

6a

Die Greibach-Normalform und Eigenschaften von kontextfreien Sprachen

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik und Theorembeweisen

Stand: 4. Juni 2024
Basierend auf Folien von PD Dr. David Sabel



Satz

Das Leerheitsproblem für kontextfreie Grammatiken ist entscheidbar.

Satz

Das Leerheitsproblem für kontextfreie Grammatiken ist entscheidbar.

Beweis Sei $G = (V, \Sigma, P, S)$ eine CFG mit 1. Sonderregel.

Satz

Das Leerheitsproblem für kontextfreie Grammatiken ist entscheidbar.

Beweis Sei $G = (V, \Sigma, P, S)$ eine CFG mit 1. Sonderregel.

Prüfe zunächst, ob $S \rightarrow \varepsilon \in P$. Wenn ja, dann ist $L(G)$ nicht leer.

Satz

Das Leerheitsproblem für kontextfreie Grammatiken ist entscheidbar.

Beweis Sei $G = (V, \Sigma, P, S)$ eine CFG mit 1. Sonderregel.

Prüfe zunächst, ob $S \rightarrow \varepsilon \in P$. Wenn ja, dann ist $L(G)$ nicht leer.

Sonst: Sei $G' = (V', \Sigma, P', S')$ eine CFG in Chomsky-Normalform mit $L(G') = L(G)$.

Satz

Das Leerheitsproblem für kontextfreie Grammatiken ist entscheidbar.

Beweis Sei $G = (V, \Sigma, P, S)$ eine CFG mit 1. Sonderregel.

Prüfe zunächst, ob $S \rightarrow \varepsilon \in P$. Wenn ja, dann ist $L(G)$ nicht leer.

Sonst: Sei $G' = (V', \Sigma, P', S')$ eine CFG in Chomsky-Normalform mit $L(G') = L(G)$.

Der folgende Algorithmus markiert alle $A \in V'$ mit $\{w \in \Sigma^* \mid A \Rightarrow_{G'}^* w\} \neq \emptyset$.

Satz

Das Leerheitsproblem für kontextfreie Grammatiken ist entscheidbar.

Beweis Sei $G = (V, \Sigma, P, S)$ eine CFG mit 1. Sonderregel.

Prüfe zunächst, ob $S \rightarrow \varepsilon \in P$. Wenn ja, dann ist $L(G)$ nicht leer.

Sonst: Sei $G' = (V', \Sigma, P', S')$ eine CFG in Chomsky-Normalform mit $L(G') = L(G)$.

Der folgende Algorithmus markiert alle $A \in V'$ mit $\{w \in \Sigma^* \mid A \Rightarrow_{G'}^* w\} \neq \emptyset$.

Prüfe, ob S' markiert wird. Wenn ja, dann ist $L(G) = L(G')$ nicht leer. \square

Algorithmus 9: Markierung der „nichtleeren“ Variablen

Eingabe: Grammatik $G = (V, \Sigma, P, S)$ in Chomsky-Normalform

Ausgabe: Menge $W \subseteq V$ aller Variablen, die nicht die leere Sprache erzeugen

Beginn

$W := \{A \in V \mid A \rightarrow a \in P, a \in \Sigma\};$

wiederhole

$W_{\text{alt}} := W;$

$W := W_{\text{alt}} \cup \{A \mid A \rightarrow BC \in P, B \in W_{\text{alt}}, C \in W_{\text{alt}}\};$

bis $W = W_{\text{alt}};$

return W

Beispiel für die Markierung der „nichtleeren“ Variablen

Sei $G = (\{S, X, Y, Z\}, \{a, b\}, \{S \rightarrow XY, Z \rightarrow ZZ \mid a \mid b, X \rightarrow ZZ \mid YY, Y \rightarrow ZY\})$.

Beispiel für die Markierung der „nichtleeren“ Variablen

Sei $G = (\{S, X, Y, Z\}, \{a, b\}, \{S \rightarrow XY, Z \rightarrow ZZ \mid a \mid b, X \rightarrow ZZ \mid YY, Y \rightarrow ZY\})$.

Ausführung von Algorithmus 9:

$$0. W := \{A \in V \mid A \rightarrow a \in P\} = \{Z\}$$

Beispiel für die Markierung der „nichtleeren“ Variablen

Sei $G = (\{S, X, Y, Z\}, \{a, b\}, \{S \rightarrow XY, Z \rightarrow ZZ \mid a \mid b, X \rightarrow ZZ \mid YY, Y \rightarrow ZY\})$.

Ausführung von Algorithmus 9:

0. $W := \{A \in V \mid A \rightarrow a \in P\} = \{Z\}$

1. $W_{\text{alt}} := W = \{Z\}$

$$W := W_{\text{alt}} \cup \{A \mid A \rightarrow BC \in P, B, C \in W_{\text{alt}}\} = \{Z\} \cup \{X, Z\} = \{X, Z\}$$

$$W \neq W_{\text{alt}}$$

Beispiel für die Markierung der „nichtleeren“ Variablen

Sei $G = (\{S, X, Y, Z\}, \{a, b\}, \{S \rightarrow XY, Z \rightarrow ZZ \mid a \mid b, X \rightarrow ZZ \mid YY, Y \rightarrow ZY\})$.

Ausführung von Algorithmus 9:

0. $W := \{A \in V \mid A \rightarrow a \in P\} = \{Z\}$

1. $W_{\text{alt}} := W = \{Z\}$

$$W := W_{\text{alt}} \cup \{A \mid A \rightarrow BC \in P, B, C \in W_{\text{alt}}\} = \{Z\} \cup \{X, Z\} = \{X, Z\}$$

$$W \neq W_{\text{alt}}$$

2. $W_{\text{alt}} := W = \{X, Z\}$

$$W := W_{\text{alt}} \cup \{A \mid A \rightarrow BC \in P, B, C \in W_{\text{alt}}\} = \{X, Z\} \cup \{X, Z\} = \{X, Z\}$$

$$W = W_{\text{alt}}$$

Beispiel für die Markierung der „nichtleeren“ Variablen

Sei $G = (\{S, X, Y, Z\}, \{a, b\}, \{S \rightarrow XY, Z \rightarrow ZZ \mid a \mid b, X \rightarrow ZZ \mid YY, Y \rightarrow ZY\})$.

Ausführung von Algorithmus 9:

0. $W := \{A \in V \mid A \rightarrow a \in P\} = \{Z\}$

1. $W_{\text{alt}} := W = \{Z\}$

$$W := W_{\text{alt}} \cup \{A \mid A \rightarrow BC \in P, B, C \in W_{\text{alt}}\} = \{Z\} \cup \{X, Z\} = \{X, Z\}$$

$$W \neq W_{\text{alt}}$$

2. $W_{\text{alt}} := W = \{X, Z\}$

$$W := W_{\text{alt}} \cup \{A \mid A \rightarrow BC \in P, B, C \in W_{\text{alt}}\} = \{X, Z\} \cup \{X, Z\} = \{X, Z\}$$

$$W = W_{\text{alt}}$$

3. return $W = \{X, Z\}$

Beispiel für die Markierung der „nichtleeren“ Variablen

Sei $G = (\{S, X, Y, Z\}, \{a, b\}, \{S \rightarrow XY, Z \rightarrow ZZ \mid a \mid b, X \rightarrow ZZ \mid YY, Y \rightarrow ZY\})$.

Ausführung von Algorithmus 9:

0. $W := \{A \in V \mid A \rightarrow a \in P\} = \{Z\}$

1. $W_{\text{alt}} := W = \{Z\}$

$$W := W_{\text{alt}} \cup \{A \mid A \rightarrow BC \in P, B, C \in W_{\text{alt}}\} = \{Z\} \cup \{X, Z\} = \{X, Z\}$$

$$W \neq W_{\text{alt}}$$

2. $W_{\text{alt}} := W = \{X, Z\}$

$$W := W_{\text{alt}} \cup \{A \mid A \rightarrow BC \in P, B, C \in W_{\text{alt}}\} = \{X, Z\} \cup \{X, Z\} = \{X, Z\}$$

$$W = W_{\text{alt}}$$

3. return $W = \{X, Z\}$

Da $S \notin W$ folgt, dass G die leere Sprache erzeugt.

Definition

Eine CFG $G = (V, \Sigma, P, S)$ ist in [Greibach-Normalform](#), falls alle Produktionen in P von der Form $A \rightarrow aB_1B_2 \dots B_j$ sind, mit $j \geq 0$, $a \in \Sigma$ und $A, B_1, \dots, B_j \in V$.

Definition

Eine CFG $G = (V, \Sigma, P, S)$ ist in [Greibach-Normalform](#), falls alle Produktionen in P von der Form $A \rightarrow aB_1B_2 \dots B_j$ sind, mit $j \geq 0$, $a \in \Sigma$ und $A, B_1, \dots, B_j \in V$.

Die Normalform ist benannt nach Sheila A. Greibach.

Definition

Eine CFG $G = (V, \Sigma, P, S)$ ist in **Greibach-Normalform**, falls alle Produktionen in P von der Form $A \rightarrow aB_1B_2 \dots B_j$ sind, mit $j \geq 0$, $a \in \Sigma$ und $A, B_1, \dots, B_j \in V$.

Die Normalform ist benannt nach Sheila A. Greibach.

Reguläre Grammatiken sind ein Spezialfall der Greibach-Normalform:

Dort ist nur $j \in \{0, 1\}$ erlaubt.

Definition

Eine CFG $G = (V, \Sigma, P, S)$ ist in **Greibach-Normalform**, falls alle Produktionen in P von der Form $A \rightarrow aB_1B_2 \dots B_j$ sind, mit $j \geq 0$, $a \in \Sigma$ und $A, B_1, \dots, B_j \in V$.

Die Normalform ist benannt nach Sheila A. Greibach.

Reguläre Grammatiken sind ein Spezialfall der Greibach-Normalform:

Dort ist nur $j \in \{0, 1\}$ erlaubt.

Die Greibach-Normalform wird u.a. verwendet, um zu zeigen, dass kontextfreie Sprachen genau von den nichtdeterministischen Kellerautomaten erkannt werden (nächste Vorlesung).

Herstellen der Greibach-Normalform

Satz

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

Herstellen der Greibach-Normalform

Grundgedanke der Prozedur:

1. Bringe die Eingabe in Chomsky-Normalform.

Herstellen der Greibach-Normalform

Grundgedanke der Prozedur:

1. Bringe die Eingabe in Chomsky-Normalform.
2. Nummeriere die Variablen A_1, \dots, A_n durch.

Herstellen der Greibach-Normalform

Grundgedanke der Prozedur:

1. Bringe die Eingabe in Chomsky-Normalform.
2. Nummeriere die Variablen A_1, \dots, A_n durch.
3. Für $i := 1$ bis n :
 - 3.1 Für jedes $j := 1$ bis $i - 1$, ersetze A_j in $A_i \rightarrow A_j u$ durch alle möglichen rechten Seiten w von $A_j \rightarrow w$ („Inlining von Produktionen“).
 - 3.2 Ersetze jede Produktion $A_i \rightarrow A_j u$ („Entfernen der Linksrekursion“) mithilfe von B_j .

Herstellen der Greibach-Normalform

Grundgedanke der Prozedur:

1. Bringe die Eingabe in Chomsky-Normalform.
2. Nummeriere die Variablen A_1, \dots, A_n durch.
3. Für $i := 1$ bis n :
 - 3.1 Für jedes $j := 1$ bis $i - 1$, ersetze A_j in $A_i \rightarrow A_j u$ durch alle möglichen rechten Seiten w von $A_j \rightarrow w$ („Inlining von Produktionen“).
 - 3.2 Ersetze jede Produktion $A_i \rightarrow A_j u$ („Entfernen der Linksrekursion“) mithilfe von B_j .
(Nun gilt für $A_i \rightarrow A_j u$ stets $i < j$. Damit gilt für $A_n \rightarrow u$, dass u mit einem Terminal beginnt. Die nächste Schleife ersetzt alle restlichen $A_i \rightarrow A_j u$.)

Herstellen der Greibach-Normalform

Grundgedanke der Prozedur:

1. Bringe die Eingabe in Chomsky-Normalform.
2. Nummeriere die Variabeln A_1, \dots, A_n durch.
3. Für $i := 1$ bis n :
 - 3.1 Für jedes $j := 1$ bis $i - 1$, ersetze A_j in $A_i \rightarrow A_j u$ durch alle möglichen rechten Seiten w von $A_j \rightarrow w$ („Inlining von Produktionen“).
 - 3.2 Ersetze jede Produktion $A_i \rightarrow A_j u$ („Entfernen der Linksrekursion“) mithilfe von B_j .
(Nun gilt für $A_i \rightarrow A_j u$ stets $i < j$. Damit gilt für $A_n \rightarrow u$, dass u mit einem Terminal beginnt. Die nächste Schleife ersetzt alle restlichen $A_i \rightarrow A_j u$.)
4. Für $i = n - 1$ bis 1:
 - 3.1 Ersetze A_j in $A_i \rightarrow A_j u$ (wo $j > i$) durch alle möglichen rechten Seiten w von $A_j \rightarrow w$ („Inlining von Produktionen“).
(w fängt mit einem Terminal an, da die Schleife absteigend läuft.)

Herstellen der Greibach-Normalform

Grundgedanke der Prozedur:

1. Bringe die Eingabe in Chomsky-Normalform.
2. Nummeriere die Variablen A_1, \dots, A_n durch.
3. Für $i := 1$ bis n :
 - 3.1 Für jedes $j := 1$ bis $i - 1$, ersetze A_j in $A_i \rightarrow A_j u$ durch alle möglichen rechten Seiten w von $A_j \rightarrow w$ („Inlining von Produktionen“).
 - 3.2 Ersetze jede Produktion $A_i \rightarrow A_j u$ („Entfernen der Linksrekursion“) mithilfe von B_i .
(Nun gilt für $A_i \rightarrow A_j u$ stets $i < j$. Damit gilt für $A_n \rightarrow u$, dass u mit einem Terminal beginnt. Die nächste Schleife ersetzt alle restlichen $A_i \rightarrow A_j u$.)
4. Für $i = n - 1$ bis 1:
 - 3.1 Ersetze A_j in $A_i \rightarrow A_j u$ (wo $j > i$) durch alle möglichen rechten Seiten w von $A_j \rightarrow w$ („Inlining von Produktionen“).
(w fängt mit einem Terminal an, da die Schleife absteigend läuft.)
5. Für jede Produktion $B_i \rightarrow A_j u$, ersetze A_j durch alle möglichen rechten Seiten w von $A_j \rightarrow w$ („Inlining von Produktionen“).

Beispiel für die Herstellung der Greibach-Normalform

Sei $G_0 = (\{S, T\}, \{a\}, \{S \rightarrow TT, T \rightarrow TS \mid a\}, S)$ eine CFG.

1. G_0 befindet sich bereits in Chomsky-Normalform.

Beispiel für die Herstellung der Greibach-Normalform

Sei $G_0 = (\{S, T\}, \{a\}, \{S \rightarrow TT, T \rightarrow TS \mid a\}, S)$ eine CFG.

1. G_0 befindet sich bereits in Chomsky-Normalform.
2. Das Durchnummerieren der Variablen führt z.B. zu
 $G_2 = (\{A_1, A_2\}, \{a\}, \{A_1 \rightarrow A_2A_2, A_2 \rightarrow A_2A_1 \mid a\}, A_1).$

Beispiel für die Herstellung der Greibach-Normalform

Sei $G_0 = (\{S, T\}, \{a\}, \{S \rightarrow TT, T \rightarrow TS \mid a\}, S)$ eine CFG.

1. G_0 befindet sich bereits in Chomsky-Normalform.
2. Das Durchnummerieren der Variablen führt z.B. zu
 $G_2 = (\{A_1, A_2\}, \{a\}, \{A_1 \rightarrow A_2A_2, A_2 \rightarrow A_2A_1 \mid a\}, A_1)$.
3. Ersetze linksrekursive $A_2 \rightarrow A_2A_1 \mid a$ durch $A_2 \rightarrow aB_2 \mid a, B_2 \rightarrow A_1B_2 \mid A_1$.
Dies führt zu
 $G_3 = (\{A_1, A_2\}, \{a\}, \{A_1 \rightarrow A_2A_2, A_2 \rightarrow aB_2 \mid a, B_2 \rightarrow A_1B_2 \mid A_1\}, A_1)$.

Beispiel für die Herstellung der Greibach-Normalform

Sei $G_0 = (\{S, T\}, \{a\}, \{S \rightarrow TT, T \rightarrow TS \mid a\}, S)$ eine CFG.

1. G_0 befindet sich bereits in Chomsky-Normalform.
2. Das Durchnummerieren der Variablen führt z.B. zu
 $G_2 = (\{A_1, A_2\}, \{a\}, \{A_1 \rightarrow A_2A_2, A_2 \rightarrow A_2A_1 \mid a\}, A_1)$.
3. Ersetze linksrekursive $A_2 \rightarrow A_2A_1 \mid a$ durch $A_2 \rightarrow aB_2 \mid a, B_2 \rightarrow A_1B_2 \mid A_1$.
Dies führt zu
 $G_3 = (\{A_1, A_2\}, \{a\}, \{A_1 \rightarrow A_2A_2, A_2 \rightarrow aB_2 \mid a, B_2 \rightarrow A_1B_2 \mid A_1\}, A_1)$.
4. Ersetze mithilfe Inlining $A_1 \rightarrow A_2A_2$ durch $A_1 \rightarrow aB_2A_2 \mid aA_2$. Dies führt zu
 $G_4 = (\{A_1, A_2\}, \{a\}, \{A_1 \rightarrow aB_2A_2 \mid aA_2, A_2 \rightarrow aB_2 \mid a, B_2 \rightarrow A_1B_2 \mid A_1\}, A_1)$.

Beispiel für die Herstellung der Greibach-Normalform

Sei $G_0 = (\{S, T\}, \{a\}, \{S \rightarrow TT, T \rightarrow TS \mid a\}, S)$ eine CFG.

1. G_0 befindet sich bereits in Chomsky-Normalform.
2. Das Durchnummerieren der Variablen führt z.B. zu
 $G_2 = (\{A_1, A_2\}, \{a\}, \{A_1 \rightarrow A_2A_2, A_2 \rightarrow A_2A_1 \mid a\}, A_1)$.
3. Ersetze linksrekursive $A_2 \rightarrow A_2A_1 \mid a$ durch $A_2 \rightarrow aB_2 \mid a, B_2 \rightarrow A_1B_2 \mid A_1$.
Dies führt zu
 $G_3 = (\{A_1, A_2\}, \{a\}, \{A_1 \rightarrow A_2A_2, A_2 \rightarrow aB_2 \mid a, B_2 \rightarrow A_1B_2 \mid A_1\}, A_1)$.
4. Ersetze mithilfe Inlining $A_1 \rightarrow A_2A_2$ durch $A_1 \rightarrow aB_2A_2 \mid aA_2$. Dies führt zu
 $G_4 = (\{A_1, A_2\}, \{a\}, \{A_1 \rightarrow aB_2A_2 \mid aA_2, A_2 \rightarrow aB_2 \mid a, B_2 \rightarrow A_1B_2 \mid A_1\}, A_1)$.
5. Ersetze mithilfe Inlining $B_2 \rightarrow A_1B_2$ durch $B_2 \rightarrow aB_2A_2B_2 \mid aA_2B_2$ und
 $B_2 \rightarrow A_1$ durch $B_2 \rightarrow aB_2A_2 \mid aA_2$. Dies führt zu
 $G_5 = (\{A_1, A_2\}, \{a\}, \{A_1 \rightarrow aB_2A_2 \mid aA_2, A_2 \rightarrow aB_2 \mid a, B_2 \rightarrow aB_2A_2B_2 \mid aA_2B_2 \mid aB_2A_2 \mid aA_2\}, A_1)$.

Links- und rechtsrekursive Produktion

Definition

Seien $G = (V, \Sigma, P, S)$, $A \in V$, $u \in (V \cup \Sigma)^*$. Eine Produktion ist

- ▶ linksrekursiv, wenn sie von der Form $A \rightarrow Au$ ist
- ▶ rechtsrekursiv, wenn sie von der Form $A \rightarrow uA$ ist.

Lemma (Entfernen der Linksrekursion)

Sei $G = (V, \Sigma, Q \cup \underbrace{\{A \rightarrow Au_1 \mid \dots \mid Au_n \mid w_1 \mid \dots \mid w_m\}}_R, S)$ eine CFG mit

- ▶ R sind alle Produktionen mit A als linker Seite
- ▶ die Satzformen w_1, \dots, w_m beginnen alle nicht mit A .

Entfernen der Linksrekursion

Lemma (Entfernen der Linksrekursion)

Sei $G = (V, \Sigma, Q \cup \underbrace{\{A \rightarrow Au_1 \mid \dots \mid Au_n \mid w_1 \mid \dots \mid w_m\}}_R, S)$ eine CFG mit

- ▶ R sind alle Produktionen mit A als linker Seite
- ▶ die Satzformen w_1, \dots, w_m beginnen alle nicht mit A .

Es gilt $L(G') = L(G)$ für $G' = (V \cup \{B\}, \Sigma, Q \cup R', S)$, wobei B eine neue Variable ist und

$$R' := \{A \rightarrow w_1 B \mid \dots \mid w_m B \mid w_1 \mid \dots \mid w_m, \\ B \rightarrow u_1 B \mid \dots \mid u_n B \mid u_1 \mid \dots \mid u_n\}$$

Entfernen der Linksrekursion

Lemma (Entfernen der Linksrekursion)

Sei $G = (V, \Sigma, Q \cup \underbrace{\{A \rightarrow Au_1 \mid \dots \mid Au_n \mid w_1 \mid \dots \mid w_m\}}_R, S)$ eine CFG mit

- ▶ R sind alle Produktionen mit A als linker Seite
- ▶ die Satzformen w_1, \dots, w_m beginnen alle nicht mit A .

Es gilt $L(G') = L(G)$ für $G' = (V \cup \{B\}, \Sigma, Q \cup R', S)$, wobei B eine neue Variable ist und

$$R' := \{A \rightarrow w_1 B \mid \dots \mid w_m B \mid w_1 \mid \dots \mid w_m, \\ B \rightarrow u_1 B \mid \dots \mid u_n B \mid u_1 \mid \dots \mid u_n\}$$

Beweis Siehe Skript.



Beispiel für das Entfernen der Linksrekursion

Sei die CFG $G = (\{A, C\}, \{b, c, d\}, \{A \rightarrow ACA \mid bb, C \rightarrow Ccc \mid d\}, A)$.

Beispiel für das Entfernen der Linksrekursion

Sei die CFG $G = (\{A, C\}, \{b, c, d\}, \{A \rightarrow ACA \mid bb, C \rightarrow Ccc \mid d\}, A)$.

Entfernen der Linksrekursion für A ergibt

$$G' = (\{A, B, C\}, \{b, c, d\}, \{A \rightarrow bbB \mid bb, B \rightarrow CAB \mid CA, C \rightarrow Ccc \mid d\}, A)$$

Beispiel für das Entfernen der Linksrekursion

Sei die CFG $G = (\{A, C\}, \{b, c, d\}, \{A \rightarrow ACA \mid bb, C \rightarrow Ccc \mid d\}, A)$.

Entfernen der Linksrekursion für A ergibt

$$G' = (\{A, B, C\}, \{b, c, d\}, \{A \rightarrow bbB \mid bb, B \rightarrow CAB \mid CA, C \rightarrow Ccc \mid d\}, A)$$

Anschließendes Entfernen der Linksrekursion für C ergibt

$$G'' = (\{A, B, C, D\}, \{b, c, d\}, \\ \{A \rightarrow bbB \mid bb, B \rightarrow CAB \mid CA, C \rightarrow dD \mid d, D \rightarrow ccD \mid cc\}, A)$$

Algorithmus 7: Herstellen der Greibach-Normalform

Eingabe: CFG $G = (\{A_1, \dots, A_n\}, \Sigma, P, A_i)$ in Chomsky-Normalform mit $\varepsilon \notin L(G)$

Ausgabe: CFG G' in Greibach-Normalform mit $L(G) = L(G')$

Beginn

für $i = 1$ *bis* n **tue**

für $j = 1$ *bis* $i - 1$ **tue**

für alle $A_i \rightarrow A_j u \in P$ **tue**

 Seien $A_j \rightarrow w_1 \mid \dots \mid w_m$ alle Produktionen in P mit A_j als linker Seite;

 Ersetze $A_i \rightarrow A_j u$ durch $A_i \rightarrow w_1 u \mid \dots \mid w_m u$ in P ;

wenn $A_i \rightarrow A_i u \in P$ **dann**

 Eliminiere solche Produktionen mit der Operation „Entfernen der Linksrekursion“;

 Sei B_i die dabei neu erzeugte Variable;

für $i = n - 1$ *bis* 1 **tue**

für alle $A_i \rightarrow A_j u \in P, j > i$ **tue**

 Seien $A_j \rightarrow w_1 \mid \dots \mid w_m$ alle Produktionen in P mit A_j als linker Seite;

 Ersetze $A_i \rightarrow A_j u$ durch $A_i \rightarrow w_1 u \mid \dots \mid w_m u$ in P ;

für $i = 1$ *bis* n **tue**

für alle $B_i \rightarrow A_j u \in P$ **tue**

 Seien $A_j \rightarrow w_1 \mid \dots \mid w_m$ alle Produktionen in P mit A_j als linker Seite;

 Ersetze $B_i \rightarrow A_j u$ durch $B_i \rightarrow w_1 u \mid \dots \mid w_m u$ in P ;

Satz

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

Satz

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

Beweis Korrektheit folgt durch Korrektheit der Operationen „Inlining von Produktionen“ und „Entfernen der Linksrekursion“, mithilfe von geeigneten Invarianten. □

Abschlusseigenschaften der kontextfreien Sprachen

Theorem

Die kontextfreien Sprachen sind abgeschlossen bezüglich **Vereinigung**, **Produkt** und **Kleeneschem Abschluss**.

Abschlusseigenschaften der kontextfreien Sprachen

Theorem

Die kontextfreien Sprachen sind abgeschlossen bezüglich Vereinigung, Produkt und Kleeneschem Abschluss.

Beweis Seien L_1, L_2 CFLs und $G_i = (V_i, \Sigma_i, P_i, S_i)$ CFGs mit $L(G_i) = L_i$ für $i = 1, 2$.
O.B.d.A. sei $V_1 \cap V_2 = \emptyset$.

Abschlusseigenschaften der kontextfreien Sprachen

Theorem

Die kontextfreien Sprachen sind abgeschlossen bezüglich Vereinigung, Produkt und Kleeneschem Abschluss.

Beweis Seien L_1, L_2 CFLs und $G_i = (V_i, \Sigma_i, P_i, S_i)$ CFGs mit $L(G_i) = L_i$ für $i = 1, 2$.
O.B.d.A. sei $V_1 \cap V_2 = \emptyset$.

Seien S, S' neue Variablen (d.h. $\{S, S'\} \cap (V_1 \cup V_2) = \emptyset$).

Abschlusseigenschaften der kontextfreien Sprachen

Theorem

Die kontextfreien Sprachen sind abgeschlossen bezüglich Vereinigung, Produkt und Kleeneschem Abschluss.

Beweis Seien L_1, L_2 CFLs und $G_i = (V_i, \Sigma_i, P_i, S_i)$ CFGs mit $L(G_i) = L_i$ für $i = 1, 2$.
O.B.d.A. sei $V_1 \cap V_2 = \emptyset$.

Seien S, S' neue Variablen (d.h. $\{S, S'\} \cap (V_1 \cup V_2) = \emptyset$).

1. **Vereinigung:** Sei $G_U = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \rightarrow S_1 \mid S_2\}, S)$.
Dann gilt $L(G_U) = L(G_1) \cup L(G_2) = L_1 \cup L_2$.

Abschlusseigenschaften der kontextfreien Sprachen

Theorem

Die kontextfreien Sprachen sind abgeschlossen bezüglich **Vereinigung**, **Produkt** und **Kleeneschem Abschluss**.

Beweis Seien L_1, L_2 CFLs und $G_i = (V_i, \Sigma_i, P_i, S_i)$ CFGs mit $L(G_i) = L_i$ für $i = 1, 2$.
O.B.d.A. sei $V_1 \cap V_2 = \emptyset$.

Seien S, S' neue Variablen (d.h. $\{S, S'\} \cap (V_1 \cup V_2) = \emptyset$).

- Vereinigung:** Sei $G_U = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \rightarrow S_1 \mid S_2\}, S)$.
Dann gilt $L(G_U) = L(G_1) \cup L(G_2) = L_1 \cup L_2$.
- Produkt:** Sei $G_\cdot = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \rightarrow S_1 S_2\}, S)$.
Dann gilt $L(G_\cdot) = L(G_1)L(G_2) = L_1 L_2$.

Abschlusseigenschaften der kontextfreien Sprachen

Theorem

Die kontextfreien Sprachen sind abgeschlossen bezüglich **Vereinigung**, **Produkt** und **Kleeneschem Abschluss**.

Beweis Seien L_1, L_2 CFLs und $G_i = (V_i, \Sigma_i, P_i, S_i)$ CFGs mit $L(G_i) = L_i$ für $i = 1, 2$.
O.B.d.A. sei $V_1 \cap V_2 = \emptyset$.

Seien S, S' neue Variablen (d.h. $\{S, S'\} \cap (V_1 \cup V_2) = \emptyset$).

- Vereinigung:** Sei $G_U = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \rightarrow S_1 \mid S_2\}, S)$.
Dann gilt $L(G_U) = L(G_1) \cup L(G_2) = L_1 \cup L_2$.
- Produkt:** Sei $G_\cdot = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \rightarrow S_1 S_2\}, S)$.
Dann gilt $L(G_\cdot) = L(G_1)L(G_2) = L_1 L_2$.
- Kleenescher Abschluss:** Sei $G_* = (V_1 \cup \{S, S'\}, \Sigma, P', S')$ mit
 $P' = (P \setminus \{S_1 \rightarrow \varepsilon\}) \cup \{S' \rightarrow \varepsilon, S' \rightarrow S, S \rightarrow SS, S \rightarrow S_1\}$.
Dann gilt $L(G_*) = L(G_1)^*$. □

Abschlusseigenschaften der kontextfreien Sprachen

Theorem

Die kontextfreien Sprachen sind **nicht** abgeschlossen unter **Schnitt** und **Komplement**.

Abschlusseigenschaften der kontextfreien Sprachen

Theorem

Die kontextfreien Sprachen sind **nicht** abgeschlossen unter **Schnitt** und **Komplement**.

Beweis

1. **Schnitt:** Seien $L_1 = \{a^n b^m c^m \mid m, n \in \mathbb{N}\}$ und $L_2 = \{a^m b^m c^n \mid m, n \in \mathbb{N}\}$.

Abschlusseigenschaften der kontextfreien Sprachen

Theorem

Die kontextfreien Sprachen sind **nicht** abgeschlossen unter **Schnitt** und **Komplement**.

Beweis

1. **Schnitt:** Seien $L_1 = \{a^n b^m c^m \mid m, n \in \mathbb{N}\}$ und $L_2 = \{a^m b^m c^n \mid m, n \in \mathbb{N}\}$.

Seien

$$G_1 = (\{A, D, S\}, \{a, b, c\}, \{S \rightarrow AD, A \rightarrow \varepsilon \mid aA, D \rightarrow \varepsilon \mid bDc\}, S)$$

$$G_2 = (\{C, D, S\}, \{a, b, c\}, \{S \rightarrow DC, \varepsilon \mid C \rightarrow cC, D \rightarrow \varepsilon \mid aDb\}, S)$$

Abschlusseigenschaften der kontextfreien Sprachen

Theorem

Die kontextfreien Sprachen sind **nicht** abgeschlossen unter **Schnitt** und **Komplement**.

Beweis

1. **Schnitt:** Seien $L_1 = \{a^n b^m c^m \mid m, n \in \mathbb{N}\}$ und $L_2 = \{a^m b^m c^n \mid m, n \in \mathbb{N}\}$.

Seien

$$G_1 = (\{A, D, S\}, \{a, b, c\}, \{S \rightarrow AD, A \rightarrow \varepsilon \mid aA, D \rightarrow \varepsilon \mid bDc\}, S)$$

$$G_2 = (\{C, D, S\}, \{a, b, c\}, \{S \rightarrow DC, \varepsilon \mid C \rightarrow cC, D \rightarrow \varepsilon \mid aDb\}, S)$$

$L(G_i) = L_i$ für $i \in \{1, 2\}$, daher sind L_1 und L_2 beide kontextfrei.

Abschlusseigenschaften der kontextfreien Sprachen

Theorem

Die kontextfreien Sprachen sind **nicht** abgeschlossen unter **Schnitt** und **Komplement**.

Beweis

1. **Schnitt:** Seien $L_1 = \{a^n b^m c^m \mid m, n \in \mathbb{N}\}$ und $L_2 = \{a^m b^m c^n \mid m, n \in \mathbb{N}\}$.

Seien

$$G_1 = (\{A, D, S\}, \{a, b, c\}, \{S \rightarrow AD, A \rightarrow \varepsilon \mid aA, D \rightarrow \varepsilon \mid bDc\}, S)$$

$$G_2 = (\{C, D, S\}, \{a, b, c\}, \{S \rightarrow DC, \varepsilon \mid C \rightarrow cC, D \rightarrow \varepsilon \mid aDb\}, S)$$

$L(G_i) = L_i$ für $i \in \{1, 2\}$, daher sind L_1 und L_2 beide kontextfrei.

$L_1 \cap L_2 = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ ist aber bekanntlich nicht kontextfrei.

Abschlusseigenschaften der kontextfreien Sprachen

Theorem

Die kontextfreien Sprachen sind **nicht** abgeschlossen unter **Schnitt** und **Komplement**.

Beweis

1. **Schnitt:** Seien $L_1 = \{a^n b^m c^m \mid m, n \in \mathbb{N}\}$ und $L_2 = \{a^m b^m c^n \mid m, n \in \mathbb{N}\}$.

Seien

$$G_1 = (\{A, D, S\}, \{a, b, c\}, \{S \rightarrow AD, A \rightarrow \varepsilon \mid aA, D \rightarrow \varepsilon \mid bDc\}, S)$$

$$G_2 = (\{C, D, S\}, \{a, b, c\}, \{S \rightarrow DC, \varepsilon \mid C \rightarrow cC, D \rightarrow \varepsilon \mid aDb\}, S)$$

$L(G_i) = L_i$ für $i \in \{1, 2\}$, daher sind L_1 und L_2 beide kontextfrei.

$L_1 \cap L_2 = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ ist aber bekanntlich nicht kontextfrei.

Daher sind die CFLs nicht abgeschlossen bezüglich Schnitt.

Beweis (Fortsetzung)

2. **Komplement:** Durch Widerspruch. Wir nehmen an, dass wenn L CFL ist, dann ist auch \overline{L} CFL.

Beweis (Fortsetzung)

2. **Komplement:** Durch Widerspruch. Wir nehmen an, dass wenn L CFL ist, dann ist auch \overline{L} CFL.

Seien L_1, L_2 CFLs. Dann ist auch $\overline{\overline{L_1} \cup \overline{L_2}}$ CFL
(da CFLs abgeschlossen bezüglich $\overline{}$ und \cup).

Beweis (Fortsetzung)

2. **Komplement:** Durch Widerspruch. Wir nehmen an, dass wenn L CFL ist, dann ist auch \overline{L} CFL.

Seien L_1, L_2 CFLs. Dann ist auch $\overline{\overline{L_1} \cup \overline{L_2}}$ CFL
(da CFLs abgeschlossen bezüglich $\overline{}$ und \cup).

$\overline{\overline{L_1} \cup \overline{L_2}} = L_1 \cap L_2$. Dies würde heißen, dass die CFLs abgeschlossen bezüglich Schnitt sind. Wir haben aber unter Punkt 1 genau das Gegenteil beweisen.
Widerspruch. □