
Grundlagen der theoretischen Informatik

Kurt Sieber

Fachbereich Mathematik/Theoretische Informatik
Universität Siegen

Vorlesung vom 22.11.2004 (Stand: 22.11.2004)

Kontextfreie Sprachen

- Reguläre Sprachen haben eine sehr einfache Struktur.

Deshalb kann man mit ihnen nur die kleinsten (d.h. einfachsten) syntaktischen Einheiten einer Programmiersprache erfassen,

z.B. Schlüsselwörter, Zahldarstellungen, Bezeichner, Kommentare.

- Wie beschreibt man die komplizierteren syntaktischen Strukturen, z.B. arithmetische und boolesche Ausdrücke, Anweisungen, Deklarationen?

Am besten durch eine *Konstruktionsvorschrift*,

d.h. aus mathematischer Sicht: durch eine *induktive Definition*.

Kontextfreie Sprachen

Beispiel: Vollständig geklammerte arithmetische Ausdrücke

Konstruktionsvorschrift

1. Jede Dezimalzahl ist ein arithmetischer Ausdruck.
2. Jeder Bezeichner ist ein arithmetischer Ausdruck.
3. Wenn e ein arithmetischer Ausdruck ist, dann ist auch $(-e)$ ein arithmetischer Ausdruck.
4. Wenn e_1, e_2 arithmetische Ausdrücke sind, dann ist auch $(e_1 + e_2)$ ein arithmetischer Ausdruck.
5. Wenn e_1, e_2 arithmetische Ausdrücke sind, dann ist auch $(e_1 - e_2)$ ein arithmetischer Ausdruck.
6. Wenn e_1, e_2 arithmetische Ausdrücke sind, dann ist auch $(e_1 * e_2)$ ein arithmetischer Ausdruck.

Kontextfreie Sprachen

Mathematische Formulierung

Die Sprache L der vollständig geklammerten arithmetischen Ausdrücke ist *induktiv definiert* durch:

1. Jede Dezimalzahl liegt in L .
2. Jeder Bezeichner liegt in L .
3. Wenn $e \in L$, dann ist auch $(-e) \in L$.
4. Wenn $e_1, e_2 \in L$, dann ist auch $(e_1 + e_2) \in L$.
5. Wenn $e_1, e_2 \in L$, dann ist auch $(e_1 - e_2) \in L$.
6. Wenn $e_1, e_2 \in L$, dann ist auch $(e_1 * e_2) \in L$.

Was bedeutet '*induktiv definiert*'? Es bedeutet, dass L die *kleinste* Menge mit den Eigenschaften 1. bis 6. ist (eine andere Menge mit diesen Eigenschaften ist z.B. Σ^*).

Kontextfreie Sprachen

Die mathematische Formulierung genügt uns nicht (so wie uns die Mengenausdrücke nicht genügten). Wir brauchen eine *formale Schreibweise* (wie die regulären Ausdrücke).

Definition 2.1 Eine *kontextfreie Grammatik* (kurz: *KFG*) ist ein 4-Tupel $G = (\Sigma, N, S, P)$, wobei gilt:

- Σ und N sind Alphabete mit $\Sigma \cap N = \emptyset$. Die Zeichen aus Σ heißen *Terminalzeichen*, die aus N heißen *Nichtterminalzeichen* von G .
- $S \in N$. S heißt *Startzeichen* von G .
- $P \subseteq N \times (\Sigma \cup N)^*$. Die Elemente von P heißen *Regeln* oder *Produktionen* von G .

Eine Produktion ist also ein *Paar* (A, u) , wobei A ein Nichtterminalzeichen ist und u ein Wort, das sowohl Terminal- als auch Nichtterminalzeichen enthalten darf. Statt (A, u) schreibt man $A \rightarrow u$.

Kontextfreie Sprachen

Konvention:

Als Nichtterminalzeichen benutzen wir Großbuchstaben.

Beispiel:

$G = (\Sigma, N, S, P)$ mit

- $\Sigma = \{0, 1, x, y, -, +, *, (,)\}$
- $N = \{E\}$
- $S = E$
- $P = \{E \rightarrow 0, E \rightarrow 1, E \rightarrow x, E \rightarrow y, E \rightarrow (-E),$
 $E \rightarrow (E + E), E \rightarrow (E - E), E \rightarrow (E * E)\}$

ist eine KFG für vollständig geklammerte arithmetische Ausdrücke (in denen nur die Zahlen 0, 1 und nur die Bezeichner x, y vorkommen).

Kontextfreie Sprachen

Kurzschreibweise:

Man schreibt

$$A \rightarrow u_1 \mid \dots \mid u_n$$

als Abkürzung für eine *Menge* von Produktionen

$$A \rightarrow u_1, \dots, A \rightarrow u_n$$

mit dem gleichen Nichtterminalzeichen auf der linken Seite.

Im Beispiel kann man also *alle* Produktionen zusammenfassen zu

$$E \rightarrow 0 \mid 1 \mid x \mid y \mid (-E) \mid (E + E) \mid (E - E) \mid (E * E)$$

wobei man “|” als “oder” liest.

Kontextfreie Sprachen

Wie ist eine KFG zu verstehen, d.h. welche Sprache beschreibt sie?

- Entweder als *Konstruktionsvorschrift* für eine Sprache:
Dann dienen die Produktionen dazu, die Wörter der Sprache *abzuleiten* oder zu *erzeugen*.
- Oder als *induktive Definition* einer Sprache:
Dann liest man die Produktionen als *Regeln*, die die Sprache erfüllen muss.

Beide Auffassungen sind äquivalent zueinander, üblich ist aber die Formulierung als Konstruktionsvorschrift.

Kontextfreie Sprachen

Definition 2.2 Sei $G = (\Sigma, N, S, P)$ eine KFG.

Auf der Menge $(\Sigma \cup N)^*$ definieren wir eine Relation \Rightarrow_G (oder kürzer: \Rightarrow) durch:

$$u \Rightarrow_G v \Leftrightarrow \begin{array}{l} \text{es gibt eine Produktion } A \rightarrow y \text{ in } P \\ \text{und Wörter } x, z \in (\Sigma \cup N)^* \\ \text{so dass } u = xAz \text{ und } v = xyz \end{array}$$

Man bezeichnet $u \Rightarrow_G v$ als einen **Ableitungsschritt** (in G).

In Worten: Ein Ableitungsschritt $u \Rightarrow_G v$ besteht darin, ein Vorkommen eines Nichtterminalzeichens A im Wort u durch die rechte Seite y einer Produktion $A \rightarrow y$ zu ersetzen.

Kontextfreie Sprachen

Mit \xrightarrow{n}_G ($n \geq 0$), $\xrightarrow{+}_G$ und $\xrightarrow{*}_G$ bezeichnen wir wieder die n -te Potenz, den transitiven Abschluss und den reflexiven, transitiven Abschluss der Relation \Rightarrow_G .

Definition 2.3 Sei G eine KFG.

1. v heißt **ableitbar** aus u (in G), wenn $u \xrightarrow{*}_G v$, d.h. wenn eine Folge $u = w_0 \Rightarrow_G \dots \Rightarrow_G w_n = v$ ($n \geq 0$) von Ableitungsschritten in G existiert.

Eine solche Folge bezeichnet man als **Ableitung** von v aus u .

2. Die von G **erzeugte** Sprache $L(G)$ ist definiert durch

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

$L(G)$ besteht also aus allen Wörtern über dem Terminalalphabet Σ , die aus dem Startzeichen S ableitbar sind.

Kontextfreie Sprachen

Definition 2.4 Eine Sprache $L \subseteq \Sigma^*$ heißt **kontextfrei**, wenn es eine kontextfreie Grammatik G gibt mit $L = L(G)$.

Beispiel:

Sei G die oben definierte KFG für arithmetische Ausdrücke.

Dann gilt z.B.

$E \Rightarrow (E + E) \Rightarrow ((E * E) + E) \Rightarrow ((x * E) + E) \Rightarrow ((x * y) + E) \Rightarrow ((x * y) + 1)$
also $((x * y) + 1) \in L(G)$.

Die Sprache $L(G)$ ist (per Definition) kontextfrei.

Aber ist $L(G)$ die gewünschte Sprache L der vollständig geklammerten arithmetischen Ausdrücke?

Um diese Frage zu beantworten, bräuchte man eine Definition für L , die unabhängig von der Grammatik ist. Die ist schwer zu finden!

Also stellen wir uns auf den Standpunkt, dass L durch die Grammatik G **definiert** ist. Dann ist nichts mehr zu beweisen.

Kontextfreie Sprachen

Sei G wie oben.

Neben der bereits genannten Ableitung

$$E \Rightarrow (E + E) \Rightarrow ((E * E) + E) \Rightarrow ((x * E) + E) \Rightarrow ((x * y) + E) \Rightarrow ((x * y) + 1)$$

gibt es andere Ableitungen für das gleiche Wort, z.B.

$$E \Rightarrow (E + E) \Rightarrow (E + 1) \Rightarrow ((E * E) + 1) \Rightarrow ((E * y) + 1) \Rightarrow ((x * y) + 1)$$

In beiden Ableitungen werden die 'gleichen' Ableitungsschritte in unterschiedlicher Reihenfolge benutzt.

Die Reihenfolge der Ableitungsschritte spielt keine Rolle, weil die Ableitungsschritte in einer kontextfreien Grammatik sehr einfach aussehen: Es wird stets ein Nichtterminalzeichen A durch ein Wort y ersetzt. Der *Kontext* des Zeichens A , d.h. der Text vor und hinter A , spielt dabei keine Rolle. Er beeinflusst den Ableitungsschritt nicht, und er wird durch den Ableitungsschritt nicht verändert (daher die Bezeichnung '*kontextfrei*'). Also ist es unerheblich, ob man zuerst A oder zuerst ein Nichtterminalzeichen im 'Kontext' von A ersetzt.

Kontextfreie Sprachen

Diese Überlegungen kann man präzisieren,

- entweder indem man eine spezielle Reihenfolge für die Ableitungsschritte *vorschreibt*, und dann zeigt, dass jede Ableitung so umgeformt werden kann, dass sie der Vorschrift genügt,
- oder indem man eine Schreibweise benutzt, die von der Reihenfolge der Ableitungsschritte *abstrahiert*.

Definition 2.5 *Ein Ableitungsschritt $u \Rightarrow v$ heißt Linksableitungsschritt, wenn es Wörter $x \in \Sigma^*$, $z \in (N \cup \Sigma)^*$ und eine Produktion $A \rightarrow y$ in G gibt mit $u = xAz$ und $v = xyz$. Eine Linksableitung ist eine Ableitung, die nur aus Linksableitungsschritten besteht.*

Eine Linksableitung ist also eine Ableitung, bei der stets das erste, d.h. das am weitesten links stehende Nichtterminalzeichen ersetzt wird.

Kontextfreie Sprachen

Ein Beispiel für eine Linksableitung haben wir schon gesehen:

$$E \Rightarrow (E + E) \Rightarrow ((E * E) + E) \Rightarrow ((x * E) + E) \Rightarrow ((x * y) + E) \Rightarrow ((x * y) + 1)$$

Das folgende Lemma besagt, dass Linksableitungen ausreichen.

Lemma 2.6 *$u \xRightarrow{*}_G v$ gilt genau dann, wenn es eine Linksableitung von v aus u in G gibt.*

Beweisidee:

Man kann die Schritte einer beliebigen Ableitung so umordnen, dass eine Linksableitung entsteht. Die Details sind mühsam! \square

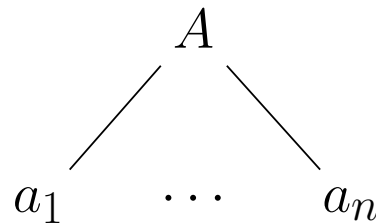
Man kann Lemma 2.6 so interpretieren:

Wenn wir beliebige Ableitungen betrachten, so gibt es (fast in jeder Grammatik) viele Ableitungen, die sich nur 'unwesentlich' unterscheiden, nämlich nur in der Reihenfolge der Ersetzungen. Betrachten wir nur Linksableitungen, so entfallen diese unwesentlichen Unterschiede, weil die Reihenfolge der Ableitungsschritte fest vorgeschrieben ist.

Kontextfreie Sprachen

Ein alternativer Ansatz besteht darin, von der Reihenfolge der Ableitungsschritte zu abstrahieren, indem man *Ableitungsbäume* betrachtet.

Definition 2.7 Ein *Ableitungsbaum* (oder *Syntaxbaum*) für die Grammatik $G = (\Sigma, N, S, P)$ ist ein Baum, dessen Knoten mit Zeichen aus $\Sigma \cup N$ markiert sind, und zwar so, dass jeder innere Knoten die Form

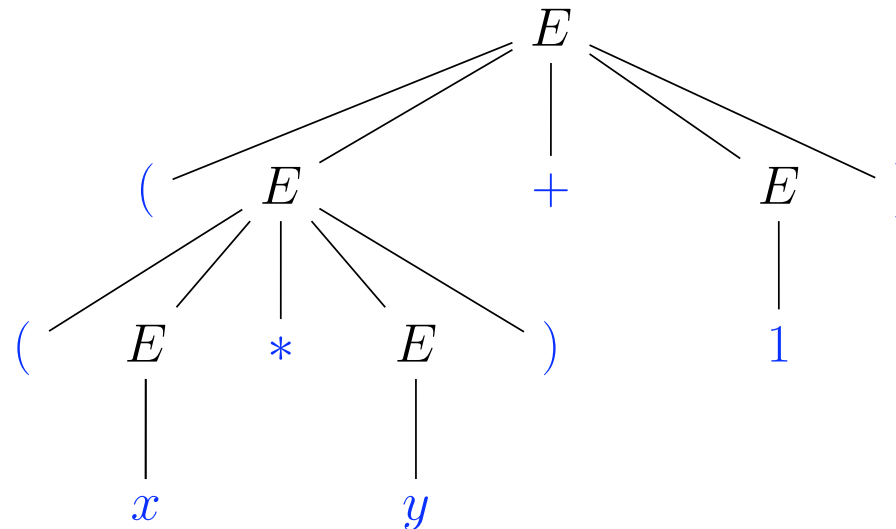


hat, wobei $A \rightarrow a_1 \dots a_n$ eine Produktion in P ist.

Kontextfreie Sprachen

Beispiel:

Ein Ableitungsbaum für unsere Grammatik G ist



An diesem Baum sieht man, dass $((x * y) + 1)$ aus E ableitbar ist (ohne dass die Reihenfolge der Ableitungsschritte festgelegt wird).

Kontextfreie Sprachen

Lemma 2.8

- *Zu jeder Ableitung $A \xRightarrow{*} u$ existiert ein ‘entsprechender’ Ableitungsbaum mit Wurzel A und Blattwort u .*
- *Umgekehrt existiert zu jedem Ableitungsbaum mit Wurzel A und Blattwort u genau eine Linksableitung $A \xRightarrow{*} u$.*

Ein Ableitungsbaum steht also für eine ganze *Menge* von Ableitungen, die sich nur in der Reihenfolge der Ableitungsschritte unterscheiden. Unter all diesen Ableitungen gibt es genau eine Linksableitung, die man erhält, indem man den Baum ‘von links nach rechts’ durchläuft.

Ableitungsbäume sind einerseits abstrakter als Ableitungen (weil sie von der Reihenfolge der Ableitungsschritte abstrahieren), andererseits sind sie anschaulicher, weil sie die *Struktur* des abgeleiteten Wortes erkennen lassen.

Kontextfreie Sprachen

In der Praxis (Compilerbau) ist es wichtig, dass jedes Wort eine eindeutige Struktur besitzt, denn ein Compiler soll ja nicht nur testen, ob eine eingegebene Zeichenreihe ein arithmetischer Ausdruck ist, sondern er soll diesen Ausdruck auch weiterverarbeiten, d.h. letzten Endes in Maschinencode übersetzen.

Definition 2.9

- Eine kontextfreie Grammatik heißt **eindeutig**, wenn für jedes Wort $w \in L(G)$ genau ein Ableitungsbaum mit Wurzel S und Blattwort w existiert (oder äquivalent dazu: genau eine Linksableitung $S \xRightarrow{*} w$). Andernfalls heißt sie **mehrdeutig**.
- Eine kontextfreie Sprache L heißt **inhärent mehrdeutig**, wenn jede kontextfreie Grammatik G mit $L = L(G)$ mehrdeutig ist.

Kontextfreie Sprachen

Beispiel:

Unsere Grammatik G für vollständig geklammerte arithmetische Ausdrücke ist eindeutig.

Intuitive Begründung:

Durch die vollständige Klammerung ist die Struktur jedes Ausdrucks eindeutig festgelegt.

Beweisskizze:

Man beweist zunächst (durch eine einfache Induktion über die Länge der Ableitung von e), dass

- jedes Wort $e \in L(G)$ genauso viele öffnende wie schließende Klammern hat,
- jedes echte nichtleere Präfix eines Wortes $e \in L(G)$ *mehr* öffnende als schließende Klammern hat.

Es folgt sofort, dass kein Wort $e \in L(G)$ echtes Präfix eines anderen Wortes $e' \in L(G)$ sein kann.

Kontextfreie Sprachen

Damit kann man nun zeigen, dass jeder Ausdruck $e \in L(G)$ eine eindeutige Struktur hat: Nehmen wir z.B. an, dass ein Ausdruck $e \in L(G)$ sich sowohl in der Form $(e_1 + e_2)$ als auch in der Form $(e'_1 * e'_2)$ darstellen lässt (mit $e_1, e_2, e'_1, e'_2 \in L(G)$).

Dann ist einer der Ausdrücke e_1, e'_1 kürzer als der andere, und müsste deshalb echtes nichtleeres Präfix des anderen sein (weil die beiden Zeichenreihen $(e_1$ und $(e'_1$ Präfixe von e sind). Das ist aber nicht möglich. \square

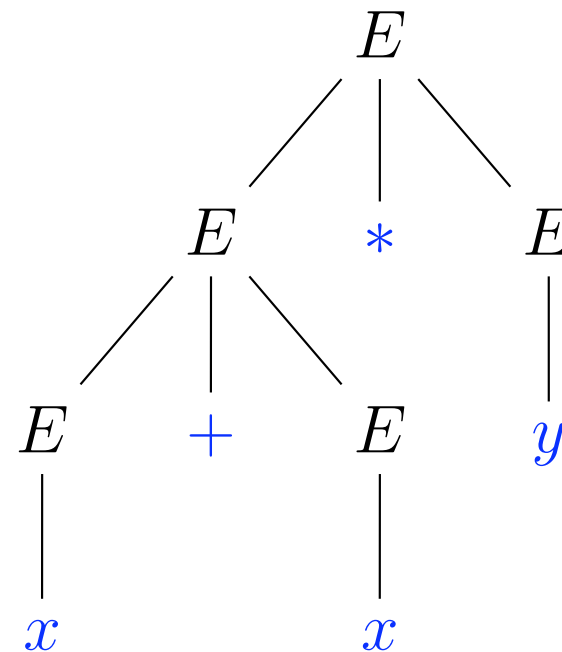
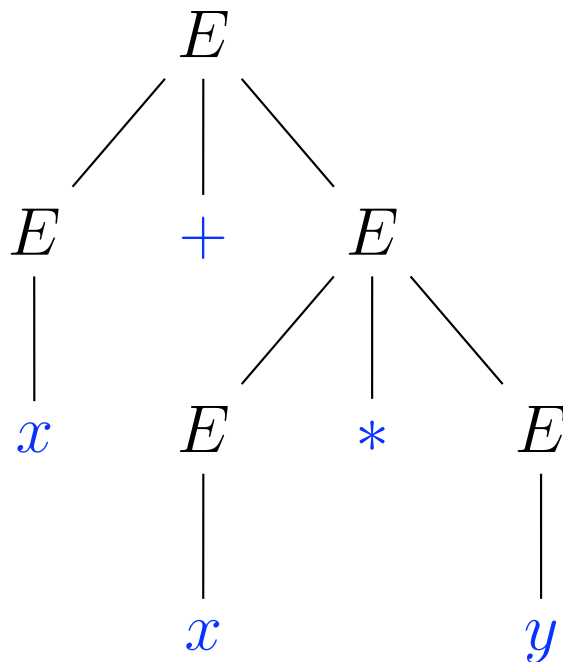
Beispiel für eine mehrdeutige Grammatik:

Sei $G_1 = (\Sigma, N, E, P)$ mit

- $\Sigma = \{0, 1, x, y, -, +, *, (,)\}$
- $N = \{E\}$
- $P = \{E \rightarrow 0, E \rightarrow 1, E \rightarrow x, E \rightarrow y, E \rightarrow -E,$
 $E \rightarrow E + E, E \rightarrow E - E, E \rightarrow E * E, E \rightarrow (E)\}$

Kontextfreie Sprachen

G_1 erzeugt die Sprache der unvollständig geklammerten arithmetischen Ausdrücke und ist (in hohem Maße) mehrdeutig, z.B. hat das Wort $x + x * y$ die beiden folgenden Ableitungsbäume.



Kontextfreie Sprachen

Gibt es eine eindeutige KFG für $L(G_1)$?

Sei $G_2 = (\Sigma, N, E, P)$ mit:

- $\Sigma = \{0, 1, x, y, -, +, *, (,)\}$
- $N = \{E, T, F\}$
- $P = \{$

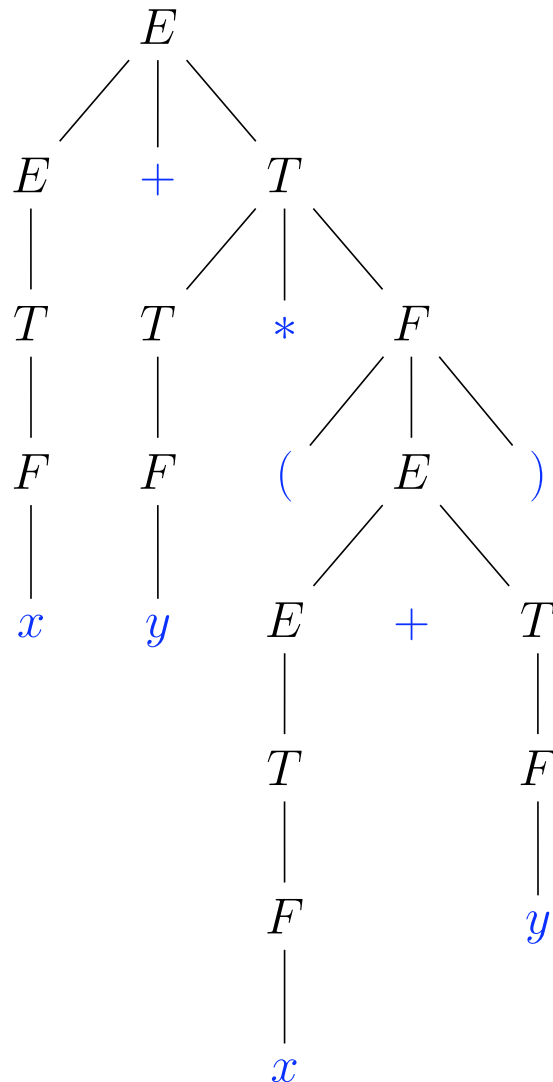
$E \rightarrow E + T,$	(1)
$E \rightarrow E - T,$	(2)
$E \rightarrow T,$	(3)
$T \rightarrow T * F,$	(4)
$T \rightarrow F,$	(5)
$F \rightarrow (E),$	(6)
$F \rightarrow -E,$	(7)
$F \rightarrow 0 \mid 1 \mid x \mid y\}$	(8)

Die eindeutige Linksableitung für $E \xRightarrow{*} x + y * (x + y)$:

- | | |
|-------------------------------|-----|
| $E \Rightarrow E + T$ | (1) |
| $\Rightarrow E + T * F$ | (4) |
| $\Rightarrow E + T * (E)$ | (6) |
| $\Rightarrow E + T * (E + T)$ | (1) |
| $\Rightarrow T + T * (E + T)$ | (3) |
| $\Rightarrow F + T * (E + T)$ | (5) |
| $\Rightarrow x + T * (E + T)$ | (7) |
| \vdots | |
| $\Rightarrow x + y * (x + y)$ | |

Kontextfreie Sprachen

und der zugehörige Ableitungsbaum



Jetzt wäre noch zu zeigen, dass G_2 eindeutig ist und dass $L(G_2) = L(G_1)$. Das beweisen wir nicht.

Man beachte, dass im Ableitungsbaum die üblichen Prioritäten der Operatoren zum Ausdruck kommen: $*$ bindet stärker als $+$, deshalb ist der Gesamtausdruck von der Form $E + T$. Etwas anderes lässt die Grammatik G_2 nicht zu, weil links von $*$ niemals ein $+$ stehen kann, das nicht durch Klammern 'geschützt' ist.