<u>**פתרון בחינה לדוגמה 3 סמסטר 2018א**</u>

שאלה 1

למכונה עם אינסוף מצבים **יש יותר כוח** מאשר למכונה עם מספר סופי של מצבים:

לכל שפה L מעל אלפבית Σ אפשר לבנות מכונה עם אינסוף מצבים **שתכריע** את L:

אלפבית הסרט של המכונה יהיה $\Gamma = \Sigma\cup\{\sqcup\}$.

לכל מילה w מעל האלפבית Σ יהיה במכונה מצב שיזכור שעד עתה קראנו את w:

המצב ההתחלתי יזכור שעד עתה קראנו את המילה הריקה.

לכל סמל a של Σ, תצא קשת מן המצב ההתחלתי למצב שיזכור שעד עכשיו קראנו את a.

מכל מצב כזה, תצא, לכל סמל b של Σ, קשת למצב שיזכור שעד עכשיו קראנו את ab, וכך הלאה.

מכל מצב שמתאים למילה **ששייכת לשפה** תצא קשת עם סמל הרווח **למצב המקבל** $\boldsymbol{q_{\text{accept}}}$.

מכל מצב שמתאים למילה **שלא שייכת לשפה** תצא קשת עם סמל הרווח **למצב הדוחה** $\boldsymbol{q_{\text{reject}}}$.

אין בקיום מכונה כזו סתירה לתזה של צ'רץ' וטיורינג, משום שמכונה **בעלת אינסוף מצבים** איננה מודל של מכונה מציאותית.

שאלה 2

כדי להוכיח שהשפה C מזוהה-טיורינג, נתאר מכונת טיורינג שמזהה אותה:

"על קלט $<M, w>$ כאשר M היא מכונת טיורינג ו-w היא מחרוזת סמלים:

1. הרץ את M על w. אם M דחתה, דחה.
2. בדוק את אורך המילה שכתובה על הסרט של M. אם הוא גדול מ-$|w|$, דחה. אחרת, קבל."

כדי להוכיח ש-C איננה כריעה, נשתמש בשיטת האלכסון:

נניח בשלילה ש-C כריעה. אז יש מכונה H שמכריעה שייכות ל-C.

נבנה את המכונה D הבאה:

"על קלט $<M>$ כאשר M היא מכונת טיורינג:

1. הרץ את המכונה H על $<M, <M>>$.
2. אם H הכריעה ש-$<M, <M>>$ שייכת ל-C, דחה.
3. אם H הכריעה ש-$<M, <M>>$ לא שייכת ל-C, מחק את תוכן הסרט, כתוב על הסרט את $<M>$ וקבל."

המכונה D מתנהגת באופן הבא: אם $<M, <M>>$ שייכת ל-C, D דוחה את $<M>$.

אם $<M, <M>>$ לא שייכת ל-C, D מקבלת את $<M>$, ובסיום ריצתה של D על $<M>$ רשומה על הסרט מילה שאינה ארוכה מ-$<M>$.

מה יקרה כאשר נריץ את D על הקלט $<D>$?

אם $<D, <D>>$ שייכת ל-C, כלומר, המכונה D מקבלת את $<D>$ ובסיום ריצתה רשומה על הסרט

מילה שאינה ארוכה מ-$\langle D\rangle$, אז D תדחה את $\langle D\rangle$. כלומר, $\langle D, \langle D\rangle\rangle$ לא שייכת ל-C.
אם $\langle D, \langle D\rangle\rangle$ לא שייכת ל-C, אז D תקבל את $\langle D\rangle$, ובסיום ריצתה תהיה רשומה על הסרט מילה שאינה ארוכה מ-$\langle D\rangle$. כלומר $\langle D, \langle D\rangle\rangle$ כן שייכת ל-C.
בכל מקרה הגענו לסתירה.

שאלה 3

א. **השפה שייכת ל-P.**
מריצים את האוטומט על v.
אם בשלב כלשהו בריצה עוברים במצב מקבל, מקבלים. אחרת, דוחים.
הזמן להרצת האוטומט על v פולינומיאלי בגודל הקלט.

ב. **השפה שייכת ל-P.**
בודקים לכל תחילית w של v האם $w \in L(G)$.
אם נמצאה w כזו, מקבלים. אחרת, דוחים.
מספר התחיליות של מילה w לינארי ב-$|w|$. זמן הבדיקה של כל תחילית פולינומיאלי בגודל התחילית ובגודל הדקדוק שהוא חלק מן הקלט.

ג. **השפה לא שייכת ל-P**. היא לא כריעה.
הראו, למשל, רדוקציה של A_{TM}.

שאלה 4

רדוקציה של $3SAT$:
"על קלט $\langle\phi\rangle$ כאשר ϕ היא נוסחה ב-3CNF:
1. יהי n מספר המשתנים בנוסחה ϕ, ויהי k מספר הפסוקיות ב-ϕ.
2. לכל פסוקית C_i ב-ϕ, $C_i = (l_1 \vee l_2 \vee l_3)$, בנה אוטומט סופי דטרמיניסטי A_i שמזהה את שפת כל המחרוזות הבינאריות שאורכן n והן מייצגות השמות שמספקות את C_i.
3. החזר את $\langle A_1, A_2, \ldots, A_k\rangle$."

נסביר מהי השפה שמזהה האוטומט A_i:
כל מחרוזת בינארית באורך n מייצגת השמה של 0-ים ו-1-ים ל-n המשתנים של הנוסחה.
השמה כזו מספקת את הפסוקית $C_i=(l_1 \vee l_2 \vee l_3)$, אם לפחות באחד המקומות המתאימים לליטרלים של הפסוקית יש בהשמה ערך שהופך את הליטרל הזה ל-1.
מספר המצבים של האוטומט A_i לינארי ב-n: על משתנים שלא מופיעים בפסוקית אפשר לעבור ממצב למצב גם על 0 וגם על 1. על משתנים שכן מופיעים בפסוקית יש התפצלות. על הערך שמספק את הליטרל שבפסוקית עוברים למצב שמאפשר קריאה של 0-ים ו-1-ים עד להשלמת האורך של המחרוזת ל-n, ואז כניסה למצב המקבל היחיד של האוטומט. על הערך שלא מספק את הליטרל שבפסוקית ממשיכים לליטרל הבא. אם מדובר בליטרל השלישי, עוברים למצב מלכודת לא מקבל.

למצב הזה עוברים גם מן המצב המקבל בקריאה של כל סמל (מחרוזת הקלט ארוכה מ-n).

הרדוקציה תקפה: ϕ ספיקה, אם ורק אם יש השמה של 0-ים ו-1-ים למשתני הנוסחה שבה בכל פסוקית יש לפחות ליטרל אחד שערכו 1, אם ורק אם החיתוך של קבוצות ההשמות שמספקות כל אחת מן הפסוקיות לא ריק, אם ורק אם $L(A_1)\cap L(A_2)\cap\cdots\cap L(A_k) \neq \varnothing$, אם ורק אם $\langle A_1, A_2, \ldots, A_k\rangle$ שייכת ל-$NONDISJOINT_{\mathrm{DFA}}$.

הרדוקציה חשיבה בזמן פולינומיאלי: כל אחד מן השלבים של הרדוקציה מתבצע פעם אחת. שלב 1 ושלב 3 פולינומיאליים. גם שלב 2 פולינומיאלי, כי מספר האוטומטים שבונים שווה למספר הפסוקיות, והגודל של כל אחד מהם לינארי במספר המשתנים.

שאלה 5

א. אפשר להוכיח שהשפה #SAT שייכת ל- $\mathrm{SPACE}(n)$:

בונים מכונת טיורינג דומה למכונה M_1 מדוגמה 8.3 בספר (עמוד 332).

המכונה תחזיק **מונה של השמות מספקות**. בתחילה ערכו של המונה הוא 0.

המכונה תעבור על ההשמות האפשריות של ערכי אמת למשתנים של הנוסחה.

בכל פעם שנמצאת השמה מספקת, מגדילים את ערכו של המונה ב-1, ומשווים את ערכו ל-k.

אם הוא שווה ל-k, מקבלים. אחרת, ממשיכים להשמה הבאה של ערכי אמת למשתני הנוסחה.

אם סיימנו לעבור על כל ההשמות, דוחים. (לא נמצאו k השמות מספקות).

ב. התשובה לא תשתנה. גם שפה זו שייכת ל- $\mathrm{SPACE}(n)$.

ההוכחה כמעט זהה לסעיף הקודם, פרט לכך שברגע שנמצאו k+1 השמות מספקות, דוחים.

אם סיימנו לעבור על כל ההשמות, ולא נמצאו יותר מ-k השמות מספקות, מקבלים.

ג. התשובה לא תשתנה. גם שפה זו שייכת ל- $\mathrm{SPACE}(n)$.

ההוכחה כמעט זהה לסעיפים הקודמים.

ברגע שנמצאו k+1 השמות מספקות, דוחים.

אם סיימנו לעבור על כל ההשמות, בודקים האם יש בדיוק k השמות מספקות. אם כן, מקבלים. אם לא, דוחים.

שאלה 6

אם $\mathrm{P} = \mathrm{NP}$, אז יש אלגוריתם בעל זמן ריצה פולינומיאלי ל-SAT.

נתאר אלגוריתם למציאת השמה מספקת לנוסחה אם היא ספיקה.

האלגוריתם מקבל כקלט נוסחה בוליאנית ϕ מעל המשתנים $x_1, \ldots, x_n$.

תחילה האלגוריתם בודק, בעזרת האלגוריתם הפולינומיאלי ל-SAT, האם ϕ ספיקה.

אם לא, האלגוריתם מחזיר "לא".

אם כן, בודקים האם $\phi \wedge x_1$ ספיקה.

אם כן, מציבים $\phi = \phi \wedge x_1$ (x_1 יהיה *true*). אם לא, מציבים $\phi = \phi \wedge \neg x_1$ (x_1 יהיה *false*).

כעת בודקים ביחס ל-ϕ החדשה האם $\phi \wedge x_2$ ספיקה, וכך קובעים את ערך האמת של x_2.

ממשיכים כך עד לקביעת ערך האמת של כל המשתנים.

האלגוריתם מבצע n+1 קריאות לאלגוריתם הפולינומיאלי של SAT, כאשר n הוא מספר המשתנים ב-ϕ. כל קריאה כזו היא עם נוסחה שגודלה לינארי בגודל של ϕ.

לכן זמן הריצה של האלגוריתם פולינומיאלי בגודל של ϕ.