Grammatiken

Programmieren und Software-Engineering
Homomorphismen, Formale Sprachen und Syntax-Analyse

22. Februar 2023

Worte der Länge n

- Ein **Alphabet** ist eine nichtleere endliche Menge, deren Elemente Zeichen oder Symbole genannt werden.
- Ein **Wort der Länge** n ist eine Folge $x_1 x_2 \ldots x_n$ von n Symbolen des Alphabets A
- Jede beliebige geschlossene Gruppe von Zeichen innerhalb eines Wortes wird als **Teilwort** bezeichnet.
- Zwei Wörter x und y werden als *gleich* bezeichnet, wenn sie die gleiche Länge besitzen und auch die Reihenfolge der Zeichen gleich ist.
- Sei T das Alphabet, so bezeichnet T^+ die Menge aller daraus konstruierbaren Wörter.

Verkettung

Definition (Verkettung)

Sind $x = x_1x_2 \dots x_n$ und $y = y_1y_2 \dots y_m$ zwei beliebige Wörter aus der Menge T^+, dann ist die **Zusammensetzung oder Verkettung** definiert als

$$xy = x_1x_2 \dots x_ny_1y_2 \dots y_m.$$

- Wird zu T^+ ein spezielles Wort ε (Epsilon) hinzugefügt, so erhält man die Menge T^*.

$$T^* = T^+ \cup \{\varepsilon\}$$

- ε wird als **Leerwort** bezeichnet. Es ist kein Terminalsymbol (und somit auch nicht im Alphabet enthalten).
- Es gilt:

$$x\varepsilon = \varepsilon x = x$$

Länge eines Wortes

- Die **Länge eines Wortes** x ist durch die Anzahl der Zeichen des Wortes bestimmt:

$$|x_1 x_2 \ldots x_n| = n$$

- Es gilt:

$$|xy| = |x| + |y|$$

- Für das Leerwort gilt:

$$|\varepsilon| = 0$$

Verkettung von Wortmengen

Definition (Verkettung von Wortmengen)

Seien M und N zwei Wortmengen. Die Verkettung MN ist definiert als

$$MN = \{x \mid x = yz, y \in M, z \in N\}.$$

- Die als Ergebnis entstehende Menge enthält alle jene Worte, die aus einem Wort der Menge M verkettet mit einem Wort der Menge N bestehen.
- Die Menge MN wird als das Produkt von M und N bezeichnet.
- Für die Produktoperation gilt das Kommutativgesetzt **nicht**!
- Somit gilt im Allgemeinen: $MN \neq NM$

Verkettung von Wortmengen

Beispiel: Gegeben seien zwei Alphabete Q und R sowie die beiden Wortmengen $M \subseteq Q^+$ und $N \subseteq R^+$.

$$\begin{array}{ll} Q = \{a, b, c\} & M = \{c, bba, bcacc\} \\ R = \{D, E, F, G\} & N = \{FE, DDF\} \end{array}$$

Man beachte den Unterschied

$$MN = \{cFE, cDDF, bbaFE, bbaDDF, bcaccFE, bcaccDDF\}$$

$$NM = \{FEc, FEbba, FEbcacc, DDFc, DDFbba, DDFbcacc\}$$

Verkettung von Wortmengen

Die Operationen "+" und "$*$" können auch auf Wortmengen angewandt werden.

Beispiel: Gegeben sei die Wortmenge $M = \{c, bba, bcacc\}$.
Eine mögliche Teilmenge Z von M^* ist:

$$Z = \{\varepsilon, c, ccccc, cbbac, cbbabcacc, bbabbabcaccc, cccbbacc\}$$

WH: Formale Sprache

- Eine *formale Sprache* L über einem Alphabet T ist eine beliebige Teilmenge von T^*.
- Grammatiken sind formale Systeme die zur Erzeugung formaler Sprachen dienen.
- Grammatiken sind definiert als

$$G = (N, T, P, S),$$

  für die gilt:
  - Alphabet der Nichtterminalsymbole N
  - Alphabet der Terminalsymbole T mit $N \cap T = \emptyset$.
    Weiters sei $V := N \cup T$.
  - Menge der Produktionsregeln P
    Eine Regel schreiben wir abstrakt als $\omega_1 \to \omega_2$, bzw. $(\omega_1, \omega_2) \in P$.
  - Startvariable S mit $S \in N$

Produktionsregeln

- Wir untersuchen nun die linke Seite ω_1 und die rechte Seite ω_2 von Produktionsregeln $\omega_1 \to \omega_2$ genauer.
- Sei $\omega_1 \in N$, so besteht die linke Seite der Regel jeweils nur aus einem Nonterminal
- Sei $\omega_2 \in NT$, so ist die rechte Seite der Regel von der Form: ein Nonterminalsymbol gefolgt von einem Terminalsymbol
  - Hier wurde die Verkettung NT der Mengen N und T verwendet.
- Sei $\omega_2 \in T \cup TN$, so besteht die rechte Seite der Regel entweder aus einem Terminalsymbol, oder aus einem Terminalsymbol gefolgt von einem Nonterminalsymbol.

Chomsky-Hierarchie

- Die **Chomsky-Hierarchie** unterscheidet verschiedene Arten von Grammatiken aufgrund der Beschaffenheit ihrer Produktionsregeln.
- Grammatiken vom **Typ-0**, **Typ-1**, **Typ-2** und **Typ-3** unterscheiden sich durch ihre Mächtigkeit.
- Die Typ-0 Grammatiken weisen die größte Mächtigkeit auf, während Typ-3 Grammatiken nur relativ einfache Sprachen erzeugen können.

Chomsky-Hierarchie

- **(Rechts-)Reguläre Grammatik (Typ-3-Grammatik):**
  Links von $\rightarrow$ steht genau ein Nonterminal. Rechts von $\rightarrow$ steht entweder ein Terminalsymbol oder ein Terminalsymbol gefolgt von genau einem Nonterminal
- **Kontextfreie Grammatik (Typ-2-Grammatik):**
  Links von $\rightarrow$ steht genau ein Nonterminal, rechts davon eine beliebige Folge von Terminalen und Nonterminalen (genauer aus V^*)
- **Kontextsensitive (umgebungsabhängige) Grammatik (Typ-1-Grammatik):**
  Hier sind zusätzlich Produktionsregeln erlaubt, auf deren linker Seite auch Terminalsymbole vorkommen, jedoch mindestens ein Nonterminal enthalten ist. Dieser "Kontext" der Terminalsymbole muss auf der rechten Seite der Produktionsregel erhalten bleiben.
- **Allgemeine Regelgrammatik (Typ-0-Grammatik):**
  Die Einschränkungen für die rechte Seite bei der Typ-1-Grammatik gelten hier nicht.

Chomsky-Hierarchie (*)

- **(Rechts-)Reguläre Grammatik (Typ-3-Grammatik):**

$$\forall(\omega_1 \to \omega_2) \in P : \omega_1 \in N \wedge \omega_2 \in T \cup TN \cup \{\varepsilon\}$$

- **Kontextfreie Grammatik (Typ-2-Grammatik):**

$$\forall(\omega_1 \to \omega_2) \in P : \omega_1 \in N \wedge \omega_2 \in V^*$$

- **Kontextsensitive (umgebungsabhängige) Grammatik (Typ-1-Grammatik):**
  Alle Produktionsregeln sind von der Form:

$$\alpha A \beta \to \alpha\gamma\beta \text{ mit } A \in N \wedge \alpha, \beta, \gamma \in V^*,$$

  oder $S \to \varepsilon$

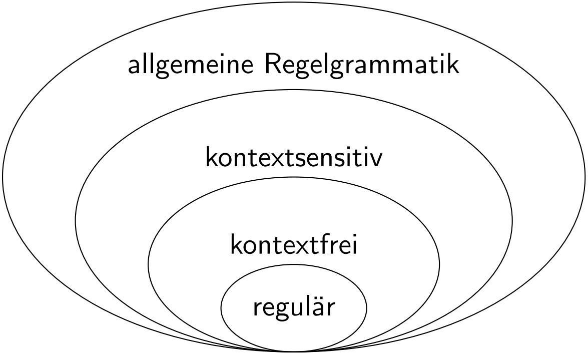
- **Allgemeine Regelgrammatik (Typ-0-Grammatik):**

$$\forall(\omega_1 \to \omega_2) \in P : \omega_1 \in V^+ \wedge \omega_2 \in V^*$$

  wobei ω_1 mindestens ein Nonterminal enthalten muss.

Chomsky-Hierarchie

Die Grammatiken höheren Typs sind in jenen niedrigeren Typs jeweils vollständig enthalten:



Beispiele zur Chomsky-Hierarchie

Beispiel:

T-3: $\langle \text{A} \rangle \rightarrow \text{x}$
$\langle \text{A} \rangle \rightarrow \text{x}\langle \text{B} \rangle$

T-2: $\langle \text{A} \rangle \rightarrow \text{x}\langle \text{A} \rangle\text{y}\langle \text{B} \rangle\text{c}$
$\langle \text{conditional statement} \rangle \rightarrow \text{if } \langle \text{expression} \rangle \text{ then } \langle \text{statementlist} \rangle \text{ fi}$

T-1: $\text{x}\langle \text{A} \rangle\text{y} \rightarrow \text{x}\langle \text{B} \rangle\text{z}\langle \text{C} \rangle\text{w}\langle \text{D} \rangle\langle \text{E} \rangle\text{y}$

T-0: .... $\rightarrow$ ..... (mindestens ein Nonterminal auf der linken Seite)
$\text{x}\langle \text{A} \rangle \rightarrow \langle \text{B} \rangle$

Chomsky-Hierarchie

Sprachen höheren Typs sind *echte Teilmengen* der Sprachklassen niederen Typs.

Grammatik	**Sprache**	**Automat**
Typ-0	rekursiv aufzählbar	nichtdeterministische Turingmaschine
Typ-1	kontextsensitiv	nichtdeterministische Turingmaschine mit linear beschränkter Bandlänge
Typ-2	kontextfrei	nichtdeterministischer Kellerautomat
Typ-3	regulär	endlicher Automat

Programmiersprachen sind im Allgemeinen kontextfreie Sprachen!

Aufgaben

Aufgabe 3.8

- Welche Regeln von SPL sind regulär?
- Wie können die anderen Regeln klassifiziert werden?

Literaturübersicht I

[1] **Berger, Krieger, Mahr: "Grundlagen der elektronischen Datenverarbeitung", Skriptum**

[2] Dirk W. Hoffmann: "Theoretische Informatik", Hanser, 3. Auflage

[3] Gernot Salzer: "Einführung in die Theorie der Informatik", Skriptum, TU Wien, 2001

[4] Wikipedia (Englisch): `https://en.wikipedia.org/`

[5] Wikipedia (Deutsch): `https://de.wikipedia.org/`