Lehrstuhl für Informatik 1
Prof. Dr. Gerhard Woeginger
Tim Hartmann, Daniel Neuen

WS 2017/18
19.02.2018

Klausur
Berechenbarkeit und Komplexität

Gelbes Papier

Lösungsvorschlag

Name: ....

Vorname: ....

Matrikelnummer: ....

Studiengang: ....

Hinweise:

- Die Bearbeitungszeit beträgt 120 Minuten.
- Bitte versehen Sie jedes Blatt mit Namen und Matrikelnummer.
- Bitte schreiben Sie deutlich. Unleserliches wird nicht korrigiert und als fehlerhaft gewertet.
- Streichen Sie Konzeptrechnungen, die nicht gewertet werden sollen, durch oder machen Sie sie anderweitig kenntlich. Bei mehreren Lösungsversuchen pro Aufgabe wird der schlechteste gewertet.
- Bitte verwenden Sie einen dokumentenechten Stift mit blauer oder schwarzer Tinte und verwenden Sie keinen Tintenkiller oder Ähnliches. Benutzen Sie ausschließlich das zur Verfügung gestellte Papier.
- Halten Sie bitte Ihren Studierendenausweis und einen Lichtbildausweis zur Kontrolle bereit.
- Bitte schalten Sie Ihre Mobiltelefone aus!

Ich versichere, die Klausur selbstständig bearbeitet zu haben, und mir ist bekannt, dass die Klausur bei einem Täuschungsversuch mit "nicht bestanden" bewertet wird.

....
(Unterschrift)

(Name/Matrikelnummer)

Aufgabe 1:

(a) Eine (deterministische, 1-Band) Turingmaschine ist durch das 7-Tupel $(Q, \Sigma, \Gamma, B, q_0, \bar{q}, \delta)$ definiert. Geben Sie Definitionsmenge und Bildmenge der Überführungsfunktion δ an. **(3 Punkte)**

Definitionsmenge $= Q \setminus \{\bar{q}\} \times \Gamma$; Bildmenge $= Q \times \Gamma \times \{R, L, N\}$

(b) Definieren Sie die Laufzeitkosten für einen Rechenschritt auf der RAM im logarithmischen Kostenmaß. **(3 Punkte)**

Im logarithmischen Kostenmaß sind die Laufzeitkosten eines Schrittes proportional zur binären Länge der Zahlen in den angesprochenen Registern.

(c) Formulieren Sie das zehnte Hilbert'sche Problem. **(3 Punkte)**

Man beschreibe einen Algorithmus, der entscheidet, ob ein gegebenes Polynom mit ganzzahligen Koeffizienten eine ganzzahlige Nullstelle hat.

(d) Formulieren Sie das Entscheidungsproblem **SUBSET-SUM**. **(3 Punkte)**

Eingabe: Positive ganze Zahlen $a_1, \ldots, a_n$ und b.
Frage: Existiert eine Indexmenge $I \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in I} a_i = b$?

(Name/Matrikelnummer)

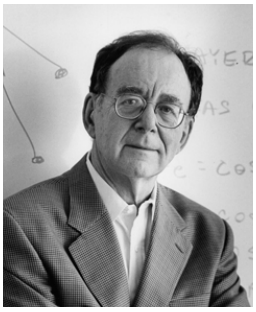
(e) Definieren Sie die Komplexitätsklasse **coNP**. **(3 Punkte)**

coNP ist die Klasse aller (Entscheidungs)probleme, deren Komplement durch eine NTM M erkannt wird, deren (Worst Case) Laufzeit polynomiell beschränkt ist.

(f) Geben Sie (ohne weitere Begründung) ein Entscheidungsproblem an, das entweder in **EXPTIME** oder in der Klasse der rekursiv aufzählbaren Sprachen liegt (aber nicht in beiden). **(3 Punkte)**

Das Halteproblem H.

(g) Wie heißen die folgenden drei Superstars der BuK? **(2 Punkte)**



Alan Turing; David Hilbert; Richard Karp

(Name/Matrikelnummer)

Aufgabe 2:

(a) Formulieren Sie den Satz von Rice. **(4 Punkte)**

Sei $\mathcal{R}$ die Menge der von Turingmaschinen berechenbaren partiellen Funktionen und S eine Teilmenge von $\mathcal{R}$ mit $\emptyset \neq S \neq \mathcal{R}$. Dann ist die Sprache

$$L(S) = \{\langle M \rangle \mid M \text{ berechnet eine Funktion aus } S\}$$

nicht rekursiv.

(b) Über einige der folgenden sieben Sprachen $L_A, \ldots, L_G$ sagt der Satz von Rice nichts aus, da seine Voraussetzungen nicht erfüllt sind. Tragen Sie eine der aufgelisteten Sprachen in das Kästchen ein, für die alle Voraussetzungen des Satzes von Rice erfüllt sind: **(2 Punkte)**

$$\begin{aligned}
L_A &= \{\ \langle M \rangle \ \mid\ L(M) \text{ ist unentscheidbar } \} \\
L_B &= \{\ \langle M \rangle \ \mid\ L(M) \text{ ist semi-entscheidbar } \} \\
L_C &= \{\ \langle M \rangle \ \mid\ L(M) = H_{tot}\ \} \\
L_D &= \{\ \langle M \rangle \ \mid\ L(M) \neq H_{tot}\ \} \\
L_E &= \{\ \langle M \rangle \ \mid\ L(M) = \overline{H}_{tot}\ \} \\
L_F &= \{\ \langle M_1 \rangle \langle M_2 \rangle \ \mid\ L(M_1) = L(M_2)\ \} \\
L_G &= \{\ \langle M_1 \rangle \langle M_2 \rangle \ \mid\ L(M_1) \cap L(M_2) \neq \emptyset\ \}
\end{aligned}$$

Anmerkung: $H_{tot} = \{\ \langle M \rangle \mid M \text{ hält auf jeder Eingabe } \}$ bezeichnet das totale Halteproblem.

L_A

(Name/Matrikelnummer)

(c) Wenden Sie nun den Satz von Rice auf die von Ihnen im Aufgabenteil (b) **(7 Punkte)** gewählte Sprache an: Definieren Sie die entsprechende Menge $\mathcal{S}$.

Zeigen Sie insbesondere, dass $\mathcal{S}$ alle vom Satz von Rice geforderten Eigenschaften besitzt.

Sei $\mathcal{S}$ die Menge aller berechenbaren partiellen Funktionen f_M, so dass $L(f_M) := \{x \in \{0,1\}^* \mid f_M(x) = 1\}$ unentscheidbar ist.

Das spezielle Halteproblem H_ϵ ist semi-entscheidbar durch eine TM M. Daher is f_M eine berechenbare *partielle* Funktion. Da H_ϵ nicht entscheidbar ist, gibt es keine TM M^* mit $L(M) = H_\epsilon$. Also ist $L(f_M)$ unentscheidbar und $f_M \in \mathcal{S}$. Also $\mathcal{S} \neq \emptyset$.

Die Funktion $f \equiv 0$ entspricht der entscheidbaren leeren Sprache. Da $L(f) \equiv 0$ berechenbar ist, gilt $f \notin \mathcal{S}$. Also $\mathcal{S} \neq \mathcal{R}$.

$$\begin{aligned} L(\mathcal{S}) &= \{\langle M \rangle \mid M \text{ berechnet eine Fkt. aus } \mathcal{S}\} \\ &= \{\langle M \rangle \mid M \text{ berechnet eine Fkt. } f_M \text{ mit } L(f_M) \text{ ist unentscheidbar.}\} \\ &= \{\langle M \rangle \mid L(M) \text{ ist unentscheidbar.}\} = L_A \end{aligned}$$

ist nach Satz von Rice unentscheidbar.

(Name/Matrikelnummer)

(d) Beweisen oder widerlegen Sie: **(7 Punkte)**

Die Sprache $L_G = \{\langle M_1\rangle\langle M_2\rangle \mid L(M_1) \cap L(M_2) \neq \emptyset\}$ aus Aufgabenteil (b) ist rekursiv aufzählbar.

Wir zeigen, dass die Sprache L_G von einer NTM M erkannt wird. Damit ist L_G semi-entscheidbar und rekursive aufzählbar.

Die NTM M schreibt zuerst nicht-deterministisch ein Wort $w \in \Sigma^*$ und eine Zahl $n \in \mathbb{N}$ aufs Band. Dann simuliert M die TM M_1 mit Eingabe w und die TM M_2 mit Eingabe w für jeweils n Schritte. M akzeptiert, falls sowohl M_1 als auch M_2 akzeptieren.

Korrektheit:

Falls $\langle M_1\rangle\langle M_2\rangle \in M$, dann gibt es ein Wort $w \in L(M_1) \cap L(M_2)$. Insbesondere gibt es $i_1 \in \mathbb{N}$ so dass M_1 Eingabe w in i_1 Schritten akzeptiert und es gibt ein $i_2 \in \mathbb{N}$ so dass M_2 Eingabe w in i_2 Schritten akzeptiert. Also akzeptieren je M_1 und M_2 Eingabe w nach höchstens $n = \max\{i_1, i_2\}$ Schritten. Daher akzeptiert M die Eingabe $\langle M_1\rangle\langle M_2\rangle$.

Falls $\langle M_1\rangle\langle M_2\rangle \notin M$, dann gilt für alle Eingaben w und $n \in \mathbb{N}$, dass entweder M_1 oder M_2 Eingabe w nicht in n Schritten akzeptiert. Demnach verwirft M die Eingabe $\langle M_1\rangle\langle M_2\rangle$.

Alternativer Beweis:

Wir zeigen, dass L_G rekursiv aufzählbar ist durch einen Aufzähler M, der wie folgt arbeitet. Für $i = 0, 1, \ldots$ betrachte alle Turingmaschinen M_j, M_k mit $j, k \leq i$ und alle Wörter w mit $|w| \leq i$. Simuliere M_j auf Eingabe w und simuliere M_k auf Eingabe w für i Schritte. Drucke $\langle M_j\rangle\langle M_k\rangle$ falls beide Simulationen in i Schritten akzeptieren.

[Korrektheit ähnlich wie oben]

(Name/Matrikelnummer)

Zusätzliches Papier

(Name/Matrikelnummer)

Aufgabe 3:

(a) Wir betrachten ein rechteckiges Schachbrett der Höhe n und Breite m. Jedes der $n \cdot m$ Felder ist entweder mit einem schwarzem Spielstein belegt, oder mit einem weißem Spielstein belegt, oder leer. Ein gegebenes Schachbrett mit Spielsteinen ist in **Mondrian-Konstellation**, **(2 Punkte)**

- falls jede (vertikale) Spalte mindestens einen Spielstein enthält, und
- falls keine (horizontale) Zeile zwei verschieden-farbige Spielsteine enthält.

Wir betrachten das folgende Entscheidungsproblem MONDRIAN:

> **Eingabe:** Ein $\boldsymbol{n \times m}$ Schachbrett und eine Belegung mit schwarzen und weißen Spielsteinen.
>
> **Frage:** Kann man durch das Entfernen von Spielsteinen vom Schachbrett eine Mondrian-Konstellation erreichen?

Betrachten Sie die folgende Instanz des MONDRIAN Problems, die aus **12** weißen und **12** schwarzen Spielsteinen auf einem $\mathbf{3 \times 8}$ Schachbrett besteht. (Diese spezielle Instanz enthält keine leeren Felder.)

○	○	○	○	●	●	●	●
○	○	●	●	○	○	●	●
○	●	○	●	○	●	○	●

Ist diese Instanz eine JA-Instanz? Tragen Sie Ihre Antwort in das Kästchen ein:

NEIN. (Aus jeder der drei Zeilen müssen entweder alle weißen oder alle schwarzen Spielsteine entfernt werden. Jede dieser acht Möglichkeiten kollidiert mit einer der acht Spalten.)

(b) Formulieren Sie die Zertifikat-Charakterisierung von NP. **(4 Punkte)**

Eine Sprache $\boldsymbol{L}$ liegt in NP genau dann, wenn es einen polynomiellen Algorithmus $\boldsymbol{V}$ und ein Polynom $\boldsymbol{p}$ gibt, sodass $\boldsymbol{x} \in \boldsymbol{L}$ genau dann wenn $\exists \boldsymbol{y} \in \{\mathbf{0}, \mathbf{1}\}^*$ mit $|\boldsymbol{y}| \leq \boldsymbol{p}(|\boldsymbol{x}|)$ und $\boldsymbol{V}$ akzeptiert $\boldsymbol{y}\#\boldsymbol{x}$.

(Name/Matrikelnummer)

(c) Zeigen Sie, dass MONDRIAN die Zertifikat-Charakterisierung von NP erfüllt. Beschreiben Sie Ihr Zertifikat und analysieren Sie seine Länge. Beschreiben Sie das Verhalten Ihres Verifizierers und analysieren Sie seine Laufzeit. **(6 Punkte)**

Das Zertifikat $\boldsymbol{y}$ kodiert die Felder der Spielsteine, durch deren Wegnehmen eine Mondrian-Konstellation entsteht.

Es werden höchstens $\boldsymbol{mn}$ Steine weggenommen, und jedes einzelne Feld kann mit $\boldsymbol{\log(mn)}$ Bits beschrieben werden. Daher ist die Länge des Zertifikats $\boldsymbol{y}$ polynomiell beschränkt durch $\boldsymbol{O(mn\log(mn))}$.

Der Verifizierer $\boldsymbol{V}$ entfernt zuerst die Spielsteine, deren Felder im Zertifikat $\boldsymbol{y}$ beschrieben sind, und überprüft, ob die verbleibenden Steine eine Mondrian-Konstellation bilden. Dazu muss $\boldsymbol{V}$ nur jede Spalte und jede Zeile einmal durchlaufen. Die Laufzeit des Verifizierers ist daher $\boldsymbol{O(mn)}$ und polynomiell beschränkt in der Grösse des Schachbretts.

(d) Beweisen Sie durch eine polynomielle Reduktion: MONDRIAN ist NP-schwer. **(8 Punkte)**

Wir zeigen 3-SAT $\leq_p$ MONDRIAN. Sei φ eine SAT-Formel über der Variablenmenge $X = \{x_1, \ldots, x_n\}$ und mit Klauseln $c_1, \ldots, c_m$. O.B.d.A. enthält keine Klausel gleichzeitig x_i und $\neg x_i$. Auf unserem $n \times m$ Schachbrett entspricht die i-te Zeile der Variablen x_i und entspricht die j-te Spalte der Klausel c_j.

- Wenn das positive Literal x_i in der j-ten Klausel vorkommt, so legen wir einen weißen Stein auf das Feld $F[i, j]$.
- Wenn das negative Literal $\neg x_i$ in der j-ten Klausel vorkommt, so legen wir einen schwarzen Stein auf das Feld $F[i, j]$.
- (Auf alle anderen Felder legen wir keinen Stein.)

Die Reduktion kann in polynomieller Zeit $O(mn)$ durchgeführt werden, da jedes der $O(mn)$ Felder in konstanter Zeit belegt werden kann.

Korrektheit: Wir zeigen, dass die gegebene SAT Formel φ erfüllbar ist genau dann, wenn die konstruierte Instanz von MONDRIAN die Antwort Ja hat.

($\Rightarrow$) Angenommen, es gibt eine erfüllende Wahrheitsbelegung für X. Wir entfernen alle Steine, die zu falschen Literalen gehören. Danach sind in der i-ten Zeile entweder nur weisse Steine übrig (für x_i) oder nur schwarze Steine übrig (für $\neg x_i$). Und da jede Klausel mindestens ein wahres Literal enthält, enthält jede Spalte mindestens einen Stein.

($\Leftarrow$) Angenommen, durch das Entfernen einiger Spielsteine kann eine Mondrian-Konstellation erreicht werden. Wenn dann in der i-ten Zeile nur weiße Steine sind, so setzen wir $x_i := 1$ und andernfalls setzen wir $x_i := 0$. Da jede Spalte mindestens einen Stein enthält, enthält jede Klausel mindestens ein wahres Literal.

(Name/Matrikelnummer)

Zusätzliches Papier

(Name/Matrikelnummer)

Aufgabe 4:

(a) **(8 Punkte)**

Beantworten Sie für jede ganze Zahl $m \geq 0$:

Wenn das folgende LOOP-Programm mit der Eingabe $x_1 = m$ gestartet wird, welchen Wert hat dann die Variable x_2 bei Terminierung? Geben Sie eine geschlossene Form an.

Beweisen Sie Ihre Antwort. (Sie können das Verhalten von LOOP-Programmen, die in der Vorlesung besprochen wurden, als bekannt annehmen.)

$x_2 := 2;$
LOOP x_1 DO
  $x_3 := 0;$
  LOOP x_2 DO
    LOOP x_2 DO $x_3 := x_3 + 1$ ENDLOOP
  ENDLOOP;
  $x_2 := x_3$
ENDLOOP

Die Antwort ist $2^{(2^m)}$ (bzw. $3^{(2^m)}$). Für $m = 0$ wird nur der erste Befehl $x_2 := 2$ (bzw. $x_2 := 3$) ausgeführt, und das Programm terminiert mit $x_2 = 2 = 2^{2^0}$ (bzw. $x_2 = 3 = 3^{2^0}$).

Für $m \geq 1$ bestimmen die beiden inneren Loops das Quadrat von x_2 und weisen es an x_3 zu: x_3 wird mit 0 initialisiert, und danach $x_2 \cdot x_2$-mal um 1 inkrementiert. Nach m Exekutionen des äusseren Loops wurde x_2 genau m-mal quadriert, und nimmt deshalb den Wert $2^{(2^m)}$ (bzw. 3^{2^0}) an.

(Name/Matrikelnummer)

(b) **(4 Punkte)**

Bestimmen Sie (mit Beweis) alle Zahlen $b \in \mathbb{N}$, für die die Funktion $g : \mathbb{N} \to \mathbb{N}$ mit $g(n) = b^{(2^n)}$ primitiv rekursiv ist.

Alle ganzen Zahl $b \in \mathbb{N}$ haben die gewünschte Eigenschaft.

Wenn man im LOOP Programm im Aufgabenteil (a) die erste Zeile "$x_2 := 2$" durch "$x_2 := b$" ersetzt, so hat die Variable x_2 bei Termination den Wert $b^{(2^m)}$. Daher ist g LOOP-berechenbar und somit primitiv rekursiv.

(c) **(8 Punkte)**

Beweisen oder widerlegen Sie:

Wenn das LOOP-Programm im Aufgabenteil (a) auf einer Eingabe $(x_1, x_2, x_3) = (a_1, a_2, a_3)$ mit $a_1 + a_2 + a_3 = s$ gestartet wird und mit einer Ausgabe $(x_1, x_2, x_3) = (b_1, b_2, b_3)$ terminiert, dann gilt $b_1 + b_2 + b_3 \leq A(3, s + 20)$.

Anmerkung: $A(\cdot, \cdot)$ bezeichnet hier die Ackermann Funktion.

Wir widerlegen die Aussage.

Wenn das LOOP-Programm mit $(x_1, x_2, x_3) = (s, 0, 0)$ gestartet wird, so terminiert es mit $(x_1, x_2, x_3) = (s, 2^{2^s}, 2^{2^s})$ (bzw. $= (s, 3^{2^s}, 3^{2^s})$). Unter $a_1 + a_2 + a_3 \leq s$ kann daher $b_1 + b_2 + b_3 = s + 2^{2^s} + 2^{2^s} > 2^{2^s}$ (bzw. $b_1 + b_2 + b_3 = s + 3^{2^s} + 3^{2^s} > 2^{2^s}$) erreicht werden. Da 2^{2^s} doppelt exponentiell wächst und da $A(3, s+20) = 2^{s+23} - 3$ nur einfach exponentiell wächst, ist die zu untersuchende Aussage falsch.

(Name/Matrikelnummer)

Zusätzliches Papier