# 76 שאלות חשובות והתשובות אליהן

| | |
|---|---|
| 1. PATH $\le_L$ SAT and SAT $\le_L$ PATH | 39. AUB is NP-complete |
| 2. MINIMAL –WORD$_{TM}$ | 40. SUB cnf |
| 3. Degree C={<G,k,d>\| … | 41. CLIQUE-SUB |
| 4. IND | 42. RL מוכל SPACE(log2n) |
| 5. Verifier – מאמת | 43. NPC |
| 6. SHORTEST | 44. A in P, AUB in P, is B in P |
| 7. A-B is NL-complete | 45. EHAMPATH-AND-IS |
| 8. UNION-ALL | 46. DOUBLE |
| 9. ALL-EXCEPT_ONE | 47. ALMOST-HP |
| 10. מונה שמפיק וכל מילה מודפסת פעם אחת ויחידה | 48. MAJORITY-PATH |
| 11. E to HALT | 49. XOR-HALT |
| 12. SHORTEST-PATH | 50. שייכות ל P |
| 13. שני מונים | 51. RP תחת איחוד |
| 14. E lba | 52. RP תחת חיתוך |
| 15. UHAMCIRCUIT | 53. Acc, Rej, Loop |
| 16. E dfa | 54. A <m A משלים |
| 17. מזוהה טיורינג במוסכמה חדשה | 55. SAME-COLR |
| 18. ביצוע ללואה במכונת טיורינג | 56. 2PATHS |
| 19. DECIDER | 57. BPP שייכות ל |
| 20. 2VC | 58. Fm(n) מספר צעדים מקסימלי |
| 21. MAX(C) – מילה מקסימלית | 59. NECESSARY |
| 22. A* | 60. דרגת כניסה ויציאה בגרף לא מכוון |
| 23. COMP-CONC | 61. POSITIVE-CNF in P |
| 24. רדוקציה בזמן ריבועי | 62. OVERLAP dfa in NL-complete |
| 25. מכונת טיורינג עם מספר אינסופי של מצבים | 63. MOVE-FIRST in RP |
| 26. INFINTE | 64. NP and coNP != P |
| 27. A nfa with LBA | 65. סימולטור מכונת טיורינג |
| 28. חשיבה בזמן לינארי/ריבועי | 66. PARTITION |
| 29. שיוויון פולינומיאלי | 67. A nfa in P |
| 30. שפה רגולרית O(1) | 68. CF tm <m A tm, !CF <m A tm |
| 31. CLIQUE <L VC | 69. DROP-MIDDLE in PSPACE |
| 32. מזוהה טיורינג לא ריקה אז יש פונקציה הניתנת לחישוב | 70. סגורה לאיחוד BPP |
| 33. זהה ל 2 | 71. מכונת טיורינג עם סרט יחיד ותנאים נוספים |
| 34. MIN(C) | 72. SUB tm <m A tm, !SUB <m A tm |
| 35. CLIQUE-AND-INDEPENDENT-SET | 73. B <l A and A <p B and check NP-complete |
| 36. שקולות -L | 74. BIHAMPATH מה לא בסדר בהוכחה |
| 37. DROP-SYMBOL | 75. HALF-CONN <l UPATH |
| 38. NOEMPTY-INTER | 76. EQ lba |

# אוסף שאלות במבוא לתורת החישוביות והסיבוכיות

1. עבור כל אחת מהטענות הבאות, קבעו האם היא נכונה או לא. נמקו את קביעתכם.
   א. אם יימצא כי $PATH \leq_L SAT$, אפשר יהיה להסיק ש- $P = NP$.
   ב. אם יימצא כי $SAT \leq_L PATH$, אפשר יהיה להסיק ש- $P = NP$.

**פתרון:**

א. ?

ב. הטענה נכונה. נניח ש- $SAT \leq_L PATH$. כלומר, קיימת מכונת טיורינג $F$ הרצה במקום לוגריתמי, $c\log n$, המחשבת רדוקציית מיפוי של $SAT$ ל- $PATH$. מהדיון בראש עמוד 351 נקבל שזמן הריצה של $F$ לא עולה על $c\log n \cdot 2^{c\log n} = O(n^2)$, כלומר $F$ היא בעצם רדוקציה פולינומית בזמן. לכן $SAT \leq_P PATH$. ממשפט 7.31 נקבל ש- $SAT \in P$ ואז ממשפטים 7.35 ו-7.37 נקבל ש- $P = NP$. מש"ל.

2. תהי $C$ שפה. מילה $w$ נקראת **מילה מינימלית** בשפה $C$ אם $w \in C$ אבל כל **תחילית ממש** של $w$ לא שייכת ל- $C$.
(מילה $v$ היא **תחילית** של מילה $w$ אם יש מילה $u$ כך ש- $w = vu$. אם $u \neq \varepsilon$ אז $v$ נקראת **תחילית ממש**.)
נגדיר את השפה $MINIMAL-WORD_{TM}$ הבאה:
$$MINIMAL-WORD_{TM} = \{<M, w> | \ M \ is\ a\ TM; w \text{ is minimal word in } L(M)\}$$
הראו **רדוקציית מיפוי** של $A_{TM}$ ל- $MINIMAL-WORD_{TM}$ (הראו: $A_{TM} \leq_m MINIMAL-WORD_{TM}$).

**פתרון:**

$F$= "קלט: $<M, w>$ כאשר $M$ היא מכונת טיורינג ו- $w$ מילה.
1. בנה את המכונה $M'$ הבאה:
   $M'$= "קלט: $x$, מילה.
   1. הרץ את $M$ על $w$. אם $M$ קיבלה, קבל אם $x = a$. אם דחתה, דחה. ($a \in \Sigma$ כאשר $\Sigma$ הוא האלפבית של $M$) "
2. החזר $<M', a>$."

**נכונות:**

ברור שהרדוקציה עוצרת, שכן כל שהיא עושה הוא בניית מכונה אחרת והחזרה.
כעת, נניח ש- $<M, w> \in A_{TM}$. פירוש הדבר הוא ש- $M$ עוצרת על $w$ ומקבלת אותה. אז במקרה זה $L(M') = \{a\}$, ואכן $a$ היא מילה מינימלית ב- $L(M')$ ולכן $<M', a> \in MINIMAL-WORD_{TM}$.
נניח ש- $<M, w> \notin A_{TM}$. אז או ש- $M'$ לא תעצור או שהיא תדחה את $w$. בשני המקרים $L(M') = \emptyset$ והיות ש- $a \notin L(M')$ יתקיים $<M', a> \notin MINIMAL-WORD_{TM}$, כדרוש.
מש"ל.

3. נסמן את ה**דרגה** של צומת $v$ בגרף לא מכוון (מספר הקשתות שנוגעות בצומת) ב- $d(v)$.
הוכיחו שהשפה הבאה היא **NP- שלמה**:
$$C = \{<G, k, d> | \ G = (V, E) \ is\ an\ undirected\ \ graph, \ \exists E' \subseteq E$$
$$(|E'| \leq k, G' = (V, E') \ is\ connected, \forall v \in V(d(v) \leq d))\}$$

**הדרכה:** מהו ה- $k$ המינימלי האפשרי כך שהגרף $G'$ יהיה קשיר? מהי הדרגה $d$ המינימלית האפשרית במקרה זה?

**פתרון:** תחילה, ניתן לבנות מאמת שיקבל כקלט אישור שהוא קבוצה $E'$ כזו. הבדיקה האם הגרף החדש קשיר ומהי דרגת כל צומת היא בבירור פולינומית.
לכן מתקיים $C \in NP$.
כעת, נראה רדוקציה בזמן ריצה פולינומי של $HAMPATH$ ל- $C$.
קלט: $<G,s,t>$, כאשר $G$ הוא גרף מכוון.

1. הוסף לגרף צומת $s'$ וקשת ממנו לצומת $s$.
2. הוסף לגרף צומת $t'$ וקשת אליו מהצומת $t$. נאמר שהגרף החדש הוא $G'$.
3. מנה את מספר הצמתים ב- $G'$. נאמר שהוא $n$.
4. הוצא לפלט $<G',n-1,2>$.

**זמן ריצה**: כל השלבים אורכים זמן ריצה פולינומי בבירור.
**נכונות**: נניח ש- $<G,s,t> \in UHAMPATH$. פירוש הדבר הוא שקיים מסלול המילטוני $P = su_1u_2...u_{n-3}t$, כאשר כל הצמתים בו שונים. אז בגרף החדש קיים המסלול $P' = s'su_1u_2...u_{n-3}tt'$, שהוא מסלול בעל $n$ צמתים שונים ולכן הוא גם תת גרף קשיר של $G'$. בפרט הוא מכיל את אותם צמתים, מכיל $n-1$ קשתות כדרוש, וכל צומת הוא בעל שכן אחד או שניים לכל היותר, ולכן $d(v) \le 2$ לכל $v \in V$ כדרוש.
כעת נניח ש- $<G',n-1,2> \in C$. נתבונן בתת גרף המובטח על פי הגדרת $C$. היות שהוא קשיר בעל $n-1$ קשתות נקבל שהוא עץ פורש של הגרף המקורי.
נתבונן באלגוריתם הבא:

1. נבחר עלה כלשהו בעץ (קיים כזה כי העץ הוא בעל יותר משני צמתים). סמן ב- $x$ את שכנו וב- $y$ את העלה עצמו.
2. אם הדרגה של $y$ היא 1 – סיים.
3. אם הדרגה של $y$ היא 2 – לך אל עבר השכן של $x$ ששונה מ- $y$. סמן $x \leftarrow y$ וסמן את השכן ב- $x$. חזור ל-2.

באלגוריתם זה, לא ייתכן שבשלב 3 ייבחר צומת שכבר היינו בו (פרט ל- $y$) כי אז היה מעגל בסתירה לכך שזהו עץ. מכיוון שזהו גרף קשיר והאלגוריתם הזה זהה בפעולתו ל- $BFS$, נקבל שמגיעים לכל הצמתים. פירוש הדבר הוא שכל הצמתים פרט לעלה הראשוני הם מדרגה 2, פרט לצומת האחרון – שחייב להיות מדרגה 1. כלומר העץ הפורש הוא "שרוך", כלומר מסלול שעובר בכל צמתי הגרף בדיוק פעם אחת. היות שהוא לא יכול להגיע ל- $s'$ מצומת אחר ולצאת ממנו בלי לעבור על אותה קשת פעמיים $s'$ חייב להיות צומת מדרגה 1, צומת תחילת המסלול, ו- $t'$ חייב להיות סוף המסלול בדומה. בהסרת $s'$ ו- $t'$ נקבל מסלול המילטוני בין $s$ ל- $t$ בגרף המקורי. מש"ל.

4. יהי $G=(V,E)$ גרף לא מכוון.

לכל קבוצה $S$ של צמתים ($S \subseteq V$) נגדיר את ה**אי-תלות** של $S$ כמספר הצמתים ב- $S$ פחות מספר הקשתות שמחברות שני צמתים ב- $S$.
פורמלית, $Independence(S) = |S| - |\{\{u,v\} \in E \mid u,v \in S\}|$.
**דוגמה**: אם $S$ היא קבוצה בלתי תלויה של 5 צמתים אז $Independence(S) = 5$.
אם $S$ היא קליקה של 4 צמתים, אז $Independence(S) = -2$.
נגדיר את השפה $IND$:

$$IND = \{ < G,k > | G = (V,E) \text{ is an undirected graph}, \exists S \subseteq V(Independenc(S) = k) \}$$

**הוכיחו**: $IND$ היא $NP$-שלמה.

??

5. **תזכורת**: **מאמת** (verifier) לשפה $A$ הוא אלגוריתם $V$ כך ש-
$A = \{w | \exists c(V \text{ accepts } < w,c >)\}$.

הוכיחו: לשפה $L$ **יש מאמת** אם ורק אם $L$ היא **מזוהה טיורינג**.
(שימו לב: יש להוכיח שני כיוונים; המאמת לא מוגבל בזמן הריצה שלו).

**פתרון:**

כיוון ראשון: נניח של-$L$ יש מאמת. נבנה מכונת טיורינג לא דטרמינסטית $N$ לזיהוי $A$.
קלט: $w$, מילה.

1. נחש מסמך אישור $c$ (באורך לא מוגבל).
2. הרץ את המאמת ל-$A$ על $< w,c >$. קבל אם $V$ קיבל, ודחה אם דחה.

ברור שאם יש מסמך אישור ל-$w$, בענף כלשהו שלב 1 ינחש אותו, ואז שלב 2 יקבל אותו. אחרת, אף ניחוש לא יצליח – ייתכן שחלק מהם ייכנסו ללופ (של האלגוריתם $V$), וייתכן שעבור חלקם $V$ ידחה. בכל מקרה המכונה תדחה ולא יהיה ענף מקבל.

לכן $M$ מזהה את $A$, אך על פי משפט קיימת גם מכונת טיורינג דטרמיניסטית שתזהה את $A$, ובכך הוכחנו שהיא מזוהה-טיורינג.

כעת, נניח ש-$A$ מזוהה טיורינג. אז קיימת מכונת טיורינג דטרמיניסטית $M$ המזהה את $A$. אלגוריתם $V$ יריץ בעצם את המכונה $A$ ללא תלות במסמך האישור. כך, אם $w \in A$, כל מסמך אישור $c$ יגרום לכך ש-$V$ יריץ את $M$ על $w$ מה שיגרום לקבלה. אחרת, כל מסמך אישור יגרום לריצה של $M$ על $w$, מה שיגרור כניסה ללופ ואי קבלה על ידי המאמת, או לחלופין דחייה ולכן דחייה על ידי המאמת. מש"ל.

6. הוכיחו שהשפה הבאה איננה מזוהה-טיורינג:

$$SHORTEST_{TM} = \{ < M,n > | M \text{ is a } TM, \forall w \in L(M)(|w| \geq n) \wedge \exists w \in L(M)(|w| = n)\}$$

**פתרון:**

נניח שקבענו את $n$ להיות קבוע. השפה כריעה על פי משפט רייס. ראשית, ברור שזוהי תכונה לא טריוויאלית שכן עבור מכונה $M$ שתכריע את השפה $\{a^n\}$ (שהיא רגולרית), $< M,n >$ יהיה בשפה אבל מכונה $M'$ שמכריעה את $\{a^{n+1}\}$ תקיים ש-$< M',n >$ לא בשפה.

היות שהשפה אינה מוגדרת כלל על המכונות עצמן אלא מערבת רק את $L(M)$ נקבל שמתקיים גם התנאי השני של משפט רייס. לכן עבור $n$ קבוע השפה איננה כריעה, ולכן לא ייתכן שהיא כריעה, שכן אז ניתן היה להכריע בפרט עבור $n$ קבוע.

7. נתונה שפה $NL$-שלמה $A$. בנוסף קיימת שפה $B \in L$ כך ש-$A \subset B$. הוכח ש-$A - B$ היא שפה $NL$-שלמה.

**פתרון**:

תחילה, השפה ב-$NL$ שכן ניתן להפעיל את מכונת ה-$NL$ של $A$ על המילה, ואז להפעיל מכונת $L$ עליה ולבדוק שהיא איננה ב-$B$ כל זאת במקום לוגריתמי (לא דטרמיניסטי).

כעת, נראה ש-$A \leq_L A - B$. נניח שקיימת מילה $x \in A - B$ והיא ידועה למכונה המממשת את הרדוקציה.

קלט: $w$, מילה.

1. בדוק האם $w \in B$ בעזרת מכונת $L$ ל-$B$.
2. אם כן, הוצא לפלט את $x$.
3. אחרת, הוצא לפלט את $w$.

**נכונות**: נניח ש-$w \in A$. ייתכנו שני מקרים: $w \in B$ או $w \notin B$. אם $w \in B$, בשלבים 1-2 מוציאים לפלט את $x$ שמקיים $x \in A - B$ כדרוש. אחרת, $w \notin B$ ואז מוציאים לפלט את $w$ שמקיים $w \in A$ ו-$w \notin B$ כלומר $w \in A - B$, כדרוש.
כעת, נניח ש-$w \notin A$. לא ייתכן ש-$w \in B$ כי $B \subseteq A$ ולכן $w \in A$. לכן הרדוקציה תחזיר את $w$ שמקיים $w \notin A$ ובפרט $w \notin A - B$, כדרוש.

**דרישת מקום**: היות ש-$B \in L$ קיימת לו מכונה דטרמיניסטית לוגריתמית, ולכן המתמר המחשב את הרדוקציה יזדקק למקום לוגריתמי בלבד. לכן סיבוכיות המקום של הרדוקציה היא לוגריתמית כדרוש.

8. נגדיר את השפה $UNION-ALL_{TM}$:
$UNION-ALL_{TM} = \{<M_1, M_2> | L(M_1) \cup L(M_2) = \Sigma^*\}$.

א. הציגו **רדוקציית מיפוי** של $A_{TM}$ ל-$UNION-ALL_{TM}$ (הראו: $A_{TM} \le_m UNION-ALL_{TM}$).

ב. הציגו **רדוקציית מיפוי** של $ALL_{TM}$ ל-$UNION-ALL_{TM}$ (הראו: $ALL_{TM} \le_m UNION-ALL_{TM}$).

תארו כל אחד מהרדוקציות, הראו שהיא ניתנת לחישוב, והוכיחו שהיא תקפה.

**פתרון:**

א. קלט: $<M, w>$ כאשר $M$ מכונת טיורינג ו-$w$ מילה.

1. בנה את המכונות $M_1, M_2$ הבאות:
2. $M_1$="קלט: מילה $x$.
   1. דחה."
3. $M_2$="קלט: מילה $x$.
   1. הרץ את $M$ על $w$.
   2. אם $M$ קיבלה, קבל את $x$. אם דחתה – דחה."
4. החזר את $<M_1, M_2>$".

ברור שהיא ניתנת לחישוב שכן כל שנעשה כאן הוא בניית שתי מכונות בסיסיות.
כעת נניח ש-$w \in L(M)$. בכל מקרה $L(M_1) = \varnothing$. כעת מכיוון ש-$w \in L(M)$ שלב 1 יקבל אותה, ואז $M_2$ תקבל כל קלט, כלומר $L(M_2) = \Sigma^*$. סה"כ קיבלנו ש-$L(M_1) \cup L(M_2) = \Sigma^*$, ולכן $<M_1, M_2> \in UNION-ALL_{TM}$.
נניח ש-$w \notin L(M)$. אז $M_2$ לא תקבל אף מילה, ולכן גם $L(M_2) = \varnothing$. בפרט לא יתקיים שאיחוד השפות הוא $\Sigma^*$ ולכן $<M_1, M_2> \notin UNION-ALL_{TM}$, כדרוש.

ב. קלט: $<M>$ כאשר $M$ היא מכונת טיורינג.

1. הוצא לפלט $<M, M'>$ כאשר $M'$ מכריעה את השפה הריקה.

ברור שהרדוקציה הזו חשיבה. כעת, נניח ש- $<M>\in ALL_{TM}$. אז $L(M)=\Sigma^*$ ואז גם $L(M)\cup L(M')=\Sigma^*$ כלומר $<M,M'>\in UNION-ALL_{TM}$, כדרוש.

אם $L(M)\neq\Sigma^*$, אז גם $L(M)\cup L(M')=L(M)\cup\varnothing=L(M)\neq\Sigma^*$, ולכן $<M,M'>\notin UNION-ALL_{TM}$, כדרוש.

9. נגדיר את השפה $ALL-EXCEPT-ONE_{TM}$ להיות שפת כל התיאורים של מכונות טיורינג $<M>$ שמקבלות את כל המילים באלפבית שלהן פרט למילה אחת.
הראה רדוקציית מיפוי מ- $ALL_{TM}$ ל- $ALL-EXCEPT-ONE_{TM}$.

**פתרון:**

קלט: $<M>$ כאשר $M$ היא מכונת טיורינג.

1. בנה את המכונה $M'$ הבאה:
   קלט: מילה $x$.
   1. הרץ את $M$ על $\varepsilon$.
   2. אם $M$ קיבלה,
      2.1. הרץ את $M$ על $x$. אם $M$ קיבלה קבל אם $x\neq\varepsilon$.
2. החזר את $M'$.

**נכונות:**

נניח ש- $<M>\in ALL_{TM}$. אז בפרט $\varepsilon\in L(M)$ ולכן בשלב 1 המכונה $M$ תקבל, ואז תריץ את $M$ על $x$ ושוב תקבל כי $M$ מקבלת את כל המילים, ולכן $M'$ תקבל את כל המילים פרט למילה $\varepsilon$. בפרט קיימת רק מילה אחת שהיא לא מקבלת ולכן $<M'>\in ALL-EXCEPT-ONE_{TM}$, כדרוש.

כעת, נניח ש- $<M>\notin ALL_{TM}$. ייתכנו שני מקרים: $\varepsilon\notin L(M)$. ואז $M'$ תדחה כל קלט או לא תעצור עליו, ולכן $L(M')=\varnothing$ ובפרט היא לא ב- $ALL-EXCEPT-ONE_{TM}$.

אם כן מתקיים $\varepsilon\in L(M)$ אז ידוע ש- $\varepsilon$ תידחה בשלב 2 על ידי $M'$, אך בנוסף חייבת להיות עוד מילה $w\notin L(M)$, וזו תידחה על ידי $M$ או ש- $M$ תיכנס ללופ עליה, ובפרט $w\notin L(M')$. לכן $M'$ לא תקבל לפחות שתי מילים ולכן $<M'>\notin ALL-EXCEPT-ONE_{TM}$.

10. הוכיחו: שפה $A$ היא מזוהה טיורינג אם ורק אם יש מונה שמפיק את $A$ וכל מילה ב- $A$ מודפסת על ידי המונה **פעם אחת ויחידה**. (כלומר, מילה ששייכת ל- $A$ מודפסת פעם אחת; מילה שלא שייכת ל- $A$ לא מודפסת אף פעם).
(ההבדל בין משפט 3.21 למה שאתם צריכים להוכיח בשאלה הוא הדרישה שכל מילה בשפה תודפס רק פעם אחת).

**פתרון:**

נשפר את המכונה שבהוכחת משפט 3.21. במקום להדפיס כל מילה אם היא מתקבלת, נבדוק מהו מספר הצעדים הנדרשו לחישובה. אם הוא שווה בדיוק ל- $i$ (משתנה הלולאה), נדפיס. אחרת, נמשיך כרגיל. במקרה זה נמשיך להדפיס את כל מילות השפה שכן מילה $w$ שמספר הצעדים לחישובה הוא $i$ תודפס בלולאה ה- $i$. עם זאת, כל מילה תודפס לכל היותר פעם אחת שכן הוא מודפס רק כאשר $i$ הוא מספר הצעדים שלו.

11. הציגו רדוקציה של $HALT_{TM}$ ל- $E_{TM}$.

**פתרון:**

נניח שקיימת מכונה $D$ המכריעה את $E_{TM}$, ונבנה באמצעותה מכונה להכרעת $HALT_{TM}$.

קלט: $<M,w>$ כאשר $M$ היא מכונת טיורינג ו-$w$ היא מילה.
1. בנה את המכונה $M'$ שלהלן:
קלט: $x$, מילה.
   1. אם $x \neq w$ דחה.
   2. הרץ את $M$ על $w$. אם $M$ עצרה – קבל.
2. הרץ את $D$ על $<M'>$. אם $D$ דחתה – קבל. אחרת – דחה.

**נכונות:** נניח ש-$<M,w> \in HALT_{TM}$. אז $M'$ תקבל את $w$ ולכן $L(M') \neq \emptyset$, כלומר $D$ תדחה את $<M'>$ ולכן המכונה שלנו תקבל, כדרוש.
אם $<M,w> \notin HALT_{TM}$, אז $M'$ תיכנס ללולאה אינסופית ובפרט $L(M') = \emptyset$, לכן $D$ תקבל אותה, והמכונה שלנו תדחה, כדרוש.

12. נגדיר את השפה $SHORTEST-PATH$:

$$SHORTEST-PATH = \{<G,s,t,k>| \text{the shortest path from } s \text{ to } t \text{ in } G \text{ has length exactly } k\}$$

(אורך של מסלול מוגדר כמספר הקשתות בו.)

א. **הוכיחו:** $SHORTEST-PATH \in NL$.
בתשובתכם אתם רשאים להשתמש בשוויון $NL = coNL$.

ב. הראו **רדוקציית מקום לוגריתמי** של המשלימה של $PATH$ ל-$SHORTEST-PATH$ (הראו: $\overline{PATH} \leq_L SHORTEST-PATH$.

ג. הסיקו: $\overline{SHORTEST-PATH}$ היא $NL$-שלמה.

**פתרון:**

1. נגדיר שתי שפות: $SP_1$ כשפת הרביעיות $<G,s,t,k>$ כאשר ב-$G$ יש מסלול באורך לכל היותר $k$ מ-$s$ ל-$t$, ו-$SP_2$ כשפת הרביעיות $<G,s,t,k>$ כך שלא קיים ב-$G$ מסלול מ-$s$ ל-$t$ באורך לכל היותר $k-1$. מתקיים בבירור $SHORTEST-PATH = SP_1 \cap SP_2$ שכן היא מכילה את כל הגרפים שבהם יש מסלול באורך $k$ אך אין מסלול באורך קטן מ-$k$.
אך $SP_1$ נמצאת ב-$NL$ בקלות שכן ניתן לנחש מסלול באורך לכל היותר $k$ (מניית מספר הצמתים היא $\log(k)$ מקום, שהוא $\log(n)$). בדומה, $SP_2 \in coNL = NL$ שכן ניתן לנחש מסלול באורך לכל היותר $k-1$, ואז מקבלים ש-$SP_1, SP_2 \in NL$, אך $NL$ סגורה לחיתוך ולכן $SP_1 \cap SP_2 = SHORTEST-PATH \in NL$. מש"ל.

2. קלט: $<G,s,t>$ כאשר $G$ הוא גרף מכוון ו-$s$ ו-$t$ הם צמתים.
   1. הוסף $|V|-1$ צמתים חדשים וקשר כל אחד לשני שכנים פרט לשניים מהצמתים, כך שיווצר "שרוך".
   2. צור קשת מקצה אחד של השרוך ל-$s$, וקשת מהקצה השני ל-$t$.
   3. הוצא לפלט $<G',s,t,|V|>$.

**נכונות:** נניח ש-$<G,s,t> \in PATH$. אז קיים מסלול בין $s$ ל-$t$, ולכן המסלול הקצר ביותר יהיה באורך הקטן מ-$|V|$, שכן אחרת היו מעגלים וניתן היה להסיר אותם. בגרף החדש רק יתווספו מסלולים, ולכן אורך המסלול הקצר ביותר יהיה קטן מ-$|V|$, כלומר $<G',s,t,|V|> \notin SHORTEST-PATH$.

נניח ש- $<G,s,t> \notin PATH$. אז בגרף החדש המסלול היחיד בין הצמתים יהיה זה החדש, אך הוא בעל $|V|$ קשתות, ולכן המסלול הקצר ביותר בין הצמתים יהיה בעל $|V|$ קשתות, כלומר $<G',s,t,|V|> \in SHORTEST-PATH$, כדרוש.

3. אנו יודעים שאם $\overline{PATH} \le_L SHORTEST-PATH$ אז גם $PATH \le_L \overline{SHORTEST-PATH}$. אך $SHORTEST-PATH \in NL$ ומהשוויון $NL = coNL$ נקבל ש- $\overline{SHORTEST-PATH} \in NL$. בנוסף, $PATH$ היא $NL$-שלמה, ולכן נקבל ש- $\overline{SHORTEST-PATH}$ היא $NL$-שלמה. מש"ל.

13. נתונים שני מונים $E_1$ ו- $E_2$.
נסמן ב- $L(E_1)$ את השפה ש- $E_1$ מפיק, וב- $L(E_2)$ את זו של $E_2$.

4. **הסבירו היטב** כיצד אפשר לבנות מונה $E_\cup$ שמפיק את השפה $L(E_1) \cup L(E_2)$.
הכוונה היא לבניית המונה $E_\cup$ מן המונים $E_1$ ו- $E_2$ בלי לעבור דרך מכונת טיורינג.
אתם רשאים להניח של- $E_\cup$ יש מספר סרטי עבודה.

5. **הסבירו היטב** כיצד אפשר לבנות מונה $E_\cap$ שמפיק את השפה $L(E_1) \cap L(E_2)$. שוב, ניתן להניח שלמונה יש כמה סרטי עבודה.

**פתרון:**

א. נבנה מונה $E_\cup$ מתאים.

1. כל עוד אחד מהמונים $E_1$ ו- $E_2$ לא סיים את פעולתו:

1.1. הרץ את $E_1$ (אם לא סיים פעולתו), וכשידפיס את המילה הבאה, המשך ל-1.2.

1.2. הרץ את $E_2$ (אם לא סיים פעולתו) וכשידפיס את המילה הבאה, חזור ל-1.

**נכונות:** תהי $w \in L(E_1) \cup L(E_2)$. ב.ה.כ. $w \in L(E_1)$ ולכן בסופו של דבר היא תודפס בשלב 1.1. בנוסף, כל מילה שמודפסת שייכת לאחת מהשפות. כך קיבלנו שהמונה אכן יוצר את האיחוד.

ב. נבנה מונה $E_\cup$ מתאים.

1. כל עוד אחד מהמונים $E_1$ ו- $E_2$ לא סיים את פעולתו:

1.1. הרץ את $E_1$ (אם לא סיים את פעולתו), ובמקום להדפיס את המילה הבאה – כתוב אותה על סרט עבודה נפרד בשביל המונה הנ"ל. בדוק האם המילה נמצאת בסרט המילים של המונה השני. אם כן – הדפס אותה. המשך ל-1.2.

1.2. הרץ את $E_2$ (אם לא סיים את פעולתו), ובמקום להדפיס את המילה הבאה – כתוב אותה על סרט עבודה נוסף בשביל המונה הנ"ל. בדוק האם המילה נמצאת בסרט המילים של המונה הראשון. אם כן – הדפס אותה. חזור ל-1.

**נכונות:**

אם $w \in L(E_1) \cap L(E_2)$ אז בפרט $w \in L(E_1)$ ולכן היא תודפס מתישהו על ידי $E_1$. כשהיא תודפס על ידו, ייתכנו שתי אפשרויות: אם היא כבר הודפסה על ידי $E_2$, ההשוואה תצליח והמילה תודפס. אם היא עוד לא הודפסה על ידי $E_2$, כאשר היא תודפס על ידה – $w$ כבר תופיע ברשימת המילים שהדפיסה $E_1$, ההשוואה תצליח, ושוב המילה תודפס. מש"ל.

14. במשפט 5.10 הוכח שהשפה $E_{LBA}$ איננה כריעה.

1. האם $E_{LBA}$ היא שפה **מזוהה-טיורינג**? הוכיחו את תשובתכם.

2. האם השפה ה**משלימה** היא שפה **מזוהה-טיורינג**? הוכיחו את תשובתכם.

**פתרון:**

א. נוכיח תחילה ש- $\overline{E_{LBA}}$ היא מזוהה טיורינג.

קלט: $<M>$, כאשר $M$ הוא אוטומט חסום לינארית.

1. עבור כל מילה בסדר הסטנדרטי: $c_1, c_2, ..., c_n, ...$:

1.1. הרץ את המכונה להכרעת $A_{LBA}$ מהוכחת משפט 5.9 על $<M, c_i>$. אם היא קיבלה – קבל. אחרת – המשך.

**נכונות:** אם $<M> \in \overline{E_{LBA}}$ אז קיימת $c_i \in L(M)$ וזו תימצא בשלב 1.1, ולכן המכונה תקבל. בכיוון השני – אם נמצאה מילה כזו אז בהכרח $L(M) \neq \varnothing$ ולכן יש לקבל.

כעת, נניח בשלילה ש- $E_{LBA}$ מזוהה-טיורינג. על פי משפט 4.22 נקבל שהיא גם כריעה. וזה בסתירה למשפט 5.10.

ב. הוכח בסעיף א'.

15. **מעגל המילטון** בגרף לא מכוון $G = (V, E)$ הוא מעגל פשוט שמכיל כל צומת של הגרף פעם אחת ויחידה.

השפה $UHAMCIRCUIT$ מוגדרת להיות שפת הקידודים של גרפים לא מכוונים שיש בהם מעגל המילטון.

1. הראו רדוקציה בזמן פולינומי של $UHAMPATH$ ל- $UHAMCIRCUIT$.
2. הוכיחו: $UHAMCIRCUIT$ היא $NP$-שלמה.

**פתרון:**

א. קלט: $<G, s, t>$ כאשר $G$ הוא גרף לא מכוון ו- $s$ ו- $t$ הם צמתים שונים.

1. צור צומת חדש $st$ וקשת בינו לבין $s$ ובינו לבין $t$.
2. החזר את הגרף החדש.

**הוכחת נכונות:**

נניח ש- $<G, s, t> \in UHAMPATH$. אז יש מסלול המילטון, $P = su_1u_2...u_nt$, בגרף הישן.

בגרף החדש יהיה קיים המסלול $(st)su_1...u_nt(st)$ והוא מעגל המילטון. כלומר $<G'> \in UHAMCIRCUIT$.

כעת, נניח ש- $<G'> \in UHAMCIRCUIT$. אז קיים בגרף החדש מעגל המילטון. היות שבמעגל המילטון כל צומת יכול להיות צומת התחלתי באופן סימטרי, והמעגל חייב לעבור בצומת $st$, נניח שהצומת ההתחלתי הוא $st$. הואיל והמעגל חייב לעבור בכל צומת פעם אחת בדיוק, נקבל שהוא משתמש בשתי הקשתות החדשות: $P' = (st)su_1...u_nt(st)$ (בלי הגבלת הכלליות הצומת השני הוא $s$ ולא $t$). ממסלול זה ניתן להוריד את הצומת $(st)$ ולקבל בגרף המקורי את המסלול $su_1u_2...u_nt$, שחייב להיות מסלול המילטון מ- $s$ ל- $t$ בגרף המקורי. מש"ל.

**ניתוח סיבוכיות:** כל שנדרש הוא הוספת צומת ושתי קשתות, ולכן זה רץ בבירור בזמן פולינומי.

ב. נוכיח ש- $UHAMCIRCUIT \in NP$: נבנה מאמת. המאמת יקבל כמסמך אישור מסלול.

קלט: $<G,c>$, כאשר $G$ הוא גרף לא מכוון ו-$c$ הוא מסלול.

1. בדוק שהצומת הראשון והאחרון של $c$ זהים.
2. בדוק שכל הצמתים במסלול שייכים ל-$G$, ושהם שונים זה מזה.
3. בדוק שכל צומת ב-$G$ מופיע ב-$c$.
4. אם כל אלו מתקיימים – קבל.

המאמת יקבל קלט אם הוא מהווה מעגל המילטוני בגרף.
כל שלב בנפרד רץ בזמן ריבועי לכל היותר, ולכן בזמן פולינומי, כדרוש.

כעת, $UHAMPATH$ היא $NP$ שלמה, והיות שהוכחנו ש-$UHAMPATH \leq_P UHAMCIRCUIT$ וש-$UHAMCIRCUIT \in NP$, נקבל ש-$UHAMCIRCUIT$ היא $NP$-שלמה. מש"ל.

16. הבעיה $E_{DFA}$ הוגדרה בספר לפני משפט 4.4.

הוכיחו: $\overline{E_{DFA}}$ היא שפה $NL$-שלמה.

**הדרכה:** הראו שהיא שייכת ל-$NL$ והראו כי $PATH \leq_L \overline{E_{DFA}}$.

**פתרון:**

תחילה נראה ש-$\overline{E_{DFA}} \in NL$. נכתוב אלגוריתם לא דטרמיניסטי להכרעת השפה:

קלט: $<D>$ כאשר $D$ הוא אוטומט סופי דטרמיניסטי.

1. הרץ את האלגוריתם (הלא דטרמיניסטי) להכרעת $PATH$ החל מהמצב ההתחלתי ועבור כל מצב מקבל של האוטומט. אם מישהו מהם קיבל – קבל.

**נכונות:** נניח ש-$<D>\in\overline{E_{DFA}}$. אז בהכרח קיים מסלול מהמצב ההתחלתי למצב מקבל כלשהו. אז מסלול חישוב אחד לפחות במכונה להכרעת $PATH$ יכריע שקיים מסלול בין המצב ההתחלתי לבין המצב המקבל הנ"ל, ומסלול חישוב זה יתקבל גם באחד ממסלולי החישוב של המכונה שלנו, ולכן $<D>$ יתקבל.

כעת, אם אחד מהאלגוריתמים להכרעת $PATH$ יחזיר אמת, אזי קיים מסלול מהמצב ההתחלתי למצב מקבל כלשהו. אם ניקח את כל האותיות המופיעות על המסלול הזה ונרכיב ממנו מילה, המילה הזו חייבת להיות בשפה ולכן $<D>\in\overline{E_{DFA}}$.

**סיבוכיות:** ניתן להריץ את כל אחד מהאלגוריתמים הלא דטרמיניסטיים על אותו מקום, וכל אחד מהם לוקח מקום לוגריתמי. לכן בסה"כ האלגוריתם ידרוש מקום לוגריתמי.

כעת נראה רדוקציה לוגריתמית במקום: $PATH \leq_L \overline{E_{DFA}}$.

קלט: $<G,s,t>$ כאשר $G$ הוא גרף מכוון ו-$s$ ו-$t$ צמתים. נניח שקבוצת הצמתים היא $\{v_1,v_2,...,v_m\}$.

1. תחילה נספור את מספר הצמתים ונאחסן אותו במונה בסרט העבודה. נאמר שהוא שווה $m$.
2. נוציא לפלט אוטומט סופי דטרמיניסטי כדלהלן:
3. $Q=V\cup\{q^*\}$.
4. $\Sigma=\{1,2,...,m\}$.
5. $F=\{t\}$.
6. $q_0=s$.
7. פונקציית המעברים תוגדר כך:

לכל קשת $(v_i,v_j)\in E$ , $\delta(v_i,j)=v_j$ .

לכל קשת $(v_i,v_j)\notin E$ , $\delta(v_i,j)=q^*$ .

$\delta(q^*,\sigma)=q^*$ לכל $\sigma\in Q$ .

**הוכחת נכונות:**

ברור שהאוטומט שנוצר הוא אוטומט סופי דטרמיניסטי.

נניח ש- $<G,s,t>\in PATH$ . אז קיים מסלול $P=su_{i_2}u_{i_3}...u_{i_{n-1}}t=u_{i_1}...u_{i_n}$ . נתבונן במילה $w=i_2i_3...i_n$ . היות שקיימות הקשתות $(u_{i_l},u_{i_{l+1}})$ נקבל שבפונקציית המעברים יש את המעבר $\delta(u_{i_l},l+1)=u_{i_{l+1}}$ . לכן בפרט בפונקציית המעברים $w$ מתחילה במצב התחלתי ועוברת, רק על ידי מעברים חוקיים, למצב מקבל. לכן $w\in L(D)$ עבור האוטומט $D$ שבפלט, ולכן $<D>\in \overline{E_{DFA}}$ , כדרוש.

כעת, נניח ש- $<D>\in \overline{E_{DFA}}$ . אז קיים מסלול מהמצב ההתחלתי למצב המקבל, כלומר קיים מסלול מ- $s$ ל- $t$ . המסלול לא עובר ב- $q^*$ שכן זהו מצב מלכודת. לכן כל הקשתות שקיימות באוטומט מתאימות לקשתות בגרף המקורי, ולכן קיים מסלול מ- $s$ ל- $t$ בגרף המקורי. מש״ל.

**ניתוח סיבוכיות:** כל שיש לעשות הוא לעבור על כל צומת ולבדוק אילו קשתות יש לו לצמתים אחרים ואילו אין, ולפי זה להוציא לפלט. לא משתמשים בזיכרון (אולי רק לשמירת אינדקסים של מקומות על סרט הקלט, וזה מקום לוגריתמי), ולכן סיבוכיות המקום היא לוגריתמית, כדרוש.

הוכחנו ש- $PATH\le_L \overline{E_{DFA}}$ , והיות ש- $PATH$ היא $NL$ -שלמה, נקבל ש- $\overline{E_{DFA}}$ היא $NL$ - קשה. אך היא גם ב- $NL$ ולכן היא $NL$ -שלמה.

17. כזכור, מכונת טיורינג יכולה בריצתה על מילת קלט $w$ לסיים במצב המקבל $q_{accept}$ , לסיים במצב הדוחה, $q_{reject}$ , או לא לעצור.

המוסכמה המקובלת היא שאם מכונה $M$ לא עוצרת על מילה $w$ אז $M$ **דוחה** את $w$ .

נניח שמסכימים שאם מכונה $M$ לא עוצרת על מילה $w$ אז $M$ **מקבלת** את $w$ .

שפה $L$ תיקרא **מזוהה-טיורינג במוסכמה החדשה** אם יש מכונת טיורינג שמקבלת את המילים של $L$ (ורק אותן) לפי המוסכמה החדשה.

לכל אחת מהטענות הבאות, קבעו האם היא נכונה, והוכיחו את קביעתכם.

1. אם $L$ היא שפה מזוהה-טיורינג במוסכמה המקובלת, אז $L$ מזוהה-טיורינג גם במוסכמה החדשה.
2. אם $L$ היא שפה מזוהה-טיורינג במוסכמה החדשה, אז $L$ מזוהה-טיורינג גם במוסכמה המקובלת.

**פתרון:**

א. ראשית נשים לב שמשפט 4.22 עדיין תקף, שכן מילה בשפה חייבת להתקבל על ידי $M_1$ או להידחות על ידי $M_2$ , ואחרת היא לא שייכת לשפה וגם לא שייכת למשלימתה. כעת, נשים לב שהשפה $A_{TM}$ היא מזוהה טיורינג במוסכמה החדשה, כאשר מתייחסים לקידודים של מכונות טיורינג בצורה החדשה:

עבור קלט: $<M,w>$

1. הרץ את $M$ על $w$ .
2. אם $M$ קיבלה – קבל. אם היא דחתה – דחה.

נשים לב שאם $w\in L(M)$ אזי $M$ תקבל או לא תעצור, ובשני המקרים המכונה שלנו "תקבל" כדרוש. כעת, אם $w\notin L(M)$ אז $M$ בהכרח עוצרת ודוחה, ולכן המכונה שלנו תדחה כדרוש.

כעת, נשים לב כי הוכחת משפט 4.11 לא מסתמכת על כך שהמכונה דוחה דווקא את הקלט שמביא ללולאה, ולכן נקבל ש- $A_{TM}$ איננה כריעה.

18. נאמר שמכונת טיורינג דטרמיניסטית $M$ **מבצעת לולאה** במהלך ריצתה על מילת קלט $w$ אם במהלך הריצה של $M$ על $w$ יש צעד חישוב שבו $M$ חוזרת למצב שבו היא נמצאת.
האם **לכל** מכונת טיורינג נתונה $M$, יש מכונת טיורינג דטרמיניסטית $M'$ שמקיימת את שני התנאים הבאים:
- $M'$ מזהה את אותה השפה ש- $M$ מזהה.
- **לכל** מילת קלט $w$ (מעל האלפבית של $M$ ו- $M'$) $M'$ לא מבצעת לולאה בריצתה על $w$.

**הוכיחו** את תשובתכם.

**פתרון:**
קיימת מכונת טיורינג כזו.
נניח ש- $M=(Q,\Sigma,\Gamma,\delta,q_0,q_{accept},q_{reject})$.
ניצור מכונת טיורינג $M'=(Q\cup Q',\Sigma,\Gamma,\delta',q_0,q_{accept},q_{reject})$, כאשר אם $Q=\{u_1,u_2,...,u_n\}$ אז $Q'=\{u_1',u_2',...,u_n'\}$.
$\delta'$ תוגדר כך: לכל $q_1\neq q_2$, $q_1,q_2\in Q$, ולכל $\sigma_1,\sigma_2\in\Gamma$,
אם $\delta(q_1,\sigma_1)=(q_2,\sigma_2,R)$ אז $\delta(q_1,\sigma_1)=(q_2,\sigma_2,R)$.
אם $\delta(q_1,\sigma_1)=(q_2,\sigma_2,L)$ אז $\delta(q_1,\sigma_1)=(q_2,\sigma_2,L)$.

ולכל $q\in Q$ ו- $\sigma_1,\sigma_2\in\Gamma$,
אם $\delta(q,\sigma_1)=(q,\sigma_2,R)$ אז $\delta'(q,\sigma_1)=(q',\sigma_2,R)$ ו- $\delta'(q',\sigma_1)=(q,\sigma_2,R)$.
אם $\delta(q,\sigma_1)=(q,\sigma_2,L)$ אז $\delta'(q,\sigma_1)=(q',\sigma_2,L)$ ו- $\delta'(q',\sigma_1)=(q,\sigma_2,L)$.
פרט למעברים האלו, פונקציית המעברים עבור $q$ זהה ל- $\delta$, ועבור $q'$ זהה ל- $\delta$ רק בהחלפת $q$ ב- $q'$.
**הוכחה:** ברור מתיאור המכונה שהיא מתנהגת בדיוק כמו $M$, רק שבמקום לעשות לולאה, היא מתנדנדת בין המצבים השקולים – והם $q$ ו- $q'$. כך היא לא עושה לולאה אבל בעצם מחשבת את אותו חישוב. לכן הן מזהות את אותה שפה, וברור שבמכונה החדשה אין לולאות.

19. **תזכורת:** מכונת טיורינג נקראת **מכונה מכריעה** (decider) אם היא **עוצרת** על כל מילת קלט $w$ (במצב המקבל או במצב הדוחה).
נגדיר את השפה $DECIDER_{TM}$ להיות שפת התיאורים של מכונות טיורינג $<M>$, המכריעות את השפה שהן מזהות.
הראו **רדוקציית מיפוי** של $DECIDER_{TM}$ ל- $ALL_{TM}$ ($DECIDER_{TM}\leq_m ALL_{TM}$).
הציגו את הרדוקציה והראו שהיא תקפה וניתנת לחישוב.

**פתרון:**
קלט: $<M>$, כאשר $M$ היא מכונת טיורינג.
1. צור את המכונה הבאה:

$M'$ = "קלט: מילה $x$.
1. הרץ את $M$ על $x$. אם $M$ עצרה, קבל."
2. החזר את $M'$.

**נכונות:**

נניח ש- $<M>\in DECIDER_{TM}$. אז לכל $w\in\Sigma^*$, $M$ עוצרת על $w$. אז $M'$ תקבל את $w$. לכן $L(M')=\Sigma^*$ ולכן $<M'>\in ALL_{TM}$ כדרוש.

כעת נניח ש- $<M>\notin DECIDER_{TM}$. אז קיימת מילה $w\in\Sigma^*$ כך ש- $M$ לא עוצרת על $w$. אז $w\notin L(M')$ ולכן $<M'>\notin ALL_{TM}$.

20. **תזכורת:** השפה $VERTEX-COVER$ היא $NP$-שלמה.

**הוכיחו:** גם הבעיה $2VC$ שלהלן היא $NP$-שלמה. השפה $2VC$ מכילה את כל הזוגות $<G,k>$ כאשר $G$ הוא גרף לא מכוון ויש ב- $G$ שתי קבוצות קודקודים (שונות אך לא בהכרח זרות), כל אחת בגודל $k$, שכל אחת מהן היא כיסוי בקודקודים בגודל $k$ של קשתות $G$.

**פתרון:**

תחילה, ברור ש- $2VC\in NP$ כי מאמת יכול לקבל שתי קבוצות צמתים בגודל $k$.

קלט: $<G,k,A_1,A_2>$, כאשר $G$ גרף לא מכוון, $k$ מספר ו- $A_1,A_2$ הן קבוצות צמתים בגודל $k$.
1. בדוק ש- $A_1\neq A_2$.
2. בדוק שכל קבוצה היא כיסוי בצמתים של הגרף.
3. אם כן, החזר "כן". אחרת החזר "לא".

כעת, נבנה רדוקציה מ- $VERTEX-COVER$ ל- $2VC$.

קלט: $<G,k>$ כאשר $G$ הוא גרף לא מכוון ו- $k$ הוא מספר.
1. צור גרף $G'$ הזהה ל- $G$.
2. הוסף ל-$G'$ שני צמתים וחבר אותם בקשת אחת.
3. החזר $<G',k+1>$.

**הוכחת נכונות:**

נניח ששני הצמתים הנוספים בגרף הם $\{x,y\}$.

נניח ש- $<G,k>\in VERTEX-COVER$. אז קיים כיסוי בצמתים ב- $G$, נאמר $S$. נתבונן בקבוצה $S\cup\{x\}$. תהי קשת $e\in E'$. אם $e$ מחברת בין $x$ ל- $y$ אז הצומת $x$ נוגע בה. אחרת, מכיוון ש- $S$ הוא כיסוי בצמתים והיא קשת גם בגרף הישן, יש צומת ב- $S$ שנוגע בה. לכן $S\cup\{x\}$ הוא כיסוי בצמתים ב-$G'$ בגודל $|S|+1=k+1$.

בנוסף, באופן סימטרי, גם $S\cup\{y\}$ הוא כיסוי בצמתים ב-$G'$. אך $S\cup\{x\}\neq S\cup\{y\}$. בכך הוכחנו ש- $<G',k+1>\in 2VC$.

כעת, נניח ש- $<G',k+1>\in 2VC$. אז קיימים שני כיסויים בצמתים בגודל $k+1$. נבחר אחד מהם, נאמר $S$. אם $S$ לא מכילה גם את $x$ וגם את $y$ (היא חייבת להכיל אחד מהם אחרת לא תכסה את הקשת הזו), אז ניתן להסיר את הצומת הנ"ל ולקבל כיסוי בצמתים של צמתי הגרף $G$ בגודל $k$, כדרוש. אחרת, $\{x,y\}\subseteq S$. ניתן להסיר את שני הצמתים האלו ולקבל כיסוי בצמתים של $G$ בגודל $k-1$. אז אפשר להוסיף לו כל צומת אחר, ולקבל כיסוי בצמתים ב- $G$ בגודל $k$, כדרוש. כלומר בכ"מ קיים ב- $G$ כיסוי בצמתים. מש"ל.

**סיבוכיות זמן:**

כל שנדרש הוא הוספת שני צמתים וקשת ביניהם, וזה נעשה בזמן פולינומי בבירור.

לכן $VERTEX-COVER \le_P 2VC$, ו- $VERTEX-COVER$ היא $NP$-שלמה, ו- $2VC \in NP$, ולכן נקבל ש- $2VC$ היא $NP$-שלמה. מש"ל.

21. תהי $C$ שפה. מילה $w$ נקראת **מילה מקסימלית** בשפה $C$ אם $w \in C$ ו- $w$ **איננה תחילית ממש** של אף מילה ששייכת ל- $C$. (מילה $w$ היא **תחילית** של מילה $v$ אם יש מילה $u$ כך ש- $v = wu$. אם $u \neq \varepsilon$ אז $w$ נקראת תחילית ממש.)

נגדיר את **שפת המילים המקסימליות** של השפה $C$ להיות שפת כל המילים המקסימליות ב- $C$.

דוגמה: אם $C$ היא שפת המילים מעל האלפבית $\{0,1\}$ שמתחילות ברצף לא ריק של אפסים ולאחר מכן בין 0 לשלושה 1-ים, אז $Max(C)$ היא שפת המילים המתחילות ברצף לא ריק של אפסים ולאחריו שלושה 1-ים.

להלן "הוכחה" שאם $C \in NP \cap coNP$ אז $\overline{Max(C)}$ שייכת למחלקה $NP$:

תהי $w \in \overline{Max(C)}$.

אז או ש- $w$ לא שייכת ל- $C$, או שיש מילה $v$ ש- $w$ היא תחילית ממש שלה ו- $v \in C$. את המקרה הראשון אפשר לאמת בזמן פולינומיאלי, משום ש- $C$ שייכת ל- $coNP$ (ולכן המשלימה של $C$ שייכת ל- $NP$). את המקרה השני אפשר לאמת בזמן פולינומיאלי, משום ש- $C$ שייכת ל- $NP$.

מה לא בסדר ב"הוכחה" הזו?

**פתרון:**

יש אינסוף מילים $v$ כאלו, כך שלא נוכל לאמת את זה.

22. תהי $A$ ששייכת למחלקה $L$.

האם בהכרח השפה $A^*$ שייכת למחלקה $SPACE(\log^2 n)$? הוכיחו את תשובתכם.

**פתרון:**

התשובה חיובית. נבנה מכונה לא דטרמיניסטית המכריעה את $A^*$ בזמן לוגריתמי.

קלט: $w$, מילה.

1. צור שני מונים, ששניהם יאותחלו לאפס.
2. בחר באופן לא דטרמיניסטי עד לאיזה מספר (שקטן מ- $n+1$) לקדם את המונה השני. אם הגעת ל- $n+1$, קבל.
3. עבור המילה שתחילתה בתו של המונה הראשון וסופה בתו של המונה השני (כלומר הם מצביעים לקלט), הרץ את המכונה הלוגריתמית להכרעת $A$.

אם המילה התקבלה, קדם את המונה הראשון לערכו של המונה השני ועוד אחת וגם את המונה השני, וחזור ל-2. אחרת – דחה.

**נכונות:**

אם $w \in A^*$ אז קיימת חלוקה של המילה לקבוצת מילים שכל אחת שייכת ל- $A$. אם עוברים על המילה משמאל לימין, כאשר מסיימים את החלק הראשון, הוא יתקבל באחד הענפים על ידי שלבים 2-3, ולאחר מכן ממשיכים לעבור על המילה. סה"כ ענף אחד לבטח יניב חלוקה מתאימה, ולכן המילה תתקבל. אם האלגוריתם מצא חלוקה מתאימה, אז ודאי שהמילה שייכת ל- $A^*$.

**סיבוכיות:**

כל מונה הוא באורך לכל היותר $\log n$, והמכונה להכרעת $A$ רצה במקום $O(\log n)$ ולכן סה"כ סיבוכיות המקום היא $O(\log n)$. כעת על פי ההרחבה של משפט סביץ', היות ש- $A^* \in NSPACE(\log n)$, נקבל ש- $A^* \in SPACE(\log^2 n)$.

23. לכל שפה $D$ נגדיר את השפה $COMP-CONC(D)=\{xy \mid x \in D, y \notin D\}$.
הוכח שאם $D \in BPP$ אז $COMP-CONC(D) \in BPP$.

24. נאמר שיש **רדוקציה בזמן ריבועי** של שפה $A$ לשפה $B$ אם קיימת פונקציה $f$, **חשיבה בזמן ריבועי**, כך שלכל $w$, $w \in A \Leftrightarrow f(w) \in B$.
נגדיר מהי **שפה $P$ -שלמה ביחס לרדוקציה בזמן ריבועי**:
שפה $L$ תיקרא $P$-שלמה ביחס לרדוקציה בזמן ריבועי, אם:
1. $L \in P$.
2. לכל שפה $A \in P$ יש **רדוקציה בזמן ריבועי** ל- $L$.

הוכיחו: **לא קיימת** שפה $P$-שלמה ביחס לרדוקציה בזמן ריבועי.
**רמז:** היעזרו במשפטי היררכיה.

**פתרון:**

נניח בשלילה שקיימת שפה $L$ כזו. היות ש- $L \in P$, $L \in TIME(n^k)$ עבור $k$ כלשהו. כעת, נניח ש- $k > 2$ (אם $k \le 2$ בוודאי היא גם שייכת למחלקות בעלות $k$ גדול יותר). נבחר שפה $A$ הנמצאת ב- $TIME(n^{2k+1})$ אך לא נמצאת ב- $TIME(n^{2k})$, המובטחת על ידי משפטי ההיררכיה לזמן. נבנה מכונת טיורינג ל- $A$ הרצה בזמן $O(n^{2k})$, וכך נקבל סתירה.
קלט: מילה $w$.
1. הפעל את הרדוקציה בזמן ריבועי של $A$ ל- $L$ על $w$, המובטחת משום שהיא $P$-שלמה ביחס לרדוקציה בזמן ריבועי. נאמר שתוצאת הרדוקציה היא $f(w)$.
2. הרץ את המכונה הרצה בזמן $O(n^k)$ על $f(w)$. קבל אם היא מקבלת ודחה אם היא דוחה.

**נכונות:** נכונות האלגוריתם ברורה מנכונות רדוקציית המיפוי.
**סיבוכיות:** היות שהרדוקציה פועלת בזמן ריבועי לכל היותר, $f(w)=O(|w|^2)$. לכן המכונה המכריעה את $L$ תרוץ על קלט בגודל $O(|w|^2)$, ולכן זמן הריצה שלה יהיה $O(|w|^{2^k})=O(|w|^{2k})$. בנוסף המכונה מחשבת עוד זמן ריבועי, אך היות ש- $k>2$, $O(n^2)=O(n^{2k})$ ולכן זמן הריצה של המכונה שלנו הוא $O(n^{2k})$. מש"ל.

25. נעיין במודל החישובי הבא: מכונת טיורינג עם מספר **אינסופי** של מצבים.
מכונה כזו זהה למכונה רגילה, פרט לכך שמספר המצבים יכול להיות אינסופי (ולכן גם התחום והטווח של פונקציית המעברים יכולים להיות אינסופיים).
האם למכונה כזו יש יותר כוח מאשר למכונה רגילה? הוכיחו את תשובתכם.
**פתרון:**
תהי שפה $A$. נוכיח שניתן להכריע באמצעות מכונה כזו את השפה $A$.
המכונה שלנו תלך על כל אות למצב שונה, ואז בעצם כל מצב ייצג מילה מסוימת בשפה. אם כאשר מגיעים לרווח, המילה עד עכשיו נמצאת בשפה, נקבל. אחרת – נדחה.

היות שהמודל הזה יכול להכריע כל שפה (הוכחנו לעיל), נקבל שהוא לא שקול למכונת טיורינג רגילה, שבה הוכחנו שיש שפות שאינן כריעות או מזוהות על ידה.

26. בבעיה 5.18 בספר (עמוד 240) מוגדרת השפה $INFINITE_{TM}$ .

1. הציגו רדוקציית מיפוי של $A_{TM}$ ל- $INFINITE_{TM}$ (הראו: $A_{TM} \leq_m INFINITE_{TM}$).
2. הציגו רדוקציית מיפוי של $A_{TM}$ ל- $\overline{INFINITE_{TM}}$ (הראו: $A_{TM} \leq \overline{INFINITE_{TM}}$).
3. הסיקו: $INFINITE_{TM}$ ו- $\overline{INFINITE_{TM}}$ **אינן מזוהות-טיורינג**.

**פתרון:**

א. קלט: $<M,w>$ כאשר $M$ היא מכונת טיורינג ו- $w$ היא מילה.

1. בנה את המכונה $M'$ הבאה:
קלט: $x$, מילה.
   1. הרץ את $M$ על $w$. אם היא קיבלה – קבל את $x$.
2. החזר את $M'$.

**נכונות:**

ברור שהרדוקציה חשיבה.

אם $<M,w> \in A_{TM}$ אז $M$ מקבלת את $w$, ואז $M'$ תקבל כל מילה, ושפתה תהיה אינסופית. כלומר $<M'> \in INFINITE_{TM}$.

אם $<M,w> \notin A_{TM}$ אז $M$ לא תקבל את $w$ ואז $M'$ לא תקבל אף מילה, כלומר שפתה תהיה ריקה ובפרט לא אינסופית. כלומר $<M'> \notin INFINITE_{TM}$. מש״ל.

ב. קלט: $<M,w>$ כאשר $M$ היא מכונת טיורינג ו- $w$ היא מילה.

1. בנה את המכונה $M'$ הבאה:
קלט: $x$, מילה.
   1. הרץ את $M$ על $w$ $|x|$ צעדים.
   אם $M$ לא עצרה או לחלופין אם היא דחתה במהלך הצעדים האלו, קבל את $x$.
   אחרת – דחה.
2. החזר את $<M'>$.

**נכונות:**

ברור שהרדוקציה חשיבה.

אם $<M,w> \in A_{TM}$, אז $M$ עוצרת על $w$ ומקבלת אותה תוך $c$ צעדים. אז כל מילה באורך הקטן מ- $c$ תתקבל בשלב 1 על ידי $M'$, אך זהו מספר סופי של מילים. כל מילה באורך גדול יותר מ- $c$, או שווה ל- $c$, תידחה שכן $M$ עצרה וקיבלה. לכן $M'$ מקבלת מספר סופי של מילים, ולכן $<M'> \in \overline{INFINITE_{TM}}$.

אם $<M,w> \notin A_{TM}$ אז לכל מספר צעדים $c$, $M$ לא תעצור על $w$ או לחלופין תדחה אותה. לכן כל מילה תתקבל על ידי $M'$ בשלב 1, ולכן שפתה תהיה אינסופית, כלומר $<M'> \notin \overline{INFINITE_{TM}}$. מש״ל.

ג. אנו יודעים שאם $A \leq_m B$ אז גם $\overline{A} \leq_m \overline{B}$ (על ידי אותה רדוקציית מיפוי). לכן נקבל ש- $\overline{A_{TM}} \leq_m INFINITE_{TM}$ ו- $\overline{A_{TM}} \leq_m \overline{INFINITE_{TM}}$. אך $\overline{A_{TM}}$ אינה מזוהה טיורינג ולכן ממשפט 5.29 נקבל ש- $INFINITE_{TM}$ ו- $\overline{INFINITE_{TM}}$ אינן מזוהות טיורינג. מש״ל.

27. תארו אוטומט חסום לינארית ($LBA$) המכריע את השפה $A_{NFA}$ (המוגדרת בספר בעמוד 195).
קלט: $<D,w>$ כאשר $D$ הוא אוטומט סופי לא דטרמיניסטי ו- $w$ היא מילה.

**פתרון:**

רעיון הפתרון הוא לסמן את קבוצת המצבים שבהם אנו נמצאים, ועבור כל אחד מהם לבדוק לאן ניתן להמשיך עם הקלט הנוכחי. בסופו של דבר בודקים האם מצב מקבל מסומן.

28. נאמר ששפה $A$ **ניתנת לרדוקציה בזמן לינארי** לשפה $B$ אם יש פונקציה $f$ **חשיבה בזמן לינארי** כך שלכל $w$, $w \in A \Leftrightarrow f(w) \in B$.

נאמר ששפה $A$ **ניתנת לרדוקציה בזמן ריבועי** לשפה $B$ אם יש פונקציה $f$ **חשיבה בזמן ריבועי** כך שלכל $w$, $w \in A \Leftrightarrow f(w) \in B$.

1. נתון ש- $B$ שייכת ל- $TIME(n)$ וש- $A$ ניתנת לרדוקציה בזמן **לינארי** ל- $B$.
האם אפשר להסיק מנתונים אלה ש- $A$ שייכת ל- $TIME(n)$? הסבירו את תשובתכם.
2. נתון ש- $B$ שייכת ל- $TIME(n^2)$ וש- $A$ ניתנת לרדוקציה בזמן **לינארי** ל- $B$.
האם אפשר להסיק מנתונים אלה ש- $A$ שייכת ל- $TIME(n^2)$? הסבירו את תשובתכם.
3. נתון ש- $B$ שייכת ל- $TIME(n)$ וש- $A$ ניתנת לרדוקציה בזמן **ריבועי** ל- $B$.
האם אפשר להסיק מנתונים אלה ש- $A$ שייכת ל- $TIME(n)$? הסבירו את תשובתכם.
4. נתון ש- $B$ שייכת ל- $TIME(n^2)$ וש- $A$ ניתנת לרדוקציה בזמן **ריבועי** ל- $B$.
האם אפשר להסיק מנתונים אלה ש- $A$ שייכת ל- $TIME(n^2)$? הסבירו את תשובתכם.

**פתרון:**

א. התשובה חיובית. נכתוב אלגוריתם לינארי להכרעת $A$.
קלט: מילה $w$.
1. הרץ את $f$ על $w$.
2. הרץ את המכונה הלינארית להכרעת $B$ על $f(w)$.

נכונות האלגוריתם נובעת מנכונות הרדוקציה. ננתח את זמן הריצה; $f$ רצה בזמן לינארי ולכן הפלט שלה הוא לינארי ביחס לקלט. כלומר אורכו הוא $cn$. בנוסף, המכונה של $B$ רצה בזמן לינארי ולכן היא רצה בזמן $dcn$ על הקלט שקיבלה. אך $dc$ הוא קבוע (שתלוי בקבועי הרדוקציה והמכונה של $B$), ולכן המכונה רצה בזמן לינארי. כלומר $A \in TIME(n)$ כדרוש.

ב. המכונה זהה למכונה בסעיף א. ננתח את זמן הריצה; שוב, הפלט של הרדוקציה הוא בגודל $cn$. כעת, המכונה של $B$ רצה בזמן ריבועי ולכן זמן הריצה שלה על הקלט הנ"ל יהיה $dc^2n^2$. אך $dc^2$ הוא קבוע, ולכן נקבל שזמן הריצה הוא $O(n^2)$, כלומר $A \in TIME(n^2)$ כדרוש.

ג. התשובה שלילית. ננתח את זמן הריצה של האלגוריתם ה"טבעי" שמצופה. הרדוקציה הריבועית מוציאה פלט בגודל $cn^2$. האלגוריתם ל- $B$ לינארי ולכן זמן הריצה שלו יהיה $dcn^2$. אך $dc$ קבוע, ולכן זמן הריצה הוא $\Omega(n^2)$ (במקרה שכל הזמנים של הרדוקציה ושל המכונה הם הדוקים ולא קטנים אסימפטוטית בפועל), כלומר לא בהכרח מתקיים $A \in TIME(n)$.

ד. התשובה שלילית. שוב ננתח את זמן הריצה; הפלט של הרדוקציה הוא בגודל $cn^2$. האלגוריתם ל- $B$ רץ בזמן ריבועי ולכן זמן הריצה שלו יהיה $dc^2n^4$, אך $dc^2$ קבוע, ואם

זמני הריצה הדוקים, נקבל שזמן הריצה של האלגוריתם הוא $\Omega(n^4)$, ולכן לא יתקיים בהכרח $A \in TIME(n^2)$.

29. יהי $\Sigma$ אלפבית נתון.
מצאו את כל השפות $L$ כך ש- $L \equiv_p \Sigma^*$ ואת כל השפות $K$ כך ש- $K \equiv_p \varnothing$.
הסבירו היטב את תשובתכם.
(היחס $\equiv_p$ מוגדר במדריך הלמידה בעמוד 70).

**פתרון:**

תהי שפה $L$ כך ש- $L \equiv_p \Sigma^*$. אז בפרט $L \leq_p \Sigma^*$ ולכן קיימת רדוקציה פולינומית $f$ מ- $L$ ל- $\Sigma^*$. אם $w \in L$ אז $f(w) \in \Sigma^*$, ואם $w \notin L$ אז $f(w) \notin \Sigma^*$. אך לא ייתכן ש- $f(w) \notin \Sigma^*$ (כי $\Sigma^*$ היא שפת כל המילים), ולכן נקבל שאין $w \notin L$, כלומר $L = \Sigma^*$. באופן דומה, $K = \varnothing$.

30. הוכיחו: אם $A$ היא שפה **רגולרית** אז $A \in SPACE(1)$.

**הוכחה:**

$A$ רגולרית ולכן קיים אוטומט סופי דטרמיניסטי $D$ המקבל אותה.
נתייחס אל האוטומט כמכונת טיורינג, ונוסיף לפונקציית המעברים, בכל מעבר, שלא משתמשים בסרט העבודה (הולכים שמאלה וכותבים רווח, לדוגמה). כך מבטיחים שקוראים את הקלט כמו באוטומט, ועושים את אותן פעולות אך לא משתמשים בסרט, ולכן סיבוכיות המקום היא $O(1)$. כלומר $A \in SPACE(1)$ כדרוש.

31. הוכיחו: $CLIQUE \leq_L VERTEX-COVER$.
עליכם לתאר את הרדוקציה, להוכיח שהיא תקפה, ולהוכיח בפירוט שהיא יכולה להתבצע במקום לוגריתמי.

**פתרון:**

קלט: $<G,k>$ כאשר $G$ הוא גרף לא מכוון ו- $k$ הוא מספר שלם.

1. מנה את מספר הצמתים בגרף, $n$, ואחסן אותו בסרט העבודה.
2. הוצא לפלט את הגרף $G$, רק שכל קשת שקיימת – לא תהיה בגרף החדש, וכל קשת שלא הייתה קיימת – תהיה.
3. הוצא לפלט את המספר $n-k$.

**נכונות:** נניח שב- $G$ יש קליקה בגודל $k$. אז יש קבוצה $S$ בגודל $k$ שבין כל שני צמתים שלה יש קשת. לכן בגרף המשלים, לא תהיה אף קשת בין שני צמתים ב- $S$, כלומר $S$ היא קבוצה בלתי תלויה ב- $\overline{G}$. כעת, על פי שאלה 4.10 סעיף 2 במדריך הלמידה, $V-S$ מכסה את קשתות הגרף, כלומר הוא כיסוי בצמתים בגודל $n-k$, כדרוש.

**סיבוכיות:** מניית מספר הצמתים בגרף תיעשה על ידי מונה בסרט העבודה, שגודלו לא עולה על $n$, ולכן אורכו יהיה לכל היותר $\log n$ (אנו מניחים שקבוצת הצמתים שמורה בתחילת הקידוד כך שלא צריך לבדוק את זה בקשתות עצמן). כעת, חיסור בין $n$ ל- $k$ גם הוא לוגריתמי בבירור במקום.
כעת, עבור כל צומת עוברים על כל הצמתים האחרים ובודקים אילו קשתות קיימות ואילו לא, ובהתאם מוציאים לפלט. אין צורך להשתמש בסרט העבודה, אלא לשמירת האינדקס של הצמתים שאנו מחפשים כאשר עוברים לחיפוש בקשתות. לצורך כך משתמשים בשני אינדקסים, שאורכם לכל היותר $\log n$. סה״כ השתמשנו במקום $O(\log n)$ ולכן זו רדוקציה לוגריתמית, כדרוש.

32. **תזכורת:** יהי $\Sigma$ אלפבית סופי. פונקציה $f:\Sigma^* \to \Sigma^*$ היא **פונקציה ניתנת לחישוב** אם יש מכונת טיורינג שעל כל קלט $w \in \Sigma^*$ עוצרת, וכשהיא עוצרת רשומה על הסרט המילה $f(w)$ בלבד.

**הטווח** של פונקציה $f$ כזו הוא הקבוצה $\{f(w) \mid w \in \Sigma^*\}$.

הוכיחו: אם $A$ היא שפה **מזוהה-טיורינג לא ריקה**, אז יש פונקציה **ניתנת לחישוב** שהטווח שלה הוא השפה $A$.

**הדרכה:** זכרו שיש ל-$A$ מונה (enumerator).

**פתרון:**

ניקח מונה לשפה $A$, $E$. הפונקציה החשיבה $f$ תיקח את המחרוזת ה-$i$ בסדר הסטנדרטי (החל מ-0), ותתאים לה את המחרוזת ה-$i+1$ שהודפסה על ידי המונה.

הדרך לעשות זאת תהיה לקחת את המונה $E$ וליצור ממנו מכונת טיורינג:

קלט: מילה $w$.

1. כתוב על הסרט את המילה $\varepsilon$.
2. אם המילה הכתובה שווה ל-$w$, הרץ את המונה עד שיגיע למצב ההדפסה שלו, וכתוב על הסרט רק את המילה שהיה אמור להדפיס (למכונה שלנו לא יהיה פלט, ובמקום זה היא תכתוב את הפלט על הסרט). עצור.
3. אחרת, קדם את המילה הכתובה על הסרט ב-1 לפי הסדר הסטנדרטי, והרץ את המונה עד שיגיע למצב ההדפסה (ללא כתיבת המילה המודפסת על הסרט). חזור ל-2.

33. תהי $C$ שפה. מילה $w$ נקראת **מילה מינימלית** בשפה $C$ אם $w \in C$ אבל כל **תחילית ממש** של $w$ איננה ב-$C$.

נגדיר את השפה $MINIMAL-WORD_{TM}$ להיות שפת כל הזוגות $<M,w>$ כך ש-$M$ היא מכונת טיורינג ו-$w$ היא מילה מינימלית ב-$L(M)$.

הראו **רדוקציית מיפוי** של $A_{TM}$ ל-$MINIMAL-WORD_{TM}$ (הראו: $A_{TM} \le_m MINIMAL-WORD_{TM}$).

**פתרון:**

קלט: $<M,w>$ כאשר $M$ היא מכונת טיורינג ו-$w$ היא מילה.

1. צור את המכונה $M'$ הבאה:
   קלט: $x$, מילה.
   1. הרץ את $M$ על $w$.
   
   אם $M$ קיבלה, קבל אם $x=w$.
2. החזר את $<M',w>$.

**נכונות:** ברור שהפונקציה חשיבה.

נניח ש-$<M,w> \in A_{TM}$. אז $M$ מקבלת את $w$, ולכן $L(M')=\{w\}$. בפרט $w$ היא המילה היחידה ב-$L(M')$, ולכן היא מינימלית. כלומר $<M',w> \in MINIMAL-WORD_{TM}$.

נניח ש-$<M,w> \notin A_{TM}$. אז $L(M')=\emptyset$, ובפרט $w \notin L(M')$ ולכן היא לא מילה מינימלית בשפה הזו. כלומר $<M',w> \notin MINIMAL-WORD_{TM}$. מש"ל.

34. נשתמש בהגדרת מילה מינימלית משאלה 32. עבור שפה $C$ נגדיר את השפה $Min(C)$ להיות שפת כל המילים המינימליות ב-$C$.

(**דוגמה:** $C=(0+1)^*1$, אז $Min(C)=0^*1$.)
נתון ש- $C\in NP\cap coNP$.
הוכח: $Min(C)\in NP\cap coNP$.
**פתרון:**
היות ש- $C\in coNP$, קיים מאמת $V_1$ שמאמת את $\overline{C}$. מכיוון ש- $C\in NP$, קיים מאמת $V_2$ שמאמת את $C$. כך נוכל לבנות מאמת ל- $Min(C)$:
קלט: $<w,c_0,c_1,c_2,...,c_{|w|-1},d>$.
1. לכל $0\le i\le|w|-1$, בדוק בעזרת $V_1$ ומסמך האישור $c_i$ שהרישא בגודל $i$ של $w$ שייכת ל- $\overline{C}$. אם כל הבדיקות הצליחו, המשך.
2. בדוק בעזרת $V_2$ ומסמך האישור $d$ ש- $w\in C$. אם הבדיקה הצליחה, קבל.

**נכונות:** אם $w\in Min(C)$, אז קיים מסמך אישור $d$ כך ש- $V_2$ יקבל את $<w,c>$. בנוסף, כל רישא באורך $i$ של $w$ לא שייכת ל- $C$, ולכן קיים מסמך אישור $c_i$ כך ש- $V_1$ יקבל את $<w_i,c_i>$ ($w_i$ היא הרישא). כל מסמכי האישור האלו יסופקו, המאמת שלנו יקבל. בנוסף, אם יסופקו מסמכי אישור כאלו, הרי שהם מהווים אסמכתא לכך שהמילה בשפה, ושכל רישא לא בשפה, ולכן המילה ב- $Min(C)$.

**זמן ריצה:** כל אחד מהמאמתים רץ בזמן פולינומי ביחס ל-$|w|$, והיות שאנו מריצים $n+1$ פעמים מאמת, נקבל שזמן הריצה הוא פולינומי ביחס ל-$|w|$, כדרוש.

כעת, נבנה מאמת ל- $\overline{Min(C)}$.
קלט: $<w,c>$, כאשר $w$ היא מילה, ומופיע בדיוק אחד מבין $c$ ו- $d$.
1. הרץ את $V_1$ על $<w,c>$. אם הוא קיבל – קבל.
2. הרץ את $V_2$ על כל אחת מהרישות של $w$, $w_i$, כך: $<w_i,c>$. אם אחת מההרצות קיבלה – קבל.

**נכונות:** אם $w\notin Min(C)$, אז או ש- $w\notin C$ ואז יהיה מסמך אישור כך ש- $V_1$ יקבל אותו בשלב 1 ואז המילה תתקבל כדרוש, או שקיימת רישא של $w$ שנמצאת ב- $C$, ואז קיים מסמך אישור עבור הרישא הזו כך ש- $V_2$ יקבל אותה ואז נקבל.
בנוסף, אם קיימים מסמכי אישור מתאימים, אזי או ש- $w\notin C$ או שקיימת רישא מתאימה, ובכל מקרה $w\notin Min(C)$. מש"ל.

**זמן ריצה:** מריצים לכל היותר פעמיים מאמת הרץ בזמן פולינומי ביחס ל-$|w|$, ולכן זמן הריצה הוא בבירור פולינומי.

35. **תזכורת:** הבעיות $CLIQUE$ ו- $INDEPENDENT-SET$ הן $NP$-שלמות.
הוכיחו: גם הבעיה $CLIQUE-AND-INDEPENDENT-SET$, המורכבת מכל הזוגות $<G,k>$ בהם $G$ הוא גרף לא מכוון בעל קליקה בגודל $k$ וקבוצה בלתי תלויה בגודל $k$, היא בעיה $NP$-שלמה.
**פתרון:**
ראשית, הבעיה נמצאת בקלות ב- $NP$, שכן מאמת יכול לקבל שתי קבוצות בגודל $k$ ולאמת שהן מהוות קליקה וקבוצה בלתי תלויה.
כעת, נראה רדוקציה מ- $CLIQUE$ ל- $CLIQUE-AND-INDEPENDENT-SET$.
קלט: $<G,k>$ כאשר $G$ הוא גרף לא מכוון ו- $k$ הוא מספר טבעי.

1. הוסף לגרף $k$ צמתים נוספים שלא מחוברים בקשתות.
2. החזר את הגרף שהתקבל ואת המספר $k$.

**נכונות:**

נניח ש- $<G,k>\in CLIQUE$. אז גם בגרף החדש תהיה קיימת הקליקה הנ"ל, ובנוסף $k$ הצמתים החדשים שנוספו לגרף מהווים קבוצה בלתי תלויה בגרף החדש, ולכן קיימת בו הן קבוצה בלתי תלויה בגודל $k$ והן קליקה בגודל $k$, ולכן $<G',k>\in CLIQUE-AND-INDEPENDENT-SET$.

נניח ש- $<G',k>\in CLIQUE-AND-INDEPENDENT-SET$.

אז בפרט קיימת קבוצה $S$ בגודל $k$ שמהווה בגרף החדש קליקה. לא ייתכן שיש צומת מקבוצת הצמתים החדשה בקליקה, כי הוא לא מחובר לאף צומת אחר, ובפרט לא לשאר צומתי הקליקה (אלא אם כן $k=1$ ואז כל צומת בגרף הישן הוא קליקה). לכן $S$ היא גם קליקה בגרף המקורי, כלומר $<G,k>\in CLIQUE$. מש"ל.

**זמן ריצה:**

מוסיפים $k$ צמתים חדשים, אך $k\le n$ כאשר $n$ הוא מספר הצמתים בגרף הישן, ולכן $k$ לכל היותר לינארי באורך הקלט, כלומר, הפעולה הזו היא בזמן פולינומי באורך הקלט. מש"ל.

36. נאמר ששפות $A$ ו- $B$ הן **L-שקולות** אם גם $A\le_L B$ וגם $B\le_L A$.
   1. הוכיחו: כל שתי שפות $NL$-שלמות הן $L$-שקולות.
   2. האם כל שתי שפות במחלקה $L$ הן $L$-שקולות? הוכיחו את תשובתכם.

**פתרון:**

א. יהיו $A,B$ שפות $NL$-שלמות. אז לכל שפה $K\in NL$, $K\le_L A$. אך $B\in NL$ ולכן $B\le_L A$. באופן סימטרי, $A\le_L B$. מש"ל.

ב. התשובה שלילית. נתבונן בשפה $A=\{a\}$ ובשפה $B=a^*$, שתיהן מעל האלפבית $\Sigma=\{a\}$ נניח בשלילה שהן $L$-שקולות. אז בפרט קיימת רדוקציית מיפוי מ- $A$ ל- $B$. כלומר לכל $w\in A$, $f(w)\in B$, ולכל $w\notin A$, $f(w)\notin B$. בפרט, $f(\varepsilon)\notin B$. אך הרדוקציה היא מעל האלפבית $\{a\}$ ולכן $f(\varepsilon)$ חייב להיות מעל האלפבית $\{a\}$, בסתירה לכך ש- $B$ מורכבת מכל המילים מעל האלפבית $\{a\}$.

37. לכל שפה $D$ מעל אלפבית נתון $\Sigma$ נגדיר את השפה $DROP-SYMBOL(D)$ המורכבת מכל המילים $w\in\Sigma^*$ כך שניתן לקבל את $w$ על ידי השמטת אחת האותיות של מילה כלשהי $x\in D$ (אות יחידה).

**דוגמה:** אם $x=1011$ שייכת ל- $D$ אז $011,111,101$ שייכות ל- $DROP-SYMBOL(D)$.

הוכיחו: אם $D$ שייכת למחלקה **RP**, אזי גם $DROP-SYMBOL(D)$ שייכת למחלקה RP.

**פתרון:**

מכיוון ש- $D\in \mathrm{RP}$, קיימת מכונת RP $M$ המכריעה את $D$.

נבנה מכונה הסתברותית המכריעה את $DROP-SYMBOL(D)$:

קלט: מילה $x$.

1. עבור כל $\sigma\in\Sigma$:
   1.1. עבור כל $i\in\{0,...,n\}$:
      1.1.1. מקם את $\sigma$ לאחר האות ה- $i$-ית ב- $w$, והרץ את $M$ על המילה החדשה. אם $M$ קיבלה, קבל. אחרת – בטל את הוספת $\sigma$ והמשך.
2. דחה.

**נכונות:** כלומר, נוכיח שהאלגוריתם דוחה קלטים שאינם בשפה בהסתברות 1 ומקבל קלטים שבשפה בהסתברות הגדולה מ- $\frac{1}{2}$.

נניח ש- $w \notin DROP - SYMBOL(D)$. אז כל אות שנוסיף בכל מיקום תקיים שהמילה החדשה לא נמצאת ב- $D$. אך היות ש- $M$ היא מכונת RP, היא תדחה את כל הניסיונות הללו בהסתברות 1. כלומר הקלט $w$ יידחה בהסתברות 1 כדרוש.

כעת, נניח ש- $w \in DROP - SYMBOL(D)$. אז קיימת מילה $x$ ב- $D$ כך ש- $w$ מתקבלת מ- $x$ על ידי הסרת אות אחת. לכן $x$ תיבחן באחד מהצעדים 1.1.1. ההסתברות שלא נקבל את המילה היא קטנה או שווה להסתברות שלא נקבל את המילה אם $x$ הנ"ל היא היחידה שמקיימת את התנאי הזה (אחרת יש לנו רק "עוד הזדמנויות" לקבל). אך ההסתברות שנטעה בבדיקה האם $x \in D$ היא פחות מ- $\frac{1}{2}$ כי זוהי מכונת RP ולכן נקבל שהקלט $w$ יידחה בהסתברות הקטנה מחצי, כלומר יתקבל בהסתברות של לפחות חצי, כדרוש. מש"ל.

**זמן ריצה**: הלולאה רצה $O(n^2)$ פעמים ובכל אחת מהן רצה בזמן פולינומי, ולכן סה"כ זמן הריצה הוא פולינומי באורך הקלט.

38. הוכיחו שהשפה $NONEMPTY - INTER_{DFA}$ שלהלן היא שפה **NP-קשה**: השפה מורכבת מכל הצירופים $< A_1, A_2, ..., A_k >$ של אוטומטים סופיים דטרמיניסטיים שחיתוך שפותיהם אינו ריק.

**הדרכה:** הראו רדוקציה פולינומיאלית של $3SAT$.

אפשר לייצג השמה של ערכים בוליאניים ל- $n$ משתנים בוליאניים $x_1, x_2, ..., x_n$ בעזרת מחרוזת באורך $n$ מעל $\{0,1\}$: המקום ה- $i$ במחרוזת יציין את ערכו של $x_i$.

לכל פסוקית בנו אוטומט שיזהה את שפת כל המחרוזות מעל $\{0,1\}$ שמתאימות להשמות שמספקות את הפסוקית.

**פתרון:**

קלט: נוסחה $< \phi >$ כאשר $\phi \in 3SAT$.

1. עבור כל פסוקית $(x_i \vee x_j \vee x_k)$, בנה את האוטומט הבא:

   קלט: מחרוזת $w$ באורך $n$ מעל $\{0,1\}$

   1. נניח ב.ה.כ. ש- $i \le j \le k$.

      קרא ללא שינוי את המחרוזת עד המקום ה- $i$.

      אם התו ה- $i$ הוא 1 ו- $x_i$ לא מתויג, או שהוא 0 ו- $x_i$ מתויג, המשך לקרוא עד סוף המחרוזת וודא שהיא באורך $n$. אם כן – קבל.

   2. בצע בדומה ל-1 עד התו ה- $j$

   3. בצע בדומה ל-1 עד התו ה- $k$.

   4. אם אף אחד מהתווים לא היה שווה 1, דחה.

   היות ש- $n$ ידוע לנו מראש, נוכל לבנות את האוטומט עם $n$ מצבים שיוודאו שיש $n$ תווים בקלט, ותוך כדי יטפלו באינדקסים המתאימים.

2. החזר את האוטומט המתאים לכל פסוקית.

**נכונות:**

נניח ש- $< \phi > \in 3SAT$. אז קיימת השמה מספקת עבור $\phi$. המחרוזת שמתאימה להשמה זו תתקבל על ידי כל אחד מהאוטומטים, שכן היא מהווה השמה מספקת לכל אחת מהפסוקיות. לכן החיתוך של האוטומטים אינו ריק.

כעת, אם החיתוך אינו ריק, אז קיימת מילה שמתקבלת על ידי כולם. אך מקבלים מילה רק אם היא באורך $n$, ולכן היא תהיה מקבילה להשמה. אך עבור כל פסוקית היא מתאימה להשמה מפסקת, ולכן היא מספקת את כל הפסוקיות גם יחד, כלומר מספקת את הנוסחה. לכן קיימת לנוסחה השמה מספקת. מש"ל.

**סיבוכיות:**

בניית כל אוטומט מתבצעת בזמן פולינומי באורך הקלט, ובונים לכל היותר $n$ (שקטן מאורך הקלט) אוטומטים כאלה, ולכן זמן הריצה הוא פולינומי באורך הקלט. מש"ל.

39. נניח ש- $P \neq NP$.

נתון ש- $A$ היא $NP$-שלמה וש- $B$ היא סופית.

**הוכח:** $A \cup B$ היא $NP$-שלמה.

**הוכחה:**

תחילה, $A \cup B \in NP$ בקלות על ידי מאמת שיקבל מסמך אישור מתאים ל- $A$:

קלט: $< w,c >$

1. בדוק האם $w \in B$ על ידי הרצת אוטומט סופי דטרמיניסטי המתאים ל- $B$ (שהיא רגולרית כי היא סופית). אם כן – קבל.
2. הרץ את המאמת של $A$ על $< w,c >$. אם קיבל, קבל.

נכונות האלגוריתם ברורה.

כעת, נראה רדוקציה של $A$ ל- $A \cup B$. היות ש- $B$ ו- $B - A$ סופיות, קיימים אוטומטים סופיים דטרמיניסטיים $D_1$, $D_2$ המקבלים אותן, בהתאמה. נסמן מילה $x \notin A \cup B$. קיימת כזו שכן $A \cup B \neq \Sigma^*$ (אחרת $A \in P$ בקלות).

קלט: מילה $w$.

1. הרץ את $D_1$ על $w$.

   אם $D_1$ דחה, החזר את $w$.
2. הרץ את $D_2$ על $w$.

   אם $D_2$ קיבל, החזר את $x_1$.
3. אחרת, החזר את $w$.

**נכונות:**

נניח ש- $w \notin A$. ייתכנו שתי אפשרויות: $w \in B$ או $w \notin B$. אם $w \in B$, אז $w \in B - A$. אז $D_1$ ו- $D_2$ יקבלו את $w$, ולכן נחזיר את $x$, המקיים $x \notin A \cup B$, כדרוש.

אם $w \notin B$ אזי $D_1$ תדחה את $w$, ולכן נחזיר את $w$ המקיימת $w \notin A \cup B$.

כעת, נניח ש- $w \in A$. שוב ייתכנו שתי אפשרויות: $w \in B$ או $w \notin B$.

אם $w \in B$, היא תתקבל על ידי $D_1$ אך יידחה על ידי $D_2$, ואז נחזיר את $w$ המקיימת $w \in A \cup B$.

אם $w \notin B$, אז היא תידחה על ידי $D_1$ ולכן נחזיר את $w$, המקיימת $w \in A \subseteq A \cup B$, כדרוש.

**זמן ריצה:** הרצת אוטומט סופי דטרמיניסטי אורכת זמן לינארי, ולכן זמן הריצה הכולל הוא פולינומי.

40. נגדיר את השפה $SUB_{CNF}$ להיות שפת הזוגות של דקדוקים חופשיי הקשר $< G_1, G_2 >$ המקיימים $L(G_1) \subset L(G_2)$.

הראו רדוקציית מיפוי של $ALL_{CNF}$ ל- $SUB_{CNF}$ (הראו: $ALL_{CNF} \leq_m SUB_{CNF}$).

**פתרון:**

קלט: $<G>$ כאשר $G$ הוא דקדוק חופשי הקשר.

1. צור דקדוק חסר הקשר $G'$ שיקבל את כל המילים ב-$\Sigma^*$ פרט למילה אחת (נאמר 0 אם אלפבית הקלט הוא $\{0,1\}$ לדוגמה).
2. הוצא לפלט את $<G',G>$.

**נכונות:**

תחילה, ברור שזוהי פונקציה ניתנת לחישוב. כל מה שצריך הוא לבנות דקדוק כזה, כאשר ברור שהוא קיים שכן שפת כל המילים פרט למילה אחת היא רגולרית (כי משלימתה סופית).

כעת, נניח ש-$<G>\in ALL_{CNF}$. אז $L(G)=\Sigma^*$, ובפרט $G$ מקבל את כל המילים של $G'$, ו-$G'$ לא מקבל את המילה היחידה לעיל, ולכן $L(G')\subset L(G)$, כלומר $<G',G>\in SUB_{CNF}$, כדרוש.

נניח ש-$<G',G>\in SUB_{CNF}$. אז $L(G')\subset L(G)$. אך $L(G')$ מכילה את כל המילים פרט לאחת, ולכן $L(G)$ חייבת להכיל את כל המילים. כלומר $L(G)=\Sigma^*$ או במילים אחרות, $<G>\in ALL_{CNF}$, כדרוש. מש"ל.

41. נגדיר את השפה $CLIQUE-SUB$ להיות שפת כל הזוגות $<G,k>$ כך ש-$G$ הוא גרף לא מכוון, וניתן לחלק את קודקודי $G$ ל-$k$ קבוצות זרות שהאיחוד שלהן הוא $V$ (קבוצת הצמתים של $G$), וכל קבוצה היא קליקה.
הראו ש-$CLIQUE-SUB$ היא $NP$-שלמה.
**רמז:** הסתכלו על הגרף המשלים.

**פתרון:**

תחילה, השפה ב-$NP$ על ידי מאמת שיקבל $k$ תת קבוצות כאלו, ויבדוק שכל אחת היא קליקה, שהן זרות (יש לבדוק כל זוג בנפרד בזמן ריבועי), ושאיחודן הוא כל הצמתים של $G$.

כעת, נראה רדוקציה של $COLORING$ ל-$CLIQUE-SUB$.

קלט: $<G,k>$ כאשר $G$ הוא גרף לא מכוון ו-$k$ הוא מספר טבעי.

1. צור גרף $G'$ הזהה ל-$\overline{G}$, על ידי הוספת כל קשת שאינה ב-$G$.
2. החזר את $<G'>$.

**נכונות:**

נניח ש-$<G,k>\in COLORING$. אז קיימת $k$-צביעה של $G$. נתבונן בחלוקה הבאה ל-$k$ קבוצות:

קבוצה אחת תכיל את כל הצמתים שנצבעו בצבע ה-1.
קבוצה שנייה תכיל את כל הצמתים שנצבעו בצבע ה-2.
וכן הלאה... הקבוצה ה-$k$ תכיל את כל הצמתים שנצבעו בצבע ה-$k$.

ראשית, ברור שאלו קבוצות זרות (אף צומת לא נצבע בשני צבעים), ובנוסף איחודן הוא $V$, כי כל צומת נצבע בצבע כלשהו.

כל קבוצה מבין הקבוצות האלו, היא קבוצה בלתי תלויה ב-$G$, שכן לו היו בה שני צמתים המחוברים בקשת, זו הייתה סתירה לכך שזו צביעה.

בשל כך, כל שני צמתים מכל קבוצה כזו בגרף המשלים יהיו מחוברים בקשת. כלומר כל קבוצה כזו היא קליקה בגרף $G'$. כלומר ב-$G'$ יש $k$ קבוצות זרות שאיחודן הוא $V$, וכל אחת היא קליקה בגרף. לכן $COLORING\leq_p CLIQUE-SUB$ ו-$CLIQUE-SUB\in NP$ ולכן היא $NP$-שלמה. מש"ל.

42. נגדיר את המחלקה **RL** להיות מחלקת השפות שקיימת מכונת טיורינג הסתברותית הרצה בזמן לוגריתמי, כך שכל קלט שאינו בשפה נדחה על ידן בהסתברות 1, וכל קלט בשפה מתקבל בהסתברות שאינה קטנה מ- $\frac{1}{2}$.

הוכיחו ש- $\mathrm{RL} \subseteq SPACE(\log^2 n)$.

**הוכחה:**

תהי $A \in \mathrm{RL}$. נתבונן במכונה ההסתברותית הלוגריתמית המכריעה אותה, $M$. במקום להתייחס אליה כאל מכונה הסתברותית, נתייחס לכל הטלת מטבע כאל בחירה אי-דטרמיניסטית. נקבל מכונה לא דטרמיניסטית, ונוכיח שגם היא מכריעה את $A$.

תהי $w \notin A$. אז $M$ דוחה את $w$ בהסתברות 1, כלומר כל ענף חישוב על $w$ מסתיים בדחייה. לכן $w$ תידחה.

תהי $w \in A$. אז $M$ מקבלת את $w$ בהסתברות שאינה קטנה מ- $\frac{1}{2}$, ובפרט קיים ענף חישוב שבו $w$ מתקבלת. כלומר המכונה הלא דטרמיניסטית תקבל את $w$.

כלומר, ל- $A$ יש מכונת טיורינג לא דטרמיניסטית הרצה במקום $O(\log n)$. אך על פי משפט סביץ' (כולל ההרחבה לפונקציות לוגריתמיות), נובע מכך שקיימת מכונה דטרמיניסטית המכריעה את $A$ במקום $O(\log^2 n)$. כלומר $A \in SPACE(\log^2 n)$. מש"ל.

43. תהי $NPC_{TM}$ שפת כל התיאורים של מכונות טיורינג, $<M>$, כך ש- $L(M)$ היא $NP$-שלמה.

הוכח ש- $NPC_{TM}$ אינה כריעה על ידי רדוקציית מיפוי מ- $A_{TM}$ ל- $NPC_{TM}$. ניקח תיאור של מכונת טיורינג המכריעה את $3SAT$, $P$.

**הוכחה:**

קלט: $<M,w>$ כאשר $M$ היא מכונת טיורינג ו- $w$ היא מילה.

1. בנה את המכונה $M'$ הבאה:

   קלט: $x$, מילה.

   1. הרץ את $M$ על $w$.

   אם $M$ קיבלה – הרץ את $P$ על $x$, והחזר את התשובה.

2. החזר את $<M'>$.

**נכונות:**

נניח ש- $<M,w> \in A_{TM}$.

אז $M$ תקבל את $w$ ולכן $M'$ תפעל כמו $P$, כלומר $L(M') = 3SAT$, כלומר $L(M')$ היא $NP$-שלמה, כדרוש.

כעת, אם $<M,w> \notin A_{TM}$, $L(M') = \varnothing$ אך זוהי אינה שפה $NP$-שלמה, ולכן $<M'> \notin NPC_{TM}$. מש"ל.

44. נתון ש- $A \in P$ וש- $A \cup B \in NP$.

1. האם בהכרח $B \in NP$? הוכח.
2. האם תשובתך תשתנה אם נתון ש- $A$ סופית?

**פתרון:**

א. התשובה שלילית. נבחר את $B$ להיות שפה לא כריעה כלשהי, לדוגמה $B = A_{TM}$. אז $B \notin NP$. אך עם זאת, אם נבחר $A = \Sigma^*$, אז $A \cup B = \Sigma^*$, ו- $\Sigma^*$ היא בוודאי ב- $NP$ (היא רגולרית). מש"ל.

ב. אם $A$ סופית אז גם $A - B$ סופית. לשתיהן יש אוטומטים סופיים דטרמיניסטיים, נאמר $D_1$ ו- $D_2$, בהתאמה.

נבנה מאמת לשפה $B$:

קלט: $< w, c >$.

1. בדוק בעזרת מאמת פולינומי ל- $A \cup B$ ש- $w \in A \cup B$. אם הוא דחה, דחה.
2. בדוק בעזרת האוטומט $D_1$ האם $w \in A$. אם לא, קבל. אם כן, בדוק האם $w \in A - B$. אם כן, קבל. אחרת – דחה.

**נכונות:**

אם $w \in B$, אז בפרט $w \in A \cup B$ וקיים מסמך אישור $c$ כך שהמאמת לאיחוד יקבל אותו. כעת, אם $w \notin A$, אז בהכרח מתקיים $w \in B$ ואפשר לקבל. אך אם הוא כן ב- $A$, יש לוודא שהוא נמצא גם ב- $B$, וזאת עושים על ידי בדיקה האם הוא ב- $A - B$.

**זמן ריצה:** המאמת לאיחוד רץ בזמן פולינומי, והאוטומטים רצים בזמן לינארי ולכן סה"כ זמן הריצה של המאמת פולינומי.

45. נגדיר את השפה $EHAMPATH - AND - IS$ להיות שפת הזוגות $< G, k >$ כך ש- $G$ הוא גרף לא מכוון בעל מסלול המילטון, ויש ב- $G$ קבוצה בלתי תלויה בגודל $k$.

הוכח שהשפה היא $NP$-שלמה.

**פתרון:**

תחילה, השפה ב- $NP$ בקלות על ידי מאמת שיקבל מסלול, יבדוק שהוא מכיל את כל צמתי הגרף בדיוק פעם אחת, ושיש קשת בין כל שני צמתים סמוכים. בנוסף, הוא יקבל קבוצה בגודל $k$ ויבדוק שאין בין שני צמתים בקבוצה זו קשת.

כעת, נראה רדוקציה פולינומית מ- $HAMPATH$ ל- $EHAMPATH - AND - IS$.

קלט: $< G, s, t >$ כאשר $G$ הוא גרף לא מכוון ו- $s$ ו- $t$ הם צמתים.

1. הוסף ל- $G$ צומת $s^*$ וקשת בינו לבין $s$ וצומת $t^*$ וקשת בינו לבין $t$. קרא לגרף החדש $G'$.
2. החזר את $< G', 1 >$.

**נכונות:**

נניח ש- $< G, s, t > \in HAMPATH$. אז למסלול ההמילטון ב- $G$ נוסיף בקצוות את $s^*$ ו- $t^*$ בהתאמה, ונקבל מסלול המילטון ב- $G'$. בנוסף, כל צומת יהווה קבוצה בלתי תלויה בגודל 1 (באופן ריק). כך הוכחנו ש- $< G', 1 > \in EHAMPATH - AND - IS$.

כעת, נניח ש- $< G', 1 > \in EHAMPATH - AND - IS$. אז קיים מסלול המילטון ב- $G'$. אך לא ייתכן שהצומת $s^*$ אינו צומת קצה שלו, שכן לא ניתן להיכנס ולצאת מהצומת ללא חזרה על צומת אחר (כי יש אליו רק קשת נכנסת), ובדומה גם $t^*$ חייב להיות צומת קצה. לכן אם נסיר אותם נקבל מסלול המילטון $s - t$ בגרף המקורי, ולכן $< G, s, t > \in HAMPATH$, כדרוש.

**זמן ריצה:** זמן הריצה פולינומי בבירור.

46. עבור שפה $A$ כלשהי, נגדיר את השפה $DOUBLE(A) = \{ww \mid w \in A\}$.

הוכח שאם $A$ היא $NL$-שלמה, אז גם $DOUBLE(A)$ היא $NL$-שלמה.

**הוכחה:**

תחילה, $A \in NL$ ולכן קיימת מכונה לוגריתמית במקום ולאדטרמיניסטית, $M$, המכריעה את $A$. נבנה מכונה $M'$ שתכריע את $A$ במקום לוגריתמי.

קלט: מילה $x$.

1. מנה את מספר התווים של $A$, וחלק מנייה זו לשתיים. עבוד על הקלט רק מתחילתו ועד לאינדקס שקיבלת. (אם הוא אי זוגי – דחה).
2. הרץ את $M$ על הקלט כמוסבר לעיל. אם היא קיבלה – קבל. אחרת – דחה.

**נכונות:** ברורה מנכונות $M$.

**סיבוכיות:** המונה אורכו לכל היותר $\log n$, וכך גם חלוקתו בשניים. מעבר לכך משתמשים במקום שבו מכונה $M$ משתמשת, אך הקלט שלה לינארי בגודל הקלט שלנו, ולכן היא משתמש במקום לוגריתמי. מש״ל.

כעת, נראה רדוקציה $A \leq_L DOUBLE(A)$.

קלט: $w$, מילה.

1. הוצא לפלט את $w$ ומיד לאחר מכן חזור עם הראש הקורא שמאלה עד הסוף והדפס שוב את $w$.

**נכונות:** אם $w \in A$ אז הפלט יהיה $ww \in DOUBLE(A)$ כדרוש. אם הפלט ב-$DOUBLE(A)$, אז $ww \in DOUBLE(A)$, כלומר בהכרח $w \in A$. מש״ל.

**סיבוכיות:** לא משתמשים בסרט העבודה ולכן סיבוכיות המקום היא $O(1)$ ולבטח לוגריתמית.

לכן $A \leq_L DOUBLE(A)$ שהיא מצידה ב-$NL$ ולכן היא $NL$-שלמה. מש״ל.

47. **תזכורת:** השפה $HAMPATH$ היא $NP$-שלמה.

נגדיר את השפה $ALMOST-HP$: מילה $<G,s,t>$ שייכת ל-$ALMOST-HP$ אם $G$ הוא גרף מכוון, $s$ ו-$t$ צמתים ב-$G$, ויש ב-$G$ שני צמתים $u$ ו-$v$ שאין ביניהם קשת, ואם נוסיף את הקשת $(u,v)$ ל-$G$ אז יהיה ב-$G$ מסלול המילטון מ-$s$ ל-$t$.

1. להלן ״הוכחה״ לכך שהשפה $ALMOST-HP$ היא שפה $NP$-שלמה:

אפשר לבנות מאמת בעל זמן ריצה פולינומיאלי לשפה:

הקלט למאמת יהיה (בנוסף לגרף $G$ ולצמתים $s$ ו-$t$) קשת $e$ שיש להוסיף ורשימה של קשתות של מסלול מ-$s$ ל-$t$.

המאמת יוודא שהקשת $e$ לא נמצאת ב-$G$, ורשימת הקשתות הנתונה אכן מהווה מסלול המילטון מ-$s$ ל-$t$ (לאחר ההוספה של הקשת $e$). אם כן, הוא יקבל. אם לא, הוא ידחה.

אפשר להראות רדוקציה בזמן פולינומיאלי של $HAMPATH$ ל-$ALMOST-HP$:

מוסיפים לגרף $G$ צומת חדש $v$. נקרא לגרף המתקבל $G'$. מחזירים את $<G',v,t>$.

ברור שהרדוקציה ניתנת לחישוב בזמן פולינומיאלי.

הרדוקציה תקפה: אם יש ב-$G$ מסלול המילטון מ-$s$ ל-$t$, אז אם נוסיף ל-$G'$ את הקשת $(v,s)$ יהיה ב-$G'$ מסלול המילטון מ-$v$ ל-$t$. אם אין ב-$G$ מסלול המילטון מ-$s$ ל-$t$, אז תוספת הקשת הזו לא תיצור מסלול המילטון מ-$v$ ל-$t$ ב-$G'$.

מה **לא בסדר** בהוכחה הזו? הסבירו היטב **מה השגיאה** בהוכחה.

2. הוכיחו: אם יש אלגוריתם **בעל זמן ריצה פולינומיאלי** להכרעת השייכות לשפה $ALMOST-HP$, אז אפשר לבנות **בעזרתו** אלגוריתם **בעל זמן ריצה פולינומיאלי** לבעיה הבאה (של **מציאת** הקשת):

הקלט: גרף מכוון $G$ ושני צמתים $s$ ו- $t$.

הפלט: קשת $(u,v)$ שלא נמצאת ב- $G$, ואם מוסיפים אותה ל- $G$, יהיה בגרף המתקבל מסלול המילטון מ- $s$ ל- $t$. אם לא קיימת קשת כזו, מוחזר "לא".

**הדרכה:** האלגוריתם יקרא מספר פולינומיאלי של פעמים לאלגוריתם ההכרעה של $ALMOST-HP$, כל פעם עם קלט אחר. (אלגוריתם ההכרעה עונה רק "כן" או "לא" על הקלט שלו).

מחפשים מסלול המילטון מ- $s$ ל- $t$. יש דרך לדאוג לכך שתהיה חסרה קשת אחת במסלול כזה.

**הערות:** מילה $<G,s,t>$ שייכת ל- $ALMOST-HP$ גם אם יש ב- $G$ מסלול המילטון מ- $s$ ל- $t$.

אינכם רשאים להניח שגם לשפה $HAMPATH$ (או לכל שפה $NP$-שלמה אחרת) יש אלגוריתם בעל זמן ריצה פולינומיאלי. אפשר להשתמש רק בנתון שלשפה $ALMOST-HP$ יש אלגוריתם בעל זמן ריצה פולינומיאלי.

**פתרון:**

א. הבעיה הוא בכיוון השני; אם לא קיים ב- $G$ מסלול המילטון $s-t$ אמנם זה נכון שהוספת הקשת לא תיצור מסלול המילטון מ- $v$ ל- $t$, אך ייתכן שניתן להוסיף קשת אחרת מ- $v$ לצומת אחר, כך שכן יווצר מסלול המילטון בגרף החדש. לכן זו לא רדוקציית מיפוי תקפה.

ב. נניח שהמכונה $M$ היא מכונת ההכרעה הפולינומיאלית ל- $ALMOST-HP$.

קלט: $<G,s,t>$ כאשר $G$ הוא גרף מכוון ו- $s$ ו- $t$ הם צמתים.

1. הרץ את $M$ על $G$. אם נדחה – דחה.
2. בדוק האם יש קשת שאינה שייכת ל- $G$. אם אין כזו – דחה.
3. שכפל את $G$, ועבוד רק על עותק אחד.
4. לכל $e \in E$:

4.1. הסר את $e$ מהגרף.

4.2. הרץ את $M$ על הגרף החדש. אם התשובה שלילית, החזר את הקשת הזו, סמן אותה, ולעולם אל תסיר אותה שוב.

4.3. אם מספר הקשתות בגרף הנוכחי קטן או שווה ל- $|V|-2$, צא מהלולאה.

5. בגרף הנוכחי קיימת קשת יחידה שאם נוסיף אותה, יתקבל מסלול המילטון. עבור על כל הקשתות האפשריות, בדוק מי מהן יוצרת מסלול המילטון. אם היא הייתה בגרף קודם (בדוק בעותק המקורי), החזר קשת כלשהי שלא הייתה שייכת לגרף. אחרת, החזר אותה.

**נכונות:**

תחילה, לא ייתכן ש"יגמרו" הקשתות להסיר בשלב 4 לפני שנגיע לתנאי העצירה, שכן כל עוד יש יותר מ- $|V|-2$ קשתות, $|V|-2$ מהן מספיקות לייצור מסלול המילטוני באמצעות קשת נוספת.

כלומר, בוודאות מסיימים את הלולאה כאשר יש בגרף $|V|-2$ קשתות. היות שמסירים קשת מהגרף רק כאשר בהוספת קשת אחת ניתן לקבל מסלול המילטוני בגרף, נקבל שקיימת קשת שנוסיף אותה ונקבל בו מסלול המילטוני. כל הקשתות בגרף שקיבלנו שייכות גם לגרף המקורי (רק הסרנו קשתות). כעת, אם הקשת הנוספת הייתה שייכת ל- $G$, יש בו מסלול המילטוני ובוודאות ניתן להחזיר כל קשת שאינה שייכת לגרף המקורי. אם היא לא הייתה שייכת ל- $G$, אז ניתן להוציא לפלט אותה.

**סיבוכיות זמן:** שלבים 1-3 הם פולינומיים בבירור. שלב 4 מריץ אלגוריתם פולינומי לכל היותר $|E|$ פעמים, ולכן רץ בזמן פולינומי. בדיקה האם גרף בעל $|V|-1$ צמתים הוא מסלול המילטון, ניתן לעשות על ידי ספירת הדרגה של כל צומת, מציאת שני הצמתים מדרגה 1, ומעבר קשת אחר קשת תוך בדיקה שמגיעים לצומת השני מדרגה 1, ושלא חוזרים על צומת פעמיים. היא ניתנת לביצוע בזמן פולינומי.

48. נגדיר את השפה $MAJORITY-PATH$ להיות שפת כל השלשות $<G,a,b>$ כך ש- $G$ הוא גרף מכוון, $a$ ו- $b$ הם צמתים ב- $G$, ויש ב- $G$ מסלול $P$ מ- $a$ ל- $b$ כך ש**יותר ממחצית הקשתות** של $G$ שייכות למסלול $P$.

הוכיחו: $PATH$ ניתנת לרדוקציית **מקום לוגריתמי** ל- $MAJORITY-PATH$ (הראו: $PATH \leq_L MAJORITY-PATH$).

תארו את הרדוקציה, הוכיחו שהיא תקפה, והראו שהיא ניתנת לחישוב במקום לוגריתמי.

**פתרון:**

קלט: $<G,s,t>$ כאשר $G$ הוא גרף מכוון ו- $s$ ו- $t$ הם צמתים.

1. מנה את מספר הקשתות ב- $G$, נאמר שהוא $n$.
2. הוצא לפלט את הגרף $G$, כאשר תוסיף לו $n$ צמתים המחוברים כשרוך (כל אחד מחובר לשני שכנים פרט לראשון), ואת הצומת האחרון חבר ל- $s$ (לצומת ה"ראשון" של השרוך קרא $a$), ו- $n$ צמתים נוספים במבנה דומה, כאשר את הראשון חבר ל- $t$. לצומת ה"אחרון" קרא $b$.
3. הוצא לפלט את $a$ ואת $b$ (כלומר סה"כ $<G',a,b>$).

**נכונות:** נניח ש- $<G,s,t>\in PATH$. אז המסלול בין $a$ ל- $b$ יעבור דרך כל $2n$ הקשתות החדשות בבירור, ובנוסף בעוד קשתות. אך סך כל הקשתות בגרף החדש הוא $3n$, ולכן בהכרח המסלול עובר ביותר מחצי מהקשתות של $G'$.

נניח ש- $<G,s,t>\notin PATH$. אך אם היה מסלול בין $a$ ל- $b$ בגרף החדש אז ניתן היה להסיר ממנו את הצמתים החדשים ולקבל מסלול מ- $s$ ל- $t$. לכן בהכרח לא קיים מסלול בין $a$ ל- $b$ בגרף החדש, ולכן $<G',a,b>\notin MAJORITY-PATH$.

**סיבוכיות:** כל מה שעושים עם סרט העבודה הוא למנות (פעמיים) את מספר הצמתים, באורך $\log n$, לחסר ממנו עד שמגיעים ל-0 (או יותר), ולכן סיבוכיות המקום לוגריתמית.

49. נתונה השפה $XOR-HALT_{TM}$, המורכבת מכל השלשות $<M,u,v>$ כך ש- $M$ **עוצרת** על אחת ורק על אחת מהמילים $u$ ו- $v$.

האם השפה $XOR-HALT_{TM}$ **מזוהה-טיורינג**? הוכיחו את תשובתכם.

**פתרון:**

התשובה שלילית. נראה רדוקציה $HALT_{TM} \leq_m XOR-HALT_{TM}$.

קלט: $<M,w>$ כאשר $M$ מכונת טיורינג ו- $w$ מילה.

1. צור את המכונה $M'$:
   קלט: מילה $x$.
   1. הרץ את $M$ על $w$.
   אם $M$ עצרה, כנס ללולאה אם ורק אם $x \neq a$.
2. החזר את $<M',a,aa>$.
   ($a \in \Sigma$ כלשהי).

**נכונות:**

נניח ש- $<M,w>\in HALT_{TM}$. אז $M$ תעצור על $w$ ולכן $M'$ תעצור רק על הקלט $x=a$. בפרט היא תעצור רק על אחת מבין המילים $a$ ו- $aa$, ולכן $<M',a,aa>\in XOR-HALT_{TM}$.

נניח ש- $<M,w>\notin HALT_{TM}$. אז $M$ לא תעצור על $w$ ולכן לא תעצור על כל מילה. כלומר $M'$ תיכנס ללולאה גם על $a$ וגם על $aa$, כלומר $<M',a,aa>\notin XOR-HALT_{TM}$.

לכן $HALT_{TM}\le_m XOR-HALT_{TM}$, ומכיוון ש- $HALT_{TM}$ לא כריעה, גם $XOR-HALT_{TM}$ לא כריעה.

50. $A$ ו- $B$ הן שתי שפות מעל אלפבית נתון $\Sigma$.
נתון שיש מכונת טיורינג דטרמיניסטית $M$ שאלפבית הקלט שלה הוא $\Sigma$, ומתקיים:
אם הקלט למכונה $M$ הוא מילה $w$ ששייכת ל- $A$, אז מספר הצעדים שהמכונה רצה עד לעצירה אינו גדול מ- $|w|^k$ עבור $k$ נתון כלשהו, ובסיום הריצה רשומה על הסרט של המכונה מילה ששייכת ל- $B$.
אם הקלט למכונה הוא מילה $v$ שלא שייכת ל- $A$, אז או שמספר הצעדים שהמכונה רצה עד לעצירה אינו גדול מ- $|w|^k$ (אותו $k$ שלעיל), ובסיום הריצה רשומה על הסרט של המכונה מילה שלא שייכת ל- $B$, או שהמכונה אף פעם לא עוצרת.
נתון שהשפה $B$ שייכת למחלקה $P$. האם **בהכרח** גם השפה $A$ שייכת ל- $P$? **הוכיחו** את תשובתכם.

**פתרון:**

התשובה חיובית.

נבנה אלגוריתם לשפה $A$.

קלט: מילה $w$

1. הרץ את המכונה $M$ על $w$ במשך $|w|^k$ צעדים.
2. אם היא לא עצרה – דחה.
3. אם היא עצרה – הרץ את המכונה הפולינומית של $B$ על תוכן הסרט הנוכחי כקלט. אם היא קיבלה – קבל. אם היא דחתה – דחה.

**נכונות:** אם $w\in A$, אז המכונה $M$ תעצור בשלב 1 עם קלט שנמצא ב- $B$ ולכן המכונה הפולינומית של $B$ תקבל. אם $M$ לא עוצרת על $w$ הוא לא שייך לשפה ואז דוחים ב-2. אם $M$ דוחה את $w$, אז המכונה $M$ תעצור עם קלט שאינו ב- $B$, ולכן המכונה הפולינומית תדחה את הקלט, כדרוש.

**סיבוכיות זמן:** מכיוון ש- $M$ רצה זמן פולינומי באורך הקלט, אורך המילה שיש על הסרט שלה בסוף הריצה הוא פולינומי באורך הקלט. אך המכונה ל- $B$ היא פולינומית, וזמן ריצה פולינומי במילה באורך פולינומי באורך הקלט הוא זמן ריצה פולינומי. לכן $A\in P$.

51. הוכח שהמחלקה $RP$ סגורה תחת פעולת האיחוד.

**פתרון:**

יהיו $A,B\in RP$, ונסמן ב- $M$ ו- $N$ את מכונות ה- $RP$ המכריעות אותן. נבנה מכונת $RP$ שמכריעה אותן.

קלט: מילה $w$.

1. הרץ את $M$ על $w$. אם היא קיבלה – קבל.
2. הרץ את $N$ על $w$. אם היא קיבלה – קבל. אם היא דחתה – דחה.

**נכונות:**

נניח ש- $w \notin A \cup B$. אז $w \notin A$ וגם $w \notin B$, ולכן, מנכונות $M$ ו- $N$, שתיהן ידחו בהסתברות 1, ולכן המכונה החדשה תדחה בהסתברות 1.
נניח ש- $w \in A \cup B$. ייתכנו שלושה מקרים:

א. $w \in A, w \notin B$. במקרה זה שלב 1 יקבל בהסתברות של לפחות חצי.

ב. $w \in B, w \notin A$. שוב שלב 2 יקבל בהסתברות של לפחות חצי.

ג. $w \in A, w \in B$. גם כאן שלב 1 יקבל בהסבתרות של לפחות חצי.

כל אחד מהמקרים מקבל בהסתברות של לפחות חצי, ולכן בהכרח המכונה שלנו מקבלת בהסתברות של לפחות חצי, כדרוש.

זמן הריצה הוא פולינומי שכן מריצים שני אלגוריתמים פולינומיים.

52. הוכח שהמחלקה $RP$ סגורה תחת פעולת החיתוך.
**פתרון:**
באותם סימונים של השאלה הקודמת.
קלט: מילה $w$.

1. הרץ את $M$ על $w$. אם היא דחתה – דחה.
2. הרץ את $N$ על $w$. אם היא דחתה – דחה. אם היא קיבלה – קבל.

**נכונות:**
נניח ש- $w \notin A \cap B$. אז $w \notin A$ או $w \notin B$. בכל מקרה אחד מהשלבים 1 או 2 ידחו כי $M$ ו- $N$ הן מכונות $RP$, ולכן המכונה תדחה בהסתברות 1.

נניח ש- $w \in A \cap B$. אז $w \in A$ ולכן 1 ידחה בהסתברות הקטנה מ- $\frac{1}{2}$, וגם ההסתברות ש-2 ידחה קטנה מ- $\frac{1}{2}$. בפרט בהכרח המכונה כולה תדחה בהסתברות הקטנה מ- $\frac{1}{2}$.

53. כזכור, מכונת טיורינג יכולה בריצתה על מילת קלט $w$ לסיים במצב המקבל $q_{accept}$, לסיים במצב הדוחה $q_{reject}$, או לא לעצור.
תהי $M$ מכונת טיורינג. נגדיר את השפות הבאות:
$Acc(M)$: קבוצת המילים שעליהן $M$ מסיימת במצב המקבל.
$Rej(M)$: קבוצת המילים שעליהן $M$ מסיימת במצב הדוחה.
$Loop(M)$: קבוצת המילים שעליהן $M$ לא עוצרת.
**הערה: סעיפים שונים בשאלה זו לקוחים מבחינות שונות.**
לכל טענה, האם היא נכונה? הוכיחו את תשובתכם.

א. אם $Loop(M)$ **סופית**, אז $Acc(M)$ **כריעה**.

ב. אם $Acc(M) \cup Loop(M)$ **כריעה**, אז $L(M)$ **כריעה**.

ג. אם $Acc(M) \cup Rej(M)$ **כריעה**, אז $L(M)$ **כריעה**.

ד. אם $Acc(M)$ **כריעה**, אז $Rej(M)$ **כריעה**.

ה. אם $Loop(M)$ **מזוהה-טיורינג**, אז היא **כריעה**.

**פתרון:**

א. הטענה נכונה. אם $Loop(M)$ סופית היא רגולרית ובפרט קיימת לה מכונת טיורינג המכריעה אותה, $N$. נבנה מכונה להכרעת $Acc(M)$:
קלט: מילה $w$.

1. הרץ את $N$ על $w$. אם היא קיבלה – דחה.
2. הרץ את $M$ על $w$. אם היא קיבלה – קבל. אם היא דחתה – דחה.

**נכונות:** אם $w \in Acc(M)$, אז $w \notin Loop(M)$ ולכן $N$ תדחה את $w$, ונמשיך. ואז $M$ בטוח עוצרת על $w$, וכל שנותר להכריע הוא האם היא מקבלת או דוחה.

אם $w \notin Acc(M)$, אז או ש- $M$ נכנסת ללולאה על $w$, ואז בשורה 1 נדחה את המילה, או שהיא דוחה את $w$, ואז בשורה 2 נדחה אותה.

ב. נתבונן במכונה הטבעית המזהה את השפה $HALT_{TM}$: בהינתן קלט $<M,w>$, המכונה תריץ את $M$ על $w$, תקבל אם המכונה עצרה. אם המכונה נכנסה ללולאה, גם המכונה שבנינו תיכנס ללולאה.

אף קלט לא **נדחה**, ולכן $Acc(K) \cup Loop(K) = \Sigma^*$, וזו ודאי שפה כריעה. עם זאת, הוכח בספר ש- $L(K) = HALT_{TM}$ איננה כריעה. בכך סתרנו את הטענה.

ג. הטענה נכונה. נבנה מכונה להכרעת $L(M)$. נניח שהמכונה $K$ מכריעה את $Acc(M) \cup Rej(M)$.

קלט: מילה $w$.

1. הרץ את $K$ על $w$. אם היא דחתה – דחה.
2. הרץ את $M$ על $w$, והוצא את אותו פלט.

**נכונות:** אם $K$ מקבלת, אז $M$ עוצרת על $w$ וניתן להריץ בבטחה ולהחזיר את התוצאה של $M$. אחרת, $M$ נכנסת ללולאה וניתן לדחות.

ד. נבנה את המכונה $M$ הבאה: בהינתן קלט $<N,w>$, המכונה תריץ את $N$ על $w$. אם היא עצרה – $M$ תדחה. במכונה זו, $Rej(M) = HALT_{TM}$, שכן זוהי קבוצת הקלטים עבורם $M$ עוצרת על $w$. ידוע שזו לא שפה כריעה. אך $Acc(M) = \varnothing$ והיא כריעה בבירור, וכך סתרנו את הטענה.

ה. הטענה נכונה. נניח ש- $Loop(M)$ מזוהה טיורינג. אך המשלימה של $Loop(M)$ היא $Acc(M) \cup Rej(M)$, והיא כמובן מזוהה על ידי מכונה שמקבלת כל קלט עליו $M$ עוצרת. לכן גם היא וגם המשלימה שלה מזוהה, וממשפט בספר נקבל שהשפה כריעה. כלומר $Loop(M)$ כריעה, כדרוש.

54. נתון ש- $A \le_m \overline{A}$.

הוכיחו: או ש- $A$ **כריעה** או ש- $A$ **איננה מזוהה טיורינג**.

**פתרון:**

נניח ש- $A$ לא כריעה. אז ממשפט 4.22, $A$ לא מזוהה או $\overline{A}$ לא מזוהה. נחלק למקרים:

- אם $A$ לא מזוהה טיורינג, סיימנו.
- אם $\overline{A}$ לא מזוהה טיורינג, נשתמש ברדוקציה הנתונה. על פי המדריך, נכון גם $\overline{A} \le_m A$. ואז על פי משפט 5.29, נקבל ש- $A$ לא מזוהה טיורינג. מש"ל.

55. **תזכורת:** השפה $3COLOR$ של כל הגרפים הלא מכוונים שיש לצומתיהם צביעה חוקית בשלושה צבעים (צביעה היא חוקית אם כל שני צמתים המחוברים בקשת צבועים בצבעים שונים) היא שפה $NP$-שלמה.

נגדיר את השפה $SAME-COLOR$ להיות שפת השלשות $<G,u,v>$ כך של- $G$ יש 3-צביעה חוקית שבה $u$ ו- $v$ **צבועים באותו הצבע**.

הראו **רדוקציה בזמן פולינומיאלי** של $SAME-COLOR$ ל- $3COLOR$ (הראו: $SAME-COLOR \le_p 3COLOR$).

תארו את הרדוקציה, הוכיחו שהיא תקפה ושהיא ניתנת לחישוב בזמן פולינומיאלי.
שימו לב היטב לכיוון של הרדוקציה שצריך להראות!

**פתרון:**

קלט: $<G,u,v>$ כך ש- $G$ הוא גרף מכוון ו- $u$ ו- $v$ הם צמתים.

1. הוסף לגרף שני צמתים $u'$ ו- $v'$, וחבר כל אחד מהם ל- $u$ ול- $v$, וחבר ביניהם.
2. החזר את $<G'>$.

**נכונות:**

נניח ש- $<G,u,v>\in SAME-COLOR$. אז ל- $G$ יש 3-צביעה חוקית, שבה $u$ ו- $v$ נצבעים באותו צבע (נאמר אדום). בגרף החדש, $u'$ ו- $v'$ מחוברים רק לצמתים האלו, ולכן אם נצבע את $u'$ בצהוב ואת $v'$ בירוק, בוודאות לא נקבל סתירה שכן אין להם שכנים אחרים (לא תהיה קשת הצבועה באותו צבע משני צדיה כי הקשת היחידה החדשה היא בין הצמתים החדשים).

כעת, אם $<G'>\in 3COLOR$, אז $u',v'$ נצבעים בשני צבעים השונים מצבעו של $u$, אך שונים גם מצבעו של $v$, ושונים זה מזה. מכיוון שיש סה"כ שלושה צבעים, נקבל ש- $u$ ו- $v$ חייבים להיצבע באותו הצבע. אם נסיר מהגרף החדש את $u'$ ו- $v'$ נקבל בהכרח 3-צביעה בגרף המקורי, ובנוסף $u$ ו- $v$ צבועים בו באותו צבע. כלומר $<G,u,v>\in SAME-COLOR$, כדרוש.

56. נגדיר את השפה $2PATHS$ להיות שפת השלשות $<G,s,t>$ כך ש- $G$ הוא גרף מכוון, $s$ ו- $t$ הם צמתים ב- $G$, ויש ב- $G$ **שני מסלולים פשוטים** שונים (או יותר) מ- $s$ ל- $t$. (המסלולים השונים יכולים להכיל קשתות משותפות. הם לא חייבים להיות זרים זה לזה.)
הוכיחו: השפה $2PATHS$ היא **שפה NL- שלמה**.

**הדרכה:** הוכיחו שהיא שייכת ל- $NL$, והראו כי $PATH\leq_L 2PATH$.

**פתרון:**

נבנה מכונה לא דטרמיניסטית שמכריעה את $2PATH$:

1. נחש במקביל באופן לא דטרמיניסטי שני מסלולים מ- $s$ ל- $t$ באורך לכל היותר $n$. כאשר המסלולים יהיו שונים, עבור למצב שמסמן שהמסלולים שונים. אם הגעת ל- $s$ אבל המסלולים שווים, דחה. אחרת, קבל.

**נכונות:** ברור שאם קיימים שני מסלולים, הם ינוחשו במקביל ובסופו של דבר יזוהו כשונים, ולכן נקבל.

**סיבוכיות:** כל מה שצריך להחזיק הוא מונה של כמות הצמתים, והוא באורך $\log n$ לכל היותר.

כעת נראה ש- $PATH\leq_L 2PATH$.

קלט: $<G,s,t>$ כאשר $G$ הוא גרף מכוון, ו- $s$ ו- $t$ הם צמתים.

1. הוסף צומת $u$, והוסף את הקשתות $(s,u)$ ו- $(u,t)$.
2. החזר את $<G',s,t>$.

**נכונות:**

נניח ש- $<G,s,t>\in PATH$. אז בגרף החדש המסלול הישן קיים, ובנוסף קיים המסלול החדש $P=sut$. לכן קיימים שני מסלולים ולכן $<G',s,t>\in 2PATH$.

נניח ש- $<G,s,t>\notin PATH$. אז בגרף החדש המסלול הפשוט היחיד שנוסף הוא $P=sut$. בפרט לא קיימים שני מסלולים מ- $s$ ל- $t$ ולכן $<G',s,t>\notin 2PATH$, כדרוש.

**סיבוכיות:** לא משתמשים בסרט העבודה, ולכן סיבוכיות המקום היא $O(1)$ כלומר לוגריתמית, כדרוש.

לכן $2PATH \leq_L PATH$ שמצידה שייכת ל- $NL$, ולכן היא $NL$-שלמה.

57. האם הטענה הבאה נכונה? הוכיחו את תשובתכם!

אם $A$ **שייכת למחלקה BPP** ו- $B \leq_L A$ (יש רדוקציית מקום לוגריתמי של $B$ ל- $A$),

אז גם $B$ **שייכת למחלקה BPP**.

**פתרון:**

הטענה נכונה. אם $A$ שייכת למחלקה BPP, אז קיימת לה מכונה מתאימה $M$.

על פי הדיון בראש עמוד 351 בספר, רדוקציה לוגריתמית במקום רצה בזמן $O(n^2)$ כלומר פולינומי באורך הקלט.

כעת, נבנה מכונת $BPP$ ל- $B$:

קלט: מילה $w$.

1. הרץ את המכונה המחשבת רדוקציה לוגריתמית במקום מ- $B$ ל- $A$.
2. הרץ את $M$ על $f(w)$. אם היא קיבלה – קבל. אם היא דחתה – דחה.

**נכונות:**

נניח ש- $w \in B$. אז מכיוון ש- $f$ היא רדוקציית מיפוי, $f(w) \in A$, ולכן $M$ דוחה אותה בהסתברות הקטנה מ- $\frac{1}{3}$. כלומר המכונה תדחה את $w$ בהסתברות הקטנה מ- $\frac{1}{3}$, כדרוש.

כעת, אם $w \notin B$ אז גם $f(w) \notin A$, ולכן $M$ תקבל את $w$ בהסתברות הקטנה מ- $\frac{1}{3}$, כלומר המכונה החדשה תקבל את $w$ בהסתברות הקטנה מ- $\frac{1}{3}$.

**סיבוכיות:** כאמור, הרדוקציה רצה בזמן $O(n\log n)$, ולכן בפרט הפלט שלה, $f(w)$, הוא בגודל $O(n\log n)$. אך $M$ רצה בזמן פולינומי ביחס לאורך הקלט שלה, כלומר בזמן פולינומי ביחס ל- $O(n\log n)$, ובסך הכל נקבל זמן פולינומי, כדרוש.

58. תהי $M$ מכונת טיורינג **דטרמיניסטית**.

נגדיר את הפונקציה $f_M : \mathbf{N} \to \mathbf{N}$ הבאה ($0 \in \mathbf{N}$):

$f_M(n)$ הוא מספר הצעדים המקסימלי שמבצעת המכונה $M$ על מילת קלט כלשהי שאורכה $n$ עד לעצירה. אם $M$ לא עוצרת על אף מילת קלט שאורכה $n$, $f_M(n) = 0$.

א. האם הטענה הבאה נכונה? **הוכיחו** את תשובתכם.

אם $f_M$ היא פונקציה **ניתנת לחישוב**, אז השפה ש- $M$ מזהה היא שפה **כריעה**.

ב. הראו שייתכן שהשפה ש- $M$ מזהה היא שפה **כריעה**, ו- $f_M$ **איננה פונקציה ניתנת לחישוב**.

בשני הסעיפים מדובר על מכונה שלא בהכרח מכריעה את השפה שהיא מזהה. כלומר, ייתכנו קלטים שעליהם המכונה לא עוצרת.

**פתרון:**

א. הטענה נכונה. נבנה מכונה שתכריע את השפה $L(M)$.

קלט: מילה $w$.

1. חשב את $f_M(|w|)$ .
2. אם $f_M(|w|)=0$ , דחה.
3. אחרת, הרץ את $M$ על $w$ במשך $f_M(|w|)$ צעדים. אם היא עצרה עד אז – קבל אם היא קיבלה ודחה אם דחתה. אם היא לא עצרה – דחה.

**נכונות:**

נניח ש- $w\in L(M)$ . אז $M$ רצה לכל היותר $f_M(|w|)$ צעדים על $w$ , ולכן היא תתקבל בשלב 3. אם $w\notin L(M)$ , ייתכנו מספר מקרים; אם $M$ נכנסת ללולאה על כל קלט בגודל $|w|$, היא תידחה בשלב 2. אחרת, בשלב 3 נריץ את $M$ על $w$ מספר כלשהו של צעדים, ומכיוון שהיא לא תעצור עד אז, נדחה.

לחלופין, $M$ עוצרת על $w$ תוך מספר צעדים הקטן מ- $f_M(|w|)$ או שווה לו, ואז נדחה את $w$ בשלב 3. מש"ל.

ב. נגדיר את השפה הבאה $A=\{<M,w,k>|M\ accepts\ w\ in\ k\ steps\ or\ less\}$ . קל לבנות מכונת טיורינג שתכריע את $A$ , $N$ : בהינתן קלט $<M,w,k>$ , $N$ תריץ את $M$ על $w$ במשך $k$ צעדים. אם היא דחתה או לא עצרה, נדחה. אחרת – נקבל.

כעת, נניח ש- $f_N$ ניתנת לחישוב. נבנה מכונת טיורינג להכרעת $A_{TM}$ :

קלט: $<M,w>$ , כאשר $M$ היא מכונת טיורינג ו- $w$ היא מילה.

1. הרץ את $N(M,w,f_N(|w|))$ . אם $N$ קיבלה – קבל. אחרת – דחה.

**נכונות:** נניח ש- $w\in L(M)$ . ברור ש- $f_N(|w|)\geq k$ המתאים עבור $A$ (כי $N$ מסמלצת את $M$ , ולכן היא לא יכולה לעשות פחות צעדים משנדרש על ידי $M$). לכן $N$ חייבת לקבל את $w$ , כי $M$ תקבל את $w$ תוך $f_N(|w|)$ צעדים או פחות, ואז המכונה שלנו תקבל את $<M,w>$ כדרוש.

נניח ש- $w\notin L(M)$ . אז כל ערך שיהיה ל- $f_N(|w|)$ לא יגרום לכך ש- $N$ תקבל כי היא מקבלת רק אם $M$ מקבלת, ולכן גם המכונה שלנו תדחה.

כך הוכחנו ש- $A_{TM}$ כריעה, בסתירה לכך שהיא לא כריעה.

לכן $f_N$ אינה ניתנת לחישוב.

59. משתנה $A$ בדקדוק חסר הקשר $G$ נקרא **הכרחי** אם **יש** מילה $w$ בשפה שהדקדוק יוצר, כך שהמשתנה $A$ משתתף **בכל גזירה** של $w$ . (משתנה $B$ **איננו** הכרחי, אם **לכל** מילה $w$ בשפה שהדקדוק יוצר, **יש גזירה** שבה המשתנה $B$ לא משתתף).

נגדיר את השפה $NECESSARY_{CFG}$ להיות שפת כל הזוגות $<G,A>$ כך ש- $G$ הוא דקדוק חופשי הקשר ו- $A$ הוא משתנה הכרחי ב- $G$ .

הוכיחו: השפה $NECESSARY_{CFG}$ **איננה כריעה**.

**הדרכה:** אפשר להראות רדוקציה של שפה בלתי כריעה בדקדוקים חסרי הקשר.

**פתרון:** נראה רדוקציה מ- $\overline{ALL_{CFG}}$ (לא כריעה כי $ALL_{CFG}$ לא כריעה).

קלט: $<G>$ כאשר $G$ הוא דקדוק חופשי הקשר.

1. צור משתנה חדש $A$ , ואת הכללים הבאים:

$S\rightarrow A$ .

לכל $\sigma\in\Sigma$ , $A\rightarrow\sigma A$ .

. $A \to \varepsilon$

2. החזר $<G', A>$.

**נכונות:**

נניח ש- $<G> \notin ALL_{CFG}$. אז קיימת מילה ש- $S$ לא גוזרת. אך $A$ גוזר את כל המילים, ולכן על מנת לגזור את המילה שאינה ב- $L(G)$, חייבים להשתמש ב- $A$, כלומר הוא הכרחי. כלומר $<G', A> \in NECESSARY_{CFG}$.

נניח ש- $<G> \in ALL_{CFG}$. אז כל מילה ניתן לגזור בדקדוק החדש על ידי הכללים הישנים, ואין צורך להשתמש ב- $A$, לכן הוא לא הכרחי. כלומר $<G', A> \notin NECESSARY_{CFG}$.
מש"ל.

60. **דרגת הכניסה** של צומת $v$ בגרף **מכוון** היא מספר הקשתות שנכנסות לצומת $v$ בגרף.
**דרגת היציאה** של צומת $v$ בגרף **מכוון** היא מספר הקשתות שיוצאות מהצומת $v$ בגרף.
נעיין בבעיה הבאה:

**הקלט**: גרף **לא מכוון** $G = (V, E)$ וקבוצת צמתים $C$ ($C \subseteq V$).

**השאלה**: האם אפשר לקבוע כיוון לכל קשת של הגרף (להפוך אותו מגרף לא מכוון לגרף מכוון) באופן שלכל צומת $u$ של $C$, או שדרגת הכניסה של $u$ תהיה 0, או שדרגת היציאה של $u$ תהיה 0, ולכל צומת $w$ ב- $V - C$, דרגת הכניסה של $w$ תהיה לפחות 1.

הוכיחו: הבעיה הזו היא **NP - שלמה**.

**הדרכה**: הוכיחו שהבעיה שייכת ל- NP, והראו רדוקציה בזמן פולינומיאלי של $3SAT$.

**פתרון:**

מאמת לבעיה יקבל את הגרף ואת קבוצת הקשתות המכוונות, ויבדוק את דרגות כל הצמתים תחת הכיוונים הנ"ל.

כעת נראה רדוקציה מ- $3SAT$. נניח שקבוצת המשתנים היא $\{x_1, ..., x_n\}$.

קלט: $<\phi>$ כאשר $\phi$ היא נוסחה ב- $3cnf$.

1. צור גרף חדש $G$. עבור הפסוקית ה- $i$ צור צומת $c_i$.
2. לכל משתנה $x_i$ צור צומת $x_i$ ו- $\overline{x_i}$. צור את הקשת $\{x_i, \overline{x_i}\}$.
3. אם הליטרל $x_i$ מופיע בפסוקית $c_j$, צור את הקשת $\{x_i, c_j\}$.
4. אם הליטרל $\overline{x_i}$ מופיע בפסוקית $c_j$, צור את הקשת $\{\overline{x_i}, c_j\}$.
5. החזר $<G, \{x_1, \overline{x_1}, x_2, \overline{x_2}, ..., x_n, \overline{x_n}\}>$.

**נכונות:**

נניח ש- $<\phi> \in 3SAT$. אז יש השמה מספקת ל- $\phi$. נראה איך לתת כיוונים מתאימים ל- $G$.
אם $x_i = 1$, ניתן לקשת $\{x_i, \overline{x_i}\}$ את הכיוון $(x_i, \overline{x_i})$. ולכל פסוקית $j$ בה מופיע $x_i$ ניתן את הכיוון $(x_i, c_j)$. לכל פסוקית $j$ בה מופיע $\overline{x_i}$ ניתן את הכיוון $(c_j, \overline{x_i})$. אם $x_i = 0$ ניתן את הכיוון $(\overline{x_i}, x_i)$, ובאופן זהה ניתן כיוונים לפסוקיות, רק הפוך.

מכיוון שכל משתנה מקבל ערך יחיד, מההשמה הנ"ל ברור שמצומת $x_i$ או שרק יוצאות קשתות (ל- $\overline{x_i}$ ולפסוקיות), או שרק נכנסות קשתות (מאותם צמתים). בנוסף, אם פסוקית $c_j$ היא אמת, אזי יש ליטרל שהוא אמת אצלה. אם הוא $x_i$ אזי $x_i = 1$, ולכן יוצאת קשת מ- $x_i$ לפסוקית $c_j$, ודרגת הכניסה גדולה מאפס. אם הוא $\overline{x_i}$, אז $x_i = 0$, ולכן יוצאת קשת מ- $\overline{x_i}$

ל- $c_j$, ודרגת הכניסה גדולה מאפס. מכיוון שכל הפסוקיות הן אמת (כי הנוסחה ספיקה), נקבל שלכל פסוקית נכנסת קשת אחת לפחות. מש"ל.

כעת, נניח ש- $<G,\{x_1,\overline{x_1},x_2,\overline{x_2},...,x_n,\overline{x_n}\}>$ נמצא בשפה. נראה כיצד לייצר השמה מספקת.

לכל משתנה $x_i$, אם הקשת היא $(x_i,\overline{x_i})$, $x_i=1$. אחרת, $x_i=0$.

ניקח פסוקית $c_j$. כל קשת נכנסת/יוצאת מ- $c_j$ מייצגת את הליטרל שמופיע בפסוקית. מכיוון שיש לה קשת נכנסת אחת לפחות, ייתכן שהקשת היא $(x_i,c_j)$ או $(\overline{x_i},c_j)$. אם הקשת היא $(x_i,c_j)$ הרי שמ- $x_i$ רק יוצאות קשתות, כלומר הקשת היא $(x_i,\overline{x_i})$ ולכן נקבל ש- $x_i=1$. אבל כאמור ב- $c_j$ מופיע $x_i$ ולכן היא אמת. אם הקשת היא $(\overline{x_i},c_j)$ הרי שמ- $\overline{x_i}$ רק יוצאות קשתות ולכן הקשת היא $(\overline{x_i},x_i)$, ועל כן שמנו $x_i=0$. כאמור ב- $c_j$ מופיע $\overline{x_i}$ ולכן היא אמת. מש"ל.

61. **תזכורת**: פסוק בתחשיב הפסוקים (נוסחה בוליאנית) ב-CNF הוא קבוצה של **פסוקיות** כך שבין הפסוקיות יש הקשר $\wedge$, וכל פסוקית היא קבוצה של **ליטרלים** שביניהם יש הקשר $\vee$.
ליטרל הוא **משתנה** או **שלילה של משתנה**.
ליטרל נקרא **חיובי** אם הוא משתנה; ליטרל נקרא **שלילי** אם הוא שלילה של משתנה.
הוכח שהשפה $POSITIVE-CNF$ הבאה **שייכת למחלקה P**.

$POSITIVE-CNF$ מורכבת מכל המילים $<\phi>$ עבורן $\phi$ הוא פסיק **ספיק** בתחשיב הפסוקים ב-CNF, **ובכל** פסוקית של $\phi$ יש **לכל היותר ליטרל שלילי אחד**. (או שאין ליטרלים שליליים בכלל, או שיש ליטרל שלילי יחיד).

דוגמה למילה בשפה: $(x_1 \vee x_2)\wedge(\neg x_2 \vee x_3)\wedge(\neg x_3)$.

**פתרון:**

קלט: נוסחה $<\phi>$.

1. עבור כל משתנה $x_i$ השם $x_i=1$.
2. עבור על הנוסחה, ובדוק האם קיימת פסוקית שבה הליטרל היחיד הוא $\overline{x_i}$. אם כן, השם $x_i=0$. הסר את הליטרל $x_i$ מכל המקומות בהם הוא מופיע, וחזור על הפעולה.
3. בדוק האם הנוסחה, עם ההצבה הנוכחית, ערכה אמת. אם כן – קבל. אחרת – דחה.

**נכונות:**

נניח ש- $<\phi>\in POSITIVE-CNF$. כל ליטרל המופיע לבדו בפסוקית חייב שהמשתנה שלו יקבל את הערך 0, אחרת הפסוקית לא תקבל ערך "אמת" וכך כל הפסוק. כל מופע של הליטרל החיובי לא תורם לאף פסוקית ולכן ניתן להסיר אותו ולהמשיך בתהליך: אם נשאר רק ליטרל שלילי, המשתנה חייב לקבל את הערך 0, אחרת הנוסחה לא תהיה ספיקה. לאחר שסיימנו עם הלולאה הזו, כל הפסוקיות שנשארו מכילות בוודאות ליטרלים שלא הושם להם אפס, והם חיוביים. השמת הערך 1 תספק את כל הפסוקיות שנותרו. ייתכן שהסרנו פסוקית שלמה שלא תוכל להיות מסופקת – לדוגמה אם היא מכילה רק משתנים שהושם להם 0 (כי חייבים). את הסתירה הזו נגלה בבדיקה בשלב 3, ונדחה.

**זמן ריצה:**

היות שמספר הפסוקיות בהן ליטרל שלילי יחיד הוא לינארי באורך הקלט, המעבר על כל הנוסחה והסרת משתנים כאלו מתבצעים מספר פולינומי של פעמים, ואורכים בעצמם זמן פולינומי. לכן שורה 2 אורכת זמן פולינומי, ושורה 3 גם היא ניתנת לביצוע בזמן פולינומי.

62. הוכיחו שהשפה $OVERLAP_{DFA}$ הבאה היא **NL-שלמה**: $OVERLAP_{DFA}$ מורכבת מכל הזוגות $< D_1, D_2 >$ של אוטומטים סופיים דטרמיניסטיים שהשפות שהם מזהים **אינן** זרות זו לזו, כלומר $L(D_1) \cap L(D_2) \neq \varnothing$.

**הדרכה:** הוכיחו שהבעיה שייכת ל- NL, והראו כי $\overline{E_{DFA}} \leq_L OVERLAP_{DFA}$.

**פתרון:**

תחילה נבנה מכונה לא דטרמיניסטית המכריעה את $OVERLAP_{DFA}$ במקום לוגריתמי.

קלט: $< D_1, D_2 >$

1. נחש מילה $w$ באורך לכל היותר $n^2$, כאשר $n$ הוא מספר המצבים המקסימלי של $D_1, D_2$.
2. סמלץ את האוטומטים על $w$, וקבל אם ורק אם שניהם מקבלים אותה.

**נכונות:** תחילה, מילה שנמצאת בחיתוך תתקבל על ידי אוטומט מכפלה, בעל $n^2$ מצבים. אך אוטומט בעל $n^2$ מצבים, שפתו אינו ריקה אם ורק אם הוא מקבל מילה באורך לכל היותר $n^2$. לכן אם יש מילה משותפת, יש אחת כזו באורך לכל היותר $n^2$, והמכונה תמצא אותה.

**סיבוכיות מקום:** חישוב $n^2$ הוא כמובן לוגריתמי ($\log(n^2) = 2\log n = O(\log n)$).

כעת, נראה רדוקציה $\overline{E_{DFA}} \leq_L OVERLAP_{DFA}$.

קלט: $< D >$ כאשר $D$ הוא אוטומט סופי דטרמיניסטי.

1. בנה אוטומט סופי דטרמיניסטי $D'$ המקבל את $\Sigma^*$.
2. החזר $< D, D' >$.

**נכונות:**

נניח ש- $< D > \notin E_{DFA}$. אז קיימת מילה $w$ שמתקבלת על ידי $D$, ובפרט היא מתקבלת גם על ידי $D'$, ולכן יש ביניהן חיתוך שאינו ריק. כלומר $< D, D' > \in OVERLAP_{DFA}$.

נניח ש- $< D > \in E_{DFA}$. אז $L(D) = \varnothing$ ובפרט $L(D) \cap L(D') = \varnothing$, כלומר $< D, D' > \notin OVERLAP_{DFA}$.

**סיבוכיות מקום:** בניית האוטומט יכולה להיעשות במצבים בלבד ולא צריכה שימוש בסרט, ולכן סיבוכיות המקום תהיה $O(1)$ ובפרט לוגריתמית.

כעת, קיימת גם רדוקציה $PATH \leq_L \overline{E_{DFA}}$, אותה תיארנו בתרגיל מספר 16. מטרנזיטיביות הרדוקציה הלוגריתמית במקום (הוכחה במדריך), נקבל ש- $PATH \leq_L OVERLAP_{DFA}$, ולכן $OVERLAP_{DFA}$ NL-שלמה.

63. תהי $w$ מילה לא ריקה. נסמן ב- $MOVE-FIRST(w)$ את קבוצת כל המילים שמתקבלות מ- $w$ על ידי העברת הסמל הראשון של $w$ למקום כלשהו במילה.

**דוגמאות:** אם $w = 1000$, אז $MOVE-FIRST(w) = \{1000,0100,0010,0001\}$;

אם $w = 111$, אז $MOVE-FIRST(w) = \{111\}$.

תהי $A$ שפה. $MOVE-FIRST(A) = \bigcup_{w \in A} MOVE-FIRST(w)$.

נתון שהשפה $A$ שייכת **למחלקה RP**. האם **בהכרח** $MOVE-FIRST(A)$ שייכת **למחלקה RP**? הוכיחו!

**פתרון:**

נניח ש- $M$ היא מכונת RP ל- $A$
נבנה מכונת RP עבור $MOVE-FIRST(A)$.
קלט: מילה $w$.
1. עבור $i=1,\ldots,n$:
   1.1. הזז את התו ה- $i$ להיות השמאלי ביותר.
   1.2. הרץ את $M$ על המילה הנוכחית. אם היא קיבלה – קבל.
   1.3. החזר את המצב לקדמותו (המילה המקורית).
2. דחה.

**נכונות:**

נניח ש- $w \notin MOVE-FIRST(A)$. אז כל מילה שמתקבלת בשלב 1.1 לא שייכת ל- $A$. מכיוון ש- $M$ היא מכונת RP, היא תדחה כל אחת מהמילים בהסתברות 1, ולכן נדחה את הקלט בהסתברות 1.

נניח ש- $w \in MOVE-FIRST(A)$. אז קיימת לפחות מילה אחת בשלב 1.1 ששייכת ל- $A$. ההסתברות שהמכונה החדשה תדחה את המילה קטנה או שווה להסתברות שיש רק מילה אחת בשלב 1.1 המתאימה, והיא תדחה אותה (כי יש "פחות אפשרויות לקבל", ואם זה לא בשפה דוחים גם ככה). אך ההסתברות שנדחה קלט שהוא ב- $A$ קטנה מחצי. לכן נקבל שההסתברות שהמכונה תדחה קלט שבשפה קטנה מחצי, כדרוש.

**זמן ריצה:**

מריצים $n$ פעמים אלגוריתם פולינומי ולכן זמן הריצה פולינומי.

64. נניח שיוכח שהמחלקה $\mathrm{NP} \cap \mathrm{coNP}$ **שונה** מן המחלקה P.
ביחס לכל אחת מן הטענות הבאות, עליכם לקבוע האם **בהכרח** היא תהיה נכונה, ולנמק בקצרה את קביעתכם.

א. **לכל** שפה NP-שלמה $L$, השפה המשלימה ל- $L$ לא שייכת ל- $P$.
ב. **יש** שפות ב- P שלא שייכות ל- $\mathrm{NP} \cap \mathrm{coNP}$.
ג. **לכל** שפה NP-שלמה $L$, גם השפה המשלימה ל- $L$ היא שפה NP-שלמה.
ד. **יש** שפה $L$ ששייכת למחלקה $\mathrm{NP} \cap \mathrm{coNP}$, והמשלימה שלה לא שייכת למחלקה זו.

**פתרון:**

א. נניח בשלילה שקיימת שפה NP-שלמה $L$ שמשלימתה שייכת ל- $P$. אך מכונה דטרמיניסטית פולינומית המקבלת את $L$ יכולה לקבל את $\overline{L}$ בזמן פולינומי על ידי החלפת המצב המקבל והדוחה, ולכן גם $L \in P$. אך היות ש- $L$ היא NP-שלמה נקבל ש-P=NP. יתקיים גם P=coNP מנימוק סימטרי (כל שפה ב-coNP המשלימה שלה נמצאת ב-P ולכן גם היא ב-P). ואז נקבל ש- $\mathrm{NP} \cap \mathrm{coNP} = \mathrm{NP} = \mathrm{P}$, בסתירה לנתון.

ב. כל שפה ב-P שייכת גם ל-NP, כי מכונה דטרמיניסטית היא בפרט מכונה לא דטרמיניסטית. אך גם משלימתה שייכת ל-P (הסברנו בסעיף הקודם), ולכן גם משלימתה שייכת ל-NP, כלומר היא שייכת ל-coNP. אם כן, כל שפה ב-P שייכת לשתי השפות, כלומר שייכת ל- $\mathrm{NP} \cap \mathrm{coNP}$, ולכן הטענה לא נכונה.

65. סטודנט בקורס "מבוא לתורת החישוביות והסיבוכיות" קנה סימולטור של מכונות טיורינג.
הסימולטור מקבל תיאור של מכונת טיורינג (קבוצת המצבים, אלפבית הקלט, אלפבית הסרט, פונקציית המעברים, המצב ההתחלתי, המצב המקבל והמצב הדוחה) וכן מחרוזת קלט למכונה.
הסימולטור מבצע סימולציה של ריצת המכונה הנתונה על הקלטים הנתונים.
לאחר כמה זמן הסטודנט הרגיש שהתוצאות המתקבלות על ידי הסימולטור שגויות.

א. בדיקת הסימולטור העלתה שמדי פעם, כאשר הראש הקורא-כותב נע ימינה (לפי פונקציית המעברים של המכונה), לא מתבצעת ההדפסה, והסמל בריבוע שממנו נע הראש הקורא כותב לא משתנה. (כאשר מבצעים צעד מן הצורה $\delta(q,a)=(p,b,R)$ לא מתבצעת ההחלפה של $a$ ב- $b$.) התופעה הזו לא קורה תמיד ואפילו לא ברוב המקרים, אך קורה מפעם לפעם.

ב. בדיקת הסימולטור העלתה שמדי פעם, כאשר הראש הקורא-כותב אמור לנוע ימינה (לפי פונקצית המעברים של המכונה), התנועה ימינה לא מתבצעת, והראש הקורא-כותב נשאר על הריבוע שבו הוא היה. התופעה הזו לא קורה תמיד ואפילו לא ברוב המקרים, אך קורה מפעם לפעם.

הסטודנט פנה לשני מומחים בתחום החישוביות.
פרופסור פסימוס טוען שיש שפות שהן מזוהות-טיורינג, ואי אפשר יהיה לזהות אותן בעזרת הסימולטור המקולקל.
פרופסור אופטימוס טוען שכל שפה מזוהה טיורינג אפשר יהיה לזהות גם בעזרת הסימולטור המקולקל. ייתכן שיידרשו שינויים במכונה המזהה, כך שהיא תתאים לתקלה של הסימולטור, אך עדיין אפשר לזהות בעזרת הסימולטור כל שפה מזוהה טיורינג.
עבור כל אחד מהמקרים א' וב', קבעו מי משני המומחים צודק. **הוכיחו** את תשובתכם.

**פתרון:**

א. נכפיל את אלפבית הסרט להיות $\Sigma \cup \Sigma'$. בכל פעם שזזים ימינה, כותבים לסרט את התו שמתכוונים לכתוב רק מתויג, כך ששני תיוגים הם התו המקורי. חוזרים אחורה ובודקים האם התו מתויג כמו שרצינו, ואחרת ממשיכים בלולאה עד שהוא משתנה. היות שהתקלה לא קורה תמיד, בשלב כלשהו התו יתחלף והמכונה תרוץ כרגיל. בכל שאר המצבים, היא תתייחס לתווים מתויגים כאילו אין בהם תיוג.

ב. בכל הליכה ימינה נחליף את התו שאמור להיות בתו מתויג. כל עוד עדיין עומדים על תו לא מתויג, נמשיך להחליפו בתו שאמור להיות תוך תיוג. אם לא, נסמן את התו הנוכחי בכוכבית, נחזור שמאלה, ונוריד את התג מהתו הנוכחי, עד שירד, ואז נגיע לכוכבית. מהכוכבית, אם השלב הבא הוא הליכה שמאלה, נוריד את הכוכבית ונשים את התו החדש. אם ימינה, נשים את התו החדש עם תיוג, ואז יתבצע אותו תהליך.

66. **תזכורת:** השפה $SUBSET-SUM$ היא **NP - שלמה**.
נגדיר את השפה $PARTITION$ :
$PARTITION$ מורכבת מכל הקבוצות $S=\{x_1,...,x_n\}$ של מספרים טבעיים כך שקיימת תת קבוצה $S$ $\{y_1,...,y_k\}\subseteq S$, ומתקיים $\sum_{i=1}^{k} y_i = \frac{\sum_{i=1}^{n} x_i}{2}$.

(למשל, $\{1,2,3\}$ שייכת ל- $PARTITION$ ; $\{1,2,3,8\}$ לא שייכת ל- $PARTITION$.)
מותרים מופעים כפולים של מספרים ב- $S$ ( $S$ היא רב-קבוצה – multiset).

א. הציגו רדוקציה בזמן פולינומיאלי של $SUBSET-SUM$ ל- $PARTITION$ (הראו: $SUBSET-SUM \leq_p PARTITION$).
תארו את הרדוקציה, והוכיחו שהיא תקפה ושהיא ניתנת לחישוב בזמן פולינומיאלי.

ב. הוכיחו: אם יש אלגוריתם בעל זמן ריצה פולינומיאלי להכרעת השייכות ל- $PARTITION$, אז אפשר לבנות **בעזרתו** אלגוריתם **בעל זמן ריצה פולינומיאלי** לבעיה הבאה (של **מציאת** התת-קבוצה):

הקלט: קבוצה $S$ של מספרים טבעיים, $S=\{x_1,x_2,...,x_k\}$.

הפלט: תת-קבוצה של $S$, $\{y_1,...,y_n\} \subseteq S$, כך ש- $\sum_{i=1}^{n} y_i = \frac{\sum_{i=1}^{k} x_i}{2}$, אם תת-קבוצה כזו קיימת. אם לא קיימת תת-קבוצה כזו, מוחזר "לא".

(האלגוריתם מקבל כקלט קבוצת מספרים $S$. אם אין ל- $S$ תת-קבוצה שסכום המספרים שלה שווה למחצית סכום המספרים ב- $S$, האלגוריתם יחזיר "לא". אם יש ל- $S$ תת קבוצה כזו, האלגוריתם יחזיר את **רשימת האיברים של תת קבוצה כזו**.)

**הדרכה:** האלגוריתם יקרא מספר פולינומיאלי של פעמים לאלגוריתם ההכרעה $PARTITION$, כל פעם עם קלט מעט שונה. (אלגוריתם ההכרעה עונה רק "כן" או "לא" על הקלט שלו.)

א. **פתרון:**

קלט: $< S,t >$ כאשר $S$ היא קבוצת המספרים הטבעיים, ו- $t$ הוא מספר.

1. סכום את כל המספרים ב- $S$ לכדי מספר $y$.
2. אם $y \le 2t$, הוצא כפלט $< S \cup \{2t-y\} >$.
3. אם $y > 2t$, הוצא לפלט $< S \cup \{y-2t\} >$.

**נכונות:**

נניח ש- $< S,t > \in SUBSET-SUM$. אז קיימת קבוצה $S'$ החלקית ל- $S$, וסכומה הוא $t$.

אם $y \le 2t$. אז בקבוצה החדשה, אותה קבוצה $S'$, סכומה הוא $t$, וסכום כל הקבוצה החדשה הוא $y+2t-y=2t$, כלומר סכומה של $S'$ הוא בדיוק מחצית סכומה של כל הקבוצה, כדרוש.

אם $y > 2t$, אז בקבוצה החדשה, $S''=S' \cup \{y-2t\}$ מקיימת שסכומה הוא $t+y-2t=y-t$. אך סכום כל הקבוצה הוא $y+y-2y=2y-2t$, כלומר סכום $S''$ הוא בדיוק מחצית סכום $S'$, כדרוש.

כעת, נניח שהפלט של הרדוקציה נמצא בשפה. אז יש קבוצה $S$ שסכומה שווה בדיוק לחצי.

אם $y \le 2t$, סכומה של הקבוצה הוא $t$. אם האיבר החדש נמצא בקבוצה $S$, שאר איברי הקבוצה קיימים בקבוצה שלפני פעולת הרדוקציה, סכומם גם הוא $t$. אחרת, הקבוצה $S$ מתאימה.

אם $y > 2t$, סכומה של הקבוצה הוא $2y-2t$, ולכן סכום הקבוצה שנבחרת הוא $y-t$. אם האיבר החדש נמצא בקבוצה הנבחרת, נסיר אותו ונקבל קבוצה שסכומה $y-t-(y-2t)=t$ כדרוש. אם האיבר החדש לא נמצא בקבוצה הנבחרת, ניקח את שאר איברי הקבוצה (לא כולל החדש). ערכם הוא $y-(y-t)=t$, כדרוש.

**סיבוכיות זמן:** סכימת המספרים היא סכימה של $n$ מספרים לכל היותר, כל אחד בזמן לינארי לכל היותר, לכן היא אורכת זמן פולינומי. השוואה ויצירת המספר החדש גם הן דורשות רק פעולות אריתמטיות פשוטות ולכן אורכות זמן פולינומי.

ב. **???**

67. הוכיחו שהשפות הבאות שייכות למחלקה P:

א. $A_{NFA}$.

ב. שפת הזוגות $<G,w>$ כך ש- $G$ הוא דקדוק חסר הקשר בצורה הנורמלית של חומסקי, $w\in L(G)$, ויש יותר מאופן אחד לגזור את $w$ בדקדוק $G$.

**פתרון:**

א. נבנה מכונת טיורינג עם שני סרטים שתכריע את $A_{NFA}$ בזמן פולינומי.

קלט: $<D,w>$ כאשר $D$ הוא אוטומט סופי לא דטרמיניסטי ו- $w$ היא מילה.

1. המר את $D$ לאוטומט סופי לא דטרמיניסטי **ללא** מסעי- $\varepsilon$.
2. כתוב את המצב ההתחלתי בסרט השני.
3. קרא את הקלט מימין לשמאל. לכל תו:
   1. לכל מצב בסרט השני:

3.1.1. בדוק לאילו מצבים ניתן להגיע מהמצב הזה עם האות שנקראת.

3.1.2. כתוב את כל המצבים האלו בהמשך.

   2. בטל כפילויות בסרט השני.
4. אם בסרט השני מופיע מצב מקבל – קבל. אחרת – דחה.

**נכונות:** נכונות האלגוריתם ברורה; מכל מצב בודקים לאילו מצבים ניתן להגיע בקריאת הקלט, ולבסוף אם מגיעים למצב מקבל – מקבלים.

**זמן ריצה:** הורדת מסעי- $\varepsilon$ מאוטומט אורכת זמן פולינומיאלי. מעבר לכך, בכל איטרציה של שלב 3.1, עוברים על לכל היותר $p(n)$ מצבים ($p(n)$ הוא פולינום שכן ייתכן שהסר המסעי- $\varepsilon$ הגדילה את מספר המצבים), אך אותה מבצעים $O(n)$ פעמים, ולכן זמן הריצה הכולל הוא פולינומי.

ב. ??

68. נגדיר את השפה $CF_{TM}$ להיות שפת התיאורים של מכונות טיורינג, $<M>$, המקיימות ש- $L(M)$ היא שפה חופשית הקשר.

א. הציגו רדוקציית מיפוי של $A_{TM}$ ל- $CF_{TM}$ (הראו: $A_{TM}\le_m CF_{TM}$).

ב. הציגו רדוקציית מיפוי של $A_{TM}$ ל- $\overline{CF_{TM}}$ (הראו: $A_{TM}\le_m \overline{CF_{TM}}$).

ג. הסיקו: $CF_{TM}$ ו- $\overline{CF_{TM}}$ **אינן מזוהות טיורינג**.

**פתרון:**

א. קלט: $<M,w>$ כאשר $M$ היא מכונת טיורינג ו- $w$ היא מילה.

1. בנה את $M'$ הבאה.
   קלט: $x$, מילה.
   1. הרץ את $M$ על $w$ במשך $|x|$ צעדים. אם $M$ קיבלה, קבל.

      אחרת, קבל אם ורק אם $x=a^nb^nc^n$ עבור $n$ כלשהו.
2. החזר את $<M'>$.

**נכונות:**

נניח ש- $<M,w>\in A_{TM}$. אז $M$ מקבלת את $w$ לאחר $m$ צעדים. לכן השפה שתקבל $M'$ מכילה את כל המילים באורך לפחות $m$, ואת כל המילים מהצורה $a^nb^nc^n$, עבור $n<m$. לכן משלימתה סופית, ולכן היא רגולרית ובפרט חופשית הקשר. כלומר $<M'>\in CF_{TM}$.

כעת, אם $<M,w>\notin A_{TM}$, אז $M'$ תקבל רק מילים מהצורה $a^nb^nc^n$, כלומר $L(M')=\{a^nb^nc^n \mid n\geq 0\}$, וזוהי כמובן שפה שאיננה חופשית הקשר, כלומר $<M'>\notin CF_{TM}$, כדרוש.

ב. קלט: $<M,w>$ כאשר $M$ היא מכונת טיורינג ו-$w$ מילה.

1. בנה את המכונה $M'$ הבאה:
קלט: $x$, מילה.
   1. הרץ את $M$ על $w$.
   אם $M$ קיבלה – קבל אם ורק אם $x=a^nb^nc^n$.
2. החזור את $<M'>$.

**נכונות:**

נניח ש-$<M,w>\in A_{TM}$. אז $M$ עוצרת ומקבלת את $w$, ולכן $L(M')=\{a^nb^nc^n \mid n\geq 0\}$ ולכן $<M'>\notin CF_{TM}$ כדרוש.

נניח ש-$<M,w>\notin A_{TM}$. אז $L(M')=\varnothing$ וזו שפה חופשית הקשר ולכן $<M'>\in CF_{TM}$ מש"ל.

ג. כמו ב-26ג.

69. לכל שפה $D$ מעל אלפבית נתון $\Sigma$ נגדיר את השפה $DROP-MIDDLE(D)$ הבאה:

$DROP-MIDDLE(D)=\{w\in\Sigma^* \mid \exists u,v\in\Sigma^*,\sigma\in\Sigma: w=uv,|u|=|v|,u\sigma v\in D\}$.

**דוגמה:** אם 10011 שייכת ל-$D$, אז 1011 שייכת ל-$DROP-MIDDLE(D)$.

**דוגמה לשפה:** אם $D$ היא שפת המילים מעל האלפבית $\{0,1\}$ שאורכן אי-זוגי והאות האמצעית שלהן היא 0, אז $DROP-MIDDLE(D)$ היא שפת המילים מעל האלפבית $\{0,1\}$ שאורכן זוגי.

א. הוכיחו: אם $D$ שייכת ל-$PSPACE$ אז גם $DROP-MIDDLE(D)$ שייכת ל-$PSPACE$.

ב. תהי $D$ שפה PSPACE-שלמה.
הראו **שלא בהכרח** $DROP-MIDDLE(D)$ היא שפה PSPACE-שלמה.
בתשובתכם אתם יכולים להיעזר בטענה הבאה:
אם $D$ היא שפה $PSPACE$-שלמה, אז גם השפה $\{ww \mid w\in D\}$ היא שפה $PSPACE$-שלמה.

ג. הוכיחו: אם $D\in BPP$ אז $DROP-MIDDLE(D)\in BPP$.

**פתרון:**

א. אם $D$ שייכת ל-PSPACE, קיימת לה מכונת טיורינג דטרמיניסטית $M$ הרצה במקום פולינומי.
נבנה את המכונה $M'$ שתכריע את $DROP-MIDDLE(D)$ במקום פולינומי.
קלט: מילה $w$.

1. אם המילה באורך אי זוגי – דחה.
2. לכל $\sigma\in\Sigma$, הוסף את $\sigma$ כאות האמצעית במילה, והרץ את $M$ על המילה שהתקבלה. אם היא קיבלה – קבל. אחרת, הסר את $\sigma$ והמשך בלולאה.
3. דחה.

**נכונות:** אם $w \in DROP-MIDDLE(D)$, אז קיימת אות כך שאם מכניסים אותה אל אמצע המילה, מקבלים מילה ב-$D$, ולכן $M'$ תקבל את המילה. אם קיימת אות כזו, אז בהכרח המילה בשפה.

**סיבוכיות:** כל ריצה של $M$ רצה במקום פולינומי ביחס ל-$|w|+1$, כלומר פולינומי ביחס ל-$|w|$. היות שניתן להשתמש שוב ושוב באותו מקום, נקבל שסיבוכיות המקום הכוללת היא פולינומית באורך הקלט, כדרוש.

ב. תהי $D$ שפה PSPACE-שלמה. על פי הטענה, גם $DD=\{ww \mid w \in D\}$ היא PSPACE-שלמה. אך כל המילים בשפה זו זוגיות, ולכן $DROP-MIDDLE(DD)=\varnothing$, והיא לא PSPACE-שלמה כי אין רדוקצית מיפוי ממנה לאף שפה פרט לעצמה.

ג. $D \in BPP$ ולכן על פי למת ההגברה, קיימת מכונת טיורינג דטרמיניסטית מסוג BPP $M$, שהשגיאה שלה קטנה כרצוננו, נאמר שהשגיאה היא $k$. נבנה מכונת BPP ל-$M'$.

קלט: מילה $w$.

1. אם $|w|$ אי זוגי, דחה.
2. לכל $\sigma \in \Sigma$, הוסף את $\sigma$ לאמצע המילה $w$, והרץ את $M$ על $w$. אם $M$ קיבלה, קבל. אם היא דחתה, הסר את $\sigma$ והמשך.
3. דחה.

**נכונות:** נניח ש-$w \in DROP-MIDDLE(D)$. נניח ש-$i$ מהמילים שנבדקות שייכות ל-$D$. ההסתברות שהמכונה בכל זאת תדחה שווה ל-$k^i(1-k)^{|\Sigma|-i}$. היות שאנו מניחים ש-$k<\frac{1}{2}$, ההסתברות הזו היא לכל היותר $(1-k)^{|\Sigma|}$. אם נניח $|\Sigma|>1$, אז אם $k<1-\frac{1}{\sqrt{2}}$ נקבל שהביטוי הנ"ל קטן מ-$\frac{1}{2}$. ואז ניתן להשתמש בלמת ההגברה על מנת להשיג שגיאה של שליש.

נניח ש-$w \notin DROP-MIDDLE(D)$. אז כל אחת מהבדיקות צריכה להיכשל. ההסתברות שאכן כך יקרה היא $(1-k)^{|\Sigma|}$, וכאמור ניתן לדאוג שההסתברות הזו תהיה קטנה מחצי.

כעת, השגיאה קטנה מחצי ולכן על פי למת ההגברה, ניתן לדאוג שהיא תהיה קטנה משליש, ולכן השפה תהיה ב-BPP.

**סיבוכיות זמן:**

70. האם המחלקה $BPP$ סגורה לאיחוד? הוכיחו את תשובתכם!

**פתרון:**

התשובה חיובית. נניח ש-$A_1, A_2 \in BPP$. אז קיימות מכונות $M_1$ ו-$M_2$ המכריעות את השפות בהתאמה, עם שגיאה הסתברותית $k$.

נבנה את המכונה הבאה שתכריע את $A_1 \cup A_2$.

קלט: מילה $w$.

1. הרץ את $M_1$ על $w$. אם היא קיבלה – קבל.
2. הרץ את $M_2$ על $w$. אם היא קיבלה – קבל. אם היא דחתה – דחה.

**נכונות:**

אם $w \in A_1 \cup A_2$, ההסתברות שנדחה שווה להסתברות ש-$M_1$ תדחה וש-$M_2$ תדחה. אך יש שלושה מקרים: $w \in A_1 \oplus A_2$, ואז ההסתברות ששתיהן ידחו היא $k\cdot(1-k)$, או

$w \in A_1 \cap A_2$ ואז ההסתברות לדחייה היא $k^2$. אך היות ש- $k < \frac{1}{2}$ נקבל שההסתברות תהיה לכל היותר $k - k^2$.

אם $w \notin A_1 \cup A_2$, ההסתברות שנדחה אותו היא ההסתברות שגם $M_1$ תצדוק וגם $M_2$, כלומר $(1-k)^2 = 1 - 2k + k^2$, וההסתברות שנקבל שלא לצורך היא $2k - k^2$.

אם נבחר $k = \frac{1}{3}$, אז ההסתברות לדחייה שלא לצורך תהיה $\frac{1}{9}$, וההסתברות לקבלה שלא לצורך היא $\frac{5}{9} = \frac{2}{3} - \frac{1}{9}$, וזה חסם לא מספיק. נבחר $k = \frac{1}{9}$. ואז ההסתברות לדחייה שלא לצורך היא $\frac{1}{81}$, וההסתברות לקבלה שלא לצורך תהיה $\frac{1}{3} > \frac{17}{81} = \frac{2}{9} - \frac{1}{81}$. כלומר השגיאה הדו צדדית קטנה מ- $\frac{1}{3}$ ולכן זוהי מכונת $BPP$, כי בנוסף ברור שהיא רצה בזמן פולינומי.

71. האם לכל מכונת טיורינג דטרמיניסטית $M$ בעלת סרט אחד, יש מכונת טיורינג דטרמיניסטית $M'$ בעלת סרט אחד שמקיימת את התנאים הבאים?

- הראש הקורא-כותב של $M'$ יכול לנוע רק ימינה או שמאלה (ולא להישאר במקום).
- $M'$ מזהה את אותה השפה כמו של $M$.
- אם $w \in L(M)$, אז במהלך הריצה של $M'$ על $w$, $M'$ מבקרת בכל מצב מהמצבים שלה, למעט המצב הדוחה $q_{reject}$.

**פתרון:**

תחילה, נהפוך את המצב המקבל למצב מקבל פיקטיבי. המצב המקבל האמיתי יהיה $q_{accept}^*$.

נניח שקבוצת המצבים של המכונה, ללא $q_{accept}, q_{reject}, a_{accept}^*$, היא $\{q_1, ..., q_n\}$.

בנוסף, ניקח תו # שאינו באלפבית הקלט, וניצור מעבר מכל מצב פרט ל- $q_{accept}, q_{reject}, a_{accept}^*$ מעבר $\delta(q_i, \#) = (q_i^*, \#, R)$. מטרתו של $q_i^*$ היא לשים סולמית בתא הבא של הסרט, ולהחזיר למצב $q_{i+1}$ את הראש הקורא-כותב כאשר הוא על התא הזה.

נוסיף גם $q_0^*$ שמתפקד באופן זהה.

לכל $\sigma \in \Sigma$ ולכל $0 \le i \le n$, נגדיר:

$\delta(q_i^*, \sigma) = (q_i^{**}, \#, L)$.

$\delta(q_i^{**}, \sigma) = (q_{i+1}, \sigma, R)$.

בנוסף, נוסיף לכל $\sigma \in \Sigma$ את הכלל $\delta(q_{accept}^*, \sigma) = (q_0^*, \#, R)$ ואת הכלל $\delta(q_n, \sigma) = (q_{accept}^*, \#, R)$.

כך, כאשר מגיעים למצב המקבל המקורי, עוברים על כל המצבים ובסוף חוזרים למצב המקבל של המכונה הנ"ל.

72. נגדיר את השפה $SUB_{TM}$ להיות שפת הזוגות $< M_1, M_2 >$ כך ש- $M_1, M_2$ הן מכונות טיורינג, ו- $L(M_1) \subset L(M_2)$.

א. הראו רדוקציית מיפוי של $A_{TM}$ ל- $SUB_{TM}$ (הראו: $A_{TM} \le_m SUB_{TM}$).

ב. הראו רדוקציית מיפוי של $A_{TM}$ ל- $\overline{SUB_{TM}}$ (הראו $A_{TM} \leq_m \overline{SUB_{TM}}$).

**פתרון:**

א. קלט: $<M,w>$ כאשר $M$ היא מכונת טיורינג ו- $w$ היא מילה.

1. בנה את המכונה $M'$ הבאה:
   קלט: מילה $x$.
   1. הרץ את $M$ על $w$ במשך $|x|$ צעדים.
      אם $M$ קיבלה – דחה את $x$.
      אחרת – קבל את $x$.
2. בנה מכונת טיורינג $M''$ שמזהה את $\Sigma^*$.
3. החזר $<M',M''>$.

**נכונות:**

ברור שהרדוקציה ניתנת לחישוב.

נניח ש- $<M,w> \in A_{TM}$. אז $M$ מקבלת את $w$ בשלב כלשהו, ולכן $M'$ לא תקבל את כל המילים. היות ש- $L(M'') = \Sigma^*$, נקבל ש- $L(M') \subset L(M'')$, כלומר $<M',M''> \in SUB_{TM}$.

נניח ש- $<M,w> \notin A_{TM}$. אז $M'$ תקבל כל מילה, ולכן $L(M') = L(M'')$ כלומר $<M',M''> \notin SUB_{TM}$. מש"ל.

ב. קלט: $<M,w>$ כאשר $M$ היא מכונת טיורינג ו- $w$ היא מילה.

1. בנה את המכונה $M'$ הבאה:
   קלט: מילה $x$.
   1. הרץ את $M$ על $w$.
      אם $M$ קיבלה – קבל את $x$.
2. בנה מכונת טיורינג $M'$ המזהה את $\Sigma^*$.
3. החזר את $<M',M''>$.

**נכונות:**

ברור שהרדוקציה ניתנת לחישוב.

נניח ש- $<M,w> \in A_{TM}$. אז $M'$ תקבל את כל המילים, ולכן $L(M') = L(M'')$, כלומר $<M',M''> \notin SUB_{TM}$.

נניח ש- $<M,w> \notin A_{TM}$. אז $M'$ לא מקבלת את כל המילים, ולכן $L(M') \subset L(M'')$, ולכן $<M',M''> \in SUB_{TM}$. מש"ל.

73. על השפות $A$ ו- $B$ נתון:
$A \leq_p B$.
$B \leq_L A$.

א. אם נתון ש- $A$ היא NP-שלמה, האם אפשר להסיק שגם $B$ היא NP-שלמה? **הוכיחו**.

ב. אם נתון ש- $B$ היא NP-שלמה, האם אפשר להסיק שגם $A$ היא NP-שלמה? **הוכיחו**.

**פתרון:**

מהדיון בראש עמוד 351 ניתן להסיק שזמן הריצה של מתמר לוגריתמי הוא $O(n^2)$. לכן זוהי רדוקציה בזמן פולינומיאלי. לכן $A \equiv_p B$. אך אם $A \leq_p B$ ו- $B \in NP$, אז גם $A \in NP$, אז ממשפט 7.36 נסיק את נכונות שתי הטענות.

74. $UHAMPATH$ היא שפת השלשות $<G,s,t>$ כך ש- $G$ גרף לא מכוון, ויש מסלול המילטון מ- $s$ ל- $t$ ב- $G$.

גרף **לא מכוון** נקרא דו-צדדי אם אפשר לחלק את קבוצת הצמתים $V$ לשתי קבוצות זרות כך שקצה של כל קשת בגרף שייך לקבוצה אחת, וקצה השני לקבוצה השנייה.

השפה $BIHAMPATH$ היא שפת הגרפים הדו-צדדיים שיש בהם מסלול המילטון מ- $s$ ל- $t$.

להלן מוצגת הוכחה ש- $BIHAMPATH$ היא $NP$ -שלמה.

שייכות ל-NP מוכחת על ידי אותו מאמת כמו ל- $HAMPATH$, רק שהוא בודק האם הגרף הוא דו צדדי (זאת ניתן לעשות בזמן פולינומיאלי).

להלן רדוקציה בזמן פולינומיאלי של $UHAMPATH$ ל- $BIHAMPATH$.

קלט: $<G,s,t>$ כאשר $G=(V,E)$ הוא גרף לא מכוון ו- $s$ ו- $t$ הם צמתים של $G$.

1. לכל קשת $(u,v)\in E$, הוסף לגרף $G$ צומת $uv$.
2. החלף כל קשת $(u,v)$ בשתי הקשתות $(u,uv)$ ו- $(uv,v)$.
3. יהי $H$ הגרף המתקבל. החזר את $<H,s,t>$.

הרדוקציה ניתנת לחישוב בזמן פולינומיאלי: כל קשת הוחלפה בצומת ובשתי קשתות.

הרדוקציה תקפה: $H$ הוא דו צדדי כי קבוצת הצמתים שלו מתחלקת לקבוצת הצמתים המקורית ולקבוצת הצמתים החדשה.

יש ב- $G$ מסלול המילטון מ- $s$ ל- $t$ אם ורק אם יש ב- $H$ מסלול המילטון מ- $s$ ל- $t$ (אותו המסלול שכל קשת בו מוחלפת בשתי הקשתות שהחליפו אותה).

מה לא נכון ב"הוכחה" הזו? הסבירו **במדויק** איזו נקודה בהוכחה שגויה.

**פתרון:**

מסלול המילטון בגרף המקורי לא בהכרח יהיה מסלול המילטון בגרף החדש, כי אם קשת לא נכללה במסלול המקורי, נאמר $e=(u,v)$ אז הצומת $uv$ לא יכלל במסלול בגרף החדש. לכן זוהי לא רדוקציית מיפוי ובפרט לא רדוקציה פולינומית.

75. קבוצת צמתים **קשירה** בגרף **לא מכוון** $G=(V,E)$ היא קבוצת צמתים $U\subseteq V$ כך שיש מסלול בין כל שני צמתים של $U$.

נגדיר את השפה $HALF-CONN$ להיות שפת התיאורים של גרפים לא מכוונים, $<G>$, כך שיש ב- $G$ קבוצת צמתים קשירה המכילה **לפחות מחצית** מן הצמתים של הגרף.

הראו רדוקציה **במקום לוגריתמי** של השפה $UPATH$ ( $PATH$ עבור גרפים לא מכוונים) לשפה $HALF-CONN$ (הראו $UPATH\le_L HALF-CONN$).

תארו את הרדוקציה, הראו שהיא תקפה והראו שהיא ניתנת לחישוב במקום לוגריתמי.

**פתרון:**

קלט: $<G,s,t>$ כאשר $G$ הוא גרף לא מכוון ו- $s$ ו- $t$ הם צמתים.

1. מנה את מספר הצמתים ב- $G$, $n$.
2. הוסף $n$ צמתים וחבר כל אחד מהם ל- $s$.
3. הוסף $n$ צמתים וחבר כל אחד מהם ל- $t$.
4. הוסף $n+2$ צמתים מבודדים.
5. הוצא את $<G'>$.

**נכונות:**

נניח ש- $<G,s,t>\in UPATH$. אז בגרף החדש יש $4n+2$ צמתים, ולכן הקבוצה שמורכבת מ- $n$ הצמתים שמחוברים ל- $s$, מ- $s$, מהמסלול בין $s$ ל- $t$, מ- $t$ ומ- $n$ הצמתים שמחוברים ל- $t$ מכילה לפחות $2n+2$ צמתים, והיא גם קשירה. לכן היא מתאימה.

נניח ש- $<G'>\in HALF-CONN$. מכיוון שהקבוצה הקשירה צריכה להכיל לפחות $2n+1$ צמתים והיא לא יכולה להכיל אף צומת מבודד (אין מסלול ממנו לאף צומת אחר), היא חייבת להכיל לפחות צומת אחד מה- $n$ שמקושרים ל- $t$. אך הקשר בין הצמתים הללו הוא רק דרך $s$ ו- $t$, ומכיוון שיש ביניהם מסלול נקבל שיש מסלול מ- $s$ ל- $t$, כלומר $<G,s,t>\in UPATH$.

**סיבוכיות זמן ריצה:** כל שצריך הוא למנות את מספר הצמתים, שהוא לכל היותר $n$, ולכן דרישת המקום היא $O(\log n)$, להכפיל את המספר הזה במספר קבוע של גורמים כדרוש על מנת ליצור את הצמתים החדשים, ולהוציאם לפלט.

76. נגדיר את השפה $EQ_{LBA}$ להיות שפת הזוגות $<M_1,M_2>$ כך ש- $M_1$ ו- $M_2$ הם אוטומטים חסומים לינארית, ו- $L(M_1)=L(M_2)$.

א. הוכיחו ש- $EQ_{LBA}$ איננה כריעה.

ב. האם $EQ_{LBA}$ מזוהה-טיורינג?

**פתרון:**

א. נראה רדוקציית מיפוי של $E_{LBA}$ ל- $EQ_{LBA}$.

קלט: $<M>$ כאשר $M$ הוא אוטומט חסום לינארית.

1. צור אוטומט חסום לינארית $M'$ שמכריע את השפה $\varnothing$.
2. החזר את $<M,M'>$.

**נכונות:**

נניח ש- $<M>\in E_{LBA}$. אז $L(M)=\varnothing$ ולכן השפות של האוטומטים זהות, כלומר $<M,M'>\in EQ_{LBA}$.

נניח ש- $<M>\notin E_{LBA}$. אז $L(M)\neq\varnothing$ ולכן שפותיהם שונות, כלומר $<M,M'>\notin EQ_{LBA}$.

ב. נראה ש- $\overline{EQ_{LBA}}$ מזוהה טיורינג: מכונת טיורינג דטרמיניסטית יכולה לעבור על כל המילים בסדר הסטדנרטי, ולבדוק את השייכות של כל אחת מהן לשני האוטומטים על ידי המכונה להכרעת $A_{LBA}$. אם השפות שונות, בסופו של דבר תתקבל מילה שונה, והמכונה תקבל.

לכן לא ייתכן ש- $EQ_{LBA}$ תהיה מזוהה טיורינג, כי אז $EQ_{LBA}$ הייתה כריעה, בסתירה לסעיף א.