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

SoSe 2024
05.08.2024

Lösung zur Erstklausur 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 1–12**. 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 | 26 | 18 | 24 | 16 | 16 | 100 |
| Erreicht | | | | | | |

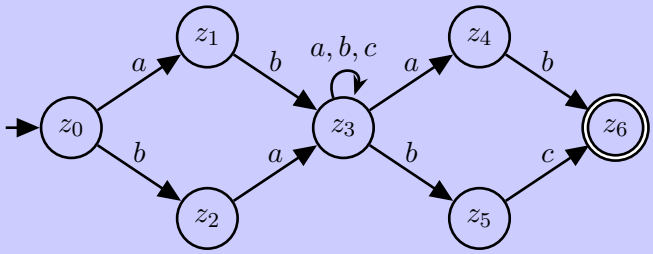
**Lösung Aufgabe 1 (Reguläre Sprachen):** **(26 Punkte)**

**a)** Sei $L_1$ die Sprache über dem Alphabet $\{a, b, c\}$, die vom regulären Ausdruck

$$(ab|ba)(a|b|c)^*(ab|bc)$$

erzeugt wird. Geben Sie den Zustandsgraphen eines nichtdeterministischen endlichen Automaten (ohne $\varepsilon$-Übergänge) an, der $L_1$ akzeptiert. (8 Punkte)

**LÖSUNGSVORSCHLAG:**



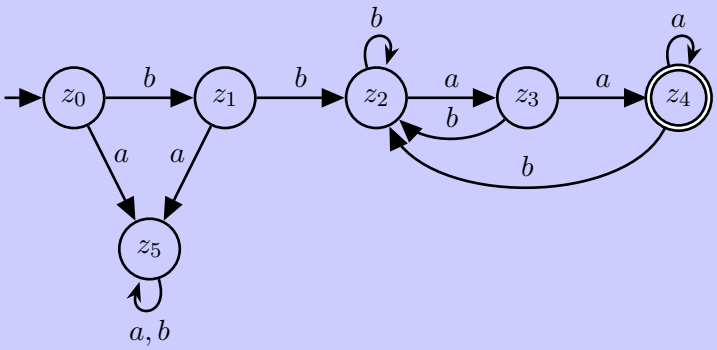

- 8 Punkte, wenn alles stimmt:
  - 1 Punkt Abzug bei fehlendem Startzustand
  - 1 Punkt Abzug bei fehlendem Endzustand
  - 1 Punkt Abzug bei jedem falschen Übergang oder Zustand

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

$$bb(a|b)^*aa$$

erzeugt wird. Geben Sie den Zustandsgraphen eines deterministischen endlichen Automaten an, der $L_2$ akzeptiert. Bitte zeichnen Sie alle Übergänge und Zustände sorgfältig, inklusive des Müllzustands (falls vorhanden). (8 Punkte)

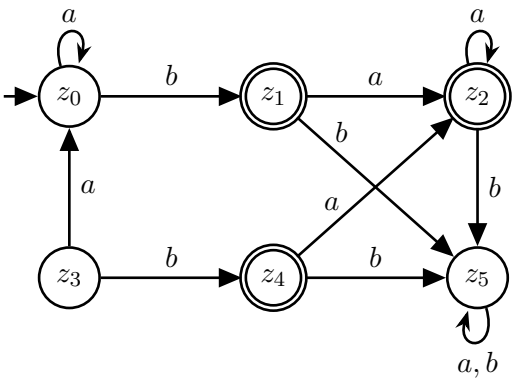
**LÖSUNGSVORSCHLAG:**




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

**c)** Minimieren Sie den folgenden deterministischen endlichen Automaten (d.h. konstruieren Sie einen deterministischen endlichen Automaten, der die gleiche Sprache akzeptiert und eine minimale Anzahl an Zuständen benutzt). Nutzen Sie dazu das Verfahren aus der Vorlesung. Geben Sie eine Partitionstabelle und den minimierten Automaten als Zustandsgraph an.

(10 Punkte)



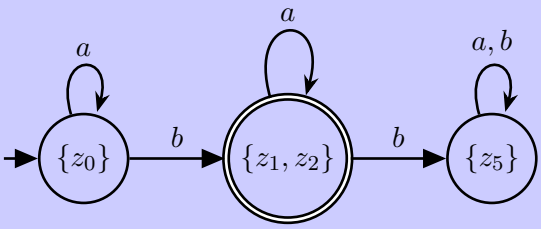

**LÖSUNGSVORSCHLAG:**

Unerreichbare Zustände: $z_3, z_4$.

Partitionstabelle:

0. $z_0 \quad z_5 \mid z_1 \quad z_2$
1. $z_0 \mid z_5 \mid z_1 \quad z_2$ mit $b$

Zustandsgraph:




- 1 Punkt für unerreichbare Zustände
- 6 Punkte für die Partitionstabelle:
  3 Punkte pro Zeile (inklusive „mit $b$“ für die zweite Zeile)
  2 Punkte Abzug, falls eine andere Minimierungsmethode verwendet wurde
- 3 Punkte für den minimierten Automaten:
  1 Punkt Abzug, wenn Start- und/oder Endzustand nicht gekennzeichnet sind
  1 Punkt Abzug bei jedem falschen Übergang oder Zustand

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

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

$$L_3 = \{ab^i cd^i \mid i \in \mathbb{N}\}$$

über dem Alphabet $\{a, b, c, d\}$ 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 $w = ab^n cd^n$. 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 $i \in \mathbb{N}$. Dann gibt es zwei Fälle:

Wenn $u = \varepsilon$, dann $v = ab^{n_1}$ und $w = b^{n_2} cd^n$ mit $n_1 + n_2 = n$. Dann ist $uv^0 w \notin L_3$, denn $uv^0 w = b^{n_2} cd^n$ und es gibt kein $a$. Widerspruch zur Annahme, dass $uv^i w \in L_3$ für jedes $i \in \mathbb{N}$.

Sonst ist $u = ab^{n_1}$, $v = b^{n_2}$ und $w = b^{n_3} cd$ mit $n_1 + n_2 + n_3 = n$. Dann ist $uv^0 w \notin L_3$, denn $uv^0 w = ab^{n_1+n_3} cd^n$ und es gibt weniger $b$'s als $d$'s. Widerspruch zur 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:
  - 1 Punkt Abzug, wenn der Fall $u = \varepsilon$ vergessen wurde
  - 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

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

über dem Alphabet $\{a, b, c, d\}$ nicht regulär ist. $\overline{L}$ bezeichnet dabei das Komplement einer Sprache $L$. Sie dürfen annehmen, dass die Sprache $L_3 = \{ab^i cd^i \mid i \in \mathbb{N}\}$ aus Teilfrage a) 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 $\overline{\overline{\{a^i b^j c d^j \mid i, j \in \mathbb{N}\}}} = \{a^i b^j c d^j \mid i, j \in \mathbb{N}\}$ auch regulär ist. Somit ist der Schnitt der offensichtlichen regulären Sprache $\{ab^i cd^j \mid i, j \in \mathbb{N}\}$ mit $\{a^i b^j c d^j \mid i, j \in \mathbb{N}\}$ auch regulär. Aber dies ist genau die Sprache $L_3$, die nicht regulär ist. Widerspruch.

- 1 Punkt: Annahme, dass $L_4$ regulär ist.
- 2 Punkte: Komplement ist regulär.
- 2 Punkte: $\{ab^i cd^j \mid i, j \in \mathbb{N}\}$ ist regulär.
- 2 Punkte: Schnitt ist regulär.
- 1 Punkt: Verweis auf Nichtregularität von $L_3$.

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

**a)** Die Sprache $L_5$ über dem Alphabet $\Sigma_1 = \{a, b, c, d, e\}$ sei definiert als

$$L_5 = \{ab^i c^j d^i e \mid i, j \in \mathbb{N}_{>0}\}$$

Geben Sie eine kontextfreie Grammatik $G_1$, die $L_5$ erzeugt, als 4-Tupel an. Die Grammatik darf keine $\varepsilon$-Produktionen enthalten. Erläutern Sie, warum $G_1$ die Sprache $L_5$ erzeugt. Beschreiben Sie beispielsweise, welche „Aufgabe“ die einzelnen Nichtterminale bei der Erzeugung übernehmen. (5 Punkte)

Geben Sie zusätzlich eine Linksableitung für das Wort *abbbccddde* für Ihre Grammatik an. (3 Punkte)

**LÖSUNGSVORSCHLAG:** $G_1 = (V, \Sigma_1, P, S)$ mit $V = \{S, T, U\}$ und

$$P = \{S \to aTe, T \to bTd \mid bUd, U \to Uc \mid c\}$$

$S$ erzeugt zunächst $a$ und $e$, mit $T$ in der Mitte. Aus $T$ lassen sich ein oder mehr $b/d$-Paare erzeugen, mit $U$ in der Mitte. Aus $U$ lassen sich ein oder mehr $c$'s erzeugen.

Linksableitung:
$S \Rightarrow aTe \Rightarrow abTde \Rightarrow abbTdde \Rightarrow abbbUddde \Rightarrow abbbUcddde \Rightarrow abbbccddde$.

- 5 Punkte für die Grammatik:
  - 3 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 $T$ und/oder $U$
- 3 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 $V_2 = \{S, T, U\}$, $\Sigma_2 = \{a, b, c, d, e\}$ und

$$P_2 = \{S \to TU, T \to aTb, T \to c, U \to dUe, U \to de\}$$

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

**LÖSUNGSVORSCHLAG:** $L(G_2) = \{a^i c b^i d^j e^j \mid i \in \mathbb{N}, j \in \mathbb{N}_{>0}\}$.

- 2 Punkte für Lösung von der allgemeinen Form $a^i c b^k d^\ell l e^m$
- 1 Punkt: richtige Anzahl an $a$'s, $b$'s
- 1 Punkt: $i \in \mathbb{N}$
- 1 Punkt: richtige Anzahl an $d$'s und $e$'s und eigene Variable $j$
- 1 Punkt: $j \in \mathbb{N}_{>0}$

**c)** Sei $G_3 = (V_3, \Sigma_3, P_3, S)$ mit $V_3 = \{S, A, B\}$, $\Sigma_3 = \{a, b\}$ und

$$P_3 = \{S \to AB \mid BA,\ A \to AA \mid AB \mid a,\ B \to BA \mid b\}$$

eine kontextfreie Grammatik.

Entscheiden Sie, ob $aababa \in L(G_3)$ ist, indem Sie den Algorithmus von Cocke, Younger und Kasami (CYK-Algorithmus) ausführen. Geben Sie die dabei entstehende Tabelle und die Ausgabe des Algorithmus an. (10 Punkte)

**LÖSUNGSVORSCHLAG:** Tabelle:

| Wort | $a$ | $a$ | $b$ | $a$ | $b$ | $a$ |
|---|---|---|---|---|---|---|
| $j\backslash i$ | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | $A$ | $A$ | $B$ | $A$ | $B$ | $A$ |
| 2 | $A$ | $S, A$ | $S, B$ | $S, A$ | $S, B$ | |
| 3 | $A, S$ | $S, A$ | $S, B$ | $S, A$ | | |
| 4 | $A, S$ | $S, A$ | $S, B$ | | | |
| 5 | $A, S$ | $S, A$ | | | | |
| 6 | $A, S$ | | | | | |

Da $S \in V(1, 6)$ ist, ist $aababa \in L(G_3)$.

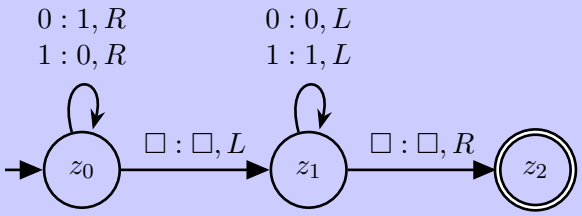
- 1 Punkt für richtige Antwort (ja)
- 9 Punkte für Tabelle
  - 1 Punkt Abzug per falsche Zelle

**Lösung Aufgabe 4 (Berechenbarkeit und Entscheidbarkeit): (16 Punkte)**

**a)** Geben Sie den Zustandsgraphen einer deterministischen Turingmaschine $M = (Z, \Sigma, \Gamma, \delta, z_0, \square, E)$ mit $\Sigma = \{0, 1\}$ an, sodass $Start_M(a_1 \cdots a_n) \vdash_M^* \square \cdots \square z(1 - a_1) \cdots (1 - a_n) \square \cdots \square$ für ein $z \in E$. Für die Eingabe 10110 würde die Turingmaschine zum Beispiel den Bandinhalt durch 01001 ersetzen, den Schreib-Lesekopf wieder zum Anfang der Ausgabe positionieren und anhalten.

(8 Punkte)

**LÖSUNGSVORSCHLAG:**




- 8 Punkte, wenn alles stimmt:
  - 1 Punkt Abzug bei fehlendem Startzustand
  - 1 Punkt Abzug bei fehlendem Endzustand
  - 1 Punkt Abzug bei jedem falschen Übergang oder Zustand
  - 1 Punkt Abzug bei fehlenden Übergängen aus $z_2$
  - 1 Punkt Abzug wenn $\varepsilon$ nicht terminiert

**b)** 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, oder erklären Sie, warum der Satz nicht anwendbar ist.

(i) Es ist unentscheidbar, ob eine gegebene deterministische Turingmaschine nach höchstens 1000 Schritten anhält. (3 Punkte)

**LÖSUNGSVORSCHLAG:** Der Satz ist nicht anwendbar, weil die Eigenschaft eine der Turingmaschine und nicht ihrer Sprache ist.

Die Sprache ist sogar entscheidbar. Man kann sie entscheiden, indem man versucht, die Turingmaschine 1001 Schritte laufen zu lassen. Wenn dies gelingt, ist die Antwort „Nein“. Sonst ist die Antwort „Ja“. Bei nichtdeterministschen Übergängen werden alle Möglichkeiten parallel (oder nacheinander) geprüft.

- 1 Punkt für richtige Antwort (nicht anwendbar)
- 2 Punkte für kurze Begründung

(ii) Es ist unentscheidbar, ob eine gegebene deterministische Turingmaschine für mindestens eine Eingabe $i \in \mathbb{N}$ die Zahl $i + 1$ berechnet. (5 Punkte)

**LÖSUNGSVORSCHLAG:** Der Satz ist anwendbar. Sei $\mathcal{S}$ die Menge aller (partiellen oder totalen) Funktionen $f$, sodass es $i$ mit $f(i) = i + 1$ 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:

$$\begin{aligned} C(\mathcal{S}) &= \{w \mid \text{die von der deterministischen Turingmaschine } M_w \text{ berechnete Funktion liegt in } \mathcal{S}\} \\ &= \{w \mid \text{die Turingmaschine } M_w \text{ berechnet eine Funktion } f, \text{ welche für mindestens eine Eingabe } i \in \mathbb{N} \text{ die Zahl } i + 1 \text{ berechnet}\} \end{aligned}$$

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

**Lösung Aufgabe 5 (Komplexität):** **(16 Punkte)**

**a)** Wir erinnern zunächst 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? |

In der Vorlesung wurde bewiesen, dass GRAPH-COLORING $\mathcal{NP}$-schwer ist. Zur Erinnerung bestand der Beweis aus folgenden Schritten:

- Wir zeigen 3-CNF-SAT $\leq_{\text{p}}$ GRAPH-COLORING.
- Sei $F = K_1 \wedge \cdots \wedge K_m$ eine 3-CNF, sodass jede Klausel $K_i$ genau 3 Literale enthält.
- Wir erzeugen ein GRAPH-COLORING-Problem mit $k = 3$ Farben.
- Wir konstruieren die „Palette“ und für jede Klausel ein „Oder-Gatter“.
- Der Graph ist $k$-färbbar g.d.w. $F$ erfüllbar ist.
- Die Übersetzung von $F$ in einen Graphen ist total und polynomiell.

Ein verwandtes Problem ist das GRAPH-3-COLORING-Problem:

| | |
|---|---|
| gegeben: | ein ungerichteter Graph $G = (V, E)$ |
| gefragt: | Gibt es eine Färbung der Knoten in $V$ mit höchstens **drei** Farben, sodass keine zwei benachbarten Knoten in $G$ die gleiche Farbe erhalten? |

Ist GRAPH-3-COLORING $\mathcal{NP}$-schwer? Begründen Sie kurz Ihre Antwort. (6 Punkte)

**LÖSUNGSVORSCHLAG:** Ja, denn die Reduktion 3-CNF-SAT $\leq_p$ GRAPH-COLORING, die wir in der Vorlesung gesehen haben, nutzt nur drei Farben. Daher kann man die gleiche Beweisidee für 3-CNF-SAT $\leq_p$ GRAPH-3-COLORING verwenden, um $\mathcal{NP}$-Schwere von GRAPH-3-COLORING zu beweisen.

- 2 Punkt für richtige Antwort (ja)
- 4 Punkte für „$\mathcal{NP}$-Schwere“-Beweis

**b)** In der Vorlesung wurden die Probleme CLIQUE, INDEPENDENT-SET und VERTEX-COVER vorgestellt. Das CLIQUE-Problem lässt sich so definieren:

| | |
|---|---|
| gegeben: | ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$ |
| gefragt: | Besitzt $G$ eine Clique der Größe mindestens $k$, d.h. eine Menge $V' \subseteq V$, sodass $\lvert V'\rvert \geq k$ und für alle $u, v \in V'$ mit $u \neq v$ gilt $\{u, v\} \in E$? |

Das INDEPENDENT-SET-Problem lässt sich so definieren:

| | |
|---|---|
| gegeben: | ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$ |
| gefragt: | Besitzt $G$ eine unabhängige Knotenmenge der Größe mindestens $k$, d.h., eine Menge $V' \subseteq V$ mit $\lvert V'\rvert \geq k$, sodass keine zwei Knoten aus $V'$ über eine Kante aus $E$ verbunden sind? |

Das VERTEX-COVER-Problem lässt sich so definieren:

| | |
|---|---|
| gegeben: | ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$ |
| gefragt: | Besitzt $G$ eine überdeckende Knotenmenge der Größe höchstens $k$, d.h. eine Menge $V' \subseteq V$ mit $\lvert V'\rvert \leq k$, sodass jede Kante aus $E$ mindestens einen ihrer beiden Knoten in der Menge hat? |

Sie dürfen als bekannt annehmen, dass diese drei Probleme $\mathcal{NP}$-vollständig sind.

Nun führen wir folgendes STRANGERS-Problem ein:

| | |
|---|---|
| gegeben: | eine Menge von Personen $P$,<br>eine binäre Relation $K \subseteq P \times P$, wobei $x\,K\,y$ intuitiv „$x$ kennt $y$“ heißt, und eine Zahl $n \in \mathbb{N}$ |
| gefragt: | Gibt es eine Teilmenge $S \subseteq P$ mit $\lvert S\rvert \geq n$, sodass alle Elemente von $S$ „Fremde“ sind (formal: $\neg\, s\,K\,t$ für alle $s, t \in S$)? |

Beweisen Sie mithilfe einer Polynomialzeit-Reduktion, dass STRANGERS $\mathcal{NP}$-schwer ist.

(10 Punkte)

**LÖSUNGSVORSCHLAG:** Wir zeigen INDEPENDENT-SET $\leq_p$ STRANGERS.

Sei $((V, E), k)$ eine INDEPENDENT-SET-Instanz. Wir setzen $f((V, E), k) = (P, K, n)$ mit $P = V$, $K = \{(x, y) \mid \{x, y\} \in E\}$ und $n = k$.

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

Korrektheit:

$((V, E), k) \in$ INDEPENDENT-SET
g.d.w. $G$ besitzt eine unabhängige Knotenmenge der Größe mindestens $k$
g.d.w. es gibt $V' \subseteq V$ mit $\lvert V'\rvert \geq k$, sodass $\{u, v\} \notin E$ für alle $u, v \in V'$
g.d.w. es gibt $S \subseteq P$ mit $\lvert S\rvert \geq n$, sodass $\{s, t\} \notin E$ für alle $s, t \in S$
g.d.w. es gibt $S \subseteq P$ mit $\lvert S\rvert \geq n$, sodass $(s, t) \notin \{(x, y) \mid \{x, y\} \in E\}$ für alle $s, t \in S$
g.d.w. es gibt $S \subseteq P$ mit $\lvert S\rvert \geq n$, sodass $\neg\, s\,K\,t$ für alle $s, t \in S$
g.d.w. $(P, K, n) \in$ STRANGERS

- 2 Punkte: „INDEPENDENT-SET $\leq_p$ STRANGERS“ (eventuell „CLIQUE $\leq_p$ STRANGERS“)

- 4 Punkte: Definition von $f$
  - 1 Punkt für $P$
  - 2 Punkte für $K$
    - 0 Punkte, falls $K$ einfach als $E$ defininiert oder die ungerichtete Natur von $E$ sonst nicht berücksichtigt
  - 1 Punkt für $n$
- 1 Punkt: $f$ ist (total und) polynomiell
- 3 Punkte: „g.d.w.“-Beweis
  - 2 Punkte für Ausfalten der Definitionen von INDEPENDENT-SET und STRANGERS
  - 1 Punkt für Ausfalten der Definition von $K$
- Insgesamt höchstens 2 Punkte, falls die Reduktion falsch herum ist