Grundbegriffe der Informatik

Einheit 15: Reguläre Ausdrücke und rechtslineare Grammatiken

Thomas Worsch

Universität Karlsruhe, Fakultät für Informatik

Wintersemester 2008/2009

Was kann man mit endlichen Akzeptoren?

- manche Sprachen kann man mit endlichen Akzeptoren erkennen
  - z. B. $\{\mathtt{a}\}^+\{\mathtt{b}\} \cup \{\mathtt{b}\}^+\{\mathtt{a}\}$
- manche Sprachen *nicht*
  - z. B. $\{\mathtt{a}^k\mathtt{b}^k \mid k \in \mathbb{N}_0\}$
- *Charakterisierung* der erkennbaren Sprachen?
  - i. e. Beschreibung ohne Benutzung endlicher Akzeptoren

Überblick

Reguläre Ausdrücke

Rechtslineare Grammatiken

Überblick

Reguläre Ausdrücke

Rechtslineare Grammatiken

Zum Begriff

regulärer Ausdruck

- Ursprung:
  Stephen Kleene: *Representation of Events in Nerve Nets and Finite Automata*,
  in: Shannon, McCarthy, Ashby (eds.): *Automata Studies*, 1956
- heute: verschiedene Bedeutungen
- in dieser Vorlesung: die „klassische“ Definition
- Verallgemeinerung: *regular expressions*
  - sehr nützlich
  - verschiedene Varianten (bei Syntax, bei Semantik)
  - emacs, grep, sed, ...
  - Java (java.util.regex), Python, Perl, ...

Definition regulärer Ausdrücke (1)

- sei Z das Alphabet $Z = \{|, (,), *, \emptyset\}$
- sei A ein Alphabet, das kein Zeichen aus Z enthält
- *regulärer Ausdruck* über A ist eine Zeichenfolge über dem Alphabet $A \cup Z$, die gewissen Vorschriften genügt.
- Menge der regulären Ausdrücke ist wie folgt festgelegt:
  - $\emptyset$ ist ein regulärer Ausdruck.
  - Für jedes $a \in A$ ist a ein regulärer Ausdruck.
  - Wenn R_1 und R_2 reguläre Ausdrücke sind, dann sind auch $(R_1 | R_2)$ und $(R_1 R_2)$ reguläre Ausdrücke.
  - Wenn R ein regulärer Ausdruck ist, dann auch $(R*)$.
  - Nichts anderes sind reguläre Ausdrücke.

Klammereinsparungsregeln

- „Stern- vor Punktrechnung“
- „Punkt- vor Strichrechnung“
- Beispiel:
  - $R_1 \mid R_2 R_3*$ ist als
  - $(R_1 \mid (R_2 (R_3*)))$ zu verstehen.
- Bei mehreren gleichen binären Operatoren gilt das als links geklammert
- Beispiel
  - $R_1 \mid R_2 \mid R_3$ ist als
  - $((R_1 \mid R_2) \mid R_3)$ zu verstehen.

Beispiele

- `∅`
- `(01)`
- `(∅|1)`
- `(∅*)`
- `((0*)*)`
- `0`
- `((01)0)`
- `(0|1)`
- `(0*)`
- `((((01)1)*)*)|(∅*)`
- `1`
- `(((01)0)0)`
- `((0(0|1))|1)`
- `((10)(1*))`
- `((01)(00))`
- `(0|(1|(0|0)))`
- `(((10)1)*)`

Mit Klammereinsparungsregeln:

- `01`
- `∅|1`
- `∅*`
- `0**`
- `010`
- `0|1`
- `0*`
- `(011)**|∅*`
- `0100`
- `0(0|1)|1`
- `101*`
- `01(00)`
- `(0|(1|(0|0)))`
- `(101)*`

Nichtbeispiele

keine regulären Ausdrücke über $\{0, 1\}$:

`(|1)`	vor `|` fehlt ein regulärer Ausdruck
`|∅|`	vor und hinter `|` fehlt je ein regulärer Ausdruck
`()01`	zwischen `(` und `)` fehlt ein regulärer Ausdruck
`((01)`	Klammern müssen „gepaart“ auftreten
`*(01)`	vor `*` fehlt ein regulärer Ausdruck
`2*`	`2` ist nicht Zeichen des Alphabetes

Definition regulärer Ausdrücke (2)

- alternative Formulierung mit Hilfe einer kontextfreien Grammatik
- reguläre Ausdrücke über Alphabet A sind die Wörter, die von der folgenden Grammatik erzeugt werden:

$$G = (\{R\},\ \{\mid, (,), *, \emptyset\} \cup A,\ R,\ P)$$

$$\text{und } P = \{R \to \emptyset,\ \ R \to (R \mid R),\ \ R \to (RR),\ \ R \to (R*)\}$$
$$\cup \{R \to a \mid a \in A\}$$

Beschriebene formale Sprache

Die *von einem regulären Ausdruck R beschriebene formale Sprache* $\langle R \rangle$ ist wie folgt definiert:

- $\langle \emptyset \rangle = \{\}$ (d. h. die leere Menge).
- Für $a \in A$ ist $\langle a \rangle = \{a\}$.
- Sind R_1 und R_2 reguläre Ausdrücke, so ist $\langle R_1 \mid R_2 \rangle = \langle R_1 \rangle \cup \langle R_2 \rangle$.
- Sind R_1 und R_2 reguläre Ausdrücke, so ist $\langle R_1 R_2 \rangle = \langle R_1 \rangle \cdot \langle R_2 \rangle$.
- Ist R ein regulärer Ausdruck, so ist $\langle R* \rangle = \langle R \rangle^*$.

- die Definition folgt der für reguläre Ausdrücke

Beschriebene formale Sprache

Die *von einem regulären Ausdruck R beschriebene formale Sprache* $\langle R \rangle$ ist wie folgt definiert:

- $\langle \emptyset \rangle = \{\}$ (d. h. die leere Menge).
- Für $a \in A$ ist $\langle a \rangle = \{a\}$.
- Sind R_1 und R_2 reguläre Ausdrücke, so ist $\langle R_1 \,|\, R_2 \rangle = \langle R_1 \rangle \cup \langle R_2 \rangle$.
- Sind R_1 und R_2 reguläre Ausdrücke, so ist $\langle R_1 R_2 \rangle = \langle R_1 \rangle \cdot \langle R_2 \rangle$.
- Ist R ein regulärer Ausdruck, so ist $\langle R* \rangle = \langle R \rangle^*$.

- die Definition folgt der für reguläre Ausdrücke

Beispiele für $\langle R \rangle$

Betrachten wir drei einfache Beispiele:

- $R = \mathtt{a|b}$: Dann ist
  $\langle R \rangle = \langle \mathtt{a|b} \rangle = \langle \mathtt{a} \rangle \cup \langle \mathtt{b} \rangle = \{\mathtt{a}\} \cup \{\mathtt{b}\} = \{\mathtt{a}, \mathtt{b}\}$.
- $R = \mathtt{(a|b)*}$: Dann ist
  $\langle R \rangle = \langle \mathtt{(a|b)*} \rangle = \langle \mathtt{a|b} \rangle^* = \{\mathtt{a}, \mathtt{b}\}^*$.
- $R = \mathtt{(a*b*)*}$: Dann ist
  $\langle R \rangle = \langle \mathtt{(a*b*)*} \rangle = \langle \mathtt{a*b*} \rangle^*$
  $= (\langle \mathtt{a*} \rangle \langle \mathtt{b*} \rangle)^* = (\langle \mathtt{a} \rangle^* \langle \mathtt{b} \rangle^*)^* = (\{\mathtt{a}\}^* \{\mathtt{b}\}^*)^*$.
- Nachdenken: $(\{\mathtt{a}\}^* \{\mathtt{b}\}^*)^* = \{\mathtt{a}, \mathtt{b}\}^*$

Beispiele für $\langle R \rangle$

Betrachten wir drei einfache Beispiele:

- $R = \texttt{a|b}$: Dann ist
  $\langle R \rangle = \langle \texttt{a|b} \rangle = \langle \texttt{a} \rangle \cup \langle \texttt{b} \rangle = \{\texttt{a}\} \cup \{\texttt{b}\} = \{\texttt{a}, \texttt{b}\}$.
- $R = \texttt{(a|b)*}$: Dann ist
  $\langle R \rangle = \langle \texttt{(a|b)*} \rangle = \langle \texttt{a|b} \rangle^* = \{\texttt{a}, \texttt{b}\}^*$.
- $R = \texttt{(a*b*)*}$: Dann ist
  $\langle R \rangle = \langle \texttt{(a*b*)*} \rangle = \langle \texttt{a*b*} \rangle^*$
  $= (\langle \texttt{a*} \rangle \langle \texttt{b*} \rangle)^* = (\langle \texttt{a} \rangle^* \langle \texttt{b} \rangle^*)^* = (\{\texttt{a}\}^* \{\texttt{b}\}^*)^*$.
- Nachdenken: $(\{\texttt{a}\}^* \{\texttt{b}\}^*)^* = \{\texttt{a}, \texttt{b}\}^*$

Beispiele für $\langle R \rangle$

Betrachten wir drei einfache Beispiele:

- $R = \texttt{a|b}$: Dann ist
  $\langle R \rangle = \langle \texttt{a|b} \rangle = \langle \texttt{a} \rangle \cup \langle \texttt{b} \rangle = \{\texttt{a}\} \cup \{\texttt{b}\} = \{\texttt{a}, \texttt{b}\}.$
- $R = \texttt{(a|b)*}$: Dann ist
  $\langle R \rangle = \langle \texttt{(a|b)*} \rangle = \langle \texttt{a|b} \rangle^* = \{\texttt{a}, \texttt{b}\}^*.$
- $R = \texttt{(a*b*)*}$: Dann ist
  $\langle R \rangle = \langle \texttt{(a*b*)*} \rangle = \langle \texttt{a*b*} \rangle^*$
  $= (\langle \texttt{a*} \rangle \langle \texttt{b*} \rangle)^* = (\langle \texttt{a} \rangle^* \langle \texttt{b} \rangle^*)^* = (\{\texttt{a}\}^* \{\texttt{b}\}^*)^* \ .$
- Nachdenken: $(\{\texttt{a}\}^* \{\texttt{b}\}^*)^* = \{\texttt{a}, \texttt{b}\}^*$

Beispiele für $\langle R \rangle$

Betrachten wir drei einfache Beispiele:

- $R = \texttt{a|b}$: Dann ist
  $\langle R \rangle = \langle \texttt{a|b} \rangle = \langle \texttt{a} \rangle \cup \langle \texttt{b} \rangle = \{\texttt{a}\} \cup \{\texttt{b}\} = \{\texttt{a}, \texttt{b}\}$.
- $R = \texttt{(a|b)*}$: Dann ist
  $\langle R \rangle = \langle \texttt{(a|b)*} \rangle = \langle \texttt{a|b} \rangle^* = \{\texttt{a}, \texttt{b}\}^*$.
- $R = \texttt{(a*b*)*}$: Dann ist
  $\langle R \rangle = \langle \texttt{(a*b*)*} \rangle = \langle \texttt{a*b*} \rangle^*$
  $= (\langle \texttt{a*} \rangle \langle \texttt{b*} \rangle)^* = (\langle \texttt{a} \rangle^* \langle \texttt{b} \rangle^*)^* = (\{\texttt{a}\}^* \{\texttt{b}\}^*)^*$.
- Nachdenken: $(\{\texttt{a}\}^* \{\texttt{b}\}^*)^* = \{\texttt{a}, \texttt{b}\}^*$

Wie ist das denn eigentlich?

- Kann man „allgemein“ von regulären Ausdrücken R_1, R_2 feststellen, ob $\langle R_1 \rangle = \langle R_2 \rangle$ ist?
- Geht das algorithmisch?

- Welche formalen Sprachen sind denn durch reguläre Ausdrücke beschreibbar?

Äquivalenz regulärer Ausdrücke

- Es gibt Algorithmen, um für reguläre Ausdrücken R_1, R_2 festzustellen, ob $\langle R_1 \rangle = \langle R_2 \rangle$ ist
  - sogar konzeptionell ziemlich einfache
- **Aber:** Dieses Problem ist PSPACE-vollständig.
  - Definition: siehe nächste Einheit
  - d. h. jedenfalls: alle bisher bekannten (!) Algorithmen sind im allgemeinen sehr sehr sehr langsam, Laufzeit z. B. 2^{n^2} ...
- Man weiß nicht, ob es vielleicht doch Algorithmen mit polynomieller Laufzeit für das Problem gibt, aber man sie „nur noch nicht gefunden“ hat.

Satz

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

1. L kann von einem endlichen Akzeptor erkannt werden.
2. L kann durch einen regulären Ausdruck beschrieben werden.
3. L kann von einer rechtslinearen Grammatik erzeugt werden.

- rechtslineare Grammatiken kommen gleich
- Eine formale Sprache, die die Eigenschaften des Satzes hat, heißt *reguläre Sprache*.
- Jede rechtslineare Grammatik ist eine kontextfreie Grammatik, also ist jede reguläre Sprache eine kontextfreie Sprache,
- *aber nicht umgekehrt*, wie man z. B. an $\{a^k b^k \mid k \in \mathbb{N}_0\}$ sieht.

Satz

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

1. L kann von einem endlichen Akzeptor erkannt werden.
2. L kann durch einen regulären Ausdruck beschrieben werden.
3. L kann von einer rechtslinearen Grammatik erzeugt werden.

- rechtslineare Grammatiken kommen gleich
- Eine formale Sprache, die die Eigenschaften des Satzes hat, heißt *reguläre Sprache*.
- Jede rechtslineare Grammatik ist eine kontextfreie Grammatik, also ist jede reguläre Sprache eine kontextfreie Sprache,
- *aber nicht umgekehrt*, wie man z. B. an $\{\mathtt{a}^k\mathtt{b}^k \mid k \in \mathbb{N}_0\}$ sieht.

Zum Beweis des Satzes

konstruktiv:

- zu gegebenem endlichen Akzeptor A ein regulärer Ausdruck R mit $\langle R \rangle = L(A)$
  - „mittel schwer“, z. B. inspiriert vom Algorithmus von Warshall
- zu gegebenem regulären Ausdruck R eine rechtslineare Grammatik G mit $L(G) = \langle R \rangle$:
  - „relativ leicht“
- zu gegebener rechtslinearer Grammatik G ein endlicher Akzeptor A mit $L(A) = L(G)$:
  - am „schwierigsten“
- beachte: für die umgekehrten Richtungen braucht man keine expliziten Konstruktionen

Zum Beweis des Satzes

konstruktiv:

- zu gegebenem endlichen Akzeptor A ein regulärer Ausdruck R mit $\langle R \rangle = L(A)$
  - „mittel schwer“, z. B. inspiriert vom Algorithmus von Warshall

  Kernidee:

  $$R_{i,0,j} = \text{reg. Aus. für } \{x \in X \mid f(z_i, x) = z_j\} \cup \begin{cases} \{\varepsilon\} & \text{falls } i = j \\ \{\} & \text{falls } i \neq j \end{cases}$$

  $$R_{i,k,j} = (R_{i,k-1,j} \mid (R_{i,k-1,k-1}\ (R_{k-1,k-1,k-1})*\ R_{k-1,k-1,j}))$$

- zu gegebenem regulären Ausdruck R eine rechtslineare Grammatik G mit $L(G) = \langle R \rangle$:
  - „relativ leicht“
- zu gegebener rechtslinearer Grammatik G ein endlicher Akzeptor A mit $L(A) = L(G)$:
  - am „schwierigsten“
- beachte: für die umgekehrten Richtungen braucht man keine expliziten Konstruktionen

Zum Beweis des Satzes

konstruktiv:

- zu gegebenem endlichen Akzeptor A ein regulärer Ausdruck R mit $\langle R \rangle = L(A)$
  - „mittel schwer“, z. B. inspiriert vom Algorithmus von Warshall
- zu gegebenem regulären Ausdruck R eine rechtslineare Grammatik G mit $L(G) = \langle R \rangle$:
  - „relativ leicht“
- zu gegebener rechtslinearer Grammatik G ein endlicher Akzeptor A mit $L(A) = L(G)$:
  - am „schwierigsten“
- beachte: für die umgekehrten Richtungen braucht man keine expliziten Konstruktionen

Zum Beweis des Satzes

konstruktiv:

- zu gegebenem endlichen Akzeptor A ein regulärer Ausdruck R mit $\langle R \rangle = L(A)$
  - „mittel schwer“, z. B. inspiriert vom Algorithmus von Warshall
- zu gegebenem regulären Ausdruck R eine rechtslineare Grammatik G mit $L(G) = \langle R \rangle$:
  - „relativ leicht“
- zu gegebener rechtslinearer Grammatik G ein endlicher Akzeptor A mit $L(A) = L(G)$:
  - am „schwierigsten“
- beachte: für die umgekehrten Richtungen braucht man keine expliziten Konstruktionen

Zum Beweis des Satzes

konstruktiv:

- zu gegebenem endlichen Akzeptor A ein regulärer Ausdruck R mit $\langle R \rangle = L(A)$
  - „mittel schwer“, z. B. inspiriert vom Algorithmus von Warshall
- zu gegebenem regulären Ausdruck R eine rechtslineare Grammatik G mit $L(G) = \langle R \rangle$:
  - „relativ leicht“

  Idee: z. B. für $(R_1 \mid R_2)$
  - gegeben $G_1 = (N_1, T_1, X_1, P_1)$ für $\langle R_1 \rangle$
  - gegeben $G_2 = (N_2, T_2, X_2, P_2)$ für $\langle R_2 \rangle$ mit $N_1 \cap N_2 = \emptyset$
  - $G = (\{X_0\} \cup N_1 \cup N_2, T_1 \cup T_2, X_0, \{X_0 \to X_1 \mid X_2\} \cup P_1 \cup P_2)$ Grammatik für $\langle (R_1 \mid R_2) \rangle$ (mit “neuem“ $X_0 \notin N_1 \cup N_2$)
- zu gegebener rechtslinearer Grammatik G ein endlicher Akzeptor A mit $L(A) = L(G)$:
  - am „schwierigsten“
- beachte: für die umgekehrten Richtungen braucht man keine expliziten Konstruktionen

Zum Beweis des Satzes

konstruktiv:

- zu gegebenem endlichen Akzeptor A ein regulärer Ausdruck R mit $\langle R \rangle = L(A)$
  - „mittel schwer“, z. B. inspiriert vom Algorithmus von Warshall
- zu gegebenem regulären Ausdruck R eine rechtslineare Grammatik G mit $L(G) = \langle R \rangle$:
  - „relativ leicht“
- zu gegebener rechtslinearer Grammatik G ein endlicher Akzeptor A mit $L(A) = L(G)$:
  - am „schwierigsten“
- beachte: für die umgekehrten Richtungen braucht man keine expliziten Konstruktionen

Zum Beweis des Satzes

konstruktiv:

- zu gegebenem endlichen Akzeptor A ein regulärer Ausdruck R mit $\langle R \rangle = L(A)$
  - „mittel schwer“, z. B. inspiriert vom Algorithmus von Warshall
- zu gegebenem regulären Ausdruck R eine rechtslineare Grammatik G mit $L(G) = \langle R \rangle$:
  - „relativ leicht“
- zu gegebener rechtslinearer Grammatik G ein endlicher Akzeptor A mit $L(A) = L(G)$:
  - am „schwierigsten“
- beachte: für die umgekehrten Richtungen braucht man keine expliziten Konstruktionen

Zum Beweis des Satzes

konstruktiv:

- zu gegebenem endlichen Akzeptor A ein regulärer Ausdruck R mit $\langle R \rangle = L(A)$
  - „mittel schwer“, z. B. inspiriert vom Algorithmus von Warshall
- zu gegebenem regulären Ausdruck R eine rechtslineare Grammatik G mit $L(G) = \langle R \rangle$:
  - „relativ leicht“
- zu gegebener rechtslinearer Grammatik G ein endlicher Akzeptor A mit $L(A) = L(G)$:
  - am „schwierigsten“
- beachte: für die umgekehrten Richtungen braucht man keine expliziten Konstruktionen

Was ist wichtig

Das sollten Sie mitnehmen:

- Definition „klassischer“ regulärer Ausdrücke
  - atomare:
    - $\emptyset$
    - $a \in A$
  - zusammengesetzte:
    - $(R_1 \mid R_2)$
    - $(R_1 R_2)$
    - $(R)*$
- wissen: reguläre Ausdrücke und die Verallgemeinerung Regular Expressions sind z. B. bei Textverarbeitungsaufgaben manchmal nützlich

Das sollten Sie üben:

- zu L ein R mit $\langle R \rangle = L$ konstruieren
- zu R das $\langle R \rangle$ bestimmen

Überblick

Reguläre Ausdrücke

Rechtslineare Grammatiken

Motivation

- Mit kontextfreien Grammatiken kann man jedenfalls zum Teil andere formale Sprachen erzeugen, als man mit endlichen Akzeptoren erkennen kann.
- Beispiel:
  - $G = (\{X\}, \{\mathtt{a}, \mathtt{b}\}, X, \{X \to \mathtt{a}X\mathtt{b} \mid \varepsilon\})$ erzeugt $\{\mathtt{a}^k\mathtt{b}^k \mid k \in \mathbb{N}_0\}$
  - und diese Sprache ist nicht regulär.
- Kann man kontextfreie Grammatiken so einschränken, dass sie zu endlichen Akzeptoren passen?

Rechtslineare Grammatiken: Definition

- Eine *rechtslineare Grammatik* ist eine kontextfreie Grammatik $G = (N, T, S, P)$, die der folgenden Einschränkung genügt: Jede Produktion ist
  - entweder von der Form $X \to w$ mit $w \in T^*$
  - oder von der Form $X \to wY$ mit $w \in T^*$ und $X, Y \in N$.
- Auf der rechten Seite einer Produktion darf also höchstens ein Nichterminalsymbol vorkommen, und wenn dann nur als letztes Symbol.

Rechtslineare Grammatiken: Beispiel

$$G = (\{X\}, \{\mathtt{a}, \mathtt{b}\}, X, \{X \to \mathtt{ab}X \mid \mathtt{bba}X \mid \varepsilon\}$$

$$L(G) = \langle \mathtt{(ab|bba)*} \rangle$$

Rechtslineare Grammatiken: Nichtbeispiel

- $G = (\{X\}, \{\mathtt{a}, \mathtt{b}\}, X, \{X \to \mathtt{a}X\mathtt{b} \mid \varepsilon\})$ ist nicht rechtslinear,
  - denn in $X \to \mathtt{a}X\mathtt{b}$ steht das Nichtterminalsymbol X nicht am rechten Ende.
- Da die erzeugte formale Sprache nicht regulär ist, kann es auch gar keine rechtslineare Grammatik geben, die $\{\mathtt{a}^k\mathtt{b}^k \mid k \in \mathbb{N}_0\}$ erzeugt.

Sprechweisen

- Rechtslineare Grammatiken heißen auch *Typ-3-Grammatiken*.
- Kontextfreien Grammatiken heißen auch *Typ-2-Grammatiken*.
- Man ahnt schon: Es gibt auch noch
  - Typ-1-Grammatiken und
  - Typ-0-Grammatiken,

  die wir hier nicht weiter betrachten werden.
- Wenn für ein $i \in \{0, 1, 2, 3\}$ eine formale Sprache L von einer Typ-i-Grammatik erzeugt wird, dann sagt man auch, L sei eine Typ-i-Sprache oder kurz vom Typ i.

Sprechweisen

- Rechtslineare Grammatiken heißen auch *Typ-3-Grammatiken*.
- Kontextfreien Grammatiken heißen auch *Typ-2-Grammatiken*.
- Man ahnt schon: Es gibt auch noch
  - Typ-1-Grammatiken und
  - Typ-0-Grammatiken,

  die wir hier nicht weiter betrachten werden.
- Wenn für ein $i \in \{0, 1, 2, 3\}$ eine formale Sprache L von einer Typ-i-Grammatik erzeugt wird, dann sagt man auch, L sei eine Typ-i-Sprache oder kurz vom Typ i.

Sprechweisen

- Rechtslineare Grammatiken heißen auch *Typ-3-Grammatiken*.
- Kontextfreien Grammatiken heißen auch *Typ-2-Grammatiken*.
- Man ahnt schon: Es gibt auch noch
  - Typ-1-Grammatiken und
  - Typ-0-Grammatiken,

  die wir hier nicht weiter betrachten werden.
- Wenn für ein $i \in \{0, 1, 2, 3\}$ eine formale Sprache L von einer Typ-i-Grammatik erzeugt wird, dann sagt man auch, L sei eine Typ-i-Sprache oder kurz vom Typ i.

Sprechweisen

- Rechtslineare Grammatiken heißen auch *Typ-3-Grammatiken*.
- Kontextfreien Grammatiken heißen auch *Typ-2-Grammatiken*.
- Man ahnt schon: Es gibt auch noch
  - Typ-1-Grammatiken und
  - Typ-0-Grammatiken,

  die wir hier nicht weiter betrachten werden.
- Wenn für ein $i \in \{0, 1, 2, 3\}$ eine formale Sprache L von einer Typ-i-Grammatik erzeugt wird, dann sagt man auch, L sei eine Typ-i-Sprache oder kurz vom Typ i.

Was ist wichtig

Das sollten Sie mitnehmen:

- Definition rechtslineare Grammatiken

Das sollten Sie üben:

- rechtslineare Grammatiken konstruieren
  (zu gegebenem Akzeptor, regulären Ausdruck, formaler Sprache)

Zusammenfassung

- reguläre Ausdrücke
  - werden von diversen „Unix-Tools“ genutzt
  - in manchen Programmiersprachen zur Textverarbeitung praktisch
- rechtslineare Grammatiken