Grundbegriffe der Informatik

Einheit 8: kontextfreie Grammatiken

Thomas Worsch

Karlsruher Institut für Technologie, Fakultät für Informatik

Wintersemester 2009/2010

Überblick

Kontextfreie Grammatiken

- Rekursive Definition syntaktischer Strukturen
- Kontextfreie Grammatiken
- Relationen (Teil 2)

Überblick

Beschreibung formaler Sprachen nur mit Hilfe einzelner Symbole und der Operation Vereinigung, Konkatenation und Konkatenationsabschluss:

- manchmal möglich
- manchmal nicht (Beweis später)

Ausschnitt der Definition der Syntax von Java

1	Block:	
		$\{$ BlockStatements$_{\text{opt}}$ $\}$
2	BlockStatements:	
		BlockStatement
		BlockStatements BlockStatement
3	BlockStatement:	
		Statement
		......
4	Statement:	
		StatementWithoutTrailingSubstatement
		......
5	StatementWithoutTrailingSubstatement:	
		Block
		......

Siehe `http://java.sun.com/docs/books/jls/third_edition/`

Rekursion

- Bei der Beschreibung der Struktur von BlockStatements wird direkt auf BlockStatements Bezug genommen.
- Bei der Definition von Block wird (indirekt) auf die Bedeutung von Statement verwiesen und
- bei der Definition von Statement (indirekt) wieder auf die Bedeutung von Block.
- Was soll das bedeuten?
- beschränken wir uns erst einmal auf den Kern des Ganzen ...

Simplizfizierungen

- schreibe X statt Block, Statement o.ä.
- schreibe runde Klammern (und) statt der geschweiften (wegen der Verwechslungsgefahr mit Mengenklammern)
- Dann besagt die Definition stark vereinfacht unter anderem:
  - K1 Ein X kann etwas "ganz einfaches" sein; schreiben für dafür einfach das leere Wort ε.
  - K2 Ein X kann ein Y sein oder die Folge XY; also kann ein X von der Form YY sein. Jedes Y seinerseits kann wieder ein X sein. Also kann ein X auch von der Form XX sein.
  - K3 Wenn man ein X hat, dann ist auch (X) wieder ein X.
  - K4 Auch gemeint: Es ist nichts ein X, was man nicht auf Grund der obigen Festlegungen als solches identifizieren kann.

Versuch einer formalen Sprache

- versuche, mit X eine formale Sprache L zu assoziieren:

$$L = \{\varepsilon\} \cup LL \cup \{(\}L\{)\}$$

- trügerische Hoffnung:
  - die Inklusion $L \supseteq \ldots$ spiegelt K1, K2, K3 wieder
  - die Inklusion $L \subseteq \ldots$ spiegelt K4 wieder
- Fragen:
  1. Gibt es überhaupt eine Sprache, die die Gleichung erfüllt?
     Das hätten wir gerne und ja, das ist so.
  2. falls ja: Ist die Lösung der Gleichung auch eindeutig?
     Das hätten wir auch gerne, aber nein, das ist nicht so.

     $\Longrightarrow$ Arbeit: Man finde und charakterisiere „irgendwie" die uns interessierende Lösung.

Versuch einer formalen Sprache

- versuche, mit X eine formale Sprache L zu assoziieren:

$$L = \{\varepsilon\} \cup LL \cup \{(\}L\{)\}$$

- trügerische Hoffnung:
  - die Inklusion $L \supseteq \ldots$ spiegelt K1, K2, K3 wieder
  - die Inklusion $L \subseteq \ldots$ spiegelt K4 wieder
- Fragen:
  1. Gibt es überhaupt eine Sprache, die die Gleichung erfüllt?
     Das hätten wir gerne und ja, das ist so.
  2. falls ja: Ist die Lösung der Gleichung auch eindeutig?
     Das hätten wir auch gerne, aber nein, das ist nicht so.

     $\Longrightarrow$ Arbeit: Man finde und charakterisiere „irgendwie" die uns interessierende Lösung.

Versuch einer formalen Sprache

- versuche, mit X eine formale Sprache L zu assoziieren:

$$L = \{\varepsilon\} \cup LL \cup \{(\}L\{)\}$$

- trügerische Hoffnung:
  - die Inklusion $L \supseteq \ldots$ spiegelt K1, K2, K3 wieder
  - die Inklusion $L \subseteq \ldots$ spiegelt K4 wieder
- Fragen:
  1. Gibt es überhaupt eine Sprache, die die Gleichung erfüllt?
     Das hätten wir gerne und ja, das ist so.
  2. falls ja: Ist die Lösung der Gleichung auch eindeutig?
     Das hätten wir auch gerne, aber nein, das ist nicht so.

  $\Longrightarrow$ Arbeit: Man finde und charakterisiere „irgendwie“ die uns interessierende Lösung.

Lösbarkeit von $L = \{\varepsilon\} \cup LL \cup \{(\}L\{)\}$

- konstruiere Folge $L_0, L_1, \ldots$ formaler Sprachen L_i für $i \in \mathbb{N}_0$
- zeige, dass die Vereinigung aller L_i die Gleichung löst

- $L_0 = \{\varepsilon\}$.
- für $i \in \mathbb{N}_0$ sei $L_{i+1} = L_i L_i \cup \{(\}L_i\{)\}$

- **Lemma.** $L = \bigcup_{i=0}^{\infty} L_i$ erfüllt die Gleichung.
- Beweisstruktur
  - zu zeigen: $L = \{\varepsilon\} \cup LL \cup \{(\}L\{)\}$
  - zeige:
    - $L \subseteq \{\varepsilon\} \cup LL \cup \{(\}L\{)\}$
    - $L \supseteq \{\varepsilon\} \cup LL \cup \{(\}L\{)\}$

Lösbarkeit von $L = \{\varepsilon\} \cup LL \cup \{(\}L\{)\}$

- konstruiere Folge $L_0, L_1, \ldots$ formaler Sprachen L_i für $i \in \mathbb{N}_0$
- zeige, dass die Vereinigung aller L_i die Gleichung löst

- $L_0 = \{\varepsilon\}$.
- für $i \in \mathbb{N}_0$ sei $L_{i+1} = L_i L_i \cup \{(\}L_i\{)\}$

- **Lemma.** $L = \bigcup_{i=0}^{\infty} L_i$ erfüllt die Gleichung.
- Beweisstruktur
  - zu zeigen: $L = \{\varepsilon\} \cup LL \cup \{(\}L\{)\}$
  - zeige:
    - $L \subseteq \{\varepsilon\} \cup LL \cup \{(\}L\{)\}$
    - $L \supseteq \{\varepsilon\} \cup LL \cup \{(\}L\{)\}$

Beweis des Lemmas

- $\forall i \in \mathbb{N}_0$: $\varepsilon \in L_i$, denn:
  - $\varepsilon \in L_0$
  - für alle $i \in \mathbb{N}_0$ ist $L_i L_i \subseteq L_{i+1}$,
    wenn $\varepsilon \in L_i$, dann auch $\varepsilon = \varepsilon\varepsilon \in L_i L_i \subseteq L_{i+1}$.
- also $\forall i \in \mathbb{N}_0$: $L_i = L_i\{\varepsilon\} \subseteq L_i L_i \subseteq L_{i+1}$.
- Zeige: $L \subseteq \{\varepsilon\} \cup LL \cup \{(\}L\{)\}$
  - Da $\varepsilon \in L_0 \subseteq L$ ist, ist $L = L\{\varepsilon\} \subseteq LL \subseteq \{\varepsilon\} \cup LL \cup \{(\}L\{)\}$.
- Zeige: $L \supseteq \{\varepsilon\} \cup LL \cup \{(\}L\{)\}$
  - sei $w \in \{\varepsilon\} \cup LL \cup \{(\}L\{)\}$.
  - 1. Fall: $w = \varepsilon$: $w = \varepsilon \in L_0 \subseteq L$.
  - 2. Fall: $w \in LL$:
    Dann $w = w_1 w_2$ mit $w_1 \in L$ und $w_2 \in L$.
    Also existieren Indizes i_1 und i_2 mit $w_1 \in L_{i_1}$ und $w_2 \in L_{i_2}$.
    Für $i = \max(i_1, i_2)$ ist $w_1 \in L_i$ und $w_2 \in L_i$,
    also $w = w_1 w_2 \in L_i L_i \subseteq L_{i+1} \subseteq L$.
  - 3. Fall: $w \in \{(\}L\{)\}$:
    für ein $i \in \mathbb{N}_0$ ist dann $w \in \{(\}L_i\{)\} \subseteq L_{i+1} \subseteq L$.

Lösung von $L = \{\varepsilon\} \cup LL \cup \{(\}L\{)\}$ nicht eindeutig

- $\{(,)\}^*$ ist auch eine Lösung, denn
  - „$\subseteq$“ zeigt man wie oben
  - „$\supseteq$“ ist trivial, da $\{(,)\}^*$ eben *alle* Wörter sind.
- Das ist eine andere Lösung, denn
  - (((ist zwar in $\{(,)\}^*$
  - aber *nicht* in $\bigcup_{i=0}^{\infty} L_i$:
    man vergleiche die Anzahlen der (und)

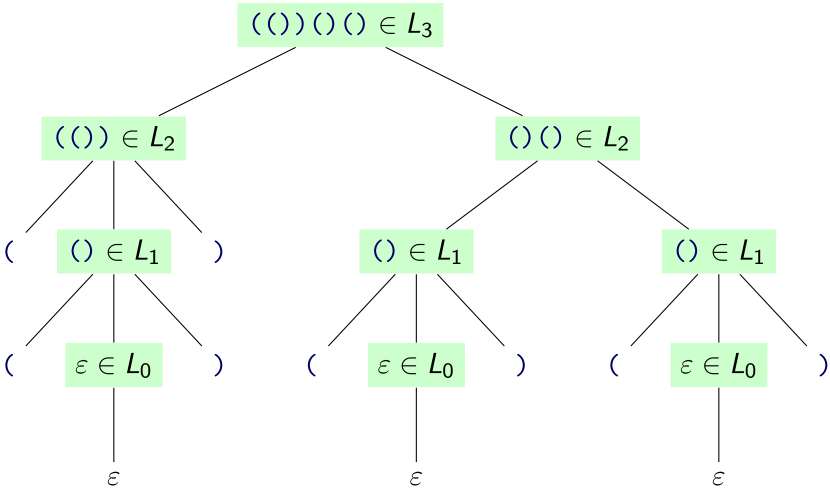
Was kann man an $\bigcup_{i=0}^{\infty} L_i$ sehen?

$$
\begin{aligned}
L_0 &= \{\varepsilon\} \\
L_1 \setminus L_0 &= \{\ ()\ \} \\
L_2 \setminus L_1 &= \{\ ()(),\ (())\ \} \\
L_3 \setminus L_2 &= \{\ ()()(),\ (())(),\ ()(()), \\
&\qquad ()()()(),\ ()()(()),\ (())()(),\ (())(()), \\
&\qquad (()()),\ ((()))\ \}
\end{aligned}
$$

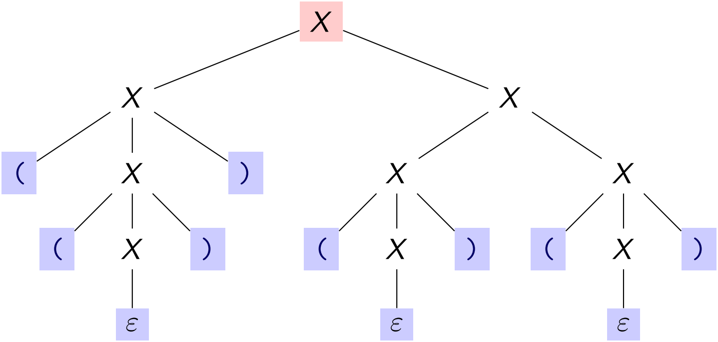
Dabei gilt z. B.:

- $(())()() \in L_3$, weil $(()) \in L_2$ und $()() \in L_2$ und Regel K2
  - $(()) \in L_2$, weil $() \in L_1$ und Regel K3.
    - $() \in L_1$, weil $\varepsilon \in L_0$ und Regel K3.
  - $()() \in L_2$, weil $() \in L_1$ und $() \in L_1$ ist und Regel K2
    - $() \in L_1$, weil $\varepsilon \in L_0$ und Regel K3.
    - $() \in L_1$, weil $\varepsilon \in L_0$ und Regel K3.

Die Erklärung für $(())()() \in L_3$ graphisch dargestellt

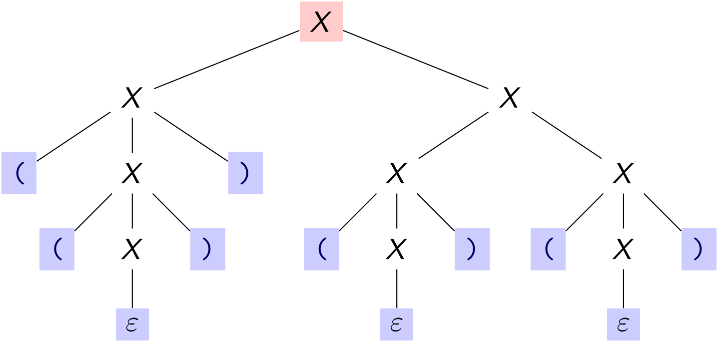


Vereinfachte Darstellung für $(())()() \in L_3$



- So etwas heißt auch Baum (Wurzel oben, Blätter unten)
- bisher: von unten nach oben interpretiert als Begründungen
- kontextfreie Grammatiken: von oben nach unten
  syntaktische Ersetzungen

Vereinfachte Darstellung für (( )) ( ) ( ) $\in L_3$



- So etwas heißt auch Baum (Wurzel oben, Blätter unten)
- bisher: von unten nach oben interpretiert als Begründungen
- kontextfreie Grammatiken: von oben nach unten
  syntaktische Ersetzungen

Was ist wichtig

Das sollten Sie mitnehmen:

- Klammerstrukturen sind wichtig.
- Manchmal kann man sich Fixpunkten „annähern“.
  - Ein Fixpunkt einer Abbildung f ist ein Argument x mit $x = f(x)$.
  - *So* kann man $L = \{\varepsilon\} \cup LL \cup \mathtt{(}L\mathtt{)}$ auch sehen …

Das sollten Sie üben:

- Angst verlieren vor dem Lesen und Finden von Beweisen
  - Bei ruhigem Hinsehen drängt sich eine passende Vorgehensweise manchmal fast auf.

Überblick

Kontextfreie Grammatik $G = (N, T, S, P)$

- N ist ein Alphabet sogenannter *Nichtterminalsymbole*
- T ist ein Alphabet sogenannter *Terminalsymbole*.
  - kein Zeichen in beiden Alphabeten: $N \cap T = \{\}$.
- $S \in N$ ist das sogenannte *Startsymbol*.
- $P \subseteq N \times V^*$ ist *endliche* Menge von *Produktionen*.
  - $V = N \cup T$ Menge aller Symbole überhaupt
  - Schreibweise: $X \to w$ (statt $(X, w) \in P$)
  - Bedeutung: man kann X ersetzen durch w

ein Ableitungsschritt

- Aus $u \in V^*$ ist in einem Schritt $v \in V^*$ ableitbar
  - in Zeichen $u \Rightarrow v$
- wenn Wörter $w_1, w_2 \in V^*$ und eine Produktion $X \to w$ in P existieren, so dass $u = w_1 X w_2$ und $v = w_1 w w_2$.
- Also: Wenn $X \to w$ in P, dann $w_1 X w_2 \Rightarrow w_1 w w_2$.

- Beispiel $G = (\{X\}, \{\mathtt{a}, \mathtt{b}\}, X, P)$ mit Produktionenmenge $P = \{X \to \varepsilon, X \to \mathtt{a}X\mathtt{b}\}$.
- Dann gilt z. B. $\mathtt{aba}X\mathtt{ba}XXXX \Rightarrow \mathtt{aba}X\mathtt{baa}X\mathtt{b}XXX$, denn

$$\underbrace{\mathtt{aba}X\mathtt{ba}}_{w_1}X\underbrace{XXX}_{w_2} \Rightarrow \underbrace{\mathtt{aba}X\mathtt{ba}}_{w_1}\mathtt{a}X\mathtt{b}\underbrace{XXX}_{w_2}$$

- Ebenso gilt $\mathtt{aba}X\mathtt{ba}XXXX \Rightarrow \mathtt{abaa}X\mathtt{bba}XXXX$, denn

$$\underbrace{\mathtt{aba}}_{w_1}X\underbrace{\mathtt{ba}XXXX}_{w_2} \Rightarrow \underbrace{\mathtt{aba}}_{w_1}\mathtt{a}X\mathtt{b}\underbrace{\mathtt{ba}XXXX}_{w_2}$$

Anmerkungen

- Die Definition von $\Rightarrow$ legt eine Relation zwischen Wörtern über dem Alphabet $V = N \cup T$ fest.
- Man könnte also auch schreiben: $R_{\Rightarrow} \subseteq V^* \times V^*$ (oder gar $\Rightarrow \subseteq V^* \times V^*$?)
- üblich: *Infixschreibweise*
  - Man schreibt $u \Rightarrow v$ und nicht $(u, v) \in R_{\Rightarrow}$,
  - so wie man auch $5 \leq 7$ schreibt und nicht $(5, 7) \in R_{\leq}$
- Im allgemeinen ist $\Rightarrow$ weder links- noch rechtstotal und weder links- noch rechtseindeutig.
- bei einer Produktion
  - linke Seite ist immer ein Nichtterminalsymbol
  - In Ableitungsschritt wird nie ein Terminalsymbol ersetzt.
  - Wo sie stehen, ist „die Ableitung zu Ende“
  - daher der Name *Terminal*symbol.

Anmerkungen

- Die Definition von $\Rightarrow$ legt eine Relation zwischen Wörtern über dem Alphabet $V = N \cup T$ fest.
- Man könnte also auch schreiben: $R_{\Rightarrow} \subseteq V^* \times V^*$ (oder gar $\Rightarrow \subseteq V^* \times V^*$?)
- üblich: *Infixschreibweise*
  - Man schreibt $u \Rightarrow v$ und nicht $(u, v) \in R_{\Rightarrow}$,
  - so wie man auch $5 \leq 7$ schreibt und nicht $(5, 7) \in R_{\leq}$
- Im allgemeinen ist $\Rightarrow$ weder links- noch rechtstotal und weder links- noch rechtseindeutig.
- bei einer Produktion
  - linke Seite ist immer ein Nichtterminalsymbol
  - In Ableitungsschritt wird nie ein Terminalsymbol ersetzt.
  - Wo sie stehen, ist „die Ableitung zu Ende“
  - daher der Name *Terminal*symbol.

Anmerkungen

- Die Definition von $\Rightarrow$ legt eine Relation zwischen Wörtern über dem Alphabet $V = N \cup T$ fest.
- Man könnte also auch schreiben: $R_{\Rightarrow} \subseteq V^* \times V^*$ (oder gar $\Rightarrow \subseteq V^* \times V^*$?)
- üblich: *Infixschreibweise*
  - Man schreibt $u \Rightarrow v$ und nicht $(u, v) \in R_{\Rightarrow}$,
  - so wie man auch $5 \leq 7$ schreibt und nicht $(5, 7) \in R_{\leq}$
- Im allgemeinen ist $\Rightarrow$ weder links- noch rechtstotal und weder links- noch rechtseindeutig.
- bei einer Produktion
  - linke Seite ist immer ein Nichtterminalsymbol
  - In Ableitungsschritt wird nie ein Terminalsymbol ersetzt.
  - Wo sie stehen, ist „die Ableitung zu Ende“
  - daher der Name *Terminal*symbol.

- Eine *Ableitung(sfolge)* ist eine Folge von Ableitungsschritten, deren Anzahl irrelevant ist.
- Formal: Für alle $u, v \in V^*$ gelte

$$\begin{aligned} u \Rightarrow^0 v &\text{ genau dann, wenn } u = v \\ \forall i \in \mathbb{N}_0 : (u \Rightarrow^{i+1} v &\text{ genau dann, wenn } \exists w \in V^* : u \Rightarrow w \Rightarrow^i v) \\ u \Rightarrow^* v &\text{ genau dann, wenn } \exists i \in \mathbb{N}_0 : u \Rightarrow^i v \end{aligned}$$

- Beispielgrammatik $G = (\{X\}, \{\mathtt{a}, \mathtt{b}\}, X, \{X \to \varepsilon, X \to \mathtt{a}X\mathtt{b}\})$:

$$X \Rightarrow \mathtt{a}X\mathtt{b} \Rightarrow \mathtt{aa}X\mathtt{bb} \Rightarrow \mathtt{aaa}X\mathtt{bbb} \Rightarrow \mathtt{aaabbb}$$

- Also gilt z. B.: $X \Rightarrow^* \mathtt{aa}X\mathtt{bb}$, $\mathtt{a}X\mathtt{b} \Rightarrow^* \mathtt{aaa}X\mathtt{bbb}$, $X \Rightarrow^* \mathtt{aaabbb}$ und viele andere.

erzeugte formale Sprache

- Hauptinteresse: Welche Wörter aus Terminalsymbolen können aus dem Startsymbol abgeleitet werden?
- *von Grammatik* $G = (N, T, S, P)$ *erzeugte formale Sprache*

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

- solche formalen Sprachen heißen auch *kontextfrei*

Beispiel einer kontextfreien Sprache

- Beispielgrammatik $G = (\{X\}, \{\mathtt{a}, \mathtt{b}\}, X, \{X \to \varepsilon, X \to \mathtt{a}X\mathtt{b}\}$
- eben schon gesehen: $\mathtt{aaabbb} \in L(G)$ wegen

$$X \Rightarrow \mathtt{a}X\mathtt{b} \Rightarrow \mathtt{aa}X\mathtt{bb} \Rightarrow \mathtt{aaa}X\mathtt{bbb} \Rightarrow \mathtt{aaabbb}$$

- leicht verallgemeinerbar: für alle $i \in \mathbb{N}_0$ gilt: $X \Rightarrow \mathtt{a}^i\mathtt{b}^i$,
- also $\{\mathtt{a}^i\mathtt{b}^i \mid i \in \mathbb{N}_0\} \subseteq L(G)$
- Beweis wird leichter, wenn man gleich zeigt:

$$\forall i \in \mathbb{N}_0 : (X \Rightarrow^* \mathtt{a}^i\mathtt{b}^i \land X \Rightarrow^* \mathtt{a}^i X \mathtt{b}^i)$$

- Umgekehrt kann man zeigen:

$$\forall i \in \mathbb{N}_0 : \text{ wenn } X \Rightarrow^{i+1} w, \text{ dann } w = \mathtt{a}^i\mathtt{b}^i \lor w = \mathtt{a}^{i+1}X\mathtt{b}^{i+1}$$

- also $L(G) \subseteq \{\mathtt{a}^i\mathtt{b}^i \mid i \in \mathbb{N}_0\}$
- Insgesamt:

$$L(G) = \{\mathtt{a}^i\mathtt{b}^i \mid i \in \mathbb{N}_0\} .$$

- Statt $\{X \to w_1, X \to w_2, X \to w_3, X \to w_4, X \to w_5\}$ schreibt man $\{X \to w_1|w_2|w_3|w_4|w_5\}$
- und liest die senkrechten Striche als „oder“.
- Beispielgrammatik:

$$P = \{\ X \to \mathtt{a}X\mathtt{b} \mid \varepsilon\ \}$$

Interpretation der Definition der Java-Syntax

eine kontextfreien Grammatik

- Block, ... sind jeweils *ein* Nichtterminalsymbol.
- Doppelpunkt entspricht Pfeil $\rightarrow$
- eingerückte Zeile: rechte Seite einer Produktion
- aufeinander folgende Zeilen denke man sich durch senkrechte Striche $|$ getrennt
- Beispiel

```
2       BlockStatements:
                BlockStatement
                BlockStatements BlockStatement
```

- bedeutet

$$\begin{aligned} \mathit{BlockStatements} &\rightarrow \mathit{BlockStatement} \\ &\mid \mathit{BlockStatements}\,\mathit{BlockStatement} \end{aligned}$$

Interpretation der Definition der Java-Syntax (2)

- ähnlich

1	Block:	
		{ BlockStatements$_{\text{opt}}$ }

- bedeutet

$$Block \rightarrow \{\ BlockStatements\ \} \mid \{\ \}$$

kontextfreie Grammatiken für Syntax von Programmiersprachen?

- (jedenfalls viele) Nichtterminalsymbole stehen für strukturelle Konzepte der Programmiersprache.
- das Ideal:
  Man kann mit der Grammatik für Java genau alle syntaktisch korrekten Javaprogramme ableiten kann, aber auch nur diese und nichts anderes.
- die Realität ist komplizierter:
  - Was mit der Grammatik nicht ableitbar ist, ist bestimmt kein Javaprogramm.
    - gut
  - Aber Dinge ableitbar, die keine korrekten Programme sind.
    - nicht gut.
    - Grund: manche Forderungen kann man überhaupt nicht mit Hilfe kontextfreier Grammatiken ausdrücken.

Ableitungsbäume

- Grammatik für unser Klammerproblem:
  $(\{X\}, \{(,)\}, X, \{X \to XX | (X) | \varepsilon\})$.
- lange Ableitungsfolgen manchmal nicht sehr erhellend:

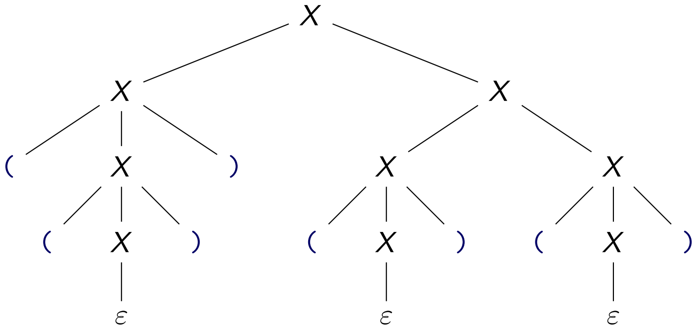
$$X \Rightarrow XX \Rightarrow (X)X \Rightarrow (X)XX \Rightarrow (X)X(X) \Rightarrow ((X))X(X)$$
$$\Rightarrow ((X))X() \Rightarrow ((X))(X)() \Rightarrow (())(X)() \Rightarrow (())()()$$

- man darf umordnen (Kontextfreiheit!)
- schon besser: *Linksableitung*

$$X \Rightarrow XX \Rightarrow (X)X \Rightarrow ((X))X \Rightarrow (())X \Rightarrow (())XX$$
$$\Rightarrow (())(X)X \Rightarrow (())()X \Rightarrow (())()(X) \Rightarrow (())()()$$

- manchmal noch übersichtlicher: *Ableitungsbaum*

ein Ableitungsbaum



- Man beginnt mit dem Startsymbol als Wurzel.
- Für jeden Ableitungsschritt werden an das ersetzte Nichtterminalsymbol Kanten nach unten dran gehängt.
- (Wir verzichten an dieser Stelle auf eine Formalisierung.)

Was ist wichtig

Das sollten Sie mitnehmen:

- kontextfreie Grammatik
- Ableitung
- erzeugte formale Sprache
- Ableitungsbaum

Das sollten Sie üben:

- (semi-)reale Produktionenmengen lesen (Java, . . .)
- zu formaler Sprache sie erzeugende kontextfreie Grammatik konstruieren
- zu kontextfreier Grammatik die erzeugte formale Sprache bestimmen

Überblick

- Es seien $R \subseteq M_1 \times M_2$ und $S \subseteq M_2 \times M_3$ zwei Relationen
- Dann heißt

  $$S \circ R = \{(x,z) \in M_1 \times M_3 \mid \exists y \in M_2 : (x,y) \in R \wedge (y,z) \in S\}$$

  das *Produkt der Relationen* R und S.
- oder in Infixschreibweise: für alle $(x,z) \in M_1 \times M_3$

  $$x(S \circ R)z \text{ gdw. } \exists y \in M_2 : xRy \wedge ySz$$

- Mitteilung: Das Relationenprodukt ist eine assoziative Operation.

- Mit Id_M bezeichnen wir die Relation

  $$\mathrm{Id}_M = \{(x,x) \mid x \in M\}$$

  Das ist die identische Abbildung auf der Menge M.
- Für jede binäre Relation $R \subseteq M \times M$ gilt:

  $$R \circ \mathrm{Id}_M = R = \mathrm{Id}_M \circ R$$

Potenzen und reflexiv-transitive Hülle einer Relation

- Ist $R \subseteq M \times M$ binäre Relation auf einer Menge M, dann definiert man *Potenzen* R^i:

$$R^0 = \mathrm{Id}_M$$
$$\forall i \in \mathbb{N}_0 : R^{i+1} = R^i \circ R$$

- Die *reflexiv-transitive Hülle* einer Relation R ist

$$R^* = \bigcup_{i=0}^{\infty} R^i$$

- Die reflexiv-transitive Hülle R^* einer Relation R hat folgende Eigenschaften:
  - R^* ist reflexiv.
  - R^* ist transitiv.
  - R^* ist die kleinste Relation, die R enthält und reflexiv und transitiv ist.
- Relation R heißt *reflexiv*, wenn $\mathrm{Id}_M \subseteq R$ ist.
- Relation R heißt *transitiv*, wenn gilt:

$$\forall x \in M : \forall y \in M : \forall z \in M : xRy \land yRz \Longrightarrow xRz$$

(Das ist ein Implikationspfeil und kein Ableitungspfeil.)

- R^* ist immer reflexiv, denn
  $\mathrm{Id}_M = R^0 \subseteq R^*$
- Man kann zeigen:
  für alle $i, j \in \mathbb{N}_0$ gilt: $R^i \circ R^j = R^{i+j}$.
- Daraus folgt: R^* ist immer transitiv, denn
  - wenn $(x, y) \in R^*$ und $(y, z) \in R^*$,
  - dann gibt es i und $j \in \mathbb{N}_0$ mit $(x, y) \in R^i$ und $(y, z) \in R^j$
  - dann ist $(x, z) \in R^i \circ R^j = R^{i+j} \subseteq R^*$.
- R^* ist die *kleinste* Relation, die R umfasst und reflexiv und transitiv ist:
  - R^* umfasst R und ist reflexiv und transitiv.
  - Es sei S eine beliebige Relation ist, die reflexiv und transitiv ist.
  - Wenn S die Relation R umfasst, also $R \subseteq S$, dann sogar
    $R^* \subseteq S$.

Was ist wichtig

Das sollten Sie mitnehmen:

- Produkte und Potenzen von Relationen
- reflexive und transitive Relationen
- reflexiv-transitive Hülle einer Relation
  - „klassisches“ Beispiel: Ableitbarkeit $\Rightarrow^*$

Das sollten Sie üben:

- Transitivität nachweisen
- Bilder von Relationen malen

Zusammenfassung

- rekursive Definitionen syntaktischer Strukturen
  - Vorsicht kann nicht schaden
  - zumindest manchmal sinnvolle Interpretation möglich
- Grammatiken
  - kontextfreie Grammatik
  - Ableitung
  - erzeugte formale Sprache
  - Ableitungsbaum
- Relationen
  - Produkte und Potenzen von Relationen
  - reflexive und transitive Relationen
  - reflexiv-transitive Hülle einer Relation