Universität Siegen Freiwillig erstellte Übung, unabhängig von der Universität

Grundlagen der theoretischen Informatik
(Übungsklausur)
—Musterlösung—

Dozent: Prof. Dr. M. L.
Autor: AKo (Student der GTI)
study.woalk.de

Sommersemester 2017
Dauer: ≈180 Minuten
Revision: 0.2

Erlaubte Hilfsmittel:

- Dokumentenechte Schreibgeräte (kein Bleistift, kein Rot)
- Ein beidseitig handschriftlich beschriebenes DIN-A4-Blatt.

Bitte beachten:

- Tragen Sie auf jedem Blatt Ihren Namen und Ihre Matrikel-Nr. ein.
- Schreiben Sie in den vorgesehenen Feldern. Sollte der Platz nicht ausreichen, schreiben Sie auf der Rückseite weiter und merken Sie es im Aufgabenfeld an.
- Prüfen Sie die Klausur auf Vollständigkeit aller 8 Seiten (ohne dieses Deckblatt).
  Die Klammerung der Klausur darf nicht gelöst werden.
- Schreiben Sie deutlich! Uneindeutige Ergebnisse sind ungültig.
- Zum Bestehen dieser Klausur benötigen Sie mindestens 50% der erreichbaren Punkte.
- Ein Täuschungsversuch (Abschreiben, unerlaubte Hilfsmittel, Kommunikation mit anderen Klausurteilnehmern, etc.) führt zum Ausschluss aus der Prüfung. Es erfolgt keine Verwarnung.

Inhaltliche Hinweise:

- Endliche Automaten können wahlweise graphisch oder tabellarisch angegeben werden.
- In LOOP- und WHILE-Programmen dürfen Addition, Multiplikation, Subtraktion (negativ $\rightarrow 0$) und die Bedingung $\text{IF } x_i = 0 \text{ THEN } P \text{ END}$ verwendet werden.
- Schreiben Sie zu LOOP- und WHILE-Programmen immer dazu, in welchen Variablen die Eingänge einer Funktion stehen, falls nötig. Die Ausgabe sollte am Ende immer in x_1 stehen.
- Addition und Multiplikation dürfen als primitiv rekursiv vorausgesetzt werden.

Name: **–Musterlösung–** Matrikelnummer: **0000000**

Diese Klausur ist eine reine studentische Übung, angefertigt für eigene Lernzwecke. Dies ist keine offizielle Klausur der Universität Siegen und wurde auch nicht unter derer Aufsicht erstellt. Keine Garantie auf Fehlerfreiheit, Vollständigkeit oder Relevanz.

Grundlagen der theoretischen Informatik – Übungsklausur

Punkteübersicht:

	Punkte:	**Bonus:**	**Erreicht:**
Aufgabe 1	10	0	
Aufgabe 2	10	0	
Aufgabe 3	10	0	
Aufgabe 4	5	0	
Aufgabe 5	10	0	
Aufgabe 6	10	0	
Aufgabe 7	10	0	
Aufgabe 8	10	0	
Aufgabe 9	10	0	
Aufgabe 10	5	0	
Aufgabe 11	10	0	
Aufgabe 12	0	10	
Gesamt	100	10	

Aufgabe 1. Fragenteil **Punkte: 10**

(a) (1 Punkt) Für alle Sprachen L gilt: $\left(L^+\right)^+ = L^+$
√ **Wahr** ○ Falsch

(b) (1 Punkt) Zu jedem DFA M mit n Zuständen gibt es einen NFA M' mit höchstens 2^n Zuständen, sodass gilt $T(M) = T\left(M'\right)$.
○ Wahr √ **Falsch**

(c) (1 Punkt) Sei $L \subseteq \Sigma^*$ eine reguläre Sprache. Es gilt $\forall w_1, w_2 \in \Sigma:\ w_1 \cdot w_2 \in L \Rightarrow w_1, w_2 \in L$.
○ Wahr √ **Falsch**

(d) (1 Punkt) Es gibt nicht-reguläre Sprachen, die die Eigenschaften des Pumping-Lemmas für reguläre Sprachen erfüllen.
√ **Wahr** ○ Falsch

(e) (1 Punkt) Unäre kontextfreie Sprachen sind immer regulär.
√ **Wahr** ○ Falsch

(f) (1 Punkt) Es gibt einen nicht-deterministischen Kellerautomaten M, zu dem kein deterministischer Kellerautomat M' mit $T(M) = T\left(M'\right)$ existiert.
√ **Wahr** ○ Falsch

(g) (1 Punkt) Semi-entscheidbare Sprachen (Typ-0) sind rekursiv aufzählbar.
√ **Wahr** ○ Falsch

(h) (1 Punkt) Es gibt eine berechenbare, partielle primitiv rekursive Funktion.
○ Wahr √ **Falsch**

(i) (1 Punkt) Seien f und g primitiv rekursive Funktionen. Dann ist $f \circ g$ ebenfalls primitiv rekursiv.
√ **Wahr** ○ Falsch

(j) (1 Punkt) Sei die Sprache A reduzierbar auf die Sprache B. Dann ist A genau dann entscheidbar, wenn B entscheidbar ist.
○ Wahr √ **Falsch**

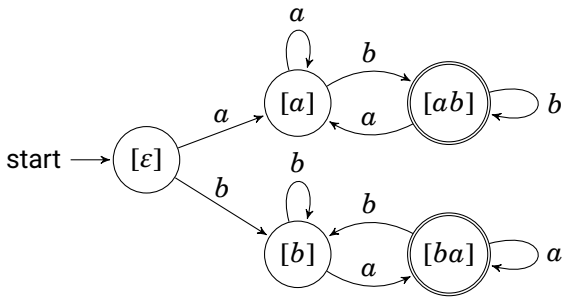
Aufgabe 2. Sei $L = \left\{w \in \{a,b\}^+ \;\middle|\; \text{das erste und letzte Zeichen von } w \text{ sind verschieden}\right\}$.

(a) (2 Punkte) Geben Sie einen regulären Ausdruck für L an.

Lösung: $L = L\left(a\,(a|b)^*\,b \;\middle|\; b\,(a|b)^*\,a\right)$

(b) (5 Punkte) Geben Sie einen DFA M mit $T(M) = L$ an.
Begründen Sie kurz in Stichworten, warum Ihr Automat korrekt ist.

Lösung: Zuerst: Anfangsbuchstaben merken. Dann: letzten Buchstaben merken, und jeweils davon abhängig in einen Endzustand oder nicht-Endzustand gehen.



(c) (3 Punkte) Geben Sie eine reguläre Grammatik G mit $L(G) = L$ an.

Lösung: $G = (V, \Sigma, P, S)$ mit $V = \{S, A, B\}$, $\Sigma = \{a, b\}$
$P = \{ S \to Ab \mid Ba,$
$\quad A \to Aa \mid Ab \mid a,$
$\quad B \to Ba \mid Bb \mid b \quad \}$

Aufgabe 3. Sei L die Menge aller Wörter über dem Alphabet $\{a, b\}$, deren Länge ungerade ist und deren mittleres Zeichen ein a ist, also: $L = \{w_1 a w_2 \mid w_1, w_2 \in \{a,b\}^* \wedge |w_1| = |w_2|\}$

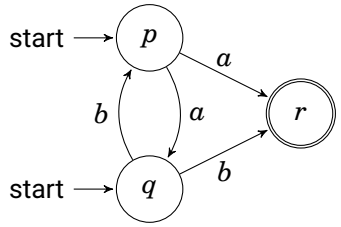
(a) (8 Punkte) Beweisen Sie mit Hilfe des Pumping-Lemmas, dass L nicht regulär ist.

Lösung: Sei n fest, aber beliebig, $n > 0$. Wähle $x = b^n a b^n \in L$. Es gilt also $|x| \geq n$. Betrachte alle Zerlegungen $x = uvw$ mit $|v| > 0$, $|uv| \leq n$:
$u = b^{k-l}$, $v = b^{n-(k-l)}$, $w = b^l a b^n$ mit $0 \leq l \leq k < n$.
Um eventuell regulär zu sein, muss $\forall i$ gelten: $uv^i w \in L$.
Sei $i = 2$: $uv^2w = b^{k-l} b^{2(n-(k-l))} b^l a b^n = b^{k-l+2n-k+l-l} a b^n = b^{2n+l} a b^n$
Da $2n > n$, gilt: $\left|b^{2n+l}\right| \neq |b^n|$ und damit $uv^2w \neq L$ $\quad \lightning L$ nicht regulär.

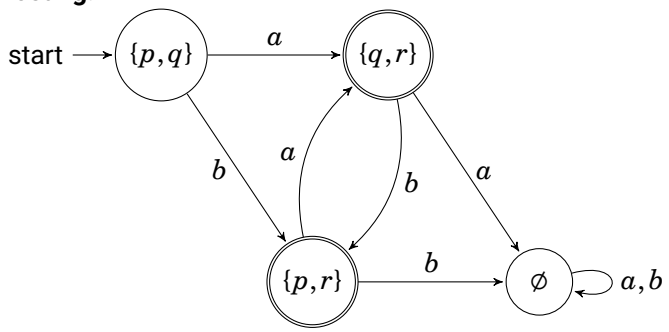
(b) (2 Punkte) Erklären Sie mit dem Satz von Myhill-Nerode, warum L nicht regulär ist.

Lösung: Um sich für den späteren Teil des Wortes zu „merken“, wie viele Zeichen w_1 hat, bräuchte man unendlich viele Äquivalenzklassen abhängig vom Wort w_1.

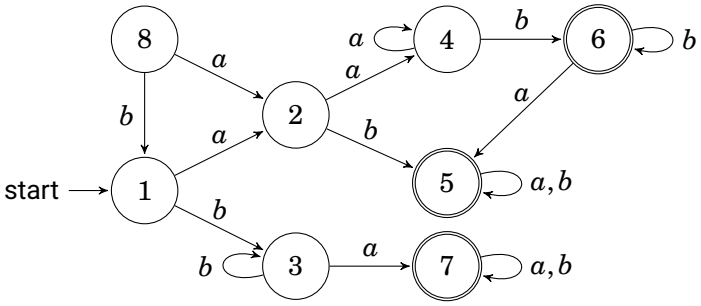
Aufgabe 4. (5 Punkte) Gegeben sei der folgende NFA M. Konstruieren Sie mit Hilfe der Potenzmengenkonstruktion einen zu M äquivalenten DFA.



Lösung:



Aufgabe 5. Gegeben sei der folgende DFA M: **Punkte: 10**

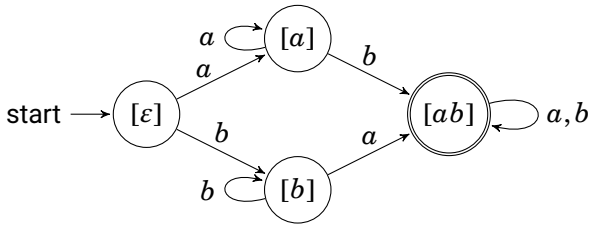


(a) (7 Punkte) Minimieren Sie M algorithmisch. Geben Sie in Ihrer Tabelle an, in welcher Reihenfolge Sie vorgegangen sind.

Lösung: Zustand 8 ist nicht vom Startzustand aus erreichbar und kann daher ignoriert werden.

2	2					
3	2	2				
4	2		2			
5	1	1	1	1		
6	1	1	1	1		
7	1	1	1	1		
	1	**2**	**3**	**4**	**5**	**6**

→ Die Zustände 2 und 4 sowie die Zustände 5, 6 und 7 sind erkennungsäquivalent.



Die Benennung der Zustände stellt hier die Äquivalenzklassen dar (nicht gefordert).

(b) (3 Punkte) Welche Sprache akzeptiert M?

Lösung: $T(M) = \{w \in \{a,b\}^* \mid w \text{ enthält sowohl } a \text{ als auch } b\}$

Aufgabe 6. (10 Punkte) Sei $L = \{a^m b^n c^{m+n} \mid m,n \geq 0\}$. Geben Sie eine kontextfreie Grammatik G an mit $L(G) = L$. Begründen Sie, warum Ihre Grammatik tatsächlich die Sprache L erzeugt.

Lösung: $G = (V, \Sigma, P, S)$ mit $\Sigma = \{a, b, c\}$, $V = \{S, A, B\}$,

$P = \{ S \to \varepsilon \mid A \mid B,$
$\quad A \to aAc \mid aBc \mid ac,$
$\quad B \to bBc \mid bc \quad \}$

unter Verwendung der ε-Sonderregel.

Diese Grammatik erzeugt zunächst entweder die Variante $n = m = 0 \to \varepsilon$, oder geht in A oder B über, die jeweils Paare von $a \ldots c$ bzw. $b \ldots c$ erzeugen. Für jedes a bzw. b wird damit auch ein c hinten hinzugefügt, sodass die Anzahl mit der Definition übereinstimmt. Nachdem einmal B genutzt wurde, können danach keine A mehr folgen und somit keine Konstrukte wie $ababcccc$ gebildet werden, die nicht zu L gehören.

Um außerdem $n = 0$ oder $m = 0$ erlauben zu können, kann A direkt nach ac, um kein einziges B nutzen zu müssen, und S direkt nach B übergehen, um kein einziges A nutzen zu müssen.

Aufgabe 7. Gegeben sei die folgende kontextfreie Grammatik. **Punkte: 10**

$G = (\{S,A,B\},\ \{a,b,c\},\ P,\ S)$

$P = \{S \to A \mid B,$
$A \to aAc \mid aBc,$
$B \to bBc \mid bc \quad \}$

(a) (7 Punkte) Wandeln Sie die Grammatik G in eine Grammatik G' in Chomsky-Normalform um. Gehen Sie schrittweise vor.

Lösung: $G' = \left(V',\ \{a,b,c\},\ P',\ S\right)$

1. Terminalsymbole in eigene Variablen einbetten:

$P = \{S \to A \mid B,$
$A \to A_aAA_c \mid A_aBA_c,$
$B \to A_bBA_c \mid A_bA_c,$
$A_a \to a,$
$A_b \to b,$
$A_c \to c \quad \}$

2. Kettenregeln eliminieren („hochziehen“):

$P = \{S \to A_aAA_c \mid A_aBA_c \mid A_bBA_c \mid A_bA_c,$
$A \to A_aAA_c \mid A_aBA_c,$
$B \to A_bBA_c \mid A_bA_c,$
$A_a \to a,$
$A_b \to b,$
$A_c \to c \quad \}$

3. Auf max. 2 Variablen pro Produkt reduzieren:

$P = \{S \to A_aA' \mid A_aB' \mid A_bB' \mid A_bA_c,$
$A \to A_aA' \mid A_aB',$
$B \to A_bB' \mid A_bA_c,$
$A' \to AA_c,$
$B' \to BA_c,$
$A_a \to a,$
$A_b \to b,$
$A_c \to c \quad \}$

$\to V' = \{S,A,B,A',B',A_a,A_b,A_c\}$

Lösung: <s.o.>

(b) (3 Punkte) Welche Sprache akzeptieren G und G'?

Lösung: $L(G) = L\left(G'\right) = \left\{a^n b^m c^{n+m} \;\middle|\; n,m \geq 1\right\}$
(nicht zu verwechseln mit der Sprache aus der vorherigen Aufgabe, wo $n,m \geq 0$ gilt)

Aufgabe 8. (10 Punkte) Zeigen Sie, dass die Sprache $L = \{a^m b^n c^{mn} \mid m, n \in \mathbb{N}\}$ nicht kontextfrei ist.

Lösung: (Ansatz) Sei n fest, aber beliebig.

Wähle $z = a^n b^n c^{nn}$. Damit ist $|z| > n$ und $z \in L$.

Betrachte alle Zerlegungen $z = uvwxy$ mit $|vx| > 0$ und $|vwx| \leq n$:

vwx kann daher nur maximal den gesamten Teil a^n, einen Teil t mit $|t| \leq n$ von $a^n b^n$, den gesamten Teil b^n, einen Teil t mit $|t| \leq n$ von $b^n c^{nn}$ oder einen Teil t mit $|t| \leq n$ von c^{nn} abdecken, aber niemals mehrere dieser gleichzeitig.

„Pumpt" man nun $uv^i wx^i y$ mit $i \in \mathbb{N}$ auf, so gibt es daher keine Möglichkeit, die Anzahl der as, bs und cs *gleichzeitig* zu erhöhen. Es gibt also i, für die $uv^i wx^i y \notin L$. Analog der Sprache $L' = \{a^n b^n c^n \mid n \geq 0\}$ ist L daher nicht kontextfrei.

Aufgabe 9. (10 Punkte) Konstruieren Sie eine Turing-Maschine, die bei Eingabe eines Wortes $w \in \{0, 1\}^*$ das gespiegelte Wort w^{R} auf das Band schreibt, den Kopf auf das erste Symbol dieses Wortes stellt und in einen Endzustand übergeht. Erklären Sie die Idee hinter ihrer Konstruktion.

Lösung: Die TM sollte das Wort nach rechts durchgehen, sich dabei jeweils einen Buchstaben merken, diesen durch ein X-Symbol ersetzen, den Buchstaben ans linke Ende des Wortes anhängen, und dann rechts hinter den Xen mit dem nächsten Buchstaben genauso fortfahren.

Am Schluss, wenn alle Symbole des originalen Wortes mit X ersetzt wurden, werden die X gelöscht und übrig bleibt das links stehende w^{R}.

$M = (Z, \Sigma, \Gamma, \delta, z_0, \square, E) \quad \Sigma = \{0,1\} \quad \Gamma = \Sigma \cup \{\square, X\} \quad E = \{z_e\} \quad Z = \{z_0, z_1, z_2, z_3, z_4, z_e\}$

$$
\begin{aligned}
\delta = \{ & (z_0, 0) \to (z_1, X, L), \\
& (z_0, 1) \to (z_2, X, L), \\
& (z_0, X) \to (z_0, X, R), \\
& (z_0, \square) \to (z_4, \square, L), \\
& (z_1, \gamma) \to (z_1, \gamma, L) \ \forall \gamma \in \{0,1,X\}, \\
& (z_1, \square) \to (z_3, 0, R), \\
& (z_2, \gamma) \to (z_2, \gamma, L) \ \forall \gamma \in \{0,1,X\}, \\
& (z_2, \square) \to (z_3, 1, R), \\
& (z_3, \gamma) \to (z_3, \gamma, R) \ \forall \gamma \in \{0,1\}, \\
& (z_3, X) \to (z_0, X, R), \\
& (z_4, X) \to (z_4, \square, L), \\
& (z_4, \gamma) \to (z_4, \gamma, L) \ \forall \gamma \in \{0,1\}, \\
& (z_4, \square) \to (z_e, \square, R) \qquad \}
\end{aligned}
$$

Aufgabe 10. (5 Punkte) Schreiben Sie ein WHILE-Programm, das die charakteristische Funktion der geraden natürlichen Zahlen berechnet.

Geben Sie an, in welchen Variablen die Eingabe steht. Die Ausgabe sollte am Ende in x_1 stehen.

Lösung: Zunächst, definiere die charakteristische Funktion der geraden natürlichen Zahlen:

$$\chi_{2\mathbb{N}}(n) = \begin{cases} 1 & \text{wenn } n \text{ gerade } (n \mod 2 = 0) \\ 0 & \text{wenn } n \text{ ungerade } (n \mod 2 = 1) \end{cases}$$

Sei x_1 die Ein- und Ausgabe.

WHILE $x_1 \neq 0$ DO
  $x_2 := x_1$;
  $x_1 := x_1 - 2$;
END;
$x_3 := 2$;
$x_3 := x_3 - x_2$;
IF $x_3 = 0$ THEN
  $x_1 := 1$
END

Aufgabe 11. Sei $f : \mathbb{N} \to \mathbb{N}$ eine totale μ-rekursive Funktion. **Punkte: 10**

Zeigen Sie, dass auch die folgenden Funktionen (I) μ-rekursiv und (II) total sind:

(a) (5 Punkte) $g : \mathbb{N} \to \mathbb{N}$, die jedem $n \in \mathbb{N}$ das Maximum der Zahlen $f(0), \ldots, f(n)$ zuordnet.

Lösung: $g(0) = f(0)$
$g(n+1) = \max(f(n+1),\ g(n))$

Nun nur noch zu zeigen, dass $\max(n,\ m)$ μ-rekursiv ist. μ-rekursive Funktionen sind äquivalent zu WHILE-Programmen. Existiert ein WHILE-Programm für die $\max$-Funktion?

$\max(x_1,\ x_2)$, Ausgabe in x_1:

$x_3 := x_1 - x_2$;
IF $x_3 = 0$ THEN
  $x_1 := x_2$
END

Ist sogar ein LOOP-Programm! Also ist $\max$ primitiv rekursiv und damit auch μ-rekursiv.

→ Also ist $g(n)$ μ-rekursiv.

Da $\max$ primitiv rekursiv, ist $\max$ immer total. Daher ist auch g total (besteht nur aus den totalen Funktionen f, $\max$ und Addition).

(b) (5 Punkte) $h : \mathbb{N} \to \mathbb{N}$, die jedem $n \in \mathbb{N}$ eine Zahl m zuordnet, sodass $f(m)$ das Maximum der Zahlen $f(0), \ldots, f(n)$ ist.

Lösung: Diese Funktion ist analog zur Funktion g, außer, dass hier nicht direkt das Maximum selbst zurückgegeben wird, sondern der Wert, der das Maximum erzeugt.

μ-rekursive Funktionen sind äquivalent zu WHILE-Programmen.
Existiert ein WHILE-Programm für h?

Sei die Ein- und Ausgabe in x_1.

$x_1 := x_1 + 1;$
LOOP x_1 DO
  $x_1 := x_1 - 1;$
  $x_4 := f(x_1);$
  $x_5 := x_2 - x_4;$
  IF $x_5 = 0$ THEN
    $x_2 := x_4;$
    $x_3 := x_1$
  END
END;
$x_1 := x_3$

$\rightarrow$ Also ist h μ-rekursiv.

Da dies sogar praktisch ein LOOP-Programm ist (bis auf die Berechnung von der totalen Funktion f), ist h auch total.

Aufgabe 12. (10 Bonuspunkte) Beweisen Sie die Unentscheidbarkeit des speziellen Halteproblems (Sprache $K = \{w \in \{0,1\}^* \mid M_w$ angesetzt auf w hält$\}$). Dabei stellt M_w die Turing-Maschine dar, die durch das Wort w kodiert wird (abgespielt durch eine universelle Turing-Maschine).

Lösung: Nehme man an, die charakteristische Funktion des speziellen Halteproblems χ_K wäre durch eine Turing-Maschine berechenbar.

Sei M' eine Turing-Maschine, die für ein eingegebenes Wort w genau dann in eine Endlosschleife übergeht, wenn das spezielle Halteproblem für w „terminiert“ ausgibt, sowie 0 ausgibt, wenn das spezielle Halteproblem für w „terminiert nicht“ ausgibt.

Sei w' so gewählt, dass $M' = M_{w'}$ (w' ist die Kodierung der TM M' selbst).

Dann gilt:

$M' = M_{w'}$ hält bei Eingabe $w' \Leftrightarrow M$ gibt 0 bei Eingabe w' aus $\Leftrightarrow \chi_K(w') = 0 \Leftrightarrow w' \notin K$
$\Leftrightarrow M_{w'}$ hält nicht bei Eingabe w' $\quad \lightning$

$\Rightarrow$ die TM kann nicht existieren, das spezielle Halteproblem ist also nicht entscheidbar.