Institut für Informatik der Ludwig-Maximilians-Universität München
Prof. Dr. Jasmin Blanchette
Jannis Limperg
Luca Maio

SoSe 2024
30.09.2024

Lösung zur Zweitklausur zur Vorlesung

Theoretische Informatik für Studierende der Medieninformatik

Die Bearbeitungszeit beträgt **120 Minuten**. Hilfsmittel sind nicht erlaubt, auch das Mitführen ausgeschalteter elektronischer Geräte wird als Betrug gewertet. Schreiben Sie Ihren vollständigen Namen und Ihre Matrikelnummer deutlich lesbar auf dieses Deckblatt, sowie Ihren Namen in die Kopfzeile auf jedem Blatt der Klausurangabe. Geben Sie alle Blätter ab. Lassen Sie diese zusammengeheftet. Verwenden Sie nur **dokumentenechte Stifte** und **weder** die Farbe **rot noch grün**.

Kontrollieren Sie, ob Sie alle Aufgabenblätter erhalten haben. Aufgabenstellungen befinden sich auf den **Seiten 1–13**. Sie dürfen die Rückseiten für Nebenrechnungen nutzen. Falls Sie die Rückseiten für Antworten nutzen, so markieren Sie klar, was zu welcher Aufgabe gehört und geben Sie in der entsprechenden Aufgabe an, wo alle Teile Ihrer Antwort zu finden sind. Streichen Sie alles durch, was nicht korrigiert werden soll.

Es gibt 5 unterschiedlich gewichtete Aufgaben zu insgesamt 100 Punkten. Die Teilaufgaben können unabhängig voneinander bearbeitet werden.

Mit Ihrer Unterschrift bestätigen Sie, dass Sie zu Beginn der Klausur in ausreichend guter gesundheitlicher Verfassung sind und diese Klausurprüfung verbindlich annehmen.

Nachname:

Vorname:

Matrikelnummer:

Studiengang:

☐ Bitte *nur* ankreuzen, wenn die Klausur entwertet und nicht korrigiert werden soll.

Please check with an X *only* if the exam should be voided and not graded.

Hiermit erkläre ich die Richtigkeit der obigen Angaben: _____

Unterschrift

Die folgende Tabelle nicht ausfüllen:

Aufgabe	1	2	3	4	5	$\sum$
Punkte	31	18	24	15	12	100
Erreicht						

Lösung Aufgabe 1 (Reguläre Sprachen): (31 Punkte)

a) Die Sprache L_1 sei definiert als die Menge aller Wörter über dem Alphabet $\{a, b, c\}$, die mindestens ein a und mindestens ein b enthalten.

Geben Sie einen regulären Ausdruck an, der L_1 erzeugt. (10 Punkte)

LÖSUNGSVORSCHLAG:

$(a|b|c)^*a(a|b|c)^*b(a|b|c)^* \mid (a|b|c)^*b(a|b|c)^*a(a|b|c)^*$

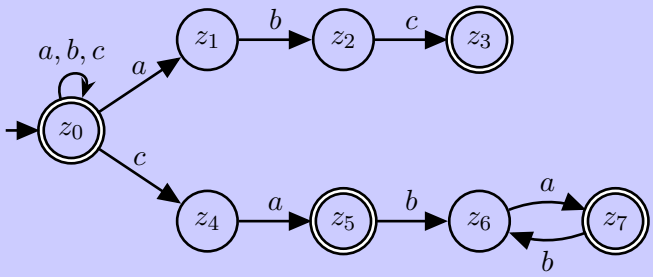
- 0 Punkte, wenn die Antwort sich nicht als regulärer Ausdruck verstehen lässt
- 1 Punkt Abzug pro Wort, das falsch vom regulären Ausdruck erkannt bzw. verworfen wird:
  - $\varepsilon \in L_1$
  - $a \in L_1$
  - $b \in L_1$
  - $c \in L_1$
  - $cc \in L_1$
  - $ab \in L_1$
  - $ba \in L_1$
- 2 Punkte Abzug, wenn „$(a|b|c)^*a(a|b|c)^*b(a|b|c)^*$“ oder „$(a|b|c)^*b(a|b|c)^*a(a|b|c)^*$“ fehlt

b) Sei L_2 die Sprache über dem Alphabet $\{a, b, c\}$, die vom regulären Ausdruck

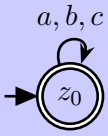
$$(a|b|c)^*(abc\,|\,ca(ba)^*\,|\,\varepsilon)$$

erzeugt wird. Geben Sie den Zustandsgraphen eines nichtdeterministischen endlichen Automaten (ohne ε-Übergänge) an, der L_2 akzeptiert. (10 Punkte)

LÖSUNGSVORSCHLAG:

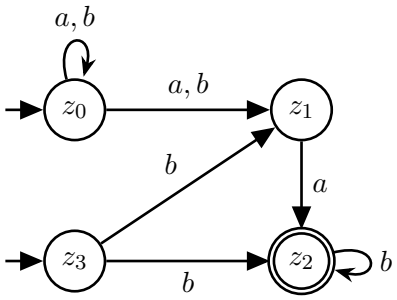


Alternative Lösung (da der reguläre Ausdruck äquivalent zu $(a|b|c)^*$ ist):



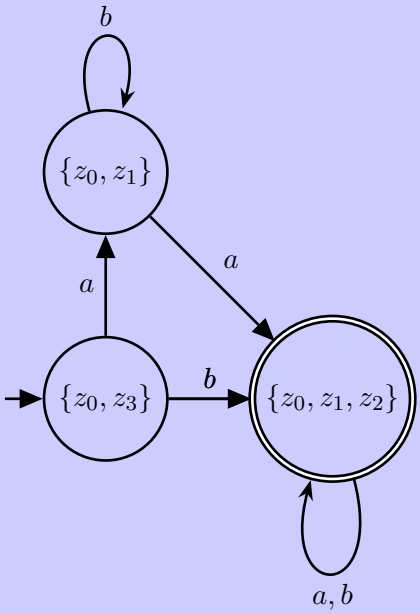
- 10 Punkte, wenn alles stimmt:
  - 1 Punkt Abzug bei fehlendem Startzustand
  - 1 Punkt Abzug bei jedem fehlendem Endzustand
  - 1 Punkt Abzug bei jedem falschen Übergang oder Zustand

c) Der nichtdeterministische endliche Automat M_1 über $\{a, b\}$ ist durch den folgenden Zustandsgraphen gegeben:



Berechnen Sie einen deterministischen endlichen Automaten M_1' aus M_1 mit der Potenzmengenkonstruktion. Es ist ausreichend, die erreichbaren Zustände von M_1' anzugeben. Sie müssen keine Begründung angeben, nur das Ergebnis. (11 Punkte)

LÖSUNGSVORSCHLAG:



d) 11 Punkte, wenn alles stimmt:

- 1 Punkt Abzug bei fehlendem Startzustand
- 1 Punkt Abzug bei jedem fehlenden Endzustand
- 1 Punkt Abzug bei jedem falschen Übergang oder Zustand
- 3 Punkte Abzug, falls der DFA mit $\{z_0\}$ startet und sonst korrekt ist unter dieser Annahme

Lösung Aufgabe 2 (Nicht reguläre Sprachen): (18 Punkte)

a) Zeigen Sie mit dem Pumping-Lemma für reguläre Sprachen, dass die Sprache

$$L_3 = \{a^i b^j a^j b^i \mid i, j \in \mathbb{N}\}$$

über dem Alphabet $\{a, b\}$ nicht regulär ist. (10 Punkte)

Zur Erinnerung:

Eine Sprache L hat die Pumping-Eigenschaft, wenn es eine Zahl $n \in \mathbb{N}_{>0}$ gibt, sodass jedes Wort $z \in L$, welches Mindestlänge n hat, als $z = uvw$ geschrieben werden kann, sodass $|uv| \leq n$, $|v| \geq 1$ und für alle $i \in \mathbb{N}$: $uv^i w \in L$.

Das Pumping-Lemma für reguläre Sprachen besagt, dass jede reguläre Sprache die Pumping-Eigenschaft hat.

LÖSUNGSVORSCHLAG: Um Nichtregularität von L_3 zu zeigen, reicht es durch das Pumping-Lemma zu zeigen, dass L_3 die Pumping-Eigenschaft nicht hat.

Der Beweis ist durch Widerspruch: Wir nehmen an, dass L_3 die Pumping-Eigenschaft hat. Wir wählen $w = a^n b^n$. Damit sind auch $w \in L_3$ und $|z| \geq n$ erfüllt.

Sei $z = uvw$ eine beliebige Zerlegung von z, sodass $|uv| \leq n$, $|v| \geq 1$ und $uv^i w \in L_3$ für jedes $i \in \mathbb{N}$.

Es gilt $u = a^{n_1}$, $v = a^{n_2}$ und $w = a^{n_3} b^n$ mit $n_1 + n_2 + n_3 = n$. Dann ist $uv^0 w \notin L_3$, denn $uv^0 w = a^{n_1+n_3} b^n$ und es gibt weniger a's als b's. Widerspruch mit der Annahme, dass $uv^i w \in L_3$ für jedes $i \in \mathbb{N}$.

- 1 Punkt: $z \in L_3$ gewählt
- 1 Punkt: $|z| \geq n$
- 2 Punkte: z ist geeignet
- 3 Punkte: Über alle Zerlegungen u, v, w argumentiert:
  - 0 Punkte Abzug, wenn unnötige Fälle betrachtet wurden
- 3 Punkte: Fall gefunden, sodass $uv^i w \notin L_3$

b) Beweisen Sie mithilfe der Abschlusseigenschaften der regulären Sprachen, dass die Sprache

$$L_4 = \overline{\{a^i b^j a^j b^i \mid i, j \in \mathbb{N}\}}$$

über dem Alphabet $\{a, b\}$ nicht regulär ist, wobei $\overline{L}$ das Komplement einer Sprache L bezeichnet. Sie dürfen annehmen, dass die Sprache $\{a^i b^i \mid i \in \mathbb{N}\}$ nicht regulär ist.

Zur Erinnerung: Die regulären Sprachen sind unter Vereinigung, Schnitt, Komplement, Produkt und Kleeneschem Abschluss abgeschlossen. (8 Punkte)

LÖSUNGSVORSCHLAG: Wir nehmen an, dass L_4 regulär ist. Das heißt, dass ihr Komplement $\overline{\overline{\{a^i b^j a^j b^i \mid i, j \in \mathbb{N}\}}} = \{a^i b^j a^j b^i \mid i, j \in \mathbb{N}\}$ auch regulär ist. Somit ist die Schnittbildung der offensichtlichen regulären Sprache $\{a^i b^j \mid i, j \in \mathbb{N}\} = L(a^* b^*)$ und von $\{a^i b^j a^j b^i \mid i, j \in \mathbb{N}\}$ auch regulär. Aber dies ist genau die Sprache $\{a^i b^i \mid i \in \mathbb{N}\}$, die bekanntermaßen nicht regulär ist. Widerspruch.

Alternativer Beweis: Wir nehmen an, dass L_4 regulär ist. Das heißt, dass ihr Komplement $\overline{\overline{\{a^i b^j a^j b^i \mid i, j \in \mathbb{N}\}}} = \{a^i b^j a^j b^i \mid i, j \in \mathbb{N}\}$. auch regulär ist. Aber wir haben in Teilfrage a) bewiesen, dass $\{a^i b^j a^j b^i \mid i, j \in \mathbb{N}\}$ nicht regulär ist. Widerspruch.

- 1 Punkt: Annahme, dass L_4 regulär ist.
- 2 Punkte: Komplement ist regulär.
- 2 Punkte: $\{a^i b^j \mid i, j \in \mathbb{N}\}$ ist regulär.
- 2 Punkte: Schnitt ist regulär.
- 1 Punkt: Verweis auf Nichtregularität von $\{a^i b^i \mid i \in \mathbb{N}\}$.

Lösung Aufgabe 3 (Kontextfreie Sprachen): **(24 Punkte)**

a) Die Sprache L_5 über dem Alphabet $\{a, b\}$ sei definiert als

$$L_5 = \{a^i b a^{2i} b^j a b^j \mid i, j \in \mathbb{N}\}$$

Geben Sie eine kontextfreie Grammatik G_1 als 4-Tupel an, die L_5 erzeugt. Die Grammatik darf keine ε-Produktionen enthalten. Erläutern Sie, warum G_1 die Sprache L_5 erzeugt. Beschreiben Sie beispielsweise, welche „Aufgabe“ die einzelnen Nichtterminale bei der Erzeugung übernehmen. (8 Punkte)

Geben Sie zusätzlich eine Linksableitung für das Wort *aabaaaabab* für Ihre Grammatik an. (4 Punkte)

LÖSUNGSVORSCHLAG: $G_1 = (V_1, \Sigma_1, P_1, S_1)$ mit $V_1 = \{S, T, U\}$, $\Sigma_1 = \{a, b\}$ und

$$P_1 = \{S \to TU,\ T \to aTaa \mid b,\ U \to bUb \mid a\}$$

S erzeugt zunächst T und U. Aus T lassen sich null oder mehr a/aa-Paare erzeugen, mit b in der Mitte. Aus U lassen sich null oder mehr b/b-Paare erzeugen, mit a in der Mitte.

Linksableitung:
$S \Rightarrow TU \Rightarrow aTaaU \Rightarrow aaTaaaaU \Rightarrow aabaaaaU \Rightarrow aabaaaabUb \Rightarrow aabaaaabab$.

- 8 Punkte für die Grammatik:
  - 6 Punkte für die Grammatik
  - 2 Punkte für die Erläuterung
  - Maximal 1 Punkt, falls Grammatik ganz falsch
  - Maximal 1 Punkt, wenn die Grammatik nicht kontextfrei ist
  - Maximal 1 Punkt für Erläuterung bei falscher Grammatik
  - 1 Punkt Abzug für „off by one“ in T bzw. U
- 4 Punkte für die Linksableitung:
  - 1 Punkt, falls keine Linksableitung oder falls Syntaxbaum

b) Gegeben sei die kontextfreie Grammatik $G_2 = (V_2, \Sigma_2, P_2, S)$ mit $V_2 = \{S, T, U\}$, $\Sigma_2 = \{a, b, c\}$ und

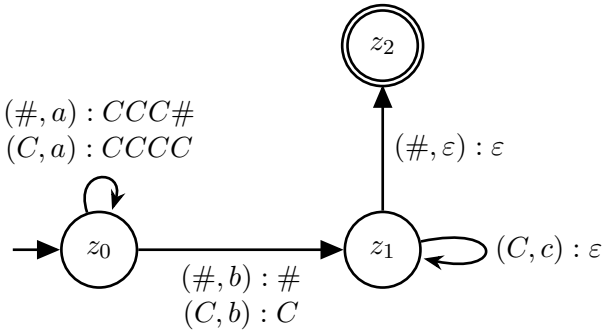
$$P_2 = \{S \to TUT, T \to aTa, T \to c, U \to bUb, U \to c\}$$

Geben Sie eine mathematische Beschreibung der Sprache an, die G_2 erzeugt. (6 Punkte)

LÖSUNGSVORSCHLAG: $L(G_2) = \{a^i c a^i b^j c b^j a^k c a^k \mid i, j, k \in \mathbb{N}\}$.

- 2 Punkte für Lösung von der allgemeinen Form $a^*ca^*b^*cb^*a^*ca^*$
- 1 Punkt: gleiche Anzahl an a's vorne
- 1 Punkt: gleiche Anzahl an b's und eigene Variable j
- 1 Punkt: gleiche Anzahl an a's hinten und eigene Variable k
- 1 Punkt: $i, j, k \in \mathbb{N}$

c) Gegeben sei der folgende Zustandsgraph eines Kellerautomaten M_2 mit Endzuständen. Geben Sie eine mathematische Beschreibung der Sprache an, die M_2 akzeptiert. (6 Punkte)



LÖSUNGSVORSCHLAG: $L(M_2) = \{a^i b c^{3i} \in \{a, b, c\}^* \mid i \in \mathbb{N}\}$.

- 2 Punkte für Lösung von der allgemeinen Form $a^*b^*c^*$
- 1 Punkt: a^i
- 1 Punkt: b
- 1 Punkt: c^{3i}
- 1 Punkt: $i \in \mathbb{N}$

Lösung Aufgabe 4 (Berechenbarkeit und Entscheidbarkeit): (15 Punkte)

a) Der Satz von Rice besagt:

> Sei $\mathcal{R}$ die Klasse aller turingberechenbaren Funktionen. Sei $\mathcal{S}$ eine nichtleere echte Teilmenge von $\mathcal{R}$. Dann ist folgende Sprache unentscheidbar:
>
> $$C(\mathcal{S}) = \{w \mid \text{die von der deterministischen Turingmaschine } M_w \text{ berechnete Funktion liegt in } \mathcal{S}\}$$
>
> wobei M_w die Turingmaschine mit der Gödelnummer w bezeichnet.

Wenden Sie für jede der beiden Aussagen unten den Satz von Rice an um sie zu beweisen.

(i) Es ist unentscheidbar, ob eine gegebene deterministische Turingmaschine für die Eingabe 42 die Zahl 43 berechnet. (4 Punkte)

LÖSUNGSVORSCHLAG: Sei $\mathcal{S}$ die Menge aller (partiellen oder totalen) Funktionen f, sodass $f(42) = 43$ gibt.
$\mathcal{S}$ ist nicht leer. Z.B. ist $(i \mapsto i+1) \in \mathcal{S}$.
$\mathcal{S}$ ist auch nicht gleich $\mathcal{R}$, weil es z.B. die Funktion $(i \mapsto i) \in \mathcal{R} \setminus \mathcal{S}$ gibt.
Dann ist folgende Sprache unentscheidbar:

$$\begin{aligned} C(\mathcal{S}) &= \{w \mid \text{die von der deterministischen Turingmaschine } M_w \text{ berechnete Funktion liegt in } \mathcal{S}\} \\ &= \{w \mid \text{die Turingmaschine } M_w \text{ berechnet eine Funktion } f, \text{ welche für die Eingabe 42 die Zahl 43 berechnet}\} \end{aligned}$$

- 1 Punkt für die Definition von $\mathcal{S}$
- 1 Punkt für „$\mathcal{S}$ nicht leer“-Beweis
- 1 Punkt für „$\mathcal{S}$ nicht gleich $\mathcal{R}$“-Beweis
- 1 Punkt für „$C(\mathcal{S}) = \cdots$“-Beweis

(ii) Es ist unentscheidbar, ob eine gegebene deterministische Turingmaschine für mindestens eine Eingabe $i \in \mathbb{N}$ die Zahl i berechnet. (4 Punkte)

LÖSUNGSVORSCHLAG: Sei $\mathcal{S}$ die Menge aller (partiellen oder totalen) Funktionen f, sodass es i mit $f(i) = i$ gibt.
$\mathcal{S}$ ist nicht leer. Z.B. ist $(i \mapsto i) \in \mathcal{S}$.
$\mathcal{S}$ ist auch nicht gleich $\mathcal{R}$, weil es z.B. die Funktion $(i \mapsto i+1) \in \mathcal{R} \setminus \mathcal{S}$ gibt.
Dann ist folgende Sprache unentscheidbar:

$$\begin{aligned} C(\mathcal{S}) &= \{w \mid \text{die von der deterministischen Turingmaschine } M_w \text{ berechnete Funktion liegt in } \mathcal{S}\} \\ &= \{w \mid \text{die Turingmaschine } M_w \text{ berechnet eine Funktion } f, \text{ welche für mindestens eine Eingabe } i \in \mathbb{N} \text{ die Zahl } i \text{ berechnet}\} \end{aligned}$$

- 1 Punkt für die Definition von $\mathcal{S}$
- 1 Punkt für „$\mathcal{S}$ nicht leer“-Beweis
- 1 Punkt für „$\mathcal{S}$ nicht gleich $\mathcal{R}$“-Beweis
- 1 Punkt für „$C(\mathcal{S}) = \cdots$“-Beweis

b) Das Postsche Korrespondenzproblem lässt sich so definieren:

> Sei Σ ein Alphabet mit $|\Sigma| > 1$. Eine Instanz des *Postschen Korrespondenzproblems* (*PCP*) besteht aus einer endlichen Folge $(x_1, y_1), \ldots, (x_k, y_k)$ von Wortpaaren mit $x_i, y_i \in \Sigma^+$. Das Entscheidungsproblem ist die Frage, ob es eine Folge von Indizes $i_1, \ldots, i_m$ mit $i_j \in \{1, \ldots, k\}$ und $m > 0$ gibt, sodass $x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m}$ gilt.

Eine weniger bekannte Variante des Problems lässt sich so definieren:

> Sei Σ ein Alphabet mit $|\Sigma| > 1$. Eine Instanz des *umgedrehten Postschen Korrespondenzproblems* (*UPCP*) besteht aus einer endlichen Folge $(x_1, y_1), \ldots, (x_k, y_k)$ von Wortpaaren mit $x_i, y_i \in \Sigma^+$. Das Entscheidungsproblem ist die Frage, ob es eine Folge von Indizes $i_1, \ldots, i_m$ mit $i_j \in \{1, \ldots, k\}$ und $m > 0$ gibt, sodass $x_{i_1} \cdots x_{i_m} = \overline{y_{i_1}} \cdot \ldots \cdot \overline{y_{i_m}}$ gilt, wobei $\overline{a_1 \cdots a_n}$ als $a_n \cdots a_1$ definiert ist.

Beweisen Sie mithilfe einer Reduktion, dass UPCP unentscheidbar ist. (7 Punkte)

LÖSUNGSVORSCHLAG: Wir zeigen PCP $\leq$ UPCP.

Sei $f((x_1, y_1), \ldots, (x_k, y_k)) = (x_1, \overline{y_1}), \ldots, (x_k, \overline{y_k})$.

Die Funktion f ist offensichtlich total und berechenbar.

Korrektheit:

$(x_1, y_1), \ldots, (x_k, y_k) \in$ PCP
g.d.w. es gibt $i_1, \ldots, i_m$, sodass $x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m}$
g.d.w. es gibt $i_1, \ldots, i_m$, sodass $x_{i_1} \cdots x_{i_m} = \overline{\overline{y_{i_1}}} \cdots \overline{\overline{y_{i_m}}}$
g.d.w. $(x_1, \overline{y_1}), \ldots, (x_k, \overline{y_k}) \in$ UPCP

- 2 Punkte: „PCP $\leq$ UPCP“
- 2 Punkte: Definition von f
- 1 Punkt: f ist (total und) berechenbar
- 2 Punkte: „g.d.w.“-Beweis
  - 1 Punkt für erste und letzte Zeile
  - 1 Punkte für die zwei mittleren Zeilen
- Insgesamt höchstens 2 Punkte, falls die Reduktion falsch herum ist

Lösung Aufgabe 5 (Komplexität): (12 Punkte)

a) Wir erinnern zunächst an die Definition des DIRECTED-HAMILTON-Problems:

gegeben:	ein gerichteter Graph $G = (V, E)$ mit $V = \{v_1, \ldots, v_n\}$
gefragt:	Gibt es einen Hamilton-Kreis in G? (Ein Hamilton-Kreis ist ein Kreis, der genau alle Knoten einmal besucht.)

In der Vorlesung wurde bewiesen, dass DIRECTED-HAMILTON $\mathcal{NP}$-vollständig ist.

Eine Variante namens FLAT-DIRECTED-HAMILTON folgt:

gegeben:	ein gerichteter Graph $G = (V, E)$ mit $V = \{v_1, \ldots, v_n\}$, sodass $w = w'$ wenn $(v, w) \in E$ und $(v, w') \in E$ für alle $v, w, w' \in V$ (d.h. jeder Knoten in G hat höchstens einen Nachfolger)
gefragt:	Gibt es einen Hamilton-Kreis in G, d.h. einen Kreis, der genau alle Knoten einmal besucht?

Nehmen Sie an, dass $\mathcal{P} \neq \mathcal{NP}$. Ist FLAT-DIRECTED-HAMILTON $\mathcal{NP}$-vollständig? Begründen Sie Ihre Antwort (z.B. mit einer Reduktion oder einer Skizze eines Algorithmus). (5 Punkte)

LÖSUNGSVORSCHLAG: Nein, FLAT-DIRECTED-HAMILTON ist in $\mathcal{P}$. Ein Polynomialzeit-Algorithmus ist wie folgt:

1. Wenn der Graph zusammenhängend ist und die Form eines Kreises hat, dann „Ja“.
2. Sonst „Nein“.

- 1 Punkt für richtige Antwort (nein)
- 1 Punkt für „es gibt einen *Polynomialzeit*-Algorithmus“
- 3 Punkte für den Algorithmus

b) Wir erinnern an die Definition des GRAPH-COLORING-Problems:

gegeben:	ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$
gefragt:	Gibt es eine Färbung der Knoten in V mit höchstens k Farben, sodass keine zwei benachbarten Knoten in G die gleiche Farbe erhalten?

Sie dürfen als bekannt annehmen, dass GRAPH-COLORING $\mathcal{NP}$-vollständig ist.

Ein weniger bekanntes Problem ist ASSIGN-GIFTS:

gegeben:	eine endliche Menge P von Personen, eine Menge $M = \{K_1, \ldots, K_n\}$ von Mengen $K_i \subseteq P$, sodass K_i einen Freundeskreis darstellt, und eine Zahl $t \in \mathbb{N}_{>0}$, die angibt, wie viele Geschenktypen es gibt
gefragt:	Gibt es eine Zuweisung $g : P \to \{0, 1, \ldots, t-1\}$ von Personen nach Geschenktypen, sodass keine zwei Personen im selben Freundeskreis denselben Geschenktypen bekommen?

Ein Beispiel für eine ASSIGN-GIFTS-Instanz ist

$$\begin{aligned} P &= \{\mathit{anita}, \mathit{bashar}, \mathit{cindy}\} \\ M &= \{\{\mathit{anita}, \mathit{bashar}\}, \{\mathit{anita}, \mathit{cindy}\}\} \\ t &= 2 \end{aligned}$$

Diese Instanz ist lösbar, wie die Zuweisung $g(\mathit{anita}) = 0$, $g(\mathit{bashar}) = 1$ und $g(\mathit{cindy}) = 1$ bezeugt.

Zeigen Sie mithilfe einer Polynomialzeit-Reduktion, dass ASSIGN-GIFTS $\mathcal{NP}$-schwer ist.

(7 Punkte)

LÖSUNGSVORSCHLAG: Wir zeigen GRAPH-COLORING $\leq_p$ ASSIGN-GIFTS.

Sei $((V, E), k)$ eine GRAPH-COLORING-Instanz. Wir setzen $f(((V, E), k)) = (P, M, t)$ mit $P = V$, $M = E$ (wobei jeder Freundeskreis die Größe 2 hat) und $t = k$.

Die Funktion f ist offensichtlich total und lässt sich in Polynomialzeit berechnen.

Korrektheit:

$((V, E), k) \in$ GRAPH-COLORING
g.d.w. es gibt eine Färbung der Knoten in V mit höchsten k Farben,
sodass keine zwei benachbarten Knoten in G die gleiche Farbe erhalten
g.d.w. es gibt eine Funktion $c : V \to \{0, 1, \ldots, k-1\}$, sodass wenn $\{v_1, v_2\} \in E$, dann $c(v_1) \neq c(v_2)$
g.d.w. es gibt eine Funktion $g : P \to \{0, 1, \ldots, t-1\}$, sodass wenn $\{p_1, p_2\} \in M$, dann $g(p_1) \neq g(p_2)$
g.d.w. es gibt eine Zuweisung $g : P \to \{0, 1, \ldots, t-1\}$, sodass keine zwei Personen im selben Freundeskreis denselben Geschenktypen bekommen
g.d.w. $(P, M, t) \in$ ASSIGN-GIFTS

- 2 Punkte: „GRAPH-COLORING $\leq_p$ ASSIGN-GIFTS“
- 2 Punkte: Definition von f
  - 1 Punkt für P und t

  - 1 Punkt für M
- 1 Punkt: f ist (total und) polynomiell
- 2 Punkte: „g.d.w.“-Beweis
  - 2 Punkte für Ausfalten der Definitionen
- Insgesamt höchstens 2 Punkte, falls die Reduktion falsch herum ist