Script generated by TTT

Title: Nipkow: Theo (16.05.2019)

Date: Thu May 16 14:16:58 CEST 2019

Duration: 87:03 min

Pages: 138

Im Folgenden sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA ohne unerreichbare Zustände.

Definition 3.50 (Äquivalenz von Zuständen)

$$p \equiv_M q \;\Leftrightarrow\; (\forall w \in \Sigma^*.\ \hat{\delta}(p,w) \in F \Leftrightarrow \hat{\delta}(q,w) \in F)$$

Intuition: Zwei Zustände sind äquivalent gwd sie die selbe Sprache erkennen.

Wir schreiben $\equiv$ statt $\equiv_M$ wenn M klar ist.

Einfache Fakten:

- $\equiv_M$ ist eine Äquivalenzrelation.
- $p \equiv_M q \implies \delta(p,a) \equiv_M \delta(q,a)$
- Algorithmus U liefert die unterscheidbaren Zustände, also $\not\equiv$.

In der weiteren Analyse beziehen wir uns direkt auf $\equiv$, nicht mehr auf den Algorithmus.

Die „Kollabierung“ von M bzgl. $\equiv$ ist der *Quotientenautomat*:

Definition 3.51 (Quotientenautomat)

$$M/\!\equiv \;:=$$

Die „Kollabierung“ von M bzgl. $\equiv$ ist der *Quotientenautomat*:

Definition 3.51 (Quotientenautomat)

$$\begin{aligned} M/\!\equiv \;&:=\; (Q/\!\equiv,\ \Sigma, \delta',\ [q_0]_\equiv,\ F/\!\equiv) \\ \delta'([p]_\equiv, a) \;&:=\; [\delta(p,a)]_\equiv \end{aligned}$$

Die Definition von δ' ist wohlgeformt da unabhängig von der Wahl des Repräsentanten p:

$$[p]_\equiv = [p']_\equiv$$

Die „Kollabierung“ von M bzgl. $\equiv$ ist der *Quotientenautomat*:

Definition 3.51 (Quotientenautomat)

$$
\begin{aligned}
M/{\equiv} &:= (Q/{\equiv},\ \Sigma, \delta',\ [q_0]_\equiv,\ F/{\equiv}) \\
\delta'([p]_\equiv, a) &:= [\delta(p,a)]_\equiv
\end{aligned}
$$

Die Definition von δ' ist wohlgeformt da unabhängig von der Wahl des Repräsentanten p:

$$
\boxed{
\begin{aligned}
\cancel{[p]_\equiv = [p']_\equiv} \quad &\Longrightarrow \quad p \equiv p' \quad \Longrightarrow \quad \delta(p,a) \equiv \delta(p',a) \\
&\Longrightarrow \quad [\delta(p,a)]_\equiv = [\delta(p',a)]_\equiv
\end{aligned}
}
$$

Lemma 3.52

$$L(M/{\equiv}) = L(M)$$

Beweis: Übung.

Minimalität des Quotientenautomaten

Eine Beobachtung: Für $p := \hat{\delta}(q_0, u)$ und $q := \hat{\delta}(q_0, v)$ gilt

$$
\begin{aligned}
p \equiv_M q \quad &\Leftrightarrow \quad \forall w \in \Sigma^*.\ \hat{\delta}(p,w) \in F \Leftrightarrow \hat{\delta}(q,w) \in F \\
&\Leftrightarrow \quad \forall w \in \Sigma^*.\ \hat{\delta}(q_0,uw) \in F \Leftrightarrow \hat{\delta}(q_0,vw) \in F \\
&\Leftrightarrow \quad \forall w \in \Sigma^*.\ uw \in L(M) \Leftrightarrow vw \in L(M)
\end{aligned}
$$

Definition 3.53

Jede Sprache $L \subseteq \Sigma^*$ induziert eine Äquivalenzrelation $\equiv_L \subseteq \Sigma^* \times \Sigma^*$:

$$u \equiv_L v \quad \Leftrightarrow \quad \forall w \in \Sigma^*.\ uw \in L \Leftrightarrow vw \in L$$

Obige Beobachtung lässt sich nun schreiben als

$$\hat{\delta}(q_0,u) \equiv_M \hat{\delta}(q_0,v) \quad \Leftrightarrow \quad u \equiv_{L(M)} v$$

Achtung

~~$p \equiv_M q$ ist eine~~ Relation auf Zuständen von M

$u \equiv_L v$ ist eine Relation auf Wörtern

$$u \equiv_{L(M)} v \quad \Leftrightarrow \quad \hat{\delta}(q_0,u) \equiv_M \hat{\delta}(q_0,v)$$

Da alle Zustände von q_0 erreichbar sind, gilt sogar: Die Abbildung

$$[u]_{\equiv_{L(M)}} \quad \mapsto \quad [\hat{\delta}(q_0,u)]_{\equiv_M}$$

ist eine Bijektion zwischen den $\equiv_{L(M)}$ und $\equiv_M$ Äquivalenzklassen.

Satz 3.54

Ist M ein DFA ohne unerreichbare Zustände, so ist der von Algorithmus U *berechnete Quotientenautomat $M/{\equiv}$ ein minimaler DFA für $L(M)$.*

$$u \equiv_{L(M)} v \quad \Leftrightarrow \quad \hat{\delta}(q_0, u) \equiv_M \hat{\delta}(q_0, v)$$

Da alle Zustände von q_0 erreichbar sind, gilt sogar: Die Abbildung

$$[u]_{\equiv_{L(M)}} \quad \mapsto \quad [\hat{\delta}(q_0, u)]_{\equiv_M}$$

ist eine Bijektion zwischen den $\equiv_{L(M)}$ und $\equiv_M$ Äquivalenzklassen.

Satz 3.54

Ist M ein DFA ohne unerreichbare Zustände, so ist der von Algorithmus U *berechnete Quotientenautomat $M/\equiv$ ein minimaler DFA für $L(M)$.*

Beweis:

Sei $L := L(M)$ und M' ein DFA mit $L(M') = L$.

Es gilt sogar:

Fakt 3.55

Alle Quotientenautomaten $M/\equiv_M$ für die gleiche Sprache $L(M)$ haben die gleiche Struktur, d.h. sie unterscheiden sich nur durch eine Umbenennung der Zustände.

Daher beschriften wir die Zustände des kanonischen Minimalautomaten für eine Sprache L mit $\equiv_L$ Äquivalenzklassen.

Definition 3.57 (Kanonischer Minimalautomat)

$$M_L \ :=$$

Definition 3.57 (Kanonischer Minimalautomat)

$$M_L \ := \ (\Sigma^*/{\equiv_L},\ \Sigma,\ \delta_L,\ [\epsilon]_{\equiv_L},\ F_L)$$

mit $\delta_L([w]_{\equiv_L}, a) := [wa]_{\equiv_L}$

Definition 3.57 (Kanonischer Minimalautomat)

$$M_L := (\Sigma^*/{\equiv_L},\ \Sigma,\ \delta_L,\ [\epsilon]_{\equiv_L},\ F_L)$$

mit $\delta_L([w]_{\equiv_L}, a) := [wa]_{\equiv_L}$ und $F_L := \{[w]_{\equiv_L} \mid w \in L\}$.

Definition 3.57 (Kanonischer Minimalautomat)

$$M_L := (\Sigma^*/{\equiv_L},\ \Sigma,\ \delta_L,\ [\epsilon]_{\equiv_L},\ F_L)$$

mit $\delta_L([w]_{\equiv_L}, a) := [wa]_{\equiv_L}$ und $F_L := \{[w]_{\equiv_L} \mid w \in L\}$.

Man sieht: δ_L ist wohldefiniert und $\hat{\delta}_L([\epsilon]_{\equiv_L}, w) = [w]_{\equiv_L}$.
Dann gilt offensichtlich $L(M_L) = L$.

Definition 3.57 (Kanonischer Minimalautomat)

$$M_L := (\Sigma^*/{\equiv_L},\ \Sigma,\ \delta_L,\ [\epsilon]_{\equiv_L},\ F_L)$$

mit $\delta_L([w]_{\equiv_L}, a) := [wa]_{\equiv_L}$ und $F_L := \{[w]_{\equiv_L} \mid w \in L\}$.

Man sieht: δ_L ist wohldefiniert und $\hat{\delta}_L([\epsilon]_{\equiv_L}, w) = [w]_{\equiv_L}$.
Dann gilt offensichtlich $L(M_L) = L$.

Satz 3.58 (Myhill-Nerode)

Eine Sprache $L \subseteq \Sigma^*$ ist genau dann regulär, wenn $\equiv_L$ endlich viele Äquivalenzklassen hat.

Beweis:

„$\Longrightarrow$“: Ist L regulär, so wird L von einem DFA M akzeptiert.

Definition 3.57 (Kanonischer Minimalautomat)

$$M_L := (\Sigma^*/{\equiv_L},\ \Sigma,\ \delta_L,\ [\epsilon]_{\equiv_L},\ F_L)$$

mit $\delta_L([w]_{\equiv_L}, a) := [wa]_{\equiv_L}$ und $F_L := \{[w]_{\equiv_L} \mid w \in L\}$.

Man sieht: δ_L ist wohldefiniert und $\hat{\delta}_L([\epsilon]_{\equiv_L}, w) = [w]_{\equiv_L}$.
Dann gilt offensichtlich $L(M_L) = L$.

Satz 3.58 (Myhill-Nerode)

Eine Sprache $L \subseteq \Sigma^*$ ist genau dann regulär, wenn $\equiv_L$ endlich viele Äquivalenzklassen hat.

Beweis:

„$\Longrightarrow$“: Ist L regulär, so wird L von einem DFA M akzeptiert.
Daher hat $\equiv_L$ so viele Äquivalenzklassen wie $M/{\equiv_M}$ Zustände.

Definition 3.57 (Kanonischer Minimalautomat)

$$M_L \ := \ (\Sigma^*/{\equiv_L},\ \Sigma,\ \delta_L,\ [\epsilon]_{\equiv_L},\ F_L)$$

mit $\delta_L([w]_{\equiv_L}, a) := [wa]_{\equiv_L}$ und $F_L := \{[w]_{\equiv_L} \mid w \in L\}$.

Man sieht: δ_L ist wohldefiniert und $\hat{\delta}_L([\epsilon]_{\equiv_L}, w) = [w]_{\equiv_L}$.
Dann gilt offensichtlich $L(M_L) = L$.

Satz 3.58 (Myhill-Nerode)

Eine Sprache $L \subseteq \Sigma^*$ ist genau dann regulär, wenn $\equiv_L$ endlich viele Äquivalenzklassen hat.

Beweis:
„$\Longrightarrow$“: Ist L regulär, so wird L von einem DFA M akzeptiert.
Daher hat $\equiv_L$ so viele Äquivalenzklassen wie $M/{\equiv_M}$ Zustände.
„$\Longleftarrow$“: Hat $\equiv_L$ endlich viele Äquivalenzklasse,
so ist M_L ein DFA mit $L(M_L) = L$. $\square$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 3.59

$L = \{w \mid w\ \#_0(w) = \#_1(w)\}$ $\qquad$ ($\#_a(w) =$ Anzahl der a in w)

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 3.59

$L = \{w \mid w\ \#_0(w) = \#_1(w)\}$ $\qquad$ ($\#_a(w) =$ Anzahl der a in w)

$$\downarrow$$

$$[\epsilon]$$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 3.59

$L = \{w \mid w\ \#_0(w) = \#_1(w)\}$ $\qquad (\#_a(w) =$ Anzahl der a in $w)$

$$\downarrow [\epsilon] \xrightarrow{1} [1]$$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 3.59

$L = \{w \mid w\ \#_0(w) = \#_1(w)\}$ $\qquad (\#_a(w) =$ Anzahl der a in $w)$

$$\downarrow [\epsilon] \xrightarrow{1} [1] \xrightarrow{1} [1^2]$$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 3.59

$L = \{w \mid w\ \#_0(w) = \#_1(w)\}$ $\qquad (\#_a(w) =$ Anzahl der a in $w)$

$$\cdots \xleftarrow[0]{} [0^2] \xleftarrow[0]{} [0] \xleftarrow[0]{} \downarrow [\epsilon] \underset{0}{\overset{1}{\rightleftarrows}} [1] \xrightarrow{1} [1^2] \xrightarrow{1} \cdots$$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 3.59

$L = \{w \mid w\ \#_0(w) = \#_1(w)\}$ $\qquad (\#_a(w) =$ Anzahl der a in $w)$

$$\cdots \xleftarrow[0]{} [0^2] \xleftarrow[0]{} [0] \xleftarrow[0]{} \downarrow [\epsilon] \underset{0}{\overset{1}{\rightleftarrows}} [1] \underset{0}{\overset{1}{\rightleftarrows}} [1^2] \xrightarrow{1} \cdots$$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 3.59

$L = \{w \mid w \ \#_0(w) = \#_1(w)\}$ $\qquad$ ($\#_a(w) =$ Anzahl der a in w)

$$\cdots \overset{1}{\underset{0}{\rightleftarrows}} [0^2] \overset{1}{\underset{0}{\rightleftarrows}} [0] \overset{1}{\underset{0}{\rightleftarrows}} \overset{\downarrow}{[\epsilon]} \overset{1}{\underset{0}{\rightleftarrows}} [1] \overset{1}{\underset{0}{\rightleftarrows}} [1^2] \overset{1}{\underset{0}{\rightleftarrows}} \cdots$$

Endzustände?

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 3.59

$L = \{w \mid w \ \#_0(w) = \#_1(w)\}$ $\qquad$ ($\#_a(w) =$ Anzahl der a in w)

$$\cdots \overset{1}{\underset{0}{\rightleftarrows}} [0^2] \overset{1}{\underset{0}{\rightleftarrows}} [0] \overset{1}{\underset{0}{\rightleftarrows}} \overset{\downarrow}{[\epsilon]} \overset{1}{\underset{0}{\rightleftarrows}} [1] \overset{1}{\underset{0}{\rightleftarrows}} [1^2] \overset{1}{\underset{0}{\rightleftarrows}} \cdots$$

Endzustände?

$$[1^i]_{\equiv_L} = \{w \mid \#_1(w) = \#_0(w) + i\}$$

$1^i 0^i \in L$ $\qquad$ $1^j 0^i \notin L$

Wie sieht M_L aus wenn L *nicht* regulär ist?

Beispiel 3.59

$L = \{w \mid w \ \#_0(w) = \#_1(w)\}$ $\qquad$ ($\#_a(w) =$ Anzahl der a in w)

$$\cdots \overset{1}{\underset{0}{\rightleftarrows}} [0^2] \overset{1}{\underset{0}{\rightleftarrows}} [0] \overset{1}{\underset{0}{\rightleftarrows}} \overset{\downarrow}{[\epsilon]} \overset{1}{\underset{0}{\rightleftarrows}} [1] \overset{1}{\underset{0}{\rightleftarrows}} [1^2] \overset{1}{\underset{0}{\rightleftarrows}} \cdots$$

Endzustände?

$$[1^i]_{\equiv_L} = \{w \mid \#_1(w) = \#_0(w) + i\}$$

Warum $1^i \not\equiv_L 1^j$ falls $i \neq j$?

Vollständige Methode um Nichtregularität von L zu zeigen:

Bemerkung

Eindeutigkeit des minimalen Automaten (modulo Umbenennung der Zustände) gilt nur bei DFAs, nicht bei NFAs!

Knobelaufgabe:
Nach Def. gilt $p \not\equiv_M q$ gdw $\exists w.\ \hat{\delta}(p,w) \in F \not\Leftrightarrow \hat{\delta}(q,w) \in F$

Man zeige: Falls $p \not\equiv_M q$, dann gibt es ein w der Länge $< |Q|$, das p und q unterscheidet, d.h. $\hat{\delta}(p,w) \in F \not\Leftrightarrow \hat{\delta}(q,w) \in F$.

Knobelaufgabe:
Nach Def. gilt $p \not\equiv_M q$ gdw $\exists w.\ \hat{\delta}(p,w) \in F \not\Leftrightarrow \hat{\delta}(q,w) \in F$

Man zeige: Falls $p \not\equiv_M q$, dann gibt es ein w der Länge $< |Q|$, das p und q unterscheidet, d.h. $\hat{\delta}(p,w) \in F \not\Leftrightarrow \hat{\delta}(q,w) \in F$.

Knobelaufgabe:
Sei $L = \{ww \mid w \in \Sigma^*\}$.

4. Kontextfreie Sprachen

4. Kontextfreie Sprachen

4.1 Kontextfreie Grammatiken

Beispiel 4.1 (Arithmetische Ausdrücke)

<Expr> → <Term>
<Expr> → <Expr> + <Term>
<Term> → <Factor>
<Term> → <Term> ∗ <Factor>
<Factor> → a
<Factor> → (<Expr>)

4. Kontextfreie Sprachen

4.1 Kontextfreie Grammatiken

Beispiel 4.1 (Arithmetische Ausdrücke)

<Expr> → <Term>
<Expr> → <Expr> + <Term>
<Term> → <Factor>
<Term> → <Term> $*$ <Factor>
<Factor> → a
<Factor> → (<Expr>)

Eine (Links)Ableitung:

<Expr> →

$\to a * (a + a)$

4. Kontextfreie Sprachen

4.1 Kontextfreie Grammatiken

Beispiel 4.1 (Arithmetische Ausdrücke)

<Expr> → <Term>
<Expr> → <Expr> + <Term>
<Term> → <Factor>
<Term> → <Term> $*$ <Factor>
<Factor> → a
<Factor> → (<Expr>)

Eine (Links)Ableitung:

<Expr> →

$\to a * (a + a)$

131

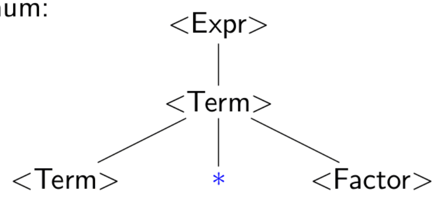
Der Syntaxbaum: <Expr>

Expr

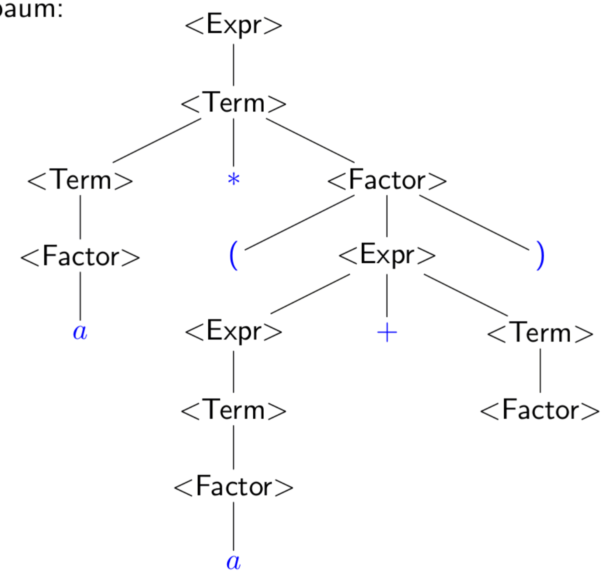
Expr + Term

Der Syntaxbaum: <Expr>

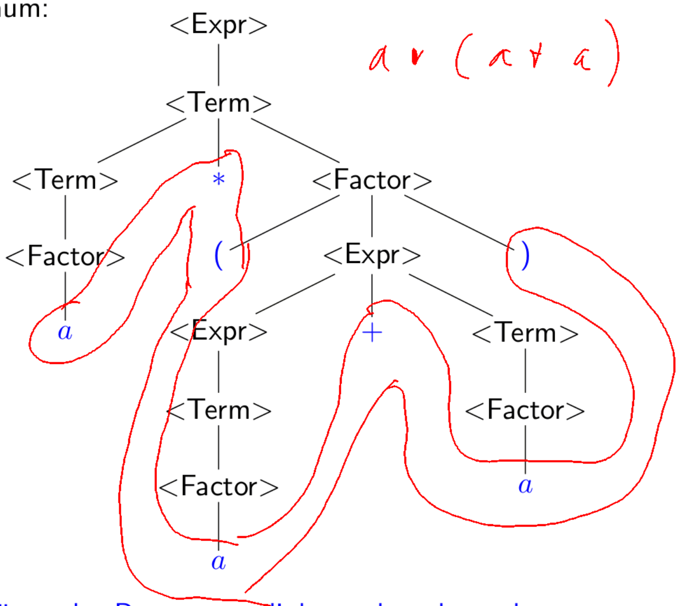
Der Syntaxbaum:



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. Das Parsen ist:
  - 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 4.2

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

Definition 4.2

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

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

Definition 4.2

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.

Definition 4.2

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 $A \to \alpha$ statt $(A, \alpha) \in P$.
- Statt $A \to \alpha_1$, $A \to \alpha_2$, $A \to \alpha_3$ schreiben wir einfach

$$A \to \alpha_1 \mid \alpha_2 \mid \alpha_3$$

Beispiel 4.3 (Arithmetische Ausdrücke)

$$V = \{E, T, F\}$$
$$\Sigma = \{a, +, *, (,)\}$$

Beispiel 4.3 (Arithmetische Ausdrücke)

$$V = \{E, T, F\}$$
$$\Sigma = \{a, +, *, (,)\}$$

$$P = \left\{ \begin{array}{lcl} E & \to & T \mid E+T \\ T & \to & F \mid T*F \\ F & \to & a \mid (E) \end{array} \right\}$$

Beispiel 4.3 (Arithmetische Ausdrücke)

$$V = \{E, T, F\}$$
$$\Sigma = \{a, +, *, (,)\}$$
$$P = \left\{ \begin{array}{lcl} E & \to & T \mid E+T \\ T & \to & F \mid T*F \\ F & \to & a \mid (E) \end{array} \right\}$$

$$S = E$$

Definition 4.4

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

$$\alpha \to_G \beta$$

gdw es eine Regel $A \to \gamma$ 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:

$$a + T + a \quad \to_G \quad a + T * F + a$$

$T \to T * F$

Definition 4.5 (Iteration von $\to_G$)

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

Definition 4.5 (Iteration von $\to_G$)

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

Wir nennen

$$\alpha_1 \to_G \alpha_2 \to_G \cdots \to_G \alpha_n$$

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

Definition 4.6 (Kontextfreie Sprache)

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

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

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

Definition 4.6 (Kontextfreie Sprache)

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

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

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 4.7

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

$$S \to aSb \mid \epsilon$$

erzeugt wird.

Beispiel 4.7

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

$$S \to aSb \mid \epsilon$$

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

$$\begin{array}{rcl} V & = & \{S\} \\ \Sigma & = & \{a, b\} \\ P & = & \{S \to aSb \mid \epsilon\} \end{array}$$

Beispiel 4.7

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

$$S \to aSb \mid \epsilon$$

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

$$\begin{array}{rcl} V & = & \{S\} \\ \Sigma & = & \{a, b\} \\ P & = & \{S \to aSb \mid \epsilon\} \end{array}$$

Der unendliche Baum aller möglichen (Links-)Ableitungen:

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

Beispiel 4.8

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

Beispiel 4.8

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

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

Beispiel 4.8

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

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

Der Anfang des unendlichen Baums aller Ableitungen (Achtung, dies ist *kein* Syntaxbaum!):

S

Lemma 4.9 (Dekompositionslemma)

$$\alpha_1\alpha_2 \to_G^n \beta$$
$$\Leftrightarrow$$
$$\exists \beta_1, \beta_2, n_1, n_2.\ \ \beta = \beta_1\beta_2 \ \wedge\ n = n_1 + n_2 \ \wedge\ \alpha_i \to_G^{n_i} \beta_i \ (i = 1, 2)$$

Lemma 4.9 (Dekompositionslemma)

$$\alpha_1\alpha_2 \to_G^n \beta$$
$$\Leftrightarrow$$
$$\exists \beta_1, \beta_2, n_1, n_2.\ \ \beta = \beta_1\beta_2 \ \wedge\ n = n_1 + n_2 \ \wedge\ \alpha_i \to_G^{n_i} \beta_i \ (i = 1, 2)$$

Beweis:
Übung! □

Definition 4.10

Eine CFG heißt rechtslinear gdw jede Produkion von der Form

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

Eine CFG heißt linkslinear gdw jede Produkion von der Form

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

Definition 4.10

Eine CFG heißt rechtslinear gdw jede Produkion von der Form

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

Eine CFG heißt linkslinear gdw jede Produkion von der Form

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

Lemma 4.11

Die rechtslinearen und linkslinearen Grammatiken erzeugen jeweils genau die regulären Sprachen.

Beweis: Übung!

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

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

Beispiel 4.13

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

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 4.13

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

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

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

Beispiel 4.13

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

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

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

Beispiel 4.13

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

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

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

Beispiel 4.13

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

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

$$\begin{array}{|rcl|} & & \epsilon \in L_G(S) \\ u \in L_G(S) & \Longrightarrow & [u] \in L_G(S) \\ u \in L_G(S) \land v \in L_G(S) & \Longrightarrow & uv \in L_G(S) \end{array}$$

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

Beispiel 4.13

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

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

$$\begin{array}{|rcl|} & & \epsilon \in L_G(S) \\ u \in L_G(S) & \Longrightarrow & [u] \in L_G(S) \\ u \in L_G(S) \land v \in L_G(S) & \Longrightarrow & uv \in L_G(S) \end{array}$$

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

Beispiel 4.13

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

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

$$\begin{array}{|rcl|} & & \epsilon \in L_G(S) \\ u \in L_G(S) & \Longrightarrow & [u] \in L_G(S) \\ u \in L_G(S) \land v \in L_G(S) & \Longrightarrow & uv \in L_G(S) \end{array}$$

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

Beispiel 4.13

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

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

$$\begin{array}{|rcl|} & & \epsilon \in L_G(S) \\ u \in L_G(S) & \Longrightarrow & [u] \in L_G(S) \\ u \in L_G(S) \land v \in L_G(S) & \Longrightarrow & uv \in L_G(S) \end{array}$$

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

Bemerkungen

- Die Produktionen ($\rightarrow$) erzeugen Wörter *top-down*, d.h. von einem Nichtterminal zu einem Wort hin.

Bemerkungen

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

Bemerkungen

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

Bemerkungen

- Die Produktionen ($\rightarrow$) erzeugen Wörter *top-down*, d.h. von einem Nichtterminal zu einem Wort hin.

Bemerkungen

- Die Produktionen ($\rightarrow$) erzeugen Wörter *top-down*, d.h. von einem Nichtterminal zu einem Wort hin.

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,

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 $\forall u \in L_G(S).\ P(u)$,

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:

Beispiel 4.13

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

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

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

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:

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

„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:

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

„Induktion über die Erzeugung von u"

Lemma 4.14
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:

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

„Induktion über die Erzeugung von u"

Lemma 4.14
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:

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

„Induktion über die Erzeugung von u"

Lemma 4.14
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:

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

] [

„Induktion über die Erzeugung von u"

Lemma 4.14
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. □

Hinweis
Die Aussage

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

ist eine Abkürzung für

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

Der Allquantor wird dann oft weggelassen:

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

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:

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

„Induktion über die Erzeugung von u"

Lemma 4.14 $u \in L(S) \Rightarrow$
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. □

Hinweis
Die Aussage

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

ist eine Abkürzung für

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

Der Allquantor wird dann oft weggelassen:

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

Definition 4.15
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 4.15
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:

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

Definition 4.15
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:

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

] [

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

Definition 4.15

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:

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

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 4.15

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:

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

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: [], [[]], [] [], [[] []] []

Definition 4.15

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:

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

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 4.16

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

Satz 4.16

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

Beweis:

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

Satz 4.16

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

Beweis:

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

Mit Ind. über die Erzeugung von u.

Satz 4.16

Die Grammatik $S \to \epsilon \mid [S] \mid SS$ 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).

Definition 4.15

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:

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

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 4.16

Die Grammatik $S \to \epsilon \mid [S] \mid SS$ 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: $p \preceq \epsilon \implies p = \epsilon$

Satz 4.16

Die Grammatik $S \to \epsilon \mid [S] \mid SS$ 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 $p = \epsilon$:

Satz 4.16

Die Grammatik $S \to \epsilon \mid [S] \mid SS$ 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 $p = [u]$: $A(p) = A(u) + 1 = B(u) + 1 = B(p)$

Satz 4.16

Die Grammatik $S \to \epsilon \mid [S] \mid SS$ 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 $p = [u]$: $A(p) = A(u) + 1 = B(u) + 1 = B(p)$

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

Satz 4.16

Die Grammatik $S \to \epsilon \mid [S] \mid SS$ 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 $p = [u]$: $A(p) = A(u) + 1 = B(u) + 1 = B(p)$

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

$A(p) =$

Satz 4.16

Die Grammatik $S \to \epsilon \mid [S] \mid SS$ 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 $p = [u]$: $A(p) = A(u) + 1 = B(u) + 1 = B(p)$

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

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

IA: $A(q) \geq B(q)$

Satz 4.16

Die Grammatik $S \to \epsilon \mid [S] \mid SS$ 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 $p = [u]$: $A(p) = A(u) + 1 = B(u) + 1 = B(p)$

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

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

Satz 4.16

Die Grammatik $S \to \epsilon \mid [S] \mid SS$ 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 $p = [u]$: $A(p) = A(u) + 1 = B(u) + 1 = B(p)$

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 $q \preceq v$:

$A(p) = A(u) + A(q) \underset{(1)}{=} B(u) + A(q) \underset{IA}{\geq} B(u) + B(q) = B(uq) = B(p)$

Beweis (Forts.)
u balanciert $\implies u \in L_G(S)$:

Beweis (Forts.)
u balanciert $\implies u \in L_G(S)$:
Mit vollständiger Induktion über $n := |u|$.

Beweis (Forts.)
u balanciert $\implies u \in L_G(S)$:
Mit vollständiger Induktion über $n := |u|$.(dh $u = a_1 \ldots a_n$)

Beweis (Forts.)
u balanciert $\implies u \in L_G(S)$:
Mit vollständiger Induktion über $n := |u|$.(dh $u = a_1 \ldots a_n$)
IA: Jedes balancierte Wort $< n$ liegt in $L_G(S)$.
Zz: $u \in L_G(S)$.

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

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 $h(w) := A(w) - B(w)$:
$h(\epsilon)$, $h(a_1)$, $h(a_1 a_2)$,..., $h(a_1 \dots a_n)$ (alle ≥ 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 $h(w) := A(w) - B(w)$:
$h(\epsilon)$, $h(a_1)$, $h(a_1 a_2)$,..., $h(a_1 \dots a_n)$ (alle ≥ 0!).
Insb. gilt $a_1 =$ [und $a_n =$].

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 $h(w) := A(w) - B(w)$:
$h(\epsilon)$, $h(a_1)$, $h(a_1 a_2)$,..., $h(a_1 \dots a_n)$ (alle ≥ 0!).

1 2 3 2 3 2 1 0

[[[ ] [ ]]]

Beweis (Forts.)

u balanciert $\implies u \in L_G(S)$:
Mit vollständiger Induktion über $n := |u|$.(dh $u = a_1 \ldots a_n$)
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 $h(w) := A(w) - B(w)$:
$h(\epsilon)$, $h(a_1)$, $h(a_1 a_2)$, $\ldots$, $h(a_1 \ldots a_n)$ (alle ≥ 0!).
Insb. gilt $a_1 =$ [und $a_n =$]. Wir betrachten zwei Fälle:

1 2 3 2 3 2 1 0

[[[ ] [ ]]]

Beweis (Forts.)

u balanciert $\implies u \in L_G(S)$:
Mit vollständiger Induktion über $n := |u|$.(dh $u = a_1 \ldots a_n$)
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 $h(w) := A(w) - B(w)$:
$h(\epsilon)$, $h(a_1)$, $h(a_1 a_2)$, $\ldots$, $h(a_1 \ldots a_n)$ (alle ≥ 0!).
Insb. gilt $a_1 =$ [und $a_n =$]. Wir betrachten zwei Fälle:

$S \to [u]$

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

Beweis (Forts.)

u balanciert $\implies u \in L_G(S)$:
Mit vollständiger Induktion über $n := |u|$.(dh $u = a_1 \ldots a_n$)
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 $h(w) := A(w) - B(w)$:
$h(\epsilon)$, $h(a_1)$, $h(a_1 a_2)$, $\ldots$, $h(a_1 \ldots a_n)$ (alle ≥ 0!).
Insb. gilt $a_1 =$ [und $a_n =$]. Wir betrachten zwei Fälle:

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

Beweis (Forts.)

u balanciert $\implies u \in L_G(S)$:
Mit vollständiger Induktion über $n := |u|$.(dh $u = a_1 \ldots a_n$)
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 $h(w) := A(w) - B(w)$:
$h(\epsilon)$, $h(a_1)$, $h(a_1 a_2)$, $\ldots$, $h(a_1 \ldots a_n)$ (alle ≥ 0!).
Insb. gilt $a_1 =$ [und $a_n =$]. Wir betrachten zwei Fälle:

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

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 $h(w) := A(w) - B(w)$:
$h(\epsilon)$, $h(a_1)$, $h(a_1a_2)$,..., $h(a_1 \dots a_n)$ (alle ≥ 0!).
Insb. gilt $a_1 =$ [und $a_n =$]. Wir betrachten zwei Fälle:

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

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 $h(w) := A(w) - B(w)$:
$h(\epsilon)$, $h(a_1)$, $h(a_1a_2)$,..., $h(a_1 \dots a_n)$ (alle ≥ 0!).
Insb. gilt $a_1 =$ [und $a_n =$]. Wir betrachten zwei Fälle:

- 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 $h(w) := A(w) - B(w)$:
$h(\epsilon)$, $h(a_1)$, $h(a_1a_2)$,..., $h(a_1 \dots a_n)$ (alle ≥ 0!).
Insb. gilt $a_1 =$ [und $a_n =$]. Wir betrachten zwei Fälle:

- 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 $h(w) := A(w) - B(w)$:
$h(\epsilon)$, $h(a_1)$, $h(a_1a_2)$,..., $h(a_1 \dots a_n)$ (alle ≥ 0!).
Insb. gilt $a_1 =$ [und $a_n =$]. Wir betrachten zwei Fälle:

- 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) $p \preceq v$: $h([p) = A([p) - B([p) > 0 \implies$
  $A(p) = A([p) - 1 > B([p) - 1 = B(p) - 1 \implies$
  $A(p) \geq B(p)$
  $\implies v \in L_G(S)$ (nach IA)

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 $h(w) := A(w) - B(w)$:
$h(\epsilon)$, $h(a_1)$, $h(a_1a_2)$,..., $h(a_1 \dots a_n)$ (alle ≥ 0!).
Insb. gilt $a_1 =$ [und $a_n =$]. Wir betrachten zwei Fälle:

- 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) $p \preceq v$: $h([p) = A([p) - B([p) > 0 \implies$
  $A(p) = A([p) - 1 > B([p) - 1 = B(p) - 1 \implies$
  $A(p) \geq B(p)$
  $\implies v \in L_G(S)$ (nach IA) $\implies u = [v] \in L_G(S)$

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.
  Sei $u_1 := a_1 \dots a_k$, $u_2 := a_{k+1} \dots a_n$

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.
  Sei $u_1 := a_1 \dots a_k$, $u_2 := a_{k+1} \dots a_n$
  Dann sind u_1 und u_2 balanciert:

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.
  Sei $u_1 := a_1 \dots a_k$, $u_2 := a_{k+1} \dots a_n$
  Dann sind u_1 und u_2 balanciert:
  (1) $A(u_1) = B(u_1)$ da $h(u_1) = 0$;
  $A(u_2) =$

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.
  Sei $u_1 := a_1 \dots a_k$, $u_2 := a_{k+1} \dots a_n$
  Dann sind u_1 und u_2 balanciert:
  (1) $A(u_1) = B(u_1)$ da $h(u_1) = 0$;

  $A(u_2) = A(u) - A(u_1) = B(u) - B(u_1) =$

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.
  Sei $u_1 := a_1 \dots a_k$, $u_2 := a_{k+1} \dots a_n$
  Dann sind u_1 und u_2 balanciert:
  (1) $A(u_1) = B(u_1)$ da $h(u_1) = 0$;

  $A(u_2) = A(u) - A(u_1) = B(u) - B(u_1) = B(u_2)$

Beweis (Forts.):

- Es gibt noch eine Nullstelle $h(a_1 \dots a_k)$.
  Sei $u_1 := a_1 \dots a_k$, $u_2 := a_{k+1} \dots a_n$
  Dann sind u_1 und u_2 balanciert:
  (1) $A(u_1) = B(u_1)$ da $h(u_1) = 0$;
  $A(u_2) = A(u) - A(u_1) = B(u) - B(u_1) = B(u_2)$

  (2) $p \preceq u_1 \implies p \preceq u \implies A(p) \geq B(p)$
  $p \preceq u_2$: $A(p) = A(u_1 p) - A(u_1) \geq B(u_1 p) - B(u_1) = B(p)$
  $\implies u_1, u_2 \in L_G(S)$ (nach IA, da $|u_i| < n$)
  $\implies u = u_1 u_2 \in L_G(S)$ $\quad S \to SS$

$\square$

Moral:

- „$w \in L_G(S) \implies P(w)$“ beweist man immer schematisch mit Induktion über die Erzeugung von w.
- „$P(w) \implies w \in L_G(S)$“ beweist man oft mit Induktion über $|w|$. Erfordert meist Kreativität.

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

$$A_i \to w_0 A_{i_1} w_1 \ldots w_{n-1} A_{i_n} w_n$$

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

u_1 u_n

$$A_i \to w_0 A_{i_1} w_1 \ldots w_{n-1} A_{i_n} w_n$$

$L(A)$

Man kann G als eine (simultane) induktive Definition von Sprachen $L_G(A_i)$ für all $A_i \in V$ sehen. Jede Produktion korrespondiert zu einer Erzeugungsregel

$$u_1 \in L_G(A_{i_1}) \wedge \cdots \wedge u_n \in L_G(A_{i_n}) \implies w_0 u_1 w_1 \ldots u_n w_n \in L_G(A_i)$$

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

$$A_i \to w_0 A_{i_1} w_1 \ldots w_{n-1} A_{i_n} w_n$$

Man kann G als eine (simultane) induktive Definition von Sprachen $L_G(A_i)$ für all $A_i \in V$ sehen. Jede Produktion korrespondiert zu einer Erzeugungsregel

$$u_1 \in L_G(A_{i_1}) \wedge \cdots \wedge u_n \in L_G(A_{i_n}) \implies w_0 u_1 w_1 \ldots u_n w_n \in L_G(A_i)$$

Aussagen der Form

$$(u \in L_G(A_1) \implies P_1(u)) \wedge \cdots \wedge (u \in L_G(A_k) \implies P_k(u))$$

werden durch simultane Induktion über die Erzeugung von u bewiesen, indem man für jede Produktion zeigt:

$$P_{i_1}(u_1) \wedge \cdots \wedge P_{i_n}(u_n) \implies P_i(w_0 u_1 w_1 \ldots w_{n-1} u_n w_n)$$

Beispiel 4.17

Grammatik:

$$A \to \epsilon \mid aB \qquad B \to Aa$$

Beispiel 4.17

Grammatik:

$$A \to \epsilon \mid aB \qquad B \to Aa$$

Induktive Definition:

Beispiel 4.17

Grammatik:

$$A \to \epsilon \mid aB \qquad B \to Aa$$

Induktive Definition:

- $\epsilon \in L(A)$
- $w \in L(B) \implies aw \in L(A)$
- $w \in L(A) \implies wa \in L(B)$

Beim induktiven Beweis von

$$(w \in L_G(A) \implies \#_a(w) \text{ ist gerade}) \wedge$$
$$(w \in L_G(B) \implies \#_a(w) \text{ ist ungerade})$$

muss man zeigen

- $\#_a(\epsilon)$ ist gerade