Technische Universität München
Fakultät für Informatik
Lehrstuhl Informatik II
Prof. Dr. Helmut Seidl
Dr. Werner Meixner

Sommersemester 2013
Lösungen der Klausur
26. September 2013

Einführung in die Theoretische Informatik

Name	Vorname	Studiengang	Matrikelnummer
....	....	☐ Diplom ☐ Inform. ☐ Bachelor ☐ BioInf. ☐ Lehramt ☐ WirtInf.	....

Hörsaal	Reihe	Sitzplatz	Unterschrift
....	....	....	....

Code:

Allgemeine Hinweise

- Bitte füllen Sie obige Felder in Druckbuchstaben aus und unterschreiben Sie!
- Bitte schreiben Sie nicht mit Bleistift oder in roter/grüner Farbe!
- Die Arbeitszeit beträgt 150 Minuten.
- Alle Antworten sind in die geheftete Angabe auf den jeweiligen Seiten (bzw. Rückseiten) der betreffenden Aufgaben einzutragen. Auf dem Schmierblattbogen können Sie Nebenrechnungen machen. Der Schmierblattbogen muss ebenfalls abgegeben werden, wird aber in der Regel nicht bewertet.

Hörsaal verlassen von ...... bis ...... / von ...... bis ......
Vorzeitig abgegeben um ......
Besondere Bemerkungen:

	A1	A2	A3	A4	A5	A6	A7	Σ	Korrektor
Erstkorrektur									
Zweitkorrektur									

Aufgabe 1 (6 Punkte)

Wahr oder falsch? Begründen Sie Ihre Antwort!
Für die richtige Antwort und für die richtige Begründung gibt es jeweils einen $\frac{1}{2}$ Punkt.

1. Es gibt ein LOOP-Programm zur Berechnung der Ackermann-Funktion.
2. Sei $L = \{w \in \Sigma^* \mid \forall n \in \mathbb{N}.\ \varphi_w(n){=}n^2{-}n\}$. Dann ist L entscheidbar.
3. Seien $L \subseteq \Sigma^*$ eine reguläre Sprache und $K \subseteq L$. Dann ist K entscheidbar.
4. Sei $L \subseteq \Sigma^*$ nicht kontextfrei. Dann ist L nicht entscheidbar.
5. Das PCP $((1, 01), (10, 1), (10, 01))$ besitzt unendlich viele Lösungen.
6. $0^*1(0|1)^* \equiv (0|1)^*10^*$.

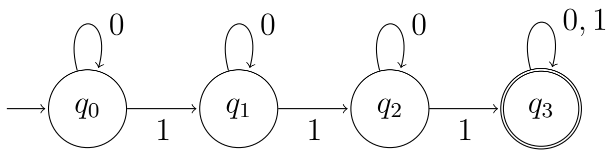
Lösung

1. Falsch! Nach Satz der Vorlesung ist die Ackermann-Funktion nicht primitiv rekursiv.
2. Falsch! Satz von Rice: Sei $F = \{f : \mathbb{N} \to \mathbb{N} \mid f \text{ berechenbar und } \forall n \in \mathbb{N}. f(n) = n^2{-}n\}$. Dann ist F einelementig, und erfüllt mindestens die Bedingungen des Satzes von Rice, mithin ist L nicht entscheidbar.
3. Falsch! Beispiel $L = \Sigma^*$ und $K = H_0$.
4. Falsch! Verweis auf Vorlesung. Oder: Alle kontextsensitiven Sprachen sind entscheidbar, Beispiel $\{a^n b^n c^n\}$ aus Übungen für kontextsensitive und nicht kontextfreie Sprachen.
5. Wahr! Konstruktion: $(10, 1)(10, 01)(10, 01) \ldots (10, 01)(1, 01)$.
6. Wahr! Sei $w \in \{0, 1\}^*$ ein Wort, das mindestens eine 1 enthält. Betrachtet man die erste 1 in w, dann liegt w offenbar in $0^*1(0|1)^*$. Umgekehrt enthält jedes Wort aus $0^*1(0|1)^*$ mindestens eine 1. Damit beschreibt $0^*1(0|1)^*$ genau die Menge aller $w \in \{0, 1\}^*$, die mindestens eine 1 enthalten.

   Analog folgt, dass $(0|1)^*10^*$ genau gleich der Menge aller $w \in \{0, 1\}^*$ ist, die mindestens eine 1 enthalten. Mithin gilt die Gleichung.

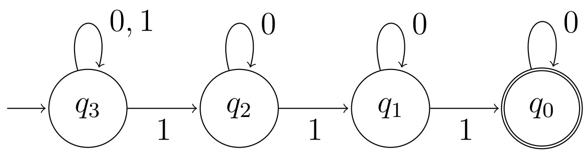
Aufgabe 2 (10 Punkte)

Sei $\Sigma = \{0, 1\}$. Der folgende Übergangsgraph definiert einen deterministischen endlichen Automat $A = (Q, \Sigma, \delta, q_0, \{q_3\})$.



1. Beweisen Sie durch Anwendung eines Minimierungsalgorithmus, dass A minimal ist in dem Sinne, dass für <u>keine</u> zwei verschiedenen Zustände $p, q \in Q$ die Äquivalenzbeziehung $p \equiv_A q$ gilt.

2. Durch geeignete „Spiegelung“ gewinnt man aus A den NFA $A^R = (Q, \Sigma, \delta^R, q_3, \{q_0\})$ wie folgt:



   Definieren Sie ein Verfahren zur Überprüfung der Gleichung $L(A) = L(A^R)$ und begründen Sie dessen Korrektheit.

   Beweisen Sie nun durch Anwendung Ihres Verfahrens die Gleichheit der Sprachen $L(A)$ und $L(A^R)$!

3. Geben Sie einen regulären Ausdruck α für die Sprache $L(A)$ an, so dass also $L(\alpha) = L(A)$ gilt.

Lösung

Bepunktung vorbehaltlich einer Änderung der Detailpunkte.

1. Anwendung des Minimierungsverfahrens, beginnend bei der letzten Zeile mit $p \not\equiv_A q_3$ für $p \neq q_3$:

   $\delta(q_1, 1) = q_2 \wedge \delta(q_2, 1) = q_3 \wedge q_2 \not\equiv_A q_3 \implies q_1 \not\equiv_A q_2,$
   $\delta(q_0, 1) = q_1 \wedge \delta(q_1, 1) = q_2 \wedge q_1 \not\equiv_A q_2 \implies q_0 \not\equiv_A q_1,$
   $\delta(q_0, 1) = q_1 \wedge \delta(q_2, 1) = q_3 \wedge q_1 \not\equiv_A q_3 \implies q_0 \not\equiv_A q_2.$

	q_0	q_1	q_2
q_1			
q_2			
q_3	X	X	X

,

	q_0	q_1	q_2
q_1	X		
q_2	X	X	
q_3	X	X	X

.

   Ergebnis: $p \neq q \implies p \not\equiv_A q$. (3P)

2. Verfahren:

   1. Durch Potenzmengenverfahren konstruiert man einen zu A^R äquivalenten DFA B.
   2. Durch Minimierungsverfahren konstruiert man einen zu B äquivalenten DFA C.
   3. Nun prüft man, ob C aus A^R durch einfache Umbezeichnung der Zustände hervorgeht.

   (2P)

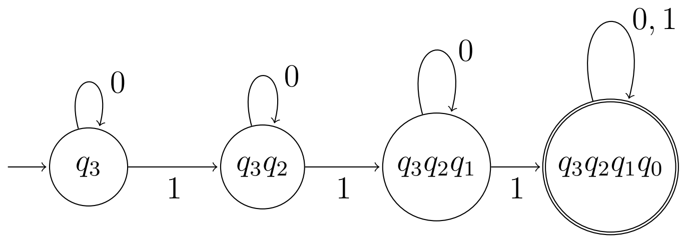
   1. Anwendung des Potenzmengenverfahrens mit Kurznotation:

p	$\delta(p,0)$	$\delta(p,1)$
q_3	q_3	q_3q_2
q_3q_2	q_3q_2	$q_3q_2q_1$
$q_3q_2q_1$	$q_3q_2q_1$	$q_3q_2q_1q_0$
$q_3q_2q_1q_0$	$q_3q_2q_1q_0$	$q_3q_2q_1q_0$

   (2P)

   mit $q_3q_2q_1q_0$ als einzigem Endzustand.

   Darstellung als Graph:



   2. und 3.: Da der Automat B bis auf Umbezeichnung der Zustände identisch ist mit Automat A, ist B bereits minimal und $L(B) = L(A)$. Es folgt $L(A^R) = L(A)$.

   (1P)

3. $\alpha = 0^*10^*10^*1(0|1)^*$. (2P)

Aufgabe 3 (10 Punkte)

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

1. Definieren Sie einen Kellerautomat $K_1 = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \emptyset)$, der die Sprache $L_1 = \{ca^n db^n \mid n \in \mathbb{N}\}$ mit leerem Keller akzeptiert, so dass also $L_\epsilon(K_1) = L_1$ gilt! Geben Sie den Übergangsgraph Ihres Automaten K_1 an.

   Ist Ihr Automat K_1 deterministisch?

2. Wir betrachten für eine beliebige aber feste natürliche Zahl $k_0 \in \mathbb{N}$ (z.B. $k_0 = 2$) die Sprache
$$L_{k_0} = \{c^{k_0} a^n d^{k_0} b^n \mid n \in \mathbb{N}\}\,.$$
   (a) Geben Sie für beliebiges $k_0 \geq 1$ ein Verfahren an zur Konstruktion einer Grammatik G_{k_0} in Chomsky-Normalform, so dass G_{k_0} die Sprache L_{k_0} erzeugt. Benützen Sie u.a. indizierte Nichtterminale U_i, V_i.

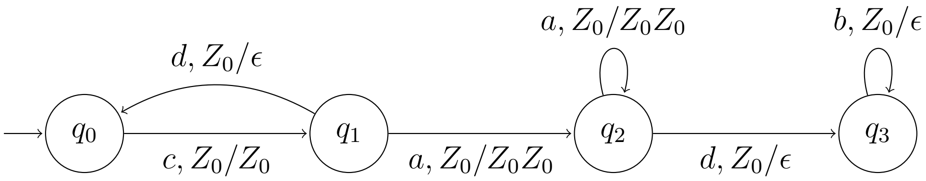
   (b) Erzeugen Sie für $k_0 = 2$ durch Anwendung Ihres Verfahrens eine konkrete Grammatik G_2, so dass $L(G_2) = L_2$ gilt.

3. Seien $k_0 \in \mathbb{N}$ beliebig aber fest und
$$L = \{c^k a^n d^k b^n \mid k, n \in \mathbb{N}, k \leq k_0\}\,.$$
   Gibt es einen Kellerautomat K, der L akzeptiert? Begründung!

Lösung

Bepunktung vorbehaltlich einer Änderung der Detailpunkte.

1. In graphischer Notation sieht K_1 wie folgt aus:



(3P)

   Da kein spontaner Übergang definiert wurde, gilt $|\delta(q, x, Z_0)| + |\delta(q, \epsilon, Z_0)| \leq 1$ für alle $x \in \Sigma, q \in Q$. Damit ist K_1 deterministisch.

(1P)

2. (a) Wir definieren $G_{k_0} = (N, \Sigma, P, S)$ mit $N = \{S, A, B, C, D, U_1, U_2 \ldots, U_{k_0}, V_1, V_2 \ldots, V_{k_0}\}$ und Produktionen aus P für alle $2 \leq i \leq k_0$ wie folgt:

$$\begin{array}{lll} S \rightarrow U_{k_0} V_{k_0}\,, & V_{k_0} \rightarrow AX\,, & X \rightarrow V_{k_0} B\,, \\ A \rightarrow a\,, & C \rightarrow c\,, & U_1 \rightarrow c\,, \\ B \rightarrow b\,, & D \rightarrow c\,, & V_1 \rightarrow d\,, \\ U_i \rightarrow CU_{i-1}\,, & V_i \rightarrow DV_{i-1}\,. & \end{array}$$

(b) Für $i = 2$ lautet die letzte Zeile: $U_2 \to CU_1$, $V_2 \to DV_1$.

(4P)

3. L ist für alle $k_0 \in \mathbb{N}$ kontextfrei, da alle L_i kontextfrei sind und L wegen

$$L = \bigcup_{i=0}^{k_0} L_i$$

eine endliche Vereinigung von kontextfreien Sprachen ist. Folglich gibt es auch einen entsprechenden Kellerautomat, der L akzeptiert.

(2P)

Aufgabe 4 (9 Punkte)

Wir betrachten die Grammatik $G = (V, \{a, b\}, P, S)$ mit den Produktionen

$$\begin{aligned} &S \to AB \mid BX\,, \\ &A \to BA \mid a\,, \qquad B \to XX \mid b\,, \qquad X \to AB \mid b\,. \end{aligned}$$

1. Beweisen Sie durch Anwendung des Cocke-Younger-Kasami-Algorithmus, dass $baabb \in L(G)$ gilt.

   Fertigen Sie dazu ein übersichtliches Protokoll an und lassen Sie genügend Platz für die Eintragung der jeweils relevanten Syntaxbäume von Ableitungen im Sinne von Teilaufgabe 2.

2. Berechnen Sie nun mit einer geeigneten Erweiterung des CYK-Algorithmus alle existierenden verschiedenen Syntaxbäume für Ableitungen des Wortes *baabb*!

   Stellen Sie alle Syntaxbäume für Ableitungen von *baabb* graphisch dar!

Lösung

Bepunktung vorbehaltlich einer Änderung der Detailpunkte.

1.

$^{15}\ S, B, S, X$				
$^{14}\ \emptyset$	$^{25}\ X$			
$^{13}\ \emptyset$	$^{24}\ \emptyset$	$^{35}\ S, X, B$		
$^{12}\ A$	$^{23}\ \emptyset$	$^{34}\ S, X$	$^{45}\ B, S$	
$^{11}\ B, X$	$^{22}\ A$	$^{33}\ A$	$^{44}\ B, X$	$^{55}\ B, X$
b	a	a	b	b

(5P)

2. Es gibt genau 2 Syntaxbäume zu folgenden Ableitungen:

   B_1: $S \to_G AB \to_G BAB \to_G^* baabb\,.$

   B_2: $S \to_G BX \to_G BAB \to_G^* baabb\,.$

(4P)

Aufgabe 5 (9 Punkte)

Gegeben sei die Grammatik $G = (\{S\}, \{a, b\}, P, S)$ mit den folgenden Produktionen:

$$S \rightarrow T \mid SS, \quad T \rightarrow aSaSb \mid \epsilon.$$

1. Zeigen Sie mit struktureller Induktion, dass für alle ableitbaren Wörter $w \in L(G)$ die Anzahl der enthaltenen a doppelt so groß ist wie die Anzahl der enthaltenen b, d. h., dass gilt $\#_a(w) = 2 \cdot \#_b(w)$.

2. Zeigen Sie induktiv, dass alle Satzformen $\alpha_n = a^n S(ab)^n$ für $n \geq 0$ in G ableitbar sind.

3. Zeigen Sie mit Hilfe des Pumping-Lemmas, dass $L(G)$ nicht regulär ist.

Lösung

Bepunktung vorbehaltlich einer Änderung der Detailpunkte.

1. Eliminierung der Kettenproduktion $S \rightarrow T$ bzw. Variablen T (1P.)
   führt zu $S \rightarrow aSbSb \mid SS \mid \epsilon$.

   Mit Induktion über die Erzeugung zeigt man für alle $w \in L(G)$ die Eigenschaft $\underbrace{\#_a(w) = 2 \cdot \#_b(w)}_{P(w)}$:

   $S \rightarrow \epsilon$: Offenbar gilt $P(\epsilon)$, denn $\#_a(\epsilon) = 2 \cdot \#_b(\epsilon) = 0$. (1P.)

   $S \rightarrow aSbSb$:
   Es gelte $P(x), P(y)$ und $x, y \in L(S)$. Dann gilt für $w' = axayb$:
   $\#_a(w') = \#_a(x) + 1 + \#_a(y) + 1 = 2 \cdot \#_b(x) + 2 \cdot \#_b(y) + 2 = 2 \cdot (\#_b(x) + \#_b(y) + 1)$
   $= 2 \cdot \#_b(w')$, d.h. es gilt $P(w')$. (1P.)

   $S \rightarrow SS$:
   Es gelte $P(x), P(y)$ und $x, y \in L(S)$. Dann gilt f'ur $w' = xy$:
   $\#_a(w') = \#_a(x) + \#_a(y) = 2 \cdot \#_b(x) + 2 \cdot \#_b(y) = 2 \cdot (\#_b(x) + \#_b(y))$
   $= 2 \cdot \#_b(w')$, d.h. es gilt $P(w')$. (1P.)

2. Induktion über n. ($\frac{1}{2}$P.)
   Für $n = 0$ gilt $S \rightarrow_G^* S$. ($\frac{1}{2}$P.)

   Falls $S \rightarrow_G^* a^n S(ab)^n$, dann folgt

   $S \rightarrow_G^* a^n S(ab)^n \rightarrow_G a^n aSaSb(ab)^n \rightarrow_G a^n aaSab(ab)^n = a^{n+1} S(ab)^{n+1}$. (1P.)
   Bewertung: Bei unklarer Beweisstruktur maximal 1P.

3. Sei $n \geq 1$ eine Pumping-Lemma-Zahl. Es gilt $z = a^n(ab)^n \in L(G)$. (1P.)

   Seien $u, v, w \in \{a, b\}^*$ mit $z = uvw$, $|uv| \leq n$ und $v \neq 1$.

   Dann gilt $v \in \{a\}^*$. Dann folgt für $i = 0$, dass $z' = uv^iw \in L(G)$ gilt.

   Andrerseits gilt aber $\#_a(z') \neq 2\#_b(z')$, und damit $z' \notin L(G)$. Widerspruch! (2P.)

Aufgabe 6 (8 Punkte)

Seien $c : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ die Cantorsche Paarungsfunktion mit den Umkehrfunktionen $p_1 : \mathbb{N} \to \mathbb{N}$ und $p_2 : \mathbb{N} \to \mathbb{N}$. Wir definieren eine Funktion $k : \mathbb{N} \to \mathbb{N}$ wie folgt:

$$k(0) = c(2,1) \quad \text{und} \quad k(n+1) = c\left(p_2(k(n)),\, 3p_2(k(n)) + p_1(k(n))\right) \;\forall n \in \mathbb{N}\,.$$

1. Wir betrachten die Funktion $f(n) = p_1(k(n))$ für $n \in \mathbb{N}$. Zeigen Sie, dass es Parameter $a, b \in \mathbb{N}$ gibt, so dass die folgende Rekursionsgleichung für alle $n \in \mathbb{N}$ erfüllt ist.
$$f(n+2) - a \cdot f(n+1) - b \cdot f(n) = 0\,. \tag{1}$$

2. Zeigen Sie, dass für alle $a, b \in \mathbb{N}$ jede Funktion $f : \mathbb{N} \to \mathbb{N}$, die die obige Rekursionsgleichung (1) erfüllt, primitiv rekursiv ist.

   Hinweis: Sie dürfen aus der Vorlesung oder den Übungen bekannte Ergebnisse über primitiv rekursive Funktionen verwenden.

Lösung

Bepunktung vorbehaltlich einer Änderung der Detailpunkte.

1. Wir rechnen für alle $n \in \mathbb{N}$
$$\begin{aligned} f(n+2) - af(n+1) - bf(n) &= p_1(k(n+2)) - ap_1(k(n+1)) - bp_1(k(n)) \\ &= p_2(k(n+1)) - ap_2(k(n)) - bp_1(k(n)) \\ &= 3p_2(k(n)) + p_1(k(n)) - ap_2(k(n)) - bp_1(k(n)) \\ &= (3-a)p_2(k(n)) + (1-b)p_1(k(n)) \\ &= 0 \qquad\qquad \text{für } a = 3 \text{ und } b = 1\,. \end{aligned}$$
(4P)

2. Sei f so, dass (1) erfüllt ist. Wir definieren eine Funktion h durch
$$h(0) = c(f(0), f(1)) \quad \text{und} \quad h(n+1) = c\left(p_2(h(n)),\, ap_2(h(n)) + bp_1(h(n))\right) \;\forall n \in \mathbb{N}\,.$$
(1P)

   c, p_1, p_2 sind PR. Damit ist h primitiv rekursiv nach dem erweiterten Rekursionsschema. (1P)

   Es folgt per Induktion $f(n) = p_1(h(n))$ für alle $n \in \mathbb{N}$. (1P)

   Damit ist f primitiv rekursiv. (1P)

   Alternativ ist auch ein Beweis mit LOOP-Programm zulässig.

Aufgabe 7 (8 Punkte)

Für alle folgenden Teilaufgaben sei $\Sigma = \{0, 1\}$. U sei eine universelle Turingmaschine, die für Eingabe $w\#x$ die Turingmaschine M_w mit Eingabe x simuliert und genau dann hält, wenn M_w hält.

1. Beweisen Sie, dass die folgende Aussage <u>falsch</u> ist:

   Sei F eine Menge von totalen Funktionen $f : \mathbb{N} \to \mathbb{N}$ mit $F \neq \emptyset$. Dann ist unentscheidbar, ob $\varphi_w \in F$ für $w \in \Sigma^*$ gilt. D.h., dass die Menge $R = \{w \in \Sigma^* \mid \varphi_w \in F\}$ nicht entscheidbar ist.

2. Seien $K = \{w \in \Sigma^* \mid M_w[w]\downarrow\}$ das spezielle Halteproblem und $S = \{w \in \Sigma^* \mid \varphi_w(w) = 0\}$ das Problem der „Selbstannullierung".

   Zeigen Sie mit Hilfe einer Reduktion von K auf S, d.h. $K \leq S$, dass das Problem S nicht entscheidbar ist.

3. Sei $A \in \Sigma^*$ in NP. Zeigen Sie, dass dann A^2 ebenfalls in NP ist.

   <u>Hinweis:</u> Verwenden Sie Zertifikate.

Lösung

Bepunktung vorbehaltlich einer Änderung der Detailpunkte.

1. Sei $f = ntime_M$, wobei die Turingmaschine M die charakteristische Funktion des Halteproblems H_0 berechne. f ist total und nicht berechenbar. (2P)

   Sei $F = \{f\}$. Dann ist R die leere Menge und natürlich entscheidbar. (1P)

2. Konstruktion einer Reduktionsabbildung $f(w)$: $M_{w'}$ ruft U mit Eingabe $w\#w$ auf. Falls $M_w[w]$ hält, dann ruft $M_{w'}$ eine Turingmaschine T auf, die 0 ausgibt bei beliebiger Eingabe. Sei w' der Kode der konstruierten Turingmaschine, dann wird schließlich $f(w) = w'$ definiert. (3P)

3. Sei c_A ein Zertifikat, so dass $w \in A$ von einem polynomiellen Verifikator M_A mit Zertifikat c_A überprüft wird. Dann wird ein Verifikator M_{A^2} konstruiert, der für w ein Zertifikat c wie folgt überprüft:

   $$w\#c \in L(M_{A^2}) :\Longleftrightarrow (c = u\#v\#c_A) \wedge (uv = w) \wedge (u\#c_A \in L(M_A)) \wedge (v\#c_A \in L(M_A))$$

   (2P)