Script generated by TTT

Title: Seidl: Theoretische_Informatik (23.05.2013)

Date: Thu May 23 16:01:06 CEST 2013

Duration: 90:41 min

Pages: 71

Moral:

- „$w \in L_G(S) \implies P(w)$" beweist man immer schematisch mit Induktion über die Erzeugung von w.
- „$P(w) \implies w \in L_G(S)$" beweist man oft mit Induktion über $|w|$. Erfordert meist Kreativität.

Wir übertragen jetzt die Idee der induktiven Erzeugung und der dazugehörige Induktionsregel auf beliebige CFGs $G = (V, \Sigma, P, S)$.

Wir übertragen jetzt die Idee der induktiven Erzeugung und der dazugehörige Induktionsregel auf beliebige CFGs $G = (V, \Sigma, P, S)$.
Sei $V = \{A_1, \dots, A_k\}$.

Wir übertragen jetzt die Idee der induktiven Erzeugung und der dazugehörige Induktionsregel auf beliebige CFGs $G = (V, \Sigma, P, S)$.
Sei $V = \{A_1, \ldots, A_k\}$. Damit ist jede Produktion von der Form

$$A_i \to w_0 \boxed{A_{i_1}} w_1 \ldots w_{n-1} \boxed{A_{i_n}} w_n$$

Wir übertragen jetzt die Idee der induktiven Erzeugung und der dazugehörige Induktionsregel auf beliebige CFGs $G = (V, \Sigma, P, S)$.
Sei $V = \{A_1, \ldots, A_k\}$. Damit ist jede Produktion von der Form

$$A_i \to w_0 A_{i_1} w_1 \ldots w_{n-1} A_{i_n} w_n$$

Man kann G wie folgt als eine (simultane) induktive Definition von Sprachen $L_G(A_i)$ für all $A_i \in V$ sehen:

Wir übertragen jetzt die Idee der induktiven Erzeugung und der dazugehörige Induktionsregel auf beliebige CFGs $G = (V, \Sigma, P, S)$.
Sei $V = \{A_1, \ldots, A_k\}$. Damit ist jede Produktion von der Form

$$A_i \to w_0 A_{i_1} w_1 \ldots w_{n-1} A_{i_n} w_n$$

Man kann G wie folgt als eine (simultane) induktive Definition von Sprachen $L_G(A_i)$ für all $A_i \in V$ sehen:
Jede Produktion korrespondiert zu einer Erzeugungsregel

$$u_1 \in L_G(A_{i_1}) \wedge \cdots \wedge u_n \in L_G(A_{i_n}) \implies w_0 u_1 w_1 \ldots w_{n-1} u_n w_n \in L_G(A_i)$$

Wir übertragen jetzt die Idee der induktiven Erzeugung und der dazugehörige Induktionsregel auf beliebige CFGs $G = (V, \Sigma, P, S)$.
Sei $V = \{A_1, \ldots, A_k\}$. Damit ist jede Produktion von der Form

$$A_i \to w_0 A_{i_1} w_1 \ldots w_{n-1} A_{i_n} w_n$$

Man kann G wie folgt als eine (simultane) induktive Definition von Sprachen $L_G(A_i)$ für all $A_i \in V$ sehen:
Jede Produktion korrespondiert zu einer Erzeugungsregel

$$u_1 \in L_G(A_{i_1}) \wedge \cdots \wedge u_n \in L_G(A_{i_n}) \implies w_0 u_1 w_1 \ldots w_{n-1} u_n w_n \in L_G(A_i)$$

Aussagen der Form

$$(u \in L_G(A_1) \implies P_1(u)) \wedge \cdots \wedge (u \in L_G(A_k) \implies P_k(u))$$

Wir übertragen jetzt die Idee der induktiven Erzeugung und der dazugehörige Induktionsregel auf beliebige CFGs $G = (V, \Sigma, P, S)$. Sei $V = \{A_1, \ldots, A_k\}$. Damit ist jede Produktion von der Form

$$A_i \to w_0 A_{i_1} w_1 \ldots w_{n-1} A_{i_n} w_n$$

Man kann G wie folgt als eine (simultane) induktive Definition von Sprachen $L_G(A_i)$ für all $A_i \in V$ sehen:
Jede Produktion korrespondiert zu einer Erzeugungsregel

$$u_1 \in L_G(A_{i_1}) \wedge \cdots \wedge u_n \in L_G(A_{i_n}) \implies w_0 u_1 w_1 \ldots w_{n-1} u_n w_n \in L_G(A_i)$$

Aussagen der Form

$$(u \in L_G(A_1) \implies P_1(u)) \wedge \cdots \wedge (u \in L_G(A_k) \implies P_k(u))$$

kann man durch simultane Induktion über die Erzeugung von u beweisen, indem man für jede Produktion zeigt, dass

$$P_{i_1}(u_1) \wedge \cdots \wedge P_{i_n}(u_n) \implies P_i(w_0 u_1 w_1 \ldots w_{n-1} u_n w_n)$$

Beispiel 3.17

Grammatik:

$$A \to \epsilon \mid aB \qquad B \to Aa$$

Beispiel 3.17

Grammatik:

$$A \to \epsilon \mid aB \qquad B \to Aa$$

Induktive Definition:

Beispiel 3.17

Grammatik:

$$A \to \epsilon \mid aB \qquad B \to Aa$$

Induktive Definition:

- $\epsilon \in L_G(A)$
- $w \in L_G(B) \implies aw \in L_G(A)$
- $w \in L_G(A) \implies wa \in L_G(B)$

Beispiel 3.17

Grammatik:

$$A \to \epsilon \mid aB \qquad B \to Aa$$

Induktive Definition:

- $\epsilon \in L_G(A)$
- $w \in L_G(B) \implies aw \in L_G(A)$
- $w \in L_G(A) \implies wa \in L_G(B)$

Beim induktiven Beweis von

$$(w \in L_G(A) \implies \#_a(w) \text{ ist gerade}) \wedge$$
$$(w \in L_G(B) \implies \#_a(w) \text{ ist ungerade})$$

muss man zeigen

Beispiel 3.17

Grammatik:

$$A \to \epsilon \mid aB \qquad B \to Aa$$

Induktive Definition:

- $\epsilon \in L_G(A)$
- $w \in L_G(B) \implies aw \in L_G(A)$
- $w \in L_G(A) \implies wa \in L_G(B)$

Beim induktiven Beweis von

$$(w \in L_G(A) \implies \#_a(w) \text{ ist gerade}) \wedge$$
$$(w \in L_G(B) \implies \#_a(w) \text{ ist ungerade})$$

muss man zeigen

- $\#_a(\epsilon)$ ist gerade

Beispiel 3.17

Grammatik:

$$A \to \epsilon \mid aB \qquad B \to Aa$$

Induktive Definition:

- $\epsilon \in L_G(A)$
- $w \in L_G(B) \implies aw \in L_G(A)$
- $w \in L_G(A) \implies wa \in L_G(B)$

Beim induktiven Beweis von

$$(w \in L_G(A) \implies \#_a(w) \text{ ist gerade}) \wedge$$
$$(w \in L_G(B) \implies \#_a(w) \text{ ist ungerade})$$

muss man zeigen

- $\#_a(\epsilon)$ ist gerade

- $\#_a(w)$ ist ungerade $\implies \#_a(aw)$ ist gerade

Beispiel 3.17

Grammatik:

$$A \to \epsilon \mid aB \qquad B \to Aa$$

Induktive Definition:

- $\epsilon \in L_G(A)$
- $w \in L_G(B) \implies aw \in L_G(A)$
- $w \in L_G(A) \implies wa \in L_G(B)$

Beim induktiven Beweis von

$$(w \in L_G(A) \implies \#_a(w) \text{ ist gerade}) \wedge$$
$$(w \in L_G(B) \implies \#_a(w) \text{ ist ungerade})$$

muss man zeigen

- $\#_a(\epsilon)$ ist gerade
- $\#_a(w)$ ist ungerade $\implies \#_a(aw)$ ist gerade

- $\#_a(w)$ ist gerade $\implies \#_a(wa)$ ist ungerade

Definition 3.18 (Syntaxbaum)

Ein Syntaxbaum für eine Ableitung mit einer Grammatik $G = (V, \Sigma, P, S)$ ist ein Baum, so dass

Definition 3.18 (Syntaxbaum)

Ein Syntaxbaum für eine Ableitung mit einer Grammatik $G = (V, \Sigma, P, S)$ ist ein Baum, so dass

- jedes Blatt mit einem Zeichen aus $\Sigma \cup \{\epsilon\}$ beschriftet ist,

Definition 3.18 (Syntaxbaum)

Ein Syntaxbaum für eine Ableitung mit einer Grammatik $G = (V, \Sigma, P, S)$ ist ein Baum, so dass

- jedes Blatt mit einem Zeichen aus $\Sigma \cup \{\epsilon\}$ beschriftet ist,
- jeder innere Knoten mit einem $A \in V$ beschriftet ist, und falls die Nachfolger (von links nach rechts) mit $X_1, \ldots, X_n \in V \cup \Sigma \cup \{\epsilon\}$ beschriftet sind,

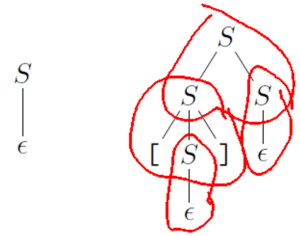
$A \to \cancel{X}_1 \ldots X_n$

Definition 3.18 (Syntaxbaum)

Ein Syntaxbaum für eine Ableitung mit einer Grammatik $G = (V, \Sigma, P, S)$ ist ein Baum, so dass

- jedes Blatt mit einem Zeichen aus $\Sigma \cup \{\epsilon\}$ beschriftet ist,
- jeder innere Knoten mit einem $A \in V$ beschriftet ist, und falls die Nachfolger (von links nach rechts) mit $X_1, \ldots, X_n \in V \cup \Sigma \cup \{\epsilon\}$ beschriftet sind, dann ist $A \to X_1 \ldots X_n$ eine Produktion in P,
- ein Blatt ϵ der einzige Nachfolger seines Vorgängers ist.

Beispiel 3.19 (Syntaxbäume für $S \to \epsilon \mid [S] \mid SS$)

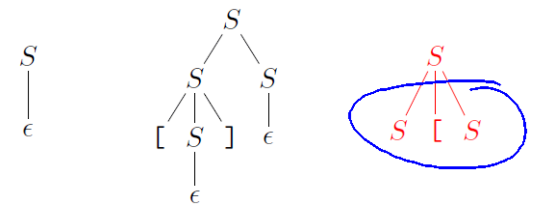


Definition 3.18 (Syntaxbaum)

Ein Syntaxbaum für eine Ableitung mit einer Grammatik $G = (V, \Sigma, P, S)$ ist ein Baum, so dass

- jedes Blatt mit einem Zeichen aus $\Sigma \cup \{\epsilon\}$ beschriftet ist,
- jeder innere Knoten mit einem $A \in V$ beschriftet ist, und falls die Nachfolger (von links nach rechts) mit $X_1, \ldots, X_n \in V \cup \Sigma \cup \{\epsilon\}$ beschriftet sind, dann ist $A \to X_1 \ldots X_n$ eine Produktion in P,
- ein Blatt ϵ der einzige Nachfolger seines Vorgängers ist.

Beispiel 3.19 (Syntaxbäume für $S \to \epsilon \mid [S] \mid SS$)



Satz 3.20 G

Für eine CFG und ein $w \in \Sigma^*$ sind folgende Bedingungen äquivalent:

1. $A \to_G^* w$
2. $w \in L_G(A)$ *(gemäß der induktiven Definition)*

Satz 3.20

Für eine CFG und ein $w \in \Sigma^*$ sind folgende Bedingungen äquivalent:

1. $A \to_G^* w$
2. $w \in L_G(A)$ *(gemäß der induktiven Definition)*

3. *Es gibt einen Syntaxbaum, dessen Rand (Blätter von links nach rechts gelesen) das Wort w ist.*

dessen Wurzel mit A ist

Satz 3.20

Für eine CFG und ein $w \in \Sigma^*$ sind folgende Bedingungen äquivalent:

1. $A \to_G^* w$
2. $w \in L_G(A)$ *(gemäß der induktiven Definition)*
3. *Es gibt einen Syntaxbaum, dessen Rand (Blätter von links nach rechts gelesen) das Wort w ist.*

Beweis:

Skizze (Details: [HMU])

Satz 3.20

Für eine CFG und ein $w \in \Sigma^*$ sind folgende Bedingungen äquivalent:

1. $A \to_G^* w$
2. $w \in L_G(A)$ *(gemäß der induktiven Definition)*
3. *Es gibt einen Syntaxbaum, dessen Rand (Blätter von links nach rechts gelesen) das Wort w ist.*

Beweis:

Skizze (Details: [HMU])

$1 \Rightarrow 2$: Sei $A \to_G^n w$. Wir konstruieren dazu einen Beweis für $w \in L_G(A)$ mit Induktion über n.

Satz 3.20

Für eine CFG und ein $w \in \Sigma^*$ sind folgende Bedingungen äquivalent:

1. $A \to_G^* w$
2. $w \in L_G(A)$ *(gemäß der induktiven Definition)*
3. *Es gibt einen Syntaxbaum, dessen Rand (Blätter von links nach rechts gelesen) das Wort w ist.*

Beweis:

Skizze (Details: [HMU])

$1 \Rightarrow 2$: Sei $A \to_G^n w$. Wir konstruieren dazu einen Beweis für $w \in L_G(A)$ mit Induktion über n. Die Ableitung muss von folgender Form sein:

$$A \to_G \; w_0 A_1 w_1 \dots A_m w_m \; \to_G^{n-1} \; w$$

Satz 3.20

Für eine CFG und ein $w \in \Sigma^*$ sind folgende Bedingungen äquivalent:

1. $A \to_G^* w$
2. $w \in L_G(A)$ *(gemäß der induktiven Definition)*
3. *Es gibt einen Syntaxbaum, dessen Rand (Blätter von links nach rechts gelesen) das Wort w ist.*

Beweis:

Skizze (Details: [HMU])

$1 \Rightarrow 2$: Sei $A \to_G^n w$. Wir konstruieren dazu einen Beweis für $w \in L_G(A)$ mit Induktion über n. Die Ableitung muss von folgender Form sein:

$$A \to_G \; w_0 A_1 w_1 \dots A_m w_m \; \to_G^{n-1} \; w$$

Nach dem Dekompositionslemma muss es u_i und $n_i \leq n-1$ geben mit $A_i \to_G^{n_i} u_i$ und $w = w_0 u_1 w_1 \dots u_m w_m$.

Satz 3.20

Für eine CFG und ein $w \in \Sigma^*$ sind folgende Bedingungen äquivalent:

1. $A \to_G^* w$
2. $w \in L_G(A)$ *(gemäß der induktiven Definition)*
3. *Es gibt einen Syntaxbaum, dessen Rand (Blätter von links nach rechts gelesen) das Wort w ist.*

Beweis:

Skizze (Details: [HMU])

$1 \Rightarrow 2$: Sei $A \to_G^n w$. Wir konstruieren dazu einen Beweis für $w \in L_G(A)$ mit Induktion über n. Die Ableitung muss von folgender Form sein:

$$A \to_G \; w_0 A_1 w_1 \dots A_m w_m \; \to_G^{n-1} \; w$$

Nach dem Dekompositionslemma muss es u_i und $n_i \leq n-1$ geben mit $A_i \to_G^{n_i} u_i$ und $w = w_0 u_1 w_1 \dots u_m w_m$. Nach IA (da $n_i < n$) gilt, gilt $u_i \in L_G(A_i)$ und daher (mit einem weiteren Erzeugungsschritt) auch $w \in L_G(A)$.

$2 \Rightarrow 3$: Parallel zur induktiven Erzeugung von $w \in L_G(A)$ kann man einen Syntaxbaum mit Rand w erzeugen. Formal: Induktion über die Erzeugung von w.
$3 \Rightarrow 1$: Jeder Baum lässt sich in eine Ableitung, sogar eine Linksableitung(!) umformen. Induktion über die Höhe des Baums: Transformiere die Unterbäume t_i mit Beschriftung A_i in Ableitungen $A_i \to_G^* u_i$ und füge diese mit dem Dekompositionslemma zu einer grossen Ableitung $A \to_G w_0 A_1 w_1 \dots A_m w_m \to_G^* w_0 u_1 w_1 \dots u_m w_m$ zusammen.

Satz 3.20

Für eine CFG und ein $w \in \Sigma^*$ sind folgende Bedingungen äquivalent:

Satz 3.20

Für eine CFG und ein $w \in \Sigma^*$ sind folgende Bedingungen äquivalent:

1. $A \to_G^* w$
2. $w \in L_G(A)$ *(gemäß der induktiven Definition)*
3. *Es gibt einen Syntaxbaum, dessen Rand (Blätter von links nach rechts gelesen) das Wort w ist.*

Beweis:
Skizze (Details: [HMU])
$1 \Rightarrow 2$: Sei $A \to_G^n w$. Wir konstruieren dazu einen Beweis für $w \in L_G(A)$ mit Induktion über n. Die Ableitung muss von folgender Form sein:

$$A \to_G w_0 A_1 w_1 \dots A_m w_m \to_G^{n-1} w$$

Nach dem Dekompositionslemma muss es u_i und $n_i \leq n-1$ geben mit $A_i \to_G^{n_i} u_i$ und $w = w_0 u_1 w_1 \dots u_m w_m$. Nach IA (da $n_i < n$) gilt, gilt $u_i \in L_G(A_i)$

Definition 3.21

- Eine CFG G heißt mehrdeutig gdw es ein $w \in L(G)$ gibt, das zwei verschiedene Syntaxbäume hat, also zwei verschiedene Syntaxbäume mit Wurzel S und Rand w.

Definition 3.21

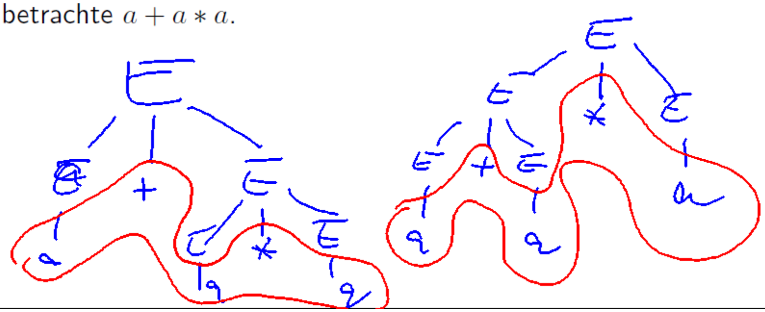
- Eine CFG G heißt mehrdeutig gdw es ein $w \in L(G)$ gibt, das zwei verschiedene Syntaxbäume hat, also zwei verschiedene Syntaxbäume mit Wurzel S und Rand w.
- Eine CFL L heißt inhärent mehrdeutig gdw jede CFG G mit $L(G) = L$ mehrdeutig ist.

Definition 3.21

- Eine CFG G heißt mehrdeutig gdw es ein $w \in L(G)$ gibt, das zwei verschiedene Syntaxbäume hat, also zwei verschiedene Syntaxbäume mit Wurzel S und Rand w.
- Eine CFL L heißt inhärent mehrdeutig gdw jede CFG G mit $L(G) = L$ mehrdeutig ist.

Beispiel 3.22

Die Grammatik $E \to E + E \mid E * E \mid (E) \mid a$ ist mehrdeutig — betrachte $a + a * a$.



Definition 3.21

- Eine CFG G heißt mehrdeutig gdw es ein $w \in L(G)$ gibt, das zwei verschiedene Syntaxbäume hat, also zwei verschiedene Syntaxbäume mit Wurzel S und Rand w.
- Eine CFL L heißt inhärent mehrdeutig gdw jede CFG G mit $L(G) = L$ mehrdeutig ist.

Beispiel 3.22

Die Grammatik $E \to E + E \mid E * E \mid (E) \mid a$ ist mehrdeutig — betrachte $a + a * a$. Die erzeugte Sprache ist aber nicht inhärent mehrdeutig (siehe Beispiel Arithmetische Ausdrücke).

Satz 3.23

Die Sprache $\{a^i b^j c^k \mid i = j \lor j = k\}$ ist inhärent mehrdeutig.

Definition 3.21

- Eine CFG G heißt mehrdeutig gdw es ein $w \in L(G)$ gibt, das zwei verschiedene Syntaxbäume hat, also zwei verschiedene Syntaxbäume mit Wurzel S und Rand w.
- Eine CFL L heißt inhärent mehrdeutig gdw jede CFG G mit $L(G) = L$ mehrdeutig ist.

Beispiel 3.22

Die Grammatik $E \to E + E \mid E * E \mid (E) \mid a$ ist mehrdeutig — betrachte $a + a * a$. Die erzeugte Sprache ist aber nicht inhärent mehrdeutig (siehe Beispiel Arithmetische Ausdrücke).

Satz 3.23

Die Sprache $\{a^i b^j c^k \mid i = j \lor j = k\}$ ist inhärent mehrdeutig.

3.3 Die Chomsky-Normalform

Definition 3.24
Eine kontextfreie Grammatik G ist in Chomsky-Normalform gdw alle Produktionen eine der Formen

$$A \to a \quad \text{oder} \quad A \to BC$$

haben.

3.3 Die Chomsky-Normalform

Definition 3.24
Eine kontextfreie Grammatik G ist in Chomsky-Normalform gdw alle Produktionen eine der Formen

$$A \to a \quad \text{oder} \quad A \to BC$$

haben.

$L = \{\varepsilon\}$

3.3 Die Chomsky-Normalform

Definition 3.24
Eine kontextfreie Grammatik G ist in Chomsky-Normalform gdw alle Produktionen eine der Formen

$$A \to a \quad \text{oder} \quad A \to BC$$

haben.

Wir konstruieren und beweisen nun schrittweise:

Satz 3.25
Zu jeder CFG G kann man eine CFG G' in Chomsky-Normalform konstruieren mit $L(G') = L(G) \setminus \{\epsilon\}$.

3.3 Die Chomsky-Normalform

Definition 3.24
Eine kontextfreie Grammatik G ist in Chomsky-Normalform gdw alle Produktionen eine der Formen

$$A \to a \quad \text{oder} \quad A \to BC$$

haben.

Wir konstruieren und beweisen nun schrittweise:

Satz 3.25
Zu jeder CFG G kann man eine CFG G' in Chomsky-Normalform konstruieren mit $L(G') = L(G) \setminus \{\epsilon\}$.

Wer auf $\epsilon \in L(G')$ nicht verzichten möchte:
Füge am Ende wieder $S \to \epsilon$ hinzu.

Wir nennen $A \to \epsilon$ eine ϵ-Produktion.

3.3 Die Chomsky-Normalform

Definition 3.24
Eine kontextfreie Grammatik G ist in Chomsky-Normalform gdw alle Produktionen eine der Formen

$$A \to a \quad \text{oder} \quad A \to BC$$

haben.

Wir konstruieren und beweisen nun schrittweise:

Satz 3.25
Zu jeder CFG G kann man eine CFG G' in Chomsky-Normalform konstruieren mit $L(G') = L(G) \setminus \{\epsilon\}$.

Wer auf $\epsilon \in L(G')$ nicht verzichten möchte:
Füge am Ende wieder $S \to \epsilon$ hinzu.

Wir nennen $A \to \epsilon$ eine ϵ-Produktion.

Lemma 3.26
Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine ϵ-Produktionen enthält, so dass gilt $L(G') = L(G) \setminus \{\epsilon\}$

Wir nennen $A \to \epsilon$ eine ϵ-Produktion.

Lemma 3.26
Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine ϵ-Produktionen enthält, so dass gilt $L(G') = L(G) \setminus \{\epsilon\}$

Beweis:
Wir erweitern P induktiv zu eine Obermenge $\hat{P}$:

Wir nennen $A \to \epsilon$ eine ϵ-Produktion.

Lemma 3.26
Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine ϵ-Produktionen enthält, so dass gilt $L(G') = L(G) \setminus \{\epsilon\}$

Beweis:
Wir erweitern P induktiv zu eine Obermenge $\hat{P}$:

1. Jede Produktion aus P ist in $\hat{P}$

Wir nennen $A \to \epsilon$ eine ϵ-Produktion.

Lemma 3.26
Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine ϵ-Produktionen enthält, so dass gilt $L(G') = L(G) \setminus \{\epsilon\}$

Beweis:
Wir erweitern P induktiv zu eine Obermenge $\hat{P}$:

1. Jede Produktion aus P ist in $\hat{P}$
2. Sind $B \to \epsilon$ und $A \to \alpha B \beta$ in $\hat{P}$, so füge auch $A \to \alpha\beta$ hinzu.

Wir nennen $A \to \epsilon$ eine ϵ-Produktion.

Lemma 3.26
Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine ϵ-Produktionen enthält, so dass gilt $L(G') = L(G) \setminus \{\epsilon\}$

Beweis:
Wir erweitern P induktiv zu eine Obermenge $\hat{P}$:

1. Jede Produktion aus P ist in $\hat{P}$
2. Sind $B \to \epsilon$ und $A \to \alpha B \beta$ in $\hat{P}$, so füge auch $A \to \alpha\beta$ hinzu.

Offensichtlich gilt $L(\hat{G}) = L(G)$:
Jede neue Produktion kann von 2 alten simuliert werden.

Wir nennen $A \to \epsilon$ eine ϵ-Produktion.

Lemma 3.26
Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine ϵ-Produktionen enthält, so dass gilt $L(G') = L(G) \setminus \{\epsilon\}$

Beweis:
Wir erweitern P induktiv zu eine Obermenge $\hat{P}$:

1. Jede Produktion aus P ist in $\hat{P}$
2. Sind $B \to \epsilon$ und $A \to \alpha B \beta$ in $\hat{P}$, so füge auch $A \to \alpha\beta$ hinzu.

Offensichtlich gilt $L(\hat{G}) = L(G)$:
Jede neue Produktion kann von 2 alten simuliert werden.
Das Ergebnis G' ist $\hat{G}$ ohne die ϵ-Produktionen.

Beweis (Forts.):

Sei $S \to_{\hat{G}}^* w$, $w \neq \epsilon$, eine Ableitung minimaler Länge.

Beweis (Forts.):

Sei $S \to_{\hat{G}}^* w$, $w \neq \epsilon$, eine Ableitung minimaler Länge.
Käme darin $B \to \epsilon$ vor, also

$$S \to_{\hat{G}}^* \gamma B \delta \to_{\hat{G}} \gamma\delta \to_{\hat{G}}^* w$$

Beweis (Forts.):

Sei $S \to_{\hat{G}}^* w$, $w \neq \epsilon$, eine Ableitung minimaler Länge.
Käme darin $B \to \epsilon$ vor, also

$$S \to_{\hat{G}}^* \gamma B \delta \to_{\hat{G}} \gamma\delta \to_{\hat{G}}^* w$$

so wäre γ oder δ nichtleer, dh B muss durch eine Produktion $A \to \alpha B \beta$ eingeführt worden sein:

Beweis (Forts.):

Sei $S \to_{\hat{G}}^* w$, $w \neq \epsilon$, eine Ableitung minimaler Länge.
Käme darin $B \to \epsilon$ vor, also

$$S \to_{\hat{G}}^* \gamma B \delta \to_{\hat{G}} \gamma\delta \to_{\hat{G}}^* w$$

so wäre γ oder δ nichtleer, dh B muss durch eine Produktion $A \to \alpha B \beta$ eingeführt worden sein:

$$S \to_{\hat{G}}^k \alpha' A \beta' \to_{\hat{G}} \alpha' \alpha B \beta \beta' \to_{\hat{G}}^m \gamma B \delta \to_{\hat{G}} \gamma\delta \to_{\hat{G}}^n w$$

Dann wäre aber folgende Ableitung mit $(A \to \alpha\beta) \in \hat{P}$ kürzer:

$$S \to_{\hat{G}}^k \alpha' A \beta' \to_{\hat{G}} \alpha' \alpha \beta \beta' \to_{\hat{G}}^m \gamma\delta \to_{\hat{G}}^n w$$

Beispiel 3.27

$$S \to AB, \quad A \to aAA \mid \epsilon, \qquad B \to bBB \mid \epsilon$$

$S \to B$ $\quad A \to aA$ $\quad B \to bB$

$S \to \varepsilon$ $\quad A \to a$ $\quad B \to b$

$S \to A$

Wir nennen $A \to B$ eine Kettenproduktion.

Wir nennen $A \to B$ eine Kettenproduktion.

Lemma 3.28
Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine Kettenproduktionen enthält, so dass gilt $L(G') = L(G)$.

Beispiel 3.27

$$S \to AB, \quad A \to aAA \mid \epsilon, \qquad B \to bBB \mid \epsilon$$

$\mid \mid bB$

$\mid b$

Wir nennen $A \to B$ eine Kettenproduktion.

Lemma 3.28

Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine Kettenproduktionen enthält, so dass gilt $L(G') = L(G)$.

Beweis:

Wir erweitern P induktiv zu eine Obermenge $\hat{P}$:

Wir nennen $A \to B$ eine Kettenproduktion.

Lemma 3.28

Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine Kettenproduktionen enthält, so dass gilt $L(G') = L(G)$.

Beweis:

Wir erweitern P induktiv zu eine Obermenge $\hat{P}$:

1. Jede Produktion aus P ist in $\hat{P}$
2. Sind $A \to B$ und $B \to \alpha$ in $\hat{P}$, so füge auch $A \to \alpha$ hinzu.

$A \to B$ $\qquad$ $B \to A$

Wir nennen $A \to B$ eine Kettenproduktion.

Lemma 3.28

Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine Kettenproduktionen enthält, so dass gilt $L(G') = L(G)$.

Beweis:

Wir erweitern P induktiv zu eine Obermenge $\hat{P}$:

1. Jede Produktion aus P ist in $\hat{P}$
2. Sind $A \to B$ und $B \to \alpha$ in $\hat{P}$, so füge auch $A \to \alpha$ hinzu.

Das Ergebnis G' ist $\hat{G}$ ohne die (nun überflüssigen) Kettenproduktionen.

Wir nennen $A \to B$ eine Kettenproduktion.

Lemma 3.28

Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine Kettenproduktionen enthält, so dass gilt $L(G') = L(G)$.

Beweis:

Wir erweitern P induktiv zu eine Obermenge $\hat{P}$:

1. Jede Produktion aus P ist in $\hat{P}$
2. Sind $A \to B$ und $B \to \alpha$ in $\hat{P}$, so füge auch $A \to \alpha$ hinzu.

Das Ergebnis G' ist $\hat{G}$ ohne die (nun überflüssigen) Kettenproduktionen.

Rest des Beweises analog zur Elimination von ϵ-Produktionen. $\square$

Beispiel 3.29

$$A \to a \mid B, \qquad B \to b \mid C, \qquad C \to A$$

$| b$ $\quad$ $| A$ $\quad$ $| a$

$| C$ $\quad$ $| a$ $\quad$ $| B$

$| A$ $\quad$ $| B$ $\quad$ $| b$

$|$ $\quad$ $| C$

Konstruktion einer Chomsky-Normalform

Eingabe: Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$

Konstruktion einer Chomsky-Normalform

Eingabe: Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$

1. Eliminiere alle ϵ-Produktionen.

Konstruktion einer Chomsky-Normalform

Eingabe: Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$

1. Eliminiere alle ϵ-Produktionen.
2. Eliminiere alle Kettenproduktionen.

Konstruktion einer Chomsky-Normalform

Eingabe: Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$

1. Eliminiere alle ϵ-Produktionen.
2. Eliminiere alle Kettenproduktionen.
3. Füge für jedes $a \in \Sigma$, das in einer rechten Seite der Länge ≥ 2 vorkommt, ein neues Nichtterminal A_a zu V hinzu,

Konstruktion einer Chomsky-Normalform

Eingabe: Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$

1. Eliminiere alle ϵ-Produktionen.
2. Eliminiere alle Kettenproduktionen.
3. Füge für jedes $a \in \Sigma$, das in einer rechten Seite der Länge ≥ 2 vorkommt, ein neues Nichtterminal A_a zu V hinzu, ersetze a in allen rechten Seiten der Länge ≥ 2 durch A_a, und füge $A_a \to a$ zu P hinzu.

Konstruktion einer Chomsky-Normalform

Eingabe: Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$

1. Eliminiere alle ϵ-Produktionen.
2. Eliminiere alle Kettenproduktionen.
3. Füge für jedes $a \in \Sigma$, das in einer rechten Seite der Länge ≥ 2 vorkommt, ein neues Nichtterminal A_a zu V hinzu, ersetze a in allen rechten Seiten der Länge ≥ 2 durch A_a, und füge $A_a \to a$ zu P hinzu.
4. Ersetze jede Produktion der Form

$$A \to B_1 B_2 \cdots B_k \ (k \geq 3)$$

durch

$$A \to B_1 C_2, \ C_2 \to B_2 C_3, \ \ldots, \ C_{k-1} \to B_{k-1} B_k$$

wobei $C_2, \ldots, C_{k-1}$ neue Nichtterminale sind.

Konstruktion einer Chomsky-Normalform

Eingabe: Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$

1. Eliminiere alle ϵ-Produktionen.
2. Eliminiere alle Kettenproduktionen.
3. Füge für jedes $a \in \Sigma$, das in einer rechten Seite der Länge ≥ 2 vorkommt, ein neues Nichtterminal A_a zu V hinzu, ersetze a in allen rechten Seiten der Länge ≥ 2 durch A_a, und füge $A_a \to a$ zu P hinzu.
4. Ersetze jede Produktion der Form

$$A \to B_1 B_2 \cdots B_k \ (k \geq 3)$$

durch

$$A \to B_1 C_2, \ C_2 \to B_2 C_3, \ \ldots, \ C_{k-1} \to B_{k-1} B_k$$

wobei $C_2, \ldots, C_{k-1}$ neue Nichtterminale sind.

Definition 3.30
Eine CFG ist in Greibach-Normalform (benannt nach Sheila Greibach, UCLA), falls jede Produktion von der Form

$$A \to aA_1 \dots A_n$$

ist.

Definition 3.30
Eine CFG ist in Greibach-Normalform (benannt nach Sheila Greibach, UCLA), falls jede Produktion von der Form

$$A \to aA_1 \dots A_n$$

ist.

Satz 3.31
Zu jeder CFG G gibt es eine CFG G' in Greibach-Normalform mit $L(G') = L(G) \setminus \{\epsilon\}$.

3.4 Das Pumping-Lemma für kontextfreie Sprachen

Zur Erinnerung: Das Pumping-Lemma für reguläre Sprachen: Für jede reguläre Sprache L gibt es ein $n \in \mathbb{N}$, so dass sich jedes Wort $z \in L$ mit $|z| \geq n$ zerlegen lässt in $z = uvw$ mit $|uv| \leq n$, $v \neq \epsilon$ und $uv^*w \subseteq L$.

Satz 3.32 (Pumping-Lemma)
Für jede kontextfreie Sprache L gibt es ein $n \geq 1$, so dass sich jedes Wort $z \in L$ mit $|z| \geq n$ zerlegen lässt in

$$z = uvwxy,$$

mit