Technische Universität München
Fakultät für Informatik
Prof. Tobias Nipkow, Ph.D.
Dr. Werner Meixner, Alexander Krauss

Sommersemester 2010
Lösungsblatt Wiederholungsklausur
15. Oktober 2010

Einführung in die Theoretische Informatik

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

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

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 180 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.
- Es sind keine Hilfsmittel außer einem handbeschriebenen DIN-A4-Blatt zugelassen.

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 im Folgenden Ihre Antworten möglichst knapp!
Sei im folgenden Σ ein Alphabet, und $a \in \Sigma$.

1. Die Menge aller μ-rekursiven Funktionen ist abzählbar.
2. Sei $t(n) = a(n, n)$ mit der Ackermann-Funktion a.
   Dann ist jede Sprache $L \in NTIME(t(n))$ entscheidbar.
3. Zu jedem ϵ-NFA gibt es einen äquivalenten NFA mit der gleichen Anzahl von Zuständen.
4. Für eine reguläre Sprache L ist in $O(|w|^2)$ Schritten entscheidbar, ob $w \in L$ liegt.
5. Jedes erzeugende Symbol in einer Grammatik ist nützlich.
6. Wenn $A \leq_p B$ und $B \leq_p A$ dann ist $A = B$.

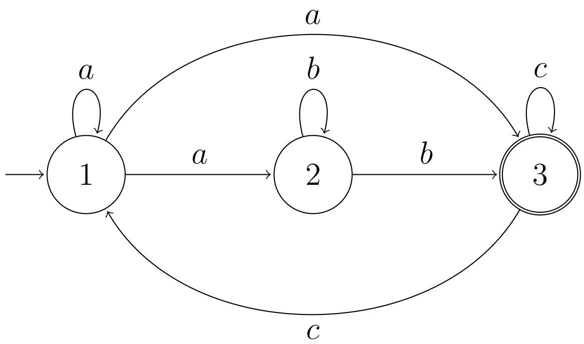
Lösungsvorschlag

1. (w) z.B. WHILE-Programme abzählen
2. (w) TM simulieren und nach t(n) Schritten abbrechen
3. (w) Die Konstruktion verändert die Zustandsmenge nicht, sondern fügt lediglich neue Übergänge hinzu.
4. (w) sogar in Linearzeit: DFA simulieren
5. (f) Es könnte z.B. nicht erreichbar sein.
6. (f) Wenn z.B. $A, B \in P$, dann gilt immer $A \leq_p B$ und $B \leq_p A$.

Richtige Antwort: 0,5 Punkte
Begründung auch richtig/sinnvoll: 0,5 Punkte

Aufgabe 2 (8 Punkte)

Seien $\Sigma = \{a, b, c\}$ und $A = (\{1, 2, 3\}, \Sigma, \delta, 1, \{3\})$ ein nichtdeterministischer endlicher Automat, der durch das folgende Übergangsdiagramm definiert ist:



1. Geben Sie einen regulären Ausdruck r_A an, der $L(A)$ darstellt.
2. Berechnen Sie mithilfe der Potenzmengenkonstruktion einen deterministischen endlichen Automaten B mit $L(B) = L(A)$.
3. Sei $\Sigma_C = \Sigma \cup \{d\}$. Geben Sie einen endlichen Automaten C an mit $L(C) = \Sigma_C^* \setminus L(A)$.

Lösungsvorschlag

1. $r_A = a^+ b^+ (c \mid c a^* b^+)^*$. (1 P.)
2. Berechnung δ_B:

$\delta_B(x, y)$	a	b	c
1	123	$\emptyset$	$\emptyset$
123	123	23	13
13	123	$\emptyset$	13
23	$\emptyset$	23	13
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

   Es gilt $B = (\{1, 13, 23, 123, \emptyset\}, \Sigma, \delta_B, 1, \{13, 23, 123\})$. (4 P.)

3. $C = (\{1, 13, 23, 123, \emptyset\}, \Sigma_C, \delta_C, 1, \{1, \emptyset\})$ und für alle $q \in Q_C$

$$\delta_C(q, x) = \begin{cases} \delta_B(q, x) : x \neq d \\ \emptyset : \text{sonst} \end{cases}$$

(3 P.)

Aufgabe 3 (8 Punkte)

Sei $\Sigma = \{a, b, c\}$. Die Anzahl der Vorkommen eines Buchstaben $x \in \Sigma$ in einem Wort $w \in \Sigma^*$ bezeichnen wir mit $\#_x(w)$. Wir betrachten die Sprache

$$L = \{w \in \Sigma^* \mid \#_a(w) = \#_b(w) \text{ oder } \#_b(w) = \#_c(w)\}\,.$$

1. Zeigen Sie, dass L nicht regulär ist.

2. Geben Sie eine kontextfreie Grammatik G an, die L erzeugt.

3. Zeigen Sie, dass das Wort $(abcba)^n$ für alle $n \geq 0$ ableitbar ist.

Lösungsvorschlag

1. Anwendung Pumping-Lemma:
   Seien n eine PL-Zahl und $z = a^n b^n$. Dann gilt $z \in L$. (1 P.)
   Seien $z = uvw$ mit $|uv| \leq n$, $|v| > 0$ und $\forall i \geq 0 : uv^i w \in L$. (1 P.)
   Dann gilt $uv \in \{a\}^+$ und $v = a^l$ mit $l > 0$.
   Es folgt $uv^0 w = a^{n-l} b^n \in L$
   im Widerspruch zur Definition. (1 P.)

2. Sei $G = (\Sigma, \{S, L, R\}, S, P)$ mit P wie folgt.

$$\begin{aligned} S &\to T \mid R \\ T &\to aTb \mid bTa \mid TT \mid c \mid \epsilon \\ R &\to bRc \mid cRb \mid RR \mid a \mid \epsilon\,. \end{aligned}$$

(3 P.)

3. $S \to_G T \to_G TT {\to_G}^* T^n$. (1 P.)
   $T {\to_G}^* TTT \to_G aTbTT \to_G abTT \to_G abcT \to_G abcbTa \to_G abcba\,.$ (1 P.)

Aufgabe 4 (10 Punkte)

Sei Σ ein Alphabet.

1. Zeigen Sie: Für beliebige reguläre Ausdrücke r, s über Σ gilt: $L(r(sr)^*) = L((rs)^*r)$.

2. Sei v ein Wort in Σ^*, welches wir das "verbotene Wort" nennen. Zur Vereinfachung nehmen wir an, dass in v kein Buchstabe doppelt vorkommt.

   Sei $k \in \mathbb{N}$. Beschreiben Sie, wie man DFAs konstruieren kann, die die Sprache

   (a) $L_0 = \{w \in \Sigma^* \mid w \text{ enthält kein Vorkommen des Teilworts } v\}$ bzw.

   (b) $L_k = \{w \in \Sigma^* \mid w \text{ enthält maximal } k \text{ Vorkommen des Teilworts } v\}$

   akzeptieren.

   Achtung: Sie sollen die Automaten nicht angeben, sondern lediglich präzise beschreiben, wie sie gefunden werden können. Bekannte Verfahren aus der Vorlesung können Sie dabei voraussetzen.

Lösungsvorschlag

1. Mit Induktion über n zeigen wir $A(BA)^n = (AB)^n A$ für beliebige $A, B \subseteq \Sigma^*$:
   Induktionsanfang: $A(BA)^0 = A = (AB)^0 A$.
   Induktionsschritt: $A(BA)^{n+1} = A(BA)^n BA = (AB)^n ABA = (AB)^{n+1} A$.

   Daraus folgt $A(BA)^* = (AB)^* A$. Die Gleichung über die regulären Ausdrücke ist eine Instanz dieser Aussage:

   $$L(r(sr)^*) = L(r)(L(s)L(r))^* = (L(r)L(s))^* L(r) = L((rs)^* r)$$

   (4P.)

2. (a) Wir konstruieren den regulären Ausdruck $r = \Sigma^* v \Sigma^*$, wobei Σ den Ausdruck bezeichnet, in dem alle Zeichen aus Σ mit $\mid$ verknüpft sind. Gesucht ist ein DFA, der das Komplement der Sprache akzeptiert. Hierzu konstruiert man einen DFA mit den üblichen Mitteln, und vertauscht am Ende Endzustände und Nicht-Endzustände. (3P.)

   (b) Wir konstruieren reguläre Ausdrücke r_k für $k \in \mathbb{N}$. Dabei beschreibt r_k genau die Wörter mit *mindestens* k Vorkommen von v:

   $$r_0 = \Sigma^* \qquad r_{n+1} = \Sigma^* v r_n$$

   Den DFA für die Sprache L_k erhält man nun indem man den Automaten für r_{k+1} konstruiert und wie oben das Komplement bildet. (3P.)

Aufgabe 5 (9 Punkte)

1. Sei $prodeven(n)$ das Produkt aller geraden Zahlen k mit $1 \leq k \leq n$. Es gelte $prodeven(0) = prodeven(1) = 1$.

   Zeigen Sie, dass *prodeven* primitiv rekursiv ist. Dabei darf die Funktion *even* mit $even(2k) = 1$ und $even(2k+1) = 0$ für $k \in \mathbb{N}$ als primitiv rekursiv vorausgesetzt werden.

2. Sei $g : \mathbb{N} \to \mathbb{N}$ primitiv rekursiv und $f : \mathbb{N} \to \mathbb{N}$ definiert durch die Startwerte $f(0) = 1$ und $f(1) = 3$ zusammen mit der Rekursion

   $$f(n) = f(n-1) + g(n) \cdot f(n-2) \quad \text{für alle} \quad n \in \mathbb{N} \text{ mit } n \geq 2\,.$$

   Zeigen Sie mit Hilfe der Paarfunktion $c : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, dass f primitiv rekursiv ist.

<u>*Hinweis*</u>:
Sie dürfen zusätzlich zu den Basisfunktionen der primitiven Rekursion die folgenden Funktionen als primitiv rekursiv annehmen: $plus(m,n)$ $(+)$, $times(m,n)$ $(\cdot)$, $dotminus(m,n)$ $(\dot{-})$, $pred(n)$, $c(m,n)$, $p_1(n)$, $p_2(n)$, $ifthen(n,a,b)$ und c_n^k (konstante k-stellige Funktion). Sie dürfen die erweiterte Komposition und das erweiterte rekursive Definitionsschema benützen. LOOP-Programme sind nicht erlaubt.

Lösungsvorschlag

1. $$\begin{aligned} prodeven(0) &= 1\,, \\ prodeven(n+1) &= \\ & ifthen(\,n, ifthen(\,even(n), prodeven(n), (n+1)\cdot prodeven(n)\,), 1\,)\,. \end{aligned}$$

   (4 P.)

2. Sei $k(n) = c(f(n), f(n+1))$. (1 P.)

   Dann gilt

   $$\begin{aligned} k(0) &= c(1,3)\,, \\ k(n+1) &= c(\,p_2(k(n)),\ p_2(k(n)) + g(n+2) \cdot p_1(k(n))\,)\,. \end{aligned}$$

   Mithin ist k primitiv rekursiv. (3 P.)

   Wegen $f(n) = p_1(k(n))$ ist damit auch f primitiv rekursiv. (1 P.)

Aufgabe 6 (10 Punkte)

1. Sei $\Sigma = \{0, 1\}$. Beweisen oder widerlegen Sie:
   (a) Wenn L_1 und L_2 rekursiv aufzählbar sind, dann ist auch $L_1 \setminus L_2$ rekursiv aufzählbar.
   (b) Die Menge $A = \{w \in \Sigma^* \mid \forall v \in \Sigma^*.\, M_w(v) = vv\}$ ist entscheidbar.
   (c) Die Menge $B = \{w \in \Sigma^* \mid \forall v \in \Sigma^*.\, M_w(v) = \chi_{PCP}(v)\}$ ist entscheidbar.

   Dabei sei $PCP \subseteq \Sigma^*$ die Menge der lösbaren Instanzen des Postschen Korrespondenzproblems, codiert als Wörter über Σ.
2. Geben Sie eine Funktion $f : \Sigma^* \to \Sigma^*$ an, so dass f nicht berechenbar ist und für die $\forall w.\, |f(w)| \geq |w|$ gilt. Beweisen Sie diese Eigenschaften.

Lösungsvorschlag

1. (a) Falsch. Gegenbeispiel: $L_1 = \{0, 1\}^*$, $L_2 = K$. Dann wäre $\Sigma^* \setminus K = \overline{K}$ rekursiv aufzählbar und damit wäre K entscheidbar. Statt K kann man jede Menge verwenden die rekursiv aufzählbar aber nicht entscheidbar ist. (2P.)
   (b) Falsch. Wir verwenden den Satz von Rice. Wir verifizieren die Voraussetzungen: Offensichtlich existiert eine TM, die stets ihre Eingabe verdoppelt. Ebenfalls hat nicht jede TM dieses Verhalten. Damit hat B die Form $B = \{w \mid \varphi_w \in F\}$ mit $F = \{f \mid \forall v.\, f(v) = vv\}$ und ist somit laut Satz von Rice unentscheidbar. (3P.)
   (c) Richtig. Das PCP ist bekanntermaßen unentscheidbar. Somit ist die charakteristische Funktion χ_{PCP} nicht berechenbar und es existiert keine TM, die die Forderung erfüllt. Damit ist $B = \emptyset$ und somit trivialerweise entscheidbar.(2P.)
2. Wir betrachten die Funktion χ_K für das spezielle Halteproblem K. Um auch die Längenbedingung zu erfüllen definieren wir f mit $f(w) = \chi_K(w)w$, hängen also die Eingabe zusätzlich an die Ausgabe an. Es bleibt zu zeigen, dass f nicht berechenbar ist. Wäre dies der Fall, dann könnte man aus dem Ergebnis von f nur das erste Zeichen behalten und somit χ_K berechnen. Das ist aber ein Widerspruch.
   Statt χ_K funktioniert hier auch eine beliebige andere nicht-berechenbare Funktion. (3P.)

Aufgabe 7 (9 Punkte)

Eine Turingmaschine heißt *N-frei*, wenn $\delta(q,x) \in Q \times \Gamma \times \{L,R\}$ für alle $q \in Q$ und $x \in \Gamma$.

1. Zeigen Sie, dass es für jede Turingmaschine M eine N-freie Turingmaschine M' gibt, so dass für alle $q, q' \in Q$ und $\alpha, \alpha', \beta, \beta' \in \Gamma^*$ gilt:

$$(\alpha, q, \beta) \rightarrow_M^* (\alpha', q', \beta') \iff (\alpha, q, \beta) \rightarrow_{M'}^* (\alpha', q', \beta') \; . \qquad (*)$$

   Gehen Sie dabei wie folgt vor:

   (a) Beschreiben Sie informell ihre Konstruktionsidee.

   (b) Geben Sie die Konstruktion von M' formal an.

2. Gibt es analog auch stets eine L-freie Maschine, so dass $(*)$ gilt? Begründung!

Lösungsvorschlag

1. (a) Idee: Immer dann wenn die TM eigentlich stehenbleiben soll (N), geht sie stattdessen nach rechts und im nächsten Schritt sofort wieder nach links. Dafür benötigt man für jeden Zustand q einen zusätzlichen Zustand $\hat{q}$, der besagt "gehe einen Schritt nach links und dann nach q". (4P.)

   (b) Wir arbeiten syntaktisch mit nichtdeterministischen Turingmaschinen. Der Nichtdeterminismus wird aber nicht wirklich benötigt.
   Sei $M = (Q, \Sigma, \Gamma, \delta, q_0, \Box, F)$. Wir definieren $M' = (Q', \Sigma, \Gamma, \delta', q_0, \Box, F)$. Dabei ist $Q' = Q \cup \{\hat{q} \mid q \in Q\}$ und

$$\begin{aligned} \delta'(q,x) &= \delta(q,x) \cap (Q \times \Gamma \times \{L,R\}) \\ &\quad \cup \{(\hat{r}, y, R) \mid (r, y, N) \in \delta(q,x)\}, \\ \delta'(\hat{q}, x) &= \{(q, x, L)\} \; . \end{aligned}$$

   (4P.)

   Alternative Konstruktion:

   (a) Idee: Bei einem N-Schritt liegt der nächste Schritt bereits fest, da Zustand und Bandzeichen unter dem Kopf schon bekannt sind. Man ersetzt also einen N-Übergang durch alle L- und R-Übergänge, die als nächstes auftreten können. Neue Zustände müssen dann keine eingeführt werden.

   (b) Wir definieren die Hüllenoperation $\mathrm{Cl} : Q \times \Gamma \to \mathcal{P}(Q \times \Gamma)$ induktiv:

   - $(q, x) \in \mathrm{Cl}(q, x)$.
   - $(q', x') \in \mathrm{Cl}(q, x) \;\wedge\; (q'', x'', N) \in \delta(q', x') \implies (q'', x'') \in \mathrm{Cl}(q, x)$.

Intuitiv gibt $\mathrm{Cl}(q,x)$ die Kombinationen aus Bandzeichen und Zustand an, die ausschließlich durch N-Schritte entstehen können. Diese Menge ist endlich.

Nun entfernen wir zunächst alle N-Übergänge aus δ:

$$\delta'(q,x) = \delta(q,x) \cap Q \times \Gamma \times \{L, R\} .$$

Die Funktion δ'' ergibt sich nun als:

$$\delta''(q,x) = \bigcup_{(q',x') \in \mathrm{Cl}(q,x)} \delta'(q',x')$$

Man beachte die Ähnlichkeit zur ϵ-Elimination bei NFAs.

Eine dritte Variante ist ähnlich wie die erste, nur das stattdessen der alte Zustand auf dem Band gemerkt wird. Dann wird nur ein neuer Zustand benötigt, dafür muss aber das Bandalphabet erweitert werden.

2. Eine TM, die nie nach links gehen könnte, hätte nach einem Rechts-Schritt niemals die Möglichkeit, ein geschriebenes Zeichen wieder zu lesen. (1P.)