

Title: Nipkow: Theo (20.05.2019)

Date: Mon May 20 14:14:54 CEST 2019

Duration: 84:55 min

Pages: 98

4.2 Induktive Definitionen, Syntaxbäume und Ableitungen

- 1 Beispiel: Balancierte Klammern
- 2 Induktive Definitionen und Syntaxbäume allgemein
- 3 Äquivalenz von Ableitung, Syntaxbaum und induktiver Erzeugung

Beispiel 4.13

$$S \rightarrow \epsilon \mid [S] \mid SS$$

Beispiel 4.13

$$S \rightarrow \epsilon \mid [S] \mid SS$$

Um zu zeigen, dass diese Grammatik die Menge aller balancierten Klammerwörter in $\{[,]\}^*$ erzeugt, betrachten wir die Grammatik als induktive Definition einer Sprache $L_G(S)$:

	$\epsilon \in L_G(S)$
$u \in L_G(S)$	$\implies [u] \in L_G(S)$
$u \in L_G(S) \wedge v \in L_G(S)$	$\implies uv \in L_G(S)$

Damit gilt zB: $\epsilon \in L(S)$

Satz 4.16

Die Grammatik $S \rightarrow \epsilon \mid [S] \mid SS$ erzeugt genau die Menge der balancierten Wörter.

Beweis:

$u \in L_G(S) \implies u$ balanciert:

Mit Ind. über die Erzeugung von u . (1) bewiesen, wir zeigen (2).

149

Beispiel 4.17

Grammatik:

$$A \rightarrow \epsilon \mid aB \quad B \rightarrow Aa$$

Induktive Definition:

154

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

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

153

Definition 4.18 (Syntaxbaum)

Ein **Syntaxbaum** für eine Ableitung mit einer Grammatik $G = (V, \Sigma, P, S)$ ist ein Baum, so dass

- jedes Blatt mit einem Zeichen aus $\Sigma \cup \{\epsilon\}$ beschriftet ist,

155

Definition 4.18 (Syntaxbaum)

Ein **Syntaxbaum** für eine Ableitung mit einer Grammatik $G = (V, \Sigma, P, S)$ ist ein Baum, so dass

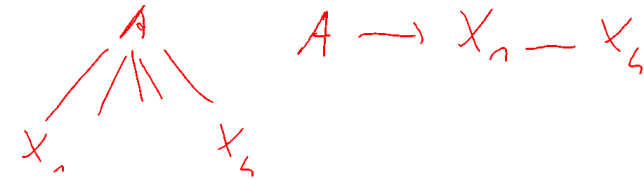
- jedes Blatt mit einem Zeichen aus $\Sigma \cup \{\epsilon\}$ beschriftet ist,
- jeder innere Knoten mit einem $A \in V$ beschriftet ist,

155

Definition 4.18 (Syntaxbaum)

Ein **Syntaxbaum** für eine Ableitung mit einer Grammatik $G = (V, \Sigma, P, S)$ ist ein Baum, so dass

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



155

Definition 4.18 (Syntaxbaum)

Ein **Syntaxbaum** für eine Ableitung mit einer Grammatik $G = (V, \Sigma, P, S)$ ist ein Baum, so dass

- jedes Blatt mit einem Zeichen aus $\Sigma \cup \{\epsilon\}$ beschriftet ist,
- jeder innere Knoten mit einem $A \in V$ beschriftet ist, und falls die Nachfolger (von links nach rechts) mit $X_1, \dots, X_n \in V \cup \Sigma \cup \{\epsilon\}$ beschriftet sind, dann ist $A \rightarrow X_1 \dots X_n$ eine Produktion in P ,



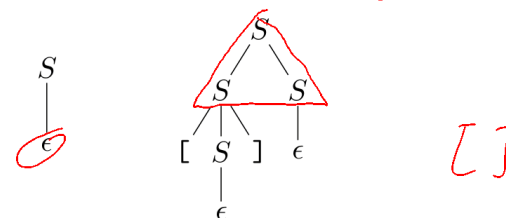
155

Definition 4.18 (Syntaxbaum)

Ein **Syntaxbaum** für eine Ableitung mit einer Grammatik $G = (V, \Sigma, P, S)$ ist ein Baum, so dass

- jedes Blatt mit einem Zeichen aus $\Sigma \cup \{\epsilon\}$ beschriftet ist,
- jeder innere Knoten mit einem $A \in V$ beschriftet ist, und falls die Nachfolger (von links nach rechts) mit $X_1, \dots, X_n \in V \cup \Sigma \cup \{\epsilon\}$ beschriftet sind, dann ist $A \rightarrow X_1 \dots X_n$ eine Produktion in P ,
- ein Blatt ϵ der einzige Nachfolger seines Vorgängers ist.

Beispiel 4.19 (Syntaxbäume für $S \rightarrow \epsilon \mid [S] \mid SS$)



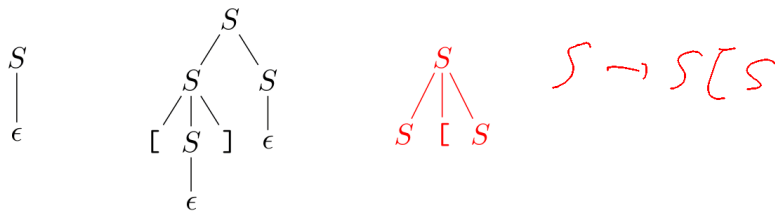
155

Definition 4.18 (Syntaxbaum)

Ein **Syntaxbaum** für eine Ableitung mit einer Grammatik $G = (V, \Sigma, P, S)$ ist ein Baum, so dass

- jedes Blatt mit einem Zeichen aus $\Sigma \cup \{\epsilon\}$ beschriftet ist,
- jeder innere Knoten mit einem $A \in V$ beschriftet ist, und falls die Nachfolger (von links nach rechts) mit $X_1, \dots, X_n \in V \cup \Sigma \cup \{\epsilon\}$ beschriftet sind, dann ist $A \rightarrow X_1 \dots X_n$ eine Produktion in P ,
- ein Blatt ϵ der einzige Nachfolger seines Vorgängers ist.

Beispiel 4.19 (Syntaxbäume für $S \rightarrow \epsilon \mid [S] \mid SS$)



155

Satz 4.20

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

- 1 $A \rightarrow_G^* w$

156

Satz 4.20

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

- 1 $A \rightarrow_G^* w$
- 2 $w \in L_G(A)$ (gemäß der induktiven Definition)

156

156

Satz 4.20

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

- ① $A \rightarrow_G^* w$
- ② $w \in L_G(A)$ (gemäß der induktiven Definition)
- ③ Es gibt einen Syntaxbaum mit Wurzel A , dessen **Rand** (Blätter von links nach rechts gelesen) das Wort w ist.

156

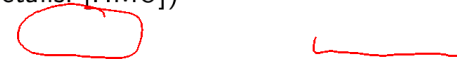
Satz 4.20

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

- ① $A \rightarrow_G^* w$
- ② $w \in L_G(A)$ (gemäß der induktiven Definition)
- ③ Es gibt einen Syntaxbaum mit Wurzel A , dessen **Rand** (Blätter von links nach rechts gelesen) das Wort w ist.

Beweis:

Skizze (Details: [HMU])



156

Satz 4.20

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

- ① $A \rightarrow_G^* w$
- ② $w \in L_G(A)$ (gemäß der induktiven Definition)
- ③ Es gibt einen Syntaxbaum mit Wurzel A , dessen **Rand** (Blätter von links nach rechts gelesen) das Wort w ist.

Beweis:

Skizze (Details: [HMU])

$1 \Rightarrow 2$: Sei $A \rightarrow_G^n w$. Wir beweisen $w \in L_G(A)$ mit Ind. über n . Die Ableitung muss von folgender Form sein:

$$A \rightarrow_G \underbrace{w_0 A_1 w_1 \dots A_m w_m}_{\text{---}} \rightarrow_G^{n-1} w$$

156

Satz 4.20

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

- ① $A \rightarrow_G^* w$
- ② $w \in L_G(A)$ (gemäß der induktiven Definition)
- ③ Es gibt einen Syntaxbaum mit Wurzel A , dessen **Rand** (Blätter von links nach rechts gelesen) das Wort w ist.

Beweis:

Skizze (Details: [HMU])

$1 \Rightarrow 2$: Sei $A \rightarrow_G^n w$. Wir beweisen $w \in L_G(A)$ mit Ind. über n . Die Ableitung muss von folgender Form sein:

$$A \rightarrow_G \underbrace{w_0 A_1 w_1 \dots A_m w_m}_{\text{---}} \rightarrow_G^{n-1} w, \quad u_i \in \Sigma^*$$

Nach dem Dekpositionslemma muss es u_i und $n_i \leq n - 1$ geben mit $A_i \rightarrow_G^{n_i} u_i$ und $w = w_0 u_1 w_1 \dots u_m w_m$.

156

Satz 4.20

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

- 1 $A \rightarrow_G^* w$
- 2 $w \in L_G(A)$ (gemäß der induktiven Definition)
- 3 Es gibt einen Syntaxbaum mit Wurzel A , dessen **Rand** (Blätter von links nach rechts gelesen) das Wort w ist.

Beweis:

Skizze (Details: [HMU])

1 \Rightarrow 2: Sei $A \rightarrow_G^n w$. Wir beweisen $w \in L_G(A)$ mit Ind. über n . Die Ableitung muss von folgender Form sein:

$$A \rightarrow_G w_0 A_1 w_1 \dots A_m w_m \rightarrow_G^{n-1} w$$

Nach dem Dekompositionslemma muss es u_i und $n_i \leq n - 1$ geben mit $A_i \rightarrow_G^{n_i} u_i$ und $w = w_0 u_1 w_1 \dots u_m w_m$. Nach IA (da $n_i < n$) gilt $u_i \in L_G(A_i)$

2 \Rightarrow 3: Parallel zur induktiven Erzeugung von $w \in L_G(A)$ kann man einen Syntaxbaum mit Rand w erzeugen. Formal: Induktion über die Erzeugung von w .

3 \Rightarrow 1: Jeder Baum lässt sich in eine (sogar Links)Ableitung transformieren. Induktion über die Höhe des Baums: Transformiere die Unterbäume t_i mit Beschriftung A_i in Ableitungen $A_i \rightarrow_G^* u_i$ und füge diese mit dem Dekompositionslemma zu einer grossen Ableitung $A \rightarrow_G w_0 A_1 w_1 \dots A_m w_m \rightarrow_G^* w_0 u_1 w_1 \dots u_m w_m$ zusammen.

156

157

Satz 4.20

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

- 1 $A \rightarrow_G^* w$
- 2 $w \in L_G(A)$ (gemäß der induktiven Definition)
- 3 Es gibt einen Syntaxbaum mit Wurzel A , dessen **Rand** (Blätter von links nach rechts gelesen) das Wort w ist.

Beweis:

Skizze (Details: [HMU])

1 \Rightarrow 2: Sei $A \rightarrow_G^n w$. Wir beweisen $w \in L_G(A)$ mit Ind. über n . Die Ableitung muss von folgender Form sein:

$$A \rightarrow_G w_0 A_1 w_1 \dots A_m w_m \rightarrow_G^{n-1} w$$

Nach dem Dekompositionslemma muss es u_i und $n_i \leq n - 1$ geben mit $A_i \rightarrow_G^{n_i} u_i$ und $w = w_0 u_1 w_1 \dots u_m w_m$. Nach IA (da $n_i < n$) gilt $u_i \in L_G(A_i)$ und daher (mit einem weiteren Erzeugungsschritt) auch $w \in L_G(A)$.

156

158

Definition 4.21

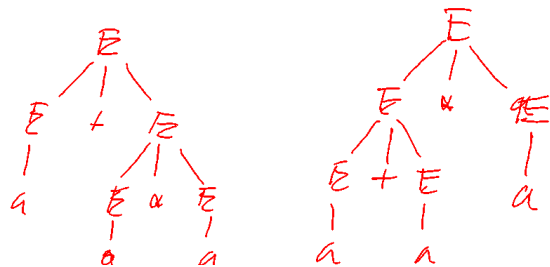
- Eine CFG G heißt **mehrdeutig** gdw es ein $w \in L(G)$ gibt, das zwei verschiedene Syntaxbäume hat, also zwei verschiedene Syntaxbäume mit Wurzel S und Rand w .
- Eine CFL L heißt **inhärent mehrdeutig** gdw jede CFG G mit $L(G) = L$ mehrdeutig ist.

Definition 4.21

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

Beispiel 4.22

Die Grammatik $E \rightarrow E + E \mid E * E \mid (E) \mid a$ ist mehrdeutig — betrachte $a + a * a$.



158

Definition 4.21

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

Beispiel 4.22

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

Satz 4.23

Die Sprache $\{a^i b^j c^k \mid i = j \vee j = k\}$ ist inhärent mehrdeutig.

158

Definition 4.21

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

Beispiel 4.22

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

158

4.3 Die Chomsky-Normalform

Definition 4.24

Eine kontextfreie Grammatik G ist in **Chomsky-Normalform** gdw alle Produktionen eine der Formen

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

haben.

159

4.3 Die Chomsky-Normalform

Definition 4.24

Eine kontextfreie Grammatik G ist in **Chomsky-Normalform** gdw alle Produktionen eine der Formen

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

haben.

Wir konstruieren und beweisen nun schrittweise:

Satz 4.25

Zu jeder CFG G kann man eine CFG G' in Chomsky-Normalform konstruieren mit $\underline{L(G') = L(G) \setminus \{\epsilon\}}$.

Wir nennen $A \rightarrow \epsilon$ eine ϵ -Produktion.

159

160

4.3 Die Chomsky-Normalform

Definition 4.24

Eine kontextfreie Grammatik G ist in **Chomsky-Normalform** gdw alle Produktionen eine der Formen

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

haben.

Wir konstruieren und beweisen nun schrittweise:

Satz 4.25

Zu jeder CFG G kann man eine CFG G' in Chomsky-Normalform konstruieren mit $L(G') = L(G) \setminus \{\epsilon\}$.

Wer auf $\epsilon \in L(G')$ nicht verzichten möchte:

Füge am Ende wieder $\underline{S \rightarrow \epsilon}$ hinzu.

Wir nennen $A \rightarrow \epsilon$ eine ϵ -Produktion.

Lemma 4.26

Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine ϵ -Produktionen enthält, so dass gilt $L(G') = L(G) \setminus \{\epsilon\}$

159

160

Wir nennen $A \rightarrow \epsilon$ eine ϵ -Produktion.

Lemma 4.26

Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine ϵ -Produktionen enthält, so dass gilt $L(G') = L(G) \setminus \{\epsilon\}$

Beweis:

Wir erweitern P induktiv zu eine Obermenge \hat{P} :

160

Wir nennen $A \rightarrow \epsilon$ eine ϵ -Produktion.

Lemma 4.26

Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine ϵ -Produktionen enthält, so dass gilt $L(G') = L(G) \setminus \{\epsilon\}$

Beweis:

Wir erweitern P induktiv zu eine Obermenge \hat{P} :

- 1 Jede Produktion aus P ist in \hat{P}

160

Wir nennen $A \rightarrow \epsilon$ eine ϵ -Produktion.

Lemma 4.26

Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine ϵ -Produktionen enthält, so dass gilt $L(G') = L(G) \setminus \{\epsilon\}$

Beweis:

Wir erweitern P induktiv zu eine Obermenge \hat{P} :

- 1 Jede Produktion aus P ist in \hat{P}
- 2 Sind $B \rightarrow \epsilon$ und $A \rightarrow \alpha B \beta$ in \hat{P} , so füge auch $A \rightarrow \alpha \beta$ hinzu.

Offensichtlich gilt $L(\hat{G}) = L(G)$: Jede neue Produktion kann von 2 alten simuliert werden.

160

Wir nennen $A \rightarrow \epsilon$ eine ϵ -Produktion.

Lemma 4.26

Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine ϵ -Produktionen enthält, so dass gilt $L(G') = L(G) \setminus \{\epsilon\}$

Beweis:

Wir erweitern P induktiv zu eine Obermenge \hat{P} :

- 1 Jede Produktion aus P ist in \hat{P}
- 2 Sind $B \rightarrow \epsilon$ und $A \rightarrow \alpha B \beta$ in \hat{P} , so füge auch $A \rightarrow \alpha \beta$ hinzu.

Offensichtlich gilt $L(\hat{G}) = L(G)$: Jede neue Produktion kann von 2 alten simuliert werden.

160

Wir nennen $A \rightarrow \epsilon$ eine ϵ -Produktion.

Lemma 4.26

Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine ϵ -Produktionen enthält, so dass gilt

$$L(G') = L(G) \setminus \{\epsilon\}$$

Beweis:

Wir erweitern P induktiv zu eine Obermenge \hat{P} :

- 1 Jede Produktion aus P ist in \hat{P}
- 2 Sind $B \rightarrow \epsilon$ und $A \rightarrow \alpha B \beta$ in \hat{P} , so füge auch $A \rightarrow \alpha \beta$ hinzu.

Offensichtlich gilt $L(\hat{G}) = L(G)$: Jede neue Produktion kann von 2 alten simuliert werden.

Wir definieren G' als \hat{G} ohne die ϵ -Produktionen.
Denn diese sind nun überflüssig:

160

Wir nennen $A \rightarrow \epsilon$ eine ϵ -Produktion.

Lemma 4.26

Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine ϵ -Produktionen enthält, so dass gilt

$$L(G') = L(G) \setminus \{\epsilon\}$$

Beweis:

Wir erweitern P induktiv zu eine Obermenge \hat{P} :

- 1 Jede Produktion aus P ist in \hat{P}
- 2 Sind $B \rightarrow \epsilon$ und $A \rightarrow \alpha B \beta$ in \hat{P} , so füge auch $A \rightarrow \alpha \beta$ hinzu.

Offensichtlich gilt $L(\hat{G}) = L(G)$: Jede neue Produktion kann von 2 alten simuliert werden.

Wir definieren G' als \hat{G} ohne die ϵ -Produktionen.
Denn diese sind nun überflüssig:

160

Beweis (Forts.):

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

161

Beweis (Forts.):

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

161

Beweis (Forts.):

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

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

Beweis (Forts.):

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

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

so wäre γ oder δ nichtleer,

Beweis (Forts.):

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

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

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

Beweis (Forts.):

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

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

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

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

Beweis (Forts.):

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

$$S \xrightarrow{\hat{G}}^* \gamma B \delta \xrightarrow{\hat{G}} \gamma \delta \xrightarrow{\hat{G}}^* w \quad ||$$

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

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

Dann wäre aber folgende Ableitung mit $(A \rightarrow \alpha \beta) \in \hat{P}$ kürzer:

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

Beispiel 4.27

$$S \rightarrow AB, \quad A \rightarrow \underline{aAA} \mid B, \quad B \rightarrow bBS \mid \epsilon$$

$$A \rightarrow \epsilon \quad B \rightarrow bS \quad S \rightarrow A \cdot$$

$$A \rightarrow aA \quad S \rightarrow B \cdot \quad S \rightarrow \epsilon$$

$$A \rightarrow a \quad B \rightarrow b$$

Beweis (Forts.):

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

$$S \xrightarrow{\hat{G}}^* \gamma B \delta \xrightarrow{\hat{G}} \gamma \delta \xrightarrow{\hat{G}}^* w$$

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

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

Dann wäre aber folgende Ableitung mit $(A \rightarrow \alpha \beta) \in \hat{P}$ kürzer:

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

Also kommen in Ableitungen minimaler Länge keine ϵ -Produktionen vor. Letztere sind also überflüssig. □

Wir nennen $A \rightarrow B$ eine Kettenproduktion.

Beispiel 4.27

$$S \rightarrow AB, \quad A \rightarrow aAA \mid B, \quad B \rightarrow bBS \mid \epsilon$$

Beweis (Forts.):

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



Beweis (Forts.):

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

$$S \xrightarrow{*}_{\hat{G}} \gamma B \delta \xrightarrow{\hat{G}} \gamma \delta \xrightarrow{*}_{\hat{G}} w$$

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

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

Dann wäre aber folgende Ableitung mit $(A \rightarrow \alpha \beta) \in \hat{P}$ kürzer:

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

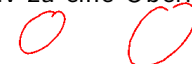
Wir nennen $A \rightarrow B$ eine Kettenproduktion.

Lemma 4.28

Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine Kettenproduktionen enthält, so dass gilt $L(G') = L(G)$.

Beweis:

Wir erweitern P induktiv zu eine Obermenge \hat{P} :



Wir nennen $A \rightarrow B$ eine Kettenproduktion.

Lemma 4.28

Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine Kettenproduktionen enthält, so dass gilt $L(G') = L(G)$.

Beweis:

Wir erweitern P induktiv zu eine Obermenge \hat{P} :

- 1 Jede Produktion aus P ist in \hat{P}
- 2 Sind $A \rightarrow B$ und $B \rightarrow \alpha$ in \hat{P} mit $\alpha \neq A$, so füge auch $A \rightarrow \alpha$ hinzu.

Wir nennen $A \rightarrow B$ eine Kettenproduktion.

Lemma 4.28

Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine Kettenproduktionen enthält, so dass gilt $L(G') = L(G)$.

Beweis:

Wir erweitern P induktiv zu eine Obermenge \hat{P} :

- 1 Jede Produktion aus P ist in \hat{P}
- 2 Sind $A \rightarrow B$ und $B \rightarrow \alpha$ in \hat{P} mit $\alpha \neq A$, so füge auch $A \rightarrow \alpha$ hinzu.

Das Ergebnis G' ist \hat{G} ohne die (nun überflüssigen) Kettenproduktionen.

Rest des Beweises analog zur Elimination von ϵ -Produktionen. □

Wir nennen $A \rightarrow B$ eine Kettenproduktion.

Lemma 4.28

Zu jeder CFG $G = (V, \Sigma, P, S)$ kann man eine CFG G' konstruieren, die keine Kettenproduktionen enthält, so dass gilt $L(G') = L(G)$.

Beweis:

Wir erweitern P induktiv zu eine Obermenge \hat{P} :

- 1 Jede Produktion aus P ist in \hat{P}
- 2 Sind $A \rightarrow B$ und $B \rightarrow \alpha$ in \hat{P} mit $\alpha \neq A$, so füge auch $A \rightarrow \alpha$ hinzu.

Das Ergebnis G' ist \hat{G} ohne die (nun überflüssigen) Kettenproduktionen.

Konstruktion einer Chomsky-Normalform

Konstruktion einer Chomsky-Normalform

Eingabe: Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$

165

Konstruktion einer Chomsky-Normalform

$A \rightarrow a$

Eingabe: Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$

- 1 Füge für jedes $a \in \Sigma$, das in einer rechten Seite der Länge ≥ 2 vorkommt, ein neues Nichtterminal A_a zu V hinzu, ersetze a in allen rechten Seiten der Länge ≥ 2 durch A_a , und füge $A_a \rightarrow a$ zu P hinzu.

$A \rightarrow a$ $A \rightarrow B_1 B_2 \dots B_n$

165

Konstruktion einer Chomsky-Normalform

$A \rightarrow a$ | $A \rightarrow BC$

Eingabe: Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$

- 1 Füge für jedes $a \in \Sigma$, das in einer rechten Seite der Länge ≥ 2 vorkommt, ein neues Nichtterminal A_a zu V hinzu, ersetze a in allen rechten Seiten der Länge ≥ 2 durch A_a , und füge $A_a \rightarrow a$ zu P hinzu.

165

Konstruktion einer Chomsky-Normalform

Eingabe: Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$

- 1 Füge für jedes $a \in \Sigma$, das in einer rechten Seite der Länge ≥ 2 vorkommt, ein neues Nichtterminal A_a zu V hinzu, ersetze a in allen rechten Seiten der Länge ≥ 2 durch A_a , und füge $A_a \rightarrow a$ zu P hinzu.
- 2 Ersetze jede Produktion der Form

$$A \rightarrow B_1 B_2 \dots B_k \quad (k \geq 3)$$

durch

$$A \rightarrow B_1 C_2, \quad C_2 \rightarrow B_2 C_3, \quad \dots, \quad C_{k-1} \rightarrow B_{k-1} B_k$$

wobei C_2, \dots, C_{k-1} neue Nichtterminale sind.

165

Konstruktion einer Chomsky-Normalform

$$A \rightarrow a \quad A \rightarrow BC$$

Eingabe: Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$

- 1 Füge für jedes $a \in \Sigma$, das in einer rechten Seite der Länge ≥ 2 vorkommt, ein neues Nichtterminal A_a zu V hinzu, ersetze a in allen rechten Seiten der Länge ≥ 2 durch A_a , und füge $A_a \rightarrow a$ zu P hinzu.
- 2 Ersetze jede Produktion der Form

$$A \rightarrow B_1 B_2 \cdots B_k \quad (k \geq 3)$$

durch

$$A \rightarrow B_1 C_2, \quad C_2 \rightarrow B_2 C_3, \quad \dots, \quad C_{k-1} \rightarrow B_{k-1} B_k$$

wobei C_2, \dots, C_{k-1} neue Nichtterminale sind.

- 3 Eliminiere alle ϵ -Produktionen.

165

Definition 4.30

Eine CFG ist in **Greibach-Normalform** (benannt nach Sheila Greibach, UCLA), falls jede Produktion von der Form

$$A \rightarrow a A_1 \dots A_n$$

ist.

167

Aufgabe:

- Zeige: Mit diesem Algorithmus ist das Ergebnis höchstens quadratisch größer als G .
- Zeige: die Reihenfolge der Schritte ist wichtig!
 - Finde eine andere Reihenfolge, die inkorrekt ist: Das resultierende Verfahren ergibt nicht immer eine Grammatik in CNF.
 - Finde eine andere Reihenfolge, die zwar zu einem korrekten Verfahren führt, aber eine Grammatik in CNF produzieren kann, die exponentiell größer als G ist.

166

Aufgabe:

- Zeige: Mit diesem Algorithmus ist das Ergebnis höchstens quadratisch größer als G .
- Zeige: die Reihenfolge der Schritte ist wichtig!
 - Finde eine andere Reihenfolge, die inkorrekt ist: Das resultierende Verfahren ergibt nicht immer eine Grammatik in CNF.
 - Finde eine andere Reihenfolge, die zwar zu einem korrekten Verfahren führt, aber eine Grammatik in CNF produzieren kann, die exponentiell größer als G ist.

166

Definition 4.30

Eine CFG ist in Greibach-Normalform (benannt nach Sheila Greibach, UCLA), falls jede Produktion von der Form

$$\underline{A \rightarrow aA_1 \dots A_n}$$

ist.

Satz 4.31

Zu jeder CFG G gibt es eine CFG G' in Greibach-Normalform mit $L(G') = L(G) \setminus \{\epsilon\}$.

167

4.4 Das Pumping-Lemma für kontextfreie Sprachen

Zur Erinnerung: Das Pumping-Lemma für reguläre Sprachen: Für jede reguläre Sprache L gibt es ein $n \in \mathbb{N}$, so dass sich jedes Wort $z \in L$ mit $|z| \geq n$ zerlegen lässt in $z = uvw$ mit $|uv| \leq n$, $v \neq \epsilon$ und $uv^*w \subseteq L$.

Zum Beweis haben wir $n = |Q|$ gewählt, wobei Q die Zustandsmenge eines L erkennenden DFA war. Das Argument war dann, dass beim Erkennen von z (mindestens) ein Zustand zweimal besucht werden muss und damit der dazwischen liegende Weg im Automaten beliebig oft wiederholt werden kann.

168

4.4 Das Pumping-Lemma für kontextfreie Sprachen

Zur Erinnerung: Das Pumping-Lemma für reguläre Sprachen: Für jede reguläre Sprache L gibt es ein $n \in \mathbb{N}$, so dass sich jedes Wort $z \in L$ mit $|z| \geq n$ zerlegen lässt in $z = uvw$ mit $|uv| \leq n$, $v \neq \epsilon$ und $uv^*w \subseteq L$.

168

4.4 Das Pumping-Lemma für kontextfreie Sprachen

Zur Erinnerung: Das Pumping-Lemma für reguläre Sprachen: Für jede reguläre Sprache L gibt es ein $n \in \mathbb{N}$, so dass sich jedes Wort $z \in L$ mit $|z| \geq n$ zerlegen lässt in $z = uvw$ mit $|uv| \leq n$, $v \neq \epsilon$ und $uv^*w \subseteq L$.

Zum Beweis haben wir $n = |Q|$ gewählt, wobei Q die Zustandsmenge eines L erkennenden DFA war. Das Argument war dann, dass beim Erkennen von z (mindestens) ein Zustand zweimal besucht werden muss und damit der dazwischen liegende Weg im Automaten beliebig oft wiederholt werden kann.

Ein ähnliches Argument kann auch auf kontextfreie Grammatiken angewendet werden.

168

Satz 4.32 (Pumping-Lemma)

Für jede kontextfreie Sprache L gibt es ein $n \geq 1$, so dass sich jedes Wort $z \in L$ mit $|z| \geq n$ zerlegen lässt in

$$z = uvwxy,$$

mit

- $vx \neq \epsilon$,

169

Satz 4.32 (Pumping-Lemma)

Für jede kontextfreie Sprache L gibt es ein $n \geq 1$, so dass sich jedes Wort $z \in L$ mit $|z| \geq n$ zerlegen lässt in

$$z = uvwxy,$$

mit

- $vx \neq \epsilon$,
- $|vwx| \leq n$, und
- $\forall i \in \mathbb{N}. uv^iwx^iy \in L$.

169

Satz 4.32 (Pumping-Lemma)

Für jede kontextfreie Sprache L gibt es ein $n \geq 1$, so dass sich jedes Wort $z \in L$ mit $|z| \geq n$ zerlegen lässt in

$$z = uvwxy,$$

mit

- $vx \neq \epsilon$,
- $|vwx| \leq n$, und

169

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine CFG in Chomsky-Normalform mit $L(G) = L \setminus \{\epsilon\}$.

170

Satz 4.32 (Pumping-Lemma)

Für jede kontextfreie Sprache L gibt es ein $n \geq 1$, so dass sich jedes Wort $z \in L$ mit $|z| \geq n$ zerlegen lässt in

$$z = uvwxy,$$

mit

- $vx \neq \epsilon$,
- $|vwx| \leq n$, und
- $\forall i \in \mathbb{N}. uv^iwx^iy \in L$.

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine CFG in Chomsky-Normalform mit $L(G) = L \setminus \{\epsilon\}$. Wähle $n = 2^{|V|}$.

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine CFG in Chomsky-Normalform mit $L(G) = L \setminus \{\epsilon\}$.

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine CFG in Chomsky-Normalform mit $L(G) = L \setminus \{\epsilon\}$.

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine CFG in Chomsky-Normalform mit $L(G) = L \setminus \{\epsilon\}$. Wähle $n = 2^{\lfloor \log_2 n \rfloor}$. Sei $z \in L(G)$ mit $|z| \geq n$.

Jeder Binärbaum mit $\geq 2^k$ Blättern hat die Höhe $\geq k$

170

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine CFG in Chomsky-Normalform mit $L(G) = L \setminus \{\epsilon\}$. Wähle $n = 2^{\lfloor \log_2 n \rfloor}$. Sei $z \in L(G)$ mit $|z| \geq n$.

Jeder Binärbaum mit $\geq 2^k$ Blättern hat die Höhe $\geq k$

Dann hat der Syntaxbaum für z (ohne die letzte Stufe für die Terminale) mindestens einen Pfad der Länge $\geq |V|$, da er wegen der Chomsky-Normalform ein Binärbaum ist. Wir betrachten einen solchen Pfad maximaler Länge.

170

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine CFG in Chomsky-Normalform mit $L(G) = L \setminus \{\epsilon\}$. Wähle $n = 2^{\lfloor \log_2 n \rfloor}$. Sei $z \in L(G)$ mit $|z| \geq n$.

Jeder Binärbaum mit $\geq 2^k$ Blättern hat die Höhe $\geq k$

Dann hat der Syntaxbaum für z (ohne die letzte Stufe für die Terminale) mindestens einen Pfad der Länge $\geq |V|$, da er wegen der Chomsky-Normalform ein Binärbaum ist.

170

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine CFG in Chomsky-Normalform mit $L(G) = L \setminus \{\epsilon\}$. Wähle $n = 2^{\lfloor \log_2 n \rfloor}$. Sei $z \in L(G)$ mit $|z| \geq n$.

Jeder Binärbaum mit $\geq 2^k$ Blättern hat die Höhe $\geq k$

Dann hat der Syntaxbaum für z (ohne die letzte Stufe für die Terminale) mindestens einen Pfad der Länge $\geq |V|$, da er wegen der Chomsky-Normalform ein Binärbaum ist. Wir betrachten einen solchen Pfad maximaler Länge.

Auf diesem Pfad der Länge $\geq |V|$ kommen $\geq |V| + 1$ Nichtterminale vor, also mindestens eins doppelt.

170

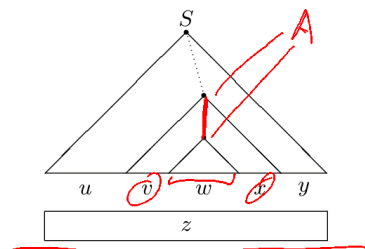
Beweis:

Sei $G = (V, \Sigma, P, S)$ eine CFG in Chomsky-Normalform mit $L(G) = L \setminus \{\epsilon\}$. Wähle $n = 2^{|V|}$. Sei $z \in L(G)$ mit $|z| \geq n$.

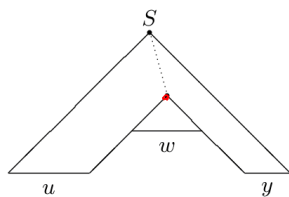
Jeder Binärbaum mit $\geq 2^k$ Blättern hat die Höhe $\geq k$

Dann hat der Syntaxbaum für z (ohne die letzte Stufe für die Terminale) mindestens einen Pfad der Länge $\geq |V|$, da er wegen der Chomsky-Normalform ein Binärbaum ist. Wir betrachten einen solchen Pfad maximaler Länge.

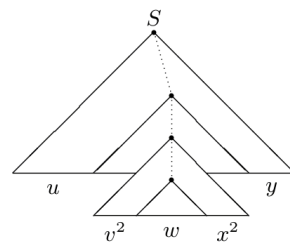
Auf diesem Pfad der Länge $\geq |V|$ kommen $\geq |V| + 1$ Nichtterminale vor, also mindestens eins doppelt. Nennen wir es A .



X



Dieser Ableitungsbaum zeigt $wy \in L$



Dieser Ableitungsbaum zeigt $uv^2wx^2y \in L$

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine CFG in Chomsky-Normalform mit $L(G) = L \setminus \{\epsilon\}$. Wähle $n = 2^{|V|}$. Sei $z \in L(G)$ mit $|z| \geq n$.

Jeder Binärbaum mit $\geq 2^k$ Blättern hat die Höhe $\geq k$

Dann hat der Syntaxbaum für z (ohne die letzte Stufe für die Terminale) mindestens einen Pfad der Länge $\geq |V|$, da er wegen der Chomsky-Normalform ein Binärbaum ist. Wir betrachten einen solchen Pfad maximaler Länge.

Auf diesem Pfad der Länge $\geq |V|$ kommen $\geq |V| + 1$ Nichtterminale vor, also mindestens eins doppelt. Nennen wir es A .

Die zwischen zwei Vorkommen von A liegende Teilableitung kann nun beliebig oft wiederholt werden.

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine CFG in Chomsky-Normalform mit $L(G) = L \setminus \{\epsilon\}$. Wähle $n = 2^{|V|}$. Sei $z \in L(G)$ mit $|z| \geq n$.

Jeder Binärbaum mit $\geq 2^k$ Blättern hat die Höhe $\geq k$

Dann hat der Syntaxbaum für z (ohne die letzte Stufe für die Terminale) mindestens einen Pfad der Länge $\geq |V|$, da er wegen der Chomsky-Normalform ein Binärbaum ist. Wir betrachten einen solchen Pfad maximaler Länge.

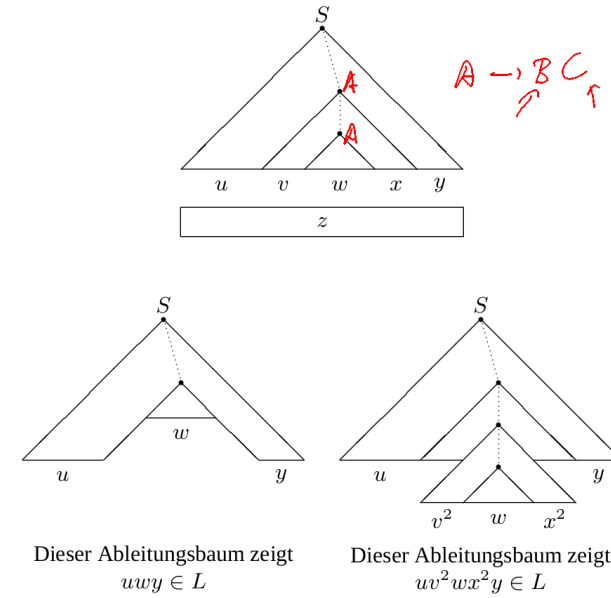
Auf diesem Pfad der Länge $\geq |V|$ kommen $\geq |V| + 1$ Nichtterminale vor, also mindestens eins doppelt. Nennen wir es A .

Die zwischen zwei Vorkommen von A liegende Teilableitung kann nun beliebig oft wiederholt werden.

$i = 0$

Beweis (Forts.):

Die Bedingung $vx \neq \epsilon$ folgt, da das obere der beiden A 's mit $A \rightarrow BC$ abgeleitet werden muss.



Beweis (Forts.):

Die Bedingung $vx \neq \epsilon$ folgt, da das obere der beiden A 's mit $A \rightarrow BC$ abgeleitet werden muss. *CNF: kein $X \rightarrow \epsilon$*

Um $|vwx| \leq n$ zu erreichen, darf das obere A maximal $|V| + 1$ Schritte von jedem Blatt entfernt sein, weil $n = 2^{|V|}$. Da der ausgewählte Pfad maximale Länge hat, genügt es, dass das obere A vom Blatt dieses Pfads $|V| + 1$ Schritte entfernt ist. Da es nur $|V|$ Nichtterminale gibt, muss ein solches doppeltes A existieren. □

Beweis (Forts.):

Die Bedingung $vx \neq \epsilon$ folgt, da das obere der beiden A 's mit $A \rightarrow BC$ abgeleitet werden muss. *A*

Um $|vwx| \leq n$ zu erreichen, darf das obere A maximal $|V| + 1$ Schritte von jedem Blatt entfernt sein, weil $n = 2^{|V|}$. Da der ausgewählte Pfad maximale Länge hat, genügt es, dass das obere A vom Blatt dieses Pfads $|V| + 1$ Schritte entfernt ist. Da es nur $|V|$ Nichtterminale gibt, muss ein solches doppeltes A existieren. □

Beweis (Forts.):

Die Bedingung $vx \neq \epsilon$ folgt, da das obere der beiden A 's mit $A \rightarrow BC$ abgeleitet werden muss.



172

Beispiel 4.33

Wir zeigen, dass die Sprache

$$L := \{a^i b^i c^i \mid i \in \mathbb{N}\}$$

nicht kontextfrei ist.

Wäre L kontextfrei, so gäbe es eine dazugehörige Pumping-Lemma Zahl n und wir könnten jedes $z \in L$ mit $|z| \geq n$, insb. $z = a^n b^n c^n$, zerlegen und aufpumpen, ohne aus der Sprache herauszufallen.

173

Beispiel 4.33

Wir zeigen, dass die Sprache

$$L := \{a^i b^i c^i \mid i \in \mathbb{N}\}$$

nicht kontextfrei ist.

173

Beispiel 4.33

Wir zeigen, dass die Sprache

$$L := \{a^i b^i c^i \mid i \in \mathbb{N}\}$$

nicht kontextfrei ist.

Wäre L kontextfrei, so gäbe es eine dazugehörige Pumping-Lemma Zahl n und wir könnten jedes $z \in L$ mit $|z| \geq n$, insb. $z = a^n b^n c^n$, zerlegen und aufpumpen, ohne aus der Sprache herauszufallen.

Eine Zerlegung $z = uvwxy$ mit $|vwx| \leq n$ bedeutet aber, dass vwx nicht a 's und c 's enthalten kann.

173

Beispiel 4.33

Wir zeigen, dass die Sprache

$$L := \{a^i b^i c^i \mid i \in \mathbb{N}\}$$

nicht kontextfrei ist.

Wäre L kontextfrei, so gäbe es eine dazugehörige Pumping-Lemma Zahl n und wir könnten jedes $z \in L$ mit $|z| \geq n$, insb. $z = a^n b^n c^n$, zerlegen und aufpumpen, ohne aus der Sprache herauszufallen.

Eine Zerlegung $z = uvwxy$ mit $|vwx| \leq n$ bedeutet aber, dass vwx nicht a 's und c 's enthalten kann. OE nehmen wir an, vwx enthält nur a 's und b 's.

173

Beispiel 4.33

Wir zeigen, dass die Sprache

$$L := \{a^i b^i c^i \mid i \in \mathbb{N}\}$$

nicht kontextfrei ist.

Wäre L kontextfrei, so gäbe es eine dazugehörige Pumping-Lemma Zahl n und wir könnten jedes $z \in L$ mit $|z| \geq n$, insb. $z = a^n b^n c^n$, zerlegen und aufpumpen, ohne aus der Sprache herauszufallen.

Eine Zerlegung $z = uvwxy$ mit $|vwx| \leq n$ bedeutet aber, dass vwx nicht a 's und c 's enthalten kann. OE nehmen wir an, vwx enthält nur a 's und b 's. Da $vx \neq \epsilon$, muss vx mindestens ein a oder ein b enthalten.

Damit gilt aber $\#_a(uv^2wx^2y) > \#_c(uv^2wx^2y)$ oder $\#_b(uv^2wx^2y) > \#_c(uv^2wx^2y)$. Also $uv^2wx^2y \notin L$. ζ

173

Beispiel 4.33

Wir zeigen, dass die Sprache

$$L := \{a^i b^i c^i \mid i \in \mathbb{N}\}$$

nicht kontextfrei ist.

Wäre L kontextfrei, so gäbe es eine dazugehörige Pumping-Lemma Zahl n und wir könnten jedes $z \in L$ mit $|z| \geq n$, insb. $z = a^n b^n c^n$, zerlegen und aufpumpen, ohne aus der Sprache herauszufallen.

Eine Zerlegung $z = uvwxy$ mit $|vwx| \leq n$ bedeutet aber, dass vwx nicht a 's und c 's enthalten kann. OE nehmen wir an, vwx enthält nur a 's und b 's. Da $vx \neq \epsilon$, muss vx mindestens ein a oder ein b enthalten.

173

Beispiel 4.34

Mit dem Pumping-Lemma kann man zeigen, dass die Sprache $\{ww \mid w \in \{a, b\}^*\}$ nicht kontextfrei ist.

174

Beispiel 4.34

Mit dem Pumping-Lemma kann man zeigen, dass die Sprache $\{\underline{ww} \mid w \in \{a, b\}^*\}$ nicht kontextfrei ist.

Dies zeigt, dass Kontextbedingungen in Programmiersprachen, etwa „Deklaration vor Benutzung“, nicht durch kontextfreie Grammatiken ausgedrückt werden können.

174

Beispiel 4.34

Mit dem Pumping-Lemma kann man zeigen, dass die Sprache $\{ww \mid w \in \{a, b\}^*\}$ nicht kontextfrei ist.

Dies zeigt, dass Kontextbedingungen in Programmiersprachen, etwa „Deklaration vor Benutzung“, nicht durch kontextfreie Grammatiken ausgedrückt werden können.

174

4.5 Algorithmen für kontextfreie Grammatiken

Wie üblich: $G = (V, \Sigma, P, S)$ ist eine CFG.

175