פקולטה: מדעי הטבע

מחלקה: מדעי המחשב

שם הקורס: חישוביות

קוד הקורס: **7035910**

תאריך הבחינה: 3/11/2022 ט' חשוון תשפ"ג

סמסטר: קיץ, מועד: ב'

משך הבחינה: שלוש שעות

מרצי הקורס: גברת שני שוב, מר דורון מור

מתרגל הקורס: מר ניר סון

חומר עזר: שני דפי נוסחות רשומים בידי הסטודנט (ארבעה עמודים). אין צורך להגיש את הדף בסוף המבחן.
אין צורך בשימוש במחשבון, אם כי מותר לעשות זאת.

פירוט הניקוד לפי שאלה:
שאלה 1 – 45 נקודות (9 נקודות לכל סעיף).
שאלה 2 – 12 נקודות.
שאלה 3 – 18 נקודות (6 נקודות לכל סעיף).
שאלה 4 – 10 נקודות (5 נקודות לכל סעיף).
שאלה 5 – 15 נקודות (3,3,9 בהתאמה).

הוראות כלליות:
- יש לענות על כל השאלות.
- יש להוכיח כל טענה בה הנכם משתמשים, אלא אם יש הנחיה אחרת בשאלה, או שזו טענה מקורס קודם (פרט לאלגוריתמים2 של אלעד).
- במידה והנכם מתבססים על טענה שהוכחה בכיתה, יש לצטט אותה במדויק.
- מספיק לתת תיאור כללי של מכונות טיורינג שאתם בונים.

1. נניח כי $P \neq NP$. סווגו את השפות הבאות למחלקות $P, NP, coNP, \overline{NP \cup coNP}$, לא ידוע. סווגו כל שפה למחלקה המדויקת ביותר והוכיחו את תשובתכם. (9 נקודות לכל סעיף). [נניח, אם כתבתם ששפה ב NP כשלמעשה היא ב P, הפתרון יזכה לכל היותר ב-3 נקודות]

א. $BigIS = \{< G >| G\ is\ undirected\ graph\ and\ exists\ a\ subset\ of\ G\ of\ 1000\ vertices\ that\ forms\ an\ independent\ set\}$
(גרף שיש בו קבוצה בלתי תלויה בגודל 1000 קודקודים.)

<u>פתרון</u>

השפה ב-P. (ולכן גם ב $NP, coNP$)

המכונה N על הקלט $< G >$:

- עוברת על כל תתי הקבוצות של הקודקודים בגודל 1000:
- עוברת על כל זוג קודקודים בקבוצה – אם יש צלע כלשהי ביניהם- עוברים לאיטרציה הבאה.
- אם אין צלעות בכלל בכל תת הקבוצה- המכונה מקבלת.
- דוחה.

המכונה פולינומית (מעבר על $O(n^{1000})$ קבוצות קודקודים, ובכל קבוצה על $O(1000^2)$ זוגות). אם קיימת קבוצה ב"ת בגודל 1000, נגיע אליה ובה המכונה תקבל. אחרת, בסוף כל הקבוצות האפשריות, נדחה.

ב. $Exactly-5-SAT = \{\phi | \phi\ is\ a\ CNF\ formula\ that\ has\ exactly\ 5\ satisfying\ assignments\}$
(נוסחת CNF שיש לה בדיוק 5 השמות מספקות)

<u>פתרון</u>

לא ידוע באיזו מחלקה השפה. אפשר בקלות למצוא חמישה עדים של השמות מספקות, אם יש, אבל כדי לוודא שאין בכלל השמות מספקות אחרות צריך לעבור על כל ההשמות האפשרויות... כנ"ל לגבי המשלים- כדי לוודא שאין בדיוק חמש השמות, צריך לעבור על כל ההשמות. דומה לשאלה 4ב ממועד א'.

ג. $L = \left\{< G, M > \middle| \begin{array}{c} G\ is\ a\ graph\ and\ M\ is\ a\ TM \\ s.t.\ M\ accepts < M > and\ G\ has\ IS\ of\ size\ 3 \end{array} \right\}$
(גרף ומכונה, כך שהמכונה M מקבלת את $< M >$ וגם לגרף יש קבוצה בלתי תלויה בגודל 3)

<u>פתרון</u>

השפה בחוץ, כלומר ב $\overline{NP \cup coNP}$. נוכיח ברדוקציה מ L_D.

$$f(< M >) \rightarrow (< G >, < M >)$$
כאשר המכונה M נשארה זהה, והגרף G הוא שלושה קודקודים ללא צלעות.

הרדוקציה פולינומית: להעתיק את המכונה, ולייצר גרף קטן מאוד.
הרדוקציה תקפה:
אם $< M > \in L_D$ אזי M מקבלת את $< M >$. לכן, $f(< M >)$ הוא גרף עם קבוצה בלתי תלויה בגודל 3, ומכונה המקבלת את הקידוד של עצמה.
אם $< M > \notin L_D$ אזי M אינה עוצרת על או אינה מקבלת את $< M >$. לכן $f(< M >) \notin L$, שכן המכונה M אינה מקבלת את $< M >$.

לפי משפט הרדוקציה, $L \notin R$ ולכן $L \in \overline{NP \cup coNP}$.

ד. $Non-photogenic = \{< G >| G\ cannot\ be\ colored\ by\ 5\ colors\}$
(הגרף G לא יכול להיצבע חוקית בחמישה צבעים)

<u>פתרון</u>

השפה ב $coNP$. נוכיח כי $\overline{Non-photogenic} = \{< G >| G\ is\ 5-colorable\} \in NP$.
המכונה N על הקלט $< G >$:

- מנחשת צביעה בחמישה צבעים.
- עוברת על כל הצלעות- אם נמצאה צלע ששני קצותיה צבועים באותו צבע, דוחה.
- מקבלת.

המכונה פולינומית: ניחוש באורך כמות הקודקודים (אפילו לוגריתמי בקידוד), וידוא שלכל צלע אין צבע זהה בשני הקצוות- ככמות הצלעות (ליניארי).
תקפות: אם הגרף 5-צביע, אזי המכונה מנחשת את הצביעה ומקבלת. אחרת, כל ניחוש יוביל לדחייה.

ה. $L_1 = \left\{< G,k > \middle| \begin{matrix} G\ is\ a\ graph\ with\ 3\ vertex-disjoint\ cliques\ of \\ sizes\ k,k-1,k-2 \end{matrix}\right\}$
(בגרף G יש שלוש קליקות זרות בקודקודים, בגדלים $k,k-1,k-2$)

<u>פתרון</u>

נוכיח שהשפה קשה ב-NP. (אפשר לסווג אותה ל NPC, אבל לא התבקשנו)
נוכיח ע"י כך שנראה רדוקציה פולינומית מהשפה CLQ שהוכח עליה היא שלמה ב-NP, בגלל טרנזטיביות הרדוקציה ינבע שגם L_1 קשה ב-NP.

$$CLQ \leq_p L_1$$

פונקציית הרדוקציה תהיה: $f(< G,k >) = < G',k >$

כאשר G' הוא הגרף G בתוספת שני רכיבי קשירות נוספים, הראשון קליקה בגודל $k-1$ והשני קליקה בגודל $k-2$.

<u>זמן ריצה:</u> הרדוקציה פולינומית כיוון שהוספת $O(k^2)$ קודקודים וצלעות הוא פולינומי בגודל בקלט.

הרדוקציה מלאה וניתנת לחישוב כיוון שלכל מילה $< G,k >$ הרדוקציה מוגדרת והיטב.

<u>תקפות:</u>

אם $< G,k > \in CLQ$ אז ב-G תהיה קליקה בגודל k, אם כך ב-G' יהיו קליקות בגודל $k,k-1,k-2$ ולכן $< G',k > \in L_1$

אם $< G,k > \notin CLQ$ אז ב-G אין קליקה בגודל k, אם כך ב-G' יהיו קליקות בגודל $k-1,k-2$ אך לא בגודל k ולכן $< G',k > \notin L_1$

2. נגדיר מחלקה חדשה:
$NP_{eq} = \{L | There\ exists\ a\ non-deterministic\ 2-tape\ polynomial\ Turing\ Machine\ M\ such\ that\ L = L(M). Moreover, for\ every\ x, all\ paths\ of\ the\ computation\ tree\ of\ M\ on\ x\ are\ of\ the\ same\ length\}$
בעברית: קבוצת כל השפות שקיימת מ"ט א"ד דו-סרטית פולינומית, בה לכל x, כל מסלולי החישוב על x דורשים אותו מספר צעדים.
הוכיחו כי $NP \subseteq NP_{eq}$. (12 נקודות)
הערה: בתרגיל מוגדר "דו-סרטית" מטעמי נוחות, שכן בפתרון הסגל המכונה דו-סרטית. אם מישהו/י מעדיף להשתמש במכונה חד-סרטית, כל עוד הפתרון נכון, יינתנו מלוא הנקודות.

<u>פתרון</u>

תהי $L \in NP$. צ"ל $L \in NP_{eq}$.
לפי ההנחה, קיימת מ"ט מכריעה א"ד פולינומית N_L עבור L, עם זמן ריצה $p(n)$.
נגדיר את המכונה הבאה:
המכונה N_{eq} על קלט x:
- חשב את $m = p(|x|)$.
- רשום את m בסרט השני בתור מונה ($counter$)
- סמלץ את ריצת N_L על x, כאשר בכל צעד, הורד את $counter$ באחד.
- אם N_L קיבלה או דחתה כאשר $counter > 0$, בצע עוד $counter$ מסעי אפסילון (מיותרים, סוג של ריפוד) וקבל/דחה בהתאם לתשובת N_L.

המכונה N_{eq} פולינומית, כי N_L פולינומית, ומספר הצעדים בכל מסלול שווה לזמן חישוב $p(|x|)$ + זמן אתחול $counter$ + זמן סמלוץ לפי $counter$ צעדים (תשלום log על סמלוץ מכונה. זה עדיין פולינומי).
נכונות נובעת מכך שמריצים את N_L ועונים כמוה. לכן $L(N_{eq}) = L(N_L) = L$.
ולכן $L \in NP_{eq}$ כנדרש.

3. לכל אחת מהשפות הבאות, קבעו והוכיחו האם היא ב-R או ב-RE או לא בשתיהן. (6 נקודות לכל סעיף)
תזכורת: $L_\epsilon = \{< M > | \epsilon \in L(M)\}$, $L_{\Sigma^*} = \{< M > | L(M) = \Sigma^*\}$.

א. $L_1 = \{< M > | L(M) \cap L_\epsilon = L_{\Sigma^*}\}$

<u>פתרון</u>

לכל מ"ט M מתקיים ש- $L(M) \in RE$. ראינו ש-$L_\epsilon \in RE$ וכן ש- RE סגורה תחת חיתוך, ולכן $L(M) \cap L_\epsilon \in RE$. מכיוון ש-$L_\epsilon \in RE$ לא ייתכן ש-$L(M) \cap L_\epsilon = L_{\Sigma^*}$. ולכן $L_1 = \phi$ שכידוע כריעה. (כלומר, $L_1 \in R$)

ב. $L_2 = \{< M > | L(M) \cup L_{\Sigma^*} \notin RE\}$

<u>פתרון</u>

נשתמש במשפט רייס. נגדיר את התכונה $S = \{L \in RE : L \cup L_{\Sigma^*} \notin RE\}$.

תכונה זו איננה טריוויאלית משום ש:

- $\phi \in RE$ ומקיימת את התכונה כי $\phi \cup L_{\Sigma^*} = L_{\Sigma^*} \notin RE$.
- $\Sigma^* \in RE$ ולא מקיימת את התכונה כי $\Sigma^* \cap L_{\Sigma^*} = \Sigma^* \in RE$.

בנוסף, $L_s = \{< M >: L(M) \cup L_{\Sigma^*} \notin RE\} = L_2$. לכן, לפי משפט רייס המורחב ומשום ש- ϕ מקיימת את התכונה, נקבל ש $L_s = L_3 \notin RE$.

ג. יהי $n > 0$ טבעי כלשהו. תהי $L_n = \{< M_1, M_2, \ldots M_n > |$
$M_1, M_2, \ldots, M_n are\ TM\ such\ that\ exists\ a\ word\ of\ length\ smaller\ than\ n\ in$
$L(M_1) \cap L(M_2) \ldots L(M_n)\}$
(שפת קידודי n מכונות עבורן קיימת מילה שאורכה עד n בחיתוך שפת כל המכונות)

<u>פתרון</u>

נגדיר את המכונה האי-דטרמיניסטית הבאה:
המכונה N_{L_n}:
- מנחשת מילה x שאורכה עד n.
- עבור $i = 1,2, \dots, n$:
  - מריצה את M_i על x, אם דחתה, דוחה.
-מקבלת.

נכונות: אם $< M_1, M_2, \dots M_n >\in L_n$ אזי קיימת מילה שאורכה חסום ב-n ומתקבלת בכל המכונות האלו. המכונה N_{L_n} תנחש אותה, ותקבל.
אם $< M_1, M_2, \dots, M_n >\notin L_n$ אזי לא קיימת מילה באורך זה המתקבלת בכולן. לכן, לכל ניחוש של N_{L_n}, לפחות אחת המכונות תדחה את המילה, וכך גם N_{L_n}.

כעת נראה כי $L_n \notin R$ ברדוקציה מ HP:

$$f(< M, x >) \to (< M_x, M_x, M_x, \dots >)$$

המכונה M_x על קלט y:
- מריצה את M על x.
- מקבלת.

הרדוקציה מוגדרת לכל קלט וניתנת לחישוב (בפרט, אין בה הרצת מכונה שעלולה לא לעצור).
נכונות:
$< M, x >\in HP$ אזי M עוצרת על x. לכן, המכונה M_x תקבל כל y, כלומר $L(M_x) = \Sigma^*$ ובפרט $L(M_x) \cap L(M_x) \dots \cap L(M_x) = L(M_x) = \Sigma^*$ ולכן קיימת מילה בשפת החיתוך שאורכה חסום ב-n.

$< M, x >\notin HP$ אזי M לא עוצרת על x. לכן, המכונה M_x לא עוצרת, ולכן שפת החיתוך ריקה. מכאן, שאין מילה בשפת החיתוך בכלל, ובפרט, אין מילה שאורכה חסום ב-n.

4. הוכיחו/הפריכו את התכונות הבאות: (5 נקודות לכל סעיף)
   א. אם $L_1, L_2 \in NP \cap coNP$ אזי $L_1 \Delta L_2 \in NP \cap coNP$.

<u>פתרון</u>
הקדמה – היות והשפות נמצאות בחיתוך של $NP, coNP$, יש להן סגירות למשלים, חיתוך ואיחוד. שהרי $\overline{NP} = coNP$, אלא שהשפות הנוכחיות נמצאות בחיתוך שלהם, ולכן גם המשלימה שלהם נמצאת בחיתוך.
הוכחה. יהיו $L_1, L_2 \in NP \cap coNP$. לכן $L_1, L_2 \in NP$ ומכאן גם $\overline{L_1}, \overline{L_2} \in NP$ (מהגדרת $coNP$).
ניזכר בהגדרת הפרש סימטרי:

$$\begin{aligned} L_1 \Delta L_2 &= (L_1 \setminus L_2) \cup (L_1 \setminus L_2) \\ &= (L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2) \end{aligned}$$

היות ו-$NP \cap coNP$ סגורה לחיתוך, נקבל כי $L_1 \cap \overline{L_2} \in NP$. בנוסף $\overline{L_1} \cap L_2 \in NP$. היות ו-$NP \cap coNP$ סגורה לאיחוד, כל הביטוי עדיין שייך ל $NP \cap coNP$.

   ב. אם $L_1 \in NP$ וגם $L_1 \leq_p L_2$ אזי $L_2 \in NP$.

<u>פתרון</u>

הפרכה. בקורס הוכחנו ש-$HP \in NPH$ וגם $HP \notin NP$ ולכן, עבור כל שפה מ-NP קיימת רדוקציה פולינומית אל HP אבל אין זה מוכיח שהיא ב-NP בעצמה.
(באותה מידה אפשר לכתוב ש HP היא RE-שלמה, ולכן מכל שפה שאינה טריוויאלית ב-R יש רדוקציה אליה. אבל $HP \in RE \setminus R$ ולכן בפרט $HP \notin NP$ שכן $NP \subseteq R$.

5. א. נגדיר מודל חדש של מכונת טיורינג. המכונה M^* נעה על הסרט בצורה הבאה:

$\{LLL, R^k\}$

כלומר, היא יכולה לבחור לבצע שלושה צעדים שמאלה, או לבצע כמות סופית כלשהו של צעדים ימינה, $k \le 0$).

הוכיחו/הפריכו: המודל החדש שקול למכונה רגילה. (9 נקודות)

<u>פתרון</u>

המודלים שקולים. כדי להוכיח זאת, נראה כיצד המודל החדש מממש את הצעדים המקוריים L, S, R, וכיצד המודל החדש מממש את הצעדים החדשים LLL, R^k.

כיוון ראשון:

כדי לממש צעד L, יש לבצע R^2 ולאחריו LLL. כלומר, בכל פעם שמכונה במודל המקורי מבצעת L, המכונה במודל החדש תבצע R^2 ותיכנס למצב q_{LLL}, ותבצע LLL מבלי לשנות את תוכן התא בצעד זה.

כדי לממש צעד R המכונה החדשה תבצע R^1. כדי לממש צעד S המכונה החדשה תבצע R^0.

כיוון שני:

בכל פעם שהמכונה החדשה מבצעת צעד LLL ועוברת ממצב q_i ל q_j, יש לבצע את הצעדים הבאים:

- צעד L שבו נכנסים למצב q^{iL}.
- צעד L מבלי לשנות את תוכן הסרט, ונכנסים למצב q^{iLL}.
- צעד L מבלי לשנות את תוכן הסרט, ועוברים למצב q_j.

כלומר, לכל מצב הוספנו עוד שני מצבים חדשים, q^{iL}, q^{iLL}.

נסרוק את δ ונמצא מהו k המקסימלי (נתון כי k סופי).

בכל פעם שהמכונה החדשה מבצעת צעד R^k ועוברת ממצב q_i ל q_j, יש לבצע את אותו רעיון, אבל k פעמים במקום 3 פעמים.

כלומר, לכל מצב הוספנו עוד k מצבעים חדשים $q^{iR}, q^{iRR}, \ldots, q^{iR^{k-1}}$. היות ו-$k$ סופי, התהליך אכן סופי, וכמות המצבים שהוספנו סופית גם היא.

ב. נגדיר מודל חדש: מכונה שאינה כותבת. המכונה עושה את כל הדברים של מכונה רגילה, אבל היא לא יכולה לשנות את תוכן הסרט כלל.

הוכיחו/הפריכו:

1. כל שפה רגולרית ניתן להכריע עם מכונה ממודל זה. (3 נקודות)
2. כל שפה שניתן לקבל במכונה ממודל זה, היא ב-R. (3 נקודות)

<u>פתרון</u>

1. היות ו"לקחנו" את יכולת הכתיבה של המכונה, כל שנותר לה הוא לעבור על הסרט לשני הכיוונים, ולעבור בין מצבים. לכן, קיבלנו אוטומט עם יכולת נוספת של חזרה אחורה בקלט. בפרט, למדנו בקורס אוטומטים שאוטומט, אפילו דטרמיניסטי, הוא המודל הקטן ביותר שכוחו שקול לשפות רגולריות (משפט קליני, הרצאה 4, שם שם). לכן, מכונת טיורינג שאינה כותבת, שכוחה גדול יותר מאוטומט רגיל בכך שהיא יכולה לקרוא את הקלט כמה פעמים ובשני הכיוונים, בוודאי יכולה להכריע שפות רגולריות.

2. תהי שפה שניתן לקבל עם מכונה ממודל זה. כלומר, $\forall x \in L$ מתקיים $M(x) = 1$. נבנה מכונה מכריעה לשפה L במודל הרגיל:

המכונה M_{reg} על הקלט x:

- מריצה את M על x למשך $|Q(M)| \cdot |x+1|$ צעדים, ומתעדת את הקונפיגורציות בדרך.
- אם הגענו לקונפיגורציה מקבלת – מקבלת.
- אם הגענו לקונפיגורציה דוחה – דוחה.
- אם חזרנו לקונפיגורציה פעמיים- דוחה.
- אם סיימנו את כל הקונפיגורציות ללא הגעה לקונפיגורציה מקבלת – דוחה.

נכונות: אם $x \in L$ אזי המכונה M תקבל את x (תגיע לקונפיגורציה מקבלת) ולכן M_{reg} תקבל את x.
אם $x \notin L$ אזי המכונה M לא תקבל את x או לא תעצור. היות והיא לא יכולה לכתוב, כמות הקונפיגורציות שלה אינה תלויה ב $|\Gamma|$, כי אם רק באורך הקלט המקורי ובכמות המצבים. לכן, לאחר כמות סופית של קונפיגורציות, היא תחזור על אותה קונפיגורציה פעמיים, ונדחה; או שנגיע מתישהו לקונפיגורציה דוחה ונדחה; בכל מקרה, לא צריך לרוץ לנצח, אלא אפשר לדחות.

בהצלחה!