Theoretische Informatik Formale Sprachen und Grammatiken

Prof. Dr. Juho Mäkiö

04.12. 2018

- Datenverarbeitung erfolgt durch formale, d.h. künstliche, Sprachen
  - Beispiele …..?
- Warum gibt es formale Sprachen?
  - Exaktheit: zulässige Ausdrücke + Bedeutung
  - Bedeutung kontextunabhängig → automatische Verarbeitung

Hier geht es um….

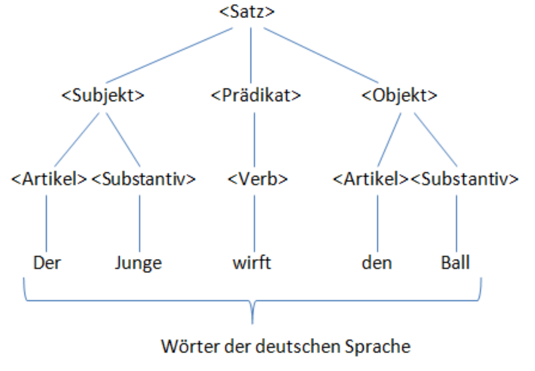
- formale Sprachen:
  $L \subset \sum^*$
  über eine Grammatik-Struktur.
- L wird über eine Syntax/Grammatik beschrieben.
- $\omega \in L$ gilt dann, wenn ω dieser Syntax genügt.
- Bsp.:
  Syntax -> Linguistik
  <Satz> -> <Subjekt> <Prädikat> <Objekt>
  z.B. Der Junge wirft den Ball
  < … > ≙ syntaktische Hilfssymbole

Formale Sprachen…

- noch etwas genauer (Syntax-Baum):

→ In der Linguistik wird das weiter vertieft

→ Dort wird auch das Problem der Fälle (z.B. Nominativ für Subjekt und z.B. Akkusativ für Objekt) weiter behandelt.



Grammatiken

- In der Informatik Grammatiken beschreiben formale Sprachen.
- Komponenten einer Grammatik G = (T, N, P, S):
  - endliches Terminalalphabet T,
  - endliche Menge N Nichtterminalsymbolen (Variablen),
  - endliche Menge P von Regeln a → b, wobei a und b aus N und T gebildet sind,
  - einer Startvariablen S aus N.

Formale Beschreibung

Eine Grammatik G ist ein 4-Tupel.

G = (V, ∑, P, S) mit

- V ≙ endliche Menge von Variablen (syntaktische Hilfssymbole)
- ∑ ≙ endliches (Eingabe-) Alphabet (Terminalalphabet)
- P ≙ endliche Menge von Produktionen (Regeln)

der Form l -> r

mit $l \in (V \cup \sum)^+ \setminus \sum^*$ (d. h. mindestens 1 Variable)

$r \in (V \cup \sum)^*$

- $S \in V$ ≙ Startvariable

Übliche Notation

- Variable : große Buchstaben
- Terminale : kleine Buchstaben
  $\in \Sigma$ bzw. Ziffern
- $l \to r_1 \quad l \to r_2 \quad l \to r_3 \quad \ldots$
  $l \to r_1 \mid r_2 \mid r_3 \mid \ldots$

Beispiel

- $G = (V, \Sigma, P, S)$ mit
- $V = \{S\}$, $\Sigma = \{0, 1\}$,
- $P = \{S \to 0S1, S \to 01\}$, S

- Welche Wörter könnten zu L(G) gehören?
  $S \Rightarrow 0S1 \Rightarrow 0\,0S1\,1 \Rightarrow 000111$
  $S \to 0S1$
- Man erkennt: $L(G) = \{0^n 1^n \mid n \geq 1\}$

- Produktionen werden solange angewendet, bis nur noch
- Terminalzeichen vorhanden sind.
- Solche Wörter gehören dann zu L(G).

Formal: L(G) = ?

- $G = (V, \Sigma, P, S)$ ~ Grammatik
- Ableitungsrelation: $\Rightarrow_G$
  $vlu \Rightarrow_G vru$ mit $v, u \in (V \cup \Sigma)^*$ und $(l \rightarrow r) \in P$
- $\Rightarrow^*_G$ (Kleensche Hülle von $\Rightarrow_G$)
  – mehrfache Anwendung von $\Rightarrow_G$
- $L(G) = \{\omega \in \Sigma^* \mid S \Rightarrow^*_G \omega\}$

- 2 Grammatiken (G_1, G_2) sind äquivalent
  $G_1 \equiv G_2$, falls es gilt: $L(G_1) = L(G_2)$
  (vgl. Automaten)

Beispiel

- $G = (V, \Sigma, P, S)$ mit
- $V = \{S, A, B\}$ $\Sigma = \{0, 1\}$
- P : $S \rightarrow AB$
- $A \rightarrow 0|0A$
- $B \rightarrow 1|1B$

S

$S \Rightarrow_G AB \Rightarrow_G 0AB$
$\Rightarrow_G 00B$
$\Rightarrow_G 001B$
$\Rightarrow_G 0011B$
$\Rightarrow 00111$

$L(G) = \{0^n 1^m \mid n, m \geq 1\}$

Beispiel

- G = ({S, A, B}, {a, b}, P, S) mit
- P : S -> AB|ABA
- A -> aA|a
- B -> Bb|ε

z.B.
S => ABA
=> aABA
=> aaBA
=> aaBbA
=> aaBbbA
=> aabbA
=> aabbaA
=> aabbaa ∈ L(G)
L(G) = $\{a^i\, b^j\, a^k \mid i \geq 1;\ j, k \geq 0\}$

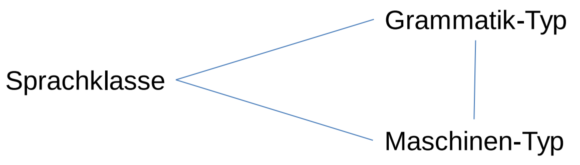
Beispiel…

G = ({S, B, C}, {a, b, c}, P, S)
P : S -> aSBC|aBC
CB -> BC
aB -> ab
bB -> bb
bC -> bc
cC -> cc
z.B.

z.B.:
S => aSBC => aaBCBC
=> aaBBCC
=> aabBCC
=> aabbCC
=> aabbcC
=> aabbcc
L(G) = $\{a^n\, b^n\, c^n \mid n \geq 1\}$

Typen von Grmmatiken

- verschiedene Typen von Grammatiken werden Maschinen bzw. Sprachklassen zugeordnet:



- Die verschiedenen Grammatik-Typen unterscheiden sich im Aussehen der Produktionen/Regeln.

Reguläre Grammatiken (Typ-3-Grammatiken)

- Eine Grammatik heißt rechtslinear, wenn alle Regeln von der Form
  - $A \rightarrow aB$ oder $A \rightarrow \varepsilon$ mit $A, B \in V$, $a \in \Sigma$ sind.
- Eine Grammatik heißt linkslinear, wenn alle Regeln von der Form
  - $A \rightarrow Ba$ oder $A \rightarrow \varepsilon$ mit $A, B \in V$, $a \in \Sigma$ sind.

Eine rechts- oder linkslineare Grammatik heißt regulär oder auch Typ-3-Grammatik.

Beispiel

G_1 = ({S, A, B, C}, {0, 1}, P, S)
mit

S -> 0S|1S|0A
A -> 1B ~ „rechtslinear“
B -> 0C ~ Wort wächst von links
C -> ε nach rechts

z.B.

S => 1S => 11S => 110A => 1101B => 11010C => 11010
$L(G_1) = L((0+1)^* 010)$

≥ 0-mal (Notationserweiterung: …$^+$: >= 1-mal)

[Automat leicht zu konstruieren]

Beispiel…

G_2 = ({S, A, B, C}, {0, 1}, P, S)
mit

S -> A0
A -> B1 ~ „linkslinear“
B -> C0 ~ Wort wächst von rechts
C -> C0|C1|ε nach links

S => A0 => B10 => C010 => C11010 => 11010
$L(G_2) = L(G_1)$

Varianten aus der Literatur:

$A \;\text{->}\; \omega B|\omega \quad$ mit $A, B \in V$, $\omega \in \Sigma^*$

rechtslinear

$A \;\text{->}\; B\omega|\omega \quad$ mit $A, B \in V$, $\omega \in \Sigma^*$

linkslinear

Beispiel

- Gibt es für $L = \{(ab)^n \mid n \geq 1\}$ eine reguläre Grammatik?

  ? $L = \{ab, abab, ababab, \ldots\}$

  → gar nicht so einfach!

$G_3 = (\{S, B, C\}, \{a, b\}, P, S)$

mit

S -> aB

B -> bC

C -> ε|aB

S => aB => abC => abaB => ababC => ababaB
=> abababC => ababab

In vereinfachter Form
S -> abS |ab

Aufgabe

- Geben Sie einen NEA A an, der Wörter über $\sum = \{a,b\}$ erkennt, die mit einem a beginnen und enden.
- Geben Sie eine rechtslineare Grammatik G an, mit L(G) = L(A)

Aufgaben

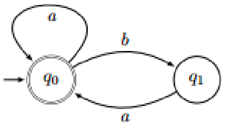
(1) Gegeben sei die Grammatik G = (N,T,P,S) mit N = {S,A,B}, T = {0,1}, S = S und P = {(S -> 0A | 1B), (A -> 1 | 1S | 0AA), (B -> 0 | 0S | 1BB)}. Welche Sprache erzeugt G?

(2) Gebe die Grammatik an, die die Sprache $L = \{a^n b^{2n} \mid n > 0\}$ erzeugt.

(3) Sei G = (N, T, P, S) eine Grammatik mit N = {S, A, B}, T = {a, b} und P wie folgt:
$S \rightarrow AB \mid SAB$
$A \rightarrow Aa \mid \varepsilon$
$B \rightarrow Bb \mid \varepsilon$
Geben Sie die von G erzeugte Sprache L(G) an (ohne Beweis).

(4) Geben Sie eine rechtslineare Grammatik an, die die Sprache generiert, die von dem Automaten akzeptiert wird.

(5) Geben Sie einen regulären Ausdruck an, der die Sprache beschreibt, die von dem Automaten akzeptiert wird.



Aufgabe

- Betrachten Sie die formale Sprache
  $L = \{w \mid w = a^n c b^m \wedge n,m \in \mathbb{N}_0\}$.
- a) Geben Sie eine reguläre Grammatik *G* an mit *L* = *L*(*G*).
- b) Zeichnen Sie jeweils den Ableitungsbaum der Zeichenketten "c" und "acbb" für *G*.
- c) Schreiben Sie die Ableitung von "aaacbb" für *G*.

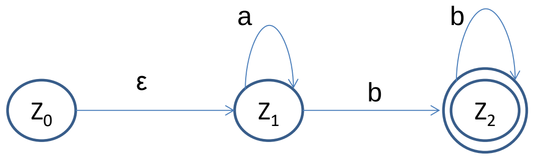
Aufgabe

Gegeben sei die rechtslineare Grammatik G = <{S, A, B, C}, {a, b, c, d}, P, S> mit
P = {S → aA,S → ε, A → aA, A → bB, A → cC,
B → bB, B → cC, B → ε, C → cC, C → dS}.

1. Geben Sie die Ableitung des Wortes aabccd an.
2. Geben Sie den die L(G) beschreibenden regulären Ausdruck an.

Aufgabe

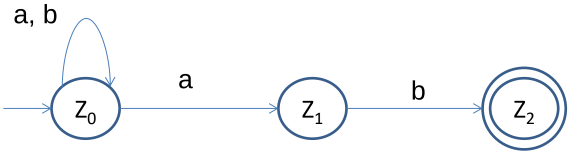
- Betrachten Sie den folgenden endlichen Automaten:



a. Wie lautet der regulärer Ausdruck r mit L(r)=L(A)?
b. Geben Sie eine rechtslineare Grammatik G mit L(A) = L(G) an.
c. Geben Sie einen endlichen Automaten A' an mit L(A) = L(A'). A' soll einen zustand weniger haben.
d. Geben Sie zum A' die entsprechende Grammatik G' an.

Aufgabe

- Betrachten Sie den folgenden endlichen Automaten:



a. Wie lautet der regulärer Ausdruck r mit L(r)=L(A)?
b. Geben Sie eine rechtslineare Grammatik G mit L(A) = L(G) an.

Aufgabe

- Geben Sie eine linkslineare Grammatik G an, die die Sprache L = {ab^nc} erzeugt.

Yes we can…