

Theoretische Informatik

Rainer Schrader

Zentrum für Angewandte Informatik Köln

14. Juli 2009

1/40

kontextfreie Grammatiken

Gliederung

- **kontextfreie Grammatiken**
- Syntaxbäume
- Chomsky-Normalform
- Spracherkennung
- Greibach-Normalform
- Kellerautomaten

2/40

kontextfreie Grammatiken

- Aus den bisher gemachten Überlegungen ergibt sich:
 - aus der Chomsky-Hierarchie bleiben nur noch die kontextfreien Sprachen als Basis für höhere Programmiersprachen übrig
 - insbesondere können reguläre Sprachen Klammersausdrücke syntaktisch nicht überprüfen
- wir wollen uns kurz überzeugen, dass kontextfreie Sprachen zumindest Klammerkonstruktionen überprüfen können.

3/40

kontextfreie Grammatiken

Beispiele:

- sei wiederum L die Sprache über $\Sigma = \{0, 1\}$ mit gleicher Anzahl von Nullen und Einsen
- definiere eine kontextfreie Grammatik $G = (\Sigma, V, S, \mathcal{R})$ wie folgt:
 - $V = \{S\}$
 - $\mathcal{R} = \{S \rightarrow \varepsilon, S \rightarrow 0S1S, S \rightarrow 1S0S\}$
- wir behaupten: $L(G) = L$
- offensichtlich ist $L(G) \subseteq L$, da alle erzeugten Wörter die gleiche Anzahl von Nullen und Einsen enthalten

4/40

kontextfreie Grammatiken

- umgekehrt: per Induktion nehmen wir an, dass für $k < n$ alle Wörter aus L der Länge $2k$ erzeugt werden können
- sei $x \in L$ ein Wort der Länge $2n$
- wir können annehmen, dass x mit 0 anfängt
- sei y das kürzeste Präfix von x mit $y \in L$
- dann ist y von der Form $y = 0z1$ und $x = 0z1w$
- es gilt $S \rightarrow 0S1S$ und per Induktion auch $S \xrightarrow{*} z$ und $S \xrightarrow{*} w$
- somit folgt auch $S \rightarrow 0S1S \xrightarrow{*} 0z1w = x$. ✓

- auch die Palindrome lassen sich als kontextfreie Sprache auffassen:
 - $V = \{S\}$
 - $\mathcal{R} = \{S \rightarrow \varepsilon, S \rightarrow 0, S \rightarrow 1, S \rightarrow 0S0, S \rightarrow 1S1\}$.

5/40

kontextfreie Grammatiken

Schreibweise:

- anstelle der Regeln:
 - $S \rightarrow \varepsilon$
 - $S \rightarrow 01$
 - $S \rightarrow 10$
 - $S \rightarrow 0S1$
 - $S \rightarrow 1S0$
- schreiben wir auch kurz:

$$S \rightarrow \varepsilon \mid 01 \mid 10 \mid 0S1 \mid 1S0$$

6/40

kontextfreie Grammatiken

Gliederung

- kontextfreie Grammatiken
- **Syntaxbäume**
- Chomsky-Normalform
- Spracherkennung
- Greibach-Normalform
- Kellerautomaten

7/40

kontextfreie Grammatiken

- die Regeln kontextfreier Grammatiken haben auf der linken Seite nur genau eine Variable
- daher lässt sich jede Ableitung eines Wortes in G durch einen Wurzelbaum darstellen
- ein **Syntaxbaum** ist ein Wurzelbaum mit den folgenden Eigenschaften:
 - (i) in der Wurzel steht das Startsymbol S ,
 - (ii) jeder innere Knoten enthält eine Variable aus V ,
 - (iii) die Blätter enthalten ein Element aus $\Sigma \cup \{\varepsilon\}$,
 - (iv) ist ein Knoten mit A markiert und seine Söhne von links nach rechts mit x_1, x_2, \dots, x_k , so existiert eine Regel $A \rightarrow x_1x_2 \dots x_k$,
 - (v) ist ein Knoten mit ε markiert, so ist er ein Blatt und der einzige Sohn seines Vaters.

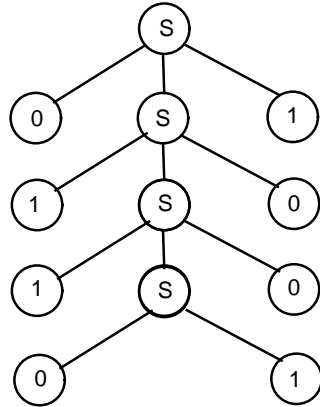
8/40

kontextfreie Grammatiken

im Beispiel der Kammersetzung liefert die Ableitung

$$S \rightarrow 0S1 \rightarrow 01S01 \rightarrow 011S001 \rightarrow 01101001$$

den Syntaxbaum



9/40

kontextfreie Grammatiken

- zu jeder Ableitung existiert ein eindeutiger Baum
- das Ergebnis der Ableitung steht in den Blättern des Baums, wenn wir sie von links nach rechts lesen
- umgekehrt kann jedoch ein Syntaxbaum mehr als eine Ableitung ergeben, wenn ein Wort mehr als nur eine Variable enthält
- wegen der Kontextfreiheit ist die Reihenfolge, in der Variablen ersetzt werden, ohne Einfluss auf das Ergebnis
- als Standardableitung eines Wortes werden wir entweder die **Linksableitung** oder die **Rechtsableitung** betrachten
- dabei wird in jedem String jeweils die am weitesten links (rechts) stehende Variable ersetzt.

10/40

kontextfreie Grammatiken

Beispiel

- sei G gegeben durch die Regeln $S \rightarrow 0AS \mid 0, A \rightarrow S1A \mid SS \mid 10$
- sei $x = 001100$
- die zugehörige Linksableitung ist

$$S \rightarrow 0AS \rightarrow 0S1AS \rightarrow 001AS \rightarrow 00110S \rightarrow 001100$$

- die Rechtsableitung ist

$$S \rightarrow 0AS \rightarrow 0A0 \rightarrow 0S1A0 \rightarrow 0S1100 \rightarrow 001100.$$

11/40

kontextfreie Grammatiken

Gliederung

- kontextfreie Grammatiken
- Syntaxbäume
- **Chomsky-Normalform**
- Spracherkennung
- Greibach-Normalform
- Kellerautomaten

12/40

kontextfreie Grammatiken

- wir betrachten im weiteren nur kontextfreie Sprachen, die ε nicht enthalten
- wir können ε nachträglich wieder hinzufügen:
 - erzeuge ein neues Startsymbol S'
 - füge die Regeln $S' \rightarrow \varepsilon \mid S$ hinzu

13/40

kontextfreie Grammatiken

Zur einfacheren Behandlung des Spracherkennungsproblems wollen wir die Grammatiken formal vereinfachen

- eine kontextfreie Grammatik ist in **Chomsky-Normalform**, wenn alle Regeln von der Form sind:

$$A \rightarrow BC \mid a$$

- wobei $A, B, C \in V$ und $a \in \Sigma$
- (offensichtlich kann durch die Normalform ε nicht erzeugt werden)
- wir wollen zeigen, dass jede kontextfreie Grammatik in Normalform gebracht werden kann
- dazu vereinfachen wir die Grammatik schrittweise

14/40

kontextfreie Grammatiken

1. auf der rechten Seite der Regeln sollen nur Variablen oder ein Buchstabe stehen:
 - für jedes $a \in \Sigma$ erzeuge eine neue Variable Y_a
 - ersetze auf den rechten Seiten der Regeln a durch Y_a
 - füge die Regeln $Y_a \rightarrow a$ hinzu
- dadurch wird die erzeugte Sprache nicht verändert
- für jedes $a \in \Sigma$ fügen wir eine Regel mit zwei Symbolen hinzu

15/40

kontextfreie Grammatiken

2. ersetze Regeln mit langen rechten Seiten:

- betrachte Regeln der Form $A \rightarrow B_1 B_2 \dots B_m$ mit $m \geq 3$
- erzeuge $m - 2$ neue Variablen C_1, \dots, C_{m-2}
- füge die neuen Regeln hinzu:

$$A \rightarrow B_1 C_1, C_1 \rightarrow B_2 C_2, \dots, C_{m-2} \rightarrow B_{m-1} B_m.$$

- dadurch wird die erzeugte Sprache nicht verändert
- jede Regel mit $m + 1$ Variablen wird durch $m - 1$ Regeln mit jeweils drei Variablen ersetzt

16/40

kontextfreie Grammatiken

3. ersetze Regeln der Form $A \rightarrow \varepsilon$:

- dazu berechnen wir in einem ersten Schritt die Menge V' aller Variablen A mit $A \xrightarrow{*} \varepsilon$:

$$V' = \emptyset, U = \emptyset$$

für jedes $A \in V$ **do**
 ist $A \rightarrow \varepsilon$ eine Regel **do**
 $V' = V' \cup \{A\}, U = U \cup \{A\}$
end do
end for

while $U \neq \emptyset$
 wähle $A \in U$
 entferne A aus U
 ersetze in jeder Regel A durch ε
 entsteht dadurch eine Regel $B \rightarrow \varepsilon$ **do**
 $V' = V' \cup \{B\}, U = U \cup \{B\}$
end do
end while

17/40

kontextfreie Grammatiken

- per Induktion über die Länge der Ableitung lässt sich zeigen:

- V' enthält alle Variablen A mit $A \xrightarrow{*} \varepsilon$

- entferne alle Regeln der Form $A \rightarrow \varepsilon$
- ersetze eine Regel $A \rightarrow BC$ durch:

$$A \rightarrow C, \text{ falls } B \in V'$$

$$A \rightarrow B, \text{ falls } C \in V'$$

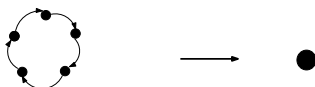
- man überlegt sich, dass dadurch:
 - die erzeugte Sprache nicht verändert wird,
 - höchstens jede Regel $A \rightarrow BC$ durch die Regeln $A \rightarrow B, A \rightarrow C$ ersetzt wird

18/40

kontextfreie Grammatiken

3. ersetze Regeln der Form $A \rightarrow B$:

- bilde einen gerichteten Graphen G mit Knotenmenge V und Kanten $A \rightarrow B$
- mittels Tiefensuche suche in G nach einem Kreis $A_1 \rightarrow A_2 \rightarrow \dots \rightarrow A_i \rightarrow A_1$
- schrumpfe den Kreis zum Knoten A_1



- wiederhole so lange, bis G kreisfrei ist
- danach ist G ein Netzwerk (gerichtet, kreisfrei)

19/40

kontextfreie Grammatiken

- sei A eine Senke in G
- für jede Regel $A \rightarrow u$ mit $u = a \in \Sigma$ oder $u = BC$
- und für jeden gerichteten Weg B_1, B_2, \dots, B_k, A in G
- ersetze die Regeln $B_i \rightarrow B_{i+1}$ durch $B_i \rightarrow u$ für $i = 1, \dots, k - 1$



- die nach diesen Operationen verbliebenen Kanten von G können entfernt werden
- dann gilt wieder: die Sprache bleibt unter den Ersetzungen unverändert
- danach ist die Grammatik in Chomsky-Normalform
- damit haben wir gezeigt:

20/40

kontextfreie Grammatiken

Satz

Jede kontextfreie Sprache L mit $\varepsilon \notin L$ wird von einer Grammatik in Chomsky-Normalform generiert. \square

das Verfahren ist im folgenden Sinne effizient:

- sei die **Größe** $s(G)$ einer Grammatik G die Gesamtsumme der Anzahl der in den Regeln enthaltenen Variablen und Buchstaben
- dann liefert eine Untersuchung der Laufzeit das folgende Ergebnis:

Korollar

Jede kontextfreie Grammatik G der Größe $s(G)$ kann in $\mathcal{O}(|V|s(G))$ Schritten in eine kontextfreie Grammatik G' in Chomsky-Normalform der Größe $\mathcal{O}(|V|s(G))$ mit $L(G') = L(G)$ umgewandelt werden. \square

21/40

kontextfreie Grammatiken

Als weitere Folgerung ergibt sich die Verifizierung der Chomsky-Hierarchie:

Korollar

Für die von Grammatiken vom Typ i erzeugten Sprachklassen gilt:

$$\mathcal{L}_3 \subseteq \mathcal{L}_2 \subseteq \mathcal{L}_1 \subseteq \mathcal{L}_0.$$

Beweis:

- wir können annehmen, dass $L \in \mathcal{L}_2$ in Chomsky-Normalform vorliegt
- falls $\varepsilon \notin L$, so folgt die Behauptung unmittelbar
- andernfalls fügen wir eine neue Startvariable S' und die Regeln $S' \rightarrow \varepsilon \mid S$ hinzu
- und ersetzen in den rechten Seiten der anderen Regeln S durch S'
- damit folgt auch die noch fehlende Inklusion $\mathcal{L}_2 \subseteq \mathcal{L}_1$. \square

Ferner lässt sich zeigen, dass alle Inklusionen echt sind.

22/40

kontextfreie Grammatiken

Gliederung

- kontextfreie Grammatiken
- Syntaxbäume
- Chomsky-Normalform
- **Spracherkennung**
- Greibach-Normalform
- Kellerautomaten

23/40

kontextfreie Grammatiken

- reguläre Sprachen sind für die Zwecke des Compilerbaus zu speziell (können keine Klammerausdrücke überprüfen)
- die Sprachen in \mathcal{L}_2 sind bereits zu allgemein (wir können keine Syntaxanalyse durchführen)
- der folgende Satz besagt, dass dieses Problem für kontextfreie Sprachen effizient (wenn auch noch nicht schnell) lösbar ist

Satz

Sei G eine kontextfreie Grammatik in Chomsky-Normalform mit Alphabet Σ und Regeln \mathcal{R} . Sei $x \in \Sigma^*$ mit $|x| = n$. Dann kann in Zeit $\mathcal{O}(|\mathcal{R}|n^3)$ entschieden werden, ob $x \in L(G)$.

24/40

kontextfreie Grammatiken

Beweis:

- der Beweis beruht auf dynamischer Programmierung
- sei $x = x_1 \dots x_n$
- für $1 \leq i \leq j \leq n$ sei $x_{ij} = x_i \dots x_j$
- wir erzeugen rekursiv die Menge V_{ij} aller Variablen A mit $A \xrightarrow{*} x_{ij}$
- dann ist $x \in L(G) \iff S \in V_{1n}$.

25/40

kontextfreie Grammatiken

- sei $x_{ij} = x_i \dots x_j$ und $V_{ij} = \{A : A \xrightarrow{*} x_{ij}\}$
- wir bauen die Mengen V_{ij} induktiv über $l = j - i$ auf
- und verwenden die Tatsache, dass G in Chomsky-Normalform vorliegt
- für $l = 0$ bestehen die V_{ij} aus genau den Variablen A , für die es eine Regel $A \rightarrow x_i$ gibt
- allgemein gibt es eine Ableitung $A \xrightarrow{*} x_{ij}$ genau dann, wenn gilt:
 - es gibt eine Regel $A \rightarrow BC$
 - und ein $k \in \{i, \dots, j-1\}$ mit
 - $B \xrightarrow{*} x_{ik}$ und $C \xrightarrow{*} x_{k+1,j}$
- damit testen wir für jedes Paar i, j höchstens n Zwischenpunkte k und jeweils $|\mathcal{R}|$ Regeln
- woraus sich die Laufzeit von $\mathcal{O}(n^3 |\mathcal{R}|)$ ergibt. □

26/40

kontextfreie Grammatiken

- für eine konkrete Grammatik (und damit vorgegebener Programmiersprache) ist $|\mathcal{R}|$ eine Konstante
- somit kann das obige Problem in $\mathcal{O}(n^3)$ Schritten gelöst werden
- dies ist zwar polynomiell, aber für praktische Zwecke noch zu langsam
- wir werden uns am Schluss uns kurz mit einer Teilklasse von \mathcal{L}_2 beschäftigen, die immer noch mächtig genug ist, jedoch lineare Algorithmen für den Test „Ist $x \in L$?“ erlauben
- wie schon bei den anderen Sprachklassen werden wir Automaten bestimmen, so dass die von ihnen akzeptierten Sprachen die Klasse \mathcal{L}_2 bilden
- zur Vorbereitung führen wir eine zweite Normalform für kontextfreie Sprachen ein

27/40

kontextfreie Grammatiken

Gliederung

- kontextfreie Grammatiken
- Syntaxbäume
- Chomsky-Normalform
- Spracherkennung
- **Greibach-Normalform**
- Kellerautomaten

28/40

kontextfreie Grammatiken

- wir beschränken uns wieder auf Sprachen, die das leere Wort nicht enthalten
- eine kontextfreie Grammatik ist in **Greibach-Normalform**, wenn alle Ableitungsregeln von der Form sind:

$$A \rightarrow aB_1B_2 \dots B_k$$

- mit $k \geq 0$ und $A, B_1, \dots, B_k \in V$ und $a \in \Sigma$
- d.h. bei jedem Ableitungsschritt wird genau ein Buchstabe des Alphabets erzeugt

29/40

kontextfreie Grammatiken

Es gilt (ohne Beweis):

Satz

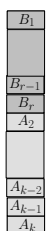
Zu einer kontextfreien Grammatik G kann in $\mathcal{O}(|\mathcal{R}|^3)$ Schritten eine kontextfreie Grammatik G' mit $|\mathcal{R}|^3$ Regeln in Greibach-Normalform und $L(G') = L(G)$ konstruiert werden. \square

- wir schon bei den anderen Sprachklassen wollen wir versuchen, Maschinen anzugeben, die die kontextfreien Sprachen erkennen
- dafür in Frage kommen nur Maschinen, die
 - mächtiger sind als deterministische Automaten,
 - aber schwächer als linear-bandbeschränkte Turingmaschinen

30/40

kontextfreie Grammatiken

- sei G eine kontextfreie Grammatik in Greibach-Normalform
- sei $x = x_1 \dots x_n \in L(G)$
- eine Linksableitung von x startet mit einer Regel $S \rightarrow x_1 A_1 \dots A_k$
- die nächste Ableitung ist von der Form $A_1 \rightarrow x_2 B_1 \dots B_r$
- wir können uns also vorstellen, dass die Variablen durch einen Stack (Keller) verwaltet werden:



- die erste Ableitung schiebt die Variablen in der Reihenfolge A_k, A_{k-1}, \dots, A_1 auf den Stack
- die zweite Ableitung entfernt A_1 vom Stack und schiebt die Variablen in der Reihenfolge B_r, B_{r-1}, \dots, B_1 auf den Stack
- ...

31/40

kontextfreie Grammatiken

- es ist daher nicht überraschend, dass wir auf die folgende Weise alle Wörter von $L(G)$ erzeugen können:
 - sei $x = x_1 \dots x_i$ bereits erzeugt und $A_1 \dots A_k$ der Inhalt des Stacks
 - entferne A_1 aus dem Stack
 - wähle nichtdeterministisch eine Regel $A_1 \rightarrow aB_1 \dots B_r$ aus
 - erweitere x um a und schiebe die Variablen in der Reihenfolge B_r, B_{r-1}, \dots, B_1 auf den Stack
 - ...

32/40

kontextfreie Grammatiken

Gliederung

- kontextfreie Grammatiken
- Syntaxbäume
- Chomsky-Normalform
- Spracherkennung
- Greibach-Normalform
- **Kellerautomaten**

33 / 40

kontextfreie Grammatiken

- durch Umkehrung dieses Prinzips können wir eine Maschine angeben, die die $x \in L(G)$ akzeptiert
- sie liegt zwischen einem NFA und einer linear-bandbeschränkten NDTM
- sie liest den Input wie ein DFA von links nach rechts
- ihr Arbeitsband ist wie ein Stack organisiert

34 / 40

kontextfreie Grammatiken

nichtdeterministischer Kellerautomat

initialisiere den Stack mit S

while Stack $\neq \emptyset$ und weiteres Eingabezeichen vorhanden

 lies nächstes Eingabezeichen x

$A = \text{pop}(\text{Stack})$

 wähle nichtdeterministisch eine Regel $A \rightarrow xB_1B_2 \dots B_k$

for $i = k$ **down to** 1 **do**

 push(B_i , Stack)

end do

end while

if Stack = \emptyset und kein weiteres Eingabezeichen vorhanden

 akzeptiere

else verwirf

35 / 40

kontextfreie Grammatiken

- ein solcher **nichtdeterministischer Kellerautomat** akzeptiert die $x \in L(G)$
- mit etwas Aufwand lässt sich auch zeigen:

Satz

Eine Sprache L ist genau dann kontextfrei, wenn sie von einem nichtdeterministischen Kellerautomaten akzeptiert wird. □

36 / 40

kontextfreie Grammatiken

Fazit:

- die kontextfreien Sprachen sind äquivalent zu den Sprachen, die von nichtdeterministischen Kellerautomaten akzeptiert werden
- sie können in $\mathcal{O}(n^3)$ Schritten entschieden werden
- damit haben wir noch keinen schnellen Algorithmus zur Spracherkennung
- zur Verdeutlichung der Schwierigkeit ein Beispiel:

37 / 40

kontextfreie Grammatiken

Beispiel:

- \mathcal{R} bestehe aus den Regeln: $S \rightarrow aA \mid aB$, $A \rightarrow a$, $B \rightarrow b$
- wir lesen die Eingabe $ab \in L(G)$ einmal von links nach rechts
- wenn wir a lesen, wissen wir nicht, welche Regel angewandt wurde
- erst nach dem Lesen von b können wir dies entscheiden
- wir benötigen somit eine „Vorausschau“

38 / 40

kontextfreie Grammatiken

- **LR(1)** besteht aus den kontextfreien Sprachen, für die
 - eine Eingabe einmal von links nach rechts gelesen wird, und
 - mit einer Vorausschau von einem Buchstaben
 - eine Rechtsableitung erzeugt werden kann
- für die Sprachen in **LR(1)** gilt:
- sie bilden eine echte Teilmenge der kontextfreien Sprachen
- sie stimmen mit den Sprachen überein, die von einem **deterministischen Kellerautomaten** erkannt werden
- für sie existiert ein linearer Spracherkennungsalgorithmus
- sie bilden die Basis für die meisten Programmiersprachen

39 / 40

kontextfreie Grammatiken

- das war's.
- Danke für
 - die Aufmerksamkeit
 - die lebhaftige Mitarbeit
 - und für die Korrektur der Fehler.
- **Viel Erfolg in der Klausur.**

40 / 40