

0. Einleitung und Grundbegriffe
1. Endliche Automaten
- 2. Formale Sprachen**
3. Berechenbarkeitstheorie
4. Komplexitätstheorie

Vorbemerkungen und ...

2.1. Chomsky-Grammatiken

2.2. Kontextfreie und reguläre Sprachen

▶ Algorithmische Sprachbeschreibung (durch Lösung des Wortproblems)

- Im letzten Kapitel lernten wir einen Ansatz kennen, Sprachen zu beschreiben, indem man direkt ein „Programm“ zur Lösung des Wortproblems für die zu beschreibende Sprache angibt:

Zur Sprache $L(A)$ gehören alle Zeichenfolgen aus Σ^* , die der Automat A über dem Alphabet Σ akzeptiert.

- Informelles Rezept:

Laufe beim Anfangszustand los - immer in Pfeilrichtung und mit dem Wort w als „Wegbeschreibung“.

Wenn die Wegbeschreibung abgearbeitet ist, schaue nach, ob du in einem der akzeptierenden Zustände in F angekommen bist. Wenn ja, gehört die Wegbeschreibung zu $L(A)$.

► Sprachbeschreibung durch Erzeugung von Wörtern

- Anstatt mit einem Automaten A beliebige Wörter darauf zu **überprüfen**, ob sie zu $L(A)$ gehören, können wir den Automaten auch dazu verwenden, genau die Wörter in $L(A)$ zu **erzeugen**:
- Informelles Rezept:
Laufe immer wieder beim Anfangszustand los, stets in irgendeine Pfeilrichtung, und das beliebig lange. Protokolliere dabei die Zeichen an den durchlaufenen Pfeilen.
Immer wenn du gerade in einem der akzeptierenden Zustände in F angekommen bist, ist dein bisheriges Reiseprotokoll ein Wort aus $L(A)$.
→ systematischer?

*Bei den Automaten Sprachen dient der Automat **beidem**, sowohl der Worterzeugung als auch der Wortprüfung.*

*Ab jetzt lernen wir Sprachen kennen, bei denen sich die **Mechanismen** zur Worterzeugung von denen zur Wortprüfung **unterscheiden**.*

*Dann aber neu zu lösen → **Wortproblem***

- ▶ Grammatiken – zunächst in natürlicher Sprache
 - Regeln, Beispielmenge (stark vereinfacht)
 - <Hauptsatz> → <Subjekt> <Prädikat> <Objekt>.
 - <Subjekt> → Hans | Paul | Maria | Wir
 - <Prädikat> → mag | züchtet | rechnen
 - <Objekt> → Feten | Hunde | TI-Aufgaben
 - Mithilfe dieser Regeln lassen sich u.a. die folgenden Hauptsätze bilden:
 - Hans mag Feten.
 - Wir rechnen TI-Aufgaben.
- ... aber auch
 - Paul rechnen Hunde. ★

0. Einleitung und Grundbegriffe
1. Endliche Automaten
- 2. Formale Sprachen**
3. Berechenbarkeitstheorie
4. Komplexitätstheorie

2.1. Chomsky-Grammatiken

- **Definition und Breitensuche**
- Kontextsensitive Sprachen
- Typen von Chomsky-Grammatiken

2.2. Kontextfreie Sprachen

2.3. Reguläre Sprachen

Kapitel 2: Formale Sprachen

Chomsky-Grammatiken

► Chomsky*-Grammatiken

Bestandteile einer **Chomsky-Grammatik** $G = (\Sigma, V, S, R)$

- ein endliches Alphabet Σ von Zeichen (**Terminalzeichen**)
- eine dazu disjunkte endliche Zeichenmenge V (**Variablen**, Nichtterminale)
- eine ausgezeichnete **Startvariable** $S \in V$
- eine endliche Menge R von **Regeln** (Produktionen)
der Art $l \rightarrow r$ mit $l \in (\Sigma \cup V)^* \setminus \{ \varepsilon \}$ und $r \in (\Sigma \cup V)^*$

beabsichtigte Verwendung:

- *Eine Chomsky-Grammatik G soll die Sprache beschreiben, die alle Wörter $w \in \Sigma^*$ enthält, die man mit Ersetzung(en) gemäß den Regeln dieser Grammatik aus der Startvariablen ableiten kann.*

*) Noam Chomsky, Sprachwissenschaftler und Linksintellektueller,
zwischen 1980 und 1992 die am häufigsten zitierte lebende Person der Welt

Kapitel 2: Formale Sprachen

Chomsky-Grammatiken

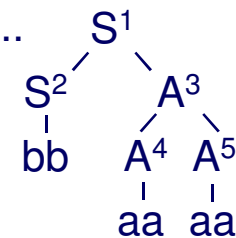
► Beispiel einer Ableitung

- Grammatik $G = (\Sigma, V, S, R)$ mit:
 - $\Sigma = \{ a, b \}$
 - $V = \{ S, A \}$
 - den folgenden Regeln in der Regelmenge R :
 - $S \rightarrow SA$ und $S \rightarrow bb$
 - $A \rightarrow AA$ und $A \rightarrow aa$
- aus der Startvariablen S kann man das Wort $w = bbaaaa$ wie folgt ableiten:

• $S \rightarrow_G SA \rightarrow_G bbA \rightarrow_G bbAA \rightarrow_G bbaaA \rightarrow_G bbaaaa$

(linke Seite \rightarrow rechte Seite)

vgl. mit ...



Kapitel 2: Formale Sprachen

Chomsky-Grammatiken

► zentrale Hilfsbegriffe: Ableitung in einem Schritt

- Es sei $G = [\Sigma, V, S, R]$ eine Grammatik.
- Es seien $u, v \in (\Sigma \cup V)^*$.

v ist genau dann **in einem Schritt** (bzw. **unmittelbar**) mit den Regeln von G aus u **ableitbar** (Formelschreibweise: $u \rightarrow_G v$), wenn es Zeichenketten $x, y \in (\Sigma \cup V)^*$ und eine Regel $l \rightarrow r \in R$ gibt, so dass gilt:

- $u = x \circ l \circ y$
- $v = x \circ r \circ y$

Eine Regel $l \rightarrow r$ kann auf das Wort u angewendet werden, wenn das Wort u die linke Seite l dieser Regel als Teilwort enthält.

Das Ergebnis der Anwendung der Regel $l \rightarrow r$ ist das Wort v , das man erhält, wenn man in u das Teilwort l durch die rechte Seite r dieser Regel ersetzt.

Kapitel 2: Formale Sprachen

Chomsky-Grammatiken

► zentrale Hilfsbegriffe: Ableitung

- Es sei $G = [\Sigma, V, S, R]$ eine Grammatik.
- Es seien $u, v \in (\Sigma \cup V)^*$.

w ist (in endlich vielen Schritten) mit den Regeln von G aus u **ableitbar** (Formelschreibweise: $u \rightarrow_G^* w$),

wenn sich dies induktiv aus folgenden Regeln ergibt:

Induktionsanfang: u ist aus u ableitbar ($u \rightarrow_G^* u$)

Induktionsschritt: Ist v aus u ableitbar und w unmittelbar aus v ableitbar, so ist w aus u ableitbar;
(Wenn $u \rightarrow_G^* v$ und $v \rightarrow_G w$, dann $u \rightarrow_G^* w$)

$$u = u_0 \rightarrow_G u_1 \rightarrow_G \dots \rightarrow_G u_n = w$$

\rightarrow_G^* ist die reflexive transitive Hülle von \rightarrow_G .

Kapitel 2: Formale Sprachen

Chomsky-Grammatiken

- ▶ zentraler Begriff: die durch eine Grammatik beschriebene Sprache
 - Es sei $G = [\Sigma, V, S, R]$ eine Chomsky-Grammatik.
 - Es sei $w \in \Sigma^*$.
 - w gehört zur **durch G beschriebenen Sprache $L(G)$** gdw. sich w (in endlichen vielen Schritten) aus dem Startsymbol S ableiten lässt.

$$L(G) = \{ w \in \Sigma^* \mid S \rightarrow_G^* w \}.$$

$L(G)$ enthält also genau alle Wörter $w \in \Sigma^$, die man mit den Regeln der Grammatik G aus der Startvariablen ableiten kann.*

Kapitel 2: Formale Sprachen

Chomsky-Grammatiken

► Suche nach passender Grammatik – Beispiel 1

- halbformale Beschreibung der interessierenden Sprache L :
 $L = \{ w \mid w \text{ ist Binärdarstellung einer durch 4 teilbaren Zahl} \}$
- *L enthält nur Zeichenketten, die die Symbole 0 und 1 enthalten;*
- *$0 \in L$, und jede Zeichenkette $w \in L$ außer $w = 0$ hat das Präfix 1 und das Suffix 00.*
- Chomsky-Grammatik $G = [\Sigma, V, S, R]$ für L :

Kapitel 2: Formale Sprachen

Chomsky-Grammatiken

► Suche nach passender Grammatik – Beispiel 1

- halbformale Beschreibung der interessierenden Sprache L:
 $L = \{ w \mid w \text{ ist Binärdarstellung einer durch 4 teilbaren Zahl} \}$
- *L enthält nur Zeichenketten, die die Symbole 0 und 1 enthalten;*
- *$0 \in L$, und jede Zeichenkette $w \in L$ außer $w = 0$ hat das Präfix 1 und das Suffix 00.*
- Chomsky-Grammatik $G = [\Sigma, V, S, R]$ für L:

$$\begin{aligned}\Sigma &= \{ 0, 1 \} \\ V &= \{ S, A, B \} \\ S &\end{aligned}$$
$$\begin{aligned}S &\rightarrow 1A & S &\rightarrow 0 \\ A &\rightarrow 1A & A &\rightarrow 0B \\ B &\rightarrow 0 & B &\rightarrow 0B & B &\rightarrow 1A\end{aligned}$$

- *Variable \approx welche Art Wort aus $\{0,1\}^*$ wird hier erwartet?*
S: ist 0 oder hat Präfix 1 und Suffix 00
A: hat Suffix 00
B: ist 0 oder hat Suffix 00

Kapitel 2: Formale Sprachen

Chomsky-Grammatiken

► Suche nach passender Grammatik – Beispiel 2

- Sei $L = \{ w \in \{0,1\}^* \mid w \text{ enthält das Teilwort } 000 \}$
- *Man kann sich leicht überlegen, dass sich jedes Wort $w \in L$ in der Form $w = u \circ 000 \circ v$ mit $u, v \in \Sigma^*$ schreiben lässt*
- Chomsky-Grammatik $G = [\Sigma, V, S, R]$ für L ?

Kapitel 2: Formale Sprachen

Chomsky-Grammatiken

► Suche nach passender Grammatik – Beispiel 2

- Sei $L = \{ w \in \{0,1\}^* \mid w \text{ enthält das Teilwort } 000 \}$
- *Man kann sich leicht überlegen, dass sich jedes Wort $w \in L$ in der Form $w = u \circ 000 \circ v$ mit $u, v \in \Sigma^*$ schreiben lässt.*
- Chomsky-Grammatik $G = [\Sigma, V, S, R]$ für L :

$$\begin{aligned}\Sigma &= \{ 0, 1 \} \\ V &= \{ S, A \} \\ S &\end{aligned}$$

$$\begin{aligned}S &\rightarrow A000A \\ A &\rightarrow \varepsilon \\ A &\rightarrow 0A \\ A &\rightarrow 1A\end{aligned}$$

S: besteht aus 000 und davor und danach einem beliebigen 0-1 Wort
A: beliebiges 0-1-Wort

*In einer der Regeln ist die rechte Seite kürzer als die linke.
Das will man oft nicht; warum sehen wir später.
Können wir das hier vermeiden? (Ad-hoc-Übung)*

Kapitel 2: Formale Sprachen

Chomsky-Grammatiken

- ▶ Suche nach passender Grammatik – Beispiel 3
 - informelle und mengentheoretische Beschreibung der interessierenden Sprache L:
$$L = \{ ab, aabb, aaabbb, \dots \} = \{ a^n b^n \mid n \in \mathbb{N} \}$$
 - Chomsky-Grammatik $G = [\Sigma, V, S, R]$ für L?

Kapitel 2: Formale Sprachen

Chomsky-Grammatiken

► Suche nach passender Grammatik – Beispiel 3

- informelle und mengentheoretische Beschreibung der interessierenden Sprache L:

$$L = \{ ab, aabb, aaabbb, \dots \} = \{ a^n b^n \mid n \geq 1 \}$$

- Chomsky-Grammatik $G = [\Sigma, V, S, R]$ für L?

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

$$V = \{ S \}$$

S

$$S \rightarrow aSb$$

$$S \rightarrow ab$$

- *Zwiebelschichtentechnik*
... auch z.B. für Klammerungen bei arithmetischen Termen verwendbar

(Zwiebelschichten wachsen laut Internet von innen dazu, Bäume übrigens außen, unter der Rinde ☺)

Kapitel 2: Formale Sprachen

Chomsky-Grammatiken

- ▶ Suche nach passender Grammatik – Beispiel 4
 - informelle und mengentheoretische Beschreibung der interessierenden Sprache L:
$$L = \{ abc, aabbcc, aaabbbccc, \dots \} = \{ a^n b^n c^n \mid n \geq 1 \}$$
 - Chomsky-Grammatik $G = [\Sigma, V, S, R]$ für L?

Kapitel 2: Formale Sprachen

Chomsky-Grammatiken

► Suche nach passender Grammatik – Beispiel 4

- informelle und mengentheoretische Beschreibung der interessierenden Sprache L:

$$L = \{ abc, aabbcc, aaabbbccc, \dots \} = \{ a^n b^n c^n \mid n \in \mathbb{N} \}$$

- Chomsky-Grammatik $G = [\Sigma, V, S, R]$ für L:

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

$$V = \{ S, H, C \}$$

S

$$S \rightarrow abc$$

$$S \rightarrow aHbc$$

$$H \rightarrow abC$$

$$H \rightarrow aHbC$$

$$Cb \rightarrow bC$$

$$Cc \rightarrow cc$$

- Zwiebelschichten- und Samenflugtechnik:*
- C ist der Same für ein c, kann nach rechts treiben und verwandelt sich nur an geeigneten Stellen in ein c.*

Beispiel:

$$\begin{aligned} S &\rightarrow_G aHbc \rightarrow_G aaHbCbc \rightarrow_G aaabCbCbc \rightarrow_G aaabbCCbc \rightarrow_G \\ &aaabbCbCc \rightarrow_G aaabbCbcc \rightarrow_G aaabbbCcc \rightarrow_G aaabbbccc \end{aligned}$$

Kapitel 2: Formale Sprachen

Chomsky-Grammatiken

- ▶ Das Wortproblem kann unüberwindlich sein.
 - Wir kümmern uns jetzt um den algorithmischen Aspekt, es geht also um die Frage:
 - ob und wie man das Wortproblem für eine Sprache lösen kann, die man mit einer Chomsky-Grammatik beschrieben hat.
 - Die generelle Antwort auf die Frage ist enttäuschend, da man zeigen kann, dass es Sprachen L mit folgenden Eigenschaften gibt:
 - Die Sprache L kann man mit einer Chomsky-Grammatik beschreiben.
 - Es gibt nachweislich keinen Algorithmus, mit dem man das Wortproblem für die Sprache L lösen kann.

Um dieser Problematik aus dem Weg zu gehen, lässt man nicht alle Chomsky-Grammatiken als Beschreibungsmittel zu

Kapitel 2: Formale Sprachen

Chomsky-Grammatiken

- ▶ Einschub: Immerhin gibt es ein **Halbentscheidungs**-Verfahren (1)
 - Es sei $G = [\Sigma, V, S, R]$ eine Grammatik
 - Wenn man uns ein Wort vorlegt und fragt, ob es aus $L(G)$ ist, dann wollen wir, zumindest im ja-Falle dies auch sicher herausfinden (und im nein-Falle evtl. ewig weiterrechnen)
 - Dazu brauchen wir lediglich ein Programm, das systematisch alle Wörter von $L(G)$ der Reihe nach produziert.

... *Breitensuche (verwendet eine „Queue“ Q)*

1. Setze $Q = [S]$, $M = \emptyset$.
2. Wenn $Q \neq \emptyset$:
 - Bestimme das erste Element u in Q und
 - streiche u aus Q .
 - Für alle v mit $u \rightarrow_G v$:
 - Hänge v ans Ende von Q ($Q := \text{Add}(Q, v)$).
 - Falls $v \in \Sigma^*$, schreibe v in M ($M := M \cup \{v\}$).
3. Gehe nach 2.

Kapitel 2: Formale Sprachen

Chomsky-Grammatiken

► Einschub: Immerhin gibt es ein Halbentscheidungs-Verfahren (2)

Durch systematisches sukzessives Ableiten kann man so alle Wörter, die zur Sprache $L(G)$ gehören, erzeugen.

- **Wenn** ein Wort w zur Sprache $L(G)$ gehört, genügt ein regelmäßiger Blick in die Menge M der bisher abgeleiteten Wörter, denn dann muss w irgendwann darin auftauchen.
- **Aber** solange das Ableitungsprogramm noch läuft und das Wort noch nicht erzeugt wurde, weiß man **nichts**: w könnte ja später noch abgeleitet werden....
- **Wenn** also ein Wort w **nicht** zur Sprache $L(G)$ gehört und das Ableitungsprogramm ewig weiterläuft, was oft der Fall ist, dann weiß man mit diesem Verfahren „stets nichts“, d.h. **nie** irgend etwas.

0. Einleitung und Grundbegriffe
1. Endliche Automaten
- 2. Formale Sprachen**
3. Berechenbarkeitstheorie
4. Komplexitätstheorie

2.1. Chomsky-Grammatiken

- Definition und Breitensuche
- **Kontextsensitive Sprachen**
- Typen von Chomsky-Grammatiken

2.2. Reguläre Sprachen

2.3. Kontextfreie Sprachen

Kapitel 2: Formale Sprachen

Kontextsensitive Sprachen

► kontextsensitive Grammatiken/Sprachen

- Es sei $G = [\Sigma, V, S, R]$ eine Grammatik

G heißt **kontextsensitive** oder **monotone Grammatik**, falls für alle Regeln $l \rightarrow r$ von G gilt:

- $|l| \leq |r|$ (d.h. die Regel $l \rightarrow r$ ist nicht **verkürzend**)

Eine Sprache L heißt **kontextsensitive Sprache**, falls es eine kontextsensitive Grammatik G mit $L = L(G)$ oder $L(G) \cup \{\varepsilon\}$ gibt¹.

*Das leere Wort kann man
mit einer kontextsensitiven Grammatik
nicht aus S ableiten.*

1) $L = L(G)$ oder $L(G) \cup \{\varepsilon\}$ und $\varepsilon \notin L(G)$ kann auch so geschrieben werden: $L(G) = L \setminus \{\varepsilon\}$.

Kapitel 2: Formale Sprachen

Kontextsensitive Sprachen

► Anmerkungen

- Alle Chomsky-Grammatiken, die wir bisher definiert haben (→ Folien 12 bis 18), enthalten keine verkürzenden Regeln, sind also kontextsensitive Grammatiken – außer in Beispiel 2. Aber auch dort finden wir leicht eine geeignete kontextsensitive Grammatik.
- Eine Beschränkung auf kontextsensitive Grammatiken hat Vor- und Nachteile. Verzichtet man darauf, verkürzende Regeln zu benutzen, kann man ...
 - einerseits bestimmte Sprachen nicht mehr beschreiben (Manche Sprachen kann man nur dann mit Chomsky-Grammatiken beschreiben, wenn man dabei verkürzende Regeln benutzt.),
 - sich andererseits aber sicher sein, dass das Wortproblem für jede so beschriebene Sprache lösbar ist.

Kapitel 2: Formale Sprachen

Kontextsensitive Sprachen

- ▶ kontextsensitive Grammatiken/Sprachen: Wortproblem

Für jede kontextsensitive Sprache L
gibt es einen Algorithmus, der für jedes Wort $w \in \Sigma^*$ entscheidet,
ob $w \in L$ oder $w \notin L$ gilt.

*Ist $\varepsilon \in L$, kann in einem ersten Schritt $w = \varepsilon$ überprüft werden.
Falls nein, geht es anschließend um die Zugehörigkeit zur
Sprache $L(G) = L \setminus \{\varepsilon\}$ einer kontextsensitiven Grammatik G .*

- ▶ kontextsensitive Grammatiken/Sprachen: Lösung des Wortproblems
Algorithmische Werkzeuge
 - Als Datenstruktur wird eine Warteschlange Q benutzt
Dabei werden folgende Bezeichnungen verwendet:
 - $[]$ bezeichnet die leere Warteschlange.
 - $[S]$ bezeichnet die Warteschlange mit dem einzigen Eintrag S .
 - $\text{Head}(Q)$ bezeichnet das erste Element in einer nichtleeren Warteschlange Q .
 - $\text{Tail}(Q)$ bezeichnet die Warteschlange, die übrig bleibt, wenn aus der nichtleeren Warteschlange Q das erste Element entfernt wird.
 - $\text{Add}(Q,v)$ bezeichnet die Warteschlange, die entsteht, wenn das Element v ans Ende der Warteschlange Q angefügt wird.

Kapitel 2: Formale Sprachen

Kontextsensitive Sprachen

► kontextsensitive Grammatiken/Sprachen: Lösung des Wortproblems

- Es sei $G = [\Sigma, V, S, R]$ eine kontextsensitive Grammatik;
- es sei $w \in \Sigma^*$.

Idee: Breitensuche mit Länge als Abbruch-Kriterium

(Verwende eine „Queue“ Q und eine Menge E für alle bereits „erledigten“ (irgendwann in Q eingereichten) abgeleiteten Wörter.)

1. Setze $Q = [S]$, $E = \emptyset$.
2. Wenn $Q \neq []$:
 - Setze $u = \text{Head}(Q)$, $Q = \text{Tail}(Q)$.
 - Für alle v mit $u \rightarrow_G v$:
 - Falls $v = w$, gib „ $w \in L(G)$ “ aus und stoppe;
 - Falls $|v| \leq |w|$ und $v \notin E$ gilt,
setze $E = E \cup \{v\}$, $Q = \text{Add}(Q, v)$.
 - Gehe nach (2).
3. Gib „ $w \notin L(G)$ “ aus und stoppe.

- E ist verzichtbar. Es verhindert nur unnötige Wiederholungen.

Kapitel 2: Formale Sprachen

Kontextsensitive Sprachen

► Beurteilung der algorithmischen Idee

- Nach endlich vielen Schritten wird ein korrektes **Ergebnis** ausgegeben, denn ...
 - Es gibt überhaupt nur endlich viele Zeichenketten der Länge kleiner gleich $|w|$ über $\Sigma \cup V$;
 - es werden sukzessive alle überhaupt ableitbaren Zeichenketten der Länge kleiner gleich $|w|$ zur „Queue“ Q hinzugefügt,
 - wenn auch nicht unbedingt in einer besonderen Reihenfolge.
- Der Algorithmus ist leider (trotz E-Benutzung) hoffnungslos **ineffizient**. Aber es ist kein qualitativ besserer Algorithmus bekannt, und man vermutet, dass es keinen besseren gibt.
- Wir werden in diesem Kapitel noch spezielle Typen von Chomsky-Grammatiken kennenlernen, für deren Sprachen das Wortproblem **effizient** lösbar ist.

0. Einleitung und Grundbegriffe
1. Endliche Automaten
- 2. Formale Sprachen**
3. Berechenbarkeitstheorie
4. Komplexitätstheorie

2.1. Chomsky-Grammatiken

- Definition
- Kontextsensitive Sprachen
- **Typen von Chomsky-Grammatiken**

2.2. Kontextfreie und reguläre Sprachen

- ▶ Begriffe: Weitere Typen von Chomsky-Grammatiken
 - Es werden vier Typen unterschieden:
 - **Typ-0** Grammatiken (/ * Synonym: **Chomsky**-Grammatiken)
Sprachen, die man mit Typ-0 Grammatiken beschreiben kann, nennt man **rekursiv aufzählbare** Sprachen.
 - **Typ-1** Grammatiken (/ * Synonym: **kontextsensitive** Grammatiken)
Sprachen, die man mit Typ-1 Grammatiken beschreiben kann, nennt man kontextsensitive Sprachen.
 - **Typ-2** Grammatiken (/ * Synonym: **kontextfreie** Grammatiken)
Sprachen, die man mit Typ-2 Grammatiken beschreiben kann, nennt man kontextfreie Sprachen.
 - **Typ-3** Grammatiken (/ * Synonym: **reguläre** Grammatiken)
Sprachen, die man mit Typ-3 Grammatiken beschreiben kann, nennt man reguläre Sprachen.

L beschreiben soll heißen: $L \setminus \{\varepsilon\}$ erzeugen.

Kapitel 2: Formale Sprachen

Typen von Chomsky-Grammatiken

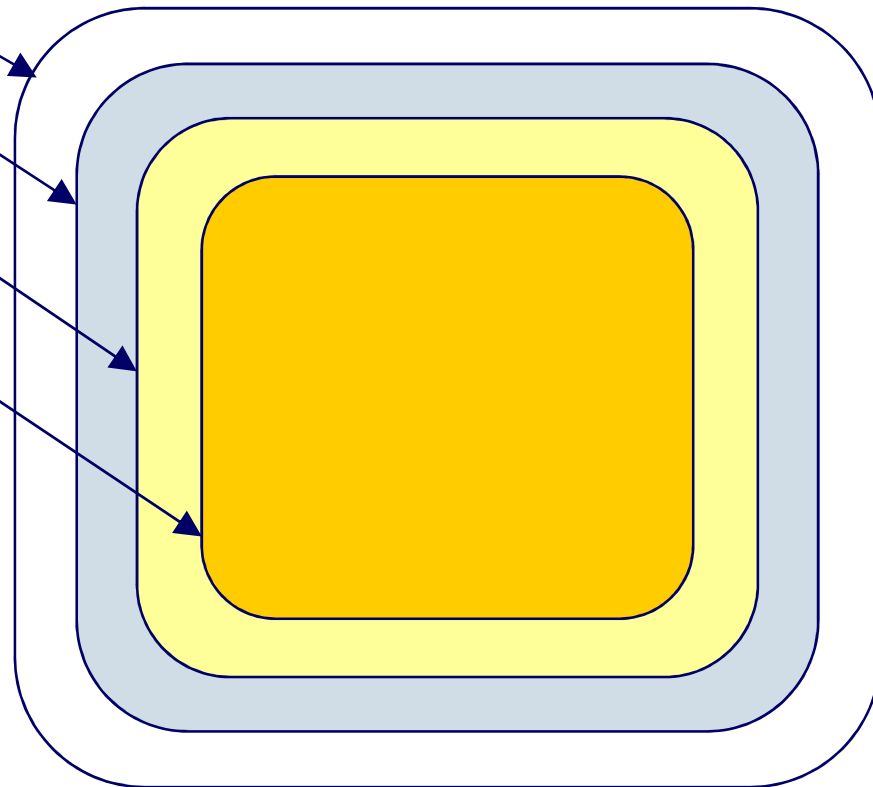
► Hierarchie der Sprachklassen

rekursiv aufzählbare Sprachen

kontextsensitive Sprachen

kontextfreie Sprachen

reguläre Sprachen



0. Einleitung und Grundbegriffe
1. Endliche Automaten
- 2. Formale Sprachen**
3. Berechenbarkeitstheorie
4. Komplexitätstheorie

2.1. Chomsky-Grammatiken

- Definition
- Kontextsensitive Sprachen
- Typen von Chomsky-Grammatiken

2.2. Kontextfreie und reguläre Sprachen

Kapitel 2: Formale Sprachen

Typen von Chomsky-Grammatiken

► kontextfreie (Typ-2-) Grammatiken/Sprachen

Es sei $G = [\Sigma, V, S, R]$ eine Chomsky-Grammatik.

G heißt **kontextfreie Grammatik**, wenn für alle Regeln $l \rightarrow r$ von G gilt:

- $l \in V$ (d.h. die linke Seite der Regel ist eine Variable)
- $r \neq \varepsilon$ (d.h. die rechte Seite der Regel ist ein nichtleeres Wort aus $(\Sigma \cup V)^*$)

Eine Sprache L heißt **kontextfreie Sprache**, falls es eine kontextfreie Grammatik G mit $L = L(G)$ oder $L(G) \cup \{\varepsilon\}$ gibt.

Auch eine kontextfreie Grammatik kann keine verkürzenden Regeln enthalten, ist also monoton (kontextsensitiv).

Kapitel 2: Formale Sprachen

Typen von Chomsky-Grammatiken

► Anmerkungen

- Da keine kontextfreie Grammatik verkürzende Regeln enthält, also jede gleichzeitig eine kontextsensitive Grammatik ist, weiß man folgendes:
 - Jede kontextfreie Sprache ist gleichzeitig eine kontextsensitive Sprache.
 - Für jede kontextfreie Sprache ist das Wortproblem lösbar (sogar effizient, wie wir noch sehen werden).
- Wir haben bereits einige Beispiele kontextfreier Sprachen gesehen:
 - Beispiel 1: durch 4 teilbare Binärzahlen
 - Beispiel 2: $\{ w \in \Sigma^* \mid w \text{ enthält Teilwort } 000 \}$
(kontextfrei zumindest nach der „Reparatur“ der Grammatik)
 - Beispiel 3: $\{ a^n \cdot b^n \mid n \geq 1 \}$

▶ Mathematik/Informatik-Deutsch

Mathematische Bezeichnungen folgen nicht unbedingt sprachlich intuitiven Gepflogenheiten, was zwei Beispiele aus der Algebra zeigen sollen:

- Eine Halbgruppe ist weder eine besondere noch ein Teil einer Gruppe. Vielmehr ist eine Gruppe eine besondere Art von Halbgruppe.
- Ein Schiefkörper ist nicht eine besondere Art von Körper. Vielmehr ist ein Körper eine besondere Art von Schiefkörper.

Und in der theoretischen Informatik haben wir jetzt kontextfreie und kontextsensitive Grammatiken kennengelernt,

- ohne dass irgendwelche Parallelen oder Gegensätze in den Definitionen erkennbar wären,
- noch der Bezug zu einem „Kontext“ (Umgebung, Zusammenhang).

Kapitel 2: Formale Sprachen

Typen von Chomsky-Grammatiken

► kontextsensitiv vs. kontextfrei, Wort-Herkunft

Der Begriff „kontextsensitiv“ wird im Vergleich zu „kontextfrei“ klarer wenn wir eine andere Art von Grammatiken betrachten:

- Es sei $G = [\Sigma, V, S, R]$ eine Grammatik.

G heißt (hier zur Unterscheidung) **klassisch-kontextsensitive Grammatik**, falls für alle Regeln $l \rightarrow r$ von G gilt:

- Es existieren $u, v, w \in (\Sigma \cup V)^*$ mit $v \neq \varepsilon$, und ein $x \in V$, so dass $l = uxw$ und $r = uvw$.

„ $x \rightarrow v$ in der Umgebung (Kontext) $u \dots w$.“

Spezialfall: Wenn stets $u = w = \varepsilon$ gilt, dann ist G kontextfrei.

Ohne Beweis:

Kontextsensitive (monotone) und klassisch-kontextsensitive Grammatiken erzeugen die gleiche Familie von (kontextsensitiven) Sprachen.

► reguläre (Typ-3-) Grammatiken/Sprachen

- Es sei $G = [\Sigma, V, S, R]$ eine Chomsky-Grammatik

G heißt **reguläre Grammatik**, falls für alle Regeln $l \rightarrow r$ von G gilt:

- $l \in V$ (d.h. die linke Seite ist eine Variable)
- $r \in \Sigma$ (d.h. die rechte Seite ist ein Terminalzeichen)
oder
 $r = xA$ mit $x \in \Sigma$ und $A \in V$ (die rechte Seite ist eine Konstante gefolgt von einer Variablen)

Eine Sprache L heißt **reguläre Sprache**, falls es eine reguläre Grammatik G mit $L(G) = L$ oder $L(G) \cup \{\varepsilon\}$ gibt.

Jede reguläre Grammatik ist kontextfrei.

Kapitel 2: Formale Sprachen

Typen von Chomsky-Grammatiken

► Beispiele für Regelmengen

- kontextsensitive Grammatik

```
S → abc  
S → aHbc  
H → abC  
H → aAbC  
Cb → bC  
Cc → cc
```

- kontextfreie Grammatik

```
S → aSb  
S → ab
```

- reguläre Grammatik

```
S → 1A  
A → 0B  
A → 1A  
B → 0  
B → 0B  
B → 1A
```

Kapitel 2: Formale Sprachen

Typen von Chomsky-Grammatiken

► Anmerkungen

- Da jede reguläre Grammatik gleichzeitig sowohl eine kontextsensitive als auch eine kontextfreie Grammatik ist, weiß man folgendes:
 - Jede reguläre Sprache ist gleichzeitig sowohl eine kontextsensitive als auch eine kontextfreie Sprache
 - Für jede reguläre Sprache ist das Wortproblem lösbar.
Wir werden sehen, dass man es sogar besonders effizient lösen kann.
- Wir haben bisher in Beispiel 1 eine reguläre Grammatik kennengelernt.
- Wir werden aber sehen, dass auch in Beispiel 2 mit $L = \{ w \in \Sigma^* \mid w \text{ enthält das Teilwort } 000 \}$ mit $\Sigma = \{ 0,1 \}$, eine reguläre Sprache vorliegt.

Kapitel 2: Formale Sprachen

Typen von Chomsky-Grammatiken

► Eine „reguläre Regelmenge“ für Beispiel 2

$$\Sigma = \{ 0,1 \}, \quad L = \{ w \in \Sigma^* \mid w \text{ enthält Teilwort } 000 \}$$

- Beginnend mit S erzeugen wir:
 - Eine beliebiges 0-1-Wort, $S \rightarrow 0S, S \rightarrow 1S$
 - und irgendwann 000, gefolgt von einem beliebigen 0-1-Wort, d.h. eine 0, gefolgt von einem Wort, das mit **Zwei** Nullen beginnt:
 $S \rightarrow 0Z$
- Aus Z erzeugen wir eine 0, gefolgt von einem Wort, das mit **Einer** Null beginnt:
 $Z \rightarrow 0E$
- E liefert 0 oder 0, gefolgt von einem **B**eliebigen nichtleeren 0-1-Wort:
 $E \rightarrow 0, E \rightarrow 0B$
- B erzeugt beliebiges nichtleeres 0-1-Wort, d.h. 0 oder 1, entweder alleine oder gefolgt von einem beliebigen nichtleeren 0-1-Wort :
 $B \rightarrow 0, B \rightarrow 0B, B \rightarrow 1, B \rightarrow 1B$