Theoretische Informatik III (T3INF2002)

Formale Sprachen und Automaten | Einführung Compilerbau

Vorlesung im Wintersemester 2022/23

Formale Sprachen und Automaten

- Formale Sprachen
- Grammatiken

Formale Sprachen

Theorie und Konzepte der formalen Sprachen

Theorie der formalen Sprachen beschäftigt sich mit der systematischen Analyse, Klassifikation und Konstruktion von Wortmengen, die über einem endlichen Alphabet gebildet werden.

- *Alphabet* Σ ist eine endliche Menge von Symbolen.
- Jedes Element $\sigma \in \Sigma$ ist ein *Zeichen* des Alphabets.
- Jedes Element $\omega \in \Sigma^*$ wird als *Wort* über Σ bezeichnet.
- Jede Teilmenge $L \subseteq \Sigma^*$ ist eine *formale Sprache* über Σ.

Formale Sprachen

Sprachen, die zur computergerechten Darstellung von Information und der Festlegung einer automatisieren Verarbeitung von Daten verwendet werden, müssen hohe Anforderungen an die Präzision und Ausdrucksweise erfüllen.

-> Syntax und Semantik solcher Sprachen werden daher präzise festgelegt.

Ziele:
Formulierung von Algorithmen in eindeutiger und für Computer verständlicher Weise.

Mittel:

Formalismen, die gewisse Ähnlichkeiten mit gesprochenen Sprachen haben, sich aber in Bezug auf Zweckmäßigkeit und Eindeutigkeit von gesprochenen Sprachen abgrenzen.

Formale Sprachen

Gesprochene Sprache hat u. a.

- Formalen Aufbau (Grammatik, d.h. Regeln)
- Bedeutung (Semantik)

→ auch bei formalen Sprachen

„kleine“ grammatisch korrekte Unterschiede können zu großen Bedeutungsunterschieden führen; auch jenseits von Gegenseitigkeit

Bsp.: Der Weg ist das Ziel. Weg ist das Ziel.

→ auch in formalen Sprachen möglich

Formale Sprachen

- Dasselbe Wort, d. h. dieselbe Buchstabenfolge kann in verschiedenen gesprochenen Sprachen vorkommen und dann verschiedene Bedeutungen haben

- Das ist auch in verschiedenen formalen Sprachen zulässig.

Formale Sprachen

Formale Sprachen vs. gesprochene Sprachen

Zeichen aus Alphabet	Buchstaben aus Alphabet
Wörter	Wörter
Ausdrücke, Anweisungen, Wörter, (Sätze)	Sätze

Formale Sprachen

An vielen Stellen der Informatik muss man entscheiden, ob eine Instanz zu einer Gesamtheit gehört, bzw. formal: ob es Element einer Menge ist.

Beispiele:

- Ist eine Textdatei ein syntaktisch korrektes Java-Programm?
- Ist eine aussagenlogische Formel eine Tautologie?
- Ist eine Graph ein Hamilton-Kreis? usw.

Formale Sprachen

- Allgemein beschreibt man dies durch eine Obermenge Y (z.B. die Menge aller Texte) und eine Teilmenge X (z.B. die Menge aller syntaktisch korrekten Java-Programme).
- Das Zugehörigkeitsproblem ist folgende Frage:
  - „Gegeben sei ein Element $x \in Y$ der Obermenge. Gilt $x \in X$ - oder gilt $x \in (Y \setminus X)$?“
- Problem ist einfach zu lösen, falls X eine endliche Menge ist.
- Ist X allerdings unendlich, dann muss die Menge geeignet in endlicher Form repräsentiert werden.

Formale Sprachen

- In der Mathematik sind im wesentlichen zwei Klassen von unendlichen Mengen bekannt:
  - Klasse der abzählbaren und Klasse der überabzählbaren Mengen
- In der Informatik ist man meist an Mengen von Zeichenketten über einem Alphabet interessiert, wobei das zwei-elementige Alphabet $\Sigma = \{0, 1\}$ der Binärwerte 0 und 1 gerade im Bereich der Hardware üblich ist.
- Menge aller Zeichenketten über einem Alphabet Σ wird mit Σ^* bezeichnet.
- Eine Teilmenge $L \subseteq \Sigma^*$ aller Zeichenkettenmenge heißt formale Sprache.

Formale Sprachen

Sei nun A eine endliche Repräsentation, die die Sprache $L(A) \subseteq \Sigma^*$ beschreibt:

- In diesem Fall stellt sich die Frage, ob man das Zugehörigkeitsproblem „$x \in L(A)$?“ auch auf der Repräsentation A algorithmisch lösen kann, d.h.: „Gegeben die endliche Repräsentation A und eine Zeichenkette $x \in \Sigma^*$, gilt nun $x \in L(A) \subseteq \Sigma^*$ oder $x \in (\Sigma^* \setminus L(A))$?“

In der Vorlesung behandeln wir zwei Arten der Repräsentation : ***Automaten und Grammatiken***

- Automaten charakterisieren Sprachen durch akzeptierende Verarbeitung der Zeichenketten, während Grammatiken diese generieren.
- Fasst man alle Sprachen, die Automaten eines Typs generieren können, zu einer Menge zusammen, erhält man eine sogenannte Sprachfamilie.

Alphabete, Wörter, Wortmengen

- Mengen von Wörtern sind von besonderer Wichtigkeit, wenn die Objekte, mit denen gearbeitet werden soll, zu kodieren sind:

  - > Beispiel : Binärwörter.

- Sie können als eine Kodierung der natürlichen Zahlen angesehen werden, genau wie die Dezimal-darstellung.

- Allen Kodierungen gemein ist die Tatsache, dass nur eine endliche Menge von Zeichen zur Bildung der Darstellung erlaubt ist :
  - Bei der Binärdarstellung nur 0 und 1,
  - bei der Dezimaldarstellung die zehn Ziffern 0 bis 9.

Eine solche Darstellung wird im Allgemeinen als ein Wort über einem endlichen Alphabet bezeichnet.

Alphabet

- Ein Alphabet ist eine total geordnete, endliche Menge von unterschiedlichen Zeichen (oder Symbolen).
- In der Mathematik und Logik bezeichnet man gelegentlich auch unendliche Mengen von Symbolen als Alphabete, allerdings sind dies dann stets abzählbare, total geordnete Mengen von Zeichen.
- Sofern nicht ausdrücklich anders erwähnt, wird im Rahmen der Vorlesung unter einem Alphabet immer ein endlicher Zeichenvorrat verstanden.

Alphabet

Häufig verwendetes Alphabet in der Informatik ist der ASCII-Code (American Standard Code for Information Interchange), ein 7-bit Code, dessen achtes Bit in der Byte- Darstellung als Prüfbit, oder zur Erweiterung des Zeichensatzes verwendet wird.

- Liste von 128 Schrift- und Steuerzeichen, die als Hexadezimalzahlen kodiert sind:
- In der 16 × 8 Matrix wird die Spaltennummer j als Hexadezimalziffer vor die Hexadezimalziffer i der Zeilennummer gesetzt und dann die Zeichenkette ji zur Basis 16 interpretiert.
- Dadurch ergibt sich eine lineare Ordnung der Zeichen entsprechend ihrer Codenummern.

Spalte	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	‘	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	’	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

Alphabet

- Für das Symbol „A“ ergibt sich gerade $[41]_{16} = 65$, für "EOT“ (end of transmission) $[4]_{16} = 4$, für „SP“ (space) $[20]_{16} = 32$ und für „DEL“ (delete) entsprechend $[7F]_{16} = 127$.
- Es ergibt sich die Reihenfolge: $\cdots \prec A \prec B \prec \cdots \prec Z \prec \cdots \prec a \prec \cdots \prec z \prec \cdots$.
- In der Regel wird die Ordnung bei Alphabeten als implizit gegeben betrachtet, und jeder endliche Zeichenvorrat als Alphabet bezeichnet.
- Falls man die Ordnung der Symbole bezeichnen möchte, nutzt man dazu das Zeichen $\prec$.

Spalte	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	‘	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	’	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

Alphabet

- Um aus einem bekannten Wort w über dem Alphabet {A, B, . . . , Z} ein neues zu konstruieren, kann man einzelne Symbole davor, dazwischen oder dahinter schreiben
  - man kann auch zwei bekannte Wörter aneinanderschreiben.
- Aus den Wörtern „S“, (auch ein einzelnes Symbol ist ein Wort!) „HAUS“, „BOOT“ und „BESITZER“ kann man so die Wörter BOOTS, BOOTSHAUS, HAUSBOOT, HAUSBESITZER, BOOTSBESITZER oder auch HAUSBOOT- BESITZER erzeugen.
- Hintereinanderschreibung von Zeichenketten bezeichnet man als **Konkatenation**, und mit ihr als assoziativer Operation zwischen den Zeichenketten (Wörtern) ist eine Struktur, hier eine Halbgruppe definiert.
- Wenn man an einer bestimmten Stelle kein Wort notieren will, aber diese Stelle dennoch bezeichnen muss, so benötigt man dafür ein spezielles Symbol (ähnlich der Null bei den Zahlen).
- Man nennt dieses das leere Wort und notieren es als λ (kleines lambda).

Das leere Wort λ ist niemals in einem Alphabet enthalten!
Es handelt sich ja gerade um dasjenige Wort, welches ohne jegliches Symbol aus dem Alphabet gebildet wird.

Wörter

- Konkatenation von Wörtern ist eine formale Operation, die man mit dem Operator · notiert, d.h. die Konkatenation der Wörter u, v ∈ Σ^* ist das Wort (u · v).
- Im folgenden schreiben wir meist nur kurz uv.
- Konkatenation ist assoziativ, d.h. für alle Wörter u, v ∈ Σ^* gilt:

$$(u \cdot v) \cdot w = u \cdot (v \cdot w)$$

- Das leere Wort λ ist das neutrale Element der Konkatenation, d.h. für alle Wörter u ∈ Σ^* gilt: u·λ=λ·u=u
- Also bildet (Σ^*, ·, λ) die algebraische Struktur eines Monoids.
- Man beachte, dass Konkatenation nicht kommutativ ist, d.h. im allgemeinen gilt u · v = v · u gerade nicht (so ist bspw. haus · boot = hausboot≠ boot · haus = boothaus).

Definition Formale Sprache

- Für ein Alphabet Σ ist $(\Sigma^*,\cdot,\lambda)$ das freie Monoid mit der Konkatenation (oder: Hintereinanderschreibung) der einzelnen Zeichen als Monoidoperation und dem leeren Wort λ als neutralem Element.
- Jede Menge $L \subseteq \Sigma^*$ heißt formale Sprache.
- $L := \{w \mid w \in \{a,b\}^* \wedge$ (w enthält das Symbol a doppelt so oft wie das Symbol b)} ist zum Beispiel eine recht einfache formale Sprache mit umständlicher Beschreibung.
- Auch die Menge $\{w \mid w \in \{a, b, c\}* \wedge$ (w beginnt mit dem Symbol a)} ist eine einfache formale Sprache mit unendlich vielen Elementen.

Definition Wort

Einige Eigenschaften von Zeichenketten werden wir bei der Beschreibung und Spezifikation von formalen Sprachen benötigen, um solche umständlichen Definitionen vermeiden zu können:

- Zu jedem Wort $w \in \Sigma^*$ und jedem Symbol $x \in \Sigma$ bezeichne $\mathbf{|w|_x}$ die **Häufigkeit des Vorkommens** des Symbols x im Wort w.
- Mit **|w|** wird die **Länge des Wortes** w bezeichnet, d.h. $|w| := \Sigma\, |w|_x$. $x \in \Sigma$
- Man schreibt $u \sqsubseteq v$, wenn es ein Wort $w \in \Sigma^*$ gibt, so dass $v = uw$ ist.
  - Das Wort u wird Präfix von v genannt.
- Das Wort u wird **echtes Präfix** von v genannt, wenn $u \sqsubseteq v$ und weder $u=v$ noch $u=\lambda$ gilt.
- Wir notieren dies als $u \sqsubset v$.
- Gilt $z = uvw$ für Wörter $u,v,w,z \in \Sigma^*$, so ist v ein **Teilwort** von z.

Beispiel Wort

Für jedes Wort w gilt $\lambda \sqsubseteq w$.

- Ebenso ist das leere Wort Teilwort eines jeden beliebigen Wortes!
- Sei $\Sigma := \{l,r\}$, dann ist für $w := llrlrllr$ gerade $|w| = 8$, $|w|_l = 5$, und $|w|_r = 3$.

- Sei $D1 := \{w \in \Sigma^* \mid |w|_L = |w|_R \text{ und } \forall u \in \Sigma^* : (u \sqsubseteq w) \rightarrow (|u|_L \geq |u|_R)\}$.
- Diese (formale) Sprache wird (einseitige) Dyck-Sprache über einem Klammerpaar genannt, mit dem Symbol L als öffnender und R als schließender Klammer.
- (D1, ·) ist ein Monoid, weil auch das leere Wort λ dazugehört und mit $u, v \in D1$, stets auch $w := u \cdot v = uv \in D1$ gilt, d.h. $D1^* = D1$ ist.

Untermonoid

- Ein Untermonoid kann niemals von einer endlichen Menge frei erzeugt werden.
- Betrachten man M := {l,lr,rl} $\subseteq/ \Sigma*$ und bildet M^*, so ist auch M^* ein Untermonoid von Σ^*, jedoch ist dieses von M nicht frei erzeugt.
- Das Element lrl $\in M*$ hat zwei verschiedene Zerlegungen in einzelne Erzeugende: lrl = l · rl = lr · l.
- Konkatenation kann auch auf Sprachen erweitert werden. Sei L1, L2 $\subseteq \Sigma^*$, dann definiert man:

$$L1 \cdot L2 = \{u \cdot v \mid u \in L1, v \in L2\}$$

Neben der Konkatenation, benutzen man eine große Zahl weiterer Operationen auf Wörtern, von denen wir hier nicht alle sofort ohne ihren Zusammenhang zu möglichen „An- und Verwendungen“ definieren.

Alphabet, Zeichen, Wort, Sprache

Alphabet Σ ist eine endliche Menge von Symbolen.
Jedes Element $\sigma \in \Sigma$ ist ein *Zeichen* des Alphabets.
Jedes Element $\omega \in \Sigma^*$ wird als *Wort* über Σ bezeichnet.
Jede Teilmenge $L \subseteq \Sigma^*$ ist eine *formale Sprache* über Σ.

- Definition fordert ausdrücklich, dass der Zeichenvorrat Σ einer Sprache nur aus endlich vielen Elementen besteht.
- Menge Σ^* wird als Kleene'sche Hülle bezeichnet und fasst alle endlichen Symbolsequenzen zusammen, die mit Zeichen aus dem Alphabet Σ aufgebaut werden können.

Formale Charakterisierung der Wortmengen

- Wie in den Abbildungen a) und b) in mathematischer Notation beschrieben, unterscheidet sie sich von der Menge Σ+ lediglich dadurch, dass Σ^* auch das leere Wort ε enthält.
- Anders als die Wörter einer Sprache, die stets eine endliche Länge aufweisen, kann eine Sprache L aus unendlich vielen Wörtern bestehen.
- Die Beispiele in Abbildung c) verdeutlichen den Sprachbegriff.

a)

Definition

$$\Sigma^0 := \{\varepsilon\}$$
$$\Sigma^1 := \Sigma$$
$$\Sigma^{n+1} := \{xy \mid x \in \Sigma, y \in \Sigma^n\}$$
$$\Sigma^+ := \bigcup_{i=1}^{\infty} \Sigma^i$$
$$\Sigma^* := \bigcup_{i=0}^{\infty} \Sigma^i$$

b)

Beispiel: $\Sigma := \{a, b\}$

$$\Sigma^0 = \{\varepsilon\}$$
$$\Sigma^1 = \{a, b\}$$
$$\Sigma^2 = \{aa, ab, ba, bb\}$$
$$\ldots$$
$$\Sigma^+ = \{a, b, aa, ab, ba, bb, \ldots\}$$
$$\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, \ldots\}$$

c)

Beispiel: $L := \{ab, ba\}$

$$L^0 = \{\varepsilon\}$$
$$L^1 = \{ab, ba\}$$
$$L^2 = \{abab, abba, baab, baba\}$$
$$\ldots$$
$$L^+ = \{ab, ba, abab, abba, baab, baba, \ldots\}$$
$$L^* = \{\varepsilon, ab, ba, abab, abba, baab, baba, \ldots\}$$

Beispiel Formale Sprachen: Dyck-Sprache D_2

Σ	{(,),[,]}
L	Menge der korrekt geklammerten Ausdrücke
$\in L$	(), [()(())], ()[()](), ...
$\notin L$	(, [()[(])], (x), ...

Beispiel Formale Sprachen: Primzahlen

Σ	{0,1,2,…,9}
L	Menge der Ziffernfolgen, die einer Primzahl entsprechen
$\in L$	2, 3, 5, 7, 11, 13, …
$\notin L$	0, 1, 4, 6, 8, 9, 10, 12, …

Beispiel Formale Sprachen: ABC-Sprache

Σ	{a, b, c}
L	Menge aller geordneten Folgen aus a‘s, b’s und c’s
$\in L$	abc, aabbc, aaaabbbcc, ...
$\notin L$	cba, ababc, abcba, ...

Beispiel Formale Sprachen: Palindromsprache

Σ	{a,b,c,...,z}
L	Menge aller spiegelbildlich angeordneten Zeichenketten
$\in L$	aabaa, reittier, anna, otto
$\notin L$	abab, aaba, abcab

Fragestellungen im Bereich formaler Sprachen

Wortproblem

- Gilt für ein Wort $\omega \in \Sigma^*$ und eine Sprache L die Beziehung $\omega \in L$?

Leerheitsproblem

- Enthält eine Sprache L mindestens ein Wort, gilt also $L \neq \emptyset$?

Endlichkeitsproblem

- Besitzt eine Sprache L nur endlich viele Elemente?

Äquivalenzproblem

- Gilt für zwei Sprachen $L1$ und $L2$ die Beziehung $L1 = L2$?

Spracherzeugung

- Gibt es für eine Sprache L eine Beschreibung, aus der sich alle Wörter systematisch ableiten lassen?

Fragestellungen im Bereich formaler Sprachen

Wortproblem
- Gilt für ein Wort $\omega \in \Sigma^*$ und eine Sprache L die Beziehung $\omega \in L$?

Leerheitsproblem
- Enthält eine Sprache L mindestens ein Wort, gilt also $L \neq \emptyset$?

Endlichkeitsproblem
- Besitzt eine Sprache L nur endlich viele Elemente?

Äquivalenzproblem
- Gilt für zwei Sprachen $L1$ und $L2$ die Beziehung $L1 = L2$?

Spracherzeugung
- Gibt es für eine Sprache L eine Beschreibung, aus der sich alle Wörter systematisch ableiten lassen?

— Die ersten vier Fragestellungen adressieren die *analytischen* Aspekte einer Sprache, auf die wir später im Detail eingehen werden.

— Aktuell interessiert die letzte Fragestellung, die sich mit dem *generativen* Aspekt einer Sprache beschäftigt.

Strukturierte Regeln und elementare Sprachkonstrukte

- Folgt der Aufbau strukturierten Regeln, lassen sich die Wörter einer Sprache mithilfe einer *Grammatik* erzeugen.
- Bei natürlichen Sprache ist man mit diesem Vorgehen vertraut.
- Anstatt alle korrekt geformten Sätze nacheinander aufzulisten, wird eine Reihe von Regeln vereinbart, mit denen sich elementare Sprachkonstrukte systematisch zu komplexen Gebilden zusammensetzen lassen.
- Als Beispiel betrachten wir die Grammatik, die einen kleinen Auszug aus der deutschen Sprache erzeugt:

<Satz> → <Subjekt> <Prädikat> <Objekt>
<Subjekt> → <Artikel><Adjektiv><Substantiv>
<Artikel> → Der | Die | Das
<Adjektiv> → kleine | süße | flinke
<Substantiv> → Eisbär | Elch | Kröte | Maus | Nilpferd
<Prädikat> → mag | fängt | isst
<Objekt> → Kekse | Schokolade | Käsepizza

Nonterminale und Terminale

- In der Terminologie der formalen Sprachen werden die in spitze Klammern gesetzten Platzhalter als *Nonterminale* oder *Nichtterminale* und die nicht weiter ersetzbaren Sprachbestandteile als *Terminale* bezeichnet.
- Für die Beispielgrammatik erhält man die nachstehende Einteilung:

Nonterminale:
<Satz>, <Subjekt>, <Artikel>, <Adjektiv>, <Substantiv>, <Prädikat>, <Objekt>,

Terminale:
Der, Die, Das, kleine, süße, flinke, Eisbär, Elch, Kröte, Maus, Nilpferd, mag, fängt, isst, Kekse, Schokolade, Käsepizza

Anwendung der Produktionen einer Grammatik G

- Dem Nonterminal <Satz> kommt im Beispiel eine besondere Bedeutung zu.
- Es ist immer das erste Symbol, mit dem eine Ableitung beginnt, und wird als *Startsymbol* bezeichnet.
- Das Beispiel zeigt, wie sich aus dem Startsymbol einige mehr oder weniger sinnvolle Sätze der deutschen Sprache ableiten lassen.

<Satz>
⇒ <Subjekt><Prädikat><Objekt>
⇒ <Subjekt> fängt <Objekt>
⇒ <Subjekt> fängt Kekse
⇒ <Artikel><Adjektiv><Substantiv> fängt Kekse
⇒ Das <Adjektiv><Substantiv> fängt Kekse
⇒ Das flinke <Substantiv> fängt Kekse
⇒ Das flinke Nilpferd fängt Kekse

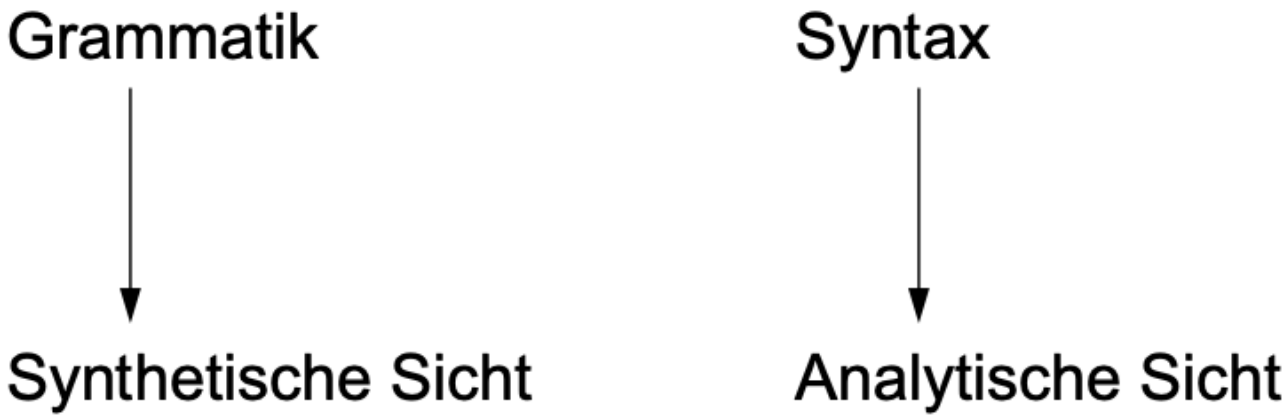
<Satz>
⇒ <Subjekt><Prädikat><Objekt>
⇒ <Subjekt> isst <Objekt>
⇒ <Subjekt> isst Käsepizza
⇒ <Artikel><Adjektiv><Substantiv> isst Käsepizza
⇒ Die <Adjektiv><Substantiv> isst Käsepizza
⇒ Die kleine <Substantiv> isst Käsepizza
⇒ Die kleine Maus isst Käsepizza

Grammatiken

Grammatiken

Um mit Sprachen, die im Allgemeinen unendliche Objekte sind, algorithmisch umgehen zu können, benötigt man endliche Beschreibungsmöglichkeiten für Sprachen.

-> Dazu dienen sowohl Grammatiken als auch Automaten.



Grammatiken

Eine Grammatik wird spezifiziert durch 4 Angaben:

$$G = (V, \Sigma, P, S)$$

V = endliche Menge der **Variablen**
Σ = endliche Menge der **Terminalzeichen**
$S \in V$ = die **Startvariable**
P = endliche Menge der **Regeln** (oder Produktionen)

auch üblich: (V, A, P, S), (N, T, P, S), (S_N, S_T, P, w_S)

- Es gilt: $V \cap \Sigma = \emptyset$
- Regeln „Produktionsregeln“ haben die Form: linke Seite -> rechte Seite
- linke und rechte Seite können aus Variablen (= Nichtterminalzeichen) und Terminalzeichen zusammengesetzt sein

Grammatiken

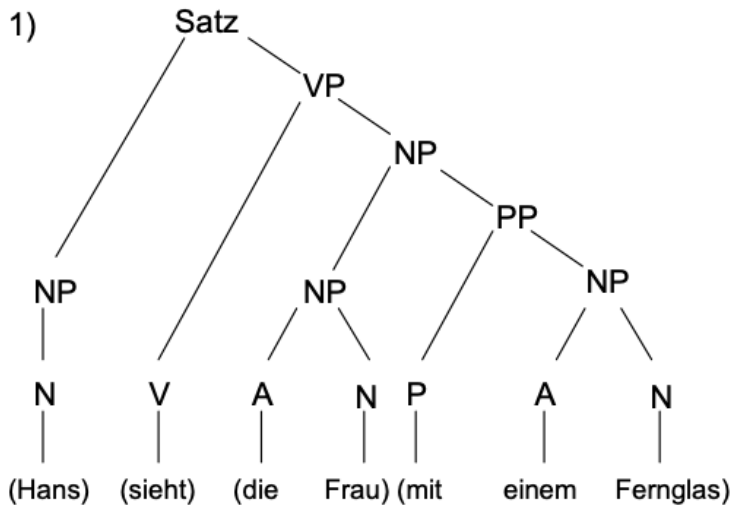
NP = Nominalphrase
VP = Verbalphrase
N = Nomen
A = Artikel
PP = Präpositionalphrase
V = Verb
P = Präposition

Satz	→	NP VP	P	→	mit	V	→	sieht
NP	→	NP PP	P	→	in	V	→	geht
NP	→	N	P	→	auf	A	→	der
NP	→	A N	N	→	Hans	A	→	die
VP	→	V	N	→	Frau	A	→	das
VP	→	V NP	N	→	Fernglas	A	→	einem
VP	→	VP PP	N	→	Park			
PP	→	P NP						

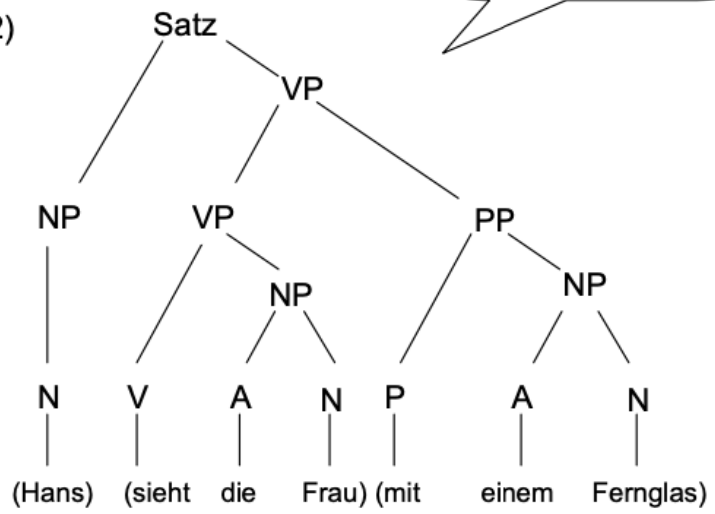
➢ Syntaxbaum zu „Hans sieht die Frau mit einem Fernglas“

Mehrdeutige Grammatik

1)



2)

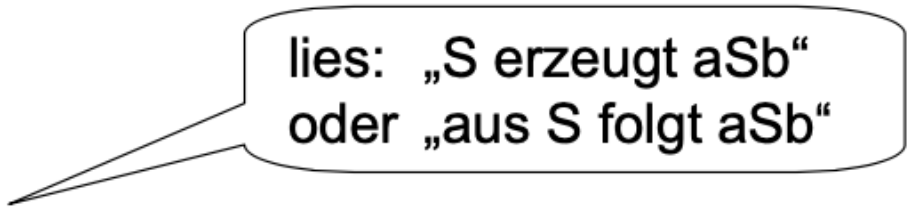


Grammatiken

Beispiel:

$V=\{S\}$

$\Sigma=\{a,b\}$



Regeln 1) $S \rightarrow aSb$

2) $S \rightarrow ab$

$S \Rightarrow ab$

$S \Rightarrow aSb \Rightarrow aabb = a^2b^2$

$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb = a^3b^3$

d. h. Bei dieser Grammatik sind ableitbar alle Wörter der Form a^nb^n, $n \geq 1$

Grammatiken

Problem:

- Sprachen enthalten i. a. unendlich viele Wörter

Ziel:

- Endlichen Formalismus angeben, der in der Lage ist, unendlich viele Sprachen zu bezeichnen.

Beispiel:

- Grammatik aus vorangegangenem Beispiel war kontextfrei, d. h. auf der linken Seite der Regeln steht nur eine Variable.

Grammatik und Ableitungsrelationen

- Durch die Menge der Produktionen definiert jede Grammatik eine Ableitungsrelation $\Rightarrow$ auf der Menge $(V \cup \Sigma)^*$.
- Haben zwei Wörter $x,y \in (V \cup \Sigma)^*$ die Form $x=lur$ und $y=lvr$ mit $l,r \in (V \cup \Sigma)^*$, so gilt $x \Rightarrow y$ genau dann, wenn die Grammatik eine Produktionsregel der Form $u \rightarrow v$ enthält.
- Mit $\Rightarrow^*$ bezeichnet man die reflexiv-transitive Hülle der Ableitungsrelation.
- Verbal ausgedrückt gilt $x \Rightarrow^* y$ genau dann, wenn das Wort y dem Wort x entspricht oder sich in endlich vielen Schritten aus x ableiten lässt.

Grammatiken und Sprachen

- Jede Grammatik *G* erzeugt eine Sprache L (*G*).
- Diese definiert man als die Menge der Wörter über dem Terminalalphabet Σ, die sich aus dem Startsymbol *S* ableiten lassen:

$$L(G) := \{y \in \Sigma^* \mid S \Rightarrow^* y\}$$

- Die Gleichung zeigt, dass die Wörter der Sprache L(*G*) ausschließlich aus Terminalsymbolen bestehen.
- Nonterminale spielen lediglich die Rolle von Platzhaltern, die nach und nach durch Symbole des Terminalalphabets oder durch weitere Nonterminale ersetzt werden.
- Erst wenn alle Nonterminale verschwunden sind, wurde ein Wort der Sprache L (*G*) erzeugt.

Übungen

Aufgabe 1

Überlegen und beschreiben Sie Beispiele von „formalen“ Sprachen, die Sie im Studium, im Alltag oder im Unternehmen kennengelernt haben.

Nutzen Sie die Mengen- und Sprachbeschreibungen, wie Sie es zum Beispiel für Primzahlen kennengelernt haben, um sich einen Überblick zu verschaffen und definieren anschließend die Sprache L.

Aufgabe 2

Gegeben seien die folgenden Alphabete:

$$\Sigma_1 := \{1,2,3,4,5,6,7,8,9\} \qquad \Sigma_3 := \{A,B,C,D,E,F\}$$

$$\Sigma_2 := \{0,1,2,3,4,5,6,7,8,9\} \qquad \Sigma_4 := \{8,9\}$$

Finden Sie umgangssprachliche Beschreibungen für die nachstehenden Sprachen:

a) $\Sigma_2 \mid \Sigma_1\Sigma_2^*$

b) $(\Sigma_2 \cup \Sigma_3) \mid (\Sigma_1 \cup \Sigma_3)(\Sigma_2 \cup \Sigma_3)^*$

c) $(\Sigma_2 \setminus \Sigma_4) \mid (\Sigma_1 \setminus \Sigma_4)(\Sigma_2 \setminus \Sigma_4)^*$

Aufgabe 3

Gegeben seien die folgenden Mengen:

$L_1 := \{aa, bb\}$ $L_2 := \{a\}^+$ $L_3 := \{b\}^+$

Erzeugen Sie die nachstehenden Sprachen:

a) $L_1 \cup L_2$ b) $L_1 \cup L_3$ c) $L_1^* \cap L_2$ d) $L_1^* \cap L_3$

Aufgabe 4

Als Beispiel einer Grammatik haben Sie in diesem Kapitel die folgenden Produktionsregeln für eine Teilmenge der deutschen Sprache kennen gelernt:

<Satz> <Subjekt> <Artikel> <Adjektiv> <Substantiv> <Prädikat> <Objekt>

→ <Subjekt> <Prädikat> <Objekt>
→ <Artikel><Adjektiv><Substantiv> → Der|Die|Das
→ kleine | süße | flinke
→ Eisbär | Elch | Kröte | Maus | Nilpferd → mag | fängt | isst
→ Kekse | Schokolade | Käsepizza

Nicht alle Sätze, die sich aus diesen Produktionen ableiten lassen, sind grammatikalisch korrekt. Wie das folgende Beispiel zeigt, lassen sich Wortsequenzen ableiten, in denen die Satzteile nicht zusammenpassen: *„Das kleine Maus mag Käsepizza“*. Schreiben Sie die Grammatik so um, dass nur solche Sätze ableitbar sind, in denen Artikel und Substantiv sprachlich korrekt kombiniert werden.