akzeptiertes Wort mit durchlaufenen Zuständen $q^{(0)}, q^{(1)}, \dots, q^{(|w|)}$. Dann ist $\delta(q^{(i)}, w[i+1]) = q^{(i+1)}$ für $0 \leq i \leq |w| - 1$, d.h. es gibt die Regeln $q^{(i)} \rightarrow w[i+1]q^{(i+1)}$ und zusätzlich $q^{(|w|)} \rightarrow \lambda$.
- Sei umgekehrt w von der oben angegebenen Grammatik erzeugt, und zwar mit der Ableitungskette

$$q^{(0)} \Rightarrow w_1 \Rightarrow \dots \Rightarrow w_n = w.$$

Dann enden alle w_i außer dem letzten auf ein Nonterminal (d.h. einen Zustand), und diese Zustandsfolge entspricht den Zuständen des Automaten beim Lesen von w .

Bemerkung

Die Umkehrung gilt auch, d.h. alle regulären Sprachen sind von DFAs akzeptierbar. Dies werden wir allerdings nicht zeigen.

Definition (reguläre Ausdrücke)

Rekursiv definierte Menge von Worten über dem Alphabet $\Sigma \cup \{ (,), |, * \}$:

- λ ist ein regulärer Ausdruck.
- Für jedes $x \in \Sigma$ ist x ein regulärer Ausdruck.
- Sind R_1 und R_2 reguläre Ausdrücke, so auch $R_1 R_2$ und $(R_1 | R_2)$.
- Ist R ein regulärer Ausdruck, so auch $(R)^*$.

Definition (reguläre Ausdrücke)

Rekursiv definierte Menge von Worten über dem Alphabet $\Sigma \cup \{(\ , \) , | , * \}$:

- λ ist ein regulärer Ausdruck.
- Für jedes $x \in \Sigma$ ist x ein regulärer Ausdruck.
- Sind R_1 und R_2 reguläre Ausdrücke, so auch $R_1 R_2$ und $(R_1 | R_2)$.
- Ist R ein regulärer Ausdruck, so auch $(R)^*$.

Von R beschriebene Sprache $L(R)$

- $L(\lambda) = \{\lambda\}$
- $L(x) = \{x\}$
- $L(R_1 R_2) = L(R_1) \circ L(R_2)$
- $L((R_1 | R_2)) = L(R_1) \cup L(R_2)$
- $L((R)^*) = L(R)^*$

In der Praxis

Häufig werden in regulären Ausdrücken vereinfachte Schreibweisen zugelassen:

- \wedge entspricht einem Zeilenanfang, $\$$ einem Zeilenende.
- $.$ kann ein beliebiges Zeichen sein.
- $[abc]$ entspricht $(a|b|c)$, $[\wedge abc]$ “jedem Zeichen außer a, b oder c”.
- Statt $(0|1|\dots|9|.|_)$ kann man auch $[0-9._]$ schreiben.
- $R?$ entspricht *höchstens einem*, R^+ *mindestens einem* Vorkommen von R .
- $R\{a,b\}$ bedeutet, dass R mindestens a und höchstens b mal vorkommt.

Ein Beispiel: Datumsangaben

```
^(0?[1-9] | [12] [0-9] | 3[01])\. (0?[1-9] | 1[012])\. [0-9]+$
```

Ein Beispiel: Datumsangaben

```
^(0?[1-9] | [12] [0-9] | 3[01])\.(0?[1-9] | 1[012])\.[0-9]+$
```

Und noch eins: Email-Adressen

```
^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$
```

Zusammenfassung

- Konstanter Platz reicht aus, um viele Worte zu erkennen und innerhalb von Zeichenfolgen zu finden.
- Schon Nachzählen und beliebige Verschachtelungen sind normalerweise nicht mehr möglich.
- Reguläre Ausdrücke sind auf Grund ihrer Flexibilität sehr wichtig in der Verarbeitung von Texten.
- Mit Hilfe von regulären Ausdrücken kann man einen Ausdruck für *alle möglichen Datumsangaben* hinschreiben, ohne alle einzeln aufzählen zu müssen.
- Mit ein paar Erweiterungen lassen sich z.B. Tag, Monat und Jahr automatisch aus der Datumsangabe auslesen und getrennt abspeichern.

Danke!