6a

Die Greibach-Normalform und Eigenschaften von kontextfreien Sprachen

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik und Theorembeweisen

Stand: 10. Mai 2024
Basiert auf Folien von PD Dr. David Sabel

Leerheitsproblem

Satz

Das Leerheitsproblem für kontextfreie Grammatiken ist entscheidbar.

Leerheitsproblem

Satz

Das Leerheitsproblem für kontextfreie Grammatiken ist entscheidbar.

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

Leerheitsproblem

Satz

Das Leerheitsproblem für kontextfreie Grammatiken ist entscheidbar.

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

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

Leerheitsproblem

Satz

Das Leerheitsproblem für kontextfreie Grammatiken ist entscheidbar.

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

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

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

Leerheitsproblem

Satz

Das Leerheitsproblem für kontextfreie Grammatiken ist entscheidbar.

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

Prüfe zunächst, ob $S \to \varepsilon \in L(G)$. 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 $\{w \in \Sigma^* \mid A \Rightarrow^*_{G'} w\} \neq \emptyset$.

Leerheitsproblem

Satz

Das Leerheitsproblem für kontextfreie Grammatiken ist entscheidbar.

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

Prüfe zunächst, ob $S \to \varepsilon \in L(G)$. 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

$W := \{A \in V \mid A \to a \in P, a \in \Sigma\}$;
wiederhole
  $W_{alt} := W$;
  $W := W_{alt} \cup \{A \mid A \to BC \in P, B \in W_{alt}, C \in W_{alt}\}$;
bis $W = W_{alt}$;
return W

Beispiel für die Markierung der „nichtleeren“ Variablen

Sei $G = (\{S, A, B, C\}, \{a, b\}, \{S \to AB, C \to CC \mid a \mid b, A \to CC \mid BB, B \to CB\})$.

Beispiel für die Markierung der „nichtleeren“ Variablen

Sei $G = (\{S, A, B, C\}, \{a, b\}, \{S \to AB, C \to CC \mid a \mid b, A \to CC \mid BB, B \to CB\})$.

Ausführung von Algorithmus 9:

0. $W := \{A \in V \mid A \to a \in P\} = \{C\}$

Beispiel für die Markierung der „nichtleeren“ Variablen

Sei $G = (\{S, A, B, C\}, \{a, b\}, \{S \to AB,\ C \to CC \mid a \mid b,\ A \to CC \mid BB,\ B \to CB\})$.

Ausführung von Algorithmus 9:

0. $W := \{A \in V \mid A \to a \in P\} = \{C\}$
1. $W_{alt} := W = \{C\}$
   $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{C\} \cup \{A, C\} = \{A, C\}$
   $W \neq W_{alt}$

Beispiel für die Markierung der „nichtleeren“ Variablen

Sei $G = (\{S, A, B, C\}, \{a, b\}, \{S \to AB, C \to CC \mid a \mid b, A \to CC \mid BB, B \to CB\})$.

Ausführung von Algorithmus 9:

0. $W := \{A \in V \mid A \to a \in P\} = \{C\}$
1. $W_{alt} := W = \{C\}$
   $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{C\} \cup \{A, C\} = \{A, C\}$
   $W \neq W_{alt}$
2. $W_{alt} := W = \{A, C\}$
   $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{A, C\} \cup \{A, C\} = \{A, C\}$
   $W = W_{alt}$

Beispiel für die Markierung der „nichtleeren“ Variablen

Sei $G = (\{S, A, B, C\}, \{a, b\}, \{S \to AB, C \to CC \mid a \mid b, A \to CC \mid BB, B \to CB\})$.

Ausführung von Algorithmus 9:

0. $W := \{A \in V \mid A \to a \in P\} = \{C\}$
1. $W_{alt} := W = \{C\}$
   $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{C\} \cup \{A, C\} = \{A, C\}$
   $W \neq W_{alt}$
2. $W_{alt} := W = \{A, C\}$
   $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{A, C\} \cup \{A, C\} = \{A, C\}$
   $W = W_{alt}$
3. return $W = \{A, C\}$

Beispiel für die Markierung der „nichtleeren“ Variablen

Sei $G = (\{S, A, B, C\}, \{a, b\}, \{S \to AB,\ C \to CC \mid a \mid b,\ A \to CC \mid BB,\ B \to CB\})$.

Ausführung von Algorithmus 9:

0. $W := \{A \in V \mid A \to a \in P\} = \{C\}$
1. $W_{alt} := W = \{C\}$
   $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{C\} \cup \{A, C\} = \{A, C\}$
   $W \neq W_{alt}$
2. $W_{alt} := W = \{A, C\}$
   $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{A, C\} \cup \{A, C\} = \{A, C\}$
   $W = W_{alt}$
3. return $W = \{A, C\}$

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

Die Greibach-Normalform

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 Greibach-Normalform

Definition

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

Die Normalform ist benannt nach Sheila A. Greibach.

Die Greibach-Normalform

Definition

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

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

Definition

Eine CFG $G = (V, \Sigma, P, S)$ ist in Greibach-Normalform, falls alle Produktionen in P von der Form $A \to aB_1B_2 \dots B_j$ 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 (später).

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 Variabeln $A_1, \ldots, A_n$ durch.

Herstellen der Greibach-Normalform

Grundgedanke der Prozedur:

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

Herstellen der Greibach-Normalform

Grundgedanke der Prozedur:

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

   (Nun gilt für $A_i \to A_j u$ stets $i < j$. Damit gilt für $A_n \to u$, dass u mit einem Terminal beginnt. Die nächste Schleife ersetzt alle restlichen $A_i \to A_j u$.)

Herstellen der Greibach-Normalform

Grundgedanke der Prozedur:

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

   (Nun gilt für $A_i \to A_j u$ stets $i < j$. Damit gilt für $A_n \to u$, dass u mit einem Terminal beginnt. Die nächste Schleife ersetzt alle restlichen $A_i \to A_j u$.)
4. Für $i = n - 1$ bis 1:
   3.1 Ersetze A_j in $A_i \to A_j u$ (wo $j > i$) durch alle möglichen rechten Seiten w von $A_j \to w$ („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 Variabeln $A_1, \ldots, A_n$ durch.
3. Für $i := 1$ bis n:
   3.1 Für jedes $j := 1$ bis $i - 1$, ersetze A_j in $A_i \to A_j u$ durch alle möglichen rechten Seiten w von $A_j \to w$ („Inlining von Produktionen“).
   3.2 Ersetze jede Produktion $A_i \to A_i u$ („Entfernen der Linksrekursion“) mithilfe von B_i.

   (Nun gilt für $A_i \to A_j u$ stets $i < j$. Damit gilt für $A_n \to u$, dass u mit einem Terminal beginnt. Die nächste Schleife ersetzt alle restlichen $A_i \to A_j u$.)
4. Für $i = n - 1$ bis 1:
   3.1 Ersetze A_j in $A_i \to A_j u$ (wo $j > i$) durch alle möglichen rechten Seiten w von $A_j \to w$ („Inlining von Produktionen“).
   (w fängt mit einem Terminal an, da die Schleife absteigend läuft.)
5. Für jede Produktion $B_i \to A_j u$, ersetze A_j durch alle möglichen rechten Seiten w von $A_j \to w$ („Inlining von Produktionen“).

Beispiel für die Herstellung der Greibach-Normalform

0. Eingabe: CFG $G_0 = (\{S, T\}, \{a\}, \{S \to TT, T \to TS \mid a\}, S)$.

Beispiel für die Herstellung der Greibach-Normalform

0. Eingabe: CFG $G_0 = (\{S, T\}, \{a\}, \{S \to TT, T \to TS \mid a\}, S)$.
1. G_0 befindet sich bereits in Chomsky-Normalform.

Beispiel für die Herstellung der Greibach-Normalform

0. Eingabe: CFG $G_0 = (\{S, T\}, \{a\}, \{S \to TT, T \to TS \mid a\}, S)$.
1. G_0 befindet sich bereits in Chomsky-Normalform.
2. Das Durchnummerieren der Variablen führt z.B. zu $G_2 = (\{A_1, A_2\}, \{a\}, \{A_1 \to A_2A_2, A_2 \to A_2A_1 \mid a\}, A_1)$.

Beispiel für die Herstellung der Greibach-Normalform

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

Beispiel für die Herstellung der Greibach-Normalform

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

Beispiel für die Herstellung der Greibach-Normalform

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

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 $A \to Au$ ist
- rechtsrekursiv, wenn sie von der Form $A \to uA$ ist.

Entfernen der Linksrekursion

Lemma (Entfernen der Linksrekursion)

Sei $G = (V, \Sigma, Q \cup \underbrace{\{A \to Au_1 \mid \cdots \mid Au_n \mid w_1 \mid \cdots \mid w_m\}}_{R}, S)$ eine CFG mit

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

Entfernen der Linksrekursion

Lemma (Entfernen der Linksrekursion)

Sei $G = (V, \Sigma, Q \cup \underbrace{\{A \to Au_1 \mid \cdots \mid Au_n \mid w_1 \mid \cdots \mid w_m\}}_{R}, S)$ eine CFG mit

- R sind alle Produktionen mit A als linker Seite
- die Satzformen $w_1, \ldots, w_m$ 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

$$\begin{aligned} R' := \{&A \to w_1 B \mid \cdots \mid w_m B \mid w_1 \mid \cdots \mid w_m, \\ &B \to u_1 B \mid \cdots \mid u_n B \mid u_1 \mid \cdots \mid u_n\} \end{aligned}$$

Entfernen der Linksrekursion

Lemma (Entfernen der Linksrekursion)

Sei $G = (V, \Sigma, Q \cup \underbrace{\{A \to Au_1 \mid \cdots \mid Au_n \mid w_1 \mid \cdots \mid w_m\}}_{R}, S)$ eine CFG mit

- R sind alle Produktionen mit A als linker Seite
- die Satzformen $w_1, \ldots, w_m$ 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

$$\begin{aligned} R' := \{A &\to w_1 B \mid \cdots \mid w_m B \mid w_1 \mid \cdots \mid w_m, \\ B &\to u_1 B \mid \cdots \mid u_n B \mid u_1 \mid \cdots \mid u_n\} \end{aligned}$$

Beweis Siehe Skript. □

Beispiel für das Entfernen der Linksrekursion

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

Beispiel für das Entfernen der Linksrekursion

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

Entfernen der Linksrekursion für A ergibt

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

Beispiel für das Entfernen der Linksrekursion

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

Entfernen der Linksrekursion für A ergibt

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

Anschließendes Entfernen der Linksrekursion für C ergibt

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

Algorithmus 7: Herstellen der Greibach-Normalform

Eingabe: CFG $G = (\{A_1, \ldots, A_n\}, \Sigma, P, A_i)$ 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** $A_i \to A_j u \in P$ **tue**
      - Seien $A_j \to w_1 \mid \cdots \mid w_m$ alle Produktionen in P mit A_j als linker Seite;
      - Ersetze $A_i \to A_j u$ durch $A_i \to w_1 u \mid \ldots \mid w_m u$ in P;
  - **wenn** $A_i \to A_i u \in P$ **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** $A_i \to A_j u \in P$, $j > i$ **tue**
    - Seien $A_j \to w_1 \mid \cdots \mid w_m$ alle Produktionen in P mit A_j als linker Seite;
    - Ersetze $A_i \to A_j u$ durch $A_i \to w_1 u \mid \cdots \mid w_m u$ in P;
- **für** $i = 1$ *bis* n **tue**
  - **für alle** $B_i \to A_j u \in P$ **tue**
    - Seien $A_j \to w_1 \mid \cdots \mid w_m$ alle Produktionen in P mit A_j als linker Seite;
    - Ersetze $B_i \to A_j u$ durch $B_i \to w_1 u \mid \cdots \mid w_m u$ in P;

Korrektheit von Algorithmus 7

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.

Korrektheit von Algorithmus 7

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. $\square$

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 $G_\cup = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \to S_1 \mid S_2\}, S)$.
   Dann gilt $L(G_\cup) = L(G_1) \cup L(G_2) = L_1 \cup L_2$.

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. $\{S, S'\} \cap (V_1 \cup V_2) = \emptyset$).

1. Vereinigung: Sei $G_\cup = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \to S_1 \mid S_2\}, S)$. Dann gilt $L(G_\cup) = L(G_1) \cup L(G_2) = L_1 \cup L_2$.
2. Produkt: Sei $G_\circ = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \to S_1 S_2\}, S)$. Dann gilt $L(G_\circ) = L(G_1)L(G_2) = L_1 L_2$.

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. $\{S, S'\} \cap (V_1 \cup V_2) = \emptyset$).

1. Vereinigung: Sei $G_\cup = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \to S_1 \mid S_2\}, S)$.
   Dann gilt $L(G_\cup) = L(G_1) \cup L(G_2) = L_1 \cup L_2$.
2. Produkt: Sei $G_\circ = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \to S_1 S_2\}, S)$.
   Dann gilt $L(G_\circ) = L(G_1)L(G_2) = L_1 L_2$.
3. Kleenescher Abschluss: Sei $G_* = (V_1 \cup \{S', S\}, \Sigma, P', S')$ mit
   $P' = (P \setminus \{S_1 \to \varepsilon\}) \cup \{S' \to \varepsilon, S' \to S, S \to SS, S \to S_1\}$.
   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

$$G_1 = (\{A, D, S\}, \{a, b, c\}, \{S \to AD,\ A \to \varepsilon \mid aA,\ D \to \varepsilon \mid bDc\}, S)$$
$$G_2 = (\{C, D, S\}, \{a, b, c\}, \{S \to DC, \varepsilon \mid C \to cC,\ D \to \varepsilon \mid aDb\}, S)$$

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

$$G_1 = (\{A, D, S\}, \{a, b, c\}, \{S \to AD, A \to \varepsilon \mid aA, D \to \varepsilon \mid bDc\}, S)$$
$$G_2 = (\{C, D, S\}, \{a, b, c\}, \{S \to DC, \varepsilon \mid C \to cC, D \to \varepsilon \mid aDb\}, S)$$

   $L(G_i) = L_i$ 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

$$G_1 = (\{A, D, S\}, \{a, b, c\}, \{S \to AD, A \to \varepsilon \mid aA, D \to \varepsilon \mid bDc\}, S)$$
$$G_2 = (\{C, D, S\}, \{a, b, c\}, \{S \to DC, \varepsilon \mid C \to cC, D \to \varepsilon \mid aDb\}, S)$$

   $L(G_i) = L_i$ 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 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

   $$G_1 = (\{A, D, S\}, \{a, b, c\}, \{S \to AD,\ A \to \varepsilon \mid aA,\ D \to \varepsilon \mid bDc\}, S)$$
   $$G_2 = (\{C, D, S\}, \{a, b, c\}, \{S \to DC, \varepsilon \mid C \to cC,\ D \to \varepsilon \mid aDb\}, S)$$

   $L(G_i) = L_i$ 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 nicht kontextfrei.

   Daher sind die CFLs nicht abgeschlossen bezüglich Schnitt.

Abschlusseigenschaften der kontextfreien Sprachen

Beweis (Fortsetzung)

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

Abschlusseigenschaften der kontextfreien Sprachen

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 $^{-}$ 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 $^-$ 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. $\square$