HA-Lösung

TA-Lösung

Einführung in die theoretische Informatik – Aufgabenblatt 6

Beachten Sie: Soweit nicht explizit angegeben, sind Ergebnisse stets zu begründen!

Hausaufgaben: Abgabe bis zum 01.06.2016 (Mittwoch) um 12:00

Aufgabe 6.1 Quiz — 1P

Geben Sie jeweils ein Beispiel für formale Sprachen mit folgenden Eigenschaften:

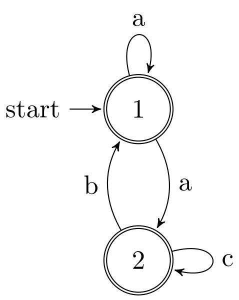
(a) endlich (b) unendlich und regulär (c) nicht regulär und kontextfrei (d) nicht kontextfrei

Lösung:

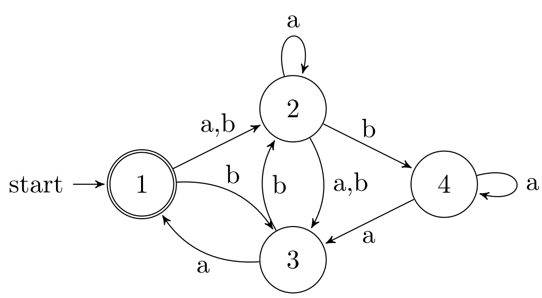
(a) $L_a = \emptyset$ (b) $L_b = L(a^*)$ (c) $L_c = \{a^n b^n \mid n \geq 0\}$ (d) $L_d = \{ww \mid w \in \Sigma^*\}$

Aufgabe 6.2 Ardens Lemma — 1P+2P

Stellen Sie entsprechend der Vorlesung für jeden der folgenden NFAs das lineare Gleichungssystem auf. Bestimmen Sie dann mittels dem Gauß-Verfahren unter Verwendung von Ardens Lemma einen rationalen Ausdruck, der die vom jeweiligen FA akzeptierte Sprache, beschreibt.



(a) NFA N_a



(b) NFA N_b

Lösung: Die Variable X_i für einen Zustand i wird gleichgesetzt mit der Vereinigung der Variablen der in einem Schritt erreichbaren Zuständen mit der Kantenbeschriftung als Präfix. Zusätzlich fügen wir die Konstante $\{\varepsilon\}$ für Endzustände ein.

(a) Aufstellen des Gleichungssystem:

$$\begin{aligned} X_1 &\equiv \varepsilon \mid aX_1 \mid aX_2 \\ X_2 &\equiv \varepsilon \mid bX_1 \mid cX_2 \end{aligned}$$

Lösen nach X_1:

$$\begin{aligned} X_2 &\equiv cX_2 \mid (\varepsilon \mid bX_1) && \text{Umformen} \\ X_2 &\equiv c^*(\varepsilon \mid bX_1) && \text{Ardens Lemma} \\ X_1 &\equiv \varepsilon \mid aX_1 \mid a(c^*(\varepsilon \mid bX_1)) && \text{Einsetzen von } X_2 \\ X_1 &\equiv (a \mid ac^*b)X_1 \mid (\varepsilon \mid ac^*) && \text{Umformen} \\ X_1 &\equiv (a \mid ac^*b)^*(\varepsilon \mid ac^*) && \text{Ardens Lemma} \end{aligned}$$

(b) Aufstellen des Gleichungssystem:

$$\begin{array}{lcl} X_1 & \equiv & \varepsilon \mid (a|b)X_2 \mid bX_3 \\ X_2 & \equiv & aX_2 \mid (a|b)X_3 \mid bX_4 \\ X_3 & \equiv & aX_1 \mid bX_2 \\ X_4 & \equiv & aX_3 \mid aX_4 \end{array}$$

Lösen nach X_1:

$$\begin{array}{lcll} X_1 & \equiv & \varepsilon \mid baX_1 \mid (a|b|bb)X_2 & \text{Einsetzen von } X_3 \\ X_2 & \equiv & (aa|ba)X_1 \mid (a|ab|bb)X_2 \mid bX_4 & \text{Einsetzen von } X_3 \\ X_4 & \equiv & aaX_1 \mid abX_2 \mid aX_4 & \text{Einsetzen von } X_3 \\ X_4 & \equiv & a^*(aaX_1 \mid abX_2) & \text{Ardens Lemma} \\ X_2 & \equiv & (aa|baa^*)X_1 \mid (a|ab|ba^*b)X_2 & \text{Einsetzen von } X_4 \\ X_2 & \equiv & (a|ab|ba^*b)^*(aa|baa^*)X_1 & \text{Ardens Lemma} \\ X_1 & \equiv & \varepsilon \mid (ba \mid (a|b|bb)(a|ab|ba^*b)^*(aa|baa^*))X_1 & \text{Einsetzen von } X_2 \\ X_1 & \equiv & (ba \mid (a|b|bb)(a|ab|ba^*b)^*(aa|baa^*))^* & \text{Ardens Lemma} \end{array}$$

Aufgabe 6.3 Inception 2P

Geben Sie eine CFG G an, die genau die Terme über einer Kleene Algebra $\langle K, +, \cdot, ^*, 0, 1\rangle$ erzeugt, die mittels der Konstanten $0, 1, a, b$ gebildet werden können. Die Multiplikation $\cdot$ soll dabei ausgeschrieben werden. Zum Beispiel sollte Ihre Grammatik $a, a+0, a+b+a, a\cdot a+a^*\cdot a+b\cdot a^*+(a^*)$ erzeugen können, jedoch nicht $+a()$, ^{*1} und auch nicht ab.

Bitte geben Sie für diese Aufgabe nur die Produktionen von G an.

Lösung:

$$T \quad \rightarrow \quad 0 \mid 1 \mid a \mid b \mid T+T \mid T\cdot T \mid T^* \mid (T)$$

Aufgabe 6.4 Kontextfreie Sprachen 2P+3P

Mit $G = (\{S, X, Y, A, B, C\}, \{a, b, c\}, P, S)$ sei die CFG mit folgenden Produktionen P bezeichnet:

$$\begin{array}{lcl} S & \rightarrow & XC \mid AY \\ X & \rightarrow & aXb \mid aA \mid bB \\ Y & \rightarrow & bYc \mid bB \mid cC \\ A & \rightarrow & aA \mid \varepsilon \\ B & \rightarrow & bB \mid \varepsilon \\ C & \rightarrow & cC \mid \varepsilon \end{array}$$

(a) Geben Sie für jedes der folgenden Wörter jeweils eine Linksableitung und eine Rechtsableitung zzgl. des entsprechenden Syntaxbaums an:

$$aabbb \qquad aaabbcc \qquad aaccc$$

(b) Zeigen Sie: $L(G) = \{a^ib^jc^k \mid i \neq j \vee j \neq k\}$.

Lösung:

(a) Trivial.

(b) Mit $L_Z(G)$ sei die von G erzeugte Sprache bezeichnet, wenn man Z als Startsymbol (Axiom) wählt. Offensichtlich gilt $L_A(G) = L(a^*)$, $L_B(G) = L(b^*)$ und $L_C(G) = L(c^*)$ bzw. ergibt aus der induktiven Definition der von A bzw. B bzw. C erzeugten Sprache.

Hiermit folgt mittels Induktion nach der Anzahl der Anwendungen der Regel $X \rightarrow aXb$

$$L_X(G) = \{a^naa^kb^n, a^nbb^kb^n \mid n,k \geq 0,\} = \{a^ib^j \mid i \neq j\}$$

Entsprechend folgt

$$L_Y(G) = \{b^jc^k \mid j \neq k\}$$

Schließlich ergibt sich

$$L(G) = L_S(G) = L_X(G)L_C(G) \cup L_A(G)L_Y(G) = \{a^ib^j \mid i \neq j\}L(c^*) \cup L(a^*)\{b^jc^k \mid j \neq k\} = \{a^ib^jc^k \mid i \neq j \vee j \neq k\}$$

Aufgabe 6.5 **Kleene Algebra** **1P+1P+1P+1P+1P+1P**

Zeigen Sie, dass in jeder Kleene Algebra $\langle K, +, \cdot, ^*, 0, 1\rangle$ für beliebige $a, b, x \in K$ folgende Aussagen gelten. Bitte geben Sie über jedem Schritt in Ihrem Beweis an, welches Axiom der Kleene Algebra Sie verwenden.

(a) $1 \sqsubseteq a^*$, $aa^* \sqsubseteq a^*$ und $a \sqsubseteq a^*$. (b) $a^*a^* = a^*$. (c) $(a^*)^* = a^*$.

(d) $ax \sqsubseteq x \to a^*x \sqsubseteq x$.

Hinweis: Ax6 kann spezialisiert werden zu $1 + a^*a^* \sqsubseteq a^* \to (a^*)^* \sqsubseteq a^*$.

(e) $ax = xb \to a^*x = xb^*$ und damit $a(ba)^* = (ab)^*a$.

Hinweis: Ax6 kann spezialisiert werden zu $x + a(xb^*) \sqsubseteq xb^* \to a^*x \sqsubseteq xb^*$.

(f) $(a + b)^* = a^*(ba^*)^*$.

Hinweis: Ax6 kann spezialsiert werden zu $1 + (a + b)(a^*(ba^*)^*) \sqsubseteq a^*(ba^*)^* \to (a + b)^* \sqsubseteq a^*(ba^*)^*$.

Lösung:

(a) $1 \sqsubseteq 1 + aa^* \sqsubseteq a^*$.

$aa^* \sqsubseteq 1 + aa^* \sqsubseteq a^*$.

$a = a \cdot 1 \sqsubseteq aa^* \sqsubseteq 1 + aa^* \sqsubseteq a^*$.

(b) $\text{Ax6}[a^*/b][a^*/x] =_{\text{syn}} a^* + aa^* \sqsubseteq a^* \to a^*a^* \sqsubseteq a^*$.

Mit Ax5: $a^* + aa^* \sqsubseteq 1 + aa^* + a^* = a^* + a^* = a^*$.

Also: $a^*a^* \sqsubseteq a^*$.

Mit (a) $1 \sqsubseteq a^*$: $a^* = a^* \cdot 1 \sqsubseteq a^*a^*$.

Insgesamt: $a^*a^* = a^*$.

(c) Da * monoton und $a \sqsubseteq a^*$: $a^* \sqsubseteq (a^*)^*$.

$\text{Ax6}[a^*/a][1/b][a^*/x] =_{\text{syn}} 1 + a^*a^* \sqsubseteq a^* \to (a^*)^* \sqsubseteq a^*$.

Mit (b) $a^*a^* = a^*$ und (a) $1 \sqsubseteq a^*$: $1 + a^*a^* \sqsubseteq a^* + a^* = a^*$.

Also $(a^*)^* \sqsubseteq a^*$.

(d) $\text{Ax6}[x/b] =_{\text{syn}} x + ax \sqsubseteq x \to a^*x \sqsubseteq x$.

Gilt somit $ax \sqsubseteq x$, dann $x + ax \sqsubseteq x + x = x$, also auch $a^*x \sqsubseteq x$.

(e) Wir zeigen $ax \sqsubseteq xb \to a^*x \sqsubseteq xb^*$.

Es gelte $ax \sqsubseteq xb$.

$\text{Ax6}[x/b, xb^*/x] =_{\text{syn}} x + a(xb^*) \sqsubseteq xb^* \to a^*x \sqsubseteq xb^*$ (gleichzeitige Substitution).

Es reicht somit $x + a(xb^*) \sqsubseteq xb^*$ zu zeigen.

Mit $1 \sqsubseteq b^*$: $axb^* \sqsubseteq xbb^*$

Damit $x + axb^* \sqsubseteq x + xbb^* = x(1 + bb^*) \sqsubseteq xb^*$, was zu zeigen war.

Symmtrisch $xb \sqsubseteq ax \to xb^* \sqsubseteq a^*x$.

Also: $ax = bx \to a^*x = xb^*$.

Damit $(ax = bx \to a^*x = xb^*)[a/x, ab/a, ba/b] =_{\text{syn}} (aba = aba \to (ab)^*a = a(ba)^*)$.

(f) Wir zeigen $(a + b)^* \sqsubseteq a^*(ba^*)^*$:

Wegen $\text{Ax6}[1/b, (a + b)/a, a^*(ba^*)^*/x] =_{\text{syn}} 1 + (a + b)(a^*(ba^*)^*) \sqsubseteq a^*(ba^*)^* \to (a + b)^* \sqsubseteq a^*(ba^*)^*$

reicht es $1 + (a + b)(a^*(ba^*)^*)$ zu zeigen.

Es gilt $1 = 1 \cdot 1 \sqsubseteq a^*(ba^*)^*$.

Damit: $(a + b) \sqsubseteq 1 + (a + b)a^*(ba^*)^* \sqsubseteq 1 + aa^*(ba^*)^* + ba^*(ba^*)^*$

Mit (a) $ba^*(ba^*)^* \sqsubseteq (ba^*)^*$ und $1 + aa^* \sqsubseteq a^*$:

$1 + aa^*(ba^*)^* + ba^*(ba^*)^* \sqsubseteq 1 + aa^*(ba^*)^* + (ba^*)^* \sqsubseteq 1 + a^*(ba^*)^* = a^*(ba^*)^*$.

Bleibt $a^*(ba^*)^* \sqsubseteq (a + b)^*$:

$a^*(ba^*)^* \sqsubseteq (a + b)^*((a + b)(a + b)^*)^* \sqsubseteq (a + b)^*((a + b)^*)^* = (a + b)^*(a + b)^* = (a + b)^*$.

Tutoraufgaben: Besprechung in KW22

Aufgabe 6.1

Die CFG G bestehe aus folgenden Produktionen über dem Alphabet $\Sigma = \{a, b\}$:

$$\begin{array}{lcl} S & \to & ASA \mid aB \\ A & \to & B \mid S \mid CB \\ B & \to & b \mid \varepsilon \\ C & \to & aC \\ D & \to & aSCb \mid a \end{array}$$

(a) Reduzieren Sie die Grammatik G auf die nützlichen Nichtterminale.

(b) Überführen Sie die reduzierte Grammatik dann in Chomsky-Normalform.

Lösung:

(a) *Erreichbare Variablen*:

- $R_0 := \{S\}$
- $R_{k+1} := R_k \cup \{Y \in V \mid \exists (X, \gamma) \in P \colon X \in R_k \wedge |\gamma|_Y > 0\}$ bis $R_{k+1} = R_k$.

▷ Führt auf:

$$R_0 = \{S\} \qquad R_1 = \{S, A, B\} \qquad R_2 = \{S, A, B, C\} = R_3$$

Damit kann D und $D \to aSCb \mid a$ entfernt werden.

Für Implementierung: muss nur Regeln für Variablen in $R_{k+1} \setminus R_k$ jeweils überprüfen, daher wie bei Erreichbarkeit auf Graphen immer nur neu "besuchte" Variablen in die Queue (work list) schieben.

Produktive Variablen:

- $P_0 := \{X \mid \exists (X, \gamma) \in P \colon \gamma \in \Sigma^*\}$
- $P_{k+1} := P_k \cup \{X \in V \mid \exists (X, \gamma) \in P \forall Y \in V \colon (|\gamma|_Y > 0 \to Y \in P_k)\}$ bis $P_{k+1} = P_k$.

▷ Führt auf:

$$P_0 = \{B\} \qquad P_1 = \{B, A, S\} = P_2$$

Damit kann C samt $C \to aC$ und $A \to CB$ entfernt werden.

Für Implementierung: für jede neu als produktiv erkannte Variable $Y \in P_{k+1} \setminus P_k$ alle Vorkommen von Y in einer Regel (X, γ) mit $|\gamma|_Y > 0$ markieren; wenn dann alle Variablen in γ markiert sind, direkt X markieren und in work list schieben.

Ab jetzt steht G für die reduzierte (bereinigte) Grammatik:

$$S \to ASA \mid aB \qquad A \to B \mid S \qquad B \to b \mid \varepsilon$$

(b) *Erkennen von ε-Regeln*:

- $E_0 := \{X \in V \mid (X, \varepsilon) \in P\}$
- $E_{k+1} := E_k \cup \{X \in V \mid \exists (X, \gamma) \in P \colon \gamma \in E_k^*\}$ bis $E_{k+1} = E_k$.

$$E_0 = \{B\} \qquad E_1 = \{B, A\} = E_2$$

- Erzeugen zusätzlicher Produktionen, welche alle möglichen Kombinationen von ε-Produktionen beachten:

$$\begin{array}{lcl} S \to ASA & \rightsquigarrow & S \to ASA \mid AS \mid SA \mid S \\ S \to aB & \rightsquigarrow & S \to aB \mid a \\ A \to B & \rightsquigarrow & A \to B \mid \varepsilon \\ A \to S & \rightsquigarrow & A \to S \\ B \to b & \rightsquigarrow & B \to b \\ B \to \varepsilon & \rightsquigarrow & B \to \varepsilon \end{array}$$

d.h. man ersetzt z.B. $S \to ASA$ durch $S \to (A \mid \varepsilon)S(A \mid \varepsilon)$ und multipliziert einfach aus.

- Entfernen aller ε-Produktionen:

$$G' \colon S \to ASA \mid AS \mid SA \mid S \mid aB \mid a \qquad A \to B \mid S \qquad B \to b$$

Zusammenziehen von Kettenproduktionen:

- $T_0 := \{(X,Y) \in P \cap V \times V\}$
- $T_{k+1} := T_k \cup \{(X,Y) \in V \times V \mid \exists Z \in V\colon (X,Z) \in T_k \wedge (Z,Y) \in T_k\}$ bis $T_{k+1} = T_k =: T_*$ (einfach transitiver Abschluss über durch Kettenproduktionen gegebener Kantenrelation auf V).

$$T_0 = \{(S,S),(A,B),(A,S)\} = T_1$$

- Dann (1) Entfernen aller Kettenproduktion und (2) anschließend, falls $(X,Y) \in T_*$, füge (X,γ) zu P hinzu für jede Regel $(Y,\gamma) \in P$.

$$G''\colon S \to ASA \mid AS \mid SA \mid aB \mid a \qquad A \to b \mid ASA \mid AS \mid SA \mid aB \mid a \qquad B \to b$$

Überführung in CNF:

- In jeder Regel (X,γ) mit $|\gamma| \geq 2$ Ersetzen jedes Vorkommen eines Terminals x durch X_x und Ergänzen die benötigten Produktionen $X_x \to x$:

$$G'''\colon S \to ASA \mid AS \mid SA \mid X_aB \mid a \qquad A \to b \mid ASA \mid AS \mid SA \mid X_aB \mid a \qquad B \to b \qquad X_a \to a$$

  Alle rechten Seiten sind jetzt von der Form $VVV^* \cup \Sigma$.

- Überführen aller rechten Seiten, welche aus mindestens drei Variablen bestehen, in quadratische Monome über V. Die einfachste Variante ist dabei, aus XYZ einfach XX_{YZ} machen, wobei X_{YZ} einfach eine Hilfsvariable ist, die das Ergebnis von YZ mittels $X_{YZ} \to YZ$ zugewiesen bekommt, ganz analog zu Gleichungssystemen über $\mathbb{R}$ (auch üblich, aber vermutlich etwas verwirrend: $X[YZ]$ oder $X\langle YZ\rangle$ mit $[YZ]$ bzw. $\langle YZ\rangle$ jetzt Variable). Damit:

$$G''''\colon S \to AX_{SA} \mid AS \mid SA \mid X_aB \mid a \qquad A \to b \mid AX_{SA} \mid AS \mid SA \mid X_aB \mid a \qquad B \to b \qquad X_a \to a \qquad X_{SA} \to SA$$

Aufgabe 6.2 Pumping-Lemma (CFL-Edition)

Wir betrachten Pfeil-Sprachen über dem Alphabet $\Sigma = \{\uparrow,\downarrow,\leftarrow,\rightarrow\}$. Wir interpretieren ein Wort $w \in \Sigma^*$ als einen Pfad in einem 2D-Gitter. Verwenden Sie das Pumping-Lemma für kontextfreie Sprachen, um zu zeigen, dass folgende Mengen von Pfaden nicht kontextfrei sind:

(a) Pfade, die „umkehren“ — beliebig weit nach rechts fahren, dann entweder nach oben oder unten gehen und dann nach links über die Höhe vom Ursprung nach links fahren:

$$L_a = \bigcup_{i<j<k} L(\rightarrow^i (\uparrow^j \mid \downarrow^j) \leftarrow^k)$$

(b) Pfade, die in den Ursprung zurückkehren:

$$L_b = \{w \in \Sigma^* \mid |w|_\uparrow = |w|_\downarrow \wedge |w|_\leftarrow = |w|_\rightarrow\}$$

Lösung:

(a) Wir nehmen an, dass L_a kontextfrei ist und führen diese Annahme zum Widerspruch.

Sei $n \geq 1$ eine Pumping-Lemma-Zahl für die kontextfreie Sprache L_a. Dann ist $z = \rightarrow^n\uparrow^{n+1}\leftarrow^{n+2}$, d.h., $z \in L_a$ und $|z| \geq n$. Gemäß Pumping-Lemma gibt es dann also eine Zerlegung $z = uvwxy$ mit Wörtern $u,v,w,x,y \in \Sigma^*$ und den folgenden Eigenschaften:

$$(1)\ vx \neq \epsilon \qquad (2)\ |vwx| \leq n \qquad (3)\ \forall i \in \mathbb{N}_0.\ uv^iwx^iy \in L_a$$

Zuerst informell: Da $|vwx| \leq n$, kann vwx nur von der Form $\rightarrow^*\uparrow^*$ oder $\uparrow^*\leftarrow^*$ sein. Wegen $|vx| > 0$ muss mindestens ein Pfeil gepumpt werden. Gilt $vw \in L(\uparrow^*)$, dann können wir die Anzahl der $\uparrow$ über die Anzahl der $\leftarrow$ pumpen, enthält vw mindestens ein $\rightarrow$ und damit kein $\leftarrow$, so kann man die Anzahl der $\rightarrow$ beliebig groß, insbesondere größer als die Anzahl der $\leftarrow$ machen. Besteht vw nur aus $\leftarrow$, dann aus mindestens einem, so dass man durch Entfernen von vx die Anzahl der $\leftarrow$ auf $n+1$ oder weniger reduziert wird, man also höchstens so viele $\leftarrow$ wie $\uparrow$ hat.

Formal: Aufgrund von (1) unterscheiden wir die folgenden Fälle:

- $|vx|_\rightarrow > 0$: Wegen (2) gilt $|vx|_\leftarrow = 0$. Allerdings gilt auch:

$$\left|uv^3wx^3y\right|_\rightarrow = n + 2\,|vx|_\rightarrow \geq n+2 = \left|uv^3wx^3y\right|_\leftarrow$$

  Dies ist ein Widerspruch zu (3).

- $|vx|_\rightarrow = 0$ und $|vx|_\uparrow > 0$: Dann gilt:

$$\left|uv^0wx^0y\right|_\uparrow = n + 1 - |vx|_\uparrow \leq n = \left|uv^0wx^0y\right|_\rightarrow$$

  Daher ist $uv^0wx^0y \notin L_a$, ein Widerspruch zu (3).

- $|vx|_\rightarrow = 0$ und $|vx|_\uparrow = 0$: Dann muss $|vx|_\leftarrow > 0$ gelten, und es folgt:

$$\left|uv^0wx^0y\right|_\leftarrow = n + 2 - |vx|_\leftarrow < n + 1 = \left|uv^0wx^0y\right|_\uparrow$$

  Daher ist $uv^0wx^0y \notin L_a$, ein Widerspruch zu (3).

Da jeder Fall zu einem Widerspruch führt und die obigen Fälle alle möglichen Zerlegungen abdecken, kann die ursprüngliche Annahme nicht gelten. Also ist L_a nicht kontextfrei.

(b) Wir nehmen an, dass L_b kontextfrei ist und führen diese Annahme zum Widerspruch.

Sei $n \geq 1$ eine Pumping-Lemma-Zahl für die kontextfreie Sprache L_b. Sei zusätzlich $z = \rightarrow^n\uparrow^n\leftarrow^n\downarrow^n$, d.h., $z \in L_b$ und $|z| \geq n$. Gemäß Pumping-Lemma gibt es dann also eine Zerlegung $z = uvwxy$ mit Wörtern $u, v, w, x, y \in \Sigma^*$ und den folgenden Eigenschaften:

$$(1)\ vx \neq \epsilon \qquad (2)\ |vwx| \leq n \qquad (3)\ \forall i \in \mathbb{N}_0.\ uv^iwx^iy \in L_b$$

Aufgrund von (1) unterscheiden wir die folgenden Fälle:

- $|vx|_\rightarrow > 0$: Wegen (2) gilt $|vx|_\leftarrow = 0$. Allerdings gilt auch:

$$\left|uv^2wx^2y\right|_\rightarrow = n + |vx|_\rightarrow > n = \left|uv^2wx^2y\right|_\leftarrow$$

  Dies ist ein Widerspruch zu (3).

- $|vx|_\uparrow > 0$: Wegen (2) gilt $|vx|_\downarrow = 0$. Allerdings gilt auch:

$$\left|uv^2wx^2y\right|_\uparrow = n + |vx|_\uparrow > n = \left|uv^2wx^2y\right|_\downarrow$$

  Dies ist ein Widerspruch zu (3).

- $|vx|_\leftarrow > 0$: Wegen (2) gilt $|vx|_\rightarrow = 0$. Allerdings gilt auch:

$$\left|uv^2wx^2y\right|_\rightarrow = n < n + |vx|_\leftarrow = \left|uv^2wx^2y\right|_\leftarrow$$

  Dies ist ein Widerspruch zu (3).

- $|vx|_\downarrow > 0$: Wegen (2) gilt $|vx|_\uparrow = 0$. Allerdings gilt auch:

$$\left|uv^2wx^2y\right|_\uparrow = n < n + |vx|_\downarrow = \left|uv^2wx^2y\right|_\downarrow$$

  Dies ist ein Widerspruch zu (3).

Da jeder Fall zu einem Widerspruch führt und die obigen Fälle alle möglichen Zerlegungen abdecken, kann die ursprüngliche Annahme nicht gelten. Also ist L_b nicht kontextfrei.

Aufgabe 6.3

Eine Sprache $L \subseteq \Sigma^*$ heißt *linear*, falls es eine kontextfreie Grammatik G gibt, so dass auf der rechten Seite einer jeden Produktion höchstens ein Nichtterminal steht (d.h. $P \subseteq V \times (\Sigma^* V \Sigma^* \cup \Sigma^*)$). Zeigen Sie folgende Aussagen:

(a) Ist L linear mit $\varepsilon \notin L$, dann gibt es eine CFG G, so dass $L = L(G)$ und für jede Produktion $X \rightarrow \gamma$ gilt: $\gamma \in \Sigma V \cup V\Sigma \cup \Sigma$.

(b) Sei $G = (V, \Sigma, P, S)$ von der Gestalt aus (a), dann gilt:

$$\forall z \in L. \Big(|z| \geq |V| + 1 \rightarrow \exists u, v, w, x, y \in \Sigma^*. \big(z = uvwxy \land |vx| > 0 \land |uvxy| \leq |V| + 1 \land \forall i \in \mathbb{N}_0.(uv^iwx^iy \in L)\big)\Big)$$

Lösung:

(a) Start mit beliebiger linearer CFG, Entfernen aller ε-Regeln und aller Kettenproduktionen, schließlich: falls $X \rightarrow awYv$ mit $a \in \Sigma, w \in \Sigma^+, v \in \Sigma^*, Y \in V$, Übergang zu $X \rightarrow aX_{wYv}$ und $X_{wYv} \rightarrow wYv$, wiederholen bis alle Regeln von der Form $X \rightarrow aY$ oder $X \rightarrow Yv$ mit $v \in \Sigma^+$. Entsprechend $X \rightarrow Yv$ noch überführen.

(b) Sei G lineare Grammatik in der Form von (a) mit $L - \{\varepsilon\} = L(G)$.

Sei $z \in L(G)$ mit $|z| \geq n := |V| + 1$ (wie im Fall von RL und im Gegensatz zu CFL reicht das hier um sicher eine Wiederholung zu haben auf Grund der Normalform aus (a)). Dann gibt es eine Ableitung

$$X_1 = S \to_G \alpha_1 X_2 \beta_1 \to_G \alpha_1\alpha_2 X_3 \beta_2\beta_1 \to_G^* \alpha_1\alpha_2 \dots \alpha_k X_{k+1} \beta_k \dots \beta_2\beta_1 \to_G \alpha_1\alpha_2 \dots \alpha_k \gamma \beta_k \dots \beta_2\beta_1 = z$$

Nun gilt $1 = |\gamma| = |\alpha_i \beta_i|$ und damit

$$|V| + 1 = n \leq |z| = |\alpha_1\alpha_2 \dots \alpha_k \gamma \beta_k \dots \beta_2\beta_1| = k + 1$$

und damit $k \geq |V|$, womit unter den $k+1$ Nichtterminalen $X_1, \dots, X_{k+1}$ (Schubfachprinzip) mindestens eines doppelt vorkommen muss, d.h. es ein Paar $1 \leq i < j \leq k$ mit $X_i = X_j$ gibt. Unter allen möglichen Paaren (i, j) wählen wir das Paar mit j minimal, also die Variable, die als erstes zum zweiten Mal besucht wird. Insbesondere sind damit alle Variablen $X_1, \dots, X_{j-1}$ von einander paarweise verschieden.

Damit ergibt sich die Zerlegung $u = \alpha_1 \dots \alpha_i$, $v = \alpha_{i+1} \dots \alpha_j$, $w = \alpha_{j+1} \dots \gamma \dots \beta_{j+1}$, $x = \beta_{i+1} \dots \beta_j$ und $y = \beta_i \dots \beta_1$, welche sich durch Weglassen bzw. Wiederholen der Ableitung $X_i \to_G^* vX_ix$ beliebig pumpen lässt.

Da G normalisiert ist, wird in jedem Ableitungsschritt mindestens ein Terminal erzeugt, weswegen $|vx| > 0$ folgt.

Wäre $|uvxy| > n$, dann würde entlang der Ableitung

$$X_1 \to_G^* uvX_jxy$$

wieder ein Nichtterminal mehrmals vorkommen, was aber der Wahl von j widerspricht.