

Script generated by TTT

Title: Seidl: Theoretische_Informatik
(10.05.2012)

Date: Thu May 10 16:00:49 CEST 2012

Duration: 88:07 min

Pages: 114

3. Kontextfreie Sprachen

3. Kontextfreie Sprachen

3. Kontextfreie Sprachen

3.1 Kontextfreie Grammatiken

Beispiel 3.1 (Arithmetische Ausdrücke)

$\langle \text{Expr} \rangle \rightarrow \langle \text{Term} \rangle$
 $\langle \text{Expr} \rangle \rightarrow \langle \text{Expr} \rangle + \langle \text{Term} \rangle$
 $\langle \text{Term} \rangle \rightarrow \langle \text{Factor} \rangle$
 $\langle \text{Term} \rangle \rightarrow \langle \text{Term} \rangle * \langle \text{Factor} \rangle$
 $\langle \text{Factor} \rangle \rightarrow a$
 $\langle \text{Factor} \rangle \rightarrow (\langle \text{Expr} \rangle)$

Eine (Links)Ableitung:

$\langle \text{Expr} \rangle \rightarrow$

$\rightarrow a * (a + a)$

3. Kontextfreie Sprachen

3.1 Kontextfreie Grammatiken

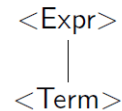
Beispiel 3.1 (Arithmetische Ausdrücke)

$\langle \text{Expr} \rangle \rightarrow \langle \text{Term} \rangle$
 $\langle \text{Expr} \rangle \rightarrow \langle \text{Expr} \rangle + \langle \text{Term} \rangle$
 $\langle \text{Term} \rangle \rightarrow \langle \text{Factor} \rangle$
 $\langle \text{Term} \rangle \rightarrow \langle \text{Term} \rangle * \langle \text{Factor} \rangle$
 $\langle \text{Factor} \rangle \rightarrow a$
 $\langle \text{Factor} \rangle \rightarrow (\langle \text{Expr} \rangle)$

Eine (Links)Ableitung:

$\langle \text{Expr} \rangle \rightarrow \langle \text{Term} \rangle \rightarrow \langle \text{Term} \rangle * \langle \text{Factor} \rangle$
 $\rightarrow \langle \text{Factor} \rangle * \langle \text{Factor} \rangle \rightarrow a * \langle \text{Factor} \rangle$
 $\rightarrow a * (\langle \text{Expr} \rangle) \rightarrow a * (\langle \text{Expr} \rangle + \langle \text{Term} \rangle)$
 $\rightarrow a * (a + a)$

Der Syntaxbaum:



3. Kontextfreie Sprachen

3.1 Kontextfreie Grammatiken

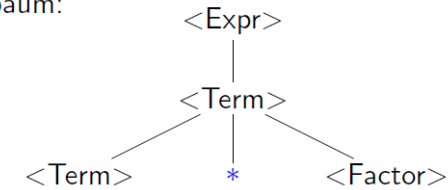
Beispiel 3.1 (Arithmetische Ausdrücke)

$\langle \text{Expr} \rangle \rightarrow \langle \text{Term} \rangle$
 $\langle \text{Expr} \rangle \rightarrow \langle \text{Expr} \rangle + \langle \text{Term} \rangle$
 $\langle \text{Term} \rangle \rightarrow \langle \text{Factor} \rangle$
 $\langle \text{Term} \rangle \rightarrow \langle \text{Term} \rangle * \langle \text{Factor} \rangle$
 $\langle \text{Factor} \rangle \rightarrow a$
 $\langle \text{Factor} \rangle \rightarrow (\langle \text{Expr} \rangle)$

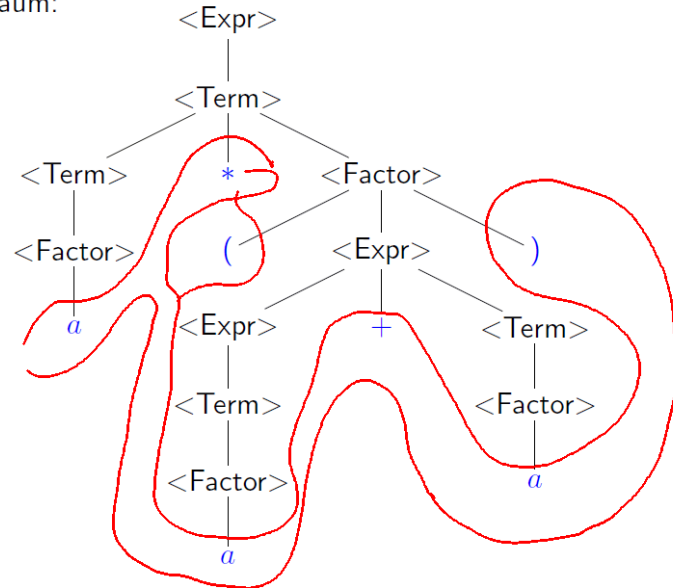
Eine (Links)Ableitung:

$\langle \text{Expr} \rangle \rightarrow \langle \text{Term} \rangle \rightarrow \langle \text{Term} \rangle * \langle \text{Factor} \rangle$
 $\rightarrow \langle \text{Factor} \rangle * \langle \text{Factor} \rangle \rightarrow a * \langle \text{Factor} \rangle$
 $\rightarrow a * (\langle \text{Expr} \rangle) \rightarrow a * (\langle \text{Expr} \rangle + \langle \text{Term} \rangle)$
 $\rightarrow a * (\langle \text{Term} \rangle + \langle \text{Term} \rangle) \rightarrow a * (\langle \text{Factor} \rangle + \langle \text{Term} \rangle)$
 $\rightarrow a * (a + \langle \text{Term} \rangle) \rightarrow a * (a + \langle \text{Factor} \rangle) \rightarrow a * (a + a)$

Der Syntaxbaum:



Der Syntaxbaum:



Die Blätter des Baums, von links nach rechts gelesen, ergeben das abgeleitete Wort

Bemerkungen

- Der vollständige Syntaxbaum enthält die gesamte Information über die Ableitung, bis auf die (irrelevante) Reihenfolge des Aufbaus.
- Kontextfreie Grammatiken dienen zur Spezifikation von Sprachen. Ihre Implementierung ist das Parsen:

Bemerkungen

- Der vollständige Syntaxbaum enthält die gesamte Information über die Ableitung, bis auf die (irrelevante) Reihenfolge des Aufbaus.

Bemerkungen

- Der vollständige Syntaxbaum enthält die gesamte Information über die Ableitung, bis auf die (irrelevante) Reihenfolge des Aufbaus.
- Kontextfreie Grammatiken dienen zur Spezifikation von Sprachen. Ihre Implementierung ist das Parsen:
 - Die Überprüfung, ob ein Wort von einer Grammatik abgeleitet werden kann, bzw
 - Die Erzeugung des Syntaxbaums (*parse tree*).

Parsen ist die Transformation eines Worts in einen Syntaxbaum.

Definition 3.2

Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ ist ein 4-Tupel:

Definition 3.2

Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ ist ein 4-Tupel:

V ist eine endlichen Menge, die Nichtterminalzeichen (oder Variablen),

Σ ist ein Alphabet, die Terminalzeichen, disjunkt von V ,

$P \subseteq V \times (V \cup \Sigma)^*$ eine endlichen Menge, die Produktionen, und

Definition 3.2

Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ ist ein 4-Tupel:

V ist eine endlichen Menge, die Nichtterminalzeichen (oder Variablen),

Definition 3.2

Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ ist ein 4-Tupel:

V ist eine endlichen Menge, die Nichtterminalzeichen (oder Variablen),

Σ ist ein Alphabet, die Terminalzeichen, disjunkt von V ,

$P \subseteq V \times (V \cup \Sigma)^*$ eine endlichen Menge, die Produktionen, und $S \in V$ ist das Startsymbol.

Konventionen:

- A, B, C, \dots sind Nichtterminale,

Definition 3.2

Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ ist ein 4-Tupel:

- V ist eine endlichen Menge, die Nichtterminalzeichen (oder Variablen),
- Σ ist ein Alphabet, die Terminalzeichen, disjunkt von V ,
- $P \subseteq V \times (V \cup \Sigma)^*$ eine endlichen Menge, die Produktionen, und
- $S \in V$ ist das Startsymbol.

Konventionen:

- A, B, C, \dots sind Nichtterminale,
- a, b, c, \dots (und Sonderzeichen wie $+, *, \dots$) sind Terminale,
- $\alpha, \beta, \gamma, \dots \in (V \cup \Sigma)^*$
- Produktionen schreiben wir $A \rightarrow \alpha$ statt $(A, \alpha) \in P$.

~~$A \rightarrow \alpha$~~

Beispiel 3.3 (Arithmetische Ausdrücke)

$V = \{E, T, F\}$

$\Sigma = \{a, +, *, (,)\}$

$P =$

$$\left\{ \begin{array}{l} E \rightarrow T \mid E + T \\ T \rightarrow F \mid T * F \\ F \rightarrow a \mid (E) \end{array} \right\}$$

Definition 3.2

Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ ist ein 4-Tupel:

- V ist eine endlichen Menge, die Nichtterminalzeichen (oder Variablen),
- Σ ist ein Alphabet, die Terminalzeichen, disjunkt von V ,
- $P \subseteq V \times (V \cup \Sigma)^*$ eine endlichen Menge, die Produktionen, und
- $S \in V$ ist das Startsymbol.

Konventionen:

- A, B, C, \dots sind Nichtterminale,
- a, b, c, \dots (und Sonderzeichen wie $+, *, \dots$) sind Terminale,
- $\alpha, \beta, \gamma, \dots \in (V \cup \Sigma)^*$
- Produktionen schreiben wir $A \rightarrow \alpha$ statt $(A, \alpha) \in P$.
- Statt $A \rightarrow \alpha_1, A \rightarrow \alpha_2, A \rightarrow \alpha_3$ schreiben wir einfach

$$A \rightarrow (\alpha_1 \mid \alpha_2 \mid \alpha_3)$$

Beispiel 3.3 (Arithmetische Ausdrücke)

$V = \{E, T, F\}$

$\Sigma = \{a, +, *, (,)\}$

$P =$

$$\left\{ \begin{array}{l} E \rightarrow T \mid E + T \\ T \rightarrow F \mid T * F \\ F \rightarrow a \mid (E) \end{array} \right\}$$

$S = E$

Definition 3.4

Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ induziert eine **Ableitungsrelation** \rightarrow_G auf Wörtern über $V \cup \Sigma$:

$$\alpha \rightarrow_G \beta$$

Definition 3.4

Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ induziert eine **Ableitungsrelation** \rightarrow_G auf Wörtern über $V \cup \Sigma$:

$$\alpha \rightarrow_G \beta$$

gdw es eine Regel $A \rightarrow \gamma$ in P gibt, und Wörter α_1, α_2 , so dass

$$\alpha = \alpha_1 A \alpha_2$$

Definition 3.4

Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ induziert eine **Ableitungsrelation** \rightarrow_G auf Wörtern über $V \cup \Sigma$:

$$\alpha \rightarrow_G \beta$$

gdw es eine Regel $A \rightarrow \gamma$ in P gibt, und Wörter α_1, α_2 , so dass

$$\alpha = \alpha_1 A \alpha_2 \quad \text{und} \quad \beta = \alpha_1 \gamma \alpha_2$$

Definition 3.4

Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ induziert eine **Ableitungsrelation** \rightarrow_G auf Wörtern über $V \cup \Sigma$:

$$\alpha \rightarrow_G \beta$$

gdw es eine Regel $A \rightarrow \gamma$ in P gibt, und Wörter α_1, α_2 , so dass

$$\alpha = \alpha_1 A \alpha_2 \quad \text{und} \quad \beta = \alpha_1 \gamma \alpha_2$$

Beispiel:

$$a + T + a \rightarrow_G a + T * F + a$$

Definition 3.5 (Reflexive transitive Hülle)

$$\alpha \rightarrow_G^0 \alpha$$

Definition 3.5 (Reflexive transitive Hülle)

$$\alpha \rightarrow_G^0 \alpha$$

$$\alpha \rightarrow_G^{n+1} \gamma \Leftrightarrow \exists \beta. \alpha \rightarrow_G^n \beta \rightarrow_G \gamma$$

Definition 3.5 (Reflexive transitive Hülle)

$$\alpha \rightarrow_G^0 \alpha$$

$$\alpha \rightarrow_G^{n+1} \gamma \Leftrightarrow \exists \beta. \alpha \rightarrow_G^n \beta \rightarrow_G \gamma$$

$$\alpha \rightarrow_G^* \beta \Leftrightarrow \exists n. \alpha \rightarrow_G^n \beta$$

Definition 3.5 (Reflexive transitive Hülle)

$$\alpha \rightarrow_G^0 \alpha$$

$$\alpha \rightarrow_G^{n+1} \gamma \Leftrightarrow \exists \beta. \alpha \rightarrow_G^n \beta \rightarrow_G \gamma$$

$$\alpha \rightarrow_G^* \beta \Leftrightarrow \exists n. \alpha \rightarrow_G^n \beta$$

$$\alpha \rightarrow_G^+ \beta \Leftrightarrow \exists n > 0. \alpha \rightarrow_G^n \beta$$

Definition 3.5 (Reflexive transitive Hülle)

$$\begin{aligned}\alpha &\rightarrow_G^0 \alpha \\ \alpha \rightarrow_G^{n+1} \gamma &:\Leftrightarrow \exists \beta. \alpha \rightarrow_G^n \beta \rightarrow_G \gamma \\ \alpha \rightarrow_G^* \beta &:\Leftrightarrow \exists n. \alpha \rightarrow_G^n \beta \\ \alpha \rightarrow_G^+ \beta &:\Leftrightarrow \exists n > 0. \alpha \rightarrow_G^n \beta\end{aligned}$$

Beispiel: $E \rightarrow_G^{11} a * (a + a)$ und daher $E \rightarrow_G^* a * (a + a)$.

Definition 3.6 (Kontextfreie Sprache)

Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ erzeugt die Sprache

$$L(G) := \{w \in \Sigma^* \mid S \rightarrow_G^* w\}$$

Definition 3.5 (Reflexive transitive Hülle)

$$\begin{aligned}\alpha &\rightarrow_G^0 \alpha \\ \alpha \rightarrow_G^{n+1} \gamma &:\Leftrightarrow \exists \beta. \alpha \rightarrow_G^n \beta \rightarrow_G \gamma \\ \alpha \rightarrow_G^* \beta &:\Leftrightarrow \exists n. \alpha \rightarrow_G^n \beta \\ \alpha \rightarrow_G^+ \beta &:\Leftrightarrow \exists n > 0. \alpha \rightarrow_G^n \beta\end{aligned}$$

Beispiel: $E \rightarrow_G^{11} a * (a + a)$ und daher $E \rightarrow_G^* a * (a + a)$.

Wir nennen

$$\alpha_1 \rightarrow_G \alpha_2 \rightarrow_G \cdots \rightarrow_G \alpha_n$$

eine **Linksableitung** gdw in jedem Schritt das linkeste Nichtterminal in α_i ersetzt wird.

Definition 3.6 (Kontextfreie Sprache)

Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ erzeugt die Sprache

$$L(G) := \{w \in \Sigma^* \mid S \rightarrow_G^* w\}$$

Eine Sprache $L \subseteq \Sigma^*$ heißt **kontextfrei** gdw es eine kontextfreie Grammatik G gibt mit $L = L(G)$.

Definition 3.6 (Kontextfreie Sprache)

Eine kontextfreien Grammatik $G = (V, \Sigma, P, S)$ erzeugt die Sprache

$$L(G) := \{w \in \Sigma^* \mid S \rightarrow_G^* w\}$$

Eine Sprache $L \subseteq \Sigma^*$ heißt **kontextfrei** gdw es eine kontextfreie Grammatik G gibt mit $L = L(G)$.

Abkürzungen:

CFG Kontextfreie Grammatik (*context-free grammar*)

CFL Kontextfreie Sprache (*context-free language*)

Beispiel 3.7

Die nicht-reguläre Sprache $L = \{a^n b^n \mid n \in \mathbb{N}\}$ ist kontextfrei, da sie von der CFG

$$S \rightarrow aSb \mid \epsilon$$

erzeugt wird.

Definition 3.6 (Kontextfreie Sprache)

Eine kontextfreien Grammatik $G = (V, \Sigma, P, S)$ erzeugt die Sprache

$$L(G) := \{w \in \Sigma^* \mid S \rightarrow_G^* w\}$$

Eine Sprache $L \subseteq \Sigma^*$ heißt **kontextfrei** gdw es eine kontextfreie Grammatik G gibt mit $L = L(G)$.

Abkürzungen:

CFG Kontextfreie Grammatik (*context-free grammar*)

CFL Kontextfreie Sprache (*context-free language*)

Konvention:

Ist G aus dem Kontext eindeutig ersichtlich, so schreibt man auch nur $\alpha \rightarrow \beta$ statt $\alpha \rightarrow_G \beta$.

Beispiel 3.7

Die nicht-reguläre Sprache $L = \{a^n b^n \mid n \in \mathbb{N}\}$ ist kontextfrei, da sie von der CFG

$$S \rightarrow aSb \mid \epsilon$$

erzeugt wird. Genauer: $L = L(G)$ wobei $G = (V, \Sigma, P, S)$ mit

$$\begin{aligned} V &= \{S\} \\ \Sigma &= \{a, b\} \\ P &= \{S \rightarrow aSb \mid \epsilon\} \end{aligned}$$

Beispiel 3.7

Die nicht-reguläre Sprache $L = \{a^n b^n \mid n \in \mathbb{N}\}$ ist kontextfrei, da sie von der CFG

$$S \rightarrow aSb \mid \epsilon$$

erzeugt wird. Genauer: $L = L(G)$ wobei $G = (V, \Sigma, P, S)$ mit

$$\begin{aligned} V &= \{S\} \\ \Sigma &= \{a, b\} \\ P &= \{S \rightarrow aSb \mid \epsilon\} \end{aligned}$$

Der unendliche Baum aller möglichen (Links-)Ableitungen:

$$\begin{array}{ccccccc} S & \rightarrow & aSb & \rightarrow & a^2Sb^2 & \rightarrow & a^3Sb^3 & \rightarrow & \dots \\ & \searrow & & \searrow & & \searrow & & \searrow & \\ & & \epsilon & & ab & & a^2b^2 & & a^3b^3 \end{array}$$

Beispiel 3.7

Die nicht-reguläre Sprache $L = \{a^n b^n \mid n \in \mathbb{N}\}$ ist kontextfrei, da sie von der CFG

$$S \rightarrow aSb \mid \epsilon$$

erzeugt wird. Genauer: $L = L(G)$ wobei $G = (V, \Sigma, P, S)$ mit

$$\begin{aligned} V &= \{S\} \\ \Sigma &= \{a, b\} \\ P &= \{S \rightarrow aSb \mid \epsilon\} \end{aligned}$$

Der unendliche Baum aller möglichen (Links-)Ableitungen:

$$\begin{array}{ccccccc} S & \rightarrow & aSb & \rightarrow & a^2Sb^2 & \rightarrow & a^3Sb^3 & \rightarrow & \dots \\ & \searrow & & \searrow & & \searrow & & \searrow & \\ & & \epsilon & & ab & & a^2b^2 & & a^3b^3 \end{array}$$

Beispiel 3.7

Die nicht-reguläre Sprache $L = \{a^n b^n \mid n \in \mathbb{N}\}$ ist kontextfrei, da sie von der CFG

$$S \rightarrow aSb \mid \epsilon$$

erzeugt wird. Genauer: $L = L(G)$ wobei $G = (V, \Sigma, P, S)$ mit

$$\begin{aligned} V &= \{S\} \\ \Sigma &= \{a, b\} \\ P &= \{S \rightarrow aSb \mid \epsilon\} \end{aligned}$$

Der unendliche Baum aller möglichen (Links-)Ableitungen:

$$\begin{array}{ccccccc} S & \rightarrow & aSb & \rightarrow & a^2Sb^2 & \rightarrow & a^3Sb^3 & \rightarrow & \dots \\ & \searrow & & \searrow & & \searrow & & \searrow & \\ & & \epsilon & & ab & & a^2b^2 & & a^3b^3 \end{array}$$

≠ Syntaxbaum!

Beispiel 3.8

Die nicht-reguläre Sprache der Palindrome ($w = w^R$) über $\{a, b\}$

Beispiel 3.8

Die nicht-reguläre Sprache der Palindrome ($w = w^R$) über $\{a, b\}$ wird von folgender CFG erzeugt:

$$S \rightarrow \epsilon \mid a \mid b \mid aSa \mid bSb$$

Beispiel 3.8

Die nicht-reguläre Sprache der Palindrome ($w = w^R$) über $\{a, b\}$ wird von folgender CFG erzeugt:

$$S \rightarrow \epsilon \mid a \mid b \mid aSa \mid bSb$$

Der Anfang des unendlichen Baums aller möglichen (Links-)Ableitungen:

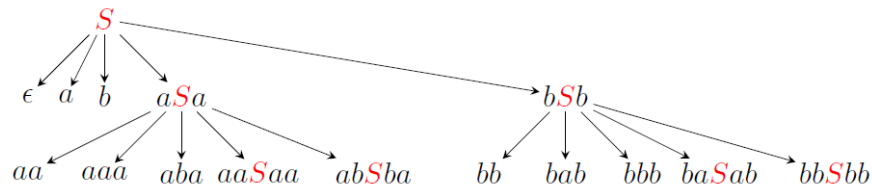


Beispiel 3.8

Die nicht-reguläre Sprache der Palindrome ($w = w^R$) über $\{a, b\}$ wird von folgender CFG erzeugt:

$$S \rightarrow \epsilon \mid a \mid b \mid aSa \mid bSb$$

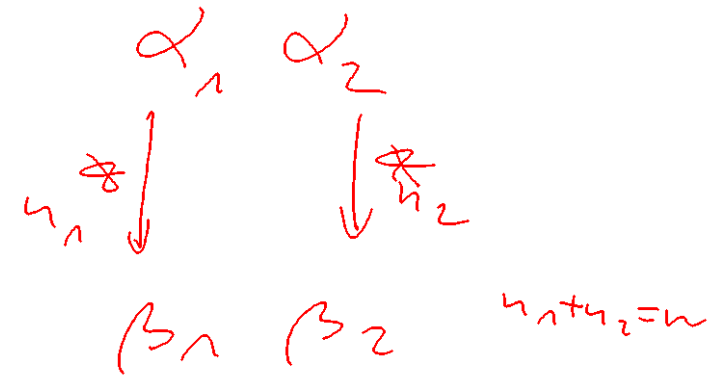
Der Anfang des unendlichen Baums aller möglichen (Links-)Ableitungen:



Lemma 3.9 (Dekompositionslemma)

$$\alpha_1 \alpha_2 \xrightarrow{G}^n \beta$$

$$\Leftrightarrow \exists \beta_1, \beta_2, n_1, n_2. \beta = \beta_1 \beta_2 \wedge n = n_1 + n_2 \wedge \alpha_i \xrightarrow{G}^{n_i} \beta_i \quad (i = 1, 2)$$



Definition 3.10

Eine CFG heißt **rechts-linear** gdw jede Produktion von der Form

$$A \rightarrow aB \quad \text{oder} \quad A \rightarrow \epsilon \quad \text{ist.}$$

Definition 3.10

Eine CFG heißt **rechts-linear** gdw jede Produktion von der Form

$$A \rightarrow aB \quad \text{oder} \quad A \rightarrow \epsilon \quad \text{ist.}$$

Eine CFG heißt **links-linear** gdw jede Produktion von der Form

$$A \rightarrow Ba \quad \text{oder} \quad A \rightarrow \epsilon \quad \text{ist.}$$

Definition 3.10

Eine CFG heißt **rechts-linear** gdw jede Produktion von der Form

$$A \rightarrow aB \quad \text{oder} \quad A \rightarrow \epsilon \quad \text{ist.}$$

Eine CFG heißt **links-linear** gdw jede Produktion von der Form

$$A \rightarrow Ba \quad \text{oder} \quad A \rightarrow \epsilon \quad \text{ist.}$$

Definition 3.10

Eine CFG heißt **rechts-linear** gdw jede Produktion von der Form

$$A \rightarrow aB \quad \text{oder} \quad A \rightarrow \epsilon \quad \text{ist.}$$

Eine CFG heißt **links-linear** gdw jede Produktion von der Form

$$A \rightarrow Ba \quad \text{oder} \quad A \rightarrow \epsilon \quad \text{ist.}$$

Lemma 3.11

Die rechts-linearen und links-linearen Grammatiken erzeugen jeweils genau die regulären Sprachen.

Lemma 3.11

Die rechts-linearen und links-linearen Grammatiken erzeugen jeweils genau die regulären Sprachen.

Beweis: Übung!

Korollar 3.12

Die regulären Sprachen sind eine echte Teilklasse der kontextfreien Sprachen.

3.2 Induktive Definitionen, Syntaxbäume und Ableitungen

Beispiel 3.13

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

3.2 Induktive Definitionen, Syntaxbäume und Ableitungen

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

Beispiel 3.13

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

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

Beispiel 3.13

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

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

$$\epsilon \in L_G(S)$$

$L(G)$

Beispiel 3.13

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

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

$$\epsilon \in L_G(S)$$

$$u \in L_G(S) \implies [u] \in L_G(S)$$

Beispiel 3.13

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

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

$$\epsilon \in L_G(S)$$

$$u \in L_G(S) \implies [u] \in L_G(S)$$

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

Beispiel 3.13

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

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:

$$\epsilon \in L_G(S) \implies [] \in L_G(S)$$

Beispiel 3.13

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

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:

$$\epsilon \in L_G(S) \implies [] \in L_G(S) \implies [[]] \in L_G(S)$$

Bemerkungen

- Die Produktionen (\rightarrow) erzeugen Wörter *top-down*, d.h. von einem Nichtterminal zu einem Wort hin.
- Die induktive Definition (\implies) erzeugt Wörter *bottom-up*, d.h. sie setzt kleinere Wörter zu größeren zusammen.

Beispiel 3.13

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

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:

$$\begin{aligned} \epsilon \in L_G(S) &\implies [] \in L_G(S) \implies [[]] \in L_G(S) \\ &\implies [[]] [] \in L_G(S) \end{aligned}$$

Bemerkungen

- Die Produktionen (\rightarrow) erzeugen Wörter *top-down*, d.h. von einem Nichtterminal zu einem Wort hin.
- Die induktive Definition (\implies) erzeugt Wörter *bottom-up*, d.h. sie setzt kleinere Wörter zu größeren zusammen.
- Die induktive Definition betrachtet nur Wörter aus Σ^* .

Beispiel 3.13

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

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:

$$\epsilon \in L_G(S) \implies [] \in L_G(S) \implies [[]] \in L_G(S)$$

Zur induktiven Definition von $L_G(S)$ gehört ein Induktionsprinzip:

Bemerkungen

- Die Produktionen (\rightarrow) erzeugen Wörter *top-down*, d.h. von einem Nichtterminal zu einem Wort hin.
- Die induktive Definition (\implies) erzeugt Wörter *bottom-up*, d.h. sie setzt kleinere Wörter zu größeren zusammen.
- Die induktive Definition betrachtet nur Wörter aus Σ^* .

Zur induktiven Definition von $L_G(S)$ gehört ein Induktionsprinzip:

Um zu zeigen, dass für alle $u \in L_G(S)$ eine Eigenschaft $P(u)$ gilt,



$$x \in L_n(S)$$

Zur induktiven Definition von $L_G(S)$ gehört ein Induktionsprinzip:

Um zu zeigen, dass für alle $u \in L_G(S)$ eine Eigenschaft $P(u)$ gilt, dh $\forall u \in L_G(S). P(u)$, zeige:

$$\begin{array}{rcl} & & P(\epsilon) \\ P(u) & \implies & P([u]) \\ P(u) \wedge P(v) & \implies & P(uv) \end{array}$$

„Induktion über die Erzeugung von u “

Zur induktiven Definition von $L_G(S)$ gehört ein Induktionsprinzip:

Um zu zeigen, dass für alle $u \in L_G(S)$ eine Eigenschaft $P(u)$ gilt, dh $\forall u \in L_G(S). P(u)$, zeige:

$$\begin{array}{rcl} & & P(\epsilon) \\ P(u) & \implies & P([u]) \\ P(u) \wedge P(v) & \implies & P(uv) \end{array}$$

„Induktion über die Erzeugung von u “

Zur induktiven Definition von $L_G(S)$ gehört ein Induktionsprinzip:

Um zu zeigen, dass für alle $u \in L_G(S)$ eine Eigenschaft $P(u)$ gilt, dh $\forall u \in L_G(S). P(u)$, zeige:

$$\begin{array}{rcl} & & P(\epsilon) \\ P(u) & \implies & P([u]) \\ P(u) \wedge P(v) & \implies & P(uv) \end{array}$$

„Induktion über die Erzeugung von u “

Lemma 3.14

Alle $u \in L_G(S)$ enthalten gleich viele [wie].

Zur induktiven Definition von $L_G(S)$ gehört ein Induktionsprinzip:

Um zu zeigen, dass für alle $u \in L_G(S)$ eine Eigenschaft $P(u)$ gilt, dh $\forall u \in L_G(S). P(u)$, zeige:

$$\begin{array}{rcl} & & P(\epsilon) \\ P(u) & \implies & P([u]) \\ P(u) \wedge P(v) & \implies & P(uv) \end{array}$$

„Induktion über die Erzeugung von u “

Lemma 3.14

Alle $u \in L_G(S)$ enthalten gleich viele [wie].

Beweis:

Mit Induktion über die Erzeugung von u :

Zur induktiven Definition von $L_G(S)$ gehört ein Induktionsprinzip:

Um zu zeigen, dass für alle $u \in L_G(S)$ eine Eigenschaft $P(u)$ gilt, dh $\forall u \in L_G(S). P(u)$, zeige:

$$\begin{array}{l} P(\epsilon) \\ P(u) \implies P([u]) \\ P(u) \wedge P(v) \implies P(uv) \end{array}$$

„Induktion über die Erzeugung von u “

Lemma 3.14

Alle $u \in L_G(S)$ enthalten gleich viele [wie].

Beweis:

Mit Induktion über die Erzeugung von u :

- ϵ enthält 0 [und].

Zur induktiven Definition von $L_G(S)$ gehört ein Induktionsprinzip:

Um zu zeigen, dass für alle $u \in L_G(S)$ eine Eigenschaft $P(u)$ gilt, dh $\forall u \in L_G(S). P(u)$, zeige:

$$\begin{array}{l} P(\epsilon) \\ P(u) \implies P([u]) \\ P(u) \wedge P(v) \implies P(uv) \end{array}$$

„Induktion über die Erzeugung von u “

Lemma 3.14

Alle $u \in L_G(S)$ enthalten gleich viele [wie].

Beweis:

Mit Induktion über die Erzeugung von u :

- ϵ enthält 0 [und].
- Enthält u gleich viele [wie], so auch $[u]$.
- Enthalten u und v gleich viele [wie], so auch uv . □

Zur induktiven Definition von $L_G(S)$ gehört ein Induktionsprinzip:

Um zu zeigen, dass für alle $u \in L_G(S)$ eine Eigenschaft $P(u)$ gilt, dh $\forall u \in L_G(S). P(u)$, zeige:

$$\begin{array}{l} P(\epsilon) \\ P(u) \implies P([u]) \\ P(u) \wedge P(v) \implies P(uv) \end{array}$$

„Induktion über die Erzeugung von u “

Lemma 3.14

Alle $u \in L_G(S)$ enthalten gleich viele [wie].

Beweis:

Mit Induktion über die Erzeugung von u :

- ϵ enthält 0 [und].
- Enthält u gleich viele [wie], so auch $[u]$.

Hinweis

Die Aussagen

$$\forall x \in M. P(x)$$

und

$$\forall x. x \in M \implies P(x)$$

sind logisch völlig äquivalent.

Hinweis

Die Aussagen

$$\forall x \in M. P(x)$$

und

$$\forall x. x \in M \implies P(x)$$

sind logisch völlig äquivalent.

Der Allquantor wird dann (wie üblich) oft weggelassen:

$$x \in M \implies P(x)$$

Definition 3.15

Präfix:

$$u \preceq w \quad :\Leftrightarrow \quad \exists v. uv = w$$

⌋⌊

Hinweis

Die Aussagen

$$\forall x \in M. P(x)$$

und

$$\forall x. x \in M \implies P(x)$$

sind logisch völlig äquivalent.

Der Allquantor wird dann (wie üblich) oft weggelassen:

$$x \in M \implies P(x)$$

Definition 3.15

Präfix:

$$u \preceq w \quad :\Leftrightarrow \quad \exists v. uv = w$$

Definition 3.15

Präfix:

$$u \preceq w \quad :\Leftrightarrow \quad \exists v. uv = w$$

Anzahl der Vorkommen:

$$\#_a(w) := \text{Anzahl der Vorkommen von } a \text{ in } w$$

Definition 3.15

Präfix:

$$u \preceq w \quad :\Leftrightarrow \quad \exists v. uv = w$$

Anzahl der Vorkommen:

$$\#_a(w) := \text{Anzahl der Vorkommen von } a \text{ in } w$$

Für unser Beispiel führen wir zwei Abk. ein:

$$A(w) := \#_{[}(w) \quad B(w) := \#_{]}(w)$$

Definition 3.15

Präfix:

$$u \preceq w \quad :\Leftrightarrow \quad \exists v. uv = w$$

Anzahl der Vorkommen:

$$\#_a(w) := \text{Anzahl der Vorkommen von } a \text{ in } w$$

Für unser Beispiel führen wir zwei Abk. ein:

$$A(w) := \#_{[}(w) \quad B(w) := \#_{]}(w)$$

Wir nennen $w \in \{[,]\}^*$ **balanciert** gdw

Definition 3.15

Präfix:

$$u \preceq w \quad :\Leftrightarrow \quad \exists v. uv = w$$

Anzahl der Vorkommen:

$$\#_a(w) := \text{Anzahl der Vorkommen von } a \text{ in } w$$

Für unser Beispiel führen wir zwei Abk. ein:

$$A(w) := \#_{[}(w) \quad B(w) := \#_{]}(w)$$

Wir nennen $w \in \{[,]\}^*$ **balanciert** gdw

- (1) $A(w) = B(w)$ und
- (2) für alle Präfixe u von w gilt $A(u) \geq B(u)$

Definition 3.15

Präfix:

$$u \preceq w \iff \exists v. uv = w$$

Anzahl der Vorkommen:

$$\#_a(w) := \text{Anzahl der Vorkommen von } a \text{ in } w$$

Für unser Beispiel führen wir zwei Abk. ein:

$$A(w) := \#_[(w) \quad B(w) := \#_](w)$$

Wir nennen $w \in \{[,]\}^*$ **balanciert** gdw

- (1) $A(w) = B(w)$ und
- (2) für alle Präfixe u von w gilt $A(u) \geq B(u)$

- Balanciert: [], [[]], [] [], [[[]] []] []

Satz 3.16

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

Definition 3.15

Präfix:

$$u \preceq w \iff \exists v. uv = w$$

Anzahl der Vorkommen:

$$\#_a(w) := \text{Anzahl der Vorkommen von } a \text{ in } w$$

Für unser Beispiel führen wir zwei Abk. ein:

$$A(w) := \#_[(w) \quad B(w) := \#_](w)$$

Wir nennen $w \in \{[,]\}^*$ **balanciert** gdw

- (1) $A(w) = B(w)$ und
- (2) für alle Präfixe u von w gilt $A(u) \geq B(u)$

- Balanciert: [], [[]], [] [], [[[]] []] []
- Nicht balanciert:] [, []] []

Satz 3.16

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:

Satz 3.16

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).

ϵ : $p \preceq \epsilon$

Satz 3.16

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).

ϵ : $p \preceq \epsilon \implies p = \epsilon \implies A(p) = B(p)$

$[u]$: Sei $p \preceq [u]$

Fall $p = \epsilon$: $A(p) = B(p)$

Fall $p = [u]$: $A(p) = A(u) + 1 =$

Satz 3.16

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).

ϵ : $p \preceq \epsilon \implies p = \epsilon \implies A(p) = B(p)$

$[u]$: Sei $p \preceq [u]$

Satz 3.16

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).

ϵ : $p \preceq \epsilon \implies p = \epsilon \implies A(p) = B(p)$

$[u]$: Sei $p \preceq [u]$

Fall $p = \epsilon$: $A(p) = B(p)$

Fall $p = [u]$: $A(p) = A(u) + 1 = B(u) + 1 =$

Satz 3.16

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).

$\epsilon: p \preceq \epsilon \implies p = \epsilon \implies A(p) = B(p)$

[u]: Sei $p \preceq [u]$

Fall $p = \epsilon: A(p) = B(p)$

Fall $p = [u]: A(p) = A(u) + 1 = B(u) + 1 = B(p)$

Satz 3.16

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).

$\epsilon: p \preceq \epsilon \implies p = \epsilon \implies A(p) = B(p)$

[u]: Sei $p \preceq [u]$

Fall $p = \epsilon: A(p) = B(p)$

Fall $p = [u]: A(p) = A(u) + 1 = B(u) + 1 = B(p)$

Fall $p = [q$ mit $q \preceq u$:

$A(p) = A(q) + 1$

Satz 3.16

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).

$\epsilon: p \preceq \epsilon \implies p = \epsilon \implies A(p) = B(p)$

[u]: Sei $p \preceq [u]$

Fall $p = \epsilon: A(p) = B(p)$

Fall $p = [u]: A(p) = A(u) + 1 = B(u) + 1 = B(p)$

Fall $p = [q$ mit $q \preceq u$:

Satz 3.16

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).

$\epsilon: p \preceq \epsilon \implies p = \epsilon \implies A(p) = B(p)$

[u]: Sei $p \preceq [u]$

Fall $p = \epsilon: A(p) = B(p)$

Fall $p = [u]: A(p) = A(u) + 1 = B(u) + 1 = B(p)$

Fall $p = [q$ mit $q \preceq u$:

$A(p) = A(q) + 1 \geq B(q) + 1$

Satz 3.16

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).

$$\epsilon: p \preceq \epsilon \implies p = \epsilon \implies A(p) = B(p)$$

[u]: Sei $p \preceq [u]$

$$\text{Fall } p = \epsilon: A(p) = B(p)$$

$$\text{Fall } p = [u]: A(p) = A(u) + 1 = B(u) + 1 = B(p)$$

Fall $p = [q]$ mit $q \preceq u$:

$$A(p) = A(q) + 1 \geq B(q) + 1 > B(q)$$

Satz 3.16

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).

$$\epsilon: p \preceq \epsilon \implies p = \epsilon \implies A(p) = B(p)$$

[u]: Sei $p \preceq [u]$

$$\text{Fall } p = \epsilon: A(p) = B(p)$$

$$\text{Fall } p = [u]: A(p) = A(u) + 1 = B(u) + 1 = B(p)$$

Fall $p = [q]$ mit $q \preceq u$:

$$A(p) = A(q) + 1 \geq B(q) + 1 > B(q) = B(p)$$

uv: Sei $p \preceq uv$.

Fall $p \preceq u$: $A(p) \geq B(p)$ mit IA für u

Fall $p = uq$ und $q \preceq v$:

$$\begin{aligned} A(p) &= A(u) + A(q) = B(u) + A(q) \geq B(u) + B(q) = B(uq) = \\ B(p) & \end{aligned}$$

Satz 3.16

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).

$$\epsilon: p \preceq \epsilon \implies p = \epsilon \implies A(p) = B(p)$$

[u]: Sei $p \preceq [u]$

$$\text{Fall } p = \epsilon: A(p) = B(p)$$

$$\text{Fall } p = [u]: A(p) = A(u) + 1 = B(u) + 1 = B(p)$$

Fall $p = [q]$ mit $q \preceq u$:

$$A(p) = A(q) + 1 \geq B(q) + 1 > B(q) = B(p)$$

Satz 3.16

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).

$$\epsilon: p \preceq \epsilon \implies p = \epsilon \implies A(p) = B(p)$$

[u]: Sei $p \preceq [u]$

$$\text{Fall } p = \epsilon: A(p) = B(p)$$

$$\text{Fall } p = [u]: A(p) = A(u) + 1 = B(u) + 1 = B(p)$$

Fall $p = [q]$ mit $q \preceq u$:

$$A(p) = A(q) + 1 \geq B(q) + 1 > B(q) = B(p)$$

uv: Sei $p \preceq uv$.

Fall $p \preceq u$: $A(p) \geq B(p)$ mit IA für u

Fall $p = uq$ und $q \preceq v$:

$$\begin{aligned} A(p) &= A(u) + A(q) = B(u) + A(q) \geq B(u) + B(q) = B(uq) = \\ B(p) & \end{aligned}$$

Beweis (Forts.)

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

Beweis (Forts.)

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

Mit vollständiger Induktion über $n := |u|$. (dh $u = a_1 \dots a_n$)

IA: Jedes balancierte Wort $< n$ liegt in $L_G(S)$.

Beweis (Forts.)

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

Mit vollständiger Induktion über $n := |u|$.

Beweis (Forts.)

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

Mit vollständiger Induktion über $n := |u|$. (dh $u = a_1 \dots a_n$)

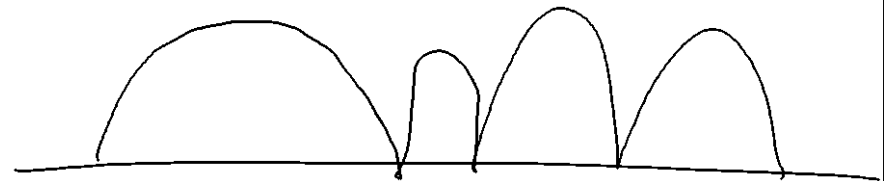
IA: Jedes balancierte Wort $< n$ liegt in $L_G(S)$.

Zz: $u \in L_G(S)$.

Falls $n = 0$, so $u = \epsilon \in L_G(S)$.

Sei $n > 0$.

Betrachte Werteverlauf von $h(w) := A(w) - B(w)$:



Beweis (Forts.)

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

Mit vollständiger Induktion über $n := |u|$. (dh $u = a_1 \dots a_n$)

IA: Jedes balancierte Wort $< n$ liegt in $L_G(S)$.

Zz: $u \in L_G(S)$.

Falls $n = 0$, so $u = \epsilon \in L_G(S)$.

Sei $n > 0$.

Betrachte Werteverlauf von $h(w) := A(w) - B(w)$:

$h(\epsilon)$

Beweis (Forts.)

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

Mit vollständiger Induktion über $n := |u|$. (dh $u = a_1 \dots a_n$)

IA: Jedes balancierte Wort $< n$ liegt in $L_G(S)$.

Zz: $u \in L_G(S)$.

Falls $n = 0$, so $u = \epsilon \in L_G(S)$.

Sei $n > 0$.

Betrachte Werteverlauf von $h(w) := A(w) - B(w)$:

$h(\epsilon), h(a_1), h(a_1 a_2), \dots, h(a_1 \dots a_n)$ (alle $\geq 0!$).

Insb. gilt $a_1 = [$ und $a_n =]$.

- Es gibt nur die Nullstellen $h(\epsilon)$ und $h(a_1 \dots a_n)$.
Dann ist $v := a_2 \dots a_{n-1}$ balanciert:

Beweis (Forts.)

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

Mit vollständiger Induktion über $n := |u|$. (dh $u = a_1 \dots a_n$)

IA: Jedes balancierte Wort $< n$ liegt in $L_G(S)$.

Zz: $u \in L_G(S)$.

Falls $n = 0$, so $u = \epsilon \in L_G(S)$.

Sei $n > 0$.

Betrachte Werteverlauf von $h(w) := A(w) - B(w)$:

$h(\epsilon), h(a_1), h(a_1 a_2), \dots, h(a_1 \dots a_n)$ (alle $\geq 0!$).

Insb. gilt $a_1 = [$ und $a_n =]$.

- Es gibt nur die Nullstellen $h(\epsilon)$ und $h(a_1 \dots a_n)$.

Beweis (Forts.)

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

Mit vollständiger Induktion über $n := |u|$. (dh $u = a_1 \dots a_n$)

IA: Jedes balancierte Wort $< n$ liegt in $L_G(S)$.

Zz: $u \in L_G(S)$.

Falls $n = 0$, so $u = \epsilon \in L_G(S)$.

Sei $n > 0$.

Betrachte Werteverlauf von $h(w) := A(w) - B(w)$:

$h(\epsilon), h(a_1), h(a_1 a_2), \dots, h(a_1 \dots a_n)$ (alle $\geq 0!$).

Insb. gilt $a_1 = [$ und $a_n =]$.

- Es gibt nur die Nullstellen $h(\epsilon)$ und $h(a_1 \dots a_n)$.
Dann ist $v := a_2 \dots a_{n-1}$ balanciert:
(1) $A(v) = A([v]) - 1 = B([v]) - 1$

Beweis (Forts.)

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

Mit vollständiger Induktion über $n := |u|$. (dh $u = a_1 \dots a_n$)

IA: Jedes balancierte Wort $< n$ liegt in $L_G(S)$.

Zz: $u \in L_G(S)$.

Falls $n = 0$, so $u = \epsilon \in L_G(S)$.

Sei $n > 0$.

Betrachte Werteverlauf von $h(w) := A(w) - B(w)$:

$h(\epsilon), h(a_1), h(a_1a_2), \dots, h(a_1 \dots a_n)$ (alle $\geq 0!$).

Insb. gilt $a_1 = [$ und $a_n =]$.

- Es gibt nur die Nullstellen $h(\epsilon)$ und $h(a_1 \dots a_n)$.
Dann ist $v := a_2 \dots a_{n-1}$ balanciert:
(1) $A(v) = A([v]) - 1 = B([v]) - 1 = B(v)$
(2) $p \preceq v$: $h([p]) = A([p]) - B([p]) > 0$

Beweis (Forts.)

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

Mit vollständiger Induktion über $n := |u|$. (dh $u = a_1 \dots a_n$)

IA: Jedes balancierte Wort $< n$ liegt in $L_G(S)$.

Zz: $u \in L_G(S)$.

Falls $n = 0$, so $u = \epsilon \in L_G(S)$.

Sei $n > 0$.

Betrachte Werteverlauf von $h(w) := A(w) - B(w)$:

$h(\epsilon), h(a_1), h(a_1a_2), \dots, h(a_1 \dots a_n)$ (alle $\geq 0!$).

Insb. gilt $a_1 = [$ und $a_n =]$.

- Es gibt nur die Nullstellen $h(\epsilon)$ und $h(a_1 \dots a_n)$.
Dann ist $v := a_2 \dots a_{n-1}$ balanciert:
(1) $A(v) = A([v]) - 1 = B([v]) - 1 = B(v)$
(2) $p \preceq v$: $h([p]) = A([p]) - B([p]) > 0 \implies$
 $A(p) = A([p]) - 1 > B([p]) - 1$

Beweis (Forts.)

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

Mit vollständiger Induktion über $n := |u|$. (dh $u = a_1 \dots a_n$)

IA: Jedes balancierte Wort $< n$ liegt in $L_G(S)$.

Zz: $u \in L_G(S)$.

Falls $n = 0$, so $u = \epsilon \in L_G(S)$.

Sei $n > 0$.

Betrachte Werteverlauf von $h(w) := A(w) - B(w)$:

$h(\epsilon), h(a_1), h(a_1a_2), \dots, h(a_1 \dots a_n)$ (alle $\geq 0!$).

Insb. gilt $a_1 = [$ und $a_n =]$.

- Es gibt nur die Nullstellen $h(\epsilon)$ und $h(a_1 \dots a_n)$.
Dann ist $v := a_2 \dots a_{n-1}$ balanciert:
(1) $A(v) = A([v]) - 1 = B([v]) - 1 = B(v)$
(2) $p \preceq v$: $h([p]) = A([p]) - B([p]) > 0 \implies$
 $A(p) = A([p]) - 1$

Beweis (Forts.)

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

Mit vollständiger Induktion über $n := |u|$. (dh $u = a_1 \dots a_n$)

IA: Jedes balancierte Wort $< n$ liegt in $L_G(S)$.

Zz: $u \in L_G(S)$.

Falls $n = 0$, so $u = \epsilon \in L_G(S)$.

Sei $n > 0$.

Betrachte Werteverlauf von $h(w) := A(w) - B(w)$:

$h(\epsilon), h(a_1), h(a_1a_2), \dots, h(a_1 \dots a_n)$ (alle $\geq 0!$).

Insb. gilt $a_1 = [$ und $a_n =]$.

- Es gibt nur die Nullstellen $h(\epsilon)$ und $h(a_1 \dots a_n)$.
Dann ist $v := a_2 \dots a_{n-1}$ balanciert:
(1) $A(v) = A([v]) - 1 = B([v]) - 1 = B(v)$
(2) $p \preceq v$: $h([p]) = A([p]) - B([p]) > 0 \implies$
 $A(p) = A([p]) - 1 > B([p]) - 1 = B(p) - 1$

Beweis (Forts.)

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

Mit vollständiger Induktion über $n := |u|$. (dh $u = a_1 \dots a_n$)

IA: Jedes balancierte Wort $< n$ liegt in $L_G(S)$.

Zz: $u \in L_G(S)$.

Falls $n = 0$, so $u = \epsilon \in L_G(S)$.

Sei $n > 0$.

Betrachte Werteverlauf von $h(w) := A(w) - B(w)$:

$h(\epsilon), h(a_1), h(a_1 a_2), \dots, h(a_1 \dots a_n)$ (alle $\geq 0!$).

Insb. gilt $a_1 = [$ und $a_n =]$.

- Es gibt nur die Nullstellen $h(\epsilon)$ und $h(a_1 \dots a_n)$.

Dann ist $v := a_2 \dots a_{n-1}$ balanciert:

$$(1) A(v) = A([v]) - 1 = B([v]) - 1 = B(v)$$

$$(2) p \preceq v: h([p]) = A([p]) - B([p]) > 0 \implies$$

$$A(p) = A([p]) - 1 > B([p]) - 1 = B(p) - 1 \implies$$

$$A(p) \geq B(p)$$

Beweis (Forts.)

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

Mit vollständiger Induktion über $n := |u|$. (dh $u = a_1 \dots a_n$)

IA: Jedes balancierte Wort $< n$ liegt in $L_G(S)$.

Zz: $u \in L_G(S)$.

Falls $n = 0$, so $u = \epsilon \in L_G(S)$.

Sei $n > 0$.

Betrachte Werteverlauf von $h(w) := A(w) - B(w)$:

$h(\epsilon), h(a_1), h(a_1 a_2), \dots, h(a_1 \dots a_n)$ (alle $\geq 0!$).

Insb. gilt $a_1 = [$ und $a_n =]$.

- Es gibt nur die Nullstellen $h(\epsilon)$ und $h(a_1 \dots a_n)$.

Dann ist $v := a_2 \dots a_{n-1}$ balanciert:

$$(1) A(v) = A([v]) - 1 = B([v]) - 1 = B(v)$$

$$(2) p \preceq v: h([p]) = A([p]) - B([p]) > 0 \implies$$

$$A(p) = A([p]) - 1 > B([p]) - 1 = B(p) - 1 \implies$$

$$A(p) \geq B(p)$$

$$\implies v \in L_G(S) \text{ (nach IA)} \implies u = [v] \in L_G(S)$$

Beweis (Forts.)

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

Mit vollständiger Induktion über $n := |u|$. (dh $u = a_1 \dots a_n$)

IA: Jedes balancierte Wort $< n$ liegt in $L_G(S)$.

Zz: $u \in L_G(S)$.

Falls $n = 0$, so $u = \epsilon \in L_G(S)$.

Sei $n > 0$.

Betrachte Werteverlauf von $h(w) := A(w) - B(w)$:

$h(\epsilon), h(a_1), h(a_1 a_2), \dots, h(a_1 \dots a_n)$ (alle $\geq 0!$).

Insb. gilt $a_1 = [$ und $a_n =]$.

- Es gibt nur die Nullstellen $h(\epsilon)$ und $h(a_1 \dots a_n)$.

Dann ist $v := a_2 \dots a_{n-1}$ balanciert:

$$(1) A(v) = A([v]) - 1 = B([v]) - 1 = B(v)$$

$$(2) p \preceq v: h([p]) = A([p]) - B([p]) > 0 \implies$$

$$A(p) = A([p]) - 1 > B([p]) - 1 = B(p) - 1 \implies$$

$$A(p) \geq B(p)$$

$$\implies v \in L_G(S) \text{ (nach IA)}$$