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

Sommersemester 2013
Lösungen der Klausur
31. Juli 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.

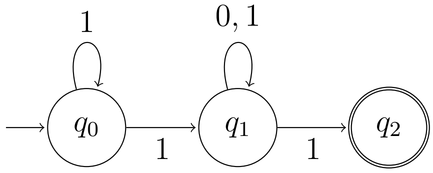
1. Falls R die Menge aller $w \in \{0,1\}^*$ ist, so dass die von der Turingmaschine M_w akzeptierte Sprache semi-entscheidbar ist, dann ist R entscheidbar.
2. Das PCP $((01, 1), (10, 01), (0, 01))$ besitzt eine Lösung.
3. Die Sprache $L = \{w \in \Sigma^* \mid \exists x.\ \varphi_w(x) = 0\}$ ist entscheidbar.
4. Falls eine nicht entscheidbare Sprache A von einer Turingmaschine M akzeptiert wird, dann ist $ntime_M$ nicht berechenbar.
5. Es gibt ein NP-vollständiges Problem A, so dass die charakteristische Funktion χ_A von A nicht μ-rekursiv ist.
6. Der Wertebereich der Ackermann-Funktion ist entscheidbar.

Lösung

1. Wahr! Die Menge R ist identisch mit $\{0,1\}^*$.
2. Falsch! Es kommt nur $(0, 01)$ als erstes Element in Frage. Dann können nur Elemente $(10, 01)$ folgen, die aber nicht abschließen.
3. Falsch! Satz von Rice, denn es gibt M_w, so dass $\varphi_w(0) = 0$. Und es gibt ein $M_{w'}$, so dass $\varphi_{w'}(x) \neq 0$ für alle x.
4. Wahr! Wäre $ntime_M$ berechenbar könnte mit $ntime_M(w) = 0$ entschieden werden, ob M hält.
5. Falsch! NP-vollständige Probleme sind entscheidbar. Damit ist χ_A berechenbar und somit μ-rekursiv.
6. Wahr! Der Wertebereich der Ackermann-Funktion ist $\mathbb{N} \setminus \{0\}$.

Aufgabe 2 (9 Punkte)

Durch die folgende Grafik sei ein nichtdeterministischer endlicher Automat $A = (Q, \Sigma, \delta, q_0, F)$ gegeben.



1. Geben Sie einen regulären Ausdruck α an, dessen Sprache von A akzeptiert wird, d. h. $L(\alpha) = L(A)$.
2. Bestimmen Sie $\overline{\delta}(\{q_0, q_1, q_2\}, 0)$!
3. Konstruieren Sie mit dem Potenzmengenverfahren einen deterministischen endlichen Automat B, der $L(A)$ akzeptiert. Das angewandte Verfahren muss deutlich gemacht werden.

   Stellen Sie den erhaltenen Automat B als Übergangsgraph dar.
4. Wie ist der Automat B abzuändern, so dass B das Komplement von $L(A)$ akzeptiert, d. h., so dass $L(B) = \{0, 1\}^* \setminus L(A)$ gilt?

Lösung

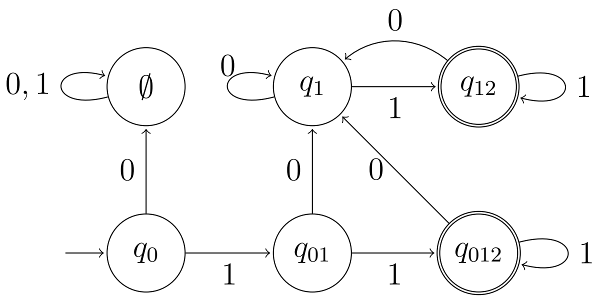
Bepunktung vorbehaltlich der Änderung der Detailpunkte.

1. $\alpha = 1^*1(0|1)^*1$. (2P)
2. $\overline{\delta}(\{q_0, q_1, q_2\}, 0) = \{q_1\}$. (1P)
3. Anwendung des Potenzmengenverfahrens mit Kurznotation:

q_i	$\delta(q_i, 0)$	$\delta(q_i, 1)$
q_0	$\emptyset$	q_0q_1
q_0q_1	q_1	$q_0q_1q_2$
q_1	q_1	q_1q_2
$q_0q_1q_2$	q_1	$q_0q_1q_2$
q_1q_2	q_1	q_1q_2
$\emptyset$	$\emptyset$	$\emptyset$

(4P)

Darstellung als Graph:



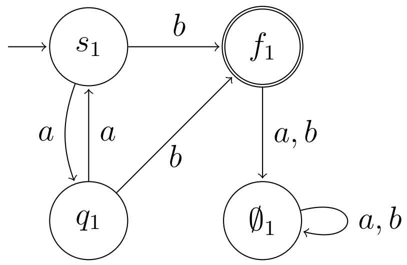
(1P)

4. Es muss F_B durch $F' = Q \setminus F_B$ ersetzt werden. (1P)

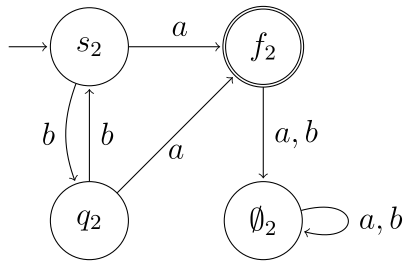
Aufgabe 3 (10 Punkte)

Wir betrachten die endlichen Automaten A_1 und A_2 über dem Alphabet $\Sigma = \{a, b\}$, die durch die folgenden Graphen gegeben sind:

A_1:



A_2:



1. Geben Sie einen regulären Ausdruck für $L(A_1)$ an und zeigen Sie $L(A_1) \cap L(A_2) = \emptyset$.

2. Geben Sie einen zu A_1 äquivalenten, minimalen DFA B_1 an und beweisen Sie dessen Minimalität.

   Der Minimalitätsbeweis kann auch durch Anwendung eines entsprechenden Konstruktionverfahrens für Quotientenautomaten geliefert werden.

3. Seien $\Gamma = \Sigma \cup \{\#\}$ und $L = L(A_1)\{\#\}L(A_2)$. Geben Sie einen minimalen DFA C an, so dass $L = L(C)$ gilt.

4. Auf Γ^* ist eine Äquivalenzrelation $\equiv_L$ wie folgt definiert: Für alle $u, v \in \Gamma^*$ gilt

$$u \equiv_L v \quad \Longleftrightarrow \quad \forall w \in \Gamma^*.\ uw \in L \Leftrightarrow vw \in L\,.$$

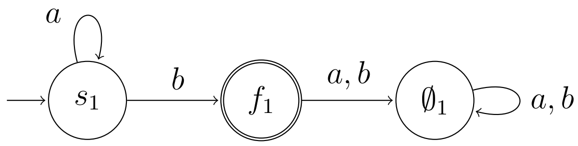
   Man zeige: Falls $u \in L(A_1)$ und $v \in L(A_2)$, dann gilt $u \not\equiv_L v$, d. h. u und v sind nicht äquivalent im Sinne von $\equiv_L$.

Lösung

Bepunktung vorbehaltlich der Änderung der Detailpunkte.

1. Es gilt $L(a^*b) = L(A_1)$ und $L(b^*a) = L(A_2)$. Wörter aus $L(A_1)$ enden mit b und Wörter aus $L(A_2)$ enden mit a. Daraus folgt $L(A_1) \cap L(A_2) = \emptyset$. (2P)

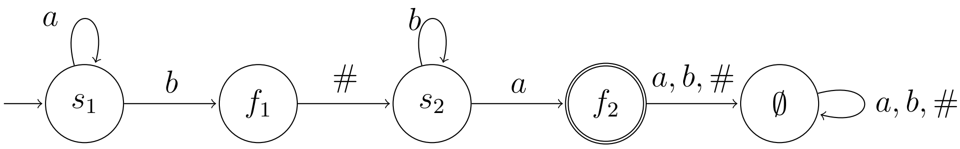
2.



(2P)

   f_1 ist als Endzustand nicht äquivalent zu Nichtendzuständen. Der Fangzustand $\emptyset$ ist nicht äquivalent zu s_1, weil $L(\emptyset) = \emptyset \neq L(A_1) = L(s_1)$. (1P)

3. Im folgenden Graph seien alle undefinierten Übergänge ergänzend zum Zustand $\emptyset$ geführt.



(3P)

4. Seien $u \in L(A_1)$ und $v \in L(A_2)$.

   Wir nehmen $u \equiv_L v$ an und führen diese Annahme wie folgt zum Widerspruch.

   Wir setzen $w = \#v$. Dann folgt $uw = u\#v \in L$.

   Und wegen $u \equiv_L v$ folgt nun auch $vw \in L$, mithin $v\#v \in L$.

   Nach Definition von L folgt $v \in L(A_1)$.

   Nach Teilaufgabe 1 gilt aber $L(A_1) \cap L(A_2) = \emptyset$. Widerspruch! (2P)

Aufgabe 4 (9 Punkte)

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

$$\begin{array}{lll} S \to AZ\,, & X \to b \mid XB\,, & B \to b\,, \\ Z \to SD \mid TD\,, & Y \to c \mid YC\,, & C \to c\,, \\ T \to XY\,, & A \to a\,, & D \to d\,. \end{array}$$

1. Zeigen Sie durch Anwendung des CYK-Verfahrens, dass a^2bc^2d nicht in der von G erzeugten Sprache enthalten ist, d. h. $a^2bc^2d \notin L(G)$.
2. Geben Sie eine Ableitung des Wortes a^2bcd^2 mit Produktionen der Grammatik G an.
3. Zeigen Sie, dass die Sprache $L = \{a^k b^m c^k d^m \mid k, m \in \mathbb{N}\backslash\{0\}\} \subseteq \{a, b, c, d\}^*$ nicht kontextfrei ist.

Lösung

Bepunktung vorbehaltlich der Änderung der Detailpunkte.

1.

16 $\emptyset$					
15 $\emptyset$	26 S				
14 $\emptyset$	25 $\emptyset$	36 T, Z			
13 $\emptyset$	24 $\emptyset$	35 T	46 Y		
12 $\emptyset$	23 $\emptyset$	34 T	45 Y	56 $\emptyset$	
11 A	22 A	33 B, X	44 C, Y	55 C, Y	66 D
a	a	b	c	c	d

(4P)

2. $S \to_G AZ \to_G ASD \to_G AAZD \to_G AATDD \to_G AAXYDD \to_G^* aabcdd\,.$ (1P)

3. Widerspruchsbeweis mit Pumping-Lemma:

   Sei n eine Pumping-Lemmazahl und $z = a^n b^n c^n d^n$.

   Es gilt $z \in L$. Damit gilt $z = uvwxy$ mit $|vwx| \leq n$ und $|vx| \geq 1$.

   Damit enthält vx entweder nicht c, d oder nicht a, d oder nicht a, b. vx enthält aber mindestens einen Buchstaben.

   Sei o.B.d.A. a in vx enthalten, und c, d seien nicht in vx enthalten.

   Nach Pumping-Lemma gilt $z' = uv^0wx^0y \in L$. In z' ist aber die Anzahl der c gleich n, wohingegen die Anzahl der a nun kleiner als n ist. Widerspruch, denn damit gilt $z' \notin L$. (4P)

Aufgabe 5 (8 Punkte)

Die <u>fallende Faktorielle</u> $f(n,k) = n^{\underline{k}}$ ist eine Funktion $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, für die $f(n,0) = 1$ für alle n, $f(n,k) = 0$ für alle $n < k$ und die folgende Rekursion für alle $n, k \in \mathbb{N}$ gelten:

$$f(n+1, k+1) = (n+1) \cdot f(n,k)\,.$$

1. Zeigen Sie ohne Verwendung von LOOP- und/oder WHILE-Programmen, dass die Fakultätsfunktion $n! = f(n,n)$ primitiv rekursiv ist. Benennen Sie die Regeln, die Sie für die Konstruktion verwenden.

2. Zeigen Sie, dass die fallende Faktorielle f primitiv rekursiv ist.

   Sie dürfen LOOP-Programme und die ganzzahlige Division $Div(n,m)$ verwenden.

3. Es sei die primitiv rekursive Funktion $h(n,m) = (n! \dot{-} m) + (m \dot{-} n!)$ gegeben und es sei μ der Operator zur Konstruktion μ-rekursiver Funktionen.

   Ist die Funktion $(\mu h)(m)$ primitiv rekursiv? Begründen Sie Ihre Antwort.

<u>Hinweis</u>: Sie dürfen aus der Vorlesung oder den Übungen bekannte Ergebnisse über primitiv rekursive Funktionen (z. B. erweiterte Komposition, erweitertes Rekursionschema, primitive Rekursivität der Addition "+", der eingeschränkten Subtraktion "$\dot{-}$", der Multiplikation "$\cdot$" und insbesondere der ganzzahligen Division $Div(n,m)$, ...) verwenden.

Lösung

Bepunktung vorbehaltlich der Änderung der Detailpunkte.

1. $n!$ wird im Folgenden mit dem erweiterten Rekursionsschema definiert. Die Konstruktion stützt sich auf die primitiv rekursive Konstante 1, die primitiv rekursive Addition und Multiplikation: (1P)

$$\begin{aligned} 0! &= 1\,, \\ (n+1)! &= (n+1) \cdot n!\,. \end{aligned} \qquad (1P)$$

2. LOOP-Programm oder direkt wie folgt:

$$f(n,k) = (1 \dot{-} (k \dot{-} n)) \cdot Div(n!, (n \dot{-} k)!)\,.$$

   (4P)

3. Nein! Die Funktion ist nicht total, weil sie für $m = 0$ nicht definiert ist. (2P)

Aufgabe 6 (10 Punkte)

Eine deterministische Turingmaschine $T = (Q, \Sigma, \Gamma, \delta, q_0, \Box, F)$ nennen wir eingeschränkt speicherfähig, wenn T bei der Abänderung einer Feldinschrift den Kopf nicht bewegt. Formal wird dies durch die folgende Eigenschaft E der Übergangsfunktion δ von T definiert, i.Z. $T \in E$:

(E) $\forall x, y \in \Gamma, p, q \in Q, d \in \{L, R, N\}. \quad [\, x \neq y \;\wedge\; \delta(p, x) = (q, y, d)\,] \implies d = N\,.$

1. Definieren Sie eine eingeschränkt speicherfähige Turingmaschine $T \in E$, die für jede nichtleere Eingabe $w = x_1 x_2 \ldots x_n$ mit Dezimalziffern $x_i \in \Sigma = \{0, 1, 2, \ldots, 9\}$ eine Berechnung durchführt, so dass bei Terminierung das Wort $w' = x_1 x_1 \ldots x_1$ der Länge n auf dem sonst leeren Band steht mit Kopf von T auf der letzten Ziffer von w'. Begründen Sie knapp Ihre Konstruktionsidee.

2. Geben Sie ein Verfahren an, das zu jeder deterministischen Turingmaschine $T = (Q, \Sigma, \Gamma, \delta, q_0, \Box, F)$ eine eingeschränkt speicherfähige Turingmaschine $T' = (Q', \Sigma, \Gamma, \delta', q_0, \Box, F) \in E$ liefert, so dass für die akzeptierten Sprachen $L(T) = L(T')$ gilt.

3. Wir definieren das Halteproblem $H_{0,E}$ auf leerem Band für eingeschränkt speicherfähige Turingmaschinen wie folgt: $H_{0,E} = \{w \in \{0,1\}^* \mid M_w[\epsilon]\!\downarrow \text{ und } M_w \in E\}$.

   Man zeige, dass $H_{0,E}$ auf $H_0 = \{w \in \{0,1\}^* \mid M_w[\epsilon]\!\downarrow\}$ reduzierbar ist, d.h., dass gilt:

   $$H_{0,E} \leq H_0\,.$$

Lösung

Bepunktung vorbehaltlich der Änderung der Detailpunkte.

1. Wir definieren zu jedem Element $x \in \Sigma$ einen (indizierten) Zustand q_x mit $q_x \neq q_y$ falls $x \neq y$.

   Seien $Q = \{q_0, f\} \cup \{q_x \mid x \in \Sigma\}$ mit $|Q| = |\Sigma| + 20$. Sei $F = \{f\}$.
   Wir definieren die Übergangsfunktion wie folgt:

   $$\begin{array}{rclrcll} \delta(q_0, \Box) & = & (f, \Box, L)\,, & \delta(q_0, x) & = & (q_x, x, R)\,, & \forall x \in \Sigma \\ \delta(q_x, \Box) & = & (f, \Box, L)\,, & \delta(q_x, y) & = & (q_0, x, N)\,, & \forall x, y \in \Sigma \end{array} \quad \text{(4P)}$$

2. Sei $Q' = Q \cup \{(q, x) \mid q \in Q, x \in \Gamma\}$. Wir schreiben q_x für (q, x) und definieren δ' wie folgt:

   Seien $q \in Q \setminus F$, $p \in Q$, $x, y \in \Gamma$, $d \in \{L, R, N\}$
   und es gelte $\delta(q, x) = (p, y, d)$. Dann und nur dann definieren wir δ', und zwar

   $$\delta'(q, x) = (p_x, y, N) \quad \text{und} \quad \delta'(p_x, y) = (p, y, d)\,. \quad \text{(3P)}$$

3. Die Eigenschaft E ist entscheidbar für Turingmaschinen, weil nur endlich viele Einträge der Übergangsfunktion geprüft werden müssen, d.h., dass die charakteristische Funktion χ_E der Menge $\{w \in \{0,1\}^* \mid M_w \in E\}$ berechenbar ist.

   Sei $M_{\widehat{w}}$ eine nirgends terminierende TM. Nun definieren wir eine berechenbare Funktion $f : \Sigma^* \to \Sigma^*$ durch

$$f(w) = \begin{cases} \widehat{w} : \chi_E(w) = 0\,, \\ w : \chi_E(w) = 1\,. \end{cases}$$

   Offenbar gilt $w \in H_{0,E} \Leftrightarrow f(w) \in H_0$. (3P)

Aufgabe 7 (8 Punkte)

Sei $\mathbb{N}$ die Menge der nichtnegativen ganzen Zahlen. Wir definieren eine unäre Operation „Komplement" (i.Z. "$^{-}$") wie folgt: $\overline{\mathrm{x}} := 0$, falls $x \neq 0$. Und $\overline{\mathrm{x}} := 1$, falls $x = 0$. Nun betrachten wir $\mathbb{N}$ zusammen mit der Addition "+" und der obigen Operation Komplement, d. h. wir betrachten die Algebra $(\mathbb{N}, +, ^{-})$. Sei außerdem $(\{true, false\}, \vee, \neg)$ die Boolesche Algebra mit Disjunktion $\vee$ und Negation $\neg$, aber ohne Konjunktion "$\wedge$".

Sei $\mathcal{F}$ die Menge aller arithmetischen Ausdrücke über beliebigen Variablen $x_1, x_2, \ldots$ und den Operationen + und Komplement $^{-}$. Der Wert $\sigma(F)$ eines Ausdrucks $F \in \mathcal{F}$ bei vorgelegter Belegung σ aller Variablen wird in üblicher Weise berechnet und notiert. Aus Gründen der Übersichtlichkeit schreiben wir $[F]_\sigma$ an Stelle von $\sigma(F)$ (Beispiel: Mit $\sigma(x_1) = 5$, $\sigma(x_2) = 0$ gilt $[x_1 + \overline{\mathrm{x}}_2]_\sigma = 6$.)

1. Ein Ausdruck $F \in \mathcal{F}$ kann durch textuelle Ersetzung von + durch $\vee$ und $^{-}$ durch $\neg$ im Formelbaum in einen Booleschen Ausdruck $F' = t(F)$ transformiert werden. (Beispiel: $t(x_2 + \overline{(x_1 + \overline{\mathrm{x}_3}\,)}\,) = x_2 \vee \neg(x_1 \vee \neg x_3)$.)

   Einer Belegung σ aller Variablen $x_1, x_2, \ldots$ mit Zahlen ordnen wir eine Belegung σ' mit Wahrheitswerten wie folgt zu: $\sigma'(x_i) := false$ bzw. $true$, falls $\sigma(x_i) = 0$ bzw. $\neq 0$ gelten.

   Man zeige durch strukturelle Induktion über den Aufbau von Ausdrücken für alle $F \in \mathcal{F}$ und Belegungen σ:

   $$[F]_\sigma = 0 \iff [F']_{\sigma'} = false\,. \tag{1}$$

2. Wir betrachten das Problem IstNull:

   Gegeben: $F \in \mathcal{F}$.

   Problem: Gibt es eine Belegung der Variablen mit nichtnegativen ganzen Zahlen, so dass F den Wert 0 annimmt?

   Zeigen Sie, dass SAT auf IstNull polynomiell reduzierbar ist, d. h.

   $$\text{SAT} \leq_p \text{IstNull}\,.$$

   Die Äquivalenz (1) darf dabei als bewiesen verwendet werden.

Lösung

Bepunktung vorbehaltlich der Änderung der Detailpunkte.

1. Regel für x_i: Nach Definition gilt $[x_i]_\sigma = 0 \iff [x_i]_{\sigma'} = false$.

   Regel für $^{-}$: $[\overline{F}]_\sigma = 0 \iff [F]_\sigma \neq 0 \iff$
   $[F']_{\sigma'} = true \iff [\neg F']_{\sigma'} = false \iff [t(\overline{F})]_{\sigma'} = false\,.$

   Regel für +: $[F_1 + F_2]_\sigma = 0 \iff [F_1]_\sigma = 0$ und $[F_2]_\sigma = 0 \iff$
   $[F_1']_{\sigma'} = false$ und $[F_2']_{\sigma'} = false \iff$
   $[F_1' \vee F_2']_{\sigma'} = false \iff [t(F_1 + F_2)]_{\sigma'} = false\,.$ (4P)

2. Sei $\mathcal{F}'$ die Menge aller Booleschen Ausdrücke über den Operationen $\vee$, $\wedge$ und $\neg$ sowie den Variablen $x_1, x_2, \ldots$. Dann sei s eine Operation, die $F' \in \mathcal{F}'$ zunächst in einen

äquivalenten Ausdruck F'' wandelt, die neben den Variablen nur Operationszeichen $\vee$ und $\neg$ enthält, und anschließend alle Operationsvorkommen von $\vee$ bzw. $\neg$ in die Operationsvorkommen $+$ und $\bar{\ }$ wandelt mit Ergebnis $s(F') \in \mathcal{F}$. s ist polynomiell zeitbeschränkt.

Wir definieren die Abbildung $f : \mathcal{F}' \to \mathcal{F}$ durch $f(F') = \overline{s(F')}$. Offenbar ist f polynomiell zeitbeschränkt und wegen Teilaufgabe 1 mit $t(s(F')) = F'$ auch Reduktionsabbildung. (4P)