

Grundlagen der Theoretischen Informatik

4. Kellerautomaten und kontextfreie Sprachen (I)

25.05.2016

Viorica Sofronie-Stokkermans

e-mail: sofronie@uni-koblenz.de

Organisatorisches

- **1. Teilklausur:** Mittwoch, 8.06.2016, D028, 14:30-15:30
Anmeldung über MeeToo bis 6.06.2016 um 8:00 (a.m.)
- Wer am 8.06 **nicht angemeldet** ist, kann an der Teilklausur **nicht teilnehmen**.
- Wenn Sie an der Teilklausur **nicht** teilnehmen möchten, melden Sie sich bitte ab.

- **2. Teilklausur:** Freitag, 28.07.2016, D-028, **14:00-15:00**
E-Mail in KLIPS.

Nachklausur: Ende des Sommersemesters

Organisatorisches

- **1. Teilklausur:** Mittwoch, 8.06.2016, D028, 14:30-15:30

Question/Answer session:

Donnerstag, 2.06.2016, in der Vorlesung

Bis jetzt

1. Motivation
2. Terminologie
3. Endliche Automaten und reguläre Sprachen
4. Kellerautomaten und kontextfreie Sprachen
5. Turingmaschinen und rekursiv aufzählbare Sprachen
6. Berechenbarkeit, (Un-)Entscheidbarkeit
7. Komplexitätsklassen P und NP

Heute

1. Motivation
2. Terminologie
3. Endliche Automaten und reguläre Sprachen
4. Kellerautomaten und kontextfreie Sprachen
5. Turingmaschinen und rekursiv aufzählbare Sprachen
6. Berechenbarkeit, (Un-)Entscheidbarkeit
7. Komplexitätsklassen P und NP

Kellerautomaten und kontextfreie Sprache

Inhalt

- Die von **Kellerautomaten** (**Push-Down-Automaten**, **PDA**s) erkannten Sprachen sind genau die vom Typ 2 (**kontextfrei**).
- **Normalformen** für kontextfreie Grammatiken.
- **Pumping-Lemma** für kontextfreie Sprachen.
- Effiziente Algorithmen für **Probleme über PDA**s

Ableitungsbäume

Zur Erinnerung: kontextfreie Grammatiken

- Kontextfreie Regel:
Eine Variable wird durch ein Wort ersetzt,
(egal in welchem Kontext die Variable steht)
- Es wird eine **einzelne** Variable ersetzt.
- Das Wort in der Conclusio kann Variablen und Terminale in **beliebiger Mischung** enthalten.

Zur Erinnerung: kontextfreie Grammatiken

Definition (Kontextfreie Grammatik)

Eine Grammatik $G = (V, T, R, S)$ heißt **kontextfrei** gdw

$$\forall (P \rightarrow Q) \in R \quad (P \in V \text{ und } Q \in (V \cup T)^*)$$

Das heißt, bei jeder Regelanwendung:

- Links eine **einzelne Variable**
- Die Prämisse macht keine Aussage, was der Kontext dieser Variablen ist („kontextfrei“)
- Rechts steht etwas beliebiges

Zur Erinnerung: kontextfreie Sprachen

Beispiel.

- $\{a^n b^n \mid n \in \mathbb{N}\}$
- $\{a^n b a^n \mid n \in \mathbb{N}\}$
- $\{ww^R \mid w \in \{a, b\}^*\}$

Ableitungsbäume

Definition (Ableitungsbaum zu einer Grammatik)

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

Ein **Ableitungsbaum (parse tree)** zu G ist ein angeordneter Baum $B = (W, E, v_0)$

Zudem muss gelten:

- Jeder Knoten $v \in W$ ist mit einem Symbol aus $V \cup T \cup \{\varepsilon\}$ markiert.
- Die Wurzel v_0 ist mit S markiert.
- Jeder innere Knoten ist mit einer Variablen aus V markiert.
- Jedes Blatt ist mit einem Symbol aus $T \cup \{\varepsilon\}$ markiert.
- Ist $v \in W$ ein innerer Knoten mit Söhnen v_1, \dots, v_k in dieser Anordnung und ist A die Markierung von v und A_i die Markierung von v_i , dann ist $A \rightarrow A_1 \dots A_k \in R$.
- Ein mit ε markiertes Blatt hat keinen Bruder (denn das entspräche einer Ableitung wie $A \rightarrow ab\varepsilon Bc$).

Ableitungsbäume

Ablezen eines Wortes vom Ableitungsbaum

Wenn Wort w von Grammatik G erzeugt wird, dann gibt es einen Ableitungsbaum mit den Buchstaben von w als Blätter von links nach rechts.

Merke:

Die Blätter eines Ableitungsbaumes sind angeordnet.

Es gibt eine Ordnung unter den Söhnen eines Knotens.

Ableitungsbäum

Definition

Seien b_1, b_2 Knoten. Dann:

$b_1 < b_2$ gdw b_1, b_2 sind Brüder, und b_1 liegt "links" von b_2 ,
oder $\exists v, v_1, v_2 \in W \quad v \rightarrow v_1, v \rightarrow v_2, v_1 < v_2$
und v_i ist Vorfahre von b_i für $i \in \{1, 2\}$.

Definition

Sei $\{b_1, \dots, b_k\}$ die Menge aller Blätter in B mit $b_1 < \dots < b_k$, und sei A_i die Markierung von b_i .

Dann heißt das Wort $A_1 \dots A_k$ die **Front** von B .

Ableitungsbäume

Theorem.

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

Dann gilt für $w \in T^*$:

$(S \Longrightarrow_G^* w)$ gdw Es existiert ein Ableitungsbaum zu G mit Front w .

Beweis. Einfach aus den Definitionen.

Ableitungsbäume: Beispiel

Grammatik für die Menge aller aussagenlogischen Formeln über den Variablen $\{x, x_0, x_1, x_2, \dots\}$:

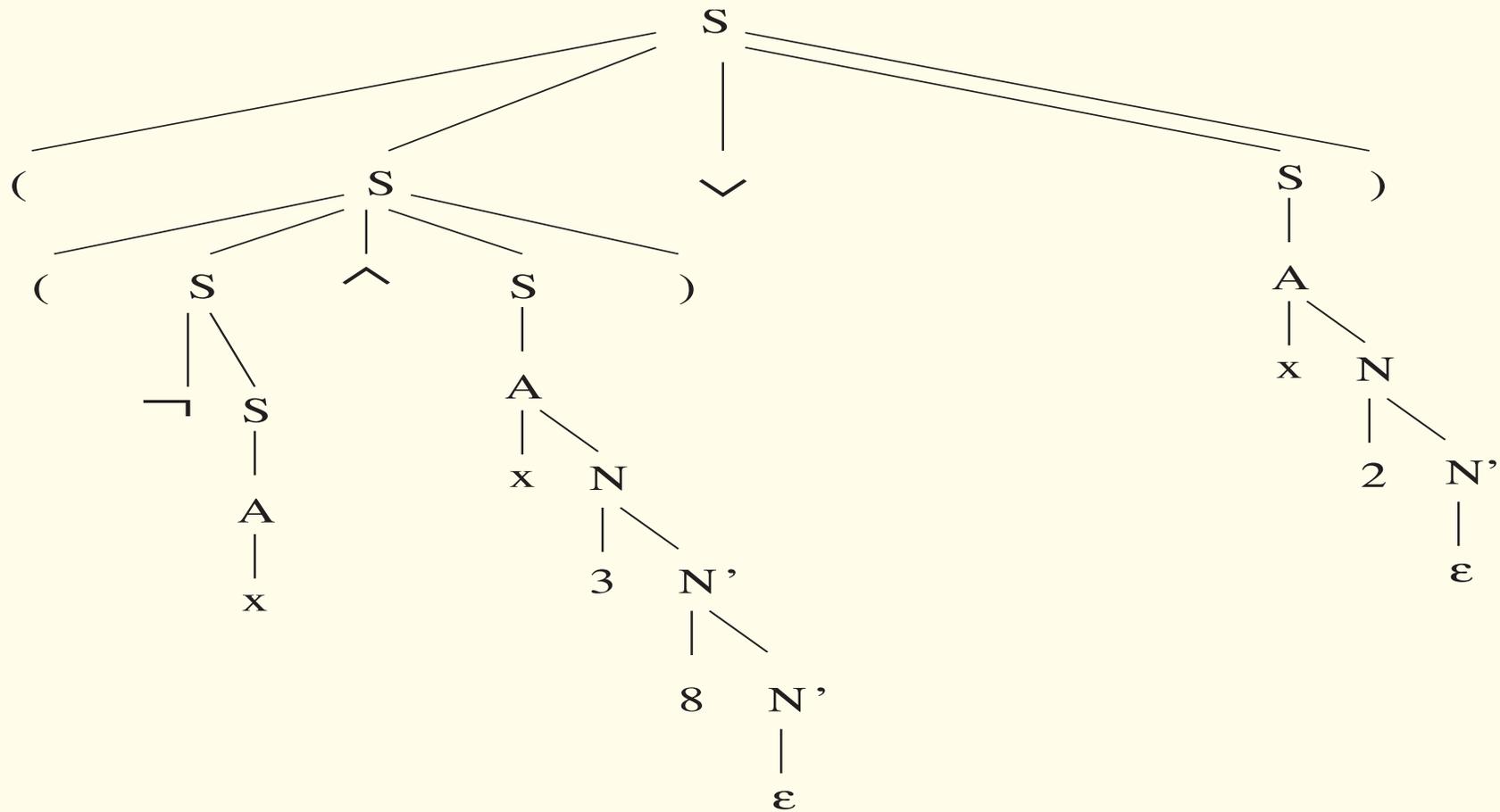
$$G = (\{S, A, N, N'\}, \{x, 0, \dots, 9, (,), \wedge, \vee, \neg\}, R, S)$$

mit der Regelmenge

$$\begin{aligned} R = \{ & S \rightarrow (S \wedge S) \mid (S \vee S) \mid \neg S \mid A \\ & A \rightarrow x \mid xN \\ & N \rightarrow 1N' \mid 2N' \mid \dots \mid 9N' \mid 0 \\ & N' \rightarrow 0N' \mid 1N' \mid \dots \mid 9N' \mid \varepsilon \} \end{aligned}$$

Ableitungsbäume: Beispiel

Ableitungsbaum für $((\neg x \wedge x38) \vee x2)$



Ableitungsbäume: Beispiel

Der Ableitungsbaum steht für viele **äquivalente** Ableitungen, darunter diese:

$$\begin{array}{lclcl}
 S & & (S \vee S) & \Rightarrow & \\
 ((S \wedge S) \vee S) & \Rightarrow & ((\neg S \wedge S) \vee S) & \Rightarrow & \\
 ((\neg A \wedge S) \vee S) & \Rightarrow & ((\neg x \wedge S) \vee S) & \Rightarrow & \\
 ((\neg x \wedge A) \vee S) & \Rightarrow & ((\neg x \wedge xN) \vee S) & \Rightarrow & \\
 ((\neg x \wedge x3N') \vee S) & \Rightarrow & ((\neg x \wedge x38N') \vee S) & \Rightarrow & \\
 ((\neg x \wedge x38) \vee S) & \Rightarrow & ((\neg x \wedge x38) \vee A) & \Rightarrow & \\
 ((\neg x \wedge x38) \vee xN) & \Rightarrow & ((\neg x \wedge x38) \vee x2N') & \Rightarrow & \\
 ((\neg x \wedge x38) \vee x2) & & & &
 \end{array}$$

Links- und Rechtsableitung

Definition (Linksableitung)

Eine Ableitung

$$w_1 \Longrightarrow_G w_2 \Longrightarrow_G \dots \Longrightarrow_G w_n$$

heißt **Linksableitung** falls w_{i+1} durch Ersetzen der linkesten Variable in w_i entsteht für alle $i < n$.

Die **Rechtsableitung** ist analog definiert.

Mehrdeutigkeit

Definition (Mehrdeutigkeit)

Eine cf-Grammatik G heißt **mehrdeutig**

gdw

es gibt ein Wort $w \in L(G)$,

zu dem es in G **zwei verschiedene Linksableitungen** gibt.

Eine **Sprache** $L \in \mathbf{L}_2$ heißt **inhärent mehrdeutig**

gdw

alle kontextfreien Grammatiken für L sind mehrdeutig.

Bemerkung

Eine Grammatik G ist mehrdeutig, gdw :

es gibt zwei verschiedene Ableitungsbäume in G mit gleicher Front.

Mehrdeutigkeit: Beispiele

Eindeutige Grammatik für aussagenlogische Formeln:

$$S \rightarrow (S \wedge S) \mid (S \vee S) \mid \neg S \mid A$$

$$A \rightarrow x \mid xN$$

$$N \rightarrow 1N' \mid 2N' \mid \dots \mid 9N' \mid 0$$

$$N' \rightarrow 0N' \mid 1N' \mid \dots \mid 9N' \mid \varepsilon$$

Mehrdeutigkeit: Beispiele

Eindeutige Grammatik für aussagenlogische Formeln:

$$S \rightarrow (S \wedge S) \mid (S \vee S) \mid \neg S \mid A$$

$$A \rightarrow x \mid xN$$

$$N \rightarrow 1N' \mid 2N' \mid \dots \mid 9N' \mid 0$$

$$N' \rightarrow 0N' \mid 1N' \mid \dots \mid 9N' \mid \varepsilon$$

Mehrdeutige Grammatik für aussagenlogische Formeln in KNF:

$$K \rightarrow K \wedge K \mid D \quad \text{Regel mit Klammer-Ersparnis!}$$

$$D \rightarrow (D \vee D) \mid L$$

$$L \rightarrow \neg A \mid A$$

$$A \rightarrow v \mid w \mid x \mid y \mid z$$

Mehrdeutigkeit: Beispiele

Inhärente Mehrdeutigkeit

Die Sprache

$$L := \{a^i b^j c^k \mid i = j \text{ oder } j = k\}$$

ist **inhärent mehrdeutig**.

Für einen Beweis siehe [Wegener “Theoretische Informatik” 1993 S.168-170], cf. Beispiel 6.1.7 Erk, Priese “Theoretische Informatik”.

Umformung von Grammatiken

Startsymbol nur links

Einfache Annahme:

Im folgenden soll für alle cf-Grammatiken gelten:

Das Startsymbol S kommt nie auf einer rechten Regelseite vor.

Umformung

Ist das bei einer Grammatik nicht gegeben, kann man es wie folgt erreichen:

- Führe ein neues Startsymbol S_{neu} ein
- Füge die Regel

$$S_{neu} \rightarrow S$$

hinzu.

Nutzlose Symbole

Nutzlose Symbole und Regeln: Intuition

- Variablen und Symbole, die vom Startsymbol aus unerreichbar sind.
- Variablen, von denen aus kein Terminalwort abgeleitet werden kann.
- Regeln, die solche Variablen und Symbole enthalten

Nutzlose Symbole

Definition ((co-)erreichbare, nutzlose Symbole)

Sei $G = (V, T, R, S)$ eine Grammatik.

Ein Symbol $x \in (V \cup T)$ heißt

erreichbar: Es gibt $\alpha, \beta \in (V \cup T)^*$: $S \xRightarrow*_G \alpha x \beta$

co-erreichbar: Es gibt $w \in T^*$: $x \xRightarrow*_G w$

nutzlos: x ist nicht erreichbar oder nicht co-erreichbar.

Nutzlose Symbole

Theorem (cf-Grammatik ohne nutzlose Symbole)

Ist $G = (V, T, R, S)$ eine cf-Grammatik mit $L(G) \neq \emptyset$,
dann existiert eine cf-Grammatik $G' = (V', T', R', S')$ mit:

- G' ist äquivalent zu G .
- Jedes $x \in (V \cup T)$ ist erreichbar und co-erreichbar.

Nutzlose Symbole

Theorem (cf-Grammatik ohne nutzlose Symbole)

Ist $G = (V, T, R, S)$ eine cf-Grammatik mit $L(G) \neq \emptyset$,
dann existiert eine cf-Grammatik $G' = (V', T', R', S')$ mit:

- G' ist äquivalent zu G .
- Jedes $x \in (V \cup T)$ ist erreichbar und co-erreichbar.

Beweis

Man kann G' aus G effektiv konstruieren:

- Wie im folgenden beschrieben, die nutzlosen Symbole bestimmen.
- Diese Symbole und alle Regeln, die sie enthalten, entfernen.

Nutzlose Symbole

Algorithmus zur Berechnung der co-erreichbaren Variablen

Input: Grammatik $G = (V, T, R, S)$

Output: co-erreichbare Variablen

Alt := \emptyset

Neu := $\{A \in V \mid \exists w \in T^* (A \rightarrow w \in R)\}$

while Alt \neq Neu

{

 Alt := Neu

 Neu := Alt $\cup \{A \in V \mid \exists \alpha \in (T \cup \text{Alt})^* (A \rightarrow \alpha \in R)\}$

}

output Neu

Nutzlose Symbole

Bestimmung einer Grammatik $G'' = (V'', T'', R'', S)$ nur mit diesen co-erreichbaren Variablen.

if $S \in \text{Neu}$ $/* S \text{ ist co-erreichbar } */$

{

$V'' := \text{Neu}$

$T'' := T$

$R'' = R \cap (V'' \times (V'' \cup T'')^*)$

}

else $/* L(G) = \emptyset */$

Nutzlose Symbole

Bestimmung der Grammatik G' ohne nutzlose Symbole:

$G' = (V', T', R', S')$ mit:

$$V' := \text{Neu2} \cap V''$$

$$T' = \text{Neu2} \cap T$$

$$R' = R'' \cap (V' \times (V' \cup T')^*)$$

$$S' = S$$

Damit gilt dann: $L(G') = L(G)$ und G' enthält keine nutzlosen Symbole.