Formale Grundlagen der Informatik 1
Kapitel 6
Kontextfreie Sprachen

Frank Heitmann
heitmann@informatik.uni-hamburg.de

27. April 2015

Motivation

Bisher hatten wir

- **Automaten**, die Wörter **akzeptieren**

Wir werden nun ein neues Modell kennenlernen:

- **Grammatiken**, die Wörter **generieren**

Informales Beispiel

Ihr kennt vielleicht schon Beispiele bei der Definition von Programmiersprachen! Hier z.B. um mögliche Identifier abzuleiten, die Ziffern und Buchstaben enthalten dürfen, aber mit einem Buchstaben beginnen müssen:

$$\begin{aligned} \langle identifier \rangle &::= \langle letter \rangle \mid \langle identifier \rangle \langle letter \rangle \mid \langle identifier \rangle \langle digit \rangle \\ \langle letter \rangle &::= A \mid B \mid C \\ \langle digit \rangle &::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \end{aligned}$$

oder kompakter:

$$\begin{aligned} I &\longrightarrow L \mid IL \mid ID \\ L &\longrightarrow a \mid b \mid c \\ D &\longrightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \end{aligned}$$

Informales Beispiel

$$\begin{aligned} I &\longrightarrow L \mid IL \mid ID \\ L &\longrightarrow a \mid b \mid c \\ D &\longrightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \end{aligned}$$

Eine mögliche Ableitung eines Identifiers (Ziel: $abc1$):

$$\begin{aligned} I &\Rightarrow ID \Rightarrow ILD \Rightarrow ILLD \Rightarrow LLLD \\ &\Rightarrow aLLD \Rightarrow abLD \Rightarrow abL1 \Rightarrow abc1 \end{aligned}$$

Hin zu Grammatiken

Dies ist genau die Idee bei Grammatiken!

- Die Großbuchstaben, die ersetzt werden können, sind die **Nonterminale**.
- Die Kleinbuchstaben, aus denen sich das abgeleitete Wort zusammensetzt, sind die **Terminale**.
- Die $I \longrightarrow IL$ usw. sind die **Regeln** oder **Produktionen**.

Wir definieren nun zunächst die Grammatik *sehr allgemein* und schränken diese später ein...

Grammatiken. Formale Definition

Definition (Grammatik)

Eine **Grammatik** ist ein Quadrupel $G = (V_N, V_T, P, S)$ mit

1. Dem endlichen Alphabet von **Nonterminalen** V_N.
2. Dem endlichen Alphabet von **Terminalen** V_T mit $V_T \cap V_N = \emptyset$. Das *Gesamtalphabet* wird mit $V := V_T \cup V_N$ bezeichnet.
3. Der endlichen Menge von **Produktionen** (oder *Regeln*) $P \subseteq (V^* \setminus V_T^*) \times V^*$.
4. Dem **Startsymbol** $S \in V_N$.

Grammatiken. Anmerkungen

Bemerkung

- Eine Regel $(u, v) \in P$ wird meist als $u \to v$ notiert.
- Mehreren Reglen $u \to v$ und $u \to w$ werden als $u \to v \mid w$ abgekürzt.
- Auf der linken Seite einer Regel steht stets mindestens ein Nonterminal!

$$P \subseteq (V^* \setminus V_T^*) \times V^*$$

Ableitung

Definition (Ableitung)

Die **einschrittige Ableitung** eines Wortes v aus einem Wort u mittels einer Produktion einer Grammatik G wird notiert als $u \underset{G}{\Longrightarrow} v$. Dabei ist die Relation $\underset{G}{\Longrightarrow} \subseteq V^* \times V^*$ für alle $u, v \in V^*$ definert durch: $u \underset{G}{\Longrightarrow} v$ gdw.

$$\exists u_1, u_2 \in V^* \exists (w_l, w_r) \in P : u = u_1 w_l u_2 \text{ und } v = u_1 w_r u_2$$

Ist der Kontext klar, wird das tief gestellte G weggelassen. Ferner bedienen wir uns wieder der reflexiven, transitiven Hülle $\overset{*}{\underset{G}{\Longrightarrow}}$ für **mehrschrittige Ableitungen**.

Grammatiken. Beispiel

Vorgehen

Eine Ableitung funktioniert also so:

- Das aktuelle Wort w betrachten
- Treten in w linke Seiten von Regeln auf?
- Falls ja, wähle eine dieser Regeln und
- ersetze die in w auftretende linke Seite der Regel
- durch die rechte Seite dieser Regel

Beispiel

Mit $S \to ASB \mid AB$, $AB \to c$, $AcB \to c$ gilt

$$S \Rightarrow ASB \Rightarrow AABB \Rightarrow AcB \Rightarrow c$$

Generierte Sprache

Definition (Generierte Sprache)

Sei $G = (V_N, V_T, P, S)$ eine Grammatik. Die von G **generierte** oder **erzeugte** Sprache ist

$$L(G) := \{w \in V_T^* \mid S \overset{*}{\underset{G}{\Longrightarrow}} w\}$$

Definition (Äquivalenz von Grammatiken)

Zwei Grammatiken G_1 und G_2 werden genau dann als **äquivalent** bezeichnet, wenn $L(G_1) = L(G_2)$ gilt.

Anmerkung

Eine Zeichenkette $u \in V^*$ mit $S \overset{*}{\underset{G}{\Longrightarrow}} u$ nennt man auch *Satzform*.

Fragen...

Wenn man zwei Regeln mit gleichen linken Seite hat, wie wird ausgewählt welche Regel benutzt wird?

1. Das darf es nicht geben!
2. Darf man frei wählen!
3. Die, die am weitesten links steht!
4. Es werden alle benutzt!

Fragen...

Kann man zwei Regeln mit gleicher rechten Seite haben?

1. Ja!
2. Nein!

Fragen...

Kann es mehrere Möglichkeiten geben, um das gleiche Wort abzuleiten?

1. Ja!
2. Nein!

Zur Nachbereitung

Zur Nachbereitung

Richtige Antworten sind:

1. 2
2. 1
3. 1

Kontextfreie Grammatiken

Die bisher eingeführten Grammatiken sind sehr mächtig.

Wir wollen sie jetzt zunächst einschränken und bei den eingeschränkten Grammatiken gucken, was man damit alles machen kann...

Kontextfreie Grammatiken. Formale Definition

Definition (Kontextfreie Grammatik (CFG))

Eine **kontextfreie Grammatik** (CFG) ist ein Quadrupel $G = (V_N, V_T, P, S)$ mit

1. Dem endlichen Alphabet von **Nonterminalen** V_N.
2. Dem endlichen Alphabet von **Terminalen** V_T mit $V_T \cap V_N = \emptyset$. Das *Gesamtalphabet* wird mit $V := V_T \cup V_N$ bezeichnet.
3. Der endlichen Menge von **Produktionen** (oder *Regeln*) $P \subseteq V_N \times V^*$.
4. Dem **Startsymbol** $S \in V_N$.

Kontextfreie Grammatiken werden auch als **Typ-2**-Grammatiken bezeichnet.

Grammatiken vs. kontextfreie Grammatiken

Anmerkung

Der Unterschied zwischen (allgemeinen) Grammatiken und kontextfreien Grammatiken liegt lediglich in der Menge der Produktionen P:

- bei allgemeinen Grammatiken:

$$P \subseteq (V^* \setminus V_T^*) \times V^*$$

- bei kontextfreien Grammatiken:

$$P \subseteq V_N \times V^*$$

Bei einer kontextfreien Grammatik ist bei einer Regel also stets genau ein Nonterminal auf der linken Seite! Daher auch der Name kontextfrei. Das Nonterminal wird unabhängig von dem Kontext, in dem es steht, ersetzt.

CFG - das λ

Definition (λ-Produktionen, λ-frei)

- Eine kontextfreie Produktion (A, λ) wird als λ-**Produktion** bezeichnet.
- Besitzt eine CFG keine λ-Produktionen, so heißt sie λ-**frei**.

Sprachfamilie CF

Definition (Sprachfamilie CF)

Ableitung und akzeptierte Sprache ist wie bei Grammatiken definiert. Die **Familie der kontextfreien Sprachen** ist dann jene Familie von Sprachen, für die es eine kontextfreie Grammatik gibt, die sie generiert. Abgekürzt wird diese Sprachfamilie mit CF.

Fragen...

Welche der folgenden Grammatiken ist *nicht* kontextfrei?

1. $S \to AB, A \to aA \mid \lambda, B \to bB \mid \lambda$
2. $S \to Ab \mid aB, A \to aA \mid a, B \to bB \mid b$
3. $S \to AB, AB \to aABb \mid ab$
4. zwei davon sind nicht kontextfrei
5. alle drei sind nicht kontextfrei
6. keine Ahnung...

Fragen...

Welche Ableitung ist mit

$$S \to AB \mid cC, A \to aAb \mid \lambda, B \to bB \mid C, C \to cC \mid \lambda$$

möglich?

1. $S \Rightarrow AB \Rightarrow AC \Rightarrow AcC \Rightarrow cC \Rightarrow c$
2. $S \Rightarrow AB \Rightarrow aAbB \Rightarrow aAbC \Rightarrow aAb \Rightarrow ab$
3. $S \Rightarrow AB \Rightarrow aAbB \Rightarrow aAbC \Rightarrow aAbc \Rightarrow abc$
4. zwei davon sind möglich
5. alle drei sind möglich
6. keine Ahnung...

Fragen...

Was gilt für die Menge der Produktionen einer kontextfreien Grammatik?

1. $P \subseteq (V^* \setminus V_T^*) \times V^*$
2. $P \subseteq V_N \times V_T^*$
3. $P \subseteq V_N \times V^*$
4. $P \subseteq (V^* \setminus V_N^*) \times V_T^*$
5. $P \subseteq V_N \times V$
6. keine Ahnung...

Zur Nachbereitung

Zur Nachbereitung

Richtige Antworten sind:

1. 3
2. 4 (nämlich 1 und 2; die 3 ist aber nur falsch, weil $aAbC \Rightarrow aAbc$ nicht in einem Schritt (sondern nur in zweien) möglich ist)
3. 3

Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

Die Grammatik G ist gegeben durch die Produktionen

$$S \to AB, A \to aA \mid \lambda, B \to bB \mid \lambda$$

Der Vollständigkeit halber:

$$\begin{aligned} V_N &:= \{S, A, B\} \\ V_T &:= \{a, b\} \end{aligned}$$

Das Startsymbol ist S.

Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

$$S \to AB, A \to aA \mid \lambda, B \to bB \mid \lambda$$

$M \subseteq L(G)$. Sei $w = a^n b^m \in M$. In G können wir nun wie folgt ableiten:

$$\begin{aligned} S &\Rightarrow AB \Rightarrow aAB \Rightarrow \ldots \Rightarrow a^n AB \Rightarrow a^n B \\ &\Rightarrow a^n bB \Rightarrow \ldots \Rightarrow a^n b^m B \Rightarrow a^n b^m \end{aligned}$$

Also gilt auch $w \in L(G)$.

Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

$$S \to AB, A \to aA \mid \lambda, B \to bB \mid \lambda$$

$L(G) \subseteq M$. An den Produktionen erkennen wir zunächst, dass $w \in \{a\}^*$ für jedes w mit $A \overset{*}{\Longrightarrow} w$ gilt, denn i-malige Anwendung von $A \to aA$ führt zu der Satzform $a^i A$. Dann kann mit $A \to \lambda$ das A gelöscht werden. Ebenso gilt $v \in \{b\}^*$ für jedes v mit $B \overset{*}{\Longrightarrow} v$.

Eine Ableitung eines Wortes $u \in L(G)$ muss nun mit $S \Rightarrow AB$ beginnen. Anschließend kann wie eben erläuert A zu a^i und B zu b^j mit $i, j \in \mathbb{N}$ abgeleitet werden, d.h. u hat die Form $a^i b^j$. Dies zeigt $u \in M$.

Beispiel 2

$$M = \{a^n b^n \mid n \in \mathbb{N}\}$$

$$S \to aSb \mid \lambda$$

$M \subseteq L(G)$. Sei $w = a^n b^n \in M$. In G können wir nun wie folgt ableiten:

$$S \Rightarrow aSb \Rightarrow \ldots \Rightarrow a^n Sb^n \Rightarrow a^n b^n$$

Also gilt auch $w \in L(G)$.

Beispiel 2

$$M = \{a^n b^n \mid n \in \mathbb{N}\}$$

$$S \to aSb \mid \lambda$$

$L(G) \subseteq M$. Sei $w \in L(G)$. Da wir nur zwei Regeln haben, ist die Argumentation recht einfach. Die einzige Möglichkeit zu einem Wort zu kommen ist die Produktion $S \to \lambda$. Vorher kann i-mal die Produktion $S \to aSb$ benutzt werden. w kann damit nur die Form $a^i b^i$ haben und damit gilt auch $w \in M$.

Zur Konstruktion von Grammatiken

Konstruktionstipps

Zwei Tipps bei der Konstruktion von Grammatiken:

1. Will man eine Zeichenkette "sequentiell" aufbauen, wie bei a^k, so kann man das Nonterminal meist an den Rand setzen $S \to aS$ (vgl. das erste Beispiel oben).
2. Will man einen "linken" und einen "rechten" Teil miteinander in Beziehung setzen, wie bei $a^n b^n$, so klappt dies meist, indem das Nonterminal nach innen wandert: $S \to aSb$ (vgl. das zweite Beispiel oben).

Fragen...

Welche Sprache generiert die Grammatik $S \to aSbc \mid \lambda$?

1. $a^n b^n c^n$
2. $a^n (bc)^n$
3. $(ab)^n c^n$
4. $(abc)^n$
5. keine davon
6. keine Ahnung ...

Fragen...

Welche Sprache generiert die Grammatik
$S \to aSb \mid C, C \to cC \mid \lambda$?

1. $a^n b^n c^n$
2. $a^n c^n b^n$
3. $a^n b^n c^m$
4. $a^n c^m b^n$
5. keine davon
6. keine Ahnung ...

Zur Nachbereitung

Zur Nachbereitung

Richtige Antworten sind:

1. 2
2. 4

Zusammenfassung

Definitionen bisher:

- Grammatik inkl.
  - Nonterminal
  - Terminal
  - Produktion
  - Startsymbol
- Ableitung
- (Generierte) Sprache, Äquivalenz

Speziell:

- Kontextfreie Grammatik / Typ-2-Grammatik
- λ-Produktionen, λ-frei
- Sprachfamilie CF

Zusammenfassung

Typische Fragestellung(en)

- Wie bei Automaten ist es oft nötig zu einer gegebenen Sprache M eine Grammatik G zu konstruieren, die M generiert, für die also $L(G) = M$ gilt. Dies ist dann wieder zu beweisen!
- Zwei Techniken zur Grammatikenkonstruktion haben wir oben kennengelernt.
- Eine weitere wichtige Fragestellung (insb. im Rahmen des einführenden Beispiels) ist zu einem Wort w zu prüfen, ob $w \in L(G)$ ist. Dies ist bei Automaten einfach (warum?), bei Grammatiken zunächst nicht so offensichtlich.

Typ-3 Grammatiken

Definition (Kontextfreie Grammatik (CFG))

Eine **kontextfreie Grammatik** $G = (V_N, V_T, P, S)$ heißt

- **linear**, falls $P \subseteq V_N \times (V_T^* \cdot V_N \cdot V_T^* \cup V_T^*)$
- **linkslinear**, falls $P \subseteq V_N \times (V_N \cdot V_T^* \cup V_T^*)$
- **rechtslinear**, falls $P \subseteq V_N \times (V_T^* \cdot V_N \cup V_T^*)$

Wir werden uns auf rechtslineare Grammatiken beschränken. Diese heißen auch **Typ-3**-Grammatiken oder **reguläre** Grammatiken.

Typische Produktionen rechtslinearer Grammatiken:

$$A \to bcX, A \to bY, A \to a$$

Grammatiken und Automaten

Satz

Eine Sprache $R \subseteq \Sigma^*$ ist regulär genau dann, wenn es eine rechtslineare Grammatik G gibt mit $L(G) = R$.

Die Beweisidee:

- Wird ein Buchstabe im Automaten gelesen, wird eine Kante (z, a, z') benutzt ("entlang gegangen").
- Man kann dies durch eine Produktion $[z] \to a[z']$ in der Grammatik "nachbauen" (wobei $[z]$ ein Nonterminal ist).
- Andersherum geht das auch!

Automat → Grammatik

Beweis.

$R \to G$. Sei $A_1 = (Z, \Sigma, K, z_0, Z_{end})$ ein DFA mit $L(A_1) = R$. Wir definieren eine rechtslineare CFG $G_R := (V_N, V_T, P, S)$ durch:

$$\begin{aligned} V_N &:= \{[z] \mid z \in Z\} \\ V_T &:= \Sigma \\ S &:= [z_0] \\ P &:= \{[z] \to a[z'] \mid (z, a, z') \in K\} \cup \\ &\quad \{[z] \to a \mid \exists z' \in Z_{end} : (z, a, z') \in K\} \cup \\ &\quad \{[z_0] \to \lambda \mid z_0 \in Z_{end}\} \end{aligned}$$

□

Automat → Grammatik

$$\begin{aligned} P &:= \{[z] \to a[z'] \mid (z, a, z') \in K\} \cup \\ &\quad \{[z] \to a \mid \exists z' \in Z_{end} : (z, a, z') \in K\} \cup \\ &\quad \{[z_0] \to \lambda \mid z_0 \in Z_{end}\} \end{aligned}$$

Gibt es eine Erfolgsrechnung in A_1 und hat diese bspw. die Kanten

$$(z_0, x_1, z_1), (z_1, x_2, z_2), \ldots, (z_{n-1}, x_n, z_n)$$

mit $z_n \in Z_{end}$, so kann man mit den Regeln

$$[z_0] \to x_1 z_1, [z_1] \to x_2 z_2, \ldots, [z_{n-1}] \to x_n$$

dieses Wort auch in der Grammatik ableiten. Die Umkehrung (zu einer Ableitung gibt es auch eine Erfolgsrechnung) gilt auch, also insgesamt $L(G_R) = L(A_1)$.

Grammatik → Automat

Beweis.

$G \to R$. Sei $G = (V_N, V_T, P, S)$ eine rechtslineare Grammatik. Wir definieren einen NFA $A_2 := (Z, \Sigma, K, Z_{start}, Z_{end})$ mit

$$\begin{aligned} Z &:= \{z_A \mid A \in V_N\} \cup \{z_\lambda\} \\ \Sigma &:= V_T \\ Z_{start} &:= \{z_S\} \\ Z_{end} &:= \{z_\lambda\} \\ K &:= \{(z_Q, u, z_R) \mid Q, R \in V_N, u \in V_T^*, Q \to uR \in P\} \cup \\ &\quad \{(z_Q, u, z_\lambda) \mid Q \in V_N, u \in V_T^*, Q \to u \in P\} \end{aligned}$$

Man kann wie eben wieder einen Zusammenhang zwischen Erfolgsrechnung und Ableitung herstellen. □

Grammatik → Automat

Hinweis

$$\begin{aligned} K &:= \{(z_Q, u, z_R) \mid Q, R \in V_N, u \in V_T^*, Q \to uR \in P\} \cup \\ &\quad \{(z_Q, u, z_\lambda) \mid Q \in V_N, u \in V_T^*, Q \to u \in P\} \end{aligned}$$

Wer sich hier daran stört, dass mehr als ein Buchstabe an der Kante steht, der kann auch Zwischenzustände für die einzelnen Buchstaben des Wortes u einführen!

Grammatiken und Automaten

Satz

Eine Sprache $C \subseteq \Sigma^*$ ist kontextfrei genau dann, wenn es einen PDA gibt, der C mit leerem Keller akzeptiert.

Beweis.

Umfangreicher als obiges, aber ähnliche Idee... □

Literaturhinweis

Interessierte finden den Beweis in [HMU] oder im Skript.

Bemerkung

Da die Akzeptanzbedingungen "mit leerem Keller" und "mit Endzustand" äquivalent sind, gilt obige Aussage auch für PDAs, die mit Endzustand akzeptieren. Man beachte aber, dass das Modell in jedem Fall *nichtdeterministisch* ist!

Zusammenfassung

Satz

Eine Sprache $R \subseteq \Sigma^*$ ist regulär genau dann, wenn es eine rechtslineare Grammatik G gibt mit $L(G) = R$.

Satz

Eine Sprache $C \subseteq \Sigma^*$ ist kontextfrei genau dann, wenn es einen PDA gibt, der C mit leerem Keller akzeptiert.

Bemerkung

Der erste Satz stellt einen Zusammenhang zwischen DFAs und rechtslinearen Grammatiken her. Der zweite Satz stellt einen Zusammenhang zwischen PDAs und kontextfreien Grammatiken her. **DFAs und rechtslineare Grammatiken sind hiernach äquivalent, ebenso wie PDAs und kontextfreie Grammatiken!**