Technische Universität München
Fakultät für Informatik
Lehrstuhl Informatik XIV
Prof. Dr. Ernst W. Mayr
Dr. Werner Meixner

Sommersemester 2014
Lösungen der Klausur
24. Juli 2014

Theoretische Informatik

Name	Vorname	Studiengang	Matrikelnummer
....	....	☐ Bachelor ☐ Inform. ☐ 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 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.

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

Max P	**A1** 4	**A2** 9	**A3** 10	**A4** 9	**A5** 10	**A6** 9	**A7** 9	Σ 60	Korrektor
Erstkorr.									
Zweitkorr.									

Aufgabe 1 (4 Punkte)

Wahr oder falsch? Begründen Sie Ihre Antwort!

1. Falls eine Grammatik Chomsky-Normalform besitzt, dann enthält sie keine nutzlosen Variablen.

2. Falls $L \subseteq \Sigma^*$ deterministisch kontextfrei ist, dann gibt es eine $LR(k)$ Grammatik, die das Komplement $\overline{L} = \Sigma^* \setminus L$ erzeugt.

3. Sei $\Sigma = \{0, 1\}$. Das Komplement $\overline{H_s} = \Sigma^* \setminus H_s$ des speziellen Halteproblems H_s ist eine Typ-0-Sprache. (H_s wurde in Übungen auch als K bezeichnet.)

4. Die Menge $\{w \in \{0, 1\}^* \,;\, \varphi_w \text{ ist } \mu\text{-rekursiv}\}$ ist entscheidbar. Dabei ist φ_w die von der Turingmaschine M_w berechnete Funktion.

Lösung

Für die richtige Antwort und für die richtige Begründung gibt es jeweils einen $\frac{1}{2}$ Punkt.

1. Falsch! Es können neue, nutzlose Variable mit entsprechenden Produktionen hinzugefügt werden.

2. Wahr! Die Klasse der deterministisch kontextfreien Sprachen ist abgeschlossen für Komplementbildung. Zu jeder DCFL gibt es eine erzeugende $LR(k)$ Grammatik.

3. Falsch! $\overline{H_s}$ ist nicht semi-entscheidbar, weil H_s nicht entscheidbar, aber semi-entscheidbar ist.

4. Wahr! Für alle w ist φ_w berechenbar und folglich μ-rekursiv.

Aufgabe 2 (9 Punkte)

Sei $\Sigma = \{a, b\}$. Für beliebige Sprachen $R, L \subseteq \Sigma^*$ ist der Rechtsquotient R/L definiert durch

$$R/L := \{x \in \Sigma^* ;\ (\exists y \in L)[\, xy \in R \,]\} .$$

Hinweis: Wenden Sie im Folgenden wenn möglich bekannte Sätze an.

1. Seien $R \subseteq \Sigma^*$ und $R_{-2} = \{x \in \Sigma^* ;\ (\exists y \in \Sigma^*)[\, |y| = 2 \wedge xy \in R \,]\}$. Man zeige:

   Falls R regulär ist, dann ist auch R_{-2} regulär.

2. Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein deterministischer endlicher Automat, der die Sprache $R := L(A)$ akzeptiert.

   Beschreiben Sie explizit, ausgehend von A, einen DFA oder NFA $A' = (Q', \Sigma, \delta', q'_0, F')$, der R_{-2} akzeptiert.

3. Seien $L \subseteq \Sigma^*$ unentscheidbar und $R \subseteq \Sigma^*$ regulär. Zeigen Sie die Entscheidbarkeit von R/L.

Lösung

Vorbehaltlich einer Punktedetaillierung:

1. Mit $L = \{y \in \Sigma^* ;\ |y| = 2\}$ gilt $R_{-2} = R/L$. (2P)

   Nach Satz der Vorlesung ist R/L regulär. (1P)

2. A' sei identisch mit A bis auf die Menge der Endzustände:

   $Q' = Q$, $\delta' = \delta$, $q'_0 = q_0$ und

   $$F' = \{q \in Q ;\ (\exists x, y \in \Sigma)[\, \hat{\delta}(q, xy) \in F \,]\}$$

   (4P)

3. Nach Satz der Vorlesung ist R/L regulär für beliebiges L. Dies schließt unentscheidbare L ein. (2P)

Aufgabe 3 (10 Punkte)

Seien $\Sigma \neq \emptyset$ und $V = \{A_1, A_2, \ldots, A_n\}$ Zeichenmengen mit $n \geq 2$ und m eine *Markierungsabbildung* der Form $m(x) = \hat{x}$ bzw. $m(A) = \widehat{A}$ für alle $x \in \Sigma$ bzw. $A \in V$. Wir definieren $\widehat{\Sigma} = \{\hat{x}\,; x \in \Sigma\}$ und $\widehat{V} = \{\widehat{A}_1, \widehat{A}_2, \ldots, \widehat{A}_n\}$. Wir setzen Mengendisjunktheit voraus, so dass $|\Sigma \cup \widehat{\Sigma} \cup V \cup \widehat{V}| = 2(n + |\Sigma|)$ gilt, und definieren $\Sigma' = \Sigma \cup \widehat{\Sigma}$ und $V' = V \cup \widehat{V}$.

Wir sagen, dass eine kontextfreie Grammatik $G' = (V', \Sigma', P', S')$ eine *Wortendemarkierung* generiert, falls S' eines der markierten Zeichen $\widehat{A}_i$, $i = 1, \ldots, n$ ist und jede Produktion aus P' eine der folgenden Formen besitzt (mit $x \in \Sigma$):

$$\begin{array}{lllcllll} A_i & \to & A_j A_k\,, & \qquad & A_i & \to & x\,, \\ \widehat{A}_i & \to & A_j \widehat{A}_k\,, & & \widehat{A}_i & \to & \hat{x}\,. \end{array}$$

1. Sei G' eine kontextfreie Grammatik, die eine Wortendemarkierung generiert. Man zeige mit <u>struktureller Induktion</u> für alle Wörter w der Sprache $L(G')$ die folgende Eigenschaft

   $$\widehat{P}(w): \quad \text{Es gibt ein } v \in \Sigma^* \text{ und ein } \hat{x} \in \widehat{\Sigma}, \text{ so dass } \; w = v\hat{x} \; \text{ gilt.}$$

   Betrachten Sie dazu geeignete Eigenschaften $P(w)$ bzw. $\widehat{P}(w)$ der aus Variablen $A \in V$ einerseits bzw. $\widehat{A} \in \widehat{V}$ andererseits ableitbaren Wörter $w \in \Sigma'^*$. Verwenden Sie die Bezeichnung $L(X) = \{w \in \Sigma'^*\,; X \xrightarrow[G']{}^* w\}$ für $X \in V'$.

2. Seien L eine kontextfreie Sprache, so dass $\epsilon \notin L$, und $E = \{x \in \Sigma^*\,; |x| = 1\}$.

   Zeigen Sie, dass der Rechtsquotient L/E kontextfrei ist. Zum Nachweis genügt eine informelle Konstruktionsbeschreibung einer kontextfreien Grammatik für L/E.

Lösung

Vorbehaltlich einer Punktedetaillierung:

1. Sei $P(w)$ die Eigenschaft: Es gilt $w \in \Sigma^*$.

   <u>Induktionsanfang:</u>
   Regel $A_i \to x$: Für $w = x$ gilt $P(w)$. (Klar!)
   Regel $\widehat{A}_i \to \hat{x}$: Für $w = \hat{x}$ gilt $\widehat{P}(w)$. (Klar!)

   <u>Induktionsschluss:</u>
   Regel $A_i \to A_j A_k$: Aus $w_j \in L(A_j) \wedge P(w_j)$ und $w_k \in L(A_k) \wedge P(w_k)$ folgt $w_i = w_j w_k \in L(A_i) \wedge P(w_i)$.
   <u>Beweis:</u> $w_j, w_k \in \Sigma^*$ impliziert $w_i = w_j w_k \in \Sigma^*$. Es folgt $P(w_i)$.

   Regel $\widehat{A}_i \to A_j \widehat{A}_k$: Aus $w_j \in L(A_j) \wedge P(w_j)$ und $w_k \in L(\widehat{A}_k) \wedge \widehat{P}(w_k)$ folgt $w_i = w_j w_k \in L(\widehat{A}_i) \wedge \widehat{P}(w_i)$.
   <u>Beweis:</u> $w_j \in \Sigma^*$ und $w_k = v\hat{x}$ impliziert $w_i = w_j w_k = w_j v \hat{x}$ mit $w_j v \in \Sigma^*$ und $\hat{x} \in \widehat{\Sigma}$. Es folgt $\widehat{P}(w_i)$.

   Da sich jedes Wort $w \in L(G')$ aus $S' \in \widehat{V}$ ableiten lässt, folgt $\widehat{P}(w)$. (6P)

2. Wenn in allen Wörtern $w \in L$ der letzte Buchstabe gestrichen wird, dann erhält man L/E. Wir konstruieren eine kontextfreie Grammatik, die L/E erzeugt, wie folgt:

   Sei G eine kontextfreie Grammatik mit $L = L(G)$ in Chomsky-Normalform. Wir ergänzen G zu einer Grammatik G', die eine Wortendemarkierung generiert, d.h., dass alle letzten Buchstaben von Wörtern w in L markiert werden.

   Dies geschieht durch

   Hinzufügen von $\widehat{\Sigma}$, $\widehat{V}$,
   Ersetzung von S durch $\widehat{S}$,
   Hinzufügen von Produktionen $\widehat{A}_i \to A_j\widehat{A}_k$ zu jeder Produktion $A_i \to A_jA_k$ und
   Hinzufügen von Produktionen $\widehat{A}_i \to \hat{x}$ zu jeder Produktion $A_i \to x$.

   Um die Grammatik für L/E zu gewinnen, werden alle Produktionen der Form $\widehat{A}_i \to \hat{x}$ von G' ersetzt durch $\widehat{A}_i \to \epsilon$. Dadurch entfällt $\widehat{\Sigma}$.

   Die erhaltene Grammatik G'' erzeugt L/E, wobei sich G'' nach Satz der Vorlesung durch Elimination der ϵ-Produktionen in eine kontextfreie Grammatik umwandeln lässt.

   (4P)

Aufgabe 4 (9 Punkte)

Seien $\Sigma = \{a, b\}$ und $L = \{a^n b^m a^n \,;\, m, n \in \mathbb{N}\}$. (Beachte: $0 \notin \mathbb{N}$.)

1. Definieren Sie einen deterministischen Kellerautomaten $K = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, der die Sprache L mit Endzustand akzeptiert, so dass also $L(K) = L$ gilt!
   Geben Sie dazu den Übergangsgraphen Ihres Automaten K an.

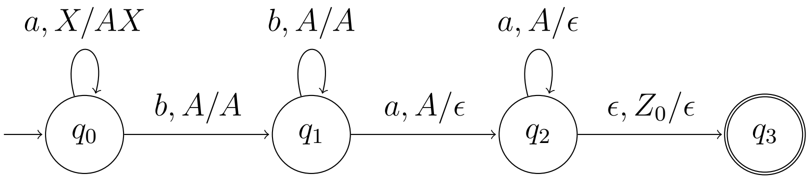
2. Zeigen Sie mit Hilfe des Pumping-Lemmas, dass kein NFA existiert, der L akzeptiert.

Lösung

Vorbehaltlich geänderter Punktedetaillierung:

1. Seien $Q = \{q_0, q_1, q_2, q_3\}$, $\Delta = \{Z_0, A\}$, $F = \{q_3\}$.

   Für alle $X \in \Delta$:



(5P)

2. Angenommen L sei regulär.

   Sei N eine Pumping-Lemma-Zahl für L und $z = a^N b a^N$ mit $z = uvw$, so dass $|uv| \leq N$, $v \neq \epsilon$ und für alle $i \in \mathbb{N}_0$ $uv^i w \in L$ gilt.

   Es folgt $uv \in a^+$, insbesondere $v = a^k$ für ein $k \in \mathbb{N}$.

   Damit folgt für $i = 0$ $z_i := uv^i w = a^{N-k} b a^N \in L$.

   Widerspruch, wegen $a^{N-k} \neq a^N$!

(4P)

Aufgabe 5 (10 Punkte)

Gegeben sei eine kontextfreie Grammatik G mit Startsymbol S in Chomsky-Normalform mit der folgenden Produktionenmenge:

$$\begin{array}{lclll} S & \to & UV \mid UA\,, & U \to BT\,, & V \to SA\,, \\ T & \to & SB \mid BT \mid b\,, & B \to b\,, & A \to a\,. \end{array}$$

1. Geben Sie eine Ableitung für $S \xrightarrow[G]{}^{*} BSBA$ an und zeigen Sie, dass $b^n(ba)^n \in L(G)$ für alle $n \in \mathbb{N}$ gilt.

2. Zeigen Sie durch Anwendung des CYK-Algorithmus, dass $T \xrightarrow[G]{}^{*} bbbab$ gilt und dass es von T aus mindestens zwei verschiedene Linksableitungen für $w = bbbab$ gibt. Gilt $w \in L(G)$?

3. Ist die Grammatik G eindeutig? Begründung!

Lösung

Vorbehaltlich geänderter Punktedetaillierung:

1. $S \to UA \to BTA \to BSBA$.

   Durch Iteration folgt:

   $S \to BSBA \to BBSBABA \to \ldots \to B^nS(BA)^n \xrightarrow[G]{}^{*} b^nS(ba)^n$.

   Außerdem gilt $S \to UA \to BTA \xrightarrow[G]{}^{*} bba$. (3P)

2.

15 U, T_{BT}, T_{SB}				
14 S	25 T			
13 U, T	24 S	35 $\emptyset$		
12 U, T	23 U, T	34 $\emptyset$	45 $\emptyset$	
11 B, T	22 B, T	33 B, T	44 A	55 B, T
b	b	b	a	b

(3P)

   Es gilt $T \to BT \xrightarrow[G]{}^{*} bbbab$ und $T \to SB \xrightarrow[G]{}^{*} bbbab$. (1P)

   S kommt nicht im Feld (15) vor, d.h., dass $bbbab$ nicht aus S ableitbar ist. (1P)

3. Nein! Denn T ist ein nützliches Symbol: es gilt $S \xrightarrow[G]{}^{*} BTA \xrightarrow[G]{}^{*} bwa \in L(G)$. Damit besitzt $w' = bbbbaba$ mindestens zwei Linksableitungen. (2P)

Aufgabe 6 (9 Punkte)

Sei $\Sigma = \{*, \#\}$. Wir kodieren ganze Zahlen $n \in \mathbb{N}_0$ als Folge $** \dots *$ der Länge n, d. h. $| ** \dots * | = n$, und stellen Paare $(x, y) \in \{*\}^* \times \{*\}^*$ als Wort $x\#y \in \Sigma^*$ dar.
Wir betrachten für $x, y, z \in \{*\}^*$ die Addition $|z| = |x| + |y|$.

1. Definieren Sie durch Angabe der Übergangsfunktion δ eine linear beschränkte Turingmaschine $M = (Q, \Sigma, \Gamma, \delta, q_0, \Box, F)$, die für $x, y, z \in \{*\}^*$ die Addition $|z| = |x| + |y|$ wie folgt durchführt:

   Startkonfiguration: $(\epsilon, q_0, x\#y)$. Endkonfiguration: (ϵ, q_e, z), mit $q_e \in F$.

   Es gilt: $(\epsilon, q_0, x\#y) \underset{M}{\rightarrow}^* (\epsilon, q_e, z)$.

   Beschreiben Sie kurz die Konstruktionsidee für Ihre Maschine.

2. Seien c_1, c_2 die Umkehrfunktionen einer Paarfunktion $c : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$. Dann ist $plus : \mathbb{N}_0 \to \mathbb{N}_0$ mit $plus(n) = c_1(n) + c_2(n)$ die Kodierung der Addition nichtnegativer ganzer Zahlen, d.h., $x + y = plus(c(x, y))$ für alle $x, y \in \mathbb{N}_0$.

   Zeigen Sie die Unentscheidbarkeit der folgenden Menge P:

   $$P = \{w \in \{0,1\}^* \,;\, \text{die von } M_w \text{ berechnete Funktion ist gleich } plus\}\,.$$

3. Sei $H_0 = \{w \in \{0,1\}^* \,;\, M_w \text{ hält auf leerem Band}\}$ das Halteproblem auf leerem Band. Zeigen Sie durch informelle Spezifikation einer Reduktionsabbildung f (wie in entsprechenden Beweisen der Vorlesung), dass H_0 reduzierbar ist auf P, i. Z. $H_0 \leq P$.

Lösung

Vorbehaltlich geänderter Punktedetaillierung:

1. Idee: Falls $x \neq \epsilon$, dann wird das erste Zeichen $*$ in x gelöscht und das Zeichen $\#$ durch $*$ ersetzt. Falls $x = \epsilon$, dann wird nur $\#$ gelöscht. Schließlich wird der Kopf nach vorne positioniert.

   Seien $Q = \{q_0, q_1, q_2, q_e\}$, $\Delta = \{*, \#, \Box\}$ und $F = \{q_e\}$.

   $$\begin{array}{rclrcl} \delta(q_0, *) &=& (q_1, \Box, R)\,, & \delta(q_0, \#) &=& (q_e, \Box, R)\,, \\ \delta(q_1, *) &=& (q_1, *, R)\,, & \delta(q_1, \#) &=& (q_2, *, L)\,, \\ \delta(q_2, *) &=& (q_2, *, L)\,, & \delta(q_2, \Box) &=& (q_e, \Box, R)\,. \end{array} \quad (5P)$$

2. Sei $S = \{plus\}$ die einelementige Menge von berechenbaren Funktionen $\mathbb{N}_0 \to \mathbb{N}_0$, mit dem berechenbaren Element $plus$.

   Es gilt $S \neq \emptyset$ und S ungleich der Menge aller Funktionen. Damit ist der Satz von Rice anwendbar, der beweist, dass P unentscheidbar ist. (2P)

3. Sei w_p der Code einer Turingmaschine M_{w_p}, die $plus$ berechnet.

   Für alle $w \in \{0,1\}^*$ sei $f(w)$ der Code einer Turingmaschine M, die wie folgt definiert ist:

M simuliert eine 2-Band-Turingmaschine, die die Eingabe von M auf Band 1 schreibt und anschließend auf Band 2 die Turingmaschine M_w auf leerem Band ausführt.

Falls M_w hält, dann wird auf Band 1 die Eingabe von M mit M_{w_p} ausgeführt und das Ergebnis auf das Band von M geschrieben.

Offenbar ist f eine totale und berechenbare Funktion, so dass $f(H_0) \subseteq P$ und $f(\overline{H_0}) \subseteq \overline{P}$ gelten. (2P)

Aufgabe 7 (9 Punkte)

1. Sei $g : \mathbb{N}_0 \to \mathbb{N}_0$ total und μ-rekursiv, und sei $f : \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch die Startwerte $f(0) = 1$ und $f(1) = 2$ zusammen mit der Rekursion

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

   Zeigen Sie die μ-Rekursivität der Funktion f, indem Sie die Erzeugungsregeln für μ-rekursive Funktionen zusammen mit einer Paarfunktion $c : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ und deren Umkehrfunktionen c_1 und c_2 anwenden.

   Hinweis: Sie dürfen zusätzlich zu den Basisfunktionen der primitiven Rekursion die folgenden Funktionen als primitiv rekursiv annehmen: $plus(m,n)$ $(+)$, $times(m,n)$ $(\cdot)$, $pred(n)$, $c(m,n)$, $c_1(n)$, $c_2(n)$ und die konstante k-stellige Funktion c_n^k. Sie dürfen die erweiterte Komposition und das erweiterte rekursive Definitionsschema benützen. LOOP- und WHILE-Programme sind nicht erlaubt.

2. Sei $f : \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch die Startwerte $f(0) = 1$ und $f(1) = 2$ zusammen mit der Rekursion

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

   Zeigen Sie, dass f primitiv-rekursiv ist, indem Sie f durch ein LOOP-Programm darstellen. *IF THEN ELSE* Konstrukte sowie arithmetische Operationen dürfen verwendet werden.

Lösung

Vorbehaltlich geänderter Punktedetaillierung:

1. Sei $k(n) = c(f(n), f(n+1))$.
   Dann gilt

$$\begin{aligned} k(0) &= c(1,2)\,, \\ k(n+1) &= c(\,c_2(k(n)),\ g(n+2) + c_2(k(n)) \cdot c_1(k(n))\,)\,. \end{aligned}$$

   Mithin ist k μ-rekursiv.
   Wegen $f(n) = c_1(k(n))$ ist damit auch f μ-rekursiv. (4P)

2. Das folgende LOOP-Programm basiert auf den Variablen $x_0, \ldots, x_5$.
   In x_0 wird das Ergebnis $f(n)$ ausgegeben, x_1 enthält beim Start das Argument n.

$x_2 := x_1 - 1;$ $x_3 := 1;$ $x_4 := 2;$
LOOP x_2 *DO*
  $x_5 := x_4 * x_3;$
  $x_5 := x_5 + 1;$
  $x_3 := x_4;$ $x_4 := x_5;$
END;
$x_0 := x_5$
IF $x_1 = 0$ *THEN* $x_0 := 1$ *END*;
IF $x_1 = 1$ *THEN* $x_0 := 2$ *END*

   Wenn wir den Satz der Vorlesung benutzen, dann folgt aus der Existenz eines LOOP-Programms für die Funktion f die primitive Rekursivität von f. (5P)