Vorlesung

Grundlagen der Theoretischen Informatik / Einführung in die Theoretische Informatik I

Bernhard Beckert

Institut für Informatik

UNIVERSITÄT KOBLENZ · LANDAU

Sommersemester 2007

Dank

Diese Vorlesungsmaterialien basieren ganz wesentlich auf den Folien zu den Vorlesungen von

Katrin Erk (gehalten an der Universität Koblenz-Landau)

Jürgen Dix (gehalten an der TU Clausthal)

Ihnen beiden gilt mein herzlicher Dank.

– Bernhard Beckert, April 2007

Beispiel

Beispiel 6.15

Grammatik $G_{ab} = (\{S\}, \{a, b\}, \{R_1, R_2\}, S)$ mit

$$R_1 = S \to aSb$$
$$R_2 = S \to \varepsilon$$

Mögliche Ableitung:

$$S \Longrightarrow_{R_1} aSb \Longrightarrow_{R_1} aaSbb \Longrightarrow_{R_1} aaaSbbb \Longrightarrow_{R_2} aaabbb$$

Also: $a^3 b^3 \in L(G_2)$

Lemma 6.16

Die Grammatik G_{ab} erzeugt die Sprache

$$L(G_{ab}) = \{a^n b^n \mid n \in \mathbb{N}_0\}$$

Beispiel

Beispiel 6.15

Grammatik $G_{ab} = (\{S\}, \{a, b\}, \{R_1, R_2\}, S)$ mit

$$R_1 = S \rightarrow aSb$$
$$R_2 = S \rightarrow \varepsilon$$

Mögliche Ableitung:

$$S \Longrightarrow_{R_1} aSb \Longrightarrow_{R_1} aaSbb \Longrightarrow_{R_1} aaaSbbb \Longrightarrow_{R_2} aaabbb$$

Also: $a^3b^3 \in L(G_2)$

Lemma 6.16

Die Grammatik G_{ab} erzeugt die Sprache

$$L(G_{ab}) = \{a^n b^n \mid n \in \mathbb{N}_0\}$$

Beispiel

Beispiel 6.15

Grammatik $G_{ab} = (\{S\}, \{a, b\}, \{R_1, R_2\}, S)$ mit

$$R_1 = S \rightarrow aSb$$
$$R_2 = S \rightarrow \varepsilon$$

Mögliche Ableitung:

$$S \Longrightarrow_{R_1} aSb \Longrightarrow_{R_1} aaSbb \Longrightarrow_{R_1} aaaSbbb \Longrightarrow_{R_2} aaabbb$$

Also: $a^3 b^3 \in L(G_2)$

Lemma 6.16

Die Grammatik G_{ab} erzeugt die Sprache

$$L(G_{ab}) = \{a^n b^n \mid n \in \mathbb{N}_0\}$$

Beispiel 6.15

Grammatik $G_{ab} = (\{S\}, \{a, b\}, \{R_1, R_2\}, S)$ mit

$$\begin{aligned} R_1 &= S \rightarrow aSb \\ R_2 &= S \rightarrow \varepsilon \end{aligned}$$

Mögliche Ableitung:

$$S \Longrightarrow_{R_1} aSb \Longrightarrow_{R_1} aaSbb \Longrightarrow_{R_1} aaaSbbb \Longrightarrow_{R_2} aaabbb$$

Also: $a^3 b^3 \in L(G_2)$

Lemma 6.16

Die Grammatik G_{ab} erzeugt die Sprache

$$L(G_{ab}) = \{a^n b^n \mid n \in \mathbb{N}_0\}$$

Beispiel

Beweis

Dass G_{ab} tatsächlich genau diese Sprache erzeugt, zeigen wir allgemein, indem wir alle möglichen Ableitungen von G_{ab} betrachten.

$\subseteq$: zu zeigen: Jedes terminale Wort, das von G_{ab} erzeugt wird, hat die Form $a^n b^n$.
Wir zeigen für alle $w \in (V \cup T)^*$: Falls $S \Longrightarrow^*_{G_{ab}} w$, dann gilt entweder $w = a^n S b^n$ oder $w = a^n b^n$ für ein $n \in \mathbb{N}_0$.

Beispiel

Beweis

Dass G_{ab} tatsächlich genau diese Sprache erzeugt, zeigen wir allgemein, indem wir alle möglichen Ableitungen von G_{ab} betrachten.

$\subseteq$: zu zeigen: Jedes terminale Wort, das von G_{ab} erzeugt wird, hat die Form $a^n b^n$.
Wir zeigen für alle $w \in (V \cup T)^*$: Falls $S \Longrightarrow^*_{G_{ab}} w$, dann gilt entweder $w = a^n S b^n$ oder $w = a^n b^n$ für ein $n \in \mathbb{N}_0$.

Beispiel

Beweis (Forts.)

Dazu verwenden wir eine **Induktion über die Länge einer Ableitung** von S nach w.

Induktionsanfang: $w = S = a^0 S b^0$

Induktionsschritt: Es gelte $S \Longrightarrow^*_{G_{ab}} w \Longrightarrow_{G_{ab}} w'$, und für w gelte nach der Induktionsvoraussetzung bereits $w = a^n b^n$ oder $w = a^n S b^n$. Außerdem sei $w \Longrightarrow_{G_{ab}} w'$ eine Ableitung in einem Schritt. Nun ist zu zeigen: $w' = a^m b^m$ oder $w' = a^m S b^m$ für irgendein m.

Beispiel

Beweis (Forts.)

Dazu verwenden wir eine **Induktion über die Länge einer Ableitung** von S nach w.

Induktionsanfang: $w = S = a^0 S b^0$

Induktionsschritt: Es gelte $S \Longrightarrow^*_{G_{ab}} w \Longrightarrow_{G_{ab}} w'$, und für w gelte nach der Induktionsvoraussetzung bereits $w = a^n b^n$ oder $w = a^n S b^n$. Außerdem sei $w \Longrightarrow_{G_{ab}} w'$ eine Ableitung in einem Schritt. Nun ist zu zeigen: $w' = a^m b^m$ oder $w' = a^m S b^m$ für irgendein m.

Beispiel

Beweis (Forts.)

Dazu verwenden wir eine **Induktion über die Länge einer Ableitung** von S nach w.

Induktionsanfang: $w = S = a^0 S b^0$

Induktionsschritt: Es gelte $S \Longrightarrow^*_{G_{ab}} w \Longrightarrow_{G_{ab}} w'$, und für w gelte nach der Induktionsvoraussetzung bereits $w = a^n b^n$ oder $w = a^n S b^n$. Außerdem sei $w \Longrightarrow_{G_{ab}} w'$ eine Ableitung in einem Schritt. Nun ist zu zeigen: $w' = a^m b^m$ oder $w' = a^m S b^m$ für irgendein m.

Beispiel

Beweis (Forts.)

Fall 1: $w = a^n b^n$. Dann konnte keine Regel angewandt werden, da w schon terminal ist, also tritt dieser Fall nie auf.

Fall 2: $w = a^n S b^n$. Dann wurde von w nach w' entweder Regel R_1 oder R_2 angewandt.

Falls R_1 angewandt wurde, dann gilt $w = a^n S b^n \Longrightarrow_{R_1} a^n a S b b^n = a^{n+1} S b^{n+1} = w'$.

Falls R_2 angewandt wurde, dann gilt $w = a^n S b^n \Longrightarrow_{R_2} a^n \varepsilon b^n = w'$. Dies Wort ist terminal und hat die geforderte Form $a^n b^n$.

Beispiel

Beweis (Forts.)

Fall 1: $w = a^n b^n$. Dann konnte keine Regel angewandt werden, da w schon terminal ist, also tritt dieser Fall nie auf.

Fall 2: $w = a^n S b^n$. Dann wurde von w nach w' entweder Regel R_1 oder R_2 angewandt.

Falls R_1 angewandt wurde, dann gilt $w = a^n S b^n \Longrightarrow_{R_1} a^n a S b b^n = a^{n+1} S b^{n+1} = w'$.

Falls R_2 angewandt wurde, dann gilt $w = a^n S b^n \Longrightarrow_{R_2} a^n \varepsilon b^n = w'$. Dies Wort ist terminal und hat die geforderte Form $a^n b^n$.

Beispiel

Beweis (Forts.)

$\supseteq$: zu zeigen: Für alle n kann $a^n b^n$ von G_{ab} erzeugt werden: $S \Longrightarrow^*_{G_{ab}} a^n b^n$ $\forall n \in \mathbb{N}_0$.

Um $a^n b^n$ zu erzeugen, wende man auf S n-mal die Regel R_1 und dann einmal die Regel R_2 an. $\square$

Beispiel: Dycksprache

Definition 6.17 (Dycksprache)

Gegeben:

- $k \in \mathbb{N}$
- $\Sigma_k := \{x_1, \overline{x_1}, x_2 \ldots, x_k, \overline{x_k}\}$ ein Alphabet mit $2k$ Symbolen

Die Dycksprache D_k ist die kleinste Menge die folgende Bedingungen erfüllt:

1. $\varepsilon \in D_k$,
2. Falls $w \in D_k$, so auch $x_i w \overline{x_i}$.
3. Falls $u, v \in D_k$, so auch uv.

Interpretiert man die x_i als öffnende, die $\overline{x_i}$ als zugehörige schließende Klammern, so kann man die Dycksprache als die **Menge aller korrekten Klammerausdrücke** sehen.

Beispiel: Dycksprache

Definition 6.17 (Dycksprache)

Gegeben:

– $k \in \mathbb{N}$

– $\Sigma_k := \{x_1, \overline{x_1}, x_2 \ldots, x_k, \overline{x_k}\}$ ein Alphabet mit $2k$ Symbolen

Die Dycksprache D_k ist die kleinste Menge die folgende Bedingungen erfüllt:

1. $\varepsilon \in D_k$,
2. Falls $w \in D_k$, so auch $x_i w \overline{x_i}$.
3. Falls $u, v \in D_k$, so auch uv.

Interpretiert man die x_i als öffnende, die $\overline{x_i}$ als zugehörige schließende Klammern, so kann man die Dycksprache als die **Menge aller korrekten Klammerausdrücke** sehen.

Beispiel: Dycksprache

Definition 6.17 (Dycksprache)

Gegeben:

- $k \in \mathbb{N}$
- $\Sigma_k := \{x_1, \overline{x_1}, x_2 \ldots, x_k, \overline{x_k}\}$ ein Alphabet mit $2k$ Symbolen

Die Dycksprache D_k ist die kleinste Menge die folgende Bedingungen erfüllt:

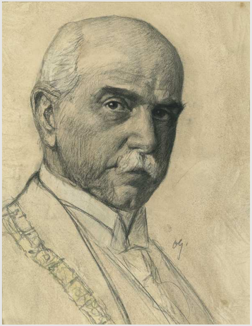
1. $\varepsilon \in D_k$,
2. Falls $w \in D_k$, so auch $x_i w \overline{x_i}$.
3. Falls u, $v \in D_k$, so auch uv.

Interpretiert man die x_i als öffnende, die $\overline{x_i}$ als zugehörige schließende Klammern, so kann man die Dycksprache als die **Menge aller korrekten Klammerausdrücke** sehen.

Beispiel: Dycksprache

Walther von Dyck $\star$ 1856, † 1934

- Mathematiker
- Hochschulpolitiker
- Erster Rektor der TU München
- Einer der Gründungsväter des Deutschen Museums



[Foto: Deutsches Museum]

Teil II

Teil II

Darstellung von Problemen

Fakt

So ziemlich alle Probleme können als Probleme über Sprachen formuliert werden.

Beispiel 7.1 (Primzahlen)

Alphabet $\Sigma_{\text{num}} := \{|\}$

Sprache $L_{\text{primes}} := \{\underbrace{||\ldots|}_{p \text{ mal}} \mid p \text{ prim}\}$

Darstellung von Problemen

Fakt

So ziemlich alle Probleme können als Probleme über Sprachen formuliert werden.

Beispiel 7.1 (Primzahlen)

Alphabet $\Sigma_{\text{num}} := \{|\}$

Sprache $L_{\text{primes}} := \{\underbrace{||\ldots|}_{p \text{ mal}} \mid p \text{ prim}\}$

Darstellung von Problemen

Eingabealphabet

$$\Sigma = \{0, 1, \ldots, n-1\}$$

erlaubt Darstellung einer Ganzzahl zur Basis *n*

Beispiel 7.2

5 binär: 101

5 unär: $|||||$ (oder auch 11111)

Darstellung von Problemen

Eingabealphabet

$$\Sigma = \{0, 1, \dots, n-1\}$$

erlaubt Darstellung einer Ganzzahl zur Basis *n*

Beispiel 7.2

5 binär: 101

5 unär: $|||||$ (oder auch 11111)

Darstellung von Problemen

Speicheraufwand

***n*-äre Darstellung** ($n > 1$) einer Zahl k führt zu einer Speicherersparnis:

$$\log_n k \quad (n\text{-är}) \qquad \text{statt} \qquad k \quad (\text{unär})$$

Nur der Schritt von unär auf binär ist wesentlich, denn

$$\log_n k = \frac{1}{\log_2 n} \cdot \log_2 k = c \cdot \log_2 k$$

(von binär auf *n*-är nur lineare Einsparung)

Darstellung von Problemen

Speicheraufwand

***n*-äre Darstellung** ($n > 1$) einer Zahl k führt zu einer Speicherersparnis:

$$\log_n k \quad (n\text{-är}) \qquad \text{statt} \qquad k \quad (\text{unär})$$

Nur der Schritt von unär auf binär ist wesentlich, denn

$$\log_n k = \frac{1}{\log_2 n} \cdot \log_2 k = c \cdot \log_2 k$$

(von binär auf n-är nur lineare Einsparung)

Darstellung des Erfüllbarkeitsproblems SAT

Problem SAT

Gegeben: Eine aussagenlogische Formel w

Frage: Gibt es eine Belegung der booleschen Variablen in w, so dass w zu *true* auswertet?

Signatur für aussagenlogische Formeln

Signatur: $\Sigma_{\text{sat}} := \{\wedge, \vee, \neg, (,), x, 0, 1\}$

Dabei Darstellung von boolscher Variablen x_i als x gefolgt von i binär kodiert.
Dadurch Formel der Länge n um (unerheblichen) Faktor $\log n$ länger.

Darstellung des Erfüllbarkeitsproblems SAT

Problem SAT

Gegeben: Eine aussagenlogische Formel w

Frage: Gibt es eine Belegung der booleschen Variablen in w, so dass w zu *true* auswertet?

Signatur für aussagenlogische Formeln

Signatur: $\Sigma_{\mathrm{sat}} := \{\wedge, \vee, \neg, (,), x, 0, 1\}$

Dabei Darstellung von boolscher Variablen x_i als x gefolgt von i binär kodiert. Dadurch Formel der Länge n um (unerheblichen) Faktor $\log n$ länger.

Darstellung des Erfüllbarkeitsproblems SAT

Definition 7.3 (Satisfiability)

Sprache

$$L_{\mathrm{sat}} := \{ w \in \Sigma_{sat}^* : \quad \begin{array}{l} w \text{ ist eine aussagenlogische Formel,} \\ \text{und es gibt eine Belegung für die } x_i, \\ \text{so dass die Formel } w \text{ zu } \textit{true} \text{ auswertet} \end{array} \}$$

Darstellung des Erreichbarkeitsproblems in Graphen

Erreichbarkeitsproblem

Gegeben: Ein Graph mit Ecken v_1 bis v_n

Frage: Gibt es einen Weg von Ecke v_1 zu Ecke v_n?

Signatur für Graphen

Signatur: $\Sigma_{\text{graph}} := \{v, e, 0, 1, (,), \#\}$

Darstellung von

Ecke v_i als v gefolgt i binär kodiert

Kante $e_{i,j}$ als $e(string_1\#string_2)$, wobei

- $string_1$ die binäre Darstellung von i,
- $string_2$ die binäre Darstellung von j

Darstellung des Erreichbarkeitsproblems in Graphen

Erreichbarkeitsproblem

Gegeben: Ein Graph mit Ecken v_1 bis v_n

Frage: Gibt es einen Weg von Ecke v_1 zu Ecke v_n?

Signatur für Graphen

Signatur: $\Sigma_{\text{graph}} := \{v, e, 0, 1, (,), \#\}$

Darstellung von

Ecke v_i als v gefolgt i binär kodiert

Kante $e_{i,j}$ als $e(string_1\#string_2)$, wobei

- $string_1$ die binäre Darstellung von i,
- $string_2$ die binäre Darstellung von j

Darstellung des Erreichbarkeitsproblems in Graphen

Erreichbarkeitsproblem

Gegeben: Ein Graph mit Ecken v_1 bis v_n

Frage: Gibt es einen Weg von Ecke v_1 zu Ecke v_n?

Signatur für Graphen

Signatur: $\Sigma_{\text{graph}} := \{v, e, 0, 1, (,), \#\}$

Darstellung von

Ecke v_i als v gefolgt i binär kodiert

Kante $e_{i,j}$ als $e(string_1\#string_2)$, wobei
- $string_1$ die binäre Darstellung von i,
- $string_2$ die binäre Darstellung von j

Darstellung des Erreichbarkeitsproblems in Graphen

Definition 7.4 (Erreichbarkeitsproblem)

Sprache

$$L_{\text{reach}} := \{ w \in \Sigma^*_{\text{graph}} : \begin{array}{l}\text{es gibt einen Weg in } w \\ \text{von der ersten Ecke } v_1 \\ \text{zur letzten Ecke } v_n \end{array} \}$$

Teil II

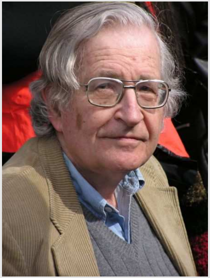
Teil II

1. Sprache, Grammatik
2. Warum Sprachen?
3. **Die Chomsky-Hierarchie**
4. Probleme über Sprachen
5. Endlich, unendlich und dann?

Die Chomsky-Hierarchie

Noam Chomsky ⋆ 1928

- Professor für Linguistik und Philosophie am MIT
- Bedeutender Linguist
- Bedeutender Beitrag zur Informatik:
  Erste Beschreibung der Chomsky-Hierarchie (1956)
- Bedeutender linker Intellektueller und Globalisierungskritiker



Die Chomsky-Hierarchie

Was muss eine Grammatik erfüllen?

- Sie darf nur **endlich viele Regeln** haben
- Jede Regelprämisse muss **mindestens eine Variable** enthalten

Das Wort kann im Lauf der Ableitung beliebig wachsen und wieder schrumpfen.

(Weitere) Beschränkung der Form, die Regeln haben dürfen, führt zu
– **Grammatiktypen** und damit auch zu
– **Sprachtypen**
von verschiedenen Schwierigkeitsgraden.

Die Chomsky-Hierarchie

Was muss eine Grammatik erfüllen?

- Sie darf nur **endlich viele Regeln** haben
- Jede Regelprämisse muss **mindestens eine Variable** enthalten

Das Wort kann im Lauf der Ableitung beliebig wachsen und wieder schrumpfen.

(Weitere) Beschränkung der Form, die Regeln haben dürfen, führt zu
- **Grammatiktypen** und damit auch zu
- **Sprachtypen**

von verschiedenen Schwierigkeitsgraden.

Die Chomsky-Hierarchie

Was muss eine Grammatik erfüllen?

- Sie darf nur **endlich viele Regeln** haben
- Jede Regelprämisse muss **mindestens eine Variable** enthalten

Das Wort kann im Lauf der Ableitung beliebig wachsen und wieder schrumpfen.

(Weitere) Beschränkung der Form, die Regeln haben dürfen, führt zu
– **Grammatiktypen** und damit auch zu
– **Sprachtypen**
von verschiedenen Schwierigkeitsgraden.

Die Chomsky-Hierarchie

Was muss eine Grammatik erfüllen?

- Sie darf nur **endlich viele Regeln** haben
- Jede Regelprämisse muss **mindestens eine Variable** enthalten

Das Wort kann im Lauf der Ableitung beliebig wachsen und wieder schrumpfen.

(Weitere) Beschränkung der Form, die Regeln haben dürfen, führt zu
– **Grammatiktypen** und damit auch zu
– **Sprachtypen**
von verschiedenen Schwierigkeitsgraden.

Die Chomsky-Hierarchie

Was muss eine Grammatik erfüllen?

- Sie darf nur **endlich viele Regeln** haben
- Jede Regelprämisse muss **mindestens eine Variable** enthalten

Das Wort kann im Lauf der Ableitung beliebig wachsen und wieder schrumpfen.

(Weitere) Beschränkung der Form, die Regeln haben dürfen, führt zu
– **Grammatiktypen** und damit auch zu
– **Sprachtypen**
von verschiedenen Schwierigkeitsgraden.

Definition 8.1 (Rechtlineare Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **rechtslinear** gdw

$$\forall (P \to Q) \in R \ \big(P \in V \text{ und } Q \in T^* \cup T^+ V\big)$$

Das heißt, bei jeder Regelanwendung:

- Links eine **einzelne Variable**
- Rechts **höchstens eine Variable**
- Wenn rechts eine Variable steht, steht sie **ganz rechts im Wort**.

Die Chomsky-Hierarchie

Definition 8.1 (Rechtlineare Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **rechtslinear** gdw

$$\forall (P \to Q) \in R \ \left(P \in V \text{ und } Q \in T^* \cup T^+ V\right)$$

Das heißt, bei jeder Regelanwendung:

- Links eine **einzelne Variable**
- Rechts **höchstens eine Variable**
- Wenn rechts eine Variable steht, steht sie **ganz rechts im Wort**.

Die Chomsky-Hierarchie

Definition 8.1 (Rechtlineare Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **rechtslinear** gdw

$$\forall (P \rightarrow Q) \in R \;\; \big(P \in V \text{ und } Q \in T^* \cup T^+ V\big)$$

Das heißt, bei jeder Regelanwendung:

- Links eine **einzelne Variable**
- Rechts **höchstens eine Variable**
- Wenn rechts eine Variable steht, steht sie **ganz rechts im Wort**.

Die Chomsky-Hierarchie

Definition 8.1 (Rechtlineare Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **rechtslinear** gdw

$$\forall (P \to Q) \in R \ \big(P \in V \text{ und } Q \in T^* \cup T^+ V\big)$$

Das heißt, bei jeder Regelanwendung:

- Links eine **einzelne Variable**
- Rechts **höchstens eine Variable**
- Wenn rechts eine Variable steht, steht sie **ganz rechts im Wort.**

Die Chomsky-Hierarchie

Definition 8.2 (Kontextfreie Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **kontextfrei** gdw

$$\forall (P \to Q) \in R \; \big(P \in V \text{ und } Q \in (V \cup T)^* \big)$$

Das heißt, bei jeder Regelanwendung:

- Links eine **einzelne Variable**
- Die Prämisse macht keine Aussage, was der Kontext dieser Variablen ist („kontextfrei")
- Rechts steht etwas beliebiges

Die Chomsky-Hierarchie

Definition 8.2 (Kontextfreie Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **kontextfrei** gdw

$$\forall (P \to Q) \in R \; \big(P \in V \text{ und } Q \in (V \cup T)^*\big)$$

Das heißt, bei jeder Regelanwendung:

- Links eine **einzelne Variable**
- Die Prämisse macht keine Aussage, was der Kontext dieser Variablen ist („kontextfrei")
- Rechts steht etwas beliebiges

Die Chomsky-Hierarchie

Definition 8.2 (Kontextfreie Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **kontextfrei** gdw

$$\forall (P \to Q) \in R \ \Big(P \in V \text{ und } Q \in (V \cup T)^* \Big)$$

Das heißt, bei jeder Regelanwendung:

- Links eine **einzelne Variable**
- Die Prämisse macht keine Aussage, was der Kontext dieser Variablen ist („kontextfrei")
- Rechts steht etwas beliebiges

Die Chomsky-Hierarchie

Definition 8.2 (Kontextfreie Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **kontextfrei** gdw

$$\forall (P \to Q) \in R \ \big(P \in V \text{ und } Q \in (V \cup T)^*\big)$$

Das heißt, bei jeder Regelanwendung:

- Links eine **einzelne Variable**
- Die Prämisse macht keine Aussage, was der Kontext dieser Variablen ist („kontextfrei“)
- Rechts steht etwas beliebiges

Die Chomsky-Hierarchie

Definition 8.3 (Kontextsensitive Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **kontextsensitive** gdw

$\forall (P \to Q) \in R$:

1. $\exists u, v, \alpha \in (V \cup T)^* \; \exists A \in V \; \big(P = uAv$ und $Q = u\alpha v$ mit $|\alpha| \geq 1\big)$, oder

   die Regel hat die Form $S \to \varepsilon$

2. S nicht in Q

Das heißt, bei jeder Regelanwendung:

- Eine Variable A wird in einen String α mit $|\alpha| \geq 1$ überführt
- Die Ersetzung von A durch α findet nur statt, wenn der in der Regel geforderte **Kontext** (u und v), vorhanden ist
- Das Wort wird nicht kürzer, außer bei $\varepsilon \in L$

Die Chomsky-Hierarchie

Definition 8.3 (Kontextsensitive Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **kontextsensitiv** gdw

$\forall (P \to Q) \in R$:

1. $\exists u, v, \alpha \in (V \cup T)^* \; \exists A \in V \; \big(P = uAv$ und $Q = u\alpha v$ mit $|\alpha| \geq 1\big)$, **oder**

   die Regel hat die Form $S \to \varepsilon$

2. S nicht in Q

Das heißt, bei jeder Regelanwendung:

- Eine Variable A wird in einen String α mit $|\alpha| \geq 1$ überführt
- Die Ersetzung von A durch α findet nur statt, wenn der in der Regel geforderte Kontext (u und v), vorhanden ist
- Das Wort wird nicht kürzer, außer bei $\varepsilon \in L$

Die Chomsky-Hierarchie

Definition 8.3 (Kontextsensitive Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **kontextsensitiv** gdw

$\forall (P \to Q) \in R$:

1. $\exists u, v, \alpha \in (V \cup T)^* \; \exists A \in V \; \big(P = uAv$ und $Q = u\alpha v$ mit $|\alpha| \geq 1\big)$, **oder**

   die Regel hat die Form $S \to \varepsilon$

2. S nicht in Q

Das heißt, bei jeder Regelanwendung:

- Eine Variable A wird in einen String α mit $|\alpha| \geq 1$ überführt
- Die Ersetzung von A durch α findet nur statt, wenn der in der Regel geforderte Kontext (u und v), vorhanden ist
- Das Wort wird nicht kürzer, außer bei $\varepsilon \in L$

Die Chomsky-Hierarchie

Definition 8.3 (Kontextsensitive Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **kontextsensitiv** gdw

$\forall (P \to Q) \in R$:

1. $\exists u, v, \alpha \in (V \cup T)^* \; \exists A \in V \; \big(P = uAv$ und $Q = u\alpha v$ mit $|\alpha| \geq 1\big)$, **oder**

   die Regel hat die Form $S \to \varepsilon$

2. S nicht in Q

Das heißt, bei jeder Regelanwendung:

- Eine Variable A wird in einen String α mit $|\alpha| \geq 1$ überführt
- Die Ersetzung von A durch α findet nur statt, wenn der in der Regel geforderte **Kontext** (u und v), vorhanden ist
- Das Wort wird nicht kürzer, außer bei $\varepsilon \in L$

Die Chomsky-Hierarchie

Definition 8.3 (Kontextsensitive Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **kontextsensitiv** gdw

$\forall (P \to Q) \in R$:

1. $\exists u, v, \alpha \in (V \cup T)^* \; \exists A \in V \; \big(P = uAv$ und $Q = u\alpha v$ mit $|\alpha| \geq 1\big)$, **oder**

   die Regel hat die Form $S \to \varepsilon$

2. S nicht in Q

Das heißt, bei jeder Regelanwendung:

- Eine Variable A wird in einen String α mit $|\alpha| \geq 1$ überführt
- Die Ersetzung von A durch α findet nur statt, wenn der in der Regel geforderte **Kontext** (u und v), vorhanden ist
- Das Wort wird nicht kürzer, außer bei $\varepsilon \in L$

Die Chomsky-Hierarchie

Definition 8.3 (Kontextsensitive Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **kontextsensitiv** gdw

$\forall (P \to Q) \in R$:

1. $\exists u, v, \alpha \in (V \cup T)^* \; \exists A \in V \; \big(P = uAv$ und $Q = u\alpha v$ mit $|\alpha| \geq 1\big)$, **oder**

   die Regel hat die Form $S \to \varepsilon$
2. S nicht in Q

Das heißt, bei jeder Regelanwendung:

- Eine Variable A wird in einen String α mit $|\alpha| \geq 1$ überführt
- Die Ersetzung von A durch α findet nur statt, wenn der in der Regel geforderte **Kontext** (u und v), vorhanden ist
- Das Wort wird nicht kürzer, außer bei $\varepsilon \in L$

Die Chomsky-Hierarchie

Definition 8.4 (Beschränkte Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **beschränkt** gdw

$\forall (P \to Q) \in R$:

1. $|P| \leq |Q|$, **oder**

   die Regel hat die Form $S \to \varepsilon$
2. S nicht in Q

Das heißt, bei jeder Regelanwendung:

- Die Conclusio ist mindestens so lang wie die Prämisse, außer bei $\varepsilon \in L$.
- Das Wort wird nicht kürzer, außer bei $\varepsilon \in L$

Die Chomsky-Hierarchie

Definition 8.4 (Beschränkte Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **beschränkt** gdw

$\forall (P \to Q) \in R$:

1. $|P| \leq |Q|$, **oder**

   die Regel hat die Form $S \to \varepsilon$

2. S nicht in Q

Das heißt, bei jeder Regelanwendung:

- Die Conclusio ist mindestens so lang wie die Prämisse, außer bei $\varepsilon \in L$.
- Das Wort wird nicht kürzer, außer bei $\varepsilon \in L$

Die Chomsky-Hierarchie

Definition 8.4 (Beschränkte Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **beschränkt** gdw

$\forall (P \to Q) \in R$:

1. $|P| \leq |Q|$, **oder**

   die Regel hat die Form $S \to \varepsilon$

2. S nicht in Q

Das heißt, bei jeder Regelanwendung:

- Die Conclusio ist mindestens so lang wie die Prämisse, außer bei $\varepsilon \in L$.
- Das Wort wird nicht kürzer, außer bei $\varepsilon \in L$

Die Chomsky-Hierarchie

Definition 8.4 (Beschränkte Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **beschränkt** gdw

$\forall (P \to Q) \in R$:

1. $|P| \leq |Q|$, **oder**

   die Regel hat die Form $S \to \varepsilon$

2. S nicht in Q

Das heißt, bei jeder Regelanwendung:

- Die Conclusio ist mindestens so lang wie die Prämisse, außer bei $\varepsilon \in L$.
- Das Wort wird nicht kürzer, außer bei $\varepsilon \in L$

Die Chomsky-Hierarchie

Definition 8.4 (Beschränkte Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **beschränkt** gdw

$\forall (P \to Q) \in R$:

1. $|P| \leq |Q|$, **oder**

   die Regel hat die Form $S \to \varepsilon$

2. S nicht in Q

Das heißt, bei jeder Regelanwendung:

- Die Conclusio ist mindestens so lang wie die Prämisse, außer bei $\varepsilon \in L$.
- Das Wort wird nicht kürzer, außer bei $\varepsilon \in L$

Die Chomsky-Hierarchie

Aufbauend auf den Grammatikarten kann man Sprachklassen definieren

Definition 8.5 (Sprachklassen)

Klasse	definiert als	Sprache heißt
$\mathbf{L}_3$, REG	$\{L(G) \mid G \text{ ist rechtslinear}\}$	Typ 3, **regulär**
$\mathbf{L}_2$, CFL	$\{L(G) \mid G \text{ ist kontextfrei}\}$	Typ 2, **kontextfrei**
$\mathbf{L}_1$, CSL	$\{L(G) \mid G \text{ ist kontextsensitiv}\}$	Typ 1, **kontextsensitiv**
$\mathbf{L}_1$, CSL	$\{L(G) \mid G \text{ ist beschränkt}\}$	Typ 1, **beschränkt**
$\mathbf{L}_0$, r.e.	$\{L(G) \mid G \text{ beliebig}\}$	Typ 0, **aufzählbar**
$\mathbf{L}$	$\{L \mid L \subseteq \Sigma^*\}$	**beliebige** Sprache

Die Chomsky-Hierarchie

Aufbauend auf den Grammatikarten kann man Sprachklassen definieren

Definition 8.5 (Sprachklassen)

Klasse	definiert als	Sprache heißt
$\mathbf{L_3}$, REG	$\{L(G) \mid G$ ist rechtslinear$\}$	Typ 3, **regulär**
$\mathbf{L_2}$, CFL	$\{L(G) \mid G$ ist kontextfrei$\}$	Typ 2, **kontextfrei**
$\mathbf{L_1}$, CSL	$\{L(G) \mid G$ ist kontextsensitiv$\}$	Typ 1, **kontextsensitiv**
$\mathbf{L_1}$, CSL	$\{L(G) \mid G$ ist beschränkt$\}$	Typ 1, **beschränkt**
$\mathbf{L_0}$, r.e.	$\{L(G) \mid G$ beliebig$\}$	Typ 0, **aufzählbar**
L	$\{L \mid L \subseteq \Sigma^* \}$	**beliebige** Sprache

Die Chomsky-Hierarchie

Grammatiken können kompliziert sein!

Beispiel 8.6 ( )

Grammatik $G_{abc} = (\{S, X_1, X_2\}, \{a, b, c\}, \{R_1, \ldots R_5\}, S)$ mit

$$
\begin{aligned}
R_1 &= & S &\rightarrow & abc \mid aX_1bc \\
R_2 &= & X_1b &\rightarrow & bX_1 \\
R_3 &= & X_1c &\rightarrow & X_2bcc \\
R_4 &= & bX_2 &\rightarrow & X_2b \\
R_5 &= & aX_2 &\rightarrow & aa \mid aaX_1
\end{aligned}
$$

- Ist diese Grammatik **kontextsensitiv**?
- Ist sie **beschränkt**?

Die Chomsky-Hierarchie

Grammatiken können kompliziert sein!

Beispiel 8.6 ( )

Grammatik $G_{abc} = (\{S, X_1, X_2\}, \{a, b, c\}, \{R_1, \ldots R_5\}, S)$ mit

$$
\begin{array}{rlcl}
R_1 = & S & \longrightarrow & abc \mid aX_1bc \\
R_2 = & X_1b & \longrightarrow & bX_1 \\
R_3 = & X_1c & \longrightarrow & X_2bcc \\
R_4 = & bX_2 & \longrightarrow & X_2b \\
R_5 = & aX_2 & \longrightarrow & aa \mid aaX_1
\end{array}
$$

- Ist diese Grammatik **kontextsensitiv**?
- Ist sie **beschränkt**?

Die Chomsky-Hierarchie

Grammatiken können kompliziert sein!

Beispiel 8.6 ( )

Grammatik $G_{abc} = (\{S, X_1, X_2\}, \{a, b, c\}, \{R_1, \ldots R_5\}, S)$ mit

$$\begin{aligned}
R_1 = \quad & S & \rightarrow \quad & abc \mid aX_1bc \\
R_2 = \quad & X_1b & \rightarrow \quad & bX_1 \\
R_3 = \quad & X_1c & \rightarrow \quad & X_2bcc \\
R_4 = \quad & bX_2 & \rightarrow \quad & X_2b \\
R_5 = \quad & aX_2 & \rightarrow \quad & aa \mid aaX_1
\end{aligned}$$

- Ist diese Grammatik **kontextsensitiv**?
- Ist sie **beschränkt**?

Die Chomsky-Hierarchie

Grammatiken können kompliziert sein!

Beispiel 8.6 ( )

Grammatik $G_{abc} = (\{S, X_1, X_2\}, \{a, b, c\}, \{R_1, \ldots R_5\}, S)$ mit

$$
\begin{array}{rlcl}
R_1 = & S & \rightarrow & abc \mid aX_1bc \\
R_2 = & X_1b & \rightarrow & bX_1 \\
R_3 = & X_1c & \rightarrow & X_2bcc \\
R_4 = & bX_2 & \rightarrow & X_2b \\
R_5 = & aX_2 & \rightarrow & aa \mid aaX_1
\end{array}
$$

- Ist diese Grammatik **kontextsensitiv**?
- Ist sie **beschränkt**?

Die Chomsky-Hierarchie

Grammatiken können kompliziert sein!

Beispiel 8.6 (Grammatik für $a^n b^n c^n$)

Grammatik $G_{abc} = (\{S, X_1, X_2\}, \{a, b, c\}, \{R_1, \ldots R_5\}, S)$ mit

$$
\begin{array}{llcl}
R_1 = & S & \rightarrow & abc \mid aX_1bc \\
R_2 = & X_1b & \rightarrow & bX_1 \\
R_3 = & X_1c & \rightarrow & X_2bcc \\
R_4 = & bX_2 & \rightarrow & X_2b \\
R_5 = & aX_2 & \rightarrow & aa \mid aaX_1
\end{array}
$$

- Ist diese Grammatik **kontextsensitiv**?
- Ist sie **beschränkt**?