

# Einführung in die Theoretische Informatik

Woche 9

Harald Zankl

Institut für Informatik © UIBK  
Wintersemester 2014/2015



## Zusammenfassung der letzten LV

Definition (Deterministischer endlicher Automat (kurz: DEA))

Ein **DEA** ist ein 5-Tupel  $A = (Q, \Sigma, \delta, s, F)$  sodass

- 1  $Q$  eine endliche Menge von **Zuständen**
- 2  $\Sigma$  eine endliche Menge von **Eingabesymbolen**
- 3  $\delta: Q \times \Sigma \rightarrow Q$  die **Übergangsfunktion**
- 4  $s \in Q$  der **Startzustand**
- 5  $F \subseteq Q$  eine endliche Menge von **akzeptierenden Zuständen**

Zu beachten:  $\delta$  muss für alle möglichen Argumente definiert sein

Satz

*Für jeden DEA  $A$  ist  $L(A)$  regulär. Umgekehrt existiert zu jeder regulären Sprache  $L$  ein DEA  $A$ , sodass  $L = L(A)$ .*

# Inhalte der Lehrveranstaltung

## Einführung in die Logik

Syntax & Semantik der Aussagenlogik, Formales Beweisen, Konjunktive und Disjunktive Normalformen

## Einführung in die Algebra

Boolesche Algebra, Universelle Algebra, Logische Schaltkreise

## Einführung in die Theorie der Formalen Sprachen

Grammatiken und Formale Sprachen, Reguläre Sprachen, **Kontextfreie Sprachen**

## Einführung in die Berechenbarkeitstheorie

Algorithmisch unlösbare Probleme, Turing Maschinen, Registermaschinen

## Einführung in die Programmverifikation

Prinzipien der Analyse von Programmen, Verifikation nach Hoare, Verschlüsselung und Sicherheit

## Beispiel

Induktive Definition von Palindromen über  $\Sigma := \{0, 1\}$ .

- 1  $\epsilon, 0, 1$  sind Palindrome
- 2 Wenn  $x$  ein Palindrom ist, dann sind auch die Wörter

$$0x0 \quad 1x1$$

Palindrome

Wir betrachten die folgenden Regeln  $R$ :

$$P \rightarrow \epsilon \mid 0 \mid 1$$

$$P \rightarrow 0P0 \mid 1P1$$

- Die Grammatik  $G_1 = (\{P\}, \Sigma, R, P)$  beschreibt die Sprache der Palindrome, d.h.  $L(G_1)$  ist genau die Menge der Palindrome über  $\Sigma$
- $G_1$  ist kontextfrei

## Beweis.

Es ist zu zeigen, dass  $x \in L(G_1)$  gdw.  $x$  ein Palindrom ist

Wir zeigen nur: Wenn  $x \in L(G_1)$ , dann ist  $x$  ein Palindrom. Dazu verwenden wir Induktion nach der Länge  $\ell$  der Ableitung  $P \xRightarrow{*} x$

**1 Basis**  $\ell = 1$ : Also gilt einer der folgenden 3 Fälle:

- $x = \epsilon$
- $x = 0$
- $x = 1$

In allen Fällen:  $x$  ist Palindrom

**2 Schritt**  $\ell > 1$ : Also hat die Ableitung eine der folgenden Gestalten:

- $P \Rightarrow 0P0 \xRightarrow{*} 0y0 = x$
- $P \Rightarrow 1P1 \xRightarrow{*} 1y1 = x$

wobei  $y \in \Sigma^*$

Somit gilt  $P \xRightarrow{*} y$ . Aus der Induktionshypothese folgt, dass  $y$  ein Palindrom ist. In beiden Fällen ist dann aber auch  $x$  ein Palindrom. ■

## Frage

Stimmt dieser Beweis wirklich? Im Beweis schließen wir nämlich, dass aus

$$P \Rightarrow OP0 \xRightarrow{*} Oy0 = x$$

auch  $P \xRightarrow{*} y$  folgt

## Beispiel

Betrachte die (kontextsensitive) Grammatik

$G = (\{S, C, B, H\}, \{a, b\}, R, S)$  mit folgender Regel in  $R$ :

$$CB \rightarrow HB$$

Dann gilt etwa  $BCB \xRightarrow{*} BHB$ , aber sicher nicht  $C \xRightarrow{*} H$

## Antwort

Der Beweis stimmt, aber nur für **kontextfreie** Grammatiken

## Satz

Sei  $G = (V, \Sigma, R, S)$  eine kontextfreie Grammatik.

Wenn  $X_1 X_2 \dots X_n \xRightarrow{*} x$  mit  $X_i \in V \cup \Sigma$  und  $x \in \Sigma^*$ , dann kann man  $x$  in die Stücke  $x_1, x_2, \dots, x_n$  zerlegen, sodass  $X_i \xRightarrow{*} x_i$  für alle  $1 \leq i \leq n$ .

## Hilfsüberlegung

- Sei  $i < j$  und betrachte:  $X_1 X_2 \dots X_i \dots X_j \dots X_n \xRightarrow{*} x$
- Dann sind in  $x$  alle aus  $X_i$  abgeleiteten Satzformen links von den aus  $X_j$  abgeleiteten zu finden

## Beweis (des Satzes).

- 1 Wenn  $X_i \in \Sigma$ , dann  $X_i = x_i$  und offensichtlich  $X_i \xRightarrow{*} x_i$
- 2 Wenn  $X_i \in V$ , dann erhalten wir  $X_i \xRightarrow{*} x_i$  aus  $X_1 X_2 \dots X_n \xRightarrow{*} x$ , indem
  - Ableitungen ausgehend von  $X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n$  ignoriert
  - Ableitungen ausgehend von  $X_i$  simuliert werden



## Lemma

Sei  $G = (V, \Sigma, R, S)$  eine Grammatik,  $A \in V$  und  $A \xrightarrow[G]{*} x$ . Dann gilt für alle  $u, v \in (V \cup \Sigma)^*$  auch  $uAv \xrightarrow[G]{*} uxv$ .

## Beweis.

Angenommen es existieren Wörter  $w_0, \dots, w_k \in (V \cup \Sigma)^*$ , sodass

$$A = w_0 \Rightarrow w_1 \Rightarrow \dots \Rightarrow w_{k-1} \Rightarrow w_k = x$$

Wir argumentieren informell:

- Wir betrachten den Schritt  $w_i \Rightarrow w_{i+1}$  für  $i \in \{0, \dots, k-1\}$
- Nach Definition gilt auch  $uw_i v \Rightarrow uw_{i+1} v$

Somit folgt

$$uAv = uw_0 v \Rightarrow uw_1 v \Rightarrow \dots \Rightarrow uw_{k-1} v \Rightarrow uw_k v = uxv$$





## Definition

Sei  $G$  eine kontextfreie Grammatik.

- Eine **Linksableitung** ist eine Ableitung sodass immer die am weitesten links stehende Variable ersetzt wird
- In einer **Rechtsableitung** wird immer die am weitesten rechts stehende Variable ersetzt

$$\Rightarrow_{\ell}, \overset{*}{\Rightarrow}_{\ell}$$

$$\Rightarrow_r, \overset{*}{\Rightarrow}_r$$

## Beispiel

Sei  $G_2 = (\{S\}, \{(, )\}, R, S)$  eine (kontextfreie) Grammatik mit Regeln  $R$ :

$$S \rightarrow \epsilon \mid (S) \mid SS$$

Dann gilt  $S \underset{r}{\Rightarrow} SS \underset{r}{\Rightarrow} S(S) \underset{r}{\Rightarrow} S() \underset{r}{\Rightarrow} (S)() \underset{r}{\Rightarrow} ()()$

## Definition (Eindeutigkeit einer Grammatik)

Eine kontextfreie Grammatik  $G$  heißt **eindeutig**, wenn jedes Wort  $x \in L(G)$  genau eine Linksableitung besitzt, ansonsten **mehrdeutig**

## Definition (rekursive Inferenz)

Sei  $G = (V, \Sigma, R, S)$  eine kontextfreie Grammatik. Für eine Regel  $A \rightarrow X_1 \dots X_n$  mit  $X_i \in V \cup \Sigma$  definieren wir  $L(A)$  induktiv:

- 1 Wenn  $X_1 \dots X_n \in \Sigma^*$ , dann  $X_1 \dots X_n \in L(A)$ .
- 2 Wenn  $x_i \in L(X_i)$  oder  $x_i = X_i \in \Sigma$ , dann  $x_1 x_2 \dots x_n \in L(A)$ .

## Beispiel

Sei  $G_3 = (\{E, T\}, \{+, \cdot, (, ), a, b, 0, 1\}, R, E)$  eine Grammatik mit Regeln

$$E \rightarrow T \mid E + E \mid E \cdot E \mid (E) \quad T \rightarrow a \mid b \mid Ta \mid Tb \mid T0 \mid T1$$

Wir zeigen  $(a + b10) \in L(E)$ :

Schritt	Wort	Variable	Regel	Rekursion
1	a	T	$T \rightarrow a$	
2	b	T	$T \rightarrow b$	
3	b1	T	$T \rightarrow T1$	2
4	b10	T	$T \rightarrow T0$	3
5	a	E	$E \rightarrow T$	1
6	b10	E	$E \rightarrow T$	4
7	a + b10	E	$E \rightarrow E + E$	5, 6
8	(a + b10)	E	$E \rightarrow (E)$	7

## Bemerkung

- Wir nennen  $L(A)$  die Sprache von  $A$ .
- Die rekursive Inferenz entspricht dem **bottom-up parsing** eines Compilers: zuerst wird der Rumpf einer Regel gelesen, dann wird das Ergebnis der Variable im Kopf übergeben
- Der Parsergenerator yacc generiert einen bottom-up parser

## Definition (Syntaxbaum)

Sei  $G = (V, \Sigma, R, S)$  eine kontextfreie Grammatik.

Ein **Syntaxbaum** für  $G$  ist ein Baum  $B$  sodass:

- 1 Jedes Blatt in  $B$  ist entweder:
  - ein Terminal aus  $\Sigma$
  - ein Nichtterminal aus  $V$ , oder
  - $\epsilon$

Im letzten Fall ist das Blatt das einzige Kind seines Vorgängers

- 2 Jeder innere Knoten von  $B$  ist eine Variable  $A \in V$  mit Kindern  $X_1, \dots, X_n$  ( $X_i \in V \cup \Sigma$ ), wobei:

$$A \rightarrow X_1 \cdots X_n \in R$$

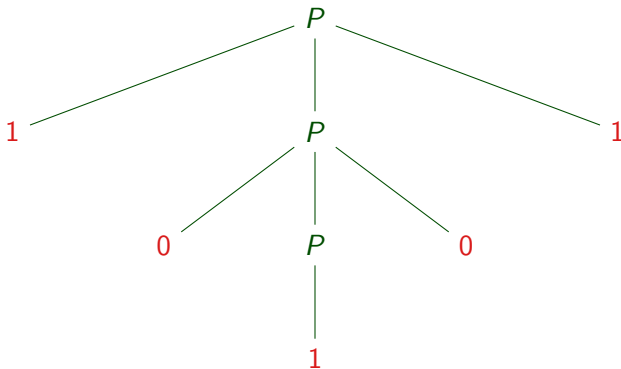
Das **Ergebnis** eines Syntaxbaums  $B$  für  $G$  ist das Wort über  $(V \cup \Sigma)^*$ , das wir erhalten, wenn wir die Blätter in  $B$  von links nach rechts lesen

## Beispiel

Wir betrachten die (kontextfreie) Grammatik  $G_1 = (\{P\}, \Sigma, R, P)$  mit:

$$P \rightarrow \epsilon \mid 0 \mid 1 \mid 0P0 \mid 1P1$$

Dann ist der folgende Baum ein Syntaxbaum für  $G_1$ :

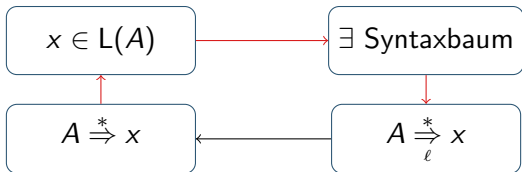


## Satz

Sei  $G = (V, \Sigma, R, S)$  eine kontextfreie Grammatik,  $A \in V$  und  $x \in \Sigma^*$ .  
Die folgenden Aussagen sind **äquivalent**:

- 1  $x \in L(A)$  nach dem rekursiven Inferenzverfahren
- 2  $A \xRightarrow{*} x$
- 3  $A \xRightarrow[\ell]{*} x$
- 4  $A \xRightarrow[r]{*} x$
- 5 Es existiert ein Syntaxbaum für  $G$  mit Wurzel  $A$  und Ergebnis  $x$

Beweisidee:



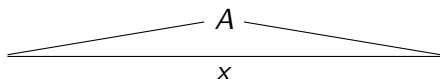
## Satz

Wenn  $x \in L(A)$  nach dem rekursiven Inferenzverfahren, dann gibt es einen Syntaxbaum mit Wurzel  $A$  und Ergebnis  $x$ .

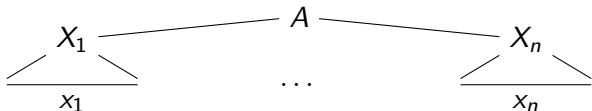
## Beweis.

Mit Induktion nach der Anzahl der Rekursionen im Inferenzverfahren

- 1 **Basis**  $x \in L(A)$  benutzt genau die Regel  $A \rightarrow x \in R$   
Dann gibt es Syntaxbaum mit Wurzel  $A$ :



- 2 **Schritt**  $x \in L(A)$  benutzt  $A \rightarrow X_1 \cdots X_n$  ( $X_i \in V \cup \Sigma$ )  
Dann gibt es Syntaxbaum mit Wurzel  $A$ :



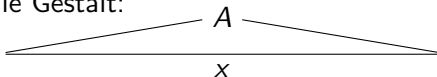
## Satz

Sei  $B$  ein Syntaxbaum mit Wurzel  $A$  und Ergebnis  $x$ , dann gibt es eine Linksableitung (eine Rechtsableitung) von  $x$  aus  $A$ .

## Beweis.

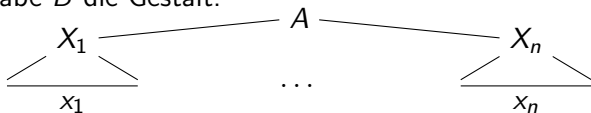
Mit Induktion nach der Höhe des Syntaxbaums

- 1 **Basis** Habe  $B$  die Gestalt:



dann existiert  $A \rightarrow x \in R$ , also  $A \xRightarrow{\ell} x$

- 2 **Schritt** Habe  $B$  die Gestalt:



dann existiert  $A \rightarrow X_1 \cdots X_n \in R$ , also:

$$A \xRightarrow{\ell} X_1 X_2 \cdots X_n \xRightarrow{\ell}^* x_1 X_2 \cdots X_n \xRightarrow{\ell}^* x_1 x_2 \cdots X_n \xRightarrow{\ell}^* x_1 x_2 \cdots x_n$$





## Satz

Sei  $G = (V, \Sigma, R, S)$  eine kontextfreie Grammatik,  $A \in V$  und  $x \in \Sigma^*$ .  
 Wenn  $A \xRightarrow{*} x$ , dann liefert das rekursive Inferenzverfahren, dass  $x \in L(A)$ .

## Beweis.

Mit Induktion nach der Länge  $\ell$  der Ableitung  $A \xRightarrow{*} x$ :

**1 Basis** Sei  $\ell = 1$ , dann gilt  $x \in L(A)$

**2 Schritt** Angenommen  $\ell = \ell' + 1$

Zunächst können wir die Ableitung  $A \xRightarrow{*} x$  wie folgt schreiben:

$$A \Rightarrow X_1 X_2 \cdots X_n \xRightarrow{*} x = x_1 x_2 \cdots x_n$$

Wir verwenden, dass wir die Ableitungen der Sätze  $x_i$  aufbrechen können, also gilt:

- Wenn  $X_i \in \Sigma$ , dann  $X_i = x_i$
- Wenn  $X_i \in V$ , dann gilt  $X_i \xRightarrow{*} x_i$  und somit  $x_i \in L(X_i)$

