Theoretische Informatik
WS 2024/2025

Kapitel 1
Formale Sprachen

1.1
Sprachen und Grammatiken

Prof. Dr. Robert Preis
Fachbereich Informatik
Fachhochschule Dortmund
Robert.Preis@fh-dortmund.de

Motivation: Sprachen

Ich brauche eine Sprache, um

- etwas zu beschreiben.
- um überhaupt mit einer Person (natürliche Sprachen) oder einem Rechner (formale Sprachen) zu kommunizieren.

Beispiele:

- *„Ich bin heute mit dem rechten Fuß zuerst aufgestanden.“*
  ist ein Satz aus der deutschen Sprache.
- *„for (i=0; i<100; i++) printf(„%d“, i) end;“*
  ist ein Satz aus einer Programmiersprache.

- *Welche Sprachen kennen Sie?*

- *Welche Sprachen können Sie?*

Motivation: Sprachprobleme

Woher wissen Sie, ob ein Wort oder ein Satz in einer Sprache ist?

Was ist eigentlich ein „Wort“?

Was ist eigentlich ein „Satz“?

Was ist eigentlich eine „Sprache“?

Und woher weiß ich, ob ein Wort in einer Sprache ist?

Ist „Bierkrug“ ein deutsches Wort?

Ist „Der Bierkrug isst einen Teller Wasser.“ ein deutscher Satz?

Es sind dabei 2 Probleme zu lösen:

1. Besteht das Wort/der Satz aus zulässigen Buchstaben (Alphabet)?
2. Ist die Syntax korrekt, d.h. sind die Buchstaben in einer korrekten Reihenfolge (Grammatik)?

Ist (17-7)*(27-35+((4+3*(2+5))) ein korrekter arithmetischer Ausdruck ?

*Was ist das **Alphabet**?* { 0,1,2,3,4,5,6,7,8,9,+,-,*,/,(,) }

*Was ist das **Wort**?* w = (17-7)*(27-35+((4+3*(2+5)))

*Was ist die **Sprache**?* L = {alle arithmetischen Ausdrücke}

*Was ist das **Wortproblem**?* $w \in L$?

Da L unendlich groß ist, geht es nicht einfach durch das Vergleichen mit den Elementen einer Menge, sondern:

Welche Eigenschaften haben die Wörter in L ?

Hat w auch diese Eigenschaften ?

*Was sind **Eigenschaften**?* **Wörter erfüllen eine Grammatik !**

Grammatik mit Startvariable E und Regeln

E → E+E|E-E|E∗E|E/E|(E)|0|1|2|3|4|5|6|7|8|9|1Z|2Z|3Z|4Z|5Z|6Z|7Z|8Z|9Z

Z → 0|1|2|3|4|5|6|7|8|9|0Z|1Z|2Z|3Z|4Z|5Z|6Z|7Z|8Z|9Z

*Wie **überprüfe** ich Eigenschaften?*

Ist E → …→… → … → (17-7)*(27-35+((4+3*(2+5))) möglich ?

Alphabet und Wort

Alphabet ∑:

Ein Alphabet ∑ ist eine endliche Menge von Symbolen, z. B.

- ∑ = {A, ..., Z, a, .., z, ., ?, !}
- ∑ = {a, b, c}
- ∑ = {0, 1}
- ∑ = {0, ..., 9}

Wort (Zeichenreihe, String):

Ein Wort w ist eine endliche Folge von Symbolen eines Alphabets, z. B.

- w=abba bei dem Alphabet ∑={a, b}
- ε leeres Wort (ohne jedes Symbol aus dem Alphabet)
- |w| Länge des Wortes w (Anzahl der Symbole)
  Bsp.: |abba| = 4, |ε| = 0
- v◦w=vw Konkatenation (Verkettung, Aneinanderhängung) von v und w
  Bsp.: „über“◦„natürlich“ = „übernatürlich“.
- w^i i-fache Konkatenation des Wortes (oder Symbols) w
  Bsp.: $(klein)^3$=klein◦klein◦klein=kleinkleinklein, ab^4c=abbbbc

Mengen von Wörtern = Sprache

Eine Sprache ist eine beliebige Menge von Wörtern über einem Alphabet (üblicherweise in abstrakter Mengennotation gegeben), z.B.

Die leere Sprache:

- L = {}

Sprachen mit nur einem Element:

- L = {ε}
- L = {a}
- L = {abba}

Sprachen mit einer endlichen Anzahl an Elementen:

- L = {1, 2, 3, 4, 5}
- L = {Januar, Februar, März, April, Mai, Juni, Juli, August, September, Oktober, November, Dezember}
- L = {Alle Wörter in einem Wörterbuch}

Sprachen können unendlich groß sein und komponiert werden

Sprachen mit einer unendlichen Anzahl an Elementen:

- L = {Alle geraden Zahlen}
- L = {Binär-String aus 0-en und 1-en mit gerader Anzahl an Ziffern}
- $L = \{0^n1^n, n \in \mathbf{N}_+\} = \{01, 0011, 000111, 00001111, ...\}$
- L = {Alle korrekten JAVA Programme}

Kompositionen von Sprachen sind möglich, z.B.:

- Vereinigung zweier Sprachen L und M ist wieder ein Sprache:
  $L \cup M = \{ w \mid w \in L \text{ oder } w \in M\}$
- Verkettung zweier Sprachen L und M ist wieder ein Sprache:
  $L \circ M = \{ v \circ w \mid v \in L, w \in M\}$

Beispiel: L = {Haus, Zimmer}, M = {Decke, Wand}

$L \cup M$ = {Haus, Zimmer, Decke, Wand}

$L \circ M$ = {HausDecke, HausWand, ZimmerDecke, ZimmerWand}

Mehrfache Verkettung von Wörtern einer Menge und Kleene‘sche Hülle

Mehrfachverkettungen (am Beispiel: L = {Karl, Josef, Heinz}):

- alle Gesamtwörter aus 0 Wörtern ist nur das leere Wort:
  $L^0 = \{\varepsilon\}$
- alle Gesamtwörter aus 1 Wort ist Sprache selbst:
  $L^1 = L = \{Karl, Josef, Heinz\}$
- alle Gesamtwörter aus 2 Wörtern:
  $L^2 = L \circ L = \{KarlKarl, KarlJosef, KarlHeinz, JosefKarl, JosefJosef, JosefHeinz, HeinzKarl, HeinzJosef, HeinzHeinz\}$
- alle Gesamtwörter aus 3 Wörtern:
  $L^3 = L \circ L \circ L = \{KarlKarlKarl, KarlKarlJosef, KarlKarlHeinz, \ldots\}$
- alle Gesamtwörter aus n Wörtern:
  $L^n = L^{n-1} \circ L$
- alle Gesamtwörter aus mindestens einem Wort:
  $L^+ = \bigcup_{n \geq 1} L^n = L^1 \cup L^2 \cup L^3 \cup \ldots$
- alle Gesamtwörter aus beliebig vielen Wörtern (Kleene‘sche Hülle):
  $L^* = \bigcup_{n \geq 0} L^n = L^0 \cup L^1 \cup L^2 \cup L^3 \cup \ldots$

Vom Entscheidungsproblem zum Wortproblem für Sprachen

Jedes Problem, bei dem eine Entscheidung gesucht wird, kann auch als Wortproblem einer Sprache aufgefasst werden:

Ist 1578571 eine Primzahl ?

ist identisch zu

Gehört das Wort „1578571“ zu der Sprache {Primzahlen} ?

Ja / Nein ?

oder

Ist 1578571 eine gerade natürliche Zahl ?

ist identisch zu

Gehört das Wort „1578571“ zu der Sprache {gerade natürlichen Zahlen} ?

Ja / Nein ?

D.h. das „Wortproblem für Sprachen“ ist ein zentrales Problem der Informatik.

Das Wortproblem einer Sprache

Gegeben sei ein Wort x und eine Sprache L:

Gehört das Wort x zu der Sprache L , d.h. x ∈ L?

- *‚17‘∈ L={3, 6, 15, 7, 11, 77, 17, 7, 13, 71} ?*
- *‚Kuckucksei‘ ∈ L={Wörter aus dem Wörterbuch} ?*

Wenn die Sprache eine endliche Aufzählung ist, dann muss man das Wort nur darin suchen…

…das ist einfach !!!

- *‚Theoretische Informatik‘ ∈ L={Wörter mit gerader Anzahl von a‘s} ?*
- *‚ABBA‘, und ‚EIN ESEL LESE NIE‘ ∈ L={alle Palindrome} ?*
- *‚174577718571‘ ∈ L={alle Primzahlen} ?*

Wenn die Sprache eine Mengendefinition ist, dann muss man es überprüfen, ob es der Definition entspricht, d.h. ob es die Eigenschaften der Sprache besitzt…

…das kann schwer werden !!!

Lösung für das Wortproblem

Gehört das Wort x zu der Sprache L ?

Für eine Lösung müssen wir...

1. **...wissen, wie die Sprache L aufgebaut ist.**
   - **L ist endlich:**
     Bei einer Sprache mit einer kleinen (endlichen) Menge an Wörtern kann man einfach alle Wörter aufzählen.
     Beispiel: Alle Wörter der deutschen Sprache im Duden.
   - **L ist unendlich:**
     Wenn L viele oder unendlich viele Wörter hat, dann müssen wir den Aufbau der Sprache kennen. Das ist die Grammatik!
     Beispiel: Alle Sätze der deutschen Sprache.

2. **...eine Maschine (Programm/Automaten) haben, die das Wort anhand der Grammatik überprüfen kann.**

Eine Grammatik bildet eine Sprache

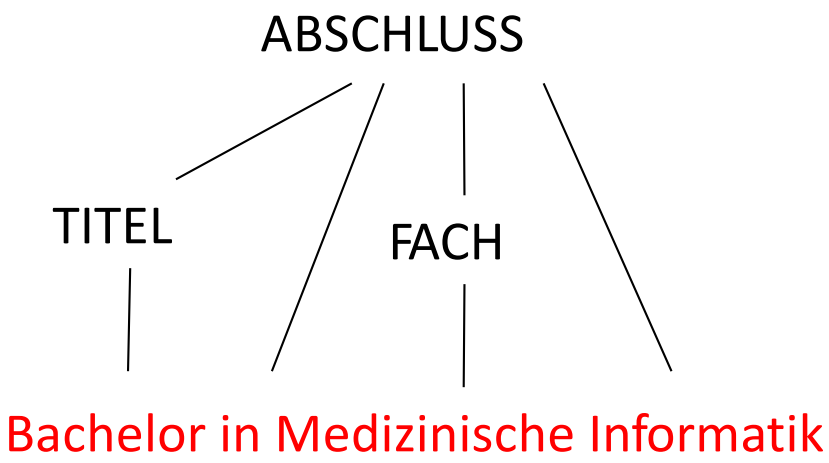
Mit dem Alphabet

{Bachelor, Master, in, Informatik, Medizinische, Wirtschafts}

bildet die Grammatik

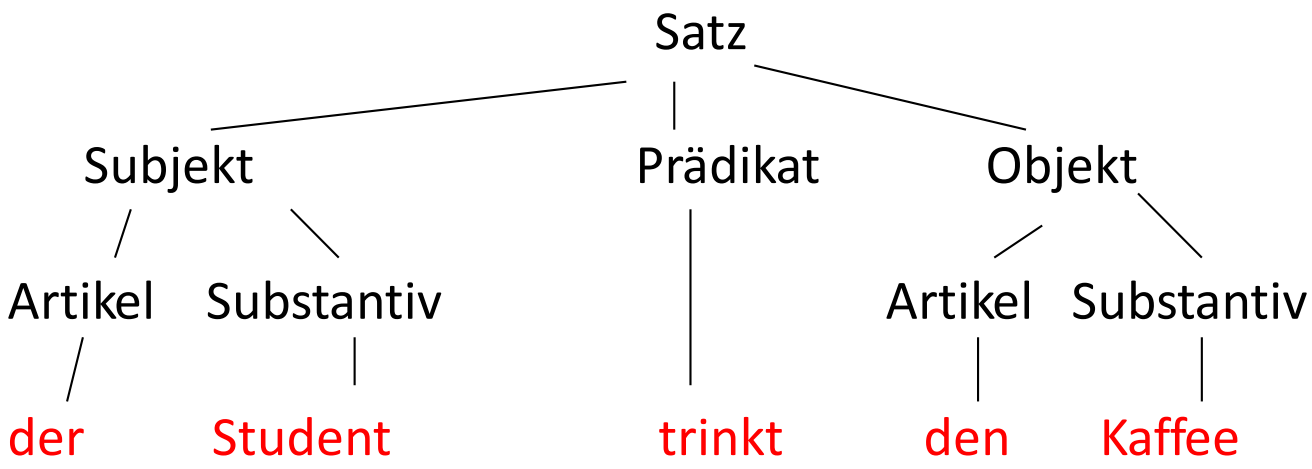
1. **ABSCHLUSS** → **TITEL** in **FACH** Informatik
2. **TITEL** → Bachelor | Master
3. **FACH** → Medizinische | Wirtschafts | ε

die Sprache der Informatik-Abschlüsse an der FH Dortmund.



Eine kleine Grammatik für ein paar deutsche Sätze

Satz → Subjekt Prädikat Objekt
Subjekt → Artikel Substantiv
Objekt → Artikel Substantiv
Artikel → der | die | das | den
Substantiv → Student | Kaffee | Dekan
Prädikat → kocht | trinkt



Alternativ: das Dekan kocht die Student

Grammatiken für formale Sprachen

Die Regeln einer Grammatik können nicht nur Sätze einer natürlichen Sprache beschreiben, sondern auch Sätze von formalen Sprachen sein.

Beispiele für formale Sprachen, bei denen die Bestandteile nach Regeln zusammengesetzt sind:

- Arithmetische Ausdrücke (3+4)*7
- Aussagenlogik: $(A \vee \neg B) \wedge C$
- Mengenlehre: $(A \cap B)^c$
- Programmiersprache if (a!=b) b=b/a;

<statement> → while (<condition>) {<statement>}

Die Bestandteile einer Grammatik

Eine Grammatik ist ein System zur Ableitung
(auch Herleitung oder Generierung) von Worten einer Sprache.

Grammatik (unvollständig): S→aA A→bB B→c

1. **Variablen (Nichtterminale):** hier **{S, A, B}**
   - Teilwörter oder Zwischenschritte
2. **Terminalsymbole: Alphabet der Sprache,** hier **{a, b, c}**
   - Symbole, aus denen die Wörter bestehen sollen
3. **Produktionsregeln (Substitutionsregeln):** hier **{S→aA, A→bB, B→c}**
   - Sind Regeln zur Erzeugung von Wörtern
   - Erklärt den syntaktischen Aufbau der Wörter
4. **Startsymbol:** hier **S**
   - Ist eine der Variablen

Grammatik (vollständig): ({S,A,B}, {a,b,c}, {S→aA,A→bB,B→c}, S)

Definition einer Grammatik

Eine Sprache L(G) wird definiert durch eine **Grammatik G = (V, T, P, S)**.

Beispiel:

G = (V, T, P, S) mit V = {S} T = {0, 1} P = {S→S1, S→S0, S→ε}

1. **V** ist eine endliche Menge der **Variablen** (Nichtterminalen).
2. **T** ist eine endliche Menge von **Terminalen** (Buchstaben), wobei $T \cap V = \emptyset$.
   Es ist $\Gamma = (V \cup T)$ die Vereinigung aus Variablen und Terminalen.
3. **P** $\subseteq (\Gamma^+ \setminus T^+) \times \Gamma^*$ ist eine endliche Menge von **Produktionsregeln** mit
   - $\Gamma^+ \setminus T^+$ (d.h. mindestens eine Variable muss vorkommen),
   - Γ^* (d.h. leeres Wort oder beliebig viele Variablen/Terminalen)

   Statt $(\alpha, \beta) \in P$ wird auch die Schreibweise $\alpha \rightarrow \beta$ verwendet.

   Die Notation $\alpha \rightarrow \beta 1 | \beta 2 | \ldots$ ist eine Abkürzung für $\alpha \rightarrow \beta 1$, $\alpha \rightarrow \beta 2$, ...
4. **S** $\in V$ ist die **Startvariable**.

Ableitung einer Grammatik

G = ({S,A,B,C}, {a,b,c}, P, S) mit

P = {
- S → ABC, (1)
- A → Aa|ε, (2)
- aB → abB|c, (3)
- bBC → Cc, (4)
- cC → b|ε } (5)

1. Starte mit der Startvariable.
2. Wende wiederholt Regeln an. Eine Regel kann angewendet werden, wenn der linke Teil der Regel in der Ableitung enthalten ist. Dieser Teil wird durch den rechten Teil der Regel ersetzt. In jedem Schritt ermittle alle Regeln, die angewendet werden können.
   - Genau eine Regel möglich: Dann wende diese an !
   - Mehrere Regeln möglich: Wähle eine aus !
3. Positives Ende, wenn nur noch Terminale existieren: dieses Terminalwort ist abgeleitet.
4. Negatives Ende, wenn man die Variablen nicht mehr los werden kann (keine anwendbare Regel oder Endlosschleife).

Ableitungen Beispiele

1. $G_1 = (\{S\}, \{0, 1\}, P, S)$ mit $P = \{S \to S1, S \to S0, S \to \varepsilon\}$
   - $S \to \varepsilon$
   - $S \to S0 \to 0$
   - $S \to S0 \to S10 \to S010 \to S0010 \to 0010$

2. $G_2 = (\{S,A,B,C\}, \{0, 1\}, P, S)$ mit
   $P = \{S \to B, S \to CA0, A \to BBB, B \to C1, B \to 0, CC1 \to \varepsilon\}$
   - $S \to B \to 0$
   - $S \to B \to C1$ Sackgasse, kein Wort der Zielsprache erreichbar
   - $S \to CA0 \to CBBB0 \to CC1BB0 \to BB0 \to 0B0 \to 000$

3. $G_3 = (\{S,A,B,C,D,E\}, \{a\}, P, S)$ mit $P = \{S \to ACaB, Ca \to aaC, CB \to DB, aD \to Da, AD \to AC, CB \to E, aE \to Ea, AE \to \varepsilon\}$
   - $S \to ACaB \to AaaCB \to AaaE \to AaEa \to AEaa \to aa$

Die Sprache einer Grammatik

Erweiterte Ableitungsregeln:

- $X \rightarrow^0 Y$ ≡ X=Y
- $X \rightarrow^n Y$ ≡ Y ist aus X in n Schritten ableitbar
- $X \rightarrow^* Y$ ≡ Y ist aus X in beliebig vielen Schritten ableitbar

Die von einer Grammatik G = ({S,…},T,P,S) erzeugte Sprache L(G)
ist die Menge der Terminalwörter,
die aus dem Startsymbol S
abgeleitet werden können:

$$L(G) = \{w \in T^* \mid S \rightarrow^* w\}$$

Brennende Grammatik Fragen

Es gibt einige Fragen, die immer wieder auftauchen:

Frage 1:

Ist die Sprache einer Grammatik leer oder wird mindestens ein Wort erzeugt?

Frage 2:

Ist ein bestimmtes Wort w Element der Sprache einer Grammatik?

Frage 3:

Sind 2 Grammatiken äquivalent, d.h. erzeugen sie dieselbe Sprache?

Frage 4:

Kann man eine Grammatik vereinfachen (kleiner machen),
d.h. mit noch weniger Variablen und Regeln dieselbe Sprache erzeugen?

Zusammenfassung

- Aus einem *Alphabet* können wir Wörter gestalten.
- Eine *Sprache* ist eine Menge, wird aber nicht immer als Aufzählung, sondern oft auch in komplexer Mengennotation angegeben.
- Es ist nicht immer sofort ersichtlich, ob ein Wort zu einer Sprache gehört (*Wortproblem*).
- Es wäre toll, eine Maschine oder ein Programm zu haben, die für ein Wort und eine Sprache immer entscheidet, ob das Wort zu der Sprache gehört oder nicht.
- Eine *Grammatik* ist ein Regelwerk das bestimmt, wie die Wörter einer Sprache aus den Buchstaben eines Alphabets aufgebaut sind.
- Man kann durch *Ableitung* vom Startsymbol alle Wörter durch mehrfache Nutzung der Regeln erzeugen.
- Die *Sprache einer Grammatik* ist die Menge der von der Grammatik vom Startsymbol aus erzeugten Wörter.