

- Frage aus der letzten Vorlesung:
Hat der Studiengang Informationswirtschaft auch die komplette Technische Informatik und Praktische Informatik - Einheit , sprich Softwaretechnik,... wie die Informatiker?

Modulhandbuch Informatik B.Sc. (WS 2011/2012)
www.informatik.kit.edu/downloads/studium/mhb_ws11_12_info_bsc_de_lang.pdf

Informatik	Informatik	Informatik	Informatik	Leistungsstufe 3 – Semester 5+6		
				+ Wahlmodule + 2 Stammmodule + 1 Proseminar + 1 Ergänzungsfachmodul		
ALG II	PROGP			Wahlbereich Informatik	Ergänzungsmodul	
	K&D		Numerik			
TheoG	SWP + BS	TI - Dig	Statistik			SETeam
ALG I	SWT I	TI - RO	HM / LA II			
GBI	PROG		HM / LA I			

Grundbegriffe der Informatik (GBI) wichtige Grundlage für:

- Algorithmen II (Approximations-, Lineare Programmierung, randomisierte, parallele, und parametrisierte Algorithmen) Profs. Wagner / Sanders
- Programmierparadigmen (rek. Fkt, Lambda-Kalkül, Log. Program. - Terme, Hornklauseln, Unifikation, Parallele Prog., Compilerbau) Profs. Snelling / Reussner

- Antwort: Nein, stattdessen gibt es Angewandte Informatik 1 und 2

ModulID	Lehrveranstaltung	SWS	LP
1. Semester			
IW1BWL1/ IW1BWL3	Betriebswirtschaftslehre: Finanzwirtschaft und Rechnungswesen	2/0/2	4.0
IW1VWL	Volkswirtschaftslehre I	3/0/2	5.0
IW1MATH1	Mathematik I	4/2/2	8.0
IW1INF1	Grundbegriffe der Informatik	2/1/2	5.0
IW1INF1	Programmieren	2/0/2	5.0
IW1JURA1	BGB für Anfänger	4/0	4.0
			31.0
2. Semester			
IW1BWL1/ IW1BWL3	Einführung in die Informationswirtschaft	2/2	4.0
IW1STAT	Statistik I	4/0/2	5.0
IW1OR	Einführung in das OR I	2/2/2	4.5
IW1MATH2	Mathematik II	4/2/2	8.0
IW2INF2	Algorithmen I	3/1/2	6.0
IW1JURA2	BGB für Fortgeschrittene	2/0	3.0
			30.5
3. Semester			
IW1BWL2/ IW1BWL4	Rechnungswesen	2/2	4.0
IW1STAT	Statistik II	4/0/2	5.0
IW1OR	Einführung in das OR II	2/2/2	4.5
IW2INF3	Theoretische Grundlagen der Informatik	3/1	7.0
IW1INF5	Angewandte Informatik I	2/1	4.0
IW1JURA2	Handels- und Gesellschaftsrecht	2/0	3.0
IW1JURA3	Öffentliches Recht I	2/0	3.0
			30.5
4. Semester			
IW1BWL2/ IW1BWL4	Betriebswirtschaftslehre: Produktionswirtschaft und Marketing	2/0/2	4.0
IW1INF5	Angewandte Informatik II	2/1	4.0
IW2INF4	Rechnerorganisation	3/1/2	6.0
IW1JURA3	Öffentliches Recht II	2/0	3.0
IW1JURA2	Privatrechtliche Übung	2/0	3.0
IW1PRAK	Betriebspraktikum		8.0
			28.0

- 5+6. Semester: 18 LP Informatik Module + 24 LP Module aus BWL, OR, VWL und Recht

Grundbegriffe der Informatik

Einheit 8: kontextfreie Grammatiken

Prof. Dr. Tanja Schultz

Karlsruher Institut für Technologie, Fakultät für Informatik

Wintersemester 2012/2013

Kontextfreie Grammatiken

Rekursive Definition syntaktischer Strukturen

Kontextfreie Grammatiken

Relationen (Teil 2)

Kontextfreie Grammatiken

Rekursive Definition syntaktischer Strukturen

Kontextfreie Grammatiken

Relationen (Teil 2)

Kontextfreie Grammatiken

Rekursive Definition syntaktischer Strukturen

Kontextfreie Grammatiken

Relationen (Teil 2)

In der Einheit 6 über „Dokumente“ hatten wir gelernt, dass die Beschreibung formaler Sprachen nur mit Hilfe einzelner Symbole und der Operation Vereinigung, Konkatenation und Konkatenationsabschluss:

- ▶ manchmal möglich ist
- ▶ manchmal nicht möglich ist (Beweis später)

Ein typischer Problemfall sind Rekursive Definitionen

Beispiel:

Definition der Syntax von Java

- 1 Block:
 - { BlockStatements_{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/

Vereinbarung um das Lesen zu vereinfachen

- ▶ Zur Unterscheidung zwischen normalem Text und Wörtern der Spezifikation
- ▶ Blockstatements - $\langle BlockStatements \rangle$; Block - $\langle Block \rangle$
- ▶ also: $\langle Block \rangle$ sehen wir als *ein* „Ding“
- ▶ dann können wir obigen Ausschnitt aus der Spezifikation der Syntax von Java wie folgt schreiben:

```
1       $\langle Block \rangle$ :  
        {  $\langle BlockStatements \rangle_{opt}$  }  
2       $\langle BlockStatements \rangle$ :  
         $\langle BlockStatement \rangle$   
         $\langle BlockStatements \rangle \langle BlockStatement \rangle$   
3       $\langle BlockStatement \rangle$ :  
         $\langle Statement \rangle$   
        .....  
USW.    ...
```

- ▶ Die Struktur von $\langle Block \rangle$ ist: ein Zeichen $\{$, eventuell gefolgt von $\langle BlockStatements \rangle$ (das Suffix $_{opt}$ bedeutet, dass $\langle BlockStatements \rangle$ optional ist), gefolgt von $\}$
- ▶ Problem: Rekursive Definition, d.h.
- ▶ Bei der Beschreibung der Struktur von $\langle BlockStatements \rangle$ wird direkt auf $\langle BlockStatements \rangle$ Bezug genommen.
- ▶ Bei der Definition von $\langle Block \rangle$ wird (indirekt) auf die Bedeutung von $\langle Statement \rangle$ verwiesen und
- ▶ bei der Definition von $\langle Statement \rangle$ wird (indirekt) wieder auf die Bedeutung von $\langle Block \rangle$ verwiesen.
- ▶ Was soll das bedeuten?
- ▶ beschränken wir uns erst einmal auf den Kern des Ganzen ...

- ▶ schreibe X statt $\langle Block \rangle$, $\langle Statement \rangle$ 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.
- ▶ was jetzt?

- ▶ Versuche, mit X eine formale Sprache L zu assoziieren:

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

- ▶ trügerische Hoffnung:

- ▶ die Inklusion $L \supseteq \dots$ spiegelt K1, K2, K3 wieder
- ▶ die Inklusion $L \subseteq \dots$ spiegelt K4 wieder

- ▶ Fragen:

1. Gibt es überhaupt eine formale Sprache L , die die obige Gleichung erfüllt?

Das hätten wir gerne ... und ja, das ist so ... und wir werden eine Lösung der Gleichung konstruieren.

2. Falls ja: Ist die formale Sprache L , die obige Gleichung erfüllt, durch diese Gleichung eindeutig festgelegt?

Das hätten wir auch gerne, aber nein, das ist nicht so.

⇒ Das bedeutet zusätzliche Arbeit. Denn, wir müssen aus den Lösungen „irgendwie“ die uns interessierende Lösung herausfinden und charakterisieren.

- ▶ Versuche, mit X eine formale Sprache L zu assoziieren:

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

- ▶ trügerische Hoffnung:

- ▶ die Inklusion $L \supseteq \dots$ spiegelt K1, K2, K3 wieder
- ▶ die Inklusion $L \subseteq \dots$ spiegelt K4 wieder

- ▶ Fragen:

1. Gibt es überhaupt eine formale Sprache L , die die obige Gleichung erfüllt?

Das hätten wir gerne ... und ja, das ist so ... und wir werden eine Lösung der Gleichung konstruieren.

2. Falls ja: Ist die formale Sprache L , die obige Gleichung erfüllt, durch diese Gleichung eindeutig festgelegt?

Das hätten wir auch gerne, aber nein, das ist nicht so.

⇒ Das bedeutet zusätzliche Arbeit. Denn, wir müssen aus den Lösungen „irgendwie“ die uns interessierende Lösung herausfinden und charakterisieren.

- ▶ Versuche, mit X eine formale Sprache L zu assoziieren:

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

- ▶ trügerische Hoffnung:

- ▶ die Inklusion $L \supseteq \dots$ spiegelt K1, K2, K3 wieder
- ▶ die Inklusion $L \subseteq \dots$ spiegelt K4 wieder

- ▶ Fragen:

1. Gibt es überhaupt eine formale Sprache L , die die obige Gleichung erfüllt?

Das hätten wir gerne ... und ja, das ist so ... und wir werden eine Lösung der Gleichung konstruieren.

2. Falls ja: Ist die formale Sprache L , die obige Gleichung erfüllt, durch diese Gleichung eindeutig festgelegt?

Das hätten wir auch gerne, aber nein, das ist nicht so.

⇒ Das bedeutet zusätzliche Arbeit. Denn, wir müssen aus den Lösungen „irgendwie“ die uns interessierende Lösung herausfinden und charakterisieren.

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

1: Konstruiere eine Lösung der Gleichung $L = \{\varepsilon\} \cup LL \cup \{()L()\}$

- ▶ Ansatz: konstruiere eine Folge L_0, L_1, \dots formaler Sprachen L_i für $i \in \mathbb{N}_0$, so dass jedes L_i einige der gewünschten Wörter enthält
- ▶ zeige dann, dass die Vereinigung aller L_i die Gleichung L 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:
 - ▶ $L \subseteq \{\varepsilon\} \cup LL \cup \{()L()\}$
 - ▶ $L \supseteq \{\varepsilon\} \cup LL \cup \{()L()\}$

1: Konstruiere eine Lösung der Gleichung $L = \{\varepsilon\} \cup LL \cup \{()L\}$

- ▶ Ansatz: konstruiere eine Folge L_0, L_1, \dots formaler Sprachen L_i für $i \in \mathbb{N}_0$, so dass jedes L_i einige der gewünschten Wörter enthält
- ▶ zeige dann, dass die Vereinigung aller L_i die Gleichung L 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:
 - ▶ $L \subseteq \{\varepsilon\} \cup LL \cup \{()L\}$
 - ▶ $L \supseteq \{\varepsilon\} \cup LL \cup \{()L\}$

- ▶ $\forall i \in \mathbb{N}_0: \varepsilon \in L_i$, 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$.

- ▶ Der Beweis des Lemmas zeigt, dass die Gleichung (mindestens) eine Lösung hat.
- ▶ Allerdings gibt es noch weitere Lösungen, z.B.
- ▶ $\{(\,,\,)\}^*$ ist auch eine Lösung, denn
 - ▶ „ \subseteq “ zeigt man wie oben
 - ▶ „ \supseteq “ ist trivial, da $\{(\,,\,)\}^*$ eben *alle* Wörter sind.
- ▶ Dies ist aber eine andere Lösung, denn beispielsweise
 - ▶ $(((($ ist zwar in $\{(\,,\,)\}^*$
 - ▶ aber *nicht* in $\bigcup_{i=0}^{\infty} L_i$:
man vergleiche die Anzahlen der (und)
 - ▶ $\bigcup_{i=0}^{\infty} L_i$ enthält nämlich nur Wörter, in denen gleich viele (und) vorkommen
 - ▶ man kann nämlich durch vollständige Induktion beweisen, dass dies für alle L_i gilt - Übung
- ▶ Also ist die Lösung der Gleichung $L = \{\varepsilon\} \cup LL \cup \{(\{L\})\}$ nicht eindeutig

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

$$L_0 = \{\varepsilon\}$$

$$L_1 \setminus L_0 = \{ () \}$$

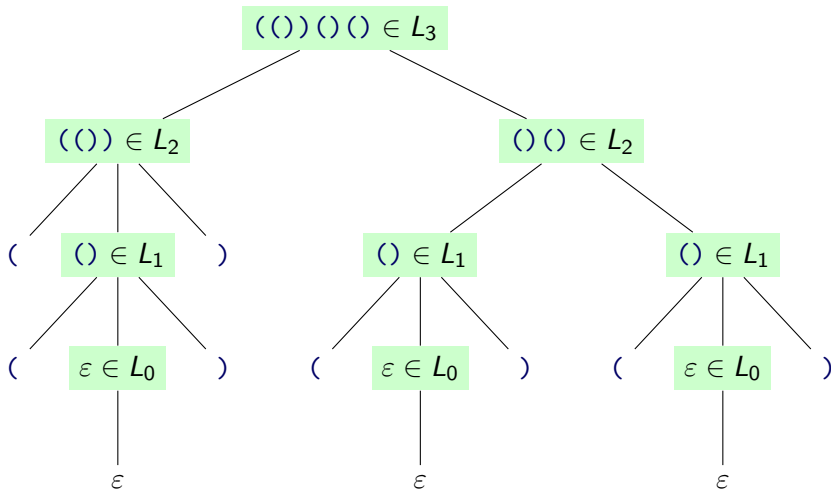
$$L_2 \setminus L_1 = \{ ()(), (()) \}$$

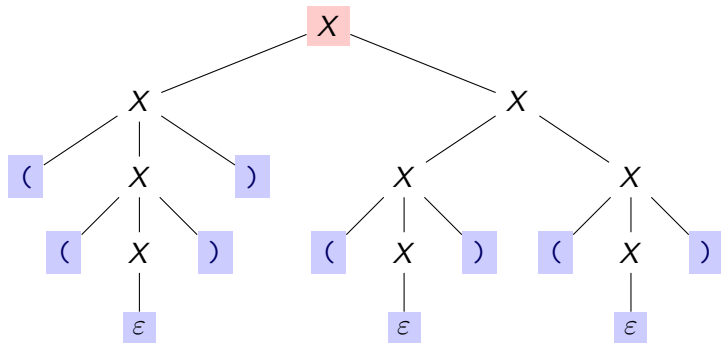
$$L_3 \setminus L_2 = \{ ()()(), (())(), ()(), ()()(), ()()(), (())(), (())(), (())(), ((())) \}$$

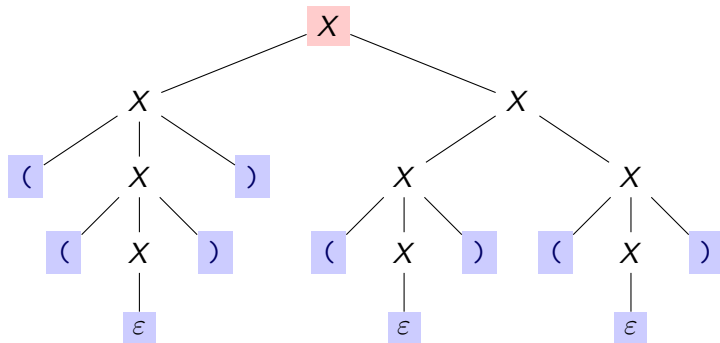
Dabei gilt z. B.:

- ▶ $(())()() \in L_3$, weil $(()) \in L_2$ und $()() \in L_2$ und Regel K2 (K2: X kann Y sein, X kann auch XY sein)
 - ▶ $(()) \in L_2$, weil $() \in L_1$ und Regel K3 (K3: wenn man X hat, dann ist (X) wieder ein X)
 - ▶ $() \in L_1$, 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







- ▶ 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

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

Kontextfreie Grammatiken

Rekursive Definition syntaktischer Strukturen

Kontextfreie Grammatiken

Relationen (Teil 2)

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: $X \rightarrow w$ (statt $(X, w) \in P$)
 - ▶ Bedeutung: man kann X ersetzen durch w

- ▶ 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 $X \rightarrow w$ in P existieren, so dass $u = w_1 X w_2$ und $v = w_1 w w_2$.
- ▶ Also: Wenn $X \rightarrow w$ in P , dann $w_1 X w_2 \Rightarrow w_1 w w_2$.
- ▶ Beispiel $G = (\{X\}, \{a, b\}, X, P)$ mit Produktionsmenge $P = \{X \rightarrow \varepsilon, X \rightarrow aXb\}$.
- ▶ Dann gilt z. B. $abaXbaXXXX \Rightarrow abaXbaaXbXXX$, denn

$$\underbrace{abaXba}_{w_1} \underbrace{X}_{w_2} XXX \Rightarrow \underbrace{abaXba}_{w_1} \underbrace{aXb}_{w_2} XXX$$

- ▶ Ebenso gilt $abaXbaXXXX \Rightarrow abaaXbbaXXXX$, denn

$$\underbrace{aba}_{w_1} \underbrace{XbaXXXX}_{w_2} \Rightarrow \underbrace{aba}_{w_1} \underbrace{aXbbaXXXX}_{w_2}$$

- ▶ Die Definition von \Rightarrow legt eine Relation zwischen Wörtern über dem Alphabet $V = N \cup T$ fest.
- ▶ Man könnte also auch schreiben: $R_{\Rightarrow} \subseteq V^* \times V^*$
- ▶ 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 ...
 - ▶ ist die linke Seite immer ein Nichtterminalsymbol.
 - ▶ wird in der Ableitung nie ein Terminalsymbol ersetzt,
 - ▶ sondern an der Stelle ist „die Ableitung zu Ende“,
 - ▶ daher der Name *Terminalsymbol*.

- ▶ Die Definition von \Rightarrow legt eine Relation zwischen Wörtern über dem Alphabet $V = N \cup T$ fest.
- ▶ Man könnte also auch schreiben: $R_{\Rightarrow} \subseteq V^* \times V^*$
- ▶ 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 ...
 - ▶ ist die linke Seite immer ein Nichtterminalsymbol.
 - ▶ wird in der Ableitung nie ein Terminalsymbol ersetzt,
 - ▶ sondern an der Stelle ist „die Ableitung zu Ende“,
 - ▶ daher der Name *Terminalsymbol*.

- ▶ Die Definition von \Rightarrow legt eine Relation zwischen Wörtern über dem Alphabet $V = N \cup T$ fest.
- ▶ Man könnte also auch schreiben: $R_{\Rightarrow} \subseteq V^* \times V^*$
- ▶ 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 ...
 - ▶ ist die linke Seite immer ein Nichtterminalsymbol.
 - ▶ wird in der Ableitung nie ein Terminalsymbol ersetzt,
 - ▶ sondern an der Stelle ist „die Ableitung zu Ende“,
 - ▶ daher der Name *Terminalsymbol*.

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

$$u \Rightarrow^0 v \text{ genau dann, wenn } u = v$$

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

$$u \Rightarrow^* v \text{ genau dann, wenn } \exists i \in \mathbb{N}_0 : u \Rightarrow^i v$$

- ▶ Beispielgrammatik $G = (\{X\}, \{a, b\}, X, \{X \rightarrow \varepsilon, X \rightarrow aXb\})$:

$$X \Rightarrow aXb \Rightarrow aaXbb \Rightarrow aaaXbbb \Rightarrow aaabbb$$

- ▶ Also gilt z. B.: $X \Rightarrow^* aaXbb$, $aXb \Rightarrow^* aaaXbbb$,
 $X \Rightarrow^* aaabbb$ und viele andere.

- ▶ 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*
- ▶ $L(G)$ heißt „kontextfreie Sprache“, weil die Regeln immer unabhängig vom Kontext (einer nichtleeren Folge von $v \in V$) angewendet werden
- ▶ also bspw. $X \rightarrow aXb$ aber NICHT $Xa \rightarrow aXb$

- ▶ Beispielgrammatik $G = (\{X\}, \{a, b\}, X, \{X \rightarrow \varepsilon, X \rightarrow aXb\})$
- ▶ eben schon gesehen: $aaabbbb \in L(G)$ wegen

$$X \Rightarrow aXb \Rightarrow aaXbb \Rightarrow aaaXbbb \Rightarrow aaabbbb$$

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

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

- ▶ Umgekehrt kann man zeigen:

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

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

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

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

$$P = \{ X \rightarrow \textcolor{blue}{a}X\textcolor{blue}{b} \mid \varepsilon \}$$

eine kontextfreien Grammatik

- ▶ $\langle Block \rangle$, ... 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

$$\begin{array}{l} \langle BlockStatements \rangle \rightarrow \langle BlockStatement \rangle \\ \quad | \quad \langle BlockStatements \rangle \langle BlockStatement \rangle \end{array}$$

- ▶ ähnlich

1 Block:

$\{ \text{BlockStatements}_{\text{opt}} \}$

- ▶ bedeutet

$$\langle \text{Block} \rangle \rightarrow \{ \langle \text{BlockStatements} \rangle \} \mid \{ \}$$

- ▶ (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.

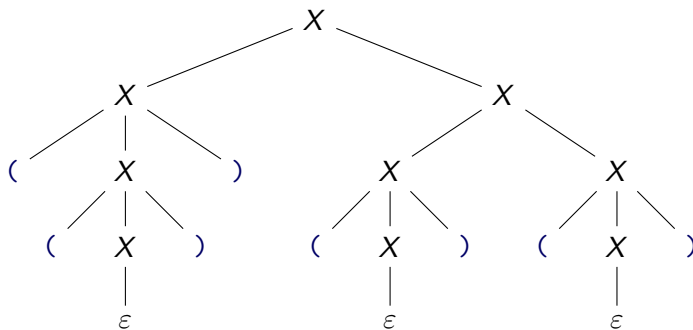
- ▶ Grammatik für unser Klammerproblem:
 $(\{X\}, \{ (,) \}, X, \{ X \rightarrow XX \mid (X) \mid \varepsilon \})$.
- ▶ lange Ableitungsfolgen manchmal nicht sehr erhellend:

$$\begin{aligned} X &\Rightarrow XX \Rightarrow (X)X \Rightarrow (X)XX \Rightarrow (X)X(X) \Rightarrow ((X))X(X) \\ &\Rightarrow ((X))X() \Rightarrow ((X))(X)() \Rightarrow (())(X)() \Rightarrow (())()() \end{aligned}$$

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

$$\begin{aligned} X &\Rightarrow XX \Rightarrow (X)X \Rightarrow ((X))X \Rightarrow (())X \Rightarrow (())XX \\ &\Rightarrow (())(X)X \Rightarrow (())()X \Rightarrow (())()(X) \Rightarrow (())()() \end{aligned}$$

- ▶ manchmal noch übersichtlicher: *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 $(\{X\}, \{(,)\}, X, \{X \rightarrow XX \mid (X) \mid \varepsilon\})$ erzeugt, nennt man *wohlgeformte* oder *korrekte Klammerausdrücke* (KKA).
- ▶ umgangssprachliche Festlegungen
 - ▶ was sich ergibt, wenn man in „normalen“ arithmetische Ausdrücken alles außer den Klammern weglässt
 - ▶ präziser
 - ▶ ε ist KKA.
 - ▶ Wenn w_1, w_2 KKA sind, dann auch $w_1 w_2$.
 - ▶ Wenn w KKA ist, dann auch (w) .
 - ▶ Nichts anderes ist KKA.

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

Kontextfreie Grammatiken

Rekursive Definition syntaktischer Strukturen

Kontextfreie Grammatiken

Relationen (Teil 2)

- ▶ 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 $(x, z) \in M_1 \times M_3$

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

- ▶ \circ ist assoziative Operation

$$(R \circ S) \circ T = R \circ (S \circ T)$$

- ▶ die identische Abbildung auf M als Relation I_M

$$I_M = \{(x, x) \mid x \in M\}$$

- ▶ I_M neutrales Element bzgl. \circ

$$R \circ I_M = R = I_M \circ R$$

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

$$\begin{aligned} R^0 &= I_M \\ \forall i \in \mathbb{N}_0 : R^{i+1} &= R^i \circ R \end{aligned}$$

- ▶ 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 $I_M \subseteq R$ ist.
- ▶ Relation R heißt *transitiv*, wenn $R \circ R \subseteq R$, also wenn:

$$\forall x \in M : \forall y \in M : \forall z \in M : xRy \wedge yRz \implies xRz$$

(Das ist ein Implikationspfeil und kein Ableitungspfeil.)

- ▶ R^* ist immer reflexiv, denn
 $I_M = R^0 \subseteq R^*$
- ▶ 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 $R^* \subseteq S$.

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

- ▶ 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