

**Klausur zur Vorlesung  
Theoretische Grundlagen der Informatik  
Wintersemester 2010/2011**

**Hier Aufkleber mit Name und Matrikelnr. anbringen**

Vorname: \_\_\_\_\_

Nachname: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

**Beachten Sie:**

- Bringen Sie den Aufkleber mit Ihrem Namen und Ihrer Matrikelnummer auf diesem Deckblatt an und beschriften Sie jedes Aufgabenblatt mit Ihrem Namen und Ihrer Matrikelnummer.
- Schreiben Sie die Lösungen auf die Aufgabenblätter und Rückseiten. Zusätzliches Papier erhalten Sie bei Bedarf von der Aufsicht.
- Zum Bestehen der Klausur sind **20** der möglichen **60** Punkte hinreichend.
- Es sind keine Hilfsmittel zugelassen.

Aufgabe	Mögliche Punkte				Erreichte Punkte			
	a	b	c	$\Sigma$	a	b	c	$\Sigma$
1	4	-	-	4		-	-	
2	4	-	-	4		-	-	
3	3	-	-	3		-	-	
4	4	-	-	4		-	-	
5	3	4	-	7			-	
6	6	-	-	6		-	-	
7	4	2	-	6			-	
8	1	2	3	6				
9	3	-	-	3		-	-	
10	2	5	-	7			-	
11	10x1			10				
$\Sigma$				60				

**Aufgabe 1:**

(4 Punkte)

Beweisen Sie, dass die Sprache  $L = \{w_1w_2 \mid w_1 \in \{a,b\}^k, w_2 = c^j, k < j, k, j \in \mathbb{N}_{>0}\}$  über dem Alphabet  $\Sigma = \{a, b, c\}$  nicht regulär ist.

**Lösung:**

Annahme:  $L$  ist regulär. Sei dann  $n$  wie im Pumping-Lemma gefordert, d.h., so dass für jedes Wort  $w \in L$  mit  $|w| > n$  eine Darstellung  $w = uvx$  mit  $|uv| \leq n$ ,  $v \neq \epsilon$ , existiert, bei der auch  $uv^i x \in L$  ist für alle  $i \in \mathbb{N}_0$ . Betrachte das Wort  $w = a^n c^{n+1} \in L$ . Für jede Zerlegung  $w = uvx$  mit  $|uv| \leq n$ ,  $v \neq \epsilon$  ist  $uv^2x = a^{n+\ell} c^{n+1} \notin L$  weil  $\ell > 0$ . Dies ist ein Widerspruch zum Pumping-Lemma.

**Aufgabe 2:**

(4 Punkte)

Gegeben sei das Alphabet  $\Sigma = \{a, b, c, d\}$ .

Definitionen:

- Es sei  $L$  die Sprache aller Wörter über  $\Sigma$  in denen  $a$  nie neben  $b$ ,  $b$  nie neben  $c$  und  $c$  nie neben  $d$  steht, wobei gilt:
- In einem Wort  $w = w_1 \dots w_n$  steht Zeichen  $x$  neben Zeichen  $y$ , wenn es ein  $i \in \{1, \dots, n-1\}$  gibt, so dass  $x = w_i$  und  $y = w_{i+1}$  oder  $y = w_i$  und  $x = w_{i+1}$ .

Entwerfen Sie eine Grammatik, die die Sprache  $L$  erzeugt.

**Lösung:**

Die Grammatik  $G = (\Sigma, V, S, R)$  mit  $V = \{S, A, B, C, D\}$  und Regeln

$$\begin{aligned} S &\rightarrow A|B|C|D|\epsilon \\ A &\rightarrow a|aA|aC|aD \\ B &\rightarrow b|bB|bD \\ C &\rightarrow c|cA|cC \\ D &\rightarrow d|dA|dB|dD \end{aligned}$$

**Aufgabe 3:**

(3 Punkte)

Gegeben sei die Grammatik  $G = (\Sigma, V, S, R)$  mit Terminalalphabet  $\Sigma = \{0, 1\}$ , Variablen  $V = \{S, A, B\}$  und Produktionen

$$R = \{ \begin{array}{l} S \rightarrow 0B|1A, \\ A \rightarrow 0|0S|1AA \\ B \rightarrow 1|1S|0BB \end{array} \} .$$

Welche Sprache erzeugt  $G$ ?

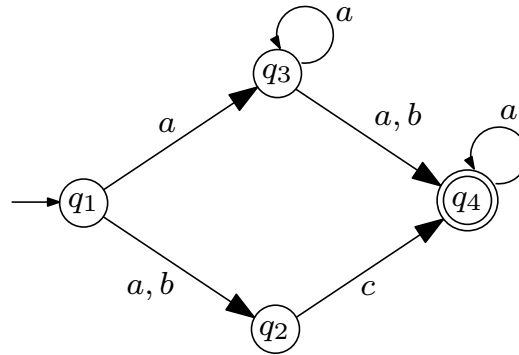
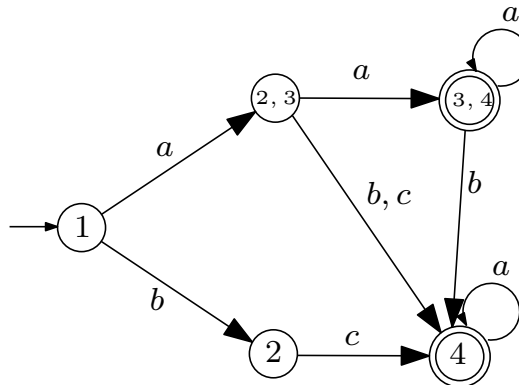
**Lösung:**

Die Sprache aller Wörter über dem Alphabet  $\{0, 1\}$  der Länge mindestens 1, die genauso häufig 1 wie 0 enthalten.

**Aufgabe 4:**

(4 Punkte)

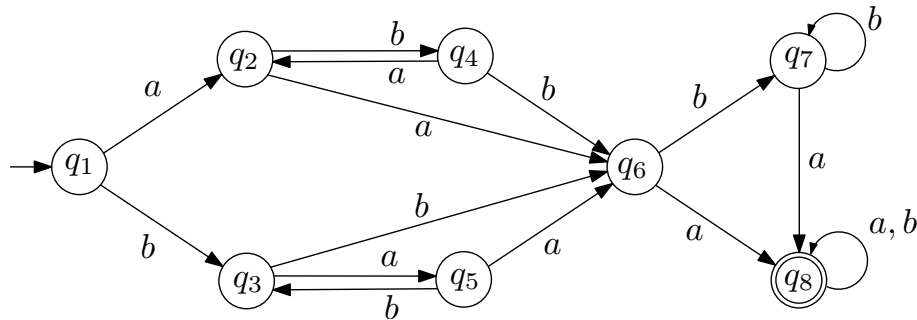
Konstruieren Sie zu folgendem NEA über dem Alphabet  $\Sigma = \{a, b, c\}$  einen DEA, der dieselbe Sprache akzeptiert und zeichnen Sie das entsprechende Zustandsübergangsdiagramm.

**Lösung:**

**Aufgabe 5:**

(3+4 Punkte)

Gegeben ist folgender endlicher Automat  $\mathcal{A}$  über dem Alphabet  $\Sigma = \{a, b\}$ .



- (a) Geben Sie einen regulären Ausdruck für die Sprache an, die  $\mathcal{A}$  akzeptiert. Hinweis: Hier ist nicht verlangt, dass sie das Verfahren aus der Vorlesung benutzen.

**Lösung:**

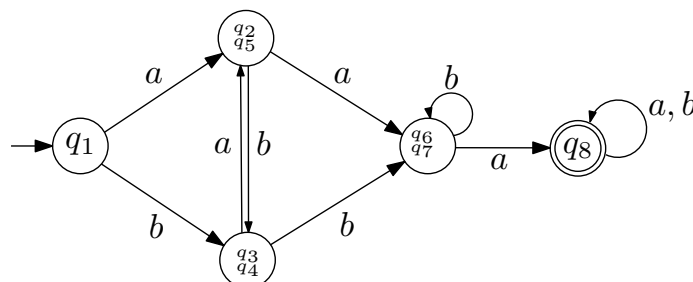
$$(a(ba)^*(a \cup bb) \cup b(ab)^*(b \cup aa))(bb^*a \cup a)(a \cup b)^*$$

- (b) Konstruieren Sie den Minimalautomaten zu  $\mathcal{A}$ . Sie dürfen benutzen, dass der Minimalautomat aus 5 Zuständen besteht.

**Lösung:**

		$\{q_1, \dots, q_8\}$
$\epsilon$	trennt	$\{q_1, \dots, q_7\}, \{q_8\}$
$a$	trennt	$\{q_1, \dots, q_5\}, \{q_6, q_7\}, \{q_8\}$
$b$	trennt	nichts
$aa$	trennt	$\{q_2, q_5\}, \{q_1, q_3, q_4\}, \{q_6, q_7\}, \{q_8\}$
$ba$	trennt	$\{q_1\}, \{q_2, q_5\}, \{q_3, q_4\}, \{q_6, q_7\}, \{q_8\}$

Das sind bereits 5 Zustände.



**Aufgabe 6:**

(6 Punkte)

**Problem SUBSET 3-INTERVALL (S-3INT)**Gegeben: Endliche Menge  $M$ , Gewichtsfunktion  $w : M \rightarrow \mathbb{N}_{>0}$ , Zahl  $k \in \mathbb{N}_{>0}$ .Frage: Gibt es eine Teilmenge  $M' \subseteq M$ , so dass  $\sum_{m \in M'} w(m) \in [k, k + 3]$  gilt?

Zeigen Sie, dass das Problem SUBSET 3-INTERVALL NP-vollständig ist. Benutzen Sie dazu, dass das Problem SUBSET SUM NP-vollständig ist:

**Problem SUBSET SUM (S-SUM)**Gegeben: Endliche Menge  $M$ , Gewichtsfunktion  $w : M \rightarrow \mathbb{N}_{>0}$ , Zahl  $k \in \mathbb{N}_{>0}$ .Frage: Gibt es eine Teilmenge  $M' \subseteq M$ , so dass  $\sum_{m \in M'} w(m) = k$  gilt?**Lösung:**Problem S-3INT liegt in NP, denn für eine gegebene Menge  $M'$  lässt sich in Polynomialzeit  $\sum_{m \in M'} w(m)$ ausrechnen und überprüfen, ob  $\sum_{m \in M'} w(m) \in [k, k + 3]$  gilt.Um zu zeigen, dass S-3INT NP-schwer ist, zeigen wir S-SUM  $\propto$  S-3INT. Gegeben sei eine S-SUM Instanz  $(M, w, k)$ . Wir konstruieren die S-3INT-Instanz  $(M, w', 4k)$  mit  $w'(m) = 4w(m)$  für jedes  $m \in M$ .Dann gilt  $(M, w, k)$  lösbar  $\Leftrightarrow (M, w', 4k)$  lösbar.

- $\Rightarrow$ . Sei  $M'$  eine Lösung von S-SUM Instanz  $(M, w, k)$ . Dann ist  $M'$  eine Lösung von S-3INT Instanz  $(M, w', 4k)$ .
- $\Leftarrow$ . Sei  $M'$  eine Lösung von S-3INT Instanz  $(M, w', 4k)$ . Dann gilt

$$\sum_{m \in M'} w'(m) = \sum_{m \in M'} 4w(m) \in [4k, 4k + 3].$$

Daraus folgt

$$\sum_{m \in M'} w(m) \in [k, k + 3/4] = \{k\},$$

da  $\sum_{m \in M'} w(m) \in \mathbb{N}$ . Die Transformation ist polynomial.

**Aufgabe 7:**

(4+2 Punkte)

- (a) Sei  $L$  eine nicht entscheidbare Sprache über dem Alphabet  $\Sigma = \{0, 1\}$  und sei  $(01)^k \notin L$  für jedes  $k \in \mathbb{N}_{>0}$ . Zeigen Sie: Die Sprache

$$L' = \{w_1 \# w_2 \mid (w_1 \in L \wedge w_2 \notin L) \text{ oder } (w_1 \notin L \wedge w_2 \in L)\}$$

über dem Alphabet  $\Sigma' = \{0, 1, \#\}$  ist nicht entscheidbar.

**Lösung:**

Annahme:  $L'$  sei entscheidbar. Sei  $M'$  eine TM die  $L'$  entscheidet. Für ein Wort  $w \in \Sigma^*$  konstruiere das Wort  $w' = w \# 01$ . Dann gilt  $w' \in L' \Leftrightarrow w \in L$ . Damit kann man  $L$  wie folgt entscheiden, im Widerspruch zur Nichtentscheidbarkeit von  $L$ :

Für  $w \in \Sigma^*$  konstruiere  $w' = w \# 01$ . Simuliere dann die TM  $M'$  auf Eingabe  $w'$  und entscheide  $w \in L$  gdw  $M'$   $w' \in L'$  entscheidet.

- (b) Sei  $\mathcal{A} = (Q = \{s, f\}, \Sigma = \{a, b\}, \Gamma = \{Y, Z\}, \delta, s, Z, \{f\})$  der Kellerautomat mit Zustandsmenge  $Q$ , Eingabealphabet  $\Sigma$ , STACK-Alphabet  $\Gamma$ , Anfangszustand  $s$ , Stack-Initialisierung  $Z$ , akzeptierendem Endzustand  $f$  und der folgenden Übergangsrelation  $\delta$ :

$$(s, a, Z) \mapsto (s, YZ)$$

$$(s, a, Y) \mapsto (s, YY)$$

$$(s, b, Y) \mapsto (s, \epsilon)$$

$$(s, \epsilon, Z) \mapsto (f, Z)$$

Sei  $L$  die Sprache, die  $\mathcal{A}$  durch akzeptierenden Endzustand erkennt. Formen Sie  $\mathcal{A}$  in einen Kellerautomaten  $\mathcal{A}'$  um, der  $L$  durch leeren Stack erkennt.

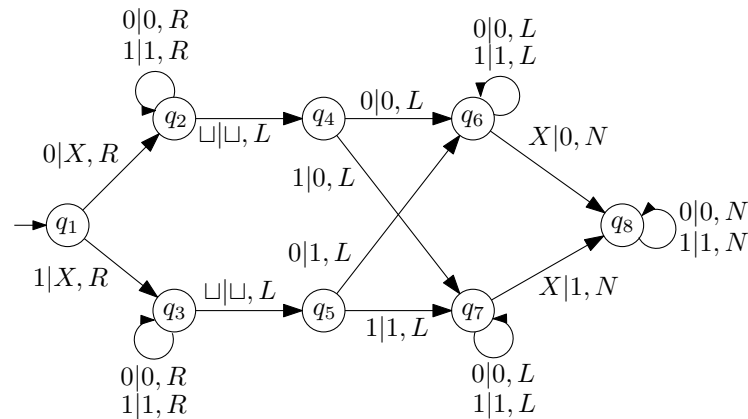
**Lösung:**

Ergänze  $\delta$  um  $(s, \epsilon, Z) \mapsto (s, \epsilon)$ .

**Aufgabe 8:**

(1+2+3 Punkte)

Gegeben sei die folgende Turingmaschine  $M$  mit Eingabealphabet  $\{0, 1\}$  und Bandalphabet  $\{0, 1, \sqcup, X\}$  wobei  $\sqcup$  das Blanksymbol ist.



(a) Ist die Turingmaschine  $M$  deterministisch? **Lösung:**

Ja.

(b) Rechnen Sie das Verhalten der Turingmaschine  $M$  bei Eingabe des Wortes 0101 durch. Geben Sie dazu alle auftretenden Konfigurationen an. **Lösung:**

$(q_1)$  0101  
 $X$   $(q_2)$  101  
 $X1$   $(q_2)$  01  
 $X10$   $(q_2)$  1  
 $X101$   $(q_2)$   $\sqcup$   
 $X10$   $(q_4)$  1  
 $X1$   $(q_7)$  00  
 $X$   $(q_7)$  100  
 $(q_7)$   $X100$   
 $(q_8)$  1100

(c) Was berechnet die Turingmaschine  $M$  für eine Eingabe der Länge mindestens 2? **Lösung:**

$M$  vertauscht erstes und letztes Zeichen, d.h. für Eingabewort  $w_1 \dots w_n$  wird  $w_n w_2 \dots w_{n-1} w_1$  berechnet.

**Aufgabe 9:**

(3 Punkte)

Gegeben sei die Grammatik  $G = (\Sigma, V, S, R)$  mit Terminalen  $\Sigma = \{a, b, c, d, x, f\}$ , Nichtterminalen  $V = \{S, A, B, C\}$  und Produktionen

$$R = \{ \begin{array}{l} S \rightarrow ABx|BCd|A \\ A \rightarrow AAf|a|CC \\ B \rightarrow b \\ C \rightarrow c \end{array} \}.$$

Berechnen Sie durch ein systematisches Verfahren eine Grammatik  $G'$  in Chomsky-Normalform, die die gleiche Sprache wie  $G$  erzeugt.

**Lösung:**

Schritt 1: Rechte Seite nur Terminale oder nur Variablen

$$R = \{ \begin{array}{l} S \rightarrow ABX|BCD|A \\ A \rightarrow AAF|a|CC \\ B \rightarrow b \\ C \rightarrow c \\ X \rightarrow x \\ D \rightarrow d \\ F \rightarrow f \end{array} \}$$

mit neuen Variablen  $X, D, F$ .

Schritt 2: Rechte Seite hat Länge höchstens 2

$$R = \{ \begin{array}{l} S \rightarrow AY_1|BY_2|A \\ A \rightarrow AY_3|a|CC \\ B \rightarrow b \\ C \rightarrow c \\ X \rightarrow x \\ D \rightarrow d \\ F \rightarrow f \\ Y_1 \rightarrow BX \\ Y_2 \rightarrow CD \\ Y_3 \rightarrow AF \end{array} \}$$

mit neuen Variablen  $Y_1, Y_2, Y_3$ .

Schritt 3: Kettenregeln beseitigen

$$R = \{ \begin{array}{l} S \rightarrow AY_1|BY_2|AY_3|a|CC \\ A \rightarrow AY_3|a|CC \\ B \rightarrow b \\ C \rightarrow c \\ X \rightarrow x \\ D \rightarrow d \\ F \rightarrow f \\ Y_1 \rightarrow BX \\ Y_2 \rightarrow CD \\ Y_3 \rightarrow AF \end{array} \}$$

**Aufgabe 10:**

(2+5 Punkte)

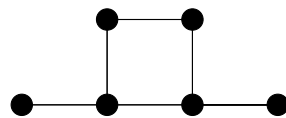
Gegeben sei ein ungerichteter Graph  $G = (V, E)$ .

- Zwei Kanten  $e_1, e_2 \in E$  sind **adjazent**, wenn sie einen gleichen Endknoten haben.
- Ein **Matching**  $E' \subseteq E$  von  $G$  ist eine Menge von Kanten die paarweise nicht adjazent sind.
- Ein **kardinalitätsmaximales Matching**  $E^*$  für  $G$  ist ein Matching mit maximal vielen Kanten, d.h. mit maximalem  $|E^*|$ .
- Ein **inklusionsmaximales Matching**  $E^{**}$  für  $G$  ist ein Matching, das nicht echte Teilmenge eines anderen Matchings ist, d.h. für das es kein Matching  $E''$  für  $G$  gibt mit  $E^{**} \subsetneq E''$ .

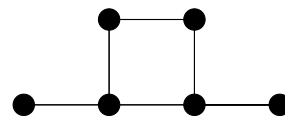
(a) Zeichnen Sie in

- Graphen (i) ein kardinalitätsmaximales Matching
- Graphen (ii) ein inklusionsmaximales Matching, das nicht kardinalitätsmaximal ist,

ein.

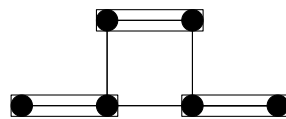


(i) kardinalitätsmaximales  
Matching

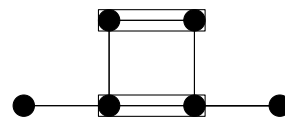


(ii) inklusionsmaximales,  
nicht kardinalitätsmaximales  
Matching

**Lösung:**



(i) kardinalitätsmaximales  
Matching



(ii) inklusionsmaximales,  
nicht kardinalitätsmaximales  
Matching

(b) Sei das folgende Maximierungsproblem MAXIMUM MATCHING gegeben:

**Problem MAXIMUM MATCHING**

Gegeben: Ein ungerichteter Graph  $G = (V, E)$ , der mindestens eine Kante enthält.

Gesucht: Ein Matching  $E'$  möglichst großer Kardinalität in  $G$ , d.h. wir haben ein Maximierungsproblem mit Zielfunktion  $|E'|$  und der Bedingung, dass  $E'$  ein Matching in  $G$  ist.

Sei  $\mathcal{A}$  ein Algorithmus, der für einen Eingabegraphen  $G$  ein beliebiges inklusionsmaximales Matching zurückliefert. Zeigen Sie:  $\mathcal{A}$  ist 1-approximativ für Problem MAXIMUM MATCHING (d.h.  $\mathcal{A}$  hat eine relative Gütegarantie von  $1 + 1 = 2$ ).

**Lösung:**

Sei  $M'$  ein beliebiges inklusionsmaximales Matching. Sei  $M^*$  ein beliebiges kardinalitätsmaximales Matching.

Jede Kante  $e \in M^*$  ist adjazent zu mindestens einer Kante in  $M'$  sonst wäre  $M' \cup \{e\}$  ein Matching mit Widerspruch zur Inklusionsmaximalität von  $M'$ .

Sei  $f : M^* \rightarrow M'$  eine Abbildung, die jeder Kante  $e^* \in M^*$  eine zu  $e^*$  adjazente Kante  $e' \in M'$  zuordnet.

Für jedes  $e' \in M'$  gilt

$$|\{e^* \in M^* \mid f(e^*) = e'\}| \leq 2 .$$

Damit ist

$$|M^*| \leq 2|M'| .$$

Dies ist äquivalent zu

$$\frac{|M^*|}{|M'|} \leq 2 .$$

Das war zu zeigen.

**Aufgabe 11:**

(10 Punkte)

Kreuzen Sie für folgende Aussagen an, ob diese wahr oder falsch sind.

*Hinweis:* Für jede richtige Antwort gibt es einen Punkt, für jede falsche Antwort wird ein Punkt abgezogen. Es wird keine negative Gesamtpunktzahl für diese Aufgabe geben.

Die Sprache  $L = \{(xyz)^i(abc)^i \mid i \in \mathbb{N}_{>0}, i < 27\}$  ist regulär.

Wahr

Falsch

**Lösung:** Wahr, denn  $L$  ist endlich.

Für jeden Kellerautomaten  $\mathcal{K}$  gilt: Die Sprache, die  $\mathcal{K}$  durch akzeptierenden Endzustand erkennt, ist gleich der Sprache, die  $\mathcal{K}$  durch leeren Stack erkennt.

Wahr

Falsch

**Lösung:** Falsch

Seien  $L_1, L_2$  Sprachen vom Chomsky-Typ 2. Dann ist  $L_1 \cap L_2$  vom Chomsky Typ 2.

Wahr

Falsch

**Lösung:** Falsch

Sei  $K$  ein Kellerautomat, der nie ein Zeichen vom Stack löscht oder auf den Stack schreibt und durch Endzustand akzeptiert. Dann gibt es einen NEA, der die gleiche Sprache akzeptiert wie  $K$ .

Wahr

Falsch

**Lösung:** Wahr

Wenn es für jedes Wort  $w \in L$  einer Sprache  $L$  einen DEA  $A_w$  gibt, der  $w$  akzeptiert, dann ist  $L$  regulär.

Wahr

Falsch

**Lösung:** Falsch, das gilt für jede Sprache, es gibt aber nichtreguläre Sprachen.

Es gibt ein Entscheidungsproblem  $\Pi \in NP$ , für das es keine polynomiale Transformation  $\Pi \propto SAT$  gibt.

Wahr

Falsch

**Lösung:** Falsch, folgt aus Definition von NP.

Jede Sprache, die von einer Grammatik erzeugt wird, ist entscheidbar.

Wahr

Falsch

**Lösung:** Falsch. Aus Vorlesung bekannt: Chomsky-0 ist gleich der Klasse der semi-entscheidbaren Sprachen.

Sei  $\Pi$  ein Optimalwertproblem und  $\mathcal{A}$  ein 4-Approximationsalgorithmus für  $\Pi$ . Dann liefert  $\mathcal{A}$  mit Wahrscheinlichkeit  $1/4$  die richtige Lösung für  $\Pi$ .

Wahr

Falsch

**Lösung:** Falsch

Das Problem 2-SAT liegt in NP.

Wahr

Falsch

**Lösung:** Wahr.

Sei  $L$  eine Sprache, die von einer nichtdeterministischen Turingmaschine  $\mathcal{M}$  in Polynomialzeit akzeptiert wird. Dann folgt daraus, dass es eine deterministische Turing-Maschine gibt, die  $L$  entscheidet.

Wahr

Falsch

**Lösung:** Wahr.