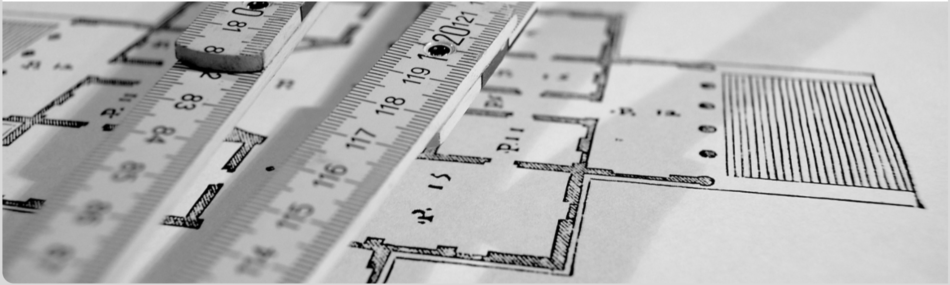
Grundbegriffe der Informatik
Tutorium 36

Termin 13 | 03.02.2017

Thassilo Helmold

KIT – Karlsruher Institut für Technologie



Inhalt

Reguläre Ausdrücke

Rechtslineare Grammatiken

Strukturelle Induktion

Abbildung: https://www.xkcd.com/1724/

In the previous episode of GBI...

Rückblick: Laufzeitbetrachtungen

- Das Master-Theorem

Rückblick: Endliche Automaten

- Mealy- und Moore-Automaten
- Formale Definition und graphische Repräsentation
- f, f_*, f_{**}
- g, g_*, g_{**}
- Endliche Akzeptoren

Wahr oder Falsch?

- Das (komplizierte) Master-Theorem kann man immer anwenden.

Wahr oder Falsch?

- Das (komplizierte) Master-Theorem kann man immer anwenden. F
  Nur bei rekursiven Algorithmen, bei denen das Problem in gleich große Teilprobleme aufgeteilt wird.

Wahr oder Falsch?

- Das (komplizierte) Master-Theorem kann man immer anwenden. F
  Nur bei rekursiven Algorithmen, bei denen das Problem in gleich große Teilprobleme aufgeteilt wird.
- Jeder Moore-Automat kann in einen Mealy-Automat gewandelt werden, der für jedes Wort die gleiche Ausgabe produziert.

Wahr oder Falsch?

- Das (komplizierte) Master-Theorem kann man immer anwenden. F
  Nur bei rekursiven Algorithmen, bei denen das Problem in gleich große Teilprobleme aufgeteilt wird.
- Jeder Moore-Automat kann in einen Mealy-Automat gewandelt werden, der für jedes Wort die gleiche Ausgabe produziert. F
  Für das leere Wort kann ein Mealy-Automat niemals eine Ausgabe produzieren.

Wahr oder Falsch?

- Das (komplizierte) Master-Theorem kann man immer anwenden. F
  Nur bei rekursiven Algorithmen, bei denen das Problem in gleich große Teilprobleme aufgeteilt wird.
- Jeder Moore-Automat kann in einen Mealy-Automat gewandelt werden, der für jedes Wort die gleiche Ausgabe produziert. F
  Für das leere Wort kann ein Mealy-Automat niemals eine Ausgabe produzieren.
- Endliche Akzeptoren sind Moore-Automaten mit dem Ausgabealphabet $\{0, 1\}$

Wahr oder Falsch?

- Das (komplizierte) Master-Theorem kann man immer anwenden. F
  Nur bei rekursiven Algorithmen, bei denen das Problem in gleich große Teilprobleme aufgeteilt wird.
- Jeder Moore-Automat kann in einen Mealy-Automat gewandelt werden, der für jedes Wort die gleiche Ausgabe produziert. F
  Für das leere Wort kann ein Mealy-Automat niemals eine Ausgabe produzieren.
- Endliche Akzeptoren sind Moore-Automaten mit dem Ausgabealphabet $\{0, 1\}$ W

Wahr oder Falsch?

- Das (komplizierte) Master-Theorem kann man immer anwenden. F
  Nur bei rekursiven Algorithmen, bei denen das Problem in gleich große Teilprobleme aufgeteilt wird.
- Jeder Moore-Automat kann in einen Mealy-Automat gewandelt werden, der für jedes Wort die gleiche Ausgabe produziert. F
  Für das leere Wort kann ein Mealy-Automat niemals eine Ausgabe produzieren.
- Endliche Akzeptoren sind Moore-Automaten mit dem Ausgabealphabet $\{0, 1\}$ W
- Mit endlichen Automaten kann jede beliebige Sprache erkannt werden

Wahr oder Falsch?

- Das (komplizierte) Master-Theorem kann man immer anwenden. F
  Nur bei rekursiven Algorithmen, bei denen das Problem in gleich große Teilprobleme aufgeteilt wird.
- Jeder Moore-Automat kann in einen Mealy-Automat gewandelt werden, der für jedes Wort die gleiche Ausgabe produziert. F
  Für das leere Wort kann ein Mealy-Automat niemals eine Ausgabe produzieren.
- Endliche Akzeptoren sind Moore-Automaten mit dem Ausgabealphabet $\{0, 1\}$ W
- Mit endlichen Automaten kann jede beliebige Sprache erkannt werden F Tatsächlich ist die Menge der akzeptierbaren Sprachen sogar sehr eingeschränkt.

Übung: Akzeptoren

Geben Sie graphisch einen endlichen Akzeptor über dem Alphabet $A = \{a, b\}$ an, der folgende Sprache akzeptiert:

- Die leere Menge
- Die Menge des leeren Worts
- Die Menge aller Worte, die genau ein b enthalten.
- Die Menge aller Worte, bei denen die Anzahl der b durch 3 teilbar ist.

Übung: Akzeptoren

Die Sprache $L \subseteq \{a, b\}^*$ sei definiert als die Menge aller Wörter w, die folgende Bedingungen erfüllen

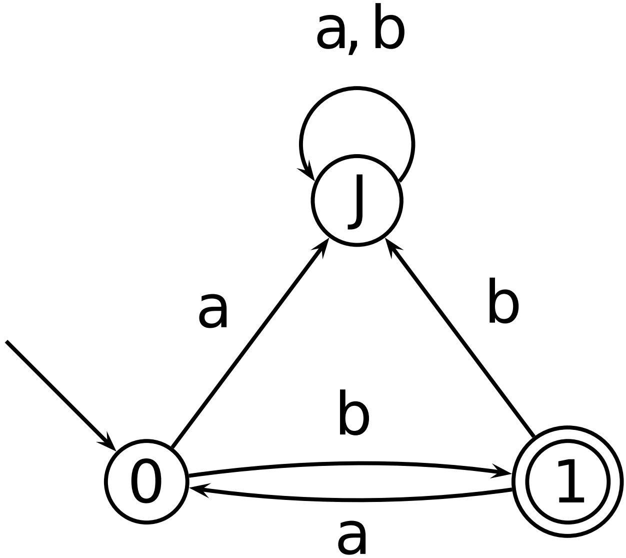
$$N_b(w) > N_a(w)$$
$$\forall v_1, v_2 \in \{a, b\}^* : w \neq v_1 bb v_2$$

Geben Sie einen endlichen Akzeptor an, der L erkennt.

Tipps

- Die 2. Bedingung bedeutet: Das Wort darf nirgends zwei b hintereinander enthalten.
- Was passiert, wenn das Wort mit einem a startet?
  Kann das Wort noch akzeptiert werden?

Übung: Akzeptoren: Lösung



Grenzen endlicher Automaten

Gibt es einen endlichen Akzeptor mit

$$L = \{a^k b^k | k \in \mathbb{N}_0\}$$

Nein! Warum nicht?
Gibt es einen endlichen Akzeptor, der alle gültigen Klammerausdrücke erkennt?
Nein, aus dem selben Grund.

Abbildung: `https://www.xkcd.com/1144/`

Kontextfreie Grammatiken „können also mehr“ als endliche Automaten.
Wir wollen nun ein „gleichmächtiges“ Konzept.

Reguläre Ausdrücke

Rechtslineare Grammatiken

Strukturelle Induktion

Reguläre Ausdrücke

Wir können uns reguläre Ausdrücke zusammenbauen aus

- den einzelnen Symbolen x aus A
- zwei regulären Ausdrücken R_1 und R_2 mit

$$(R_1 R_2) \qquad \text{oder} \qquad (R_1 \mid R_2)$$

- einem Stern $R*$
- oder dem leeren Ausdruck

Klammern dürfen nach den Klammerregeln weggelassen werden:
Stern vor Punkt vor Strich.

Reguläre Ausdrücke

Beispiel

Sei $A = \{a, b, c\}$. Dann sind gültige reguläre Ausdrücke:

$\mathtt{abc}$

$\mathtt{a \mid b \mid c}$

$\mathtt{(ab)*}$

$\emptyset *$

Sprache eines Ausdruckes

Die durch R beschriebene Sprache $\langle R \rangle$ ist wie folgt definiert:

- $\langle \emptyset \rangle = \emptyset$
- $\langle x \rangle = \{x\}$ $(x \in A)$
- $\langle R_1 \mid R_2 \rangle = \langle R_1 \rangle \cup \langle R_2 \rangle$
- $\langle R_1 R_2 \rangle = \langle R_1 \rangle \cdot \langle R_2 \rangle$
- $\langle R* \rangle = \langle R \rangle^*$

Beispiel

- $\langle \mathrm{a} \rangle = \{\mathrm{a}\}$
- $\langle \mathrm{ab} \rangle = \langle \mathrm{a} \rangle \cdot \langle \mathrm{b} \rangle = \{\mathrm{ab}\}$
- $\langle \mathrm{a} \mid \mathrm{b} \rangle = \langle \mathrm{a} \rangle \cup \langle \mathrm{b} \rangle = \{\mathrm{a}, \mathrm{b}\}$.
- $\langle (\mathrm{a} \mid \mathrm{b}) * \rangle = \langle \mathrm{a} \mid \mathrm{b} \rangle^* = \{\mathrm{a}, \mathrm{b}\}^*$.
- $\langle (\mathrm{a}{*}\mathrm{b}{*}) * \rangle = \langle \mathrm{a}{*}\mathrm{b}{*} \rangle^* = (\langle \mathrm{a}{*} \rangle \langle \mathrm{b}{*} \rangle)^* = (\langle \mathrm{a} \rangle^* \langle \mathrm{b} \rangle^*)^* = (\{\mathrm{a}\}^* \{\mathrm{b}\}^*)^*$
  $= \{\mathrm{a}, \mathrm{b}\}^*$.

Aufgabe: Reguläre Ausdrücke

In dieser Aufgabe geht um die formalen Sprachen

$$L_1 = \{a^k b^m \mid k, m \in \mathbb{N}_0\} \qquad L_2 = \{b^k a^m \mid k, m \in \mathbb{N}_0\}$$

Geben Sie für jede der folgenden formalen Sprachen L je einen regulären Ausdruck R an mit $\langle R \rangle = L$.

- $L = L_1 \cup L_2$
- $L = L_1 \cap L_2$
- $L = L_1 \cdot L_2$
- $L = L_1 *$

Aufgabe: Reguläre Ausdrücke

In dieser Aufgabe geht um die formalen Sprachen

$$L_1 = \{a^k b^m \mid k, m \in \mathbb{N}_0\} \qquad L_2 = \{b^k a^m \mid k, m \in \mathbb{N}_0\}$$

Geben Sie für jede der folgenden formalen Sprachen L je einen regulären Ausdruck R an mit $\langle R \rangle = L$.

- $L = L_1 \cup L_2$ $\quad a * b* \mid b * a*$
- $L = L_1 \cap L_2$ $\quad a* \mid b*$
- $L = L_1 \cdot L_2$ $\quad a * b * b * a*$ oder $a * b * a*$
- $L = L_1*$ $\quad (a * b*)*$ oder $(a \mid b)*$

Aufgabe: Sprachen regulärer Ausdrücke

- $\langle \texttt{(a|b)*abb(a|b)*} \rangle =$
- $\langle \texttt{a**} \rangle$
- $\langle ??? \rangle = \langle R \rangle^+$
- $\langle ??? \rangle = \{\varepsilon\}$
- $\langle ??? \rangle = \{\, w \in \{a, b\}^* \mid |w|_b > 2\}$
- $\langle ??? \rangle =$ Sprache aller Wörter über a, b, in denen das Teilwort ab nicht vorkommt.

Lösung

- $\langle (\mathtt{a}\,|\,\mathtt{b})*\mathtt{abb}(\mathtt{a}\,|\,\mathtt{b})*\rangle = \{\mathtt{a}, \mathtt{b}\}^* \cdot \{\mathtt{abb}\} \cdot \{\mathtt{a}, \mathtt{b}\}^*$
- $\langle \mathtt{a}**\rangle = \{a\}^*$
- $\langle R(R)*\rangle = \langle R\rangle^+$
- $\langle \emptyset*\rangle = \{\varepsilon\}$
- $\langle \mathtt{a}*\mathtt{ba}*\mathtt{ba}*\mathtt{b}(\mathtt{a}\,|\,\mathtt{b})*\rangle = \{\, w \in \{a, b\}^* \mid |w|_b > 2\}$
- $\langle \mathtt{b}*\mathtt{a}*\rangle =$ Sprache aller Wörter über a, b, in denen das Teilwort ab nicht vorkommt.

Reguläre Ausdrücke

Rechtslineare Grammatiken

Strukturelle Induktion

Rechtslineare Grammatiken

Definition

Eine Grammatik $G = (N, T, S, P)$ nennt man **rechtslinear** wenn bei jeder Produktion auf der rechten Seite höchstens ein Nichtterminalsymbol und dieses nur als letztes Symbol steht.

Alle Produktionen folgen dem Schema

$$X \to w \quad \text{oder} \quad X \to wY$$

mit $w \in T^*, \; X, Y \in N$.

Reguläre Sprachen

Satz

Für jede formale Sprache L sind die folgenden drei Aussagen äquivalent:

- *L kann von einem endlichen Akzeptor erkannt werden.*
- *L kann durch einen regulären Ausdruck beschrieben werden.*
- *L kann von einer rechtslinearen Grammatik erzeugt werden.*

Eine solche Sprache nennen wir **regulär**.

Beispiele für Umwandlungen

Siehe Übung 13, WS 15/16

Beispiele

$G = (\{X, Y, Z\}, \{a, b\}, X, P)$ mit

$$P = \{X \to aX \mid bY \mid \varepsilon, Y \to aX \mid bZ \mid \varepsilon, Z \to aZ \mid bZ\}$$

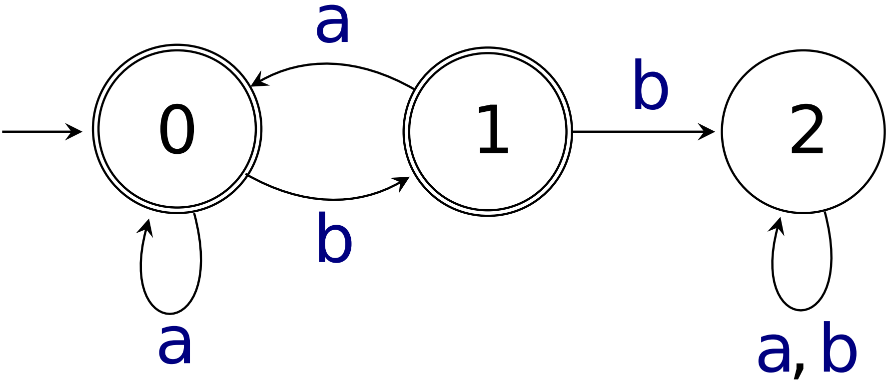
ist eine rechtslineare Grammatik. Die Sprache ist

$$L(G) = \{w \mid \forall v_1, v_2 \in \{a, b\}^* : w \neq v_1 bb v_2\}$$

Der reguläre Ausdruck ist

$$R = (a \mid ba) * (b \mid \emptyset *)$$

der Automat ist


0
1
2
a
b
b
a
a, b

Jetzt sieht man vielleicht auch

$$G = (\{X\}, \{a, b\}, X, P)$$

mit

$$P = \{X \to aX \mid baX \mid b \mid \varepsilon\}$$

Noch mehr Beispiele

- $G = (\{X\}, \{a, b\}, X, \{X \to abX \mid bbaX \mid \varepsilon\}$ wird beschrieben durch $L(G) = \langle \qquad\qquad \rangle$
- $G = (\{X, Y\}, \{a, b\}, X, \{X \to aX \mid bX \mid ababbY, Y \to aY \mid bY \mid \varepsilon\}$ wird beschrieben durch $L(G) = \langle \qquad\qquad\qquad\qquad \rangle$

Noch mehr Beispiele

- $G = (\{X\}, \{a, b\}, X, \{X \to abX \mid bbaX \mid \varepsilon\}$ wird beschrieben durch $L(G) = \langle (ab \mid bba) * \rangle$
- $G = (\{X, Y\}, \{a, b\}, X, \{X \to aX \mid bX \mid ababbY, Y \to aY \mid bY \mid \varepsilon\}$ wird beschrieben durch $L(G) = \langle (a \mid b) * ababb(a \mid b) * \rangle$

Aufgabe

Gegeben ist im folgenden jeweils eine Beschreibung einer formalen Sprache L und ein dazugehöriges Alphabet. Schreiben Sie jeweils den regulären Ausdruck R auf, für den $L(R) = L$ gilt und stellen Sie eine rechtslineare Grammatik G auf, für die $L(G) = L$ gilt:

- Die Menge aller Worte über dem Alphabet $A = \{a, b, c\}$, die genau ein c enthalten.

- Die Menge aller Worte über dem Alphabet $A = \{a, b\}$, bei denen die Anzahl der b durch 3 teilbar ist.

Aufgabe

Gegeben ist im folgenden jeweils eine Beschreibung einer formalen Sprache L und ein dazugehöriges Alphabet. Schreiben Sie jeweils den regulären Ausdruck R auf, für den $L(R) = L$ gilt und stellen Sie eine rechtslineare Grammatik G auf, für die $L(G) = L$ gilt:

- Die Menge aller Worte über dem Alphabet $A = \{a, b, c\}$, die genau ein c enthalten.
  Lösung: $(a|b) * c(a|b)*$
- Die Menge aller Worte über dem Alphabet $A = \{a, b\}$, bei denen die Anzahl der b durch 3 teilbar ist.
  Lösung: $a * (ba * ba * ba*)*$

Aufgabe

Gegeben sei die rechtslineare Grammatik

$$G = (\{S\}, \{a, b\}, S, P) \qquad P = \{S \rightarrow baaS | baS | aaS | \varepsilon\}$$

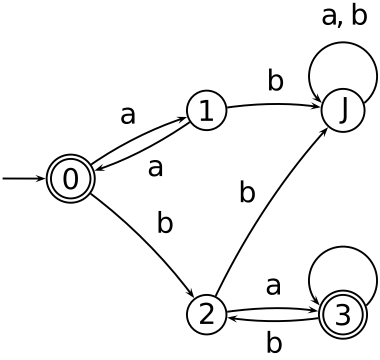
- Geben Sie einen endlichen Akzeptor A an, so dass $L(A) = L(G)$ gilt
- Geben Sie einen regulären Ausdruck R an, so dass $\langle R \rangle = L(G)$ gilt
- Geben Sie einen regulären Ausdruck R an, der nicht das Zeichen | enthält, und für den $\langle R \rangle = L(G)$ gilt.

Aufgabe

Gegeben sei die rechtslineare Grammatik

$$G = (\{S\}, \{a, b\}, S, P) \qquad P = \{S \to baaS|baS|aaS|\varepsilon\}$$

- Geben Sie einen endlichen Akzeptor A an, so dass $L(A) = L(G)$ gilt



Aufgabe

Gegeben sei die rechtslineare Grammatik

$$G = (\{S\}, \{a, b\}, S, P) \qquad P = \{S \to baaS|baS|aaS|\varepsilon\}$$

- Geben Sie einen regulären Ausdruck R an, so dass $\langle R \rangle = L(G)$ gilt

$$(baa|ba|aa)*$$

- Geben Sie einen regulären Ausdruck R an, der nicht das Zeichen | enthält, und für den $\langle R \rangle = L(G)$ gilt.

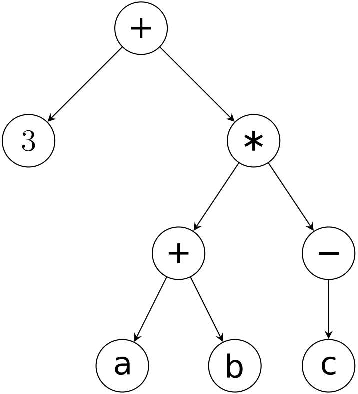
$$(aa) * (baa*)*$$

Reguläre Ausdrücke

Rechtslineare Grammatiken

Strukturelle Induktion

Man kann syntaktische Ausdrücke als Kantorowitsch-Baum schreiben.
Beispiel: einfacher arithmetischer Ausdruck: $3 + (a + b) \cdot (-c)$



Aussagen mit z.B. regulären Ausdrücken, Produktionsmengen o.Ä. lassen sich am Besten über **strukturelle Induktion beweisen** Dabei zeigt man:

Induktionsanfang Die Aussage stimmt für die Wurzel / jedes Atom

Induktionsvoraussetzung Die Aussage stimmt für einen Teilbaum / einen Teilausdruck

Induktionsschluss Die Aussage stimmt für jede Verzweigung / jeden Produktionsschritt

Aufgabe

Die Menge $M \subset \mathbb{N}_0$ sei definiert durch:

- 5 und 8 liegen in M.
- Für alle m, n gilt: Wenn n und m in M liegen, dann ist auch $n^2 + m^2$ in M.
- Keine anderen Zahlen liegen in M.

Zeigen Sie durch strukturelle Induktion:

$$\forall\, n \in M : n \bmod 3 = 2$$

Lösung

Induktionsanfang $5 \ \mathbf{mod}\ 3 = 8 \ \mathbf{mod}\ 3 = 2$

Induktionsvoraussetzung Für beliebige aber feste $n, m \in M$ gelte:

$$n \ \mathbf{mod}\ 3 = 2 \text{ und } m \ \mathbf{mod}\ 3 = 2$$

Induktionsschritt: Wir zeigen, dass dann auch $(n^2 + m^2) \ \mathbf{mod}\ 3 = 2$ gilt.
Aus $n \ \mathbf{mod}\ 3 = 2$ folgt

$$n^2 \ \mathbf{mod}\ 3 = (n \ \mathbf{mod}\ 3)^2 \ \mathbf{mod}\ 3 = 2^2 \ \mathbf{mod}\ 3 = 4 \ \mathbf{mod}\ 3 = 1$$

Ebenso gilt $m^2 \ \mathbf{mod}\ 3 = 1$, und es folgt

$$(n^2 + m^2) \ \mathbf{mod}\ 3 = 1 + 1 = 2$$

Was ihr nun wissen solltet

- Reguläre Ausdrücke
- Rechtslineare Grammatiken

Was nächstes Mal kommt

- Turingmaschinen - Mächtiger wird es nicht mehr!

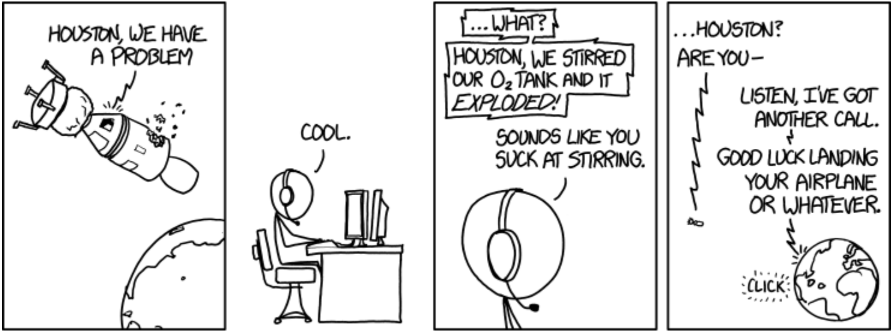


Abbildung: `http://www.xkcd.com/1438`

Oh, hi mom. No, nothing important, just work.

Danksagung

Dieser Foliensatz basiert in Teilen auf Folien von:

Philipp Basler
Nils Braun
Dominik Doerner
Ou Yue