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

15. April 2005

Sei $G = (V, \Sigma, P, S)$ eine Phrasenstrukturgrammatik.

Definition 4

Wir schreiben

1. $z \rightarrow_G z'$ gdw
$(\exists x, y \in (V \cup \Sigma)^*, l \rightarrow r \in P)[z = xly, z' = xry]$
2. $z \rightarrow_G^* z'$ gdw $z = z'$ oder
$z \rightarrow_G z^{(1)} \rightarrow_G z^{(2)} \rightarrow_G \ldots \rightarrow_G z^{(k)} = z'$. Eine solche Folge von Ableitungsschritten heißt eine Ableitung für z' von z in G (der Länge k).
3. Die von G erzeugte Sprache ist

$$L(G) := \{z \in \Sigma^*;\ s \rightarrow_G^* z'\}$$

Zur Vereinfachung der Notation schreiben wir gewöhnlich $\rightarrow$ und $\rightarrow^*$ statt $\rightarrow_G$ und $\rightarrow_G^*$

Sei $G = (V, \Sigma, P, S)$ eine Phrasenstrukturgrammatik.

Definition 4

Wir schreiben

1. $z \to_G z'$ gdw
$(\exists x, y \in (V \cup \Sigma)^*, l \to r \in P)[z = xly, z' = xry]$
2. $z \to_G^* z'$ gdw $z = z'$ oder
$z \to_G z^{(1)} \to_G z^{(2)} \to_G \ldots \to_G z^{(k)} = z'$. Eine solche Folge von Ableitungsschritten heißt eine Ableitung für z' von z in G (der Länge k).
3. Die von G erzeugte Sprache ist

$$L(G) := \{z \in \Sigma^*;\ s \to_G^* z'\}$$

Zur Vereinfachung der Notation schreiben wir gewöhnlich $\to$ und $\to^*$ statt $\to_G$ und $\to_G^*$

Sei $G = (V, \Sigma, P, S)$ eine Phrasenstrukturgrammatik.

Definition 4

Wir schreiben

1. $z \rightarrow_G z'$ gdw
   $(\exists x, y \in (V \cup \Sigma)^*, l \rightarrow r \in P)[z = xly, z' = xry]$
2. $z \rightarrow_G^* z'$ gdw $z = z'$ oder
   $z \rightarrow_G z^{(1)} \rightarrow_G z^{(2)} \rightarrow_G \ldots \rightarrow_G z^{(k)} = z'$. Eine solche Folge von Ableitungsschritten heißt eine Ableitung für z' von z in G (der Länge k).
3. Die von G erzeugte Sprache ist

$$L(G) := \{z \in \Sigma^*;\ s \rightarrow_G^* z'\}$$

Zur Vereinfachung der Notation schreiben wir gewöhnlich $\rightarrow$ und $\rightarrow^*$ statt $\rightarrow_G$ und $\rightarrow_G^*$

Sei $G = (V, \Sigma, P, S)$ eine Phrasenstrukturgrammatik.

Definition 4

Wir schreiben

1. $z \rightarrow_G z'$ gdw
$(\exists x, y \in (V \cup \Sigma)^*, l \rightarrow r \in P)[z = xly, z' = xry]$
2. $z \rightarrow_G^* z'$ gdw $z = z'$ oder
$z \rightarrow_G z^{(1)} \rightarrow_G z^{(2)} \rightarrow_G \ldots \rightarrow_G z^{(k)} = z'$. Eine solche Folge von Ableitungsschritten heißt eine Ableitung für z' von z in G (der Länge k).
3. Die von G erzeugte Sprache ist

$$L(G) := \{z \in \Sigma^*;\ s \rightarrow_G^* z'\}$$

Zur Vereinfachung der Notation schreiben wir gewöhnlich $\rightarrow$ und $\rightarrow^*$ statt $\rightarrow_G$ und $\rightarrow_G^*$

Sei $G = (V, \Sigma, P, S)$ eine Phrasenstrukturgrammatik.

Definition 4

Wir schreiben

1. $z \rightarrow_G z'$ gdw
$(\exists x, y \in (V \cup \Sigma)^*, l \rightarrow r \in P)[z = xly, z' = xry]$
2. $z \rightarrow_G^* z'$ gdw $z = z'$ oder
$z \rightarrow_G z^{(1)} \rightarrow_G z^{(2)} \rightarrow_G \ldots \rightarrow_G z^{(k)} = z'$. Eine solche Folge von Ableitungsschritten heißt eine Ableitung für z' von z in G (der Länge k).
3. Die von G erzeugte Sprache ist

$$L(G) := \{z \in \Sigma^*;\ s \rightarrow_G^* z'\}$$

Zur Vereinfachung der Notation schreiben wir gewöhnlich $\rightarrow$ und $\rightarrow^*$ statt $\rightarrow_G$ und $\rightarrow_G^*$

Beispiel 5

Wir erinnern uns:

- $L_2 = \{ab, abab, ababab, \ldots\} = \{(ab)^n, n \in \mathbb{N}\}$
  $(\Sigma_2 = \{a, b\})$
- Grammatik für L_2 mit folgenden Produktionen:

$$S \rightarrow ab, S \rightarrow abS$$

- $L_4 = \{a, b, aa, ab, bb, aaa, aab, abb, bbb \ldots\}$
  $= \{a^m b^n, m, n \in \mathbb{N}_0, m + n > 0\}$ $(\Sigma_4 = \{a, b\})$
- Grammatik für L_4 mit folgenden Produktionen:

$$S \rightarrow A, S \rightarrow B, S \rightarrow AB$$
$$A \rightarrow a, A \rightarrow aA$$
$$B \rightarrow b, B \rightarrow bB$$

Beispiel 5

Wir erinnern uns:

- $L_2 = \{ab, abab, ababab, \ldots\} = \{(ab)^n, n \in \mathbb{N}\}$
  $(\Sigma_2 = \{a, b\})$
- Grammatik für L_2 mit folgenden Produktionen:

$$S \to ab, S \to abS$$

- $L_4 = \{a, b, aa, ab, bb, aaa, aab, abb, bbb \ldots\}$
  $= \{a^m b^n, m, n \in \mathbb{N}_0, m + n > 0\}$ $(\Sigma_4 = \{a, b\})$
- Grammatik für L_4 mit folgenden Produktionen:

$$S \to A, S \to B, S \to AB$$
$$A \to a, A \to aA,$$
$$B \to b, B \to bB$$

Beispiel 5

Wir erinnern uns:

- $L_2 = \{ab, abab, ababab, \ldots\} = \{(ab)^n, n \in \mathbb{N}\}$
  $(\Sigma_2 = \{a, b\})$
- Grammatik für L_2 mit folgenden Produktionen:

$$S \to ab, S \to abS$$

  Wir bezeichnen Nichtterminale mit großen und Terminale mit kleinen Buchstaben!

- $L_4 = \{a, b, aa, ab, bb, aaa, aab, abb, bbb \ldots\}$
  $= \{a^m b^n, m, n \in \mathbb{N}_0, m + n > 0\}$ $\qquad (\Sigma_4 = \{a, b\})$
- Grammatik für L_4 mit folgenden Produktionen:

$$S \to A,\ S \to B,\ S \to AB,$$
$$A \to a,\ A \to aA,$$
$$B \to b,\ B \to bB$$

Beispiel 5

Wir erinnern uns:

- $L_4 = \{a, b, aa, ab, bb, aaa, aab, abb, bbb \ldots\}$
  $= \{a^m b^n, m, n \in \mathbb{N}_0, m + n > 0\} \qquad (\Sigma_4 = \{a, b\})$
- Grammatik für L_4 mit folgenden Produktionen:

$$S \rightarrow A,\ S \rightarrow B,\ S \rightarrow AB,$$
$$A \rightarrow a,\ A \rightarrow aA,$$
$$B \rightarrow b,\ B \rightarrow bB$$

Beispiel 5

Wir erinnern uns:

- $L_4 = \{a, b, aa, ab, bb, aaa, aab, abb, bbb \ldots\}$
  $= \{a^m b^n, m, n \in \mathbb{N}_0, m + n > 0\} \qquad (\Sigma_4 = \{a, b\})$
- Grammatik für L_4 mit folgenden Produktionen:

$$S \to A,\ S \to B,\ S \to AB,$$
$$A \to a,\ A \to aA,$$
$$B \to b,\ B \to bB$$

2.2 Die Chomsky-Hierarchie

Sei $G = (V, \Sigma, P, S)$ eine Phrasenstrukturgrammatik.

1. Jede Phrasenstrukturgrammatik (Chomsky-Grammatik) ist (zunächst) automatisch vom Typ 0.
2. Eine Chomsky-Grammatik heißt (längen-)monoton, falls für alle Regeln
$$\alpha \to \beta \in P \text{ mit } \alpha \neq S$$
gilt:
$$|\alpha| \leq |\beta|\,,$$
und, falls $S \to \epsilon \in P$, dann das Axiom S auf keiner rechten Seite vorkommt.

2.2 Die Chomsky-Hierarchie

Sei $G = (V, \Sigma, P, S)$ eine Phrasenstrukturgrammatik.

1. Jede Phrasenstrukturgrammatik (Chomsky-Grammatik) ist (zunächst) automatisch vom Typ 0.
2. Eine Chomsky-Grammatik heißt (längen-)monoton, falls für alle Regeln
$$\alpha \rightarrow \beta \in P \text{ mit } \alpha \neq S$$
gilt:
$$|\alpha| \leq |\beta| \,,$$
und, falls $S \rightarrow \epsilon \in P$, dann das Axiom S auf keiner rechten Seite vorkommt.

2.2 Die Chomsky-Hierarchie

Sei $G = (V, \Sigma, P, S)$ eine Phrasenstrukturgrammatik.

3. Eine Chomsky-Grammatik ist vom Typ 1 (auch: kontextsensitiv), falls sie monoton ist und für alle Regeln $\alpha \to \beta$ in P mit $\alpha \neq S$ gilt:

$$\alpha = \alpha' A \alpha'' \text{ und } \beta = \alpha' \beta' \alpha''$$

für geeignete $A \in V$, $\alpha', \alpha'' \in (V \cup \Sigma)^*$ und $\beta' \in (V \cup \Sigma)^+$.

4. Eine Chomsky-Grammatik ist vom Typ 2 (auch: kontextfrei), falls sie monoton ist und für alle Regeln $\alpha \to \beta \in P$ gilt:

$$\alpha \in V \ .$$

2.2 Die Chomsky-Hierarchie

Sei $G = (V, \Sigma, P, S)$ eine Phrasenstrukturgrammatik.

3. Eine Chomsky-Grammatik ist vom Typ 1 (auch: kontextsensitiv), falls sie monoton ist und für alle Regeln $\alpha \to \beta$ in P mit $\alpha \neq S$ gilt:

$$\alpha = \alpha' A \alpha'' \text{ und } \beta = \alpha' \beta' \alpha''$$

   für geeignete $A \in V$, $\alpha', \alpha'' \in (V \cup \Sigma)^*$ und $\beta' \in (V \cup \Sigma)^+$.

4. Eine Chomsky-Grammatik ist vom Typ 2 (auch: kontextfrei), falls sie monoton ist und für alle Regeln $\alpha \to \beta \in P$ gilt:

$$\alpha \in V \; .$$

2.2 Die Chomsky-Hierarchie

Sei $G = (V, \Sigma, P, S)$ eine Phrasenstrukturgrammatik.

3. Eine Chomsky-Grammatik ist vom Typ 1 (auch: kontextsensitiv), falls sie monoton ist und für alle Regeln $\alpha \rightarrow \beta$ in P mit $\alpha \neq S$ gilt:

$$\alpha = \alpha' A \alpha'' \text{ und } \beta = \alpha' \beta' \alpha''$$

   für geeignete $A \in V$, $\alpha', \alpha'' \in (V \cup \Sigma)^*$ und $\beta' \in (V \cup \Sigma)^+$.

4. Eine Chomsky-Grammatik ist vom Typ 2 (auch: kontextfrei), falls sie monoton ist und für alle Regeln $\alpha \rightarrow \beta \in P$ gilt:

$$\alpha \in V \ .$$

   Bemerkung: Manchmal wird "kontextfrei" auch ohne die Monotonie-Bedingung definiert; streng monoton schließt dann die Monotonie mit ein.

2.2 Die Chomsky-Hierarchie

Sei $G = (V, \Sigma, P, S)$ eine Phrasenstrukturgrammatik.

5. Eine Chomsky-Grammatik ist vom Typ 3 (auch: regulär, rechtslinear), falls sie monoton ist und für alle Regeln $\alpha \to \beta$ in P mit $\beta \neq \epsilon$ gilt:

$$\alpha \in V \text{ und } \beta \in \Sigma^+ \cup \Sigma^* V \ .$$

2.2 Die Chomsky-Hierarchie

Sei $G = (V, \Sigma, P, S)$ eine Phrasenstrukturgrammatik.

5. Eine Chomsky-Grammatik ist vom Typ 3 (auch: regulär, rechtslinear), falls sie monoton ist und für alle Regeln $\alpha \to \beta$ in P mit $\beta \neq \epsilon$ gilt:

$$\alpha \in V \text{ und } \beta \in \Sigma^{+} \cup \Sigma^{*}V \ .$$

   Auch hier gilt die entsprechende Bemerkung zur Monotonie-Bedingung.

Beispiel 6

- Die folgende Grammatik ist regulär:

$$S \rightarrow \epsilon,\ S \rightarrow A,$$
$$A \rightarrow aa,\ A \rightarrow aaA$$

- Die Produktion

$$A \rightarrow Bc$$

heißt linkslinear.

- Die Produktion

$$A \rightarrow aBc$$

heißt linear.

Beispiel 6

- Die folgende Grammatik ist regulär:

$$S \rightarrow \epsilon,\ S \rightarrow A,$$
$$A \rightarrow aa,\ A \rightarrow aaA$$

- Die Produktion

$$A \rightarrow Bc$$

  heißt linkslinear.

- Die Produktion

$$A \rightarrow aBc$$

  heißt linear.

Beispiel 6

- Die folgende Grammatik ist regulär:
$$S \rightarrow \epsilon,\ S \rightarrow A,$$
$$A \rightarrow aa,\ A \rightarrow aaA$$
- Die Produktion
$$A \rightarrow Bc$$
heißt linkslinear.
- Die Produktion
$$A \rightarrow aBc$$
heißt linear.

Definition 7

Eine Sprache $L \subseteq \Sigma^*$ heißt vom Typ k, $k \in \{0,1,2,3\}$, falls es eine Chomsky-k-Grammatik G mit $L(G) = L$ gibt.

In der Chomsky-Hierarchie bilden also die Typ-3- oder regulären Sprache die kleinste, unterste Stufe, darüber kommen die kontextfreien, dann die kontextsensitiven Sprachen. Oberhalb der Typ-1-Sprachen kommen die Typ-0-Sprachen, die auch rekursiv aufzählbar oder semientscheidbar genannt werden. Darüber (und nicht mehr Teil der Chomsky-Hierarchie) findet sich die Klasse aller formalen Sprachen.

In Typ-3-Grammatiken müssen entweder alle Produktionen rechtslinear oder alle linkslinear sein.
Überlegen Sie sich eine lineare Grammatik, deren Sprache nicht regulär ist!

Definition 7

Eine Sprache $L \subseteq \Sigma^*$ heißt vom Typ k, $k \in \{0, 1, 2, 3\}$, falls es eine Chomsky-k-Grammatik G mit $L(G) = L$ gibt.

In der Chomsky-Hierarchie bilden also die Typ-3- oder regulären Sprache die kleinste, unterste Stufe, darüber kommen die kontextfreien, dann die kontextsensitiven Sprachen. Oberhalb der Typ-1-Sprachen kommen die Typ-0-Sprachen, die auch rekursiv aufzählbar oder semientscheidbar genannt werden. Darüber (und nicht mehr Teil der Chomsky-Hierarchie) findet sich die Klasse aller formalen Sprachen.

In Typ-3-Grammatiken müssen entweder alle Produktionen rechtslinear oder alle linkslinear sein.

Überlegen Sie sich eine lineare Grammatik, deren Sprache nicht regulär ist!

Definition 7

Eine Sprache $L \subseteq \Sigma^*$ heißt vom Typ k, $k \in \{0, 1, 2, 3\}$, falls es eine Chomsky-k-Grammatik G mit $L(G) = L$ gibt.

In der Chomsky-Hierarchie bilden also die Typ-3- oder regulären Sprache die kleinste, unterste Stufe, darüber kommen die kontextfreien, dann die kontextsensitiven Sprachen. Oberhalb der Typ-1-Sprachen kommen die Typ-0-Sprachen, die auch rekursiv aufzählbar oder semientscheidbar genannt werden. Darüber (und nicht mehr Teil der Chomsky-Hierarchie) findet sich die Klasse aller formalen Sprachen.

In Typ-3-Grammatiken müssen entweder alle Produktionen rechtslinear oder alle linkslinear sein.

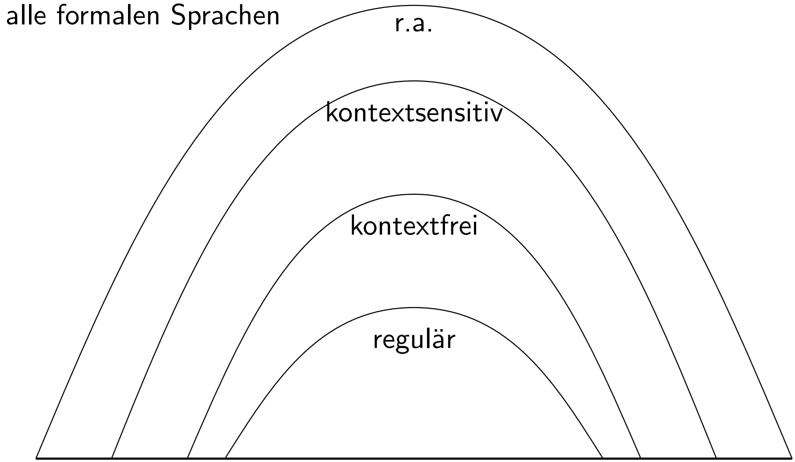
Überlegen Sie sich eine lineare Grammatik, deren Sprache nicht regulär ist!

Definition 7

Eine Sprache $L \subseteq \Sigma^*$ heißt vom Typ k, $k \in \{0, 1, 2, 3\}$, falls es eine Chomsky-k-Grammatik G mit $L(G) = L$ gibt.

In der Chomsky-Hierarchie bilden also die Typ-3- oder regulären Sprache die kleinste, unterste Stufe, darüber kommen die kontextfreien, dann die kontextsensitiven Sprachen. Oberhalb der Typ-1-Sprachen kommen die Typ-0-Sprachen, die auch rekursiv aufzählbar oder semientscheidbar genannt werden. Darüber (und nicht mehr Teil der Chomsky-Hierarchie) findet sich die Klasse aller formalen Sprachen.

In Typ-3-Grammatiken müssen entweder alle Produktionen rechtslinear oder alle linkslinear sein.
Überlegen Sie sich eine lineare Grammatik, deren Sprache nicht regulär ist!


alle formalen Sprachen
r.a.
kontextsensitiv
kontextfrei
regulär

Lemma 8

Sei $G = (V, \Sigma, P, S)$ eine Chomsky-Grammatik, so dass alle Produktionen $\alpha \to \beta$ die Bedingung $\alpha \in V$ erfüllen. Dann ist $L(G)$ kontextfrei.

Beweis:

Definition 9

Ein $A \in V$ mit $A \to^* \epsilon$ heißt nullierbar.

Bestimme alle nullierbaren $A \in V$:

$N := \{A \in V;\ (A \to \epsilon) \in P\}$
$N' := \emptyset$
while $N \neq N'$ **do**
  $N' := N$
  $N := N' \cup \{A \in V;$
    $(\exists (A \to \beta) \in P)[\beta \in N'^*]\}$
od

$\square$

Lemma 8

Sei $G = (V, \Sigma, P, S)$ eine Chomsky-Grammatik, so dass alle Produktionen $\alpha \to \beta$ die Bedingung $\alpha \in V$ erfüllen. Dann ist $L(G)$ kontextfrei.

Beweis:

Definition 9

Ein $A \in V$ mit $A \to^* \epsilon$ heißt nullierbar.

Bestimme alle nullierbaren $A \in V$:

$N := \{A \in V;\ (A \to \epsilon) \in P\}$
$N' := \emptyset$
while $N \neq N'$ **do**
  $N' := N$
  $N := N' \cup \{A \in V;\ (\exists(A \to \beta) \in P)[\beta \in N'^*]\}$
od

$\square$

Lemma 8

Sei $G = (V, \Sigma, P, S)$ eine Chomsky-Grammatik, so dass alle Produktionen $\alpha \to \beta$ die Bedingung $\alpha \in V$ erfüllen. Dann ist $L(G)$ kontextfrei.

Beweis:

Definition 9

Ein $A \in V$ mit $A \to^* \epsilon$ heißt nullierbar.

Bestimme alle nullierbaren $A \in V$:

$N := \{A \in V;\ (A \to \epsilon) \in P\}$
$N' := \emptyset$
while $N \neq N'$ **do**
  $N' := N$
  $N := N' \cup \{A \in V;$
      $(\exists (A \to \beta) \in P)[\beta \in N'^*]\}$
od

$\square$

Lemma 8

Sei $G = (V, \Sigma, P, S)$ eine Chomsky-Grammatik, so dass alle Produktionen $\alpha \to \beta$ die Bedingung $\alpha \in V$ erfüllen. Dann ist $L(G)$ kontextfrei.

Beweis:

Definition 9

Ein $A \in V$ mit $A \to^* \epsilon$ heißt nullierbar.

Bestimme alle nullierbaren $A \in V$:

$N := \{A \in V;\ (A \to \epsilon) \in P\}$
$N' := \emptyset$
while $N \neq N'$ **do**
  $N' := N$
  $N := N' \cup \{A \in V;$
      $(\exists (A \to \beta) \in P)[\beta \in N'^*]\}$
od

$\square$

Lemma 8

Sei $G = (V, \Sigma, P, S)$ eine Chomsky-Grammatik, so dass alle Produktionen $\alpha \rightarrow \beta$ die Bedingung $\alpha \in V$ erfüllen. Dann ist $L(G)$ kontextfrei.

Beweis:

Definition 9

Ein $A \in V$ mit $A \rightarrow^* \epsilon$ heißt nullierbar.

Bestimme alle nullierbaren $A \in V$:

$N := \{A \in V;\ (A \rightarrow \epsilon) \in P\}$
$N' := \emptyset$
while $N \neq N'$ **do**
  $N' := N$
  $N := N' \cup \{A \in V;$
    $(\exists (A \rightarrow \beta) \in P)[\beta \in N'^*]\}$
od

$\square$

Lemma 8

Sei $G = (V, \Sigma, P, S)$ eine Chomsky-Grammatik, so dass alle Produktionen $\alpha \to \beta$ die Bedingung $\alpha \in V$ erfüllen. Dann ist $L(G)$ kontextfrei.

Beweis:

Definition 9

Ein $A \in V$ mit $A \to^* \epsilon$ heißt nullierbar.

Bestimme alle nullierbaren $A \in V$:

$N := \{A \in V;\ (A \to \epsilon) \in P\}$
$N' := \emptyset$
while $N \neq N'$ **do**
  $N' := N$
  $N := N' \cup \{A \in V;$
    $(\exists (A \to \beta) \in P)[\beta \in N'^*]\}$
od

$\square$

Lemma 8

Sei $G = (V, \Sigma, P, S)$ eine Chomsky-Grammatik, so dass alle Produktionen $\alpha \to \beta$ die Bedingung $\alpha \in V$ erfüllen. Dann ist $L(G)$ kontextfrei.

Beweis:

Definition 9

Ein $A \in V$ mit $A \to^* \epsilon$ heißt nullierbar.

Bestimme alle nullierbaren $A \in V$:

$N := \{A \in V;\ (A \to \epsilon) \in P\}$
$N' := \emptyset$
while $N \neq N'$ **do**
  $N' := N$
  $N := N' \cup \{A \in V;$
      $(\exists (A \to \beta) \in P)[\beta \in N'^*]\}$
od

$\square$

Lemma 8

Sei $G = (V, \Sigma, P, S)$ eine Chomsky-Grammatik, so dass alle Produktionen $\alpha \to \beta$ die Bedingung $\alpha \in V$ erfüllen. Dann ist $L(G)$ kontextfrei.

Beweis:

Definition 9

Ein $A \in V$ mit $A \to^* \epsilon$ heißt nullierbar.

Bestimme alle nullierbaren $A \in V$:

$N := \{A \in V;\ (A \to \epsilon) \in P\}$
$N' := \emptyset$
while $N \neq N'$ **do**
  $N' := N$
  $N := N' \cup \{A \in V;$
    $(\exists (A \to \beta) \in P)[\beta \in N'^*]\}$
od

$\square$

Lemma 8

Sei $G = (V, \Sigma, P, S)$ eine Chomsky-Grammatik, so dass alle Produktionen $\alpha \rightarrow \beta$ die Bedingung $\alpha \in V$ erfüllen. Dann ist $L(G)$ kontextfrei.

Beweis:

Definition 9

Ein $A \in V$ mit $A \rightarrow^* \epsilon$ heißt nullierbar.

Bestimme alle nullierbaren $A \in V$:

$N := \{A \in V;\ (A \rightarrow \epsilon) \in P\}$
$N' := \emptyset$
while $N \neq N'$ **do**
  $N' := N$
  $N := N' \cup \{A \in V;$
    $(\exists (A \rightarrow \beta) \in P)[\beta \in N'^*]\}$
od

$\square$

Lemma 8

Sei $G = (V, \Sigma, P, S)$ eine Chomsky-Grammatik, so dass alle Produktionen $\alpha \rightarrow \beta$ die Bedingung $\alpha \in V$ erfüllen. Dann ist $L(G)$ kontextfrei.

Beweis:

Definition 9

Ein $A \in V$ mit $A \rightarrow^* \epsilon$ heißt nullierbar.

Bestimme alle nullierbaren $A \in V$:

$N := \{A \in V;\ (A \rightarrow \epsilon) \in P\}$
$N' := \emptyset$
while $N \neq N'$ **do**
  $N' := N$
  $N := N' \cup \{A \in V;$
    $(\exists (A \rightarrow \beta) \in P)[\beta \in N'^*]\}$
od

Wie man leicht durch Induktion sieht, enthält N zum Schluss alle nullierbaren $A \in V$. □

Sei nun G eine Grammatik, so dass alle linken Seiten $\in V$, aber die Monotoniebedingung nicht unbedingt erfüllt ist.

Modifiziere G zu G' mit Regelmenge P' wie folgt:

1. für jedes $(A \to x_1 x_2 \cdots x_n) \in P$, $n \geq 1$, füge zu P' alle Regeln $A \to y_1 y_2 \cdots y_n$ hinzu, die dadurch entstehen, dass für nicht-nullierbare x_i $y_i := x_i$ und für nullierbare x_i die beiden Möglichkeiten $y_i := x_i$ und $y_i := \epsilon$ eingesetzt werden, ohne dass die ganze rechte Seite $= \epsilon$ wird.
2. falls S nullierbar ist, sei T ein neues Nichtterminal; füge zu P' die Regeln $S \to \epsilon$ und $S \to T$ hinzu, ersetze S in allen rechten Seiten durch T und füge für jede Regel $(S \to x) \in P$, $|x| > 0$, die Regel $T \to x$ zu P' hinzu.

Sei nun G eine Grammatik, so dass alle linken Seiten $\in V$, aber die Monotoniebedingung nicht unbedingt erfüllt ist.
Modifiziere G zu G' mit Regelmenge P' wie folgt:

1. für jedes $(A \to x_1 x_2 \cdots x_n) \in P$, $n \geq 1$, füge zu P' alle Regeln $A \to y_1 y_2 \cdots y_n$ hinzu, die dadurch entstehen, dass für nicht-nullierbare x_i $y_i := x_i$ und für nullierbare x_i die beiden Möglichkeiten $y_i := x_i$ und $y_i := \epsilon$ eingesetzt werden, ohne dass die ganze rechte Seite $= \epsilon$ wird.
2. falls S nullierbar ist, sei T ein neues Nichtterminal; füge zu P' die Regeln $S \to \epsilon$ und $S \to T$ hinzu, ersetze S in allen rechten Seiten durch T und füge für jede Regel $(S \to x) \in P$, $|x| > 0$, die Regel $T \to x$ zu P' hinzu.

Sei nun G eine Grammatik, so dass alle linken Seiten $\in V$, aber die Monotoniebedingung nicht unbedingt erfüllt ist.
Modifiziere G zu G' mit Regelmenge P' wie folgt:

1. für jedes $(A \rightarrow x_1x_2 \cdots x_n) \in P$, $n \geq 1$, füge zu P' alle Regeln $A \rightarrow y_1y_2 \cdots y_n$ hinzu, die dadurch entstehen, dass für nicht-nullierbare x_i $y_i := x_i$ und für nullierbare x_i die beiden Möglichkeiten $y_i := x_i$ und $y_i := \epsilon$ eingesetzt werden, ohne dass die ganze rechte Seite $= \epsilon$ wird.
2. falls S nullierbar ist, sei T ein neues Nichtterminal; füge zu P' die Regeln $S \rightarrow \epsilon$ und $S \rightarrow T$ hinzu, ersetze S in allen rechten Seiten durch T und füge für jede Regel $(S \rightarrow x) \in P$, $|x| > 0$, die Regel $T \rightarrow x$ zu P' hinzu.

Lemma 10

$G' = (V \cup T, \Sigma, P', S)$ *ist kontextfrei, und es gilt*

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

Beweis:

Klar! $\square$

Lemma 10

$G' = (V \cup T, \Sigma, P', S)$ *ist kontextfrei, und es gilt*

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

Beweis:

Klar! $\square$