LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

Formale Sprachen und Komplexität
Sommersemester 2024

5b
Die Chomsky-Normalform

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik und Theorembeweisen

Stand: 14. Mai 2024
Basierend auf Folien von PD Dr. David Sabel

Kontextfreie Sprachen

Zusammenfassung:

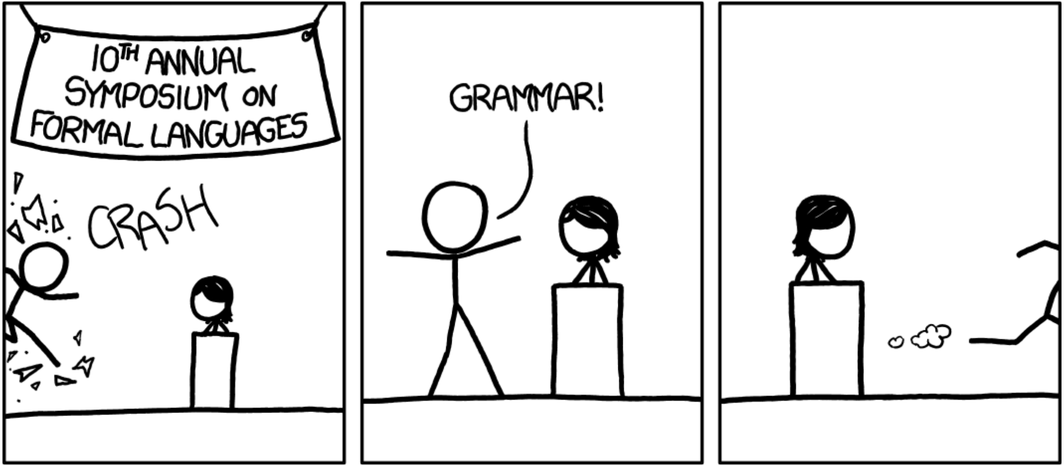
- Kontextfreie Sprachen (*context-free languages*, CFLs) werden von kontextfreien Grammatiken (*context-free grammars*, CFGs) erzeugt.
- Das sind die Typ 2-Sprachen bzw. die Typ 2-Grammatiken.
- Bedingung: Alle linken Seiten der Produktionen bestehen aus genau einer Variablen, d.h. die Produktionen sind von der Form $A \to r$.

Kontextfreie Sprachen

Zusammenfassung:

- Kontextfreie Sprachen (*context-free languages*, CFLs) werden von kontextfreien Grammatiken (*context-free grammars*, CFGs) erzeugt.
- Das sind die Typ 2-Sprachen bzw. die Typ 2-Grammatiken.
- Bedingung: Alle linken Seiten der Produktionen bestehen aus genau einer Variablen, d.h. die Produktionen sind von der Form $A \to r$.

Kontextfreie Sprachen sind insbesondere nützlich um Sprachen mit Klammerungen zu beschreiben, z.B. Programmiersprachen.



xkcd.com/1090/

Beispiele für kontextfreie Sprachen

Die CFG $G_1 = (\{S, T\}, \{a, b\}, P_1, S)$ mit

$$\begin{aligned} P_1 := \{S &\to \varepsilon \mid T, \\ T &\to aTb \mid ab\} \end{aligned}$$

erzeugt die Sprache $\{a^j b^j \mid j \in \mathbb{N}\}$, die daher kontextfrei ist.

Beispiele für kontextfreie Sprachen

Die CFG $G_1 = (\{S, T\}, \{a, b\}, P_1, S)$ mit

$$\begin{aligned} P_1 := \{S &\to \varepsilon \mid T, \\ T &\to aTb \mid ab\} \end{aligned}$$

erzeugt die Sprache $\{a^j b^j \mid j \in \mathbb{N}\}$, die daher kontextfrei ist.

Die CFG $G_2 = (\{E, M, A, Z, R\}, \{+, *, (,)\} \cup \{0, \ldots, 9\}, P_2, E)$ mit

$$\begin{aligned} P_2 := \{E &\to M \mid E + M, \\ M &\to A \mid M * A, \\ A &\to Z \mid (E), \\ Z &\to 1R \mid \ldots \mid 9R, \\ R &\to 0R \mid \ldots \mid 9R \mid \varepsilon\} \end{aligned}$$

erzeugt einfache arithmetische Ausdrücke nach der „Punkt vor Strich“-Regel.

Normalformen von Grammatiken

- Normalformen fordern eine spezielle Form der Produktionen.
- Sie sind nützlich, wenn man Grammatiken analysiert oder Algorithmen auf Grammatiken formuliert: Man muss dann nur diese Form (statt aller erlaubten) von Produktionen betrachten.
- Wir betrachten zwei Normalformen:
  die Chomsky-Normalform und die Greibach-Normalform.

Die Chomsky-Normalform

Definition

Eine CFG $G = (V, \Sigma, P, S)$ ist in Chomsky-Normalform, wenn für jede Produktion $A \to w \in P$ gilt: $w = a \in \Sigma$ oder $w = BC$ mit $B, C \in V$.

Die Chomsky-Normalform

Definition

Eine CFG $G = (V, \Sigma, P, S)$ ist in Chomsky-Normalform, wenn für jede Produktion $A \to w \in P$ gilt: $w = a \in \Sigma$ oder $w = BC$ mit $B, C \in V$.

Beispiele:

- Die CFG $G = (\{A\}, \{(,),[,]\}, \{A \to (A) \mid () \mid [A] \mid [] \mid AA\}, A)$ ist nicht in Chomsky-Normalform. Nur die Produktion $A \to AA$ passt zum vorgeschriebenen Format.

Die Chomsky-Normalform

Definition

Eine CFG $G = (V, \Sigma, P, S)$ ist in Chomsky-Normalform, wenn für jede Produktion $A \to w \in P$ gilt: $w = a \in \Sigma$ oder $w = BC$ mit $B, C \in V$.

Beispiele:

- Die CFG $G = (\{A\}, \{(,), [,]\}, \{A \to (A) \mid () \mid [A] \mid [] \mid AA\}, A)$ ist nicht in Chomsky-Normalform. Nur die Produktion $A \to AA$ passt zum vorgeschriebenen Format.
- Die CFG $G' = (\{A, B, C, D, E, F, G\}, \{(,), [,]\}, P, A)$ mit

$$
\begin{aligned}
P := \{ & A \to BF \mid BC \mid DG \mid DE \mid AA, \\
& B \to (, C \to), D \to [, E \to], F \to AC, G \to AE\}
\end{aligned}
$$

ist in Chomsky-Normalform, und erzeugt dieselbe Sprache wie G.

Eigenschaften der Chomsky-Normalform

Für CFG G in Chomsky-Normalform gilt, dass Ableitungen eines Worts $w \in L(G)$ immer genau aus $2 \cdot |w| - 1$ Ableitungsschritten bestehen.

Zudem sind die dazugehörigen Syntaxbäume immer Binärbäume (Bäume, wobei jeder Knoten 0 oder 2 Kinder hat), bis auf die letzte Schicht, die Produktionen $A \to a$ anwendet.

Eigenschaften der Chomsky-Normalform

Für CFG G in Chomsky-Normalform gilt, dass Ableitungen eines Worts $w \in L(G)$ immer genau aus $2 \cdot |w| - 1$ Ableitungsschritten bestehen.

Zudem sind die dazugehörigen Syntaxbäume immer Binärbäume (Bäume, wobei jeder Knoten 0 oder 2 Kinder hat), bis auf die letzte Schicht, die Produktionen $A \to a$ anwendet.

Sei die CFG $G' = (\{A, B, C, D, E, F, G\}, \{(,), [,]\}, P, A)$ mit

$$\begin{aligned} P := \{A &\to BF \mid BC \mid DG \mid DE \mid AA, \\ &B \to (, C \to), D \to [, E \to], F \to AC, G \to AE\} \end{aligned}$$

Eigenschaften der Chomsky-Normalform

Für CFG G in Chomsky-Normalform gilt, dass Ableitungen eines Worts $w \in L(G)$ immer genau aus $2 \cdot |w| - 1$ Ableitungsschritten bestehen.

Zudem sind die dazugehörigen Syntaxbäume immer Binärbäume (Bäume, wobei jeder Knoten 0 oder 2 Kinder hat), bis auf die letzte Schicht, die Produktionen $A \to a$ anwendet.

Sei die CFG $G' = (\{A, B, C, D, E, F, G\}, \{(,), [,]\}, P, A)$ mit

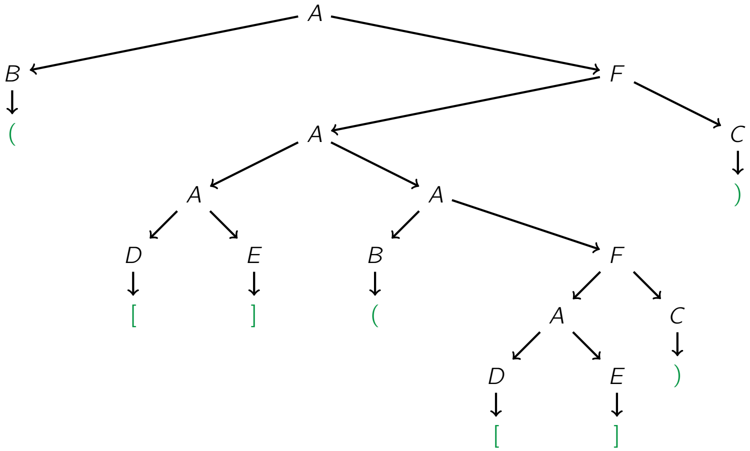
$$\begin{aligned} P := \{&A \to BF \mid BC \mid DG \mid DE \mid AA, \\ &B \to (,\ C \to),\ D \to [,\ E \to],\ F \to AC,\ G \to AE\} \end{aligned}$$

Ableitung von ([]([])):

$$\begin{aligned} A &\Rightarrow BF \Rightarrow (F \Rightarrow (AC \Rightarrow (AAC \Rightarrow (DEAC \Rightarrow ([EAC \\ &\Rightarrow ([]AC \Rightarrow ([]BFC \Rightarrow ([](FC \Rightarrow ([](ACC \Rightarrow ([](DECC \\ &\Rightarrow ([]([ECC \Rightarrow ([]([]CC \Rightarrow ([]([])C \Rightarrow ([]([])) \end{aligned}$$

Eigenschaften der Chomsky-Normalform

Der Syntaxbaum zur vorherigen Ableitung:



Herstellen der Chomsky-Normalform

Theorem

Für jede CFG G mit $\varepsilon \notin L(G)$ kann eine CFG G' in Chomsky-Normalform berechnet werden, sodass $L(G') = L(G)$ gilt.

Herstellen der Chomsky-Normalform

Theorem

Für jede CFG G mit $\varepsilon \notin L(G)$ kann eine CFG G' in Chomsky-Normalform berechnet werden, sodass $L(G') = L(G)$ gilt.

Grundgedanke der Prozedur:

1. Entferne ε-Produktionen (wie in früherer Vorlesung).
2. Entferne Produktionen von der Form $A \to B$ („Entfernen von Einheitsproduktionen").
3. Ersetze alle Terminale a in rechten Seiten, die nicht nur aus a bestehen durch neue Variablen A und füge Produktionen $A \to a$ hinzu.
4. Zerlege alle Produktionen von der Form $A \to B_1 \cdots B_m$ mit $m > 2$ in mehrere: $A \to B_1C_1, C_1 \to B_2C_3, \ldots, C_{m-2} \to B_{m-1}B_m$.

Beispiel für die Herstellung der Chomsky-Normalform

0. Eingabe: CFG $G = (\{A\}, \{(,), [,]\}, \{A \to (A) \mid () \mid [A] \mid [] \mid AA\}, A)$.

Beispiel für die Herstellung der Chomsky-Normalform

0. Eingabe: CFG $G = (\{A\}, \{(,), [,]\}, \{A \rightarrow (A) \mid () \mid [A] \mid [] \mid AA\}, A)$.
1. Es gibt keine ε-Produktionen zu entfernen.

Beispiel für die Herstellung der Chomsky-Normalform

0. Eingabe: CFG $G = (\{A\}, \{(,), [,]\}, \{A \to (A) \mid () \mid [A] \mid [] \mid AA\}, A)$.
1. Es gibt keine ε-Produktionen zu entfernen.
2. Es gibt keine Einheitsproduktionen zu entfernen.

Beispiel für die Herstellung der Chomsky-Normalform

0. Eingabe: CFG $G = (\{A\}, \{(,), [,]\}, \{A \to (A) \mid () \mid [A] \mid [] \mid AA\}, A)$.
1. Es gibt keine ε-Produktionen zu entfernen.
2. Es gibt keine Einheitsproduktionen zu entfernen.
3. Das Ersetzen von Terminale CFG führt zu $G_3 = (\{A, B, C, D, E\}, \{(,), [,]\}, P_3, A)$ mit

$$P_3 := \{A \to BAC \mid BC \mid DAE \mid DE \mid AA \\ B \to (,\ C \to),\ D \to [,\ E \to]\}$$

Beispiel für die Herstellung der Chomsky-Normalform

0. Eingabe: CFG $G = (\{A\}, \{(,), [,]\}, \{A \to (A) \mid () \mid [A] \mid [] \mid AA\}, A)$.
1. Es gibt keine ε-Produktionen zu entfernen.
2. Es gibt keine Einheitsproduktionen zu entfernen.
3. Das Ersetzen von Terminale CFG führt zu
   $G_3 = (\{A, B, C, D, E\}, \{(,), [,]\}, P_3, A)$ mit

$$\begin{aligned} P_3 := \{A \to BAC \mid BC \mid DAE \mid DE \mid AA\} \\ B \to (, C \to), D \to [, E \to]\} \end{aligned}$$

4. Die Zerlegung von langen rechten Seiten führt zu
   $G' = (\{A, B, C, D, E, F, G\}, \{(,), [,]\}, P', A)$ mit

$$\begin{aligned} P' := \{A \to BF \mid BC \mid DG \mid DE \mid AA, \\ B \to (, C \to), D \to [, E \to], F \to AC, G \to AE\} \end{aligned}$$

Entfernen von Einheitsproduktionen

Intuitiv ist klar, dass $A \to B$ entfernt werden kann:
Wenn erst $A \to B$, dann $B \to w$ angewendet wird,
kann man auch gleich $A \to w$ anwenden.

Entfernen von Einheitsproduktionen

Intuitiv ist klar, dass $A \to B$ entfernt werden kann:
Wenn erst $A \to B$, dann $B \to w$ angewendet wird,
kann man auch gleich $A \to w$ anwenden.

Algorithmisch zu beachten:

- Eliminiere in der richtigen Reihenfolge: Wenn $A \to B$ und $B \to C$, dann ist Ersetzen von $A \to B$ durch $A \to C$ nicht zielführend.
- Zyklen $A_1 \to A_2, A_2 \to A_3, \ldots, A_n \to A_1$ müssen vorher entfernt werden.

Algorithmus 5: Entfernen von Einheitsproduktionen

Eingabe: Eine CFG $G = (V, \Sigma, P, S)$
Beginn

Erzeuge gerichteten Graph $D = (V, E)$, mit $(A, B) \in E$ für jede Einheitsproduktion $A \to B \in P$;
solange *es einen Zyklus* $(A_1, A_2), \dots, (A_{n-1}, A_n), (A_n, A_1) \in E$ *gibt* **tue**
  $P := P \setminus \{A_1 \to A_2, \dots, A_{n-1} \to A_n, A_n \to A_1\}$; /* *entferne zyklische Regeln* */
  $P := P[A_1/A_2, \dots, A_1/A_n]$; /* *ersetze alle Vorkommen von* A_i *durch* A_1 *für* $i = 2, \dots, n$ */
  $V := V \setminus \{A_2, A_3, \dots, A_n\}$; /* *lösche* $A_2, \dots, A_n$ */
  $S := S[A_1/A_2, \dots, A_1/A_n]$; /* *ersetze Startsymbol durch* A_1, *falls es* A_i, $2 \leq i \leq n$ *war* */
  $E := E \setminus \{(A_1, A_2), \dots, (A_{n-1}, A_n), (A_n, A_1)\}$; /* *entferne Zyklus aus Graph* */
  $E := E[A_1/A_2, \dots, A_1/A_n]$; /* *ersetze* A_i *durch* A_1 *für* $i = 2, \dots, n$ *in anderen Kanten* */
Sortiere D topologisch und nummeriere die Variablen in V durch (und benenne entsprechend in E, P, S um), sodass gilt: $A_i \to A_j$ impliziert $i < j$;
Sei $V = \{A_1, \dots, A_k\}$;
für $i = k$ *bis* 1 **tue**
  wenn $A_i \to A_j \in P$ **dann**
    Seien $A_j \to w_1, \dots, A_j \to w_m$ alle Produktionen mit A_j als linker Seite;
    $P := P \cup \{A_i \to w_1, \dots, A_i \to w_m\}$; /* *füge* $A_i \to w_k$ *hinzu für* $A_i \to A_j$ *und* $A_j \to w_k$ */
    $P := P \setminus \{A_i \to A_j\}$; /* *entferne* $A_i \to A_j$ */
Gib die so entstandene Grammatik als G' aus;

Korrektheit von Algorithmus 5

Satz

Algorithmus 5 berechnet bei Eingabe einer CFG G mit $\varepsilon \notin L(G)$ eine CFG G', die keine Einheitsproduktionen hat, sodass gilt $L(G') = L(G)$.
Wenn G keine ε-Produktionen hat, dann hat auch G' keine ε-Produktionen.

Korrektheit von Algorithmus 5

Satz

Algorithmus 5 berechnet bei Eingabe einer CFG G mit $\varepsilon \notin L(G)$ eine CFG G', die keine Einheitsproduktionen hat, sodass gilt $L(G') = L(G)$.
Wenn G keine ε-Produktionen hat, dann hat auch G' keine ε-Produktionen.

Beweis Das Entfernen eines Zyklus ändert die erzeugte Sprache nicht.

Korrektheit von Algorithmus 5

Satz

Algorithmus 5 berechnet bei Eingabe einer CFG G mit $\varepsilon \notin L(G)$ eine CFG G', die keine Einheitsproduktionen hat, sodass gilt $L(G') = L(G)$.
Wenn G keine ε-Produktionen hat, dann hat auch G' keine ε-Produktionen.

Beweis Das Entfernen eines Zyklus ändert die erzeugte Sprache nicht.

Das Entfernen einer Einheitsproduktion $A_i \to A_j$ in der rückwärts-laufenden für-Schleife ändert die erzeugte Sprache nicht. Dies ist eine Instanz der Operation „Inlining von Produktionen", die wir gleich korrekt beweisen werden.

Korrektheit von Algorithmus 5

Beweis (Fortsetzung) Es werden keine Einheitsproduktionen eingeführt, da die Produktionen topologisch sortiert behandelt werden. Wenn $A_i \to A_j$ entfernt wird, wurden vorher alle Einheitsproduktionen $A_j \to A_k$ entfernt. D.h. zu diesem Zeitpunkt gilt: für alle Produktionen $A_j \to w$ besteht w nicht nur aus einer Variablen.

Korrektheit von Algorithmus 5

Beweis (Fortsetzung) Es werden keine Einheitsproduktionen eingeführt, da die Produktionen topologisch sortiert behandelt werden. Wenn $A_i \to A_j$ entfernt wird, wurden vorher alle Einheitsproduktionen $A_j \to A_k$ entfernt. D.h. zu diesem Zeitpunkt gilt: für alle Produktionen $A_j \to w$ besteht w nicht nur aus einer Variablen.

Es werden offensichtlich nie ε-Produktionen eingeführt.

Korrektheit von Algorithmus 5

Beweis (Fortsetzung) Es werden keine Einheitsproduktionen eingeführt, da die Produktionen topologisch sortiert behandelt werden. Wenn $A_i \to A_j$ entfernt wird, wurden vorher alle Einheitsproduktionen $A_j \to A_k$ entfernt. D.h. zu diesem Zeitpunkt gilt: für alle Produktionen $A_j \to w$ besteht w nicht nur aus einer Variablen.

Es werden offensichtlich nie ε-Produktionen eingeführt.

Der Algorithmus terminiert:

- Die solange-Schleife terminiert, da jede Iteration die Anzahl der Zyklen strikt verkleinert.
- Die für-Schleife terminiert offensichtlich. $\square$

Inlining von Produktionen

Lemma (Inlining von Produktionen)

Sei $G = (V, \Sigma, P, S)$ eine CFG mit

- $A \to uBv \in P$
- $B \to w_1 \mid \ldots \mid w_n$ alle Regeln mit B als linker Seite

und sei $G' = (V, \Sigma, (P \setminus \{A \to uBv\}) \cup \{A \to uw_1v \mid \ldots \mid uw_nv\}, S)$.
Dann gilt $L(G') = L(G)$.

Inlining von Produktionen

Beweis Das folgt durch Modifizieren der Syntaxbäume zur Ableitung mit G bzw. G':
Tausche alle Baumabschnitte

$$\begin{tikzcd}
 & A \arrow[dl] \arrow[d] \arrow[dr] & \\
u & B \arrow[d] & v \\
 & w_i &
\end{tikzcd}$$

im Syntaxbaum mit G durch

$$\begin{tikzcd}
 & A \arrow[dl] \arrow[d] \arrow[dr] & \\
u & w_i & v
\end{tikzcd}$$

im Syntaxbaum mit G'. □

Algorithmus 6: Herstellen der Chomsky-Normalform

Eingabe: CFG G mit $\varepsilon \notin L(G)$
Ausgabe: CFG G' in Chomsky-Normalform mit $L(G') = L(G)$
Beginn

Entferne die ε-Produktionen in G mit Algorithmus 1 und entferne anschließend die Einheitsproduktionen mit Algorithmus 5;
Sei $G' = (V', \Sigma, P', S')$ die entstandene Grammatik;
für alle $a \in \Sigma$ **tue**
  Sei B eine neue Variable;
  /* *Führe neue Variable B für a ein, und ersetze Vorkommen von a durch B* */
  $G' := (V' \cup \{B\}, \Sigma, \{A \to w[B/a] \mid A \to w \in P' \text{ und } |w| > 1\} \cup \{A \to w \mid A \to w \in P' \text{ und } |w| = 1\} \cup \{B \to a\}, S)$
/* *Nun sind alle Regeln von der Form $A \to a$ oder $A \to B_1 \cdots B_m$ mit $m \geq 2$* */
für alle $A \to B_1 \cdots B_m \in P'$ mit $m > 2$ **tue**
  Seien $C_1, \ldots, C_{m-2}$ neue Variablen;
  $V' := V' \cup \{C_1, \ldots, C_{m-2}\}$;
  /* *Ersetze in P' die Produktion $A \to B_1 \cdots B_m$ durch neue Regeln* */
  $P' := (P' \setminus \{A \to B_1 \cdots B_m\}) \cup \{A \to B_1 C_1\} \cup \{C_i \to B_{i+1} C_{i+1} \mid \text{für } i = 1, \ldots, m-3\} \cup \{C_{m-2} \to B_{m-1} B_m\}$;

Korrektheit von Algorithmus 6

Theorem

Für jede CFG G mit $\varepsilon \notin L(G)$ kann eine CFG G' in Chomsky-Normalform berechnet werden, sodass $L(G') = L(G)$ gilt.

Korrektheit von Algorithmus 6

Theorem

Für jede CFG G mit $\varepsilon \notin L(G)$ kann eine CFG G' in Chomsky-Normalform berechnet werden, sodass $L(G') = L(G)$ gilt.

Beweis Algorithmus 6 berechnet eine solche äquivalente CFG.

Korrektheit von Algorithmus 6

Theorem

Für jede CFG G mit $\varepsilon \notin L(G)$ kann eine CFG G' in Chomsky-Normalform berechnet werden, sodass $L(G') = L(G)$ gilt.

Beweis Algorithmus 6 berechnet eine solche äquivalente CFG.

Die Schritte „Entfernen von ε-Produktionen“ und „Entfernen von Einheitsproduktionen“ haben wir als korrekt gezeigt.

Korrektheit von Algorithmus 6

Theorem

Für jede CFG G mit $\varepsilon \notin L(G)$ kann eine CFG G' in Chomsky-Normalform berechnet werden, sodass $L(G') = L(G)$ gilt.

Beweis Algorithmus 6 berechnet eine solche äquivalente CFG.

Die Schritte „Entfernen von ε-Produktionen“ und „Entfernen von Einheitsproduktionen“ haben wir als korrekt gezeigt.

Die Schritte „Einführen von Produktionen $B \to a$“ und „Behandlung von $A \to B_1 \cdots B_m$“ sind Varianten der korrekten Operation „Boxing von Satzformen“, die wir gleich korrekt beweisen werden.

Korrektheit von Algorithmus 6

Theorem

Für jede CFG G mit $\varepsilon \notin L(G)$ kann eine CFG G' in Chomsky-Normalform berechnet werden, sodass $L(G') = L(G)$ gilt.

Beweis Algorithmus 6 berechnet eine solche äquivalente CFG.

Die Schritte „Entfernen von ε-Produktionen“ und „Entfernen von Einheitsproduktionen“ haben wir als korrekt gezeigt.

Die Schritte „Einführen von Produktionen $B \to a$“ und „Behandlung von $A \to B_1 \cdots B_m$“ sind Varianten der korrekten Operation „Boxing von Satzformen“, die wir gleich korrekt beweisen werden.

Am Ende der Berechnung haben alle Produktionen die gewünschte Form. □

Boxing von Satzformen

Lemma (Boxing von Satzformen)

Sei G eine CFG mit $G = (V, \Sigma, P \cup \{A \to w_1 \cdots w_n\}, S)$.
Seien $B_1, \ldots, B_n$ neue Variablen (d.h. $V \cap \{B_1, \ldots, B_n\} = \emptyset$) und
$G' = (V \cup \{B_1, \ldots, B_n\}, \Sigma, P \cup \{A \to B_1 \cdots B_n,\ B_1 \to w_1,\ \ldots,\ B_n \to w_n\}, S)$.
Dann gilt $L(G') = L(G)$.

Boxing von Satzformen

Lemma (Boxing von Satzformen)

Sei G eine CFG mit $G = (V, \Sigma, P \cup \{A \to w_1 \cdots w_n\}, S)$.
Seien $B_1, \ldots, B_n$ neue Variablen (d.h. $V \cap \{B_1, \ldots, B_n\} = \emptyset$) und
$G' = (V \cup \{B_1, \ldots, B_n\}, \Sigma, P \cup \{A \to B_1 \cdots B_n,\ B_1 \to w_1,\ \ldots,\ B_n \to w_n\}, S)$.
Dann gilt $L(G') = L(G)$.

Beweis

$\supseteq$ Wir zeigen: $w \in L(G)$ impliziert $w \in L(G')$.

Konstruiere aus $S \Rightarrow_G^* w$ die Ableitung $S \Rightarrow_{G'}^* w$: Übersetze jeden Schritt $uAv \Rightarrow_G uw_1 \cdots w_n v$ in $uAv \Rightarrow_{G'} uB_1 \cdots B_n v \Rightarrow_{G'}^n uw_1 \cdots w_n v$.

Boxing von Satzformen

Beweis (Fortsetzung)

$\subseteq$ Wir zeigen: $w \in L(G')$ impliziert $w \in L(G)$.

Betrachte den Syntaxbaum für $S \Rightarrow^*_{G'} w$.
Identifiziere die Anwendungen der Regeln $A \to B_1 \cdots B_n$ und $B_i \to w_i$, und modifiziere den Syntaxbaum durch Anwendung der Regel $A \to w_1 \cdots w_n$.
Lies Ableitung $S \Rightarrow^*_G w$ ab. $\square$

Weiteres Beispiel für die Herstellung der Chomsky-Normalform

Eingabe: $G_0 = (\{A, B, C, D, S\}, \{0, 1\}, P_0, S)$ mit
$P_0 = \{S \to 1A,\ A \to AB,\ A \to DA,\ A \to \varepsilon,$
$\quad B \to 0,\ B \to 1,\ C \to AAA,\ D \to 1AC\}$

Schritt 1: Entfernen von ε-Produktionen

$G_0 = (\{A, B, C, D, S\}, \{0, 1\}, P_0, S)$ mit
$P_0 = \{S \to 1A,\ A \to AB,\ A \to DA,\ A \to \varepsilon,\ B \to 0,$
$\quad B \to 1,\ C \to AAA,\ D \to 1AC\}$

Schritt 1: Entfernen von ε-Produktionen

$G_0 = (\{A, B, C, D, S\}, \{0, 1\}, P_0, S)$ mit
$P_0 = \{S \to 1A,\ A \to AB,\ A \to DA,\ A \to \varepsilon,\ B \to 0,$
$\quad B \to 1,\ C \to AAA,\ D \to 1AC\}$

- Finde Menge W der Variablen, die ε herleiten:
  $W = \{A, C\}$ da $A \to \varepsilon$ und $C \to AAA$

Schritt 1: Entfernen von ε-Produktionen

$G_0 = (\{A, B, C, D, S\}, \{0, 1\}, P_0, S)$ mit
$P_0 = \{S \to 1A,\ A \to AB,\ A \to DA,\ A \to \varepsilon,\ B \to 0,$
$\quad B \to 1,\ C \to AAA,\ D \to 1AC\}$

- Finde Menge W der Variablen, die ε herleiten:
  $W = \{A, C\}$ da $A \to \varepsilon$ und $C \to AAA$
- Starte mit
  $G_1 = (\{A, B, C, D, S\}, \{0, 1\}, P_1, S)$ mit
  $P_1 = \{S \to 1A,\ A \to AB,\ A \to DA,\ B \to 0,$
  $\quad B \to 1,\ C \to AAA,\ D \to 1AC\}$

Schritt 1: Entfernen von ε-Produktionen

$$G_0 = (\{A, B, C, D, S\}, \{0, 1\}, P_0, S) \text{ mit}$$
$$P_0 = \{S \to 1A,\ A \to AB,\ A \to DA,\ A \to \varepsilon,\ B \to 0,$$
$$B \to 1,\ C \to AAA,\ D \to 1AC\}$$

- Finde Menge W der Variablen, die ε herleiten:
  $W = \{A, C\}$ da $A \to \varepsilon$ und $C \to AAA$
- Starte mit
  $$G_1 = (\{A, B, C, D, S\}, \{0, 1\}, P_1, S) \text{ mit}$$
  $$P_1 = \{S \to 1A,\ A \to AB,\ A \to DA,\ B \to 0,$$
  $$B \to 1,\ C \to AAA,\ D \to 1AC\}$$
- Füge Produktionen für Vorkommen von A und C hinzu:
  $$P_1 = \{S \to 1A,\ S \to 1,\ A \to AB,\ A \to B,\ A \to DA,\ A \to D,$$
  $$B \to 0,\ B \to 1,\ C \to AAA,\ C \to AA,\ C \to A,$$
  $$D \to 1AC,\ D \to 1A,\ D \to 1C,\ D \to 1\}$$

Schritt 2: Entfernen von Einheitsproduktionen

$$\begin{aligned} P_1 = \{&S \to 1A,\ S \to 1,\ A \to AB,\ A \to B,\ A \to DA,\ A \to D, \\ &B \to 0,\ B \to 1\ \ C \to AAA,\ C \to AA,\ C \to A, \\ &D \to 1AC,\ D \to 1A,\ D \to 1C,\ D \to 1\} \end{aligned}$$

Schritt 2: Entfernen von Einheitsproduktionen

$$\begin{aligned} P_1 = \{&S \to 1A,\ S \to 1,\ A \to AB,\ A \to B,\ A \to DA,\ A \to D, \\ &B \to 0,\ B \to 1\ \ C \to AAA,\ C \to AA,\ C \to A, \\ &D \to 1AC,\ D \to 1A,\ D \to 1C,\ D \to 1\} \end{aligned}$$

- Der gerichtete Graph ist $D' = (\{S, A, B, C, D\}, \{(A, B), (A, D), (C, A)\})$.

Schritt 2: Entfernen von Einheitsproduktionen

$$
\begin{aligned}
P_1 = \{ & S \to 1A,\ S \to 1,\ A \to AB,\ A \to B,\ A \to DA,\ A \to D, \\
& B \to 0,\ B \to 1\ \ C \to AAA,\ C \to AA,\ C \to A, \\
& D \to 1AC,\ D \to 1A,\ D \to 1C,\ D \to 1 \}
\end{aligned}
$$

- Der gerichtete Graph ist $D' = (\{S, A, B, C, D\}, \{(A, B), (A, D), (C, A)\})$.
- Es gibt keine Zyklen, wir erhalten $G_2 = G_1$.

Schritt 2: Entfernen von Einheitsproduktionen

$$\begin{aligned}P_1 = \{&S \to 1A,\ S \to 1,\ A \to AB,\ A \to B,\ A \to DA,\ A \to D,\\ &B \to 0,\ B \to 1\ \ C \to AAA,\ C \to AA,\ C \to A,\\ &D \to 1AC,\ D \to 1A,\ D \to 1C,\ D \to 1\}\end{aligned}$$

- Der gerichtete Graph ist $D' = (\{S, A, B, C, D\}, \{(A, B), (A, D), (C, A)\})$.
- Es gibt keine Zyklen, wir erhalten $G_2 = G_1$.
- Sortieren und Umbenennen der Variablen, sodass „$A_i \to A_j$ impliziert $i < j$" gilt, erfordert eine Umbenennung ρ, welche die Beziehungen $A < B$, $A < D$, $C < A$ erzeugt.

Schritt 2: Entfernen von Einheitsproduktionen

$$P_1 = \{S \to 1A,\ S \to 1,\ A \to AB,\ A \to B,\ A \to DA,\ A \to D,$$
$$B \to 0,\ B \to 1\ \ C \to AAA,\ C \to AA,\ C \to A,$$
$$D \to 1AC,\ D \to 1A,\ D \to 1C,\ D \to 1\}$$

- Der gerichtete Graph ist $D' = (\{S, A, B, C, D\}, \{(A, B), (A, D), (C, A)\})$.
- Es gibt keine Zyklen, wir erhalten $G_2 = G_1$.
- Sortieren und Umbenennen der Variablen, sodass „$A_i \to A_j$ impliziert $i < j$“ gilt, erfordert eine Umbenennung ρ, welche die Beziehungen $A < B$, $A < D$, $C < A$ erzeugt.
- Wir wählen ρ mit $\rho(C) = A_1$, $\rho(A) = A_2$, $\rho(B) = A_3$, $\rho(D) = A_4$, $\rho(S) = A_5$.

Schritt 2: Entfernen von Einheitsproduktionen

$$\begin{aligned} P_1 = \{ & S \to 1A,\ S \to 1,\ A \to AB,\ A \to B,\ A \to DA,\ A \to D, \\ & B \to 0,\ B \to 1\ \ C \to AAA,\ C \to AA,\ C \to A, \\ & D \to 1AC,\ D \to 1A,\ D \to 1C,\ D \to 1\} \end{aligned}$$

- Der gerichtete Graph ist $D' = (\{S, A, B, C, D\}, \{(A, B), (A, D), (C, A)\})$.
- Es gibt keine Zyklen, wir erhalten $G_2 = G_1$.
- Sortieren und Umbenennen der Variablen, sodass „$A_i \to A_j$ impliziert $i < j$" gilt, erfordert eine Umbenennung ρ, welche die Beziehungen $A < B$, $A < D$, $C < A$ erzeugt.
- Wir wählen ρ mit $\rho(C) = A_1$, $\rho(A) = A_2$, $\rho(B) = A_3$, $\rho(D) = A_4$, $\rho(S) = A_5$.
- Das liefert uns $G_3 = (\{A_1, A_2, A_3, A_4, A_5\}, \Sigma, P_3, A_5)$ mit
$$\begin{aligned} P_3 = \{ & A_5 \to 1A_2,\ A_5 \to 1,\ A_2 \to A_2A_3,\ A_2 \to A_3,\ A_2 \to A_4A_2, \\ & A_2 \to A_4,\ A_3 \to 0,\ A_3 \to 1,\ A_1 \to A_2A_2A_2,\ A_1 \to A_2A_2, \\ & A_1 \to A_2,\ A_4 \to 1A_2A_1,\ A_4 \to 1A_2,\ A_4 \to 1A_1,\ A_4 \to 1\} \end{aligned}$$

Schritt 2: Entfernen von Einheitsproduktionen

$$P_3 = \{A_5 \to 1A_2,\ A_5 \to 1,\ A_2 \to A_2A_3,\ A_2 \to A_3,\ A_2 \to A_4A_2,\\ A_2 \to A_4,\ A_3 \to 0,\ A_3 \to 1,\ A_1 \to A_2A_2A_2,\ A_1 \to A_2A_2,\\ A_1 \to A_2,\ A_4 \to 1A_2A_1,\ A_4 \to 1A_2,\ A_4 \to 1A_1,\ A_4 \to 1\}$$

- Nun läuft die für-Schleife für i von 5 bis 1:

Schritt 2: Entfernen von Einheitsproduktionen

$$\begin{aligned} P_3 = \{ & A_5 \to 1A_2,\ A_5 \to 1,\ A_2 \to A_2A_3,\ A_2 \to A_3,\ A_2 \to A_4A_2, \\ & A_2 \to A_4,\ A_3 \to 0,\ A_3 \to 1,\ A_1 \to A_2A_2A_2,\ A_1 \to A_2A_2, \\ & A_1 \to A_2,\ A_4 \to 1A_2A_1,\ A_4 \to 1A_2,\ A_4 \to 1A_1,\ A_4 \to 1\} \end{aligned}$$

- Nun läuft die für-Schleife für i von 5 bis 1:
  - Für $i = 5$, $i = 4$, $i = 3$ gibt es jeweils keine Produktion der Form $A_i \to A_j$.

Schritt 2: Entfernen von Einheitsproduktionen

$$P_3 = \{A_5 \to 1A_2,\ A_5 \to 1,\ A_2 \to A_2A_3,\ A_2 \to A_3,\ A_2 \to A_4A_2,\\ A_2 \to A_4,\ A_3 \to 0,\ A_3 \to 1,\ A_1 \to A_2A_2A_2,\ A_1 \to A_2A_2,\\ A_1 \to A_2,\ A_4 \to 1A_2A_1,\ A_4 \to 1A_2,\ A_4 \to 1A_1,\ A_4 \to 1\}$$

- Nun läuft die für-Schleife für i von 5 bis 1:
  - Für $i = 5, i = 4, i = 3$ gibt es jeweils keine Produktion der Form $A_i \to A_j$.
  - Für $i = 2$ wird $A_2 \to A_3$ ersetzt durch $A_2 \to 0$, $A_2 \to 1$, und es wird $A_2 \to A_4$ ersetzt durch $A_2 \to 1A_2A_1$, $A_2 \to 1A_2$, $A_2 \to 1A_1$ und $A_2 \to 1$. Danach ist

    $$P_4 = \{A_5 \to 1A_2,\ A_5 \to 1,\ A_2 \to A_2A_3,\ A_2 \to 0,\ A_2 \to 1,\ A_2 \to A_4A_2,\\ A_2 \to 1A_2A_1,\ A_2 \to 1A_2,\ A_2 \to 1A_1,\ A_3 \to 0,\ A_3 \to 1,\\ A_1 \to A_2A_2A_2,\ A_1 \to A_2A_2,\ A_1 \to A_2,\ A_4 \to 1A_2A_1,\\ A_4 \to 1A_2,\ A_4 \to 1A_1,\ A_4 \to 1\}$$

Schritt 2: Entfernen von Einheitsproduktionen

$$\begin{aligned} P_3 = \{&A_5 \to 1A_2,\ A_5 \to 1,\ A_2 \to A_2A_3,\ A_2 \to A_3,\ A_2 \to A_4A_2,\\ &A_2 \to A_4,\ A_3 \to 0,\ A_3 \to 1,\ A_1 \to A_2A_2A_2,\ A_1 \to A_2A_2,\\ &A_1 \to A_2,\ A_4 \to 1A_2A_1,\ A_4 \to 1A_2,\ A_4 \to 1A_1,\ A_4 \to 1\} \end{aligned}$$

- Nun läuft die für-Schleife für i von 5 bis 1:
  - Für $i = 5$, $i = 4$, $i = 3$ gibt es jeweils keine Produktion der Form $A_i \to A_j$.
  - Für $i = 2$ wird $A_2 \to A_3$ ersetzt durch $A_2 \to 0$, $A_2 \to 1$, und es wird $A_2 \to A_4$ ersetzt durch $A_2 \to 1A_2A_1$, $A_2 \to 1A_2$, $A_2 \to 1A_1$ und $A_2 \to 1$.
    Danach ist
    $$\begin{aligned} P_4 = \{&A_5 \to 1A_2,\ A_5 \to 1,\ A_2 \to A_2A_3,\ A_2 \to 0,\ A_2 \to 1,\ A_2 \to A_4A_2,\\ &A_2 \to 1A_2A_1,\ A_2 \to 1A_2,\ A_2 \to 1A_1,\ A_3 \to 0,\ A_3 \to 1,\\ &A_1 \to A_2A_2A_2,\ A_1 \to A_2A_2,\ A_1 \to A_2,\ A_4 \to 1A_2A_1,\\ &A_4 \to 1A_2,\ A_4 \to 1A_1,\ A_4 \to 1\} \end{aligned}$$
  - Für $i = 1$ wird $A_1 \to A_2$ ersetzt durch $A_1 \to A_2A_3$, $A_1 \to 0$, $A_1 \to 1$, $A_1 \to A_4A_2$, $A_1 \to 1A_2A_1$, $A_1 \to 1A_2$ und $A_1 \to 1A_1$.

Schritt 2: Entfernen von Einheitsproduktionen

- Daher ist die Grammatik nach Entfernen von Einheitsproduktionen: $G_5 = (V_5, \Sigma, P_5, A_5)$ mit $V_5 = \{A_1, A_2, A_3, A_4, A_5\}$ und
$P_5 = \{A_5 \to 1A_2,\ A_5 \to 1,\ A_2 \to A_2A_3,\ A_2 \to 0,\ A_2 \to 1,$
$A_2 \to A_4A_2, A_2 \to 1A_2A_1,\ A_2 \to 1A_2,\ A_2 \to 1A_1,\ A_3 \to 0,$
$A_3 \to 1,\ A_1 \to A_2A_2A_2,\ A_1 \to A_2A_2,\ A_1 \to A_2A_3,\ A_1 \to 0,$
$A_1 \to 1,\ A_1 \to A_4A_2,\ A_1 \to 1A_2A_1,\ A_1 \to 1A_2,\ A_1 \to 1A_1,$
$A_4 \to 1A_2A_1,\ A_4 \to 1A_2,\ A_4 \to 1A_1,\ A_4 \to 1\}$

Schritt 3: Terminale durch Variablen darstellen

- Füge $B_0 \to 0$ und $B_1 \to 1$ hinzu und ersetze in rechten Seiten mit Wortlänge > 1:

$$\begin{aligned}P_6 = \{&B_0 \to 0,\ B_1 \to 1,\ A_5 \to B_1A_2,\ A_5 \to 1,\ A_2 \to A_2A_3,\ A_2 \to 0,\\ &A_2 \to 1,\ A_2 \to A_4A_2,\ A_2 \to B_1A_2A_1,\ A_2 \to B_1A_2,\\ &A_2 \to B_1A_1,\ A_3 \to 0,\ A_3 \to 1,\ A_1 \to A_2A_2A_2,\ A_1 \to A_2A_2,\\ &A_1 \to A_2A_3,\ A_1 \to 0,\ A_1 \to 1,\ A_1 \to A_4A_2,\ A_1 \to B_1A_2A_1,\\ &A_1 \to B_1A_2,\ A_1 \to B_1A_1,\ A_4 \to B_1A_2A_1,\\ &A_4 \to B_1A_2,\ A_4 \to B_1A_1,\ A_4 \to 1\}\end{aligned}$$

Schritt 4: Rechte Seiten zerlegen

- Zerlege rechte Seiten mit Wortlänge > 2:
  Ergibt $G_7 = (V_7, \Sigma, P_7, A_5)$, wobei
  $V_7 = \{A_1, A_2, A_3, A_4, A_5, B_0, B_1, C_1, C_2, C_3, C_4\}$
  $P_7 = \{B_0 \to 0,\ B_1 \to 1,\ A_5 \to B_1A_2,\ A_5 \to 1,\ A_2 \to A_2A_3,\ A_2 \to 0,$
  $A_2 \to 1,\ A_2 \to A_4A_2,\ A_2 \to B_1C_1,\ C_1 \to A_2A_1,$
  $A_2 \to B_1A_2,\ A_2 \to B_1A_1,\ A_3 \to 0,\ A_3 \to 1,\ A_1 \to A_2C_2,$
  $C_2 \to A_2A_2,\ A_1 \to A_2A_2,\ A_1 \to A_2A_3,\ A_1 \to 0,\ A_1 \to 1,$
  $A_1 \to A_4A_2,\ A_1 \to B_1C_3,\ C_3 \to A_2A_1,\ A_1 \to B_1A_2,$
  $A_1 \to B_1A_1,\ A_4 \to B_1C_4,\ C_4 \to A_2A_1,\ A_4 \to B_1A_2,$
  $A_4 \to B_1A_1,\ A_4 \to 1\}$

Schritt 4: Rechte Seiten zerlegen

- Zerlege rechte Seiten mit Wortlänge > 2:
  Ergibt $G_7 = (V_7, \Sigma, P_7, A_5)$, wobei
  $V_7 = \{A_1, A_2, A_3, A_4, A_5, B_0, B_1, C_1, C_2, C_3, C_4\}$
  $P_7 = \{B_0 \to 0,\ B_1 \to 1,\ A_5 \to B_1A_2,\ A_5 \to 1,\ A_2 \to A_2A_3,\ A_2 \to 0,$
  $A_2 \to 1,\ A_2 \to A_4A_2,\ A_2 \to B_1C_1,\ C_1 \to A_2A_1,$
  $A_2 \to B_1A_2,\ A_2 \to B_1A_1,\ A_3 \to 0,\ A_3 \to 1,\ A_1 \to A_2C_2,$
  $C_2 \to A_2A_2,\ A_1 \to A_2A_2,\ A_1 \to A_2A_3,\ A_1 \to 0,\ A_1 \to 1,$
  $A_1 \to A_4A_2,\ A_1 \to B_1C_3,\ C_3 \to A_2A_1,\ A_1 \to B_1A_2,$
  $A_1 \to B_1A_1,\ A_4 \to B_1C_4,\ C_4 \to A_2A_1,\ A_4 \to B_1A_2,$
  $A_4 \to B_1A_1,\ A_4 \to 1\}$

G_7 ist in Chomsky-Normalform.