Script **generated by TTT**

Title: Seidl: Theoretische_Informatik (14.05.2012)

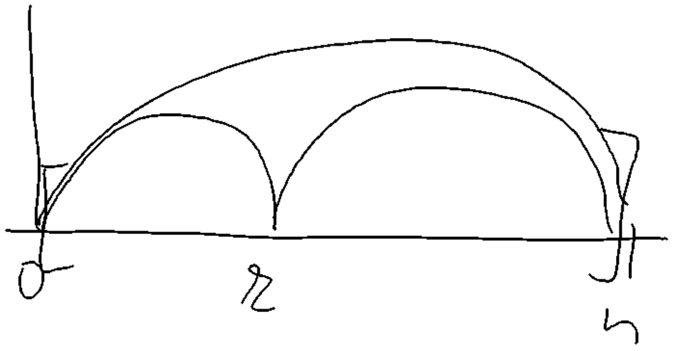
Date: Mon May 14 10:16:04 CEST 2012

Duration: 89:48 min

Pages: 82

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \ldots a_k)$.



Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \ldots a_k)$.
  Sei $u_1 := a_1 \ldots a_k$, $u_2 := a_{k+1} \ldots a_n$

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \ldots a_k)$.
  Sei $u_1 := a_1 \ldots a_k$, $u_2 := a_{k+1} \ldots a_n$
  Dann sind u_1 und u_2 balanciert:
  (1) $\quad A(u_1) = B(u_1)$ da $h(u_1) = 0$;

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.
  Sei $u_1 := a_1 \dots a_k$, $u_2 := a_{k+1} \dots a_n$
  Dann sind u_1 und u_2 balanciert:
  (1) $A(u_1) = B(u_1)$ da $h(u_1) = 0$;
  $A(u_2) = A(u) - A(u_1)$

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.
  Sei $u_1 := a_1 \dots a_k$, $u_2 := a_{k+1} \dots a_n$
  Dann sind u_1 und u_2 balanciert:
  (1) $A(u_1) = B(u_1)$ da $h(u_1) = 0$;
  $A(u_2) = A(u) - A(u_1) = B(u) - B(u_1)$

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.
  Sei $u_1 := a_1 \dots a_k$, $u_2 := a_{k+1} \dots a_n$
  Dann sind u_1 und u_2 balanciert:
  (1) $A(u_1) = B(u_1)$ da $h(u_1) = 0$;
  $A(u_2) = A(u) - A(u_1) = B(u) - B(u_1) = B(u_2)$

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.
  Sei $u_1 := a_1 \dots a_k$, $u_2 := a_{k+1} \dots a_n$
  Dann sind u_1 und u_2 balanciert:
  (1) $A(u_1) = B(u_1)$ da $h(u_1) = 0$;
  $A(u_2) = A(u) - A(u_1) = B(u) - B(u_1) = B(u_2)$
  (2) $p \preceq u_1$

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.
  Sei $u_1 := a_1 \dots a_k$, $u_2 := a_{k+1} \dots a_n$
  Dann sind u_1 und u_2 balanciert:
  (1) $A(u_1) = B(u_1)$ da $h(u_1) = 0$;
  $A(u_2) = A(u) - A(u_1) = B(u) - B(u_1) = B(u_2)$
  (2) $p \preceq u_1 \implies p \preceq u \implies A(p) \geq B(p)$
  $p \preceq u_2$: $A(p) = A(u_1 p) - A(u_1)$

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.
  Sei $u_1 := a_1 \dots a_k$, $u_2 := a_{k+1} \dots a_n$
  Dann sind u_1 und u_2 balanciert:
  (1) $A(u_1) = B(u_1)$ da $h(u_1) = 0$;
  $A(u_2) = A(u) - A(u_1) = B(u) - B(u_1) = B(u_2)$
  (2) $p \preceq u_1 \implies p \preceq u \implies A(p) \geq B(p)$
  $p \preceq u_2$: $A(p) = A(u_1 p) - A(u_1) \geq B(u_1 p) - B(u_1)$

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.
  Sei $u_1 := a_1 \dots a_k$, $u_2 := a_{k+1} \dots a_n$
  Dann sind u_1 und u_2 balanciert:
  (1) $A(u_1) = B(u_1)$ da $h(u_1) = 0$;
  $A(u_2) = A(u) - A(u_1) = B(u) - B(u_1) = B(u_2)$
  (2) $p \preceq u_1 \implies p \preceq u \implies A(p) \geq B(p)$
  $p \preceq u_2$: $A(p) = A(u_1 p) - A(u_1) \geq B(u_1 p) - B(u_1) = B(p)$

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.
  Sei $u_1 := a_1 \dots a_k$, $u_2 := a_{k+1} \dots a_n$
  Dann sind u_1 und u_2 balanciert:
  (1) $A(u_1) = B(u_1)$ da $h(u_1) = 0$;
  $A(u_2) = A(u) - A(u_1) = B(u) - B(u_1) = B(u_2)$
  (2) $p \preceq u_1 \implies p \preceq u \implies A(p) \geq B(p)$
  $p \preceq u_2$: $A(p) = A(u_1 p) - A(u_1) \geq B(u_1 p) - B(u_1) = B(p)$
  $\implies u_1, u_2 \in L_G(S)$

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.
  Sei $u_1 := a_1 \dots a_k$, $u_2 := a_{k+1} \dots a_n$
  Dann sind u_1 und u_2 balanciert:
  (1) $A(u_1) = B(u_1)$ da $h(u_1) = 0$;
  $A(u_2) = A(u) - A(u_1) = B(u) - B(u_1) = B(u_2)$
  (2) $p \preceq u_1 \implies p \preceq u \implies A(p) \geq B(p)$
  $p \preceq u_2$: $A(p) = A(u_1 p) - A(u_1) \geq B(u_1 p) - B(u_1) = B(p)$
  $\implies u_1, u_2 \in L_G(S)$ (nach IA, da $|u_i| < n$)
  $\implies u = u_1 u_2 \in L_G(S)$

$\square$

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.
  Sei $u_1 := a_1 \dots a_k$, $u_2 := a_{k+1} \dots a_n$
  Dann sind u_1 und u_2 balanciert:
  (1) $A(u_1) = B(u_1)$ da $h(u_1) = 0$;
  $A(u_2) = A(u) - A(u_1) = B(u) - B(u_1) = B(u_2)$
  (2) $p \preceq u_1 \implies p \preceq u \implies A(p) \geq B(p)$
  $p \preceq u_2$: $A(p) = A(u_1 p) - A(u_1) \geq B(u_1 p) - B(u_1) = B(p)$
  $\implies u_1, u_2 \in L_G(S)$ (nach IA, da $|u_i| < n$)
  $\implies u = u_1 u_2 \in L_G(S)$

$\square$

Moral:

- „$w \in L_G(S) \implies P(w)$“ beweist man immer schematisch mit Induktion über die Erzeugung von w.

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.

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.

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 $V = \{A_1, \ldots, A_k\}$.

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

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

ε

$n = 1$

$w_n = \varepsilon$

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

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

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

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

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 $V = \{A_1, \dots, A_k\}$. Damit ist jede Produktion von der Form

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

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

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

Aussagen der Form

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

kann man durch simultane Induktion über die Erzeugung von u beweisen,

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

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

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

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

Aussagen der Form

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

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

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

Beispiel 3.17

Grammatik:

$$\boxed{A \to \epsilon \mid aB \qquad B \to Aa}$$

Induktive Definition:

Beispiel 3.17

Grammatik:

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

Induktive Definition:

- $\epsilon \in L_G(A)$

Beispiel 3.17

Grammatik:

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

Induktive Definition:

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

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

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

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

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

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

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 $X_1, \dots, X_n \in V \cup \Sigma \cup \{\epsilon\}$ beschriftet sind, dann ist $A \to X_1 \dots X_n$ eine Produktion in P,
- ein Blatt ϵ der einzige Nachfolger seines Vorgängers ist.

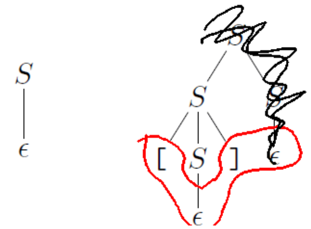
$A \to \varepsilon$

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 $X_1, \dots, X_n \in V \cup \Sigma \cup \{\epsilon\}$ beschriftet sind, dann ist $A \to X_1 \dots X_n$ eine Produktion in P,
- ein Blatt ϵ der einzige Nachfolger seines Vorgängers ist.

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



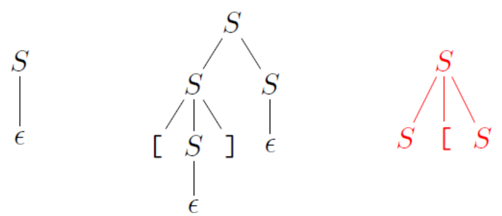
[ ]

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 $X_1, \dots, X_n \in V \cup \Sigma \cup \{\epsilon\}$ beschriftet sind, dann ist $A \to X_1 \dots X_n$ eine Produktion in P,
- ein Blatt ϵ der einzige Nachfolger seines Vorgängers ist.

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

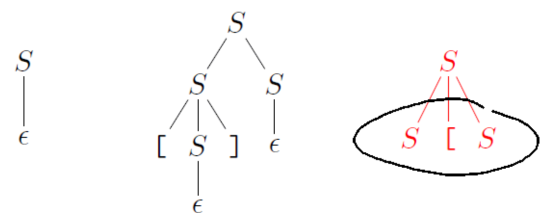


Definition 3.18 (Syntaxbaum)

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 $X_1, \dots, X_n \in V \cup \Sigma \cup \{\epsilon\}$ beschriftet sind, dann ist $A \to X_1 \dots X_n$ 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

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.* mit Wurzel A

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:

$$A \to_G w_0 A_1 w_1 \dots A_n w_n \to_G^{n-1} w$$

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:

$$A \to_G \overbrace{w_0A_1w_1\ldots A_nw_n}^{\alpha} \to_G^{n-1} \overset{\beta}{w}$$

Nach dem Dekompositionslemma muss es u_i und $n_i \leq n-1$ geben mit $A_i \to_G^{n_i} u_i$ und $w = w_0u_1w_1\ldots u_nw_n$.

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:

$$A \to_G w_0A_1w_1\ldots A_nw_n \to_G^{n-1} w$$

Nach dem Dekompositionslemma muss es u_i und $n_i \leq n-1$ geben mit $A_i \to_G^{n_i} u_i$ und $w = w_0u_1w_1\ldots u_nw_n$. Nach IA (da $n_i < n$) gilt, gilt $u_i \in L_G(A_i)$

$2 \Rightarrow 3$: 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 $A \to_G w_0A_1w_1\ldots A_nw_n \to_G^* w_0u_1w_1\ldots u_nw_n$ zusammen.

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])

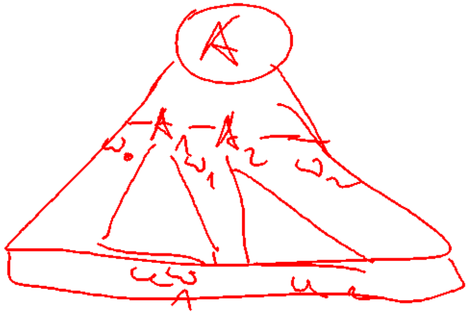
$1 \Rightarrow 2$: Sei $A \to_G^n w$. Wir konstruieren dazu einen Beweis für $w \in L_G(A)$ mit Induktion über n. Die Ableitung muss von folgender Form sein:

$$A \to_G w_0A_1w_1\ldots A_nw_n \to_G^{n-1} w$$

Nach dem Dekompositionslemma muss es u_i und $n_i \leq n-1$ geben mit $A_i \to_G^{n_i} u_i$ und $w = w_0u_1w_1\ldots u_nw_n$. Nach IA (da $n_i < n$) gilt, gilt $u_i \in L_G(A_i)$

Satz 3.20

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



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

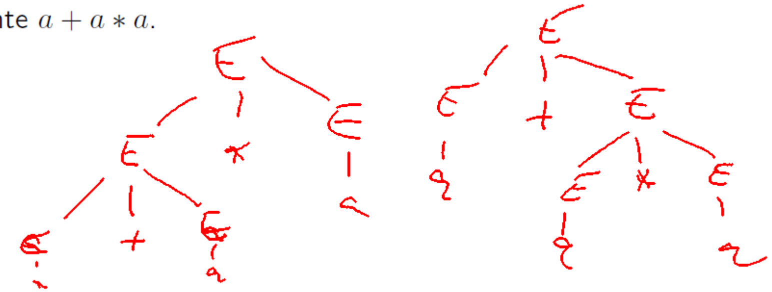
- Eine CFG G heißt mehrdeutig gdw es ein $w \in L(G)$ gibt, das zwei verschiedene Syntaxbäume hat, also zwei verschiedene Syntaxbäume mit Wurzel S und Rand w.
- Eine CFL L heißt inhärent mehrdeutig gdw jede CFG G mit $L(G) = L$ mehrdeutig ist.

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

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 $E \rightarrow E + E \mid E * E \mid (E) \mid a$ 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 $E \rightarrow E + E \mid E * E \mid (E) \mid a$ 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

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

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.

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.

S darf nicht auf rhs vorkommen!

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

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

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

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.

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.
Denn diese sind nun überflüssig:

Beweis (Forts.):

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

Beweis (Forts.):

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

$$S \rightarrow^*_{\hat{G}} \gamma B \delta \rightarrow_{\hat{G}} \gamma\delta \rightarrow^*_{\hat{G}} w$$

Beweis (Forts.):

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

$$S \rightarrow^*_{\hat{G}} \gamma B \delta \rightarrow_{\hat{G}} \gamma\delta \rightarrow^*_{\hat{G}} w$$

so wäre γ oder δ nichtleer,

Beweis (Forts.):

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

$$S \rightarrow^*_{\hat{G}} \gamma B \delta \rightarrow_{\hat{G}} \gamma\delta \rightarrow^*_{\hat{G}} w$$

so wäre γ oder δ nichtleer, dh B muss durch eine Produktion $A \rightarrow \alpha B \beta$ 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:

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

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:

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

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:

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

Also kommen in Ableitungen minimaler Länge keine ϵ-Produktionen vor.

Beispiel 3.27

$$S \to AB, \quad A \to aA \cancel{\mid \epsilon}, \quad B \to bBB \cancel{\mid \epsilon}$$

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

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

$S \to A$

~~$S \to \varepsilon$~~

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 $L(G') = L(G)$.

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.

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.

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. □

Beispiel 3.29

$$A \to a \mid \cancel{B}, \qquad B \to b \mid \cancel{C}, \qquad \cancel{C \to A}$$

$$A \to b \mid \cancel{C} \mid \cancel{A}$$

$$B \to \cancel{A} \mid a \mid \cancel{B}$$

$$C \to a \mid \cancel{B} \mid b \mid \cancel{C}$$