Formale Grundlagen der Informatik I
5. Übungsblatt

Fachbereich Mathematik
Prof. Dr. Martin Ziegler
Davorin Lešnik, Ph.D.
Carsten Rösnick

Sommersemester 2013
13. 05. 2013

Gruppenübung

Aufgabe G10 (Kontextfreie Grammatik)
Betrachten Sie die kontextfreie Grammatik $G = \big(\{a, b\}, \{X_0, X, Y\}, P, X_0\big)$ mit

$$\begin{array}{rlcl} P: & X_0 & \rightarrow & aXaY \\ & X & \rightarrow & aXa \mid Y \\ & Y & \rightarrow & bY \mid \varepsilon. \end{array}$$

(a) Bestimmen Sie die von G erzeugte Sprache.

(b) Bestimmen Sie eine zu G äquivalente Grammatik G' ohne ε-Produktionen.

Lösung:

(a) $\mathrm{L}(G) = \left\{a^n b^m a^n b^k \;\middle|\; n, m, k \in \mathbb{N},\ n \geq 1\right\}$

(b) Die Menge der Variablen Z, die nach endlich vielen Ableitungsschritten das leere Wort produzieren, ist $V_\varepsilon = \{X, Y\}$ (vgl. auch Beweis von Lemma 2.4.3). Wir entfernen $Y \rightarrow \varepsilon$ und fügen dafür die folgenden Produktionen zu G hinzu:

$$\begin{array}{lcl} X_0 & \rightarrow & aXa \\ X & \rightarrow & \varepsilon \\ Y & \rightarrow & b. \end{array}$$

Weiterhin entfernen wir $X \rightarrow \varepsilon$ und fügen auch die nachfolgenden Produktionen zu G hinzu (dabei die eben neu zu G hinzugefügten Produktionen nicht vergessen!):

$$\begin{array}{lcl} X_0 & \rightarrow & aaY \\ X & \rightarrow & aa \\ X_0 & \rightarrow & aa. \end{array}$$

Nach dem Entfernen der ε-Produktionen erhalten wir insgesamt:

$$\begin{array}{lcl} X_0 & \rightarrow & aXaY \mid aXa \mid aaY \mid aa \\ X & \rightarrow & aXa \mid Y \mid aa \\ Y & \rightarrow & bY \mid b. \end{array}$$

Aufgabe G11 (Chomsky-Normalform, CYK-Algorithmus)
Betrachten Sie die kontextfreie Grammatik $G = (\{a, b\}, \{X_0, X, Y\}, P, X_0)$ mit

$$\begin{array}{rlcl} P: & X_0 & \to & aXaY \mid aXa \mid aaY \mid aa \\ & X & \to & aXa \mid Y \mid aa \\ & Y & \to & bY \mid b. \end{array}$$

(a) Konstruieren Sie eine Grammatik G' in Chomsky-Normalform mit $\mathrm{L}(G) = \mathrm{L}(G')$.

(b) Entscheiden Sie mit Hilfe des CYK-Algorithmus, ob das Wort $w = aabaa$ in der von G' erzeugten Sprache $\mathrm{L}(G')$ liegt. Ist w in $\mathrm{L}(G')$ enthalten, so geben Sie zudem eine Ableitung in G' an.

Lösung:

(a) Erstens fügen wir Z_a und Z_b hinzu und

$$\begin{array}{lcl} X_0 & \to & Z_aXZ_aY \mid Z_aXZ_a \mid Z_aZ_aY \mid Z_aZ_a \\ X & \to & Z_aXZ_a \mid Y \mid Z_aZ_a \\ Y & \to & Z_bY \mid Z_b \\ Z_a & \to & a \\ Z_b & \to & b. \end{array}$$

Dann eliminieren wir die $X \to Y$ Produktionen:

$$\begin{array}{lcl} X_0 & \to & Z_aXZ_aY \mid Z_aXZ_a \mid Z_aZ_aY \mid Z_aZ_a \\ X & \to & Z_aXZ_a \mid Z_bY \mid b \mid Z_aZ_a \\ Y & \to & Z_bY \mid b \\ Z_a & \to & a \\ Z_b & \to & b. \end{array}$$

Letztens eliminieren wir die $X \to X_1 \dots X_n$ Produktionen mit $n \geq 3$: wir fügen Z_1 hinzu und ersetzen $X_0 \to Z_aXZ_aY$ durch $X_0 \to Z_aXZ_1$ und $Z_1 \to Z_aY$, dann fügen wir Z_2 hinzu und ersetzen $X_0 \to Z_aXZ_1$ durch $X_0 \to Z_aZ_2$ und $Z_2 \to XZ_1$, fügen Z_3 hinzu und ersetzen $X_0 \to Z_aXZ_a$ durch $X_0 \to Z_aZ_3$ und $Z_3 \to XZ_a$, fügen Z_4 hinzu und ersetzen $X_0 \to Z_aZ_aY$ durch $X_0 \to Z_aZ_4$ und $Z_4 \to Z_aY$, und schließlich fügen Z_5 hinzu und ersetzen $X \to Z_aXZ_a$ durch $X \to Z_aZ_5$ und $Z_5 \to XZ_a$. Insgesamt:

$$\begin{array}{lcl@{\qquad}lcl} X_0 & \to & Z_aZ_2 \mid Z_aZ_3 \mid Z_aZ_4 \mid Z_aZ_a & Z_3 & \to & XZ_a \\ X & \to & Z_aZ_5 \mid Z_bY \mid b \mid Z_aZ_a & Z_4 & \to & Z_aY \\ Y & \to & Z_bY \mid b & Z_5 & \to & XZ_a \\ Z_1 & \to & Z_aY & Z_a & \to & a \\ Z_2 & \to & XZ_1 & Z_b & \to & b. \end{array}$$

(b) Der CYK-Algorithmus ergibt folgende Tabelle:

w	a	a	b	a	a
$j\backslash i$	1	2	3	4	5
1	$\{Z_a\}$				
2	$\{X_0, X\}$	$\{Z_a\}$			
3	$\{X_0\}$	$\{Z_1, Z_4\}$	$\{X, Y, Z_b\}$		
4	$\emptyset$	$\{X_0, X\}$	$\{Z_3, Z_5\}$	$\{Z_a\}$	
5	$\{X_0, X\}$	$\{Z_3, Z_5\}$	$\emptyset$	$\{X_0, X\}$	$\{Z_a\}$

Damit folgt $w = aabaa \in \mathrm{L}(G')$, denn das Startsymbol X_0 ist Element der Menge des Eintrags $(i, j) = (1, 5)$. Dabei leitet sich w bspw. wie folgt ab:

$$X_0 \to_{G'} Z_aZ_3 \to_{G'} aXZ_a \to_{G'} aZ_aZ_5a \to_{G'} aaXZ_aa \to_{G'} aabaa = w$$

Aufgabe G12 (Kontextfreie Grammatik)

(a) Welche Sprache beschreibt die Grammatik $G = (\Sigma, V, P, S)$ mit $\Sigma := \{a, b, c, \$\}$, $V := \{S, A, B, X, Y\}$ und Produktionen

$$\begin{array}{ll} S \to AB & Xa \to aX \\ A \to aXA \mid \$ & X\$ \to \$Y \\ B \to bYB \mid \$ & Yb \to bY \\ & Y\$ \to \$c \end{array}$$

Beweisen Sie ihre Behauptung durch Induktion über die Menge der in G ableitbaren $(V \cup \Sigma)$-Wörter. Stellen Sie hierzu zunächst eine geeignete Induktionsbehauptung auf.

(b) Geben Sie eine kontextfreie Grammatik für die Sprache aus (a) an, d. h. eine Grammatik, bei welcher die linke Seite jeder Produktion aus einer einzigen Variablen besteht.

Lösung:

(a) Wir behaupten: $\mathrm{L}(G) = \left\{ a^k \$ b^n \$ c^{k+n} \;\middle|\; k, n \in \mathbb{N} \right\}$.

($\supseteq$) Jedes Wort der Form $w := a^k \$ b^n \$ c^{k+n}$ kann in G abgeleitet werden:

$$\begin{aligned} S &\to_G AB \to_G aXAB \to_G aXaXAB \to_G \cdots \to_G (aX)^k AB \to_G (aX)^k \$B \\ &\to_G (aX)^k \$bYB \to_G \cdots \to_G (aX)^k \$(bY)^n B \to_G (aX)^k \$(bY)^n \$ \\ &\to_G \cdots \to_G a^k X^k \$(bY)^n \$ \to_G \cdots \to_G a^k \$ Y^k (bY)^n \$ \\ &\to_G \cdots \to_G a^k \$ b^n Y^{k+n} \$ \to_G \cdots \to_G a^k \$ b^n \$ c^{k+n} \end{aligned}$$

($\subseteq$) Wir beweisen die folgenden Aussagen durch Induktion über den Ableitungsprozess.

i. Gilt $S \to_G^* w$, dann ist $w = S$ oder w ist von der Form $w = uxvyz$ für geeigenete Wörter

$$u \in \{a, X\}^*, \quad v \in \{b, Y\}^*, \quad z \in \{c\}^*, \quad x \in \{A, \$\} \quad \text{und} \quad y \in \{B, \$\}\,.$$

ii. Gilt $S \to_G^* w$, so ist $|w|_a + |w|_b = |w|_X + |w|_Y + |w|_c$.

Induktionsanfang: Für $w = S$ sind die Aussagen offensichtlich erfüllt.

Induktionsschritt: Angenommen die Ableitung ist $S \to_G^* w' \to_G w$. Falls $w' = S$, dann ist $w = AB$ und (1) und (2) erfüllt. Wir nehmen deshalb an, dass $w' \neq S$. Nach Induktionshypothese erfüllt w' die Bedingungen (2) und es gibt eine Zerlegung $w' = u'x'v'y'z'$, wie unter (1). Für jede mögliche Produktion, die von w' nach w führen könnte, müssen wir untersuchen, ob sie die Eigenschaften (1) und (2) erhält. Wir betrachten exemplarisch einige Fälle.

($A \to aXA$) In diesem Fall ist $x' = A$ und $w = u'aXAv'y'z'$. Desweiteren gilt

$$|w|_a + |w|_b = |w'|_a + |w'|_b + 1 = |w'|_X + |w'|_Y + |w'|_c + 1 = |w|_X + |w|_Y + |w|_c\,.$$

($Yb \to bY$) In diesem Fall ist $w = u'x'vy'z'$, wobei das Wort v aus v' entsteht, indem zwei Buchstaben vertauscht wurden. An der Anzahl der verschiedenen Buchstaben ändert sich nichts.

($Y\$ \to \$c$) Dann ist $w = u'x'v\$cz'$ mit $v' = vY$ und $y' = \$$. Es verringert sich die Anzahl der Y um 1. Dafür steigt die Zahl der c.

Die verbleibenden Fälle zeigt man analog.

(b)

$$\begin{aligned} S &\to aSc \mid \$X \\ X &\to bXc \mid \$ \end{aligned}$$

Hausübung

Aufgabe H10 (Pumping Lemma vs. Myhill-Nerode) (3 Punkte)
Sei $\Sigma = \{a, b\}$ und $L = \left\{ p\, q\, q^{-1} \;\middle|\; p, q \in \Sigma^+ \right\}$ (für das Symbol $^{-1}$ siehe Übung 2.5.4 im Skript).

(a) Zeigen Sie, dass L die Aussage des Pumping-Lemmas erfüllt.

(b) Zeigen Sie, dass L trotzdem nicht regulär ist.
Tipp: Benutzen Sie den Satz von Myhill-Nerode!

Lösung:

(a) $\boxed{\text{1,5 P.}}$ Wir zeigen, dass L die Behauptung des Pumping-Lemmas für $n = 4$ erfüllt. Dafür sei $x = p\,q\,q^{-1}$ ein Wort, das zu L gehört und mindestens die Länge 4 hat. Es gibt zwei Möglichkeiten:

i. Der Teil hat p mindestens die Länge 2. In diesem Fall wählen wir für u den ersten Buchstaben von p, für v den zweiten Buchstaben von p und für w den Rest von x. Dann ist es klar $uv^m w \in L$ für alle $m \in \mathbb{N}$, denn w und somit $uv^m w$ endet sich mit pp^{-1} und die Länge des Wortes davor ist $m + |p| - 1 \geq 1$ (die Annahme $|p| \geq 2$ brauchen wir für den Fall $m = 0$).

ii. Der Teil p hat die Länge 1. In diesem Fall hat q mindestens die Länge 2 (weil $|x| \geq 4$). Sei u alles vor dem letzten Buchstaben von p, v dieser letzte Buchstabe, zusammen mit dem (gleichen) ersten Buchstaben von q^{-1} und w der Rest von q^{-1}. Es ist wieder klar, dass das funktioniert.

(b) $\boxed{\text{1,5 P.}}$ Aus dem Satz von Myhill-Nerode folgt, dass es reicht, eine unendliche Menge von Wörtern zu finden, die bezüglich der Äquivalenzrelation $\sim_L$ paarweise nicht äquivalent sind. Betrachte die Wörter $\left\{ aba^{2n+1}b \;\middle|\; n \in \mathbb{N} \right\}$. Für $m, n \in \mathbb{N}$, $m \neq n$, gilt $aba^{2m+1}b \nsim_L aba^{2n+1}b$, denn $aba^{2m+1}bw \in L$ und $aba^{2n+1}bw \notin L$ für $w := ba^{2m+1}b$.

Aufgabe H11 (Chomsky-Normalform, CYK-Algorithmus) (3,5 Punkte)
Gegeben sei die Grammatik $G = \big(\{.\} \cup \{\mathtt{0}, 1, 2, \ldots, 9\}, \{S, F, X\}, P, S\big)$,

$$
\begin{aligned}
P:\quad & S \to X\ .\ F \mid X \\
& X \to \mathtt{0} \mid i\ F \qquad (i = 1, 2, \ldots, 9) \\
& F \to j\ F \mid j \qquad (j = \mathtt{0}, 1, 2, \ldots, 9)
\end{aligned}
$$

(a) Beschreiben Sie umgangssprachlich, welche Sprache von der Grammatik G akzeptiert wird.

(b) Überführen Sie G in eine äquivalente Grammatik G' in Chomsky-Normalform.

(c) Entscheiden Sie durch Ausführung des CYK-Algorithmus, ob das Wort $w = \mathtt{20.540}$ in der von G' erzeugten Sprache $\mathrm{L}\left(G'\right)$ liegt. (Der CYK-Algorithmus sei dabei wie in der Übung in einer $|w| \times |w|$-Tabelle durchzuführen.) Ist w in $\mathrm{L}\left(G'\right)$ enthalten, so geben Sie zudem eine Ableitung in G' an.

Lösung:

(a) $\boxed{\text{0,5 P.}}$ Die Menge aller Zeichenketten, die <u>endliche</u> Dezimalzahlen darstellen. Wie üblich erlaubt man keine führenden Nullen, außer wenn Null die einzige Ziffer (oder die einzige Ziffer vor dem Dezimalpunkt) ist (aber wir erlauben Nullen am Ende des Nachpunktteils).

(b) i. $\boxed{\text{☺ P.}}$ Ersetze ε-Produktionen: ✓

ii. $\boxed{\text{0,5 P.}}$ Ersetze nicht alleinstehende Terminale a durch eine Variable Z_a und füge die Produktion $Z_a \to a$ hinzu:

$$
\begin{aligned}
P:\quad & S \to X\ Z_{\bullet}\ F \mid X \\
& X \to \mathtt{0} \mid Z_i\ F \qquad (i = 1, 2, \ldots, 9) \\
& F \to Z_j\ F \mid k \qquad (j = 0, 1, 2, \ldots, 9; \quad k = \mathtt{0}, 1, 2, \ldots, 9) \\
& Z_{\bullet} \to . \\
& Z_\ell \to \ell \qquad (\ell = \mathtt{0}, 1, 2, \ldots, 9)
\end{aligned}
$$

iii. $\boxed{0{,}5\text{ P.}}$ Löse Kettenproduktionen auf, d. h. Produktionen der Form $A \to B$:

$$\begin{aligned} P: \quad S &\to X\ Z_\bullet\ F \mid \mathtt{0} \mid Z_i\ F \qquad (i = 1, 2, \dots, 9)\\ X &\to \mathtt{0} \mid Z_i\ F \qquad (i = 1, 2, \dots, 9)\\ F &\to Z_j\ F \mid k \qquad (j = 0, 1, 2, \dots, 9; \quad k = \mathtt{0}, 1, 2, \dots, 9)\\ Z_\bullet &\to .\\ Z_\ell &\to \ell \qquad (\ell = \mathtt{0}, 1, 2, \dots, 9) \end{aligned}$$

iv. $\boxed{0{,}5\text{ P.}}$ Teile Produktionen der Form $A \to B_1 \dots B_k$ mit ≥ 3 auf:

$$\begin{aligned} P: \quad S &\to X\ Z_{ZF} \mid \mathtt{0} \mid Z_i\ F \qquad (i = 1, 2, \dots, 9)\\ X &\to \mathtt{0} \mid Z_i\ F \qquad (i = 1, 2, \dots, 9)\\ F &\to Z_j\ F \mid k \qquad (j = 0, 1, 2, \dots, 9; \quad k = \mathtt{0}, 1, 2, \dots, 9)\\ Z_\bullet &\to .\\ Z_\ell &\to \ell \qquad (\ell = \mathtt{0}, 1, 2, \dots, 9)\\ Z_{ZF} &\to Z_\bullet\ F \end{aligned}$$

(c) $\boxed{1\text{ P.}}$

w	2	$\mathtt{0}$	.	5	4	$\mathtt{0}$
$j\backslash i$	1	2	3	4	5	6
1	$\{F, Z_2\}$					
2	$\{S, X, F\}$	$\{S, X, F, Z_0\}$				
3	$\emptyset$	$\emptyset$	$\{Z_\bullet\}$			
4	$\{S\}$	$\{S\}$	$\{Z_{ZF}\}$	$\{F, Z_5\}$		
5	$\{S\}$	$\{S\}$	$\{Z_{ZF}\}$	$\{S, X, F\}$	$\{F, Z_4\}$	
6	$\{S\}$	$\{S\}$	$\{Z_{ZF}\}$	$\{S, X, F\}$	$\{S, X, F\}$	$\{S, X, F, Z_0\}$

$\boxed{0{,}5\text{ P.}}$ Sind $u_{i,j}$ die Teilworte wie im Skript, Lemma 3.3.13 beschrieben, dann ergibt sich die Ableitung von $w = \mathtt{20.540}$ in G' durch

$$w = u_{1,2}u_{3,6} = (u_{1,1}u_{2,2})(u_{3,3}u_{4,6}) = (u_{1,1}u_{2,2})\big(u_{3,3}(u_{4,4}u_{5,6})\big)$$

wie folgt:

$$S \to^*_{G'} X\ Z_{ZF} \to^*_{G'} Z_2\ F\ \ Z_\bullet\ F \to^*_{G'} \mathtt{2\ 0\ .}\ \ Z_5\ F \to^*_{G'} \mathtt{2\ 0\ .\ 5\ 4\ 0}$$

Aufgabe H12 (Chomsky-Typ einer Sprache) (3,5 Punkte)

In der (fiktiven) Programmiersprache TUDL gibt es genau zwei Befehle: <u>*produce*</u> □ und <u>*consume*</u> □, wobei □ durch einen Variablennamen zu ersetzen ist. Der Einfachheit halber beschränken wir uns hier auf zwei Variablennamen: x und y. Unser Alphabet ist daher die Menge $\Sigma := \{x, y, \underline{produce}, \underline{consume}\}$. Ein gültiges TUDL-Programm erfüllt die folgenden beiden Bedingungen:

- Bevor eine Variable □ konsumiert werden kann, muss sie produziert worden sein; d. h. jedem <u>*consume*</u> □ muss ein <u>*produce*</u> □ vorausgegangen sein.
- Am Ende wollen wir keine Reste haben, d. h. die Anzahl der <u>*produce*</u> □ muss mit der Anzahl der <u>*consume*</u> □ übereinstimmen.

(a) Geben Sie eine Grammatik für die Sprache TUDL an.

(b) Beweisen oder widerlegen Sie: Die Sprache TUDL ist kontextfrei.

Lösung:

(a) $\boxed{2\ \text{P.}}$ Wir geben eine kontext-*sensitive* Grammatik in BNF (Skript, Seite 48) an: $G = (\Sigma, V, P, L)$ mit $V = \{L, P_x, P_y, C_x, C_y\}$ und

$$
\begin{aligned}
P: \quad L &\to [L]\ P_i\ [L]\ C_i\ [L] && (i = x, y) \\
C_i\ C_j &\to C_j\ C_i && (\{i,j\} = \{x,y\}) \\
P_i\ P_j &\to P_j\ P_i && (\{i,j\} = \{x,y\}) \\
P_i\ C_j &\to C_j\ P_i && (\{i,j\} = \{x,y\}) \\
P_i &\to \underline{produce}\ i && (i = x, y) \\
C_i &\to \underline{consume}\ i && (i = x, y)
\end{aligned}
$$

(b) $\boxed{1{,}5\ \text{P.}}$ Die Sprache TUDL ist kontextsensitiv, aber *nicht* kontextfrei. Der Einfachheit halber definieren wir $a := \underline{produce}\ x$, $b := \underline{produce}\ y$, $c := \underline{consume}\ x$ und $d := \underline{consume}\ y$, damit wir im Folgenden die gleichen Bezeichnungen für Zerlegungen von Worten wie im Skript wählen können.

Sei nun $n \in \mathbb{N}$, $x = a^n b^n c^n d^n$ und $x = y \cdot u \cdot v \cdot w \cdot z$ eine beliebige Zerlegung des Wortes x, wobei $|u \cdot w| > 0$ und $|u \cdot v \cdot w| \leq n$.

i. Ist $uvw = a^k$ mit $k \leq n$, so ist $y \cdot u^0 \cdot v \cdot w^0 \cdot z = y \cdot v \cdot z \notin$ TUDL.

ii. Ist $uvw = a^i \cdot (a^j b^k) \cdot b^\ell$ mit $i + j + k + \ell \leq n$, so enthält das Wort $u^0 \cdot v \cdot w^0$ nur noch j-viele a oder k-viele b. Wegen $|u \cdot w| > 0$ ist $i + \ell > 0$, womit schlussendlich $y \cdot u^0 \cdot v \cdot w^0 \cdot z = y \cdot v \cdot z = a^{n-i} b^{n-\ell} c^n d^n \notin$ TUDL folgt.

iii. Die Fälle 1. und 2. wiederholen sich für die anderen Kombinationen: uvw enthält nur b, sowohl b und c, nur c, sowohl c und d oder nur d.

Minitest

Aufgabe M13

Sei $L \subseteq \Sigma^*$ eine beliebige Sprache. Bestimmen Sie die korrekten Implikationen:

L ist regulär	☐ $\Longrightarrow$ ☐ $\Longleftarrow$	es gilt die Aussage aus dem Pumping-Lemma, d.h. es gibt ein $n \in \mathbb{N}$, so dass sich jedes Wort $x \in L$ mit $|x| \geq n$ in $x = u \cdot v \cdot w$ zerlegen lässt...
	☐ $\Longrightarrow$ ☐ $\Longleftarrow$	die Relation $\sim_L$ hat endlichen Index

Lösung:

L ist regulär	☒ $\Longrightarrow$ ☐ $\Longleftarrow$	es gilt die Aussage aus dem Pumping-Lemma, d.h. es gibt ein $n \in \mathbb{N}$, so dass sich jedes Wort $x \in L$ mit $|x| \geq n$ in $x = u \cdot v \cdot w$ zerlegen lässt...

Begründung: Die Hinrichtung ist genau das Pumping-Lemma für reguläre Sprachen. Die Rückrichtung gilt aber nicht; ein Gegenbeispiel findet man in **Aufgabe H10**.

L ist regulär	☒ $\Longrightarrow$ ☒ $\Longleftarrow$	die Relation $\sim_L$ hat endlichen Index

Begründung: Der Satz von Myhill-Nerode.

Aufgabe M14

Sei $\mathscr{A}$ ein DEA und $\mathscr{B}$ ein NEA. Welche der folgenden Aussagen sind korrekt?

(a) Es gibt einen Algorithmus, der entscheidet, ob $\mathscr{A}$ minimal ist.
☐ wahr ☐ falsch

(b) Es gibt bis auf Isomorphie genau einen minimalen DEA, der die gleiche Sprache wie $\mathscr{A}$ erkennt.
☐ wahr ☐ falsch

(c) Es gibt bis auf Isomorphie genau einen minimalen NEA, der die gleiche Sprache wie $\mathscr{B}$ erkennt.
☐ wahr ☐ falsch

Lösung:

(a) Es gibt einen Algorithmus, der entscheidet, ob $\mathscr{A}$ minimal ist.
☒ wahr ☐ falsch

Begründung: Man wende den Minimierung-Algorithmus auf $\mathscr{A}$ an. Genau dann, wenn $\mathscr{A}$ sich nicht ändert, ist $\mathscr{A}$ minimal.

(b) Es gibt bis auf Isomorphie genau einen minimalen DEA, der die gleiche Sprache wie $\mathscr{A}$ erkennt.
☒ wahr ☐ falsch

Begründung: Es gibt einen eindeutigen Minimalautomaten, siehe Skript 2.4.3.

(c) Es gibt bis auf Isomorphie genau einen minimalen NEA, der die gleiche Sprache wie $\mathscr{B}$ erkennt.
☐ wahr ☒ falsch

Begründung: Die folgenden NEAs beschreiben die gleiche Sprache, sind nicht isomorph und haben die minimale Anzahl von Zuständen.

D. h. $\mathscr{C}$ wäre auch ein „Minimalautomat" von $\mathscr{B}$. Da bei NEAs dieser nicht eindeutig bestimmt ist, spricht man bei einem NEA mit der minimalen Anzahl von Zuständen in der Regel nicht von einem Minimalautomaten.

Aufgabe M15

Betrachten Sie die Grammatiken $G_i = (\{a,b\}, \{S,X,Y\}, P_i, S)$ für $i \in \{1,2\}$ und mit

$$
\begin{array}{llll}
P_1: & S & \rightarrow & XY \\
 & X & \rightarrow & a \mid aX \\
 & Y & \rightarrow & b \mid Yb
\end{array}
\qquad
\begin{array}{llll}
P_2: & S & \rightarrow & XY \\
 & X & \rightarrow & a \mid aX \\
 & Y & \rightarrow & b \mid Yb \\
 & XY & \rightarrow & YX
\end{array}
$$

(a) Welche der folgenden Sprachen beschreibt die Grammatik G_1?
- ☐ $\{a^n b^n \mid n \geq 1\}$
- ☐ $\mathtt{L}(\mathtt{a}^*\mathtt{b}^*)$
- ☐ $\mathtt{L}(\mathtt{aa}^*\mathtt{bb}^*)$
- ☐ $\mathtt{L}((\mathtt{a}+\mathtt{b})^*)$

(b) Welchen Typ haben die Grammatiken G_1 und G_2? (Kreuzen Sie *alle* zutreffenden Antworten an.)
- G_1: ☐ 3 ☐ 2 ☐ 1 ☐ 0
- G_2: ☐ 3 ☐ 2 ☐ 1 ☐ 0

Lösung:

(a) ☐ $\{a^n b^n \mid n \geq 1\}$
☐ $\mathtt{L}(\mathtt{a}^*\mathtt{b}^*)$

☒ $\mathtt{L}(\mathtt{aa}^*\mathtt{bb}^*)$

☐ $\mathtt{L}((\mathtt{a}+\mathtt{b})^*)$

Begründung: Aus X lässt sich genau a^{n+1} und $a^n X$ für beliebiges $n \in \mathbb{N}$ ableiten. Genauso lässt sich aus Y genau b^{m+1} und $Y b^m$ für jedes $m \in \mathbb{N}$ ableiten. Damit kann man aus S genau $a^{n+1}b^{m+1}$, $a^{n+1}Y b^m$, $a^n X b^{m+1}$, $a^n X Y b^m$ für jedes $n, m \in \mathbb{N}$ ableiten. Also

$$\mathtt{L}(G_1) = \left\{ a^{n+1}b^{m+1} \;\middle|\; n, m \in \mathbb{N} \right\} = \mathtt{L}(aa^*bb^*)\,.$$

(b)
- G_1: ☐ 3 ☒ 2 ☒ 1 ☒ 0
- G_2: ☐ 3 ☐ 2 ☒ 1 ☒ 0

Begründung: G_1 ist Typ 2, weil jede Produktion der Form $X \to v$ für eine Variable X und ein v ist. Sie ist nicht Typ 3, weil $X_0 \to XY$ bei regulären Grammatiken nicht erlaubt ist.
G_2 ist Typ 1, weil jede Produktion nicht verkürzend ist. Sie ist nicht Typ 2, weil $XY \to YX$ mehr als eine Variable auf der linken Seite hat.