SS 2005

Einführung in die Informatik IV

Ernst W. Mayr

Fakultät für Informatik
TU München

http://www14.in.tum.de/lehre/2005SS/info4/index.html.de

13. Mai 2005

Definition 68

$A \in V$ heißt nutzlos, falls es keine Ableitung

$$S \to^* w, \quad w \in \Sigma^*$$

gibt, in der A vorkommt.

Satz 69

Die Menge der nutzlosen Variablen kann in Zeit $O(|V| \cdot s(G))$ bestimmt werden.

Definition 68

$A \in V$ heißt nutzlos, falls es keine Ableitung

$$S \to^* w, \quad w \in \Sigma^*$$

gibt, in der A vorkommt.

Satz 69

Die Menge der nutzlosen Variablen kann in Zeit $O(|V| \cdot s(G))$ bestimmt werden.

Beweis:

Sei V'' die Menge der nicht nutzlosen Variablen.
Offensichtlich gilt: $V'' \subseteq V'$ (V' aus dem vorigen Satz).
Falls $S \notin V'$, dann sind alle Variablen nutzlos.

Ansonsten:

$\Delta := \{S\}$; $V'' := \emptyset$;
while $\Delta \neq \emptyset$ **do**
  $V'' := V'' \cup \Delta$
  $\Delta := \{B \in V' \setminus V'';\ (\exists A \to \alpha B \beta) \in P$ mit $A \in V''$, $\alpha, \beta \in (V' \cup \Sigma)^*\}$
od

Induktion über Länge der Ableitung: Am Ende des Algorithmus ist V'' gleich der Menge der nicht nutzlosen Variablen. □

Bemerkung: Alle nutzlosen Variablen und alle Produktionen, die nutzlose Variablen enthalten, können aus der Grammatik entfernt werden, ohne die erzeugte Sprache zu ändern.

Beweis:

Sei V'' die Menge der nicht nutzlosen Variablen.
Offensichtlich gilt: $V'' \subseteq V'$ (V' aus dem vorigen Satz).
Falls $S \notin V'$, dann sind alle Variablen nutzlos.

Ansonsten:

$\Delta := \{S\};\ V'' := \emptyset;$
while $\Delta \neq \emptyset$ **do**
  $V'' := V'' \cup \Delta$
  $\Delta := \{B \in V' \setminus V'';\ (\exists A \to \alpha B \beta) \in P \text{ mit } A \in V'',\ \alpha, \beta \in (V' \cup \Sigma)^*\}$
od

Induktion über Länge der Ableitung: Am Ende des Algorithmus ist V'' gleich der Menge der nicht nutzlosen Variablen. □

Bemerkung: Alle nutzlosen Variablen und alle Produktionen, die nutzlose Variablen enthalten, können aus der Grammatik entfernt werden, ohne die erzeugte Sprache zu ändern.

Beweis:

Sei V'' die Menge der nicht nutzlosen Variablen.
Offensichtlich gilt: $V'' \subseteq V'$ (V' aus dem vorigen Satz).
Falls $S \notin V'$, dann sind alle Variablen nutzlos.

Ansonsten:

$\Delta := \{S\}$; $V'' := \emptyset$;
while $\Delta \neq \emptyset$ **do**
  $V'' := V'' \cup \Delta$
  $\Delta := \{B \in V' \setminus V'';\ (\exists A \to \alpha B \beta) \in P$ mit $A \in V'',$
  $\alpha, \beta \in (V' \cup \Sigma)^*\}$
od

Induktion über Länge der Ableitung: Am Ende des Algorithmus ist V'' gleich der Menge der nicht nutzlosen Variablen. $\square$

Bemerkung: Alle nutzlosen Variablen und alle Produktionen, die nutzlose Variablen enthalten, können aus der Grammatik entfernt werden, ohne die erzeugte Sprache zu ändern.

Beweis:

Sei V'' die Menge der nicht nutzlosen Variablen.
Offensichtlich gilt: $V'' \subseteq V'$ (V' aus dem vorigen Satz).
Falls $S \notin V'$, dann sind alle Variablen nutzlos.

Ansonsten:

$\Delta := \{S\}$; $V'' := \emptyset$;
while $\Delta \neq \emptyset$ **do**
$V'' := V'' \cup \Delta$
$\Delta := \{B \in V' \setminus V'';\ (\exists A \to \alpha B \beta) \in P$ mit $A \in V'',$
$\alpha, \beta \in (V' \cup \Sigma)^*\}$
od

Induktion über Länge der Ableitung: Am Ende des Algorithmus ist V'' gleich der Menge der nicht nutzlosen Variablen. $\square$

Bemerkung: Alle nutzlosen Variablen und alle Produktionen, die nutzlose Variablen enthalten, können aus der Grammatik entfernt werden, ohne die erzeugte Sprache zu ändern.

Beweis:

Sei V'' die Menge der nicht nutzlosen Variablen.
Offensichtlich gilt: $V'' \subseteq V'$ (V' aus dem vorigen Satz).
Falls $S \notin V'$, dann sind alle Variablen nutzlos.

Ansonsten:

$\Delta := \{S\}$; $V'' := \emptyset$;
while $\Delta \neq \emptyset$ **do**
  $V'' := V'' \cup \Delta$
  $\Delta := \{B \in V' \setminus V'';\ (\exists A \to \alpha B \beta) \in P$ mit $A \in V''$, $\alpha, \beta \in (V' \cup \Sigma)^*\}$
od

Induktion über Länge der Ableitung: Am Ende des Algorithmus ist V'' gleich der Menge der nicht nutzlosen Variablen. □

Bemerkung: Alle nutzlosen Variablen und alle Produktionen, die nutzlose Variablen enthalten, können aus der Grammatik entfernt werden, ohne die erzeugte Sprache zu ändern.

Beweis:

Sei V'' die Menge der nicht nutzlosen Variablen.
Offensichtlich gilt: $V'' \subseteq V'$ (V' aus dem vorigen Satz).
Falls $S \notin V'$, dann sind alle Variablen nutzlos.

Ansonsten:

$\Delta := \{S\}$; $V'' := \emptyset$;
while $\Delta \neq \emptyset$ **do**
$V'' := V'' \cup \Delta$
$\Delta := \{B \in V' \setminus V'';\ (\exists A \rightarrow \alpha B \beta) \in P$ mit $A \in V'',$
$\alpha, \beta \in (V' \cup \Sigma)^*\}$
od

Induktion über Länge der Ableitung: Am Ende des Algorithmus ist V'' gleich der Menge der nicht nutzlosen Variablen. $\square$

Bemerkung: Alle nutzlosen Variablen und alle Produktionen, die nutzlose Variablen enthalten, können aus der Grammatik entfernt werden, ohne die erzeugte Sprache zu ändern.

Beweis:

Sei V'' die Menge der nicht nutzlosen Variablen.
Offensichtlich gilt: $V'' \subseteq V'$ (V' aus dem vorigen Satz).
Falls $S \notin V'$, dann sind alle Variablen nutzlos.

Ansonsten:

$\Delta := \{S\}$; $V'' := \emptyset$;
while $\Delta \neq \emptyset$ **do**
$V'' := V'' \cup \Delta$
$$\Delta := \{B \in V' \setminus V'';\ (\exists A \to \alpha B \beta) \in P \text{ mit } A \in V'',\ \alpha, \beta \in (V' \cup \Sigma)^*\}$$
od

Induktion über Länge der Ableitung: Am Ende des Algorithmus ist V'' gleich der Menge der nicht nutzlosen Variablen. □

Bemerkung: Alle nutzlosen Variablen und alle Produktionen, die nutzlose Variablen enthalten, können aus der Grammatik entfernt werden, ohne die erzeugte Sprache zu ändern.

Beweis:

Sei V'' die Menge der nicht nutzlosen Variablen.
Offensichtlich gilt: $V'' \subseteq V'$ (V' aus dem vorigen Satz).
Falls $S \notin V'$, dann sind alle Variablen nutzlos.

Ansonsten:

$\Delta := \{S\}$; $V'' := \emptyset$;
while $\Delta \neq \emptyset$ **do**
$V'' := V'' \cup \Delta$
$\Delta := \{B \in V' \setminus V'';\ (\exists A \to \alpha B \beta) \in P$ mit $A \in V''$, $\alpha, \beta \in (V' \cup \Sigma)^*\}$
od

Induktion über Länge der Ableitung: Am Ende des Algorithmus ist V'' gleich der Menge der nicht nutzlosen Variablen. □

Bemerkung: Alle nutzlosen Variablen und alle Produktionen, die nutzlose Variablen enthalten, können aus der Grammatik entfernt werden, ohne die erzeugte Sprache zu ändern.

Beweis:

Sei V'' die Menge der nicht nutzlosen Variablen.
Offensichtlich gilt: $V'' \subseteq V'$ (V' aus dem vorigen Satz).
Falls $S \notin V'$, dann sind alle Variablen nutzlos.

Ansonsten:

$\Delta := \{S\}$; $V'' := \emptyset$;
while $\Delta \neq \emptyset$ **do**
$\quad V'' := V'' \cup \Delta$
$\quad \Delta := \{B \in V' \setminus V'';\ (\exists A \to \alpha B \beta) \in P \text{ mit } A \in V'',\ \alpha, \beta \in (V' \cup \Sigma)^*\}$
od

Induktion über Länge der Ableitung: Am Ende des Algorithmus ist V'' gleich der Menge der nicht nutzlosen Variablen. □

Bemerkung: Alle nutzlosen Variablen und alle Produktionen, die nutzlose Variablen enthalten, können aus der Grammatik entfernt werden, ohne die erzeugte Sprache zu ändern.

Korollar 70

Für eine kontextfreie Grammatik G kann in Zeit $O(|V| \cdot s(G))$ entschieden werden, ob $L(G) = \emptyset$.

Beweis:

$L(G) = \emptyset \iff S \notin V''$ (bzw. $S \notin V'$) $\square$

Korollar 70

Für eine kontextfreie Grammatik G *kann in Zeit* $O(|V| \cdot s(G))$ *entschieden werden, ob* $L(G) = \emptyset$.

Beweis:

$L(G) = \emptyset \Longleftrightarrow S \notin V''$ (bzw. $S \notin V'$) $\square$

Satz 71

Für eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ ohne nutzlose Variablen und in Chomsky-Normalform kann in linearer Zeit entschieden werden, ob

$$|L(G)| < \infty\,.$$

Beweis:

Definiere gerichteten Hilfsgraphen mit Knotenmenge V und

$$\text{Kante } A \to B \iff (A \to BC) \text{ oder } (A \to CB) \in P\,.$$

$L(G)$ ist endlich $\iff$ dieser Graph enthält keinen Zyklus.

Verwende DFS, um in linearer Zeit festzustellen, ob der Graph Zyklen enthält. □

Satz 71

Für eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ ohne nutzlose Variablen und in Chomsky-Normalform kann in linearer Zeit entschieden werden, ob

$$|L(G)| < \infty\,.$$

Beweis:

Definiere gerichteten Hilfsgraphen mit Knotenmenge V und

$$\text{Kante } A \to B \iff (A \to BC) \text{ oder } (A \to CB) \in P\,.$$

$L(G)$ ist endlich $\iff$ dieser Graph enthält keinen Zyklus.

Verwende DFS, um in linearer Zeit festzustellen, ob der Graph Zyklen enthält. □

Satz 71

Für eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ ohne nutzlose Variablen und in Chomsky-Normalform kann in linearer Zeit entschieden werden, ob

$$|L(G)| < \infty\,.$$

Beweis:

Definiere gerichteten Hilfsgraphen mit Knotenmenge V und

$$\text{Kante } A \to B \iff (A \to BC) \text{ oder } (A \to CB) \in P\,.$$

$L(G)$ ist endlich $\iff$ dieser Graph enthält keinen Zyklus.

Verwende DFS, um in linearer Zeit festzustellen, ob der Graph Zyklen enthält. □

Satz 72

Seien kontextfreie Grammatiken $G_1 = (V_1, \Sigma_1, P_1, S_1)$ *und* $G_2 = (V_2, \Sigma_2, P_2, S_2)$ *gegeben. Dann können in linearer Zeit kontextfreie Grammatiken für*

1. $L(G_1) \cup L(G_2)$,
2. $L(G_1)L(G_2)$,
3. $(L(G_1))^*$

konstruiert werden.

Satz 72

Seien kontextfreie Grammatiken $G_1 = (V_1, \Sigma_1, P_1, S_1)$ *und* $G_2 = (V_2, \Sigma_2, P_2, S_2)$ *gegeben. Dann können in linearer Zeit kontextfreie Grammatiken für*

1. $L(G_1) \cup L(G_2)$,
2. $L(G_1)L(G_2)$,
3. $(L(G_1))^*$

konstruiert werden.

Satz 72

Seien kontextfreie Grammatiken $G_1 = (V_1, \Sigma_1, P_1, S_1)$ *und* $G_2 = (V_2, \Sigma_2, P_2, S_2)$ *gegeben. Dann können in linearer Zeit kontextfreie Grammatiken für*

1. $L(G_1) \cup L(G_2)$,
2. $L(G_1)L(G_2)$,
3. $(L(G_1))^*$

konstruiert werden.

Satz 72

Seien kontextfreie Grammatiken $G_1 = (V_1, \Sigma_1, P_1, S_1)$ *und* $G_2 = (V_2, \Sigma_2, P_2, S_2)$ *gegeben. Dann können in linearer Zeit kontextfreie Grammatiken für*

1. $L(G_1) \cup L(G_2)$,
2. $L(G_1)L(G_2)$,
3. $(L(G_1))^*$

konstruiert werden.

Satz 72

Seien kontextfreie Grammatiken $G_1 = (V_1, \Sigma_1, P_1, S_1)$ *und* $G_2 = (V_2, \Sigma_2, P_2, S_2)$ *gegeben. Dann können in linearer Zeit kontextfreie Grammatiken für*

1. $L(G_1) \cup L(G_2)$,
2. $L(G_1)L(G_2)$,
3. $(L(G_1))^*$

konstruiert werden. Die Klasse der kontextfreien Sprachen ist also unter Vereinigung, Konkatenation und Kleene'scher Hülle abgeschlossen.

Beweis:

Ohne Beschränkung der Allgemeinheit nehmen wir an, dass $V_1 \cap V_2 = \emptyset$.

1. $V = V_1 \cup V_2 \cup \{S\}$; S neu
   $P = P_1 \cup P_2 \cup \{S \to S_1 | S_2\}$
2. $V = V_1 \cup V_2 \cup \{S\}$; S neu
   $P = P_1 \cup P_2 \cup \{S \to S_1 S_2\}$
3. $V = V_1 \cup \{S\}$; S neu
   $P = P_1 \cup \{S \to SS | S_1 | \epsilon\}$

Falls $\epsilon \in L(G_1)$ oder $\epsilon \in L(G_2)$, sind noch Korrekturen vorzunehmen, die hier als Übungsaufgabe überlassen bleiben. $\square$

Beweis:

Ohne Beschränkung der Allgemeinheit nehmen wir an, dass $V_1 \cap V_2 = \emptyset$.

1. $V = V_1 \cup V_2 \cup \{S\}$; S neu
   $P = P_1 \cup P_2 \cup \{S \to S_1 | S_2\}$
2. $V = V_1 \cup V_2 \cup \{S\}$; S neu
   $P = P_1 \cup P_2 \cup \{S \to S_1 S_2\}$
3. $V = V_1 \cup \{S\}$; S neu
   $P = P_1 \cup \{S \to SS | S_1 | \epsilon\}$

Falls $\epsilon \in L(G_1)$ oder $\epsilon \in L(G_2)$, sind noch Korrekturen vorzunehmen, die hier als Übungsaufgabe überlassen bleiben. $\square$

Beweis:

Ohne Beschränkung der Allgemeinheit nehmen wir an, dass $V_1 \cap V_2 = \emptyset$.

1. $V = V_1 \cup V_2 \cup \{S\}$; S neu
   $P = P_1 \cup P_2 \cup \{S \to S_1 | S_2\}$
2. $V = V_1 \cup V_2 \cup \{S\}$; S neu
   $P = P_1 \cup P_2 \cup \{S \to S_1 S_2\}$
3. $V = V_1 \cup \{S\}$; S neu
   $P = P_1 \cup \{S \to SS | S_1 | \epsilon\}$

Falls $\epsilon \in L(G_1)$ oder $\epsilon \in L(G_2)$, sind noch Korrekturen vorzunehmen, die hier als Übungsaufgabe überlassen bleiben. $\square$

Beweis:

Ohne Beschränkung der Allgemeinheit nehmen wir an, dass $V_1 \cap V_2 = \emptyset$.

1. $V = V_1 \cup V_2 \cup \{S\}$; S neu
   $P = P_1 \cup P_2 \cup \{S \rightarrow S_1 | S_2\}$
2. $V = V_1 \cup V_2 \cup \{S\}$; S neu
   $P = P_1 \cup P_2 \cup \{S \rightarrow S_1 S_2\}$
3. $V = V_1 \cup \{S\}$; S neu
   $P = P_1 \cup \{S \rightarrow SS | S_1 | \epsilon\}$

Falls $\epsilon \in L(G_1)$ oder $\epsilon \in L(G_2)$, sind noch Korrekturen vorzunehmen, die hier als Übungsaufgabe überlassen bleiben. $\square$

Beweis:

Ohne Beschränkung der Allgemeinheit nehmen wir an, dass $V_1 \cap V_2 = \emptyset$.

1. $V = V_1 \cup V_2 \cup \{S\}$; S neu
   $P = P_1 \cup P_2 \cup \{S \rightarrow S_1 | S_2\}$
2. $V = V_1 \cup V_2 \cup \{S\}$; S neu
   $P = P_1 \cup P_2 \cup \{S \rightarrow S_1 S_2\}$
3. $V = V_1 \cup \{S\}$; S neu
   $P = P_1 \cup \{S \rightarrow SS | S_1 | \epsilon\}$

Falls $\epsilon \in L(G_1)$ oder $\epsilon \in L(G_2)$, sind noch Korrekturen vorzunehmen, die hier als Übungsaufgabe überlassen bleiben. $\square$

Satz 73

Die Klasse der kontextfreien Sprachen ist nicht abgeschlossen unter Durchschnitt oder Komplement.

Beweis:

Es genügt zu zeigen (wegen de Morgan): nicht abgeschlossen unter Durchschnitt.

$$L_1 := \{a^i b^i c^j;\ i,j \geq 0\} \text{ ist kontextfrei}$$

$$L_2 := \{a^i b^j c^j;\ i,j \geq 0\} \text{ ist kontextfrei}$$

$$L_1 \cap L_2 = \{a^i b^i c^i;\ i \geq 0\} \text{ ist nicht kontextfrei}$$

□

Satz 73

Die Klasse der kontextfreien Sprachen ist nicht abgeschlossen unter Durchschnitt oder Komplement.

Beweis:

Es genügt zu zeigen (wegen de Morgan): nicht abgeschlossen unter Durchschnitt.

$$L_1 := \{a^i b^i c^j;\ i,j \geq 0\} \text{ ist kontextfrei}$$

$$L_2 := \{a^i b^j c^j;\ i,j \geq 0\} \text{ ist kontextfrei}$$

$$L_1 \cap L_2 = \{a^i b^i c^i;\ i \geq 0\} \text{ ist nicht kontextfrei}$$

□

Satz 73

Die Klasse der kontextfreien Sprachen ist nicht abgeschlossen unter Durchschnitt oder Komplement.

Beweis:

Es genügt zu zeigen (wegen de Morgan): nicht abgeschlossen unter Durchschnitt.

$$L_1 := \{a^i b^i c^j;\ i, j \geq 0\} \text{ ist kontextfrei}$$

$$L_2 := \{a^i b^j c^j;\ i, j \geq 0\} \text{ ist kontextfrei}$$

$$L_1 \cap L_2 = \{a^i b^i c^i;\ i \geq 0\} \text{ ist nicht kontextfrei}$$

□

Satz 73

Die Klasse der kontextfreien Sprachen ist nicht abgeschlossen unter Durchschnitt oder Komplement.

Beweis:

Es genügt zu zeigen (wegen de Morgan): nicht abgeschlossen unter Durchschnitt.

$$L_1 := \{a^i b^i c^j;\ i, j \geq 0\} \text{ ist kontextfrei}$$

$$L_2 := \{a^i b^j c^j;\ i, j \geq 0\} \text{ ist kontextfrei}$$

$$L_1 \cap L_2 = \{a^i b^i c^i;\ i \geq 0\} \text{ ist nicht kontextfrei}$$

$\square$

Satz 73

Die Klasse der kontextfreien Sprachen ist nicht abgeschlossen unter Durchschnitt oder Komplement.

Beweis:

Es genügt zu zeigen (wegen de Morgan): nicht abgeschlossen unter Durchschnitt.

$$L_1 := \{a^i b^i c^j;\ i, j \geq 0\} \text{ ist kontextfrei}$$
$$L_2 := \{a^i b^j c^j;\ i, j \geq 0\} \text{ ist kontextfrei}$$
$$L_1 \cap L_2 = \{a^i b^i c^i;\ i \geq 0\} \text{ ist nicht kontextfrei}$$

□

Satz 74

Die Klasse der kontextfreien Sprachen ist abgeschlossen gegenüber Substitution (mit kontextfreien Mengen).

Beweis:

Ersetze jedes Terminal a durch ein neues Nichtterminal S_a und füge zu den Produktionen P für jedes Terminal a die Produktionen einer kontextfreien Grammatik $G_a = (V_a, \Sigma, P_a, S_a)$ hinzu.
Forme die so erhaltene Grammatik in eine äquivalente Chomsky-2-Grammatik um. $\square$

Satz 74

Die Klasse der kontextfreien Sprachen ist abgeschlossen gegenüber Substitution (mit kontextfreien Mengen).

Beweis:

Ersetze jedes Terminal a durch ein neues Nichtterminal S_a und füge zu den Produktionen P für jedes Terminal a die Produktionen einer kontextfreien Grammatik $G_a = (V_a, \Sigma, P_a, S_a)$ hinzu.
Forme die so erhaltene Grammatik in eine äquivalente Chomsky-2-Grammatik um. □

Satz 74

Die Klasse der kontextfreien Sprachen ist abgeschlossen gegenüber Substitution (mit kontextfreien Mengen).

Beweis:

Ersetze jedes Terminal a durch ein neues Nichtterminal S_a und füge zu den Produktionen P für jedes Terminal a die Produktionen einer kontextfreien Grammatik $G_a = (V_a, \Sigma, P_a, S_a)$ hinzu.
Forme die so erhaltene Grammatik in eine äquivalente Chomsky-2-Grammatik um. □

4.6 Greibach-Normalform

Definition 75

Sei $G = (V, \Sigma, P, S)$ eine kontextfreie Grammatik. G ist in Greibach-Normalform (benannt nach Sheila Greibach), falls jede Produktion $\neq S \to \epsilon$ von der Form

$$A \to a\alpha \text{ mit } a \in \Sigma,\ \alpha \in V^*$$

ist.

Lemma 76

Sei $G = (V, \Sigma, P, S)$ kontextfrei, $(A \to \alpha_1 B \alpha_2) \in P$, und sei $B \to \beta_1 | \beta_2 | \ldots | \beta_r$ die Menge der B-Produktionen (also die Menge der Produktionen mit B auf der linken Seite). Ersetzt man $A \to \alpha_1 B \alpha_2$ durch $A \to \alpha_1\beta_1\alpha_2 | \alpha_1\beta_2\alpha_2 | \ldots | \alpha_1\beta_r\alpha_2$, so ändert sich die von der Grammatik erzeugte Sprache nicht.

4.6 Greibach-Normalform

Definition 75

Sei $G = (V, \Sigma, P, S)$ eine kontextfreie Grammatik. G ist in Greibach-Normalform (benannt nach Sheila Greibach), falls jede Produktion $\neq S \to \epsilon$ von der Form

$$A \to a\alpha \text{ mit } a \in \Sigma,\ \alpha \in V^*$$

ist.

Lemma 76

Sei $G = (V, \Sigma, P, S)$ kontextfrei, $(A \to \alpha_1 B \alpha_2) \in P$, und sei $B \to \beta_1|\beta_2|\dots|\beta_r$ die Menge der B-Produktionen (also die Menge der Produktionen mit B auf der linken Seite). Ersetzt man $A \to \alpha_1 B \alpha_2$ durch $A \to \alpha_1\beta_1\alpha_2|\alpha_1\beta_2\alpha_2|\dots|\alpha_1\beta_r\alpha_2$, so ändert sich die von der Grammatik erzeugte Sprache nicht.

Lemma 77

Sei $G = (V, \Sigma, P, S)$ kontextfrei, sei $A \to A\alpha_1|A\alpha_2|\ldots|A\alpha_r$ die Menge der linksrekursiven A-Produktionen (alle $\alpha_i \neq \epsilon$, die Produktion $A \to A$ kommt o.B.d.A. nicht vor), und seien $A \to \beta_1|\beta_2|\ldots|\beta_s$ die restlichen A-Produktionen (ebenfalls alle $\beta_i \neq \epsilon$).

Ersetzen wir alle A-Produktionen durch

$$A \to \beta_1|\ldots|\beta_s|\beta_1 A'|\ldots|\beta_s A'$$
$$A' \to \alpha_1|\ldots|\alpha_r|\alpha_1 A'|\ldots|\alpha_r A',$$

wobei A' ein neues Nichtterminal ist, so ändert sich die Sprache nicht, und die neue Grammatik enthält keine linksrekursive A-Produktion mehr.

Beweis:

Von A lassen sich vor der Transformation alle Zeichenreihen der Form

$$(\beta_1|\beta_2|\ldots|\beta_s)(\alpha_1|\alpha_2|\ldots|\alpha_r)^+$$

ableiten.

Dies ist auch nach der Transformation der Fall. Während vor der Transformation alle Zeichenreihen der obigen Form von rechts her aufgebaut werden, werden sie danach von links nach rechts erzeugt.

Die Umkehrung gilt ebenso. □

Beweis:

Von A lassen sich vor der Transformation alle Zeichenreihen der Form

$$(\beta_1|\beta_2|\dots|\beta_s)(\alpha_1|\alpha_2|\dots|\alpha_r)^+$$

ableiten.

Dies ist auch nach der Transformation der Fall. Während vor der Transformation alle Zeichenreihen der obigen Form von rechts her aufgebaut werden, werden sie danach von links nach rechts erzeugt.

Die Umkehrung gilt ebenso. □

Beweis:

Von A lassen sich vor der Transformation alle Zeichenreihen der Form

$$(\beta_1|\beta_2|\dots|\beta_s)(\alpha_1|\alpha_2|\dots|\alpha_r)^+$$

ableiten.

Dies ist auch nach der Transformation der Fall. Während vor der Transformation alle Zeichenreihen der obigen Form von rechts her aufgebaut werden, werden sie danach von links nach rechts erzeugt.

Die Umkehrung gilt ebenso. □

Satz 78

Zu jeder kontextfreien Grammatik G gibt es eine äquivalente Grammatik in Greibach-Normalform.

Beweis:

Sei o.B.d.A. $G = (V, \Sigma, P, S)$ mit $V = \{A_1, \ldots, A_m\}$ in Chomsky-Normalform und enthalte keine nutzlosen Variablen.

Beweis:

Sei o.B.d.A. $G = (V, \Sigma, P, S)$ mit $V = \{A_1, \ldots, A_m\}$ in Chomsky-Normalform und enthalte keine nutzlosen Variablen.

Bemerkung: Im folgenden Algorithmus werden ggf neue Variablen hinzugefügt, die Programmvariable m ändert sich dadurch entsprechend!

Beweis:

for $k = 1, \ldots, m$ **do**
  for $j = 1, \ldots, k-1$ **do**
    for all $(A_k \to A_j\alpha) \in P$ **do**
      ersetze die Produktion gemäß der Konstruktion
      in Lemma 76
    od
  od
  co die rechte Seite keiner $A_{k'}$-Produktion, $k' < k$, beginnt
    nun noch mit einer Variablen A_j, $j < k$ **oc**

  ersetze alle linksrekursiven A_k-Produktionen gemäß der
  Konstruktion in Lemma 77
  co die rechte Seite keiner $A_{k'}$-Produktion, $k' \leq k$, beginnt
    nun noch mit einer Variablen A_j, $j \leq k$ **oc**
od

□

Beweis:

for $k = 1, \ldots, m$ **do**
for $j = 1, \ldots, k-1$ **do**
for all $(A_k \to A_j\alpha) \in P$ **do**
ersetze die Produktion gemäß der Konstruktion in Lemma 76
od
od
co die rechte Seite keiner $A_{k'}$-Produktion, $k' < k$, beginnt nun noch mit einer Variablen A_j, $j < k$ **oc**

ersetze alle linksrekursiven A_k-Produktionen gemäß der Konstruktion in Lemma 77
co die rechte Seite keiner $A_{k'}$-Produktion, $k' \leq k$, beginnt nun noch mit einer Variablen A_j, $j \leq k$ **oc**
od

$\square$

Beweis:

for $k = 1, \ldots, m$ **do**
  for $j = 1, \ldots, k-1$ **do**
    for all $(A_k \to A_j\alpha) \in P$ **do**
      ersetze die Produktion gemäß der Konstruktion
      in Lemma 76
    od
  od
  co die rechte Seite keiner $A_{k'}$-Produktion, $k' < k$, beginnt
    nun noch mit einer Variablen A_j, $j < k$ **oc**

  ersetze alle linksrekursiven A_k-Produktionen gemäß der
  Konstruktion in Lemma 77
  co die rechte Seite keiner $A_{k'}$-Produktion, $k' \leq k$, beginnt
    nun noch mit einer Variablen A_j, $j \leq k$ **oc**
od

□

Beweis:

for $k = 1, \ldots, m$ **do**
  for $j = 1, \ldots, k-1$ **do**
    for all $(A_k \rightarrow A_j\alpha) \in P$ **do**
      ersetze die Produktion gemäß der Konstruktion
      in Lemma 76
    od
  od
  co die rechte Seite keiner $A_{k'}$-Produktion, $k' < k$, beginnt
    nun noch mit einer Variablen A_j, $j < k$ **oc**

  ersetze alle linksrekursiven A_k-Produktionen gemäß der
  Konstruktion in Lemma 77
  co die rechte Seite keiner $A_{k'}$-Produktion, $k' \leq k$, beginnt
    nun noch mit einer Variablen A_j, $j \leq k$ **oc**
od

□

Beweis:

for $k = 1, \ldots, m$ **do**
  for $j = 1, \ldots, k-1$ **do**
    for all $(A_k \rightarrow A_j\alpha) \in P$ **do**
      ersetze die Produktion gemäß der Konstruktion
      in Lemma 76
    od
  od
  co die rechte Seite keiner $A_{k'}$-Produktion, $k' < k$, beginnt
    nun noch mit einer Variablen A_j, $j < k$ **oc**

  ersetze alle linksrekursiven A_k-Produktionen gemäß der
  Konstruktion in Lemma 77
  co die rechte Seite keiner $A_{k'}$-Produktion, $k' \leq k$, beginnt
    nun noch mit einer Variablen A_j, $j \leq k$ **oc**
od

□

Beweis:

for $k = 1, \ldots, m$ **do**
  for $j = 1, \ldots, k-1$ **do**
    for all $(A_k \rightarrow A_j\alpha) \in P$ **do**
      ersetze die Produktion gemäß der Konstruktion
      in Lemma 76
    od
  od
  co die rechte Seite keiner $A_{k'}$-Produktion, $k' < k$, beginnt
    nun noch mit einer Variablen A_j, $j < k$ **oc**

  ersetze alle linksrekursiven A_k-Produktionen gemäß der
  Konstruktion in Lemma 77
  co die rechte Seite keiner $A_{k'}$-Produktion, $k' \leq k$, beginnt
    nun noch mit einer Variablen A_j, $j \leq k$ **oc**
od

□

Beweis:

for $k = 1, \ldots, m$ **do**
  for $j = 1, \ldots, k-1$ **do**
    for all $(A_k \to A_j\alpha) \in P$ **do**
      ersetze die Produktion gemäß der Konstruktion
      in Lemma 76
    od
  od
  co die rechte Seite keiner $A_{k'}$-Produktion, $k' < k$, beginnt
    nun noch mit einer Variablen A_j, $j < k$ **oc**

  ersetze alle linksrekursiven A_k-Produktionen gemäß der
  Konstruktion in Lemma 77
  co die rechte Seite keiner $A_{k'}$-Produktion, $k' \leq k$, beginnt
    nun noch mit einer Variablen A_j, $j \leq k$ **oc**
od

□