

Einführung in die Theoretische Informatik Sommersemester 2024 – Übungsblatt 6

- Das Übungsblatt ist in zwei Teile gegliedert: den Vorbereitungsteil, den Sie vor der Übung selbstständig bearbeiten sollen, und den Übungs-/Nachbereitungsteil, der Aufgaben enthält, die in der Übung besprochen werden und von Ihnen anschließend zur Nachbereitung verwendet werden können.

Vorbereitung (→ vor der Übung selbstständig zu bearbeiten)

Vorbereitungsaufgabe Ü6.1. (Wichtige Begriffe)

Überprüfen Sie, dass Sie die folgenden Begriffe oder Notationen korrekt definieren können.

- nützlich, erzeugend, erreichbar (Symbole)
- Chomsky-Normalform
- Greibach-Normalform
- Pumping-Lemma für kontextfreie Sprachen

Vorbereitungsaufgabe Ü6.2. (Automata Tutor: “Chomsky-Normalform”)

Lösen Sie die Aufgaben Ü6.2 (a–b) auf Automata Tutor.

Hinweis: Automata Tutor überprüft nur, ob Ihre Grammatik in CNF ist und die richtige Sprache erzeugt. Sie selbst sollen die Grammatik jedoch nach Verfahren der Vorlesung in CNF überführen.

Lösungsskizze.

(a) Zwischenschritte:

$$\begin{array}{l} S \rightarrow aSb \mid A \mid B \\ A \rightarrow aA \mid a \\ B \rightarrow bB \mid b \end{array} \rightsquigarrow \begin{array}{l} S \rightarrow A_a S A_b \mid A \mid B \\ A \rightarrow A_a A \mid a \\ B \rightarrow A_b B \mid b \\ A_a \rightarrow a \\ A_b \rightarrow b \end{array}$$

$$\rightsquigarrow \begin{array}{l} S \rightarrow A_a C_1 \mid A \mid B \\ C_1 \rightarrow S A_b \\ A \rightarrow A_a A \mid a \\ B \rightarrow A_b B \mid b \\ A_a \rightarrow a \\ A_b \rightarrow b \end{array} \rightsquigarrow \begin{array}{l} S \rightarrow A_a C_1 \mid A \mid B \\ C_1 \rightarrow S A_b \\ A \rightarrow A_a A \mid a \\ B \rightarrow A_b B \mid b \\ A_a \rightarrow a \\ A_b \rightarrow b \end{array}$$

Lösung:

$$\begin{aligned}
 S &\rightarrow A_a C_1 \mid A_a A \mid a \mid A_b B \mid b \\
 C_1 &\rightarrow S A_b \\
 A &\rightarrow A_a A \mid a \\
 B &\rightarrow A_b B \mid b \\
 A_a &\rightarrow a \\
 A_b &\rightarrow b
 \end{aligned}$$

(b) Zwischenschritte:

$$\begin{array}{l}
 S \rightarrow KDT \mid aT \\
 K \rightarrow c \mid j \mid ET \mid D \\
 D \rightarrow bDS \mid abb \\
 T \rightarrow \varepsilon \\
 E \rightarrow a \mid T \\
 A_a \rightarrow a \\
 A_b \rightarrow b
 \end{array}
 \quad \rightsquigarrow \quad
 \begin{array}{l}
 S \rightarrow KDT \mid A_a T \\
 K \rightarrow c \mid j \mid ET \mid D \\
 D \rightarrow A_b DS \mid A_a A_b A_b \\
 T \rightarrow \varepsilon \\
 E \rightarrow a \mid T \\
 A_a \rightarrow a \\
 A_b \rightarrow b
 \end{array}$$

$$\begin{array}{l}
 S \rightarrow KC_1 \mid A_a T \\
 C_1 \rightarrow DT \\
 K \rightarrow c \mid j \mid ET \mid D \\
 D \rightarrow A_b C_2 \mid A_a C_3 \\
 C_2 \rightarrow DS \\
 C_3 \rightarrow A_b A_b \\
 T \rightarrow \varepsilon \\
 E \rightarrow a \mid T \\
 A_a \rightarrow a \\
 A_b \rightarrow b
 \end{array}
 \quad \rightsquigarrow \quad
 \begin{array}{l}
 S \rightarrow KC_1 \mid C_1 \mid A_a T \mid A_a \\
 C_1 \rightarrow DT \mid D \\
 K \rightarrow c \mid j \mid ET \mid T \mid E \mid D \\
 D \rightarrow A_b C_2 \mid A_a C_3 \\
 C_2 \rightarrow DS \\
 C_3 \rightarrow A_b A_b \\
 E \rightarrow a \mid T \\
 A_a \rightarrow a \\
 A_b \rightarrow b
 \end{array}$$

Lösung:

$$\begin{aligned}
 S &\rightarrow KC_1 \mid A_a T \mid DT \mid a \mid A_b C_2 \mid A_a C_3 \\
 C_1 &\rightarrow DT \mid A_b C_2 \mid A_a C_3 \\
 K &\rightarrow c \mid j \mid ET \mid a \mid A_b C_2 \mid A_a C_3 \\
 D &\rightarrow A_b C_2 \mid A_a C_3 \\
 C_2 &\rightarrow DS \\
 C_3 &\rightarrow A_b A_b \\
 E &\rightarrow a \\
 A_a &\rightarrow a \\
 A_b &\rightarrow b
 \end{aligned}$$

Übung und Nachbereitung

Übungsaufgabe Ü6.3. (*Express Yourself (Whoa Do It)*)

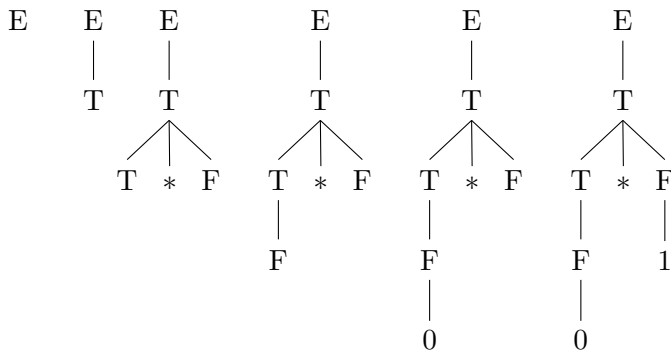
Wie in der Vorlesung besprochen, entspricht jede Ableitung $S \rightarrow \alpha_1 \rightarrow \dots \rightarrow \alpha_n$ bezüglich einer kontextfreien Grammatik $G = (V, \Sigma, P, S)$ einem Syntaxbaum. Jedes Vorkommen eines Zeichens $\mathbb{X} \in (V \cup \Sigma \cup \{\varepsilon\})$ in der Ableitung entspricht dabei einem Knoten des Syntaxbaums. Der Baum entsteht dabei induktiv:

- Die Wurzel des Baums ist das Startsymbol S .
- Wird in der Wortableitung auf das Vorkommen eines Nichtterminals A eine Regel $A \rightarrow \mathbb{X}_1 \dots \mathbb{X}_n$ mit $\mathbb{X}_i \in (V \cup \Sigma \cup \{\varepsilon\})$ angewandt, so werden dem dazugehörigen Knoten entsprechende Kinder $\mathbb{X}_1 \dots \mathbb{X}_n$ hinzugefügt.

Betrachte beispielsweise die Grammatik $G_1 = (\{E, T, F\}, \{(\cdot), 0, 1, +, *\}, P_1, E)$ für arithmetische Ausdrücke (aus der Vorlesung, Beispiel 4.1) mit folgenden Produktionen P_1 :

$$E \rightarrow T \mid E + T \quad T \rightarrow F \mid T * F \quad F \rightarrow 0 \mid 1 \mid (E)$$

Für die Ableitung $E \rightarrow T \rightarrow T * F \rightarrow F * F \rightarrow 0 * F \rightarrow 0 * 1$ entstehen schrittweise die folgenden Syntaxbäume:



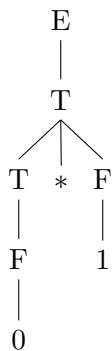
Für einen Syntaxbaum eines Wortes w produziert durch G_1 können wir die Semantik (die "mathematische Bedeutung") rekursiv berechnen:

$$\begin{array}{ll} \left[\begin{array}{c} E \\ | \\ \mathcal{B} \end{array} \right] = \llbracket \mathcal{B} \rrbracket & \left[\begin{array}{c} E \\ / \quad + \quad \backslash \\ \mathcal{B}_1 \quad \mathcal{B}_2 \end{array} \right] = \llbracket \mathcal{B}_1 \rrbracket + \llbracket \mathcal{B}_2 \rrbracket \\ \left[\begin{array}{c} T \\ | \\ \mathcal{B} \end{array} \right] = \llbracket \mathcal{B} \rrbracket & \left[\begin{array}{c} T \\ / \quad * \quad \backslash \\ \mathcal{B}_1 \quad \mathcal{B}_2 \end{array} \right] = \llbracket \mathcal{B}_1 \rrbracket * \llbracket \mathcal{B}_2 \rrbracket \\ \left[\begin{array}{c} F \\ | \\ \mathcal{B} \end{array} \right] = \llbracket \mathcal{B} \rrbracket & \left[\begin{array}{c} F \\ / \quad \backslash \\ (\quad \mathcal{B} \quad) \end{array} \right] = \llbracket \mathcal{B} \rrbracket \end{array} \quad \llbracket 0 \rrbracket = 0 \quad \llbracket 1 \rrbracket = 1$$

Dabei stehen $\mathcal{B}, \mathcal{B}_1, \mathcal{B}_2$ für beliebige Teilbäume.

Beachten Sie: die Semantik ist hier eine natürliche Zahl, die sie insbesondere mittels arithmetischen Regeln vereinfachen dürfen. Beispielsweise gilt $0 + 1 = 1$ für natürliche Zahlen (semantische Gleichheit), aber $0 + 1 \neq 1$ für Wörter über dem Alphabet Σ (syntaktische Gleichheit).

- (a) Bestimmen Sie schrittweise die Semantik des folgenden Syntaxbaums B durch Anwenden der entsprechenden Regeln.



- (b) Existiert für das Wort $0 * 1$ ein weiterer Syntaxbaum B' , sodass $\llbracket B \rrbracket \neq \llbracket B' \rrbracket$?
(c) Existiert ein Wort $w' \neq 0 * 1$ mit Syntaxbaum B' , sodass $\llbracket B \rrbracket = \llbracket B' \rrbracket$?
(d) Betrachte nun die Grammatik $G_2 = (\{S\}, \{\text{if, then, else, 0, 1, +, *, =}\}, P_2, S)$ mit folgenden Produktionen P_2 :

$$S \rightarrow 0 \mid 1 \mid S + S \mid S * S \mid \text{if } S = S \text{ then } S \text{ else } S$$

Geben Sie, analog zur Grammatik G_1 , eine mathematisch sinnvolle Semantik für Syntaxbäume bezüglich G_2 rekursiv an.

- (e) Bestimmen Sie einen Syntaxbaum und dessen Semantik für das Wort $1 + \text{if } 1 = 0 \text{ then } 0 \text{ else } 1$
(f) Existiert ein Wort w mit Syntaxbäumen B, B' bezüglich G_2 , sodass $\llbracket B \rrbracket \neq \llbracket B' \rrbracket$?

Lösungsskizze.

- (a)

$$\begin{aligned}
 \left[\begin{array}{c} E \\ | \\ T \\ / \quad | \quad \backslash \\ T \quad * \quad F \\ | \quad \quad | \\ F \quad \quad 1 \\ | \\ 0 \end{array} \right] &= \left[\begin{array}{c} T \\ / \quad | \quad \backslash \\ T \quad * \quad F \\ | \quad \quad | \\ F \quad \quad 1 \\ | \\ 0 \end{array} \right] = \left[\begin{array}{c} T \\ | \\ F \\ | \\ 0 \end{array} \right] * \left[\begin{array}{c} F \\ | \\ 1 \end{array} \right] = \left[\begin{array}{c} F \\ | \\ 0 \end{array} \right] * \left[\begin{array}{c} F \\ | \\ 1 \end{array} \right] \\
 &= \llbracket 0 \rrbracket * \left[\begin{array}{c} F \\ | \\ 1 \end{array} \right] = 0 * \left[\begin{array}{c} F \\ | \\ 1 \end{array} \right] = 0
 \end{aligned}$$

- (b) Nein, es gibt nämlich nur einen Syntaxbaum für w : Die erste Regel muss $E \rightarrow T$ sein, da w kein $+$ enthält. Die nächste Regel muss $T \rightarrow T * F$ sein, da w ein $*$ und kein $(,)$ enthält. Das rechte F muss somit mit $F \rightarrow 1$ abgeleitet werden. Das T muss mit $T \rightarrow F$ weiter abgeleitet werden, denn w enthält keine zwei $*$. Das somit entstehende F muss schlussendlich zu 0 abgeleitet werden.

(c) Ja, beispielsweise $w' = 1 * 0$.

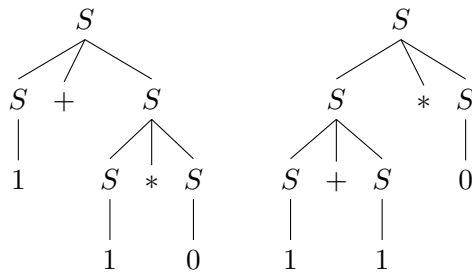
(d)

$$\begin{aligned}
 \left[\begin{array}{c} S \\ | \\ \mathcal{B} \end{array} \right] &= \llbracket \mathcal{B} \rrbracket & \left[\begin{array}{c} S \\ / \quad \backslash \\ \mathcal{B}_1 \quad + \quad \mathcal{B}_2 \end{array} \right] &= \llbracket \mathcal{B}_1 \rrbracket + \llbracket \mathcal{B}_2 \rrbracket & \left[\begin{array}{c} S \\ / \quad \backslash \\ \mathcal{B}_1 \quad * \quad \mathcal{B}_2 \end{array} \right] &= \llbracket \mathcal{B}_1 \rrbracket * \llbracket \mathcal{B}_2 \rrbracket \\
 \left[\begin{array}{c} S \\ / \quad \backslash \quad / \quad \backslash \\ \text{if } \mathcal{B}_1 = \mathcal{B}_2 \text{ then } \mathcal{B}_3 \text{ else } \mathcal{B}_4 \end{array} \right] &= \begin{cases} \llbracket \mathcal{B}_3 \rrbracket, & \text{falls } \llbracket \mathcal{B}_1 \rrbracket = \llbracket \mathcal{B}_2 \rrbracket \\ \llbracket \mathcal{B}_4 \rrbracket, & \text{sonst} \end{cases} \\
 \llbracket 0 \rrbracket &= 0 & \llbracket 1 \rrbracket &= 1
 \end{aligned}$$

(e)

$$\begin{aligned}
 & \left[\begin{array}{c} S \\ / \quad \backslash \\ S \quad + \quad S \\ | \quad / \quad \backslash \quad / \quad \backslash \\ 1 \quad \text{if } S = S \text{ then } S \text{ else } S \\ \quad \quad | \quad | \quad | \quad | \\ \quad \quad 1 \quad 0 \quad 0 \quad 1 \end{array} \right] \\
 &= \left[\begin{array}{c} S \\ | \\ 1 \end{array} \right] + \left[\begin{array}{c} S \\ / \quad \backslash \quad / \quad \backslash \\ \text{if } S = S \text{ then } S \text{ else } S \\ \quad \quad | \quad | \quad | \quad | \\ \quad \quad 1 \quad 0 \quad 0 \quad 1 \end{array} \right] \\
 &= \llbracket 1 \rrbracket + \left[\begin{array}{c} S \\ / \quad \backslash \quad / \quad \backslash \\ \text{if } S = S \text{ then } S \text{ else } S \\ \quad \quad | \quad | \quad | \quad | \\ \quad \quad 1 \quad 0 \quad 0 \quad 1 \end{array} \right] \\
 &= 1 + \left[\begin{array}{c} S \\ / \quad \backslash \quad / \quad \backslash \\ \text{if } S = S \text{ then } S \text{ else } S \\ \quad \quad | \quad | \quad | \quad | \\ \quad \quad 1 \quad 0 \quad 0 \quad 1 \end{array} \right] \\
 & \left(\text{Da } \left[\begin{array}{c} S \\ | \\ 1 \end{array} \right] = \llbracket 1 \rrbracket = 1 \neq \left[\begin{array}{c} S \\ | \\ 0 \end{array} \right] = \llbracket 0 \rrbracket = 0 \right) 1 + \left[\begin{array}{c} S \\ | \\ 1 \end{array} \right] = 1 + \llbracket 1 \rrbracket = 1 + 1 = 2
 \end{aligned}$$

(f) Ja, beispielsweise $w = 1 + 1 * 0$. Zwei Syntaxbäume:



Dabei ist die Semantik des linken Baums 1 und die des rechtens 0.

Übungsaufgabe Ü6.4. (Pumping-Lemma für Kontextfreie Sprachen)

Zeigen Sie unter Verwendung des Pumping Lemmas für kontextfreie Sprachen, dass die folgende Sprache nicht kontextfrei ist:

$$L := \{a^n b^m c^l \mid n, m, l \in \mathbb{N} \wedge n > m \wedge n > l\}$$

Lösungsskizze.

- Angenommen, L wäre kontextfrei. Dann können wir das Pumping-Lemma für kontextfreie Sprachen anwenden.
- Sei $n \in \mathbb{N}_+$ eine Pumping-Lemma-Zahl für L .
- Es gilt $z := a^{n+1} b^n c^n \in L$ und $|z| \geq n$.
- Es gibt also für z eine Zerlegung $z = uvwxy$ mit den Eigenschaften

$$(1) \quad vx \neq \varepsilon$$

$$(2) \quad |vwx| \leq n$$

$$(3) \quad \forall i \in \mathbb{N}. uv^i wx^i y \in L$$

- Wir unterscheiden die folgenden Fälle:
 - $|vx|_a > 0$: Aus (2) und der Wahl von z folgt, dass $|vwx|_c = 0$. Folglich liegt das Wort uv^0wx^0y nicht mehr in L , da es höchstens so viele a 's wie c 's enthält. Dies ist ein Widerspruch zu (3).
 - $|vx|_a = 0$: In diesem Fall muss wegen (1) $|vx|_b > 0$ oder $|vx|_c > 0$ gelten. Folglich liegt das Wort uv^3wx^3y nicht mehr in L , da es entweder mehr b 's oder mehr c 's als a 's enthält. Dies ist ein Widerspruch zu (3).
- Jeder Fall führt zu einem Widerspruch. Somit ist L nicht kontextfrei.

Übungsaufgabe Ü6.5. (Chomsky-Normalform)

Betrachten Sie die folgende Grammatik G :

$$S \rightarrow ASA \mid aB \quad A \rightarrow B \mid S \quad B \rightarrow b \mid \varepsilon$$

- (a) Überführen Sie die Grammatik in Chomsky-Normalform.

- (b) Erklären Sie in eigenen Worten, wie Sie nach Überführen einer Grammatik in CNF überprüfen können, dass Sie keine Fehler gemacht haben.¹

Lösungsskizze.

- (a) Wir überführen die Grammatik schrittweise in Chomsky-Normalform. Das Vorgehensweise für ε -Produktionen und Kettenproduktionen (Schritte 3 und 4) unterscheidet sich leicht von der Vorlesung führt aber zum selben Ergebnis. Die hier beschriebene Vorgehensweise ist allerdings leichter durchzuführen ist und hilft dabei, Fehler zu vermeiden.

- (1) *Entfernen von Terminalen in langen Produktionen.* In jeder Regel $X \rightarrow \gamma$ mit $|\gamma| \geq 2$ Ersetzen jedes Vorkommen eines Terminals x durch X_x und Ergänzen der benötigten Produktionen $X_x \rightarrow x$:

$$G': S \rightarrow ASA \mid X_a B \quad A \rightarrow B \mid S \quad B \rightarrow b \mid \varepsilon \quad X_a \rightarrow a$$

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

- (2) *Entfernen langer Produktionen.* 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 $X_{YZ} \rightarrow YZ$ zugewiesen bekommt. Damit:

$$\begin{aligned} S &\rightarrow AX_{SA} \mid X_a B \\ A &\rightarrow B \mid S \\ G'' : B &\rightarrow b \mid \varepsilon \\ X_a &\rightarrow a \\ X_{SA} &\rightarrow SA \end{aligned}$$

- (3) *Entfernen von ε -Produktionen.*

- Berechnen alle Variablen, die ε erzeugen können. Zuerst sind das alle Variablen X mit ε -Produktion ($X \rightarrow \varepsilon$). Dann kommen iterativ alle Variablen dazu, die eine Produktionen haben, deren rechte Seite nun ε werden kann. Das machen wir solange, bis wir keine neuen Variablen mehr finden. Formal:

$$\begin{aligned} E_0 &:= \{X \mid (X \rightarrow \varepsilon) \in P\} \\ E_{k+1} &:= E_k \cup \{X \mid (X \rightarrow \gamma) \in P \text{ und } \gamma \in E_k^*\} \quad \text{bis } E_{k+1} = E_k. \end{aligned}$$

In unserem Fall:

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

Hier erkennt man, ob die Grammatik ε erzeugt: $\varepsilon \in L(G) \leftrightarrow S \in E_*$.

¹Hier geht es nicht um einen formalen Beweis, dass die beiden Sprachen gleich sind, sondern um eine Strategie, wie Sie bei Aufgaben wie im vorherigen Aufgabenteil Fehler vermeiden.

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

$$\begin{aligned}
S \rightarrow AX_{SA} &\rightsquigarrow S \rightarrow AX_{SA} \mid X_{SA} \\
S \rightarrow X_a B &\rightsquigarrow S \rightarrow X_a B \mid X_a \\
A \rightarrow B &\rightsquigarrow A \rightarrow B \mid \varepsilon \\
A \rightarrow S &\rightsquigarrow A \rightarrow S \\
B \rightarrow b &\rightsquigarrow B \rightarrow b \\
B \rightarrow \varepsilon &\rightsquigarrow B \rightarrow \varepsilon \\
X_a \rightarrow a &\rightsquigarrow X_a \rightarrow a \\
X_{SA} \rightarrow SA &\rightsquigarrow X_{SA} \rightarrow SA \mid S
\end{aligned}$$

- Entfernen aller ε -Produktionen:

$$\begin{aligned}
&S \rightarrow AX_{SA} \mid X_{SA} \mid X_a B \mid X_a \\
&A \rightarrow B \mid S \\
G''' : &B \rightarrow b \\
&X_a \rightarrow a \\
&X_{SA} \rightarrow SA \mid S
\end{aligned}$$

(4) *Entfernen von Kettenproduktionen.*

- Wir berechnen, welche Variablen mit Hilfe von Kettenproduktionen in welche anderen Variablen umgewandelt werden können. Wir beginnen mit den vorhandenen Kettenproduktionen und fügen dann iterative die transitiven Kettenproduktionen hinzu. Formal:

$$\begin{aligned}
T_0 &:= \{(X \rightarrow Y) \in P \mid X, Y \in V\} \\
T_{k+1} &:= T_k \cup \{(X \rightarrow Y) \mid X, Y, Z \in V \wedge (X \rightarrow Z) \in T_k \wedge (Z \rightarrow Y) \in T_k\} \\
&\text{bis } T_{k+1} = T_k.
\end{aligned}$$

In unserem Fall:

$$\begin{aligned}
T_0 &= \{(S \rightarrow X_{SA}), (S \rightarrow X_a), (A \rightarrow B) \\
&\quad, (A \rightarrow S), (X_{SA} \rightarrow S)\} \\
T_1 &= T_0 \cup \{(S \rightarrow S), (A \rightarrow X_{SA}), (A \rightarrow X_a) \\
&\quad, (X_{SA} \rightarrow X_{SA}), (X_{SA} \rightarrow X_a)\} = T_2
\end{aligned}$$

- Entfernen der Kettenproduktionen:

$$\begin{aligned}
&S \rightarrow AX_{SA} \mid X_a B \\
&B \rightarrow b \\
&X_a \rightarrow a \\
&X_{SA} \rightarrow SA
\end{aligned}$$

- Wenn wir X mit Hilfe der entfernten Kettenproduktionen in Y umwandeln konnten, dann müssen wir X ermöglichen alle Produktionen von Y

durchzuführen. Formal:

$$G'''' : \begin{array}{l} S \rightarrow AX_{SA} \mid SA \mid X_a B \mid a \\ A \rightarrow b \mid AX_{SA} \mid SA \mid X_a B \mid a \\ B \rightarrow b \\ X_a \rightarrow a \\ X_{SA} \rightarrow SA \mid AX_{SA} \mid SA \mid X_a B \mid a \end{array}$$

- (b)
- Prüfen, dass alle Produktionen die Form $A \rightarrow BC$ oder $A \rightarrow a$ haben.
 - Insbesondere darf ε nur vom Startsymbol (oder gar nicht, wie hier) produziert werden.
 - Für einige kleine/leichte Worte überprüfen, dass Sie von beiden Grammatiken (nicht) erzeugt werden können.

Zusätzliche Übungsaufgabe Ü6.6. (Fitnessstudio)

Diese Aufgabe können Sie zusätzlich als Nachbereitung lösen. Es wird kein neuer Inhalt in dieser Aufgabe behandelt.

Wir betrachten Pfeil-Sprachen über dem Alphabet $\Sigma = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$. Wir interpretieren dabei ein Wort $w \in \Sigma^*$ als einen Pfad in einem 2D-Gitter.

- (a) Geben Sie eine formale Definition für die folgenden beiden Sprachen an:
- (1) Pfade, die in den Ursprung zurückkehren.
 - (2) Pfade, die „in großer Kurve umkehren“, d.h. *beliebig weit* nach rechts fahren, dann *noch weiter* entweder nach oben oder unten gehen und letztlich wieder umkehren und *noch weiter* nach links fahren. Diese Pfade enden dann links vom Ursprung.
- (b) Zeigen Sie mit Hilfe des Pumping-Lemma für kontextfreie Sprachen, dass diese Sprachen nicht kontextfrei sind.

Lösungsskizze. Für diese Aufgabe gibt es eine [Video-Lösung](#).

- (a) (1)

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

- (2)

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

- (b) (1)
- 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 gilt $z := \rightarrow^n \uparrow^n \leftarrow^n \downarrow^n \in L_a$ 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:

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

- Aufgrund von (1) unterscheiden wir die folgenden Fälle:
 - $|vx|_{\rightarrow} > 0$: Wegen (2) gilt $|vx|_{\leftarrow} = 0$. Allerdings gilt auch:

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

Dies ist ein Widerspruch zu (3).

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

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

Dies ist ein Widerspruch zu (3).

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

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

Dies ist ein Widerspruch zu (3).

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

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

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_a nicht kontextfrei.

- (2) • 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 .
- Dann gilt $z := \rightarrow^n \uparrow^{n+1} \leftarrow^{n+2} \in L_b$ 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:

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

- 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 $|vx|_{\rightarrow} > 0$, dann können wir die Anzahl der \rightarrow über die Anzahl der \leftarrow pumpen, enthält vx keinen \rightarrow aber mindestens ein \uparrow , so kann man die Anzahl der \rightarrow auf höchstens n reduzieren, indem man vx entfernt. Andernfalls besteht vx 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:

$$|uv^3wx^3y|_{\rightarrow} = n + 2|vx|_{\rightarrow} \geq n + 2 = |uv^3wx^3y|_{\leftarrow}$$

Dies ist ein Widerspruch zu (3).

– $|vx|_{\rightarrow} = 0$ und $|vx|_{\uparrow} > 0$: Dann gilt:

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

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

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

$$|uv^0wx^0y|_{\leftarrow} = n + 2 - |vx|_{\leftarrow} \leq n + 1 = |uv^0wx^0y|_{\uparrow}$$

Daher ist $uv^0wx^0y \notin L_b$, 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.