19 REGULÄRE AUSDRÜCKE UND RECHTSLINEARE GRAMMATIKEN

19.1 REGULÄRE AUSDRÜCKE

Klammereinsparungsregeln

- sind wohl sinnvoll und naheliegend gewählt
- muss man daher hoffentlich nicht groß auswendig lernen
- zumal es dann für $\langle R \rangle$ sowieso egal ist, ob man von links oder von rechts klammert: z. B. $\langle \texttt{((aa)b)} \rangle = \langle \texttt{(a(ab))} \rangle = \{\texttt{aab}\}$
- aber sind `((aa)b)` und `(a(ab))` verschiedene reguläre Ausdrücke (wenn sie auch die gleiche formale Sprache beschreiben)

kontextfreie Grammatik, die die regulären Ausdrücke erzeugt

- habe lange überlegt, ob ich die mit rein nehme
- **Vorsicht Gefahr:** nicht durcheinander bringen. Die *Syntax* regulärer Ausdrücke ist kontextfrei, aber die Bedeutung, i. e. *Semantik*, regulärer Ausdrücke sind nur reguläre Sprachen
- **aber auch lehrreich:** mal wieder Unterschied zwischen Syntax (Typ-2-Sprache) und Semantik (Typ-3-Sprachen)

durch regulären Ausdruck beschriebene formale Sprache

- weitere Beispiele der Form „von R zu $\langle R \rangle$“
  - $R =$ `(a|b)*abb(a|b)*`: … $\langle R \rangle$ enthält genau die Wörter, in denen das Teilwort `abb` vorkommt.
  - $R =$ `a**`: $\langle R \rangle = \{a\}^*$. Zwei Sterne unmittelbar hintereinander sind nicht besser als einer.
- weitere Beispiele der Form „von $\langle R \rangle$ zu R“
  - R für die Sprache aller Wörter, in denen mindestens drei `b` vorkommen:
    `(a|b)*b(a|b)*b(a|b)*b(a|b)*`
    wer „optimieren“ will: z. B. `a*ba*ba*b(a|b)*`
  - R für die Sprache $\{\varepsilon\}$: `∅*`, denn $\langle \emptyset * \rangle = \langle \emptyset \rangle^* = \{\}^* = \{\varepsilon\}$
  - R für die Sprache aller Wörter, in denen nirgends das Teilwort `ab` vorkommt: `b*a*`
  - Wenn R ein regulärer Ausdruck für eine formale Sprache $L = \langle R \rangle$ ist, wie sieht dann ein regulärer Ausdruck
    - für L^* aus: $\texttt{(}R\texttt{)*}$
    - für L^+ aus: $R\texttt{(}R\texttt{)*}$
- Bitte ggf. erläutern, dass $(\{\texttt{a}\}^*\{\texttt{b}\}^*)^* = \{\texttt{a},\texttt{b}\}^*$ ist: Man kann jedes Wort zerhacken in eine Folge von Blöcken, von denen jeder ein Teilwort aus `a`’s gefolgt von einem Teilwort aus `b`’s ist.

Beweis von Äquivalenzen im Kreis

- ggf. noch mal erläutern
- Konsequenz: wenn man z. B. zu regulärem Ausdruck äquivalenten endlichen Akzeptor konstruieren will, muss man dem Umweg über rechtslineare Grammatik machen. In der Praxis vielleicht unpraktisch: aber es gibt auch direkte Konstruktionen.

19.2 RECHTSLINEARE GRAMMATIKEN

Beispiel rechtslinearer Grammatiken

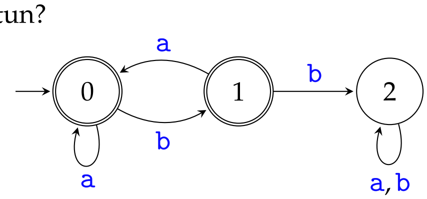
- *Das hier ist keine rechtslineare Grammatik:*
  $G = (\{X, Y\}, \{\mathtt{a}, \mathtt{b}\}, X, \{X \to \mathtt{a}Y \mid \varepsilon, Y \to X\mathtt{b}\})$

  Die Grammatik ist zwar (wie man auch sagt) linear, aber nicht *rechts*linear, denn die Produktion $Y \to X\mathtt{b}$ hat das Nichtterminalsymbol nicht am rechten Ende.

  Da $L(G) = \{\mathtt{a}^k\mathtt{b}^k \mid k \in \mathbb{N}_0\}$, sieht man deutlich, dass das Mischen von rechts- und linkslinearen Produktionen zu mehr als regulären Sprachen führt.
- von G zu $L(G)$: betrachte $G = (\{X, Y, Z\}, \{\mathtt{a}, \mathtt{b}\}, X, P)$ mit $P = \{X \to \mathtt{a}X \mid \mathtt{b}Y \mid \varepsilon,\ \ Y \to \mathtt{a}X \mid \mathtt{b}Z \mid \varepsilon,\ \ Z \to \mathtt{a}Z \mid \mathtt{b}Z\}$

  Was ist $L(G)$?

  Was hat diese Grammatik mit dem folgenden Automaten aus der vorigen Einheit zu tun?

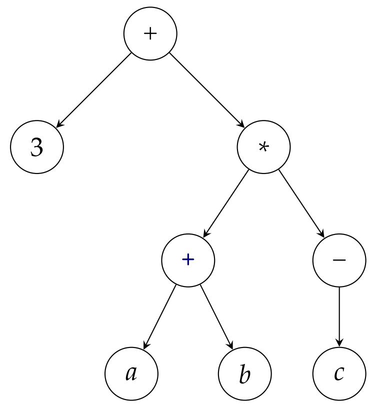
  

- Natürlich könnte man die Grammatik vereinfachen: $G = (\{X, Y\}, \{\mathtt{a}, \mathtt{b}\}, X, P)$ mit $P = \{X \to \mathtt{a}X \mid \mathtt{b}Y \mid \varepsilon,\ \ Y \to \mathtt{a}X \mid \varepsilon\}$ erzeugt die gleiche Sprache.
- Wer findet eine noch einfachere Lösung?
  $G = (\{X\}, \{\mathtt{a}, \mathtt{b}\}, X, P)$ mit $P = \{X \to \mathtt{a}X \mid \mathtt{ba}X \mid \mathtt{b} \mid \varepsilon\}$

19.3 KANTOROWITSCH-BÄUME UND STRUKTURELLE INDUKTION

Kantorowitsch-Bäume

Kantorowitsch-Bäume führ ich nicht formal ein. Zur weiteren Erläuterung vielleicht auch noch mal einen arithmetischen Ausdruck wie $3 + (a + b) * (-c)$ umwandeln in



Regex-Bäume

Das ist natürlich kein feststehender Begriff. Ich benutzt ihn nur, um mir nicht den Mund fusselig zu reden.

Höhe von Bäumen

Kann man auch definieren als Länge der längsten (wiederholungsfreien) Wege von der Wurzel zu irgendwelchen Blättern.

Eventuell die etwas lasche Formulierung des Falles „$1 + \max_i h(U_i)$, falls …“ erläutern

