חישוביות וסיבוכיות
מצגת 11-
בעיות חיפוש .VS בעיות הכרעה
והמחלקה $co-NP$

תזכורת:

- בתחילת הקורס דיברנו על מודל מכונות לחישוב **פונקציות** ועל מודל מכונות **לקבלת שפות**.
- הוכחנו שקילות בין שני המודלים.
- כעת נראה רעיון דומה עבור שפות ב-NP.

שפות ב-NP כבעיות הכרעה

- כל השפות שראינו ב-NP עד כה היו מהצורה:

$$\{< Object >|\ \exists(a\ certain\ property\ \pi)\ in\ Object\}$$

למשל ב-$clique$ היינו צריכים להכריע האם קיימת קליקה בגודל הנתון,

ב-VC, האם קיים כיסוי קודקודים בגודל הדרוש, ב-SAT, האם קיימת השמה מספקת.

כל השפות/הבעיות הללו הן **בעיות הכרעה**.

הגדרת בעיות חיפוש

לכל אחת מבעיות ההכרעה הנ"ל, ניתן להתאים בעיית חיפוש:

קלט: $\langle Object \rangle$

פלט: אם π קיים – נחזיר אותו.

אחרת – נחזיר סימן מוסכם (למשל את המחרוזת "לא קיים π כנדרש.")

בעיות אלו הן בעיות **חיפוש**.

בעיות אלה שייכות למחלקה FNP (המחלקה הזו מכילה יחסים ולא שפות).

(להרחבה ראו https://en.wikipedia.org/wiki/FNP_(complexity))

הערה קטנה (להעשרה)

יש בעיות ב-FNP אשר בעיית ההכרעה המתאימה להן לא קשה.

והן מהוות תת מחלקה $TFNP \subseteq FNP$

(להרחבה ראו https://en.wikipedia.org/wiki/TFNP)

למשל יחס R שבו לכל מילה x מובטח שקיים y, כך שמתקיים $(x,y) \in \mathrm{R}$

אך מצד שני לא ידוע על אלגוריתם דטרמיניסטי פולינומי למציאת y.

בעיית החיפוש המתאימה ל-CNF-SAT

הגדרה:

בעיית החיפוש המתאימה ל-$CNF-SAT$ מוגדרת כך:

קלט: פסוק הנתון בצורת CNF.

פלט: השמה המספקת את הפסוק (אם קיימת).

משפט:

אם $P=NP$, אזי קיים אלגוריתם דטרמיניסטי פולינומי לבעית החיפוש של $CNF-SAT$.

(כלומר: אם $CNF-SAT\in P$, אז קיים אלגוריתם **דטרמיניסטי ופולינומי** לבעיית החיפוש המתאימה ל-$CNF-SAT$)

הוכחת המשפט

ראשית, בהינתן פסוק בצורת CNF ϕ, עם n משתנים, $x_1, \dots, x_n$,

ניתן לבחור משתנה מסוים ולהציב עבורו ערך אמת מסוים. ונקבל פסוק CNF חדש עם $n-1$ משתנים.

באופן לא מדויק, נכתוב את ערך האמת בתוך הפסוק.

נשתמש בכללי הלוגיקה הבאים: (השלימו)

$$a \wedge 0 \equiv 0 \qquad a \vee 0 \equiv a$$
$$a \wedge 1 \equiv a \qquad a \vee 1 \equiv 1$$

נגדיר:

- ϕ_0 הוא הפסוק המתקבל מ-ϕ ע"י הצבת ערך 0 במשתנה עם האינדקס הקטן ביותר.
- ϕ_1 הוא הפסוק המתקבל מ-ϕ ע"י הצבת ערך 1 במשתנה עם האינדקס הקטן ביותר.

הוכחה

הוכחה: נניח ש− $P = NP$. אזי קיים אלגוריתם דטרמיניסטי פולינומי A המכריע את $CNF - SAT$.

נתאר אלגוריתם דטרמיניסטי המחזיר השמה מספקת לפסוק CNF נתון, או מחזיר את המחרוזת "לא קיימת השמה מספקת".

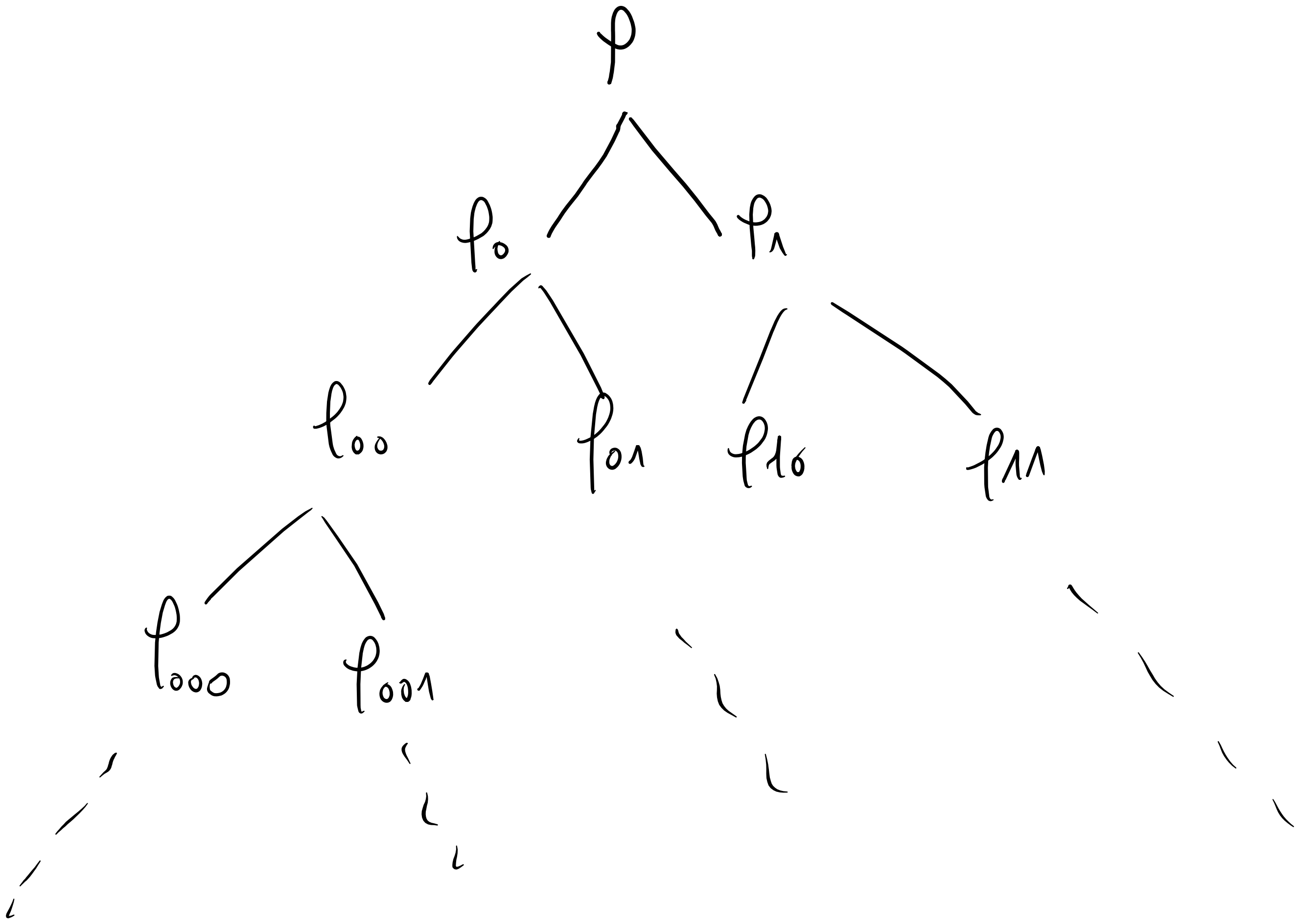
נגדיר השמה לפסוק כמילה מעל $\{0,1\}$, כך שהמקום ה-i במילה יהיה ערך האמת של המשתנה ה-i.

האלגוריתם $SearchSat$:

$SearchSat$ על קלט ϕ (=פסוק CNF עם n משתנים):

1. סמלץ את A על הקלט ϕ. אם $A(\phi) = 0$ החזר "לא קיימת השמה מספקת". אחרת, המשך ל-2.
2. אתחל השמה $\pi \leftarrow \varepsilon$
3. כל עוד (=while) $n > 0$:
הצב ערך $'0'$ במשתנה בעל האינדקס הנמוך ביותר של ϕ- יתקבל הפסוק ϕ_0.
   - סמלץ את ריצת A על הקלט ϕ_0,
     - אם $A(\phi_0) = 1$ בצע $\pi \leftarrow 0 \cdot \pi$
     $\varphi \leftarrow \varphi_0$
     - אחרת, $(A(\phi_0) = 0)$ בצע $\pi \leftarrow 1 \cdot \pi$
     $\varphi \leftarrow \varphi_1$
   - $n \leftarrow n - 1$
4. החזר את π.

ציור האלגוריתם



נכונות וסיבוכיות

- <u>נכונות</u>:

אם $CNF-SAT$ $\phi \notin$ אז בשלב הראשון A יחזיר 0, ו-$SearchSat$ יחזיר "לא קיימת השמה".

אם $CNF-SAT$ $\phi \in$ אז בכל שלב, האלגוריתם A יכריע עבור משתנה יחיד האם הוא צריך לקבל ערך $0/1$, ובסוף התהליך תהיה לנו השמה מספקת.

- <u>סיבוכיות</u>

נסמן ב-$t(\cdot)$ את הסיבוכיות של הריצה של A, על פי ההנחה, $t(\cdot)$ הוא פולינום.

האלגוריתם $\boldsymbol{SearchSat}$ קורא ל-A, n פעמים, לכן הסיבוכיות היא $O(n \cdot t(n))$ - פולינומית.

נשים לב שבכל שלב חישוב האלגוריתם "יורד" רמה **אחת** בעץ הבינארי שציירנו, ו-A מכוון אותו בכל צומת לאיזה בן לרדת. עומק העץ הוא n כך שהסיבוכיות של האלגוריתם שלנו היא אכן $O(n \cdot t(n))$.

דוגמאות לבעיות חיפוש

3. (24 נקודות)

פולינומי

א. (10 נקודות) בסעיף זה, נניח שחוקר מוכשר הצליח למצוא אלגוריתם דטרמיניסטי ל $Clique$, ומימש אותו באמצעות מכונת טיורינג M_{Clique}. תארו (במילים) מימוש של מכונת טיורינג יעילה שבהינתן קלט (G,k) מחזירה קליק ב G שגודלו k, או מחזירה "לא קיים", אם לא קיים קליק כזה.

פתרון (רעיון ההוכחה)

- בדיקה ראשונית, אם הקלט $\langle G,k \rangle$ בשפה ($Clique$)
- בכל שלב נוריד קודקוד, ואת כל הצלעות החלות בו. יהי G' הגרף המתקבל
- נבדוק אם $\langle G',k \rangle$ בשפה, אם כן, נמשיך, אחרת נחזור אל G.

- נעצור כאשר גודל הגרף G' הוא בדיוק k קודקודים.
- נחזיר את קבוצת הקודקודים של G'.

השלימו את הוכחת הנכונות וחישוב סיבוכיות הזמן של האלגוריתם.

דוגמאות לבעיות חיפוש

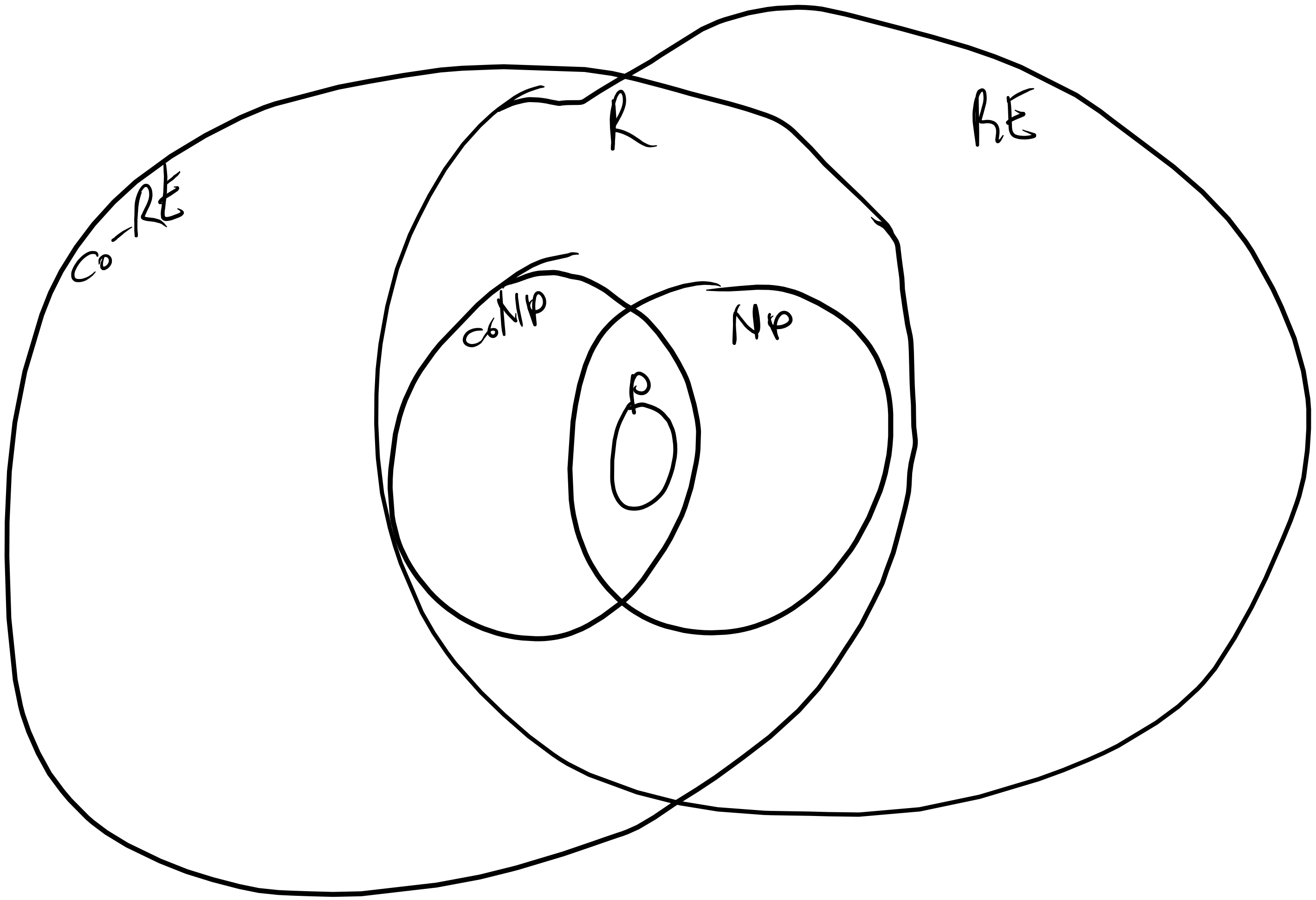
- 4. (20 נקודות) בשאלה זו, נניח כי קיים אלגוריתם דטרמיניסטי פולינומי (הממומש במכונת טיורינג $M_{HamPath}$) המכריע את השפה
$HAMPATH = \{< G >| G \text{ } is\ a\ graph\ that\ has\ a\ Hamiltonian\ path\}$
- א. מה ההשלכה של קיום האלגוריתם הזה לשאלה האם $P = NP$? (3 נקודות)
- ב. הראו אלגוריתם דטרמיניסטי פולינומי אשר בהינתן קלט $< G >$ מחזיר פרמוטציה של קודקודי הגרף אשר מהווה מסלול המילטוני, אם קיים כזה, אחרת מחזירה את המחרוזת "לא קיים מסלול המילטוני".
הסבירו את פעולת האלגוריתם במילים שלכם. אין צורך להוכיח נכונות, אך יש להראות שהסיבוכיות הינה פולינומית (17 נקודות).

פתרון (רעיון ההוכחה)

- בדיקה ראשונית, אם הקלט $\langle G\rangle$ בשפה.
- בכל שלב נוריד צלע. נקבל G'.
- נבדוק אם $\langle G'\rangle$ בשפה, אם כן, נמשיך, אחרת נחזור אל G.

- נעצור כאשר כמות הצלעות בגרף G' היא בדיוק $n-1$ קודקודים.
- נשתמש בקבוצת הצלעות שנותרה כדי להחזיר את הסידור של הקודקודים על המסלול ההמילטוני בגרף.

תמונת העולם



ויש הרבה אי וודאויות (וכיווני מחקר מעניינים לעתיד):

- ?$P = NP$
- ?$P = coNP$
- ?$NP = coNP$?
- האם פיתוחים בחישוב קוונטי, ישנו את כל מה שאנחנו יודעים היום?

תודה ובהצלחה!

(אחרי ההפסקה, נפתור מבחן.
סיימנו את החומר)

הפסקה

תשפב סמסטר א (לא קיץ ☺) מועד א

- 1. א. (11 נקודות) נניח כי הוכח $P \neq NP$. הוכיחו או הפריכו:
תהי $L_1 \in NPC$, ותהי L_2 שפה כך ש $L_2 \subseteq L_1$. אזי $L_2 \in NPC$.
- ב. (12 נקודות) נניח כי הוכח $P \neq NP$. הוכיחו או הפריכו:
אם שפה $L_1 \in NPC$ וגם $L_2 \leq_P L_1$ ו-$L_1 \leq_P L_2$ אזי גם $L_2 \in NPC$.
- ג. (11 נקודות) נניח כי הוכח $P = NP$. תארו במילים מימוש של מכונת טיורינג דטרמיניסטית פולינומית שבהינתן קלט $< G,k >$ מחזירה קבוצה בלתי תלויה (IS) ב-G בגודל k, או מחזירה "לא קיים" אם לא קיימת קבוצה כזו. אין צורך להוכיח נכונות (כן להסביר), אך חשבו את סיבוכיות הזמן של המכונה.

- 2. א. (20 נקודות) יהיו השפות הבאות:
$$3-COL=\{<G>|G\ is\ an\ undirected\ graph\ that\ is\ 3-colorable\}$$
$$4-COL=\{<G>|G\ is\ an\ undirected\ graph\ that\ is\ 4-colorable\}$$
הראו רדוקציה מ $3-COL$ אל $4-COL$.
תזכורת: גרף יקרא k-צביע אם ניתן לצבוע את הקוקדוקים שלו ב-k צבעים (או פחות) כך ששני קודקודים סמוכים אינם צבועים באותו צבע.
- ב. (14 נקודות) בהינתן שפה L הגדרנו את L^R (ההיפוך של L) להיות השפה הבאה:
$$L^R=\{w^R|w\in L\}$$
הוכיחו כי $R\setminus RE$ סגורה תחת ההיפוך. כלומר, הוכיחו כי אם $L\in RE\setminus R$ אזי $L^R\in RE\setminus R$.
- (אחת השאלות הכי קלות שראיתי בקורס הזה)

- 3. נגדיר את השפה L הבאה:
$L = \{< M >| M \text{ is a } TM, \text{and } \exists n > 0 \text{ such that for all } w \in \Sigma^n\ M \text{ accepts } w\}$
- א. הוכיחו או הפריכו: $L \in RE$.
- ב. הראו כי $HP \leq L$.
- ג. (ללא קשר בין הסעיפים)
הוכיחו כי לכל שפה L מעל Σ מתקיימת הטענה הבאה:
אם $L \in R$ וגם $L \neq \Sigma^*$ אזי קיימת מכונת טיורינג M **שאינה מכונה מכריעה** כך ש $L(M) = L$. הסבירו מדוע הטענה לא נכונה עבור $L = \Sigma^*$.

נתחיל לפתור סוף סוף

- 1. א. (11 נקודות) נניח כי הוכח $P \neq NP$. הוכיחו או הפריכו:
תהי $L_1 \in NPC$, ותהי L_2 שפה כך ש $L_2 \subseteq L_1$. אזי $L_2 \in NPC$.
- פתרון:

נתחיל לפתור סוף סוף

- 1. א. (11 נקודות) נניח כי הוכח $P \neq NP$. הוכיחו או הפריכו:
תהי $L_1 \in NPC$, ותהי L_2 שפה כך ש $L_2 \subseteq L_1$. אזי $L_2 \in NPC$.
- פתרון:
כפי שהתרגלנו מאוטומטים, הכלה אינה תכונת סגור. לכן, זוהי הפרכה.
דוגמה נגדית: תהי $L_1 = SAT$ ו-$L_2 = \Sigma^*$.
- היות ואנחנו מכירים מכונה דטרמיניסטית פולינומית מכריעה לשפה L_2, נקבל כי $L_2 \in P$. לפי ההנחה הכתובה בראש השאלה, $P \neq NP$ ולכן $L_2 \notin NP$ ובפרט $L_2 \notin NPC$.

- ב. (12 נקודות) נניח כי הוכח $P \neq NP$. הוכיחו או הפריכו:
אם שפה $L_1 \in NPC$ וגם $L_2 \leq_P L_1$ ו-$L_1 \leq_P L_2$ אזי גם $L_2 \in NPC$.
- פתרון:

- ב. (12 נקודות) נניח כי הוכח $P \neq NP$. הוכיחו או הפריכו:
אם שפה $L_1 \in NPC$ וגם $L_2 \leq_P L_1$ ו-$L_1 \leq_P L_2$ אזי גם $L_2 \in NPC$.

- פתרון:
זוהי הוכחה. החלק הראשון מעיד כי $L_2 \in NPh$ (טרנזיטיביות- מכל שפה במחלקה אל L_1, ומשם אל L_2), אך בעזרת החלק השני נראה כי $L_2 \in NP$, ובכך נראה כי $L_2 \in NPC$.

- לפי ההנחה, קיימת מכונה א"ד פולינומית M_{L_1} המכריעה את L_1. בנוסף, היות ו $L_2 \leq_P L_1$, קיימת מכונה לחישוב הרדוקציה M_f הרצה בזמן פולינומי. להלן מכונה א"ד N המכריעה את L_2 בזמן פולינומי. המכונה N על הקלט x:
  - הרץ את M_f על x וקבל $f(x)$.
  - הרץ את M_{L_1} על $f(x)$ וענה כמוה.

- נכונות נובעת מתקפות הרדוקציה ונכונות M_{L_1}. המכונה פולינומית (הרכבת פולינומים), ולכן בנינו מכונה א"ד לשפה L_2 ומכאן $L_2 \in NP$.

- ג. (11 נקודות) נניח כי הוכח $P = NP$. תארו במילים מימוש של מכונת טיורינג דטרמיניסטית פולינומית שבהינתן קלט $< G,k >$ מחזירה קבוצה בלתי תלויה (IS) ב-G בגודל k, או מחזירה "לא קיים" אם לא קיימת קבוצה כזו. אין צורך להוכיח נכונות (כן להסביר), אך חשבו את סיבוכיות הזמן של המכונה.
- פתרון:

- ג. (11 נקודות) נניח כי הוכח $P = NP$. תארו במילים מימוש של מכונת טיורינג דטרמיניסטית פולינומית שבהינתן קלט $< G,k >$ מחזירה קבוצה בלתי תלויה (IS) ב-G בגודל k, או מחזירה "לא קיים" אם לא קיימת קבוצה כזו. אין צורך להוכיח נכונות (כן להסביר), אך חשבו את סיבוכיות הזמן של המכונה.

- פתרון:
יהי אלגוריתם A להכרעת בעיית IS בזמן פולינומי. להלן אלגוריתם חיפוש מתאים:
  - הרץ את A על $< G,k >$. אם התשובה 0 החזר "לא קיים".
  - לכל $i = 1, \dots |V(G)|$:
    - הוצא את הקודקוד v_i מהגרף. (ואת הצלעות שלו) נקבל G'.
    - אם $A(G',k) = 0$ – נחזור ל-G. אחרת, נמשיך לאיטרציה הבאה עם $G := G'$.

- הקודקודים שנותרו מהווים את הקבוצה הבלתי תלויה בגודל k.

אבחנה בתרגיל האחרון

- שימו לב כי בעת הוצאת קודקוד הוצאנו גם את כל הצלעות שלו. האם לא יתכן שיצרנו קבוצה בלתי תלויה חדשה בגודל k?
- א. התחלנו בבדיקה האם בגרף המקורי קיימת קבוצה מספיק גדולה. אם לא, בכלל לא הגענו ליצירת קבוצות חדשות כאלו אלא החזרנו "לא קיימת" מיד.
- ב. יכולות להיות כמה קבוצות בלתי תלויות בגודל k. זה לא מפריע להגדרה.
- סיבוכיות ריצה:
יהי $t(\cdot)$ זמן הריצה של האלגוריתם A.
הפעלנו את האלגוריתם $|V(G)|$ פעמים, ולכן הסיבוכיות היא $O\big(|V(G)|\cdot t(\cdot)\big)$

- 2. א. (20 נקודות) יהיו השפות הבאות:
$$3-COL=\{<G>|G\ is\ an\ undirected\ graph\ that\ is\ 3-colorable\}$$
$$4-COL=\{<G>|G\ is\ an\ undirected\ graph\ that\ is\ 4-colorable\}$$
הראו רדוקציה מ $3-COL$ אל $4-COL$.
תזכורת: גרף יקרא k-צביע אם ניתן לצבוע את הקוקדוקים שלו ב-k צבעים (או פחות) כך ששני קודקודים סמוכים אינם צבועים באותו צבע.

- פתרון:

- 2. א. (20 נקודות) יהיו השפות הבאות:

$$3-COL=\{<G>|G\ is\ an\ undirected\ graph\ that\ is\ 3-colorable\}$$
$$4-COL=\{<G>|G\ is\ an\ undirected\ graph\ that\ is\ 4-colorable\}$$

הראו רדוקציה מ $3-COL$ אל $4-COL$.
תזכורת: גרף יקרא k-צביע אם ניתן לצבוע את הקוקדוקים שלו ב-k צבעים (או פחות) כך ששני קודקודים סמוכים אינם צבועים באותו צבע.

- פתרון:
אחת הרדוקציות הקלות ביותר:

$$f(<G>)=(<G'>)$$

כאשר לגרף G' הוספנו קודקוד יחיד, המחובר לכל הקודקודים המקוריים. הרדוקציה פולינומית (הוספת קודקוד ו $|V(G)|$ צלעות). נראה כי הרדוקציה תקפה:

$f(3-COL) \rightarrow (4-COL)$

- אבחנה: הקודקוד החדש ב G' לא יכול לקבל אף אחד מהצבעים המקוריים שהיו בגרף G.
  הוכחה: הוא מחובר לכולם, ולכן צביעה חוקית לא יכולה לתת לו אף אחד מהצבעים הקודמים.

- $< G > \in 3-COL \rightarrow \exists 3-coloring\ of\ V(G)$
  $\rightarrow The\ new\ vertex\ can\ take\ forth\ color \rightarrow G' \in 4-COL$

- $< G > \notin 3-COL \rightarrow G\ needs\ more\ than\ 3\ colors\ for\ equitable\ coloring$
  $\rightarrow The\ new\ vertex\ cannot\ take\ any\ of\ the\ former\ colors, he\ needs\ another$
  $\rightarrow < G' > \notin 4-COL$

- 2. ב. (14 נקודות) בהינתן שפה L הגדרנו את L^R (ההיפוך של L) להיות השפה הבאה:
$$L^R = \{w^R \mid w \in L\}$$
הוכיחו כי $RE \setminus R$ סגורה תחת ההיפוך. כלומר, הוכיחו כי אם $L \in RE \setminus R$ אזי $L^R \in RE \setminus R$.
- פתרון:

- 2. ב. (14 נקודות) בהינתן שפה L הגדרנו את L^R (ההיפוך של L) להיות השפה הבאה:
$$L^R = \{w^R | w \in L\}$$
הוכיחו כי $R \setminus RE$ סגורה תחת ההיפוך. כלומר, הוכיחו כי אם $L \in RE \setminus R$ אזי $L^R \in RE \setminus R$.

- פתרון:
תהי $L \in RE \setminus R$ ותהי M_L מכונה מקבלת עבורה. נבנה מכונה מקבלת M^R עבור L^R.
המכונה M^R על הקלט x:
- הופכת את x (לדוגמה, מעתיקה אותו לסרט נוסף, חוזרת להתחלת הסרט הראשון וכותבת אותו מסוף הסרט השני להתחלה).
- מפעילה את M_L על הקלט החדש (x^R) ועונה כמוה.

- נכונות נובעת מנכונות M_L.

- 3. נגדיר את השפה L הבאה:

$$L = \{< M > | M \; is \; a \; TM, and \; \exists n > 0 \; such \; that \; for \; all \; w \in \Sigma^n \; M \; accepts \; w\}$$

- א. הוכיחו או הפריכו: $L \in RE$.
- פתרון:

- 3. נגדיר את השפה L הבאה:
$L = \{< M > | M \text{ is a } TM, \text{and } \exists n > 0 \text{ such that for all } w \in \Sigma^n \; M \text{ accepts } w\}$

- א. הוכיחו או הפריכו: $L \in RE$.

- פתרון:
הוכחה: נבנה מ"ט א"ד מקבלת לשפה L.
המכונה N על הקלט $< M >$:
- מנחשת אורך $n > 0$.
- מריצה את $< M >$ על כל המילים באורך n.
אם כולן התקבלו, מקבלת. אם לפחות אחת נדחתה, דוחה.

3א

- נכונות:
אם $L \in> M <$ אזי קיים n עבורו כל מילה באורך n מתקבלת במכונה M. לכן, המכונה N תנחש את האורך הנכון, תריץ את M על כל המילים הללו ותקבל את $> M <$.
אם $L \notin> M <$ אזי לכל n קיימת לפחות מילה אחת באורך n שהמכונה M דוחה או לא עוצרת עליה. לכן, לכל ניחוש של n המכונה N תדחה (אם M דחתה) או לא תעצור (אם M לא עצרה) על המילה הזו. לא מפריע ל RE.

- 3. נגדיר את השפה L הבאה:
$L = \{< M >| M\ is\ a\ TM, and\ \exists n > 0\ such\ that\ for\ all\ w \in \Sigma^n\ M\ accepts\ w\}$
- ב. הראו כי $HP \leq L$.
- פתרון:

- 3. נגדיר את השפה L הבאה:
$$L = \{< M >| M\ is\ a\ TM, and\ \exists n > 0\ such\ that\ for\ all\ w \in \Sigma^n\ M\ accepts\ w\}$$
- ב. הראו כי $HP \leq L$.
- פתרון:
נגדיר את הרדוקציה $f: HP \to L$.
$$f(< M >, < x >) \to < M_x >$$
המכונה M_x על קלט y:
- מריצה את M על x.
- מקבלת.
- הרדוקציה מלאה (מוגדרת לכל קלט) וניתנת לחישוב (פעולת קומפליציה פשוטה. בפרט, אין כאן הרצת מכונה שעלולה לא לעצור). נוכיח שהרדוקציה תקפה:

3ב

$$(< M >, < x >) \in HP \;\rightarrow M\ halts\ on\ x \;\rightarrow M_x\ accepts\ each\ y \;\rightarrow L(M_x) = \Sigma^* \\ \rightarrow \forall n > 0\ \forall w \in \Sigma^n : M_x(w) = 1 \;\rightarrow < M_x > \in L$$

$$(< M >, < x >) \notin HP \rightarrow M\ doesn't\ halts\ on\ x \;\rightarrow M_x\ never\ halts \\ \rightarrow L(M_x) = \phi \rightarrow \forall n\ \forall w \in \Sigma^n : M_x(w) = 0 \rightarrow < M_x > \notin L$$

- ג. (ללא קשר בין הסעיפים)
הוכיחו כי לכל שפה L מעל Σ מתקיימת הטענה הבאה:
אם $L \in R$ וגם $L \neq \Sigma^*$ אזי קיימת מכונת טיורינג M **שאינה מכונה מכריעה** כך ש $L(M) = L$. הסבירו מדוע הטענה לא נכונה עבור $L = \Sigma^*$.
- פתרון:

- ג. (ללא קשר בין הסעיפים)
הוכיחו כי לכל שפה L מעל Σ מתקיימת הטענה הבאה:
אם $L \in R$ וגם $L \neq \Sigma^*$ אזי קיימת מכונת טיורינג M **שאינה מכונה מכריעה** כך ש $L(M) = L$. הסבירו מדוע הטענה לא נכונה עבור $L = \Sigma^*$.

- פתרון:
תהי $L \in R$. לכן קיימת לה מכונה מכריעה M_R. היות ו $RE \subseteq R$, ניתן לבנות מכונה מקבלת לכל שפה ב R.
המכונה M על קלט x:
- מריצה את M_R על x.
- אם M_R הגיעה ל q_{acc}, קבל.
- אם M_R הגיעה ל q_{rej}, היכנס ללולאה אינסופית.
נכונות נובעת מנכונות M_R. המכונה אינה מכריעה, כנדרש.

- עבור $L = \Sigma^*$, כל המילים יגיעו לקבלה, ואין מילים שיכנסו ללולאה. לכן, כל מכונה לשפה תכריע אותה.

הפסקה

תשפב סמסטר א מועד ב

- 1. א. (14 נקודות) תהי DP קבוצת כל השפות L עבורן קיימות $L_1 \in NP$, $L_2 \in coNP$ כך ש $L = L_1 \cap L_2$. תהי L השפה הבאה:
  $L = \{< G, k >| G\ is\ an\ undirected\ graph\ that\ has\ IS\ of\ size\ k, and\ does\ \boldsymbol{not}\ have\ an\ IS\ of\ size\ k+1\}$
  כלומר, ב G יש IS בגודל k, אבל לא בגודל $k+1$ ומעלה.
  הוכיחו כי $L \in DP$.
- ב. (12 נקודות) הוכיחו או הפריכו: אם $coNP \subseteq NP$ אזי $NP = coNP$.
- ג. (12 נקודות) נגדיר את השפה הבאה L:
  $L = \{< \phi >| \phi\ is\ a\ Boolean\ formula\ that\ has\ at\ most\ one\ satisfying\ assignment\}$
  הוכיחו או הפריכו: $L \in coNP$.

- 2. א. (17 נקודות) בהינתן גרף לא מכוון G, נסמן ב $\omega(G)$ את גודל הקליקה הגדולה ביותר ב G. נגדיר את השפה הבאה:
$L = \{< G_1, G_2 >| G_1, G_2 \text{ are undirected graphs and } \omega(G_1) \geq \omega(G_2)\}$
הראו רדוקציה פולינומית מ-$CLIQUE$ אל L. הוכיחו את תשובתכם.

- ב. הוכיחו את ההרחבה הבאה של משפט רייס: אם S היא תכונה לא טריוויאלית של שפות ב-RE כך ש $\Sigma^* \notin S$ אזי $L_S \notin RE$.
רמז: הראו רדוקציה מהשפה $\overline{HP}$.

- 3. בהינתן מ"ט לא דטרמיניסטית N, מילה w תיקרא **רב-מסלולית** ב-N אם יש ל-w (במכונה N) לפחות מסלול חישוב מקבל אחד, לפחות מסלול חישוב דוחה אחד, ולפחות מסלול חישוב אחד שלא עוצר. **השפה הרב-מסלולית** של המכונה הא"ד N היא קבוצת המילים שהן רב מסלוליות ב-N.
א. (15 נקודות) הוכיחו או הפריכו: אם $L \in RE$ אז קיימת מ"ט א"ד שהשפה הרב מסלולית שלה שווה ל-L.

- ב. (15 נקודות) הוכיחו או הפריכו: קיימת מ"ט א"ד שהשפה הרב מסלולית שלה היא השפה הבאה:

$$L = \{< M, w > | M \text{ } is \text{ } a \text{ } TM \text{ } that \text{ } does \text{ } not \text{ } accept \text{ } w\}$$

אם אתם טוענים שקיימת מכונה כזו, תנו תיאור של המכונה והוכיחו שהשפה שלה היא השפה הדרושה. אם אתם טוענים שלא קיימת מכונה כזו, הוכיחו מדוע לא יכולה להיות קיימת.

פתרונות

- 1. א. (14 נקודות) תהי DP קבוצת כל השפות L עבורן קיימות $L_1 \in NP$, $L_2 \in coNP$ כך ש $L = L_1 \cap L_2$. תהי L השפה הבאה:
$L = \{< G, k > | G\ is\ an\ undirected\ graph\ that\ has\ IS\ of\ size\ k, and\ does\ \boldsymbol{not}\ have\ an\ IS\ of\ size\ k+1\}$
כלומר, ב G יש IS בגודל k, אבל לא בגודל $k+1$ ומעלה.
הוכיחו כי $L \in DP$.

- פתרון:

פתרונות

- 1. א. (14 נקודות) תהי DP קבוצת כל השפות L עבורן קיימות $L_1 \in NP$, $L_2 \in coNP$ כך ש $L = L_1 \cap L_2$. תהי L השפה הבאה:
$L = \{< G, k >| G \text{ is an undirected graph that has IS of size } k, \text{and does } \textbf{not} \text{ have an IS of size } k+1\}$
כלומר, ב G יש IS בגודל k, אבל לא בגודל $k+1$ ומעלה.
הוכיחו כי $L \in DP$.

- פתרון:
תהי $L_1 = \{< G,k >| G \text{ is an undirected graph that has IS of size } k\}$, ותהי
$L_2 = \{< G, k+1 >| G \text{ is an undirected graph that doesn't have IS of size } k+1\}$

תחילה נראה כי $L_1 \in NP$ וכי $\overline{L_2} \in NP$:

$$L_1 = \{< G, k >| G\ is\ an\ undirected\ graph\ that\ has\ IS\ of\ size\ k\} \in NP$$

- נבנה מ"ט א"ד לשפה L_1. המכונה N על קלט $< G,k >$:
  - מנחשת קבוצה בעלת k קודקודים שונים.
  - מוודאת כי הקבוצה היא IS (עוברת על מטריצת השכנויות ומוודאת שאין שני קודקודים בקבוצה שיש ביניהם צלע). אם נמצאה צלע – דוחה.
  - מקבלת.
- המכונה פולינומית (ניחוש ב $O(|V(G)|)$, וידוא שאין שכנים ב $O(|V(G)|^2) = O(k^2)$).
  נכונות:
  אם $< G,k >\in L_1$ אזי קיימת קבוצה ב"ת בגודל k. הניחוש ימצא אותה, והמכונה תוודא שזו אכן קבוצה ב"ת.
  אם $< G,k >\notin L_1$ אזי לא קיימת קבוצה כנ"ל, ולכן לכל ניחוש המכונה תמצא צלע ותדחה.

$\overline{L_2} = \{< G, k+1 > | G \text{ is an undirected graph that } \textbf{does} \text{ have IS of size } k+1\}$

- מכונה א"ד פולינומית עבור $\overline{L_2}$ תהיה כמעט זהה למכונה מהשקף הקודם, פרט לכך שכעת הניחוש בגודל $k+1$, והווידוא ירוץ ב $O\left((k+1)^2\right)$. עדיין פולינומי.
- אין טעם לכתוב אותה מכונה פעמיים (גם במבחן שיהיה הסמסטר אנחנו נעריך את הקיצורים, כל עוד זה באמת אותו דבר).

- לכן $L = L_1 \cap L_2$ כנדרש, ולפי ההגדרה $L \in DP$.

- ב. (12 נקודות) הוכיחו או הפריכו: אם $coNP \subseteq NP$ אזי $NP = coNP$.
- פתרון:

• ב. (12 נקודות) הוכיחו או הפריכו: אם $NP \subseteq coNP$ אזי $NP = coNP$.

• פתרון:

נניח כי $NP \subseteq coNP$, ונרצה להראות שלכל $L \in coNP$ מתקיים $L \in NP$

$$L \in coNP \Rightarrow_1 \bar{L} \in NP \Rightarrow_2 \bar{L} \in coNP \Rightarrow_3 L \in NP$$

1. לפי הגדרת NP
2. לפי ההנחה
3. לפי הגדרת NP

- ג. (12 נקודות) נגדיר את השפה הבאה L:

$$L = \{< \phi >|\phi \text{ is a Boolean formula that has at most one satisfying assignment}\}$$

הוכיחו או הפריכו: $L \in coNP$.

- פתרון:

- ג. (12 נקודות) נגדיר את השפה הבאה L:
$L = \{<\phi>|\phi\ is\ a\ Boolean\ formula\ that\ has\ at\ most\ one\ satisfying\ assignment\}$
הוכיחו או הפריכו: $L \in coNP$.

- פתרון:
נוכיח כי $\bar{L} = \{<\phi>\ |\phi\ is\ a\ Boolean\ formula\ that\ has\ at\ least\ 2\ satisfying\ assignments\} \in NP$

- כרגיל, נסמן כי בנוסחה יש n משתנים פעילים (אם יש משתנים שלא מופיעים באף פסוקית, הם לא מעניינים אותנו).
המכונה N על קלט ϕ:
- מנחשת שני וקטורים בינאריים שונים באורך n (השמות).
- מציבה את הוקטור הראשון בנוסחה ומוודאת כי היא קיבלה ערך אמת. אם לא, דוחה.
- מציבה את הוקטור השני בנוסחה ומוודאת כי היא קיבלה ערך אמת. אם לא, דוחה;
אם כן, מקבלת.

- בנינו מכונה א"ד פולינומית לשפה $\bar{L}$ ולכן $\bar{L} \in NP$, ומכאן $L \in coNP$.
- פרטים קטנים: המכונה פולינומית (ניחוש באורך $2n$, הצבת ערכים וחישוב ערך הנוסחה פולינומי באורך הנוסחה). נכונות:
- $\bar{L} \ni <\phi>$ ולכן קיימות לפחות שתי השמות מספקות שונות. לכן, המכונה N תנחש שתיים מהן, תוודא ערך אמת בשתי ההשמות ותקבל.
$\bar{L} \notni <\phi>$ ולכן קיימת לכל היותר השמה מספקת אחת. לכן, המכונה N תנחש אותה השמה מספקת פעמיים (אבל היא מנחשת וקטורים שונים), או שלא תמצא כלל השמה מספקת, ותדחה.
- שימו לב: אם יש משתנים "לא פעילים", יתכן והכמות שלהם תהיה אקספוננציאלית באורך הפסוקית! לכן הוקטורים הם באורך של המשתנים הפעילים בלבד, ואילו שאר המשתנים לא משפיעים על הספיקות, כך שנוכל "להתעלם" מהם ולא לכלול אותם בניחוש. לכן הניחוש נותר פולינומי.

- 2. א. (17 נקודות) בהינתן גרף לא מכוון G, נסמן ב $\omega(G)$ את גודל הקליקה הגדולה ביותר ב G. נגדיר את השפה הבאה:
$$L = \{< G_1, G_2 > | G_1, G_2 \text{ are undirected graphs and } \omega(G_1) \geq \omega(G_2)\}$$
הראו רדוקציה פולינומית מ-$CLIQUE$ אל L. הוכיחו את תשובתכם.

- פתרון:

- 2. א. (17 נקודות) בהינתן גרף לא מכוון G, נסמן ב $\omega(G)$ את גודל הקליקה הגדולה ביותר ב G. נגדיר את השפה הבאה:
$$L = \{< G_1, G_2 > | G_1, G_2 \textit{ are undirected graphs and } \omega(G_1) \geq \omega(G_2)\}$$
הראו רדוקציה פולינומית מ- $CLIQUE$ אל L. הוכיחו את תשובתכם.

- פתרון:

$$f(< G, k >) = < G, K_m >$$

כאשר $m \coloneqq \min\{|V| + 1, k\}$

<u>סיבוכיות:</u> העתקת קידוד הגרף הוא זמן לינארי באורכו, בנוסף, גודל הקליקה הוא לכל היותר $O(|V|^2)$ ולכן קידודה יצריך זמן פולינומי גם כן.

- 2. א. (17 נקודות) בהינתן גרף לא מכוון G, נסמן ב $\omega(G)$ את גודל הקליקה הגדולה ביותר ב G.
נגדיר את השפה הבאה:
$$L = \{< G_1, G_2 >| G_1, G_2 \; are \; undirected \; graphs \; and \; \omega(G_1) \geq \omega(G_2)\}$$
הראו רדוקציה פולינומית מ-$CLIQUE$ אל L. הוכיחו את תשובתכם.

- פתרון:
$$f(< G, k >) = < G, K_m >$$

<u>נכונות:</u>

אם $< G,k > \in CLIQUE$ אז ב-G יש קליקה בגודל לפחות k שזוהי גודל הקליקה הגדולה ביותר ב-K_k , משום $k \leq |V|$ (אחרת לא הייתה קליקה בגודל הזה) אז $m = k$
ולכן $< G,K_m > \in L$

אם $< G,k > \notin CLIQUE$ אז ב-G יש קליקה בגודל קטן ממש מ-k, נפצל לשני מקרים משלימים:

1. אם $|V| < k$ אז $m = |V| + 1$ וב-G כמובן שאין קליקה שגדולה ממס' הקודקודים בו.
2. אם $k \leq |V|$ אז $m = k$ וב-G בצקרה זה בכל מקרה אין קליקה בגודל k, שזוהי גודל הקליקה הגדולה ביותר ב- $K_{m=k}$

ולכן $< G,K_k > \notin L$

- 2. ב. הוכיחו את ההרחבה הבאה של משפט רייס: אם S היא תכונה לא טריוויאלית של שפות ב-RE כך ש $S \notin \Sigma^*$ אזי $L_S \notin RE$.
רמז: הראו רדוקציה מהשפה $\overline{HP}$.
- פתרון:
הוכחנו בהרצאה 3. מוזמנים לעיין שם.

- 3. בהינתן מ"ט לא דטרמיניסטית N, מילה w תיקרא **רב-מסלולית** ב-N אם יש ל-w (במכונה N) לפחות מסלול חישוב מקבל אחד, לפחות מסלול חישוב דוחה אחד, ולפחות מסלול חישוב אחד שלא עוצר. **השפה הרב-מסלולית** של המכונה הא"ד N היא קבוצת המילים שהן רב מסלוליות ב-N.
  א. (15 נקודות) הוכיחו או הפריכו: אם $L \in RE$ אז קיימת מ"ט א"ד שהשפה הרב מסלולית שלה שווה ל-L.
- פתרון:

- 3. בהינתן מ"ט לא דטרמיניסטית N, מילה w תיקרא **רב-מסלולית** ב-N אם יש ל-w (במכונה N) לפחות מסלול חישוב מקבל אחד, לפחות מסלול חישוב דוחה אחד, ולפחות מסלול חישוב אחד שלא עוצר. **השפה הרב-מסלולית** של המכונה הא"ד N היא קבוצת המילים שהן רב מסלוליות ב-N.
א. (15 נקודות) הוכיחו או הפריכו: אם $L \in RE$ אז קיימת מ"ט א"ד שהשפה הרב מסלולית שלה שווה ל-L.

- פתרון:

מאחר ו- $L \in RE$ יש לה מ"ט מקבלת, תהי זו M.

$U(x)$:

תסמלץ את M על x אם דחתה, דחה.

אם קיבלה- נחש שני ביטים
אם הראשון הוא 0, קבל

אחרת, אם השני הוא 1, דחה

כנס ללולאה אינסופית

- 3. בהינתן מ"ט לא דטרמיניסטית N, מילה w תיקרא **רב-מסלולית** ב-N אם יש ל-w (במכונה N) לפחות מסלול חישוב מקבל אחד, לפחות מסלול חישוב דוחה אחד, ולפחות מסלול חישוב אחד שלא עוצר. **השפה הרב-מסלולית** של המכונה הא"ד N היא קבוצת המילים שהן רב מסלוליות ב-N.
א. (15 נקודות) הוכיחו או הפריכו: אם $L \in RE$ אז קיימת מ"ט א"ד שהשפה הרב מסלולית שלה שווה ל-L.

- פתרון:

מאחר ו- $L \in RE$ יש לה מ"ט מקבלת, תהי זו M.

$U(x)$:

תסמלץ את M על x אם דחתה, דחה.

אם קיבלה- נחש שני ביטים
אם הראשון הוא 0, קבל

אחרת, אם השני הוא 1, דחה

כנס ללולאה אינסופית

נכונות:
אם x שייכת לשפה, אז המכונה M תקבל אותה בלפחות מסלול אחד- לפי בניית המכונה, באמת x תהיה רב מסלולית ב-U

אם x לא בשפה, אז כל מסלולי החישוב של M על x דוחים או לא עוצרים, לפי בניית U כל מסלול דוחה ב-M ידחה ב-U, וכל מסלול אינסופי ב-M יהיה גם אין סופי ב-U

- 3. ב. (15 נקודות) הוכיחו או הפריכו: קיימת מ"ט א"ד שהשפה הרב מסלולית שלה היא השפה הבאה:

$$L = \{< M, w > | M \text{ is a } TM \text{ that does not accept } w\}$$

אם אתם טוענים שקיימת מכונה כזו, תנו תיאור של המכונה והוכיחו שהשפה שלה היא השפה הדרושה. אם אתם טוענים שלא קיימת מכונה כזו, הוכיחו מדוע לא יכולה להיות קיימת.

- פתרון:

- 3. ב. (15 נקודות) הוכיחו או הפריכו: קיימת מ"ט א"ד שהשפה הרב מסלולית שלה היא השפה הבאה:
$$L = \{< M, w > | M \text{ } is \text{ } a \text{ } TM \text{ } that \text{ } does \text{ } not \text{ } accept \text{ } w\}$$
אם אתם טוענים שקיימת מכונה כזו, תנו תיאור של המכונה והוכיחו שהשפה שלה היא השפה הדרושה. אם אתם טוענים שלא קיימת מכונה כזו, הוכיחו מדוע לא יכולה להיות קיימת.

- פתרון:

הטענה נכונה, נראה מ"ט א"ד שהשפה הרב מסלולית שלה היא זו.

מטרה:

אם M מקבלת את w אז נרצה שהמסלול על המילה <M,w> לא יהיה רב מסלולי,

ואם היא לא מקבלת מאיזושהי סיבה (לא עצרה או דחתה) ונרצה שיהיו כל האופציות.

נפצל למקרים לנוחות:

- אם M מקבלת את w נרצה שיחסר איזשהו סוג מסלול.
- אם M דוחה את w אז נרצה שיהיה מסלול מקבל, דוחה ומסלול שלא עוצר.
- אם M לא עוצרת על w אז נרצה שיהיה מסלול מקבל, דוחה ומסלול שלא עוצר.

$U(< M,w >)$:

1. הטל 0 או 1- אם יצא 0 דחה
2. הטל 0 או 1- אם יצא 0 קבל
3. סמלץ את M על w – אם קיבל תקבל
4. היכנס ללולאה אינסופית

נרצה להוכיח שהשפה הרב מסלולית של מכונה זו היא
$L = \{< M, w > | M \text{ is a } TM \text{ that does not accept } w\}$

$U(< M,w >)$:

1. הטל 0 או 1- אם יצא 0 דחה
2. הטל 0 או 1- אם יצא 0 קבל
3. סמלץ את M על w – אם קיבל תקבל
4. היכנס ללולאה אינסופית

נשים לב שלכל קלט ב-U יהיו לנו מסלול דוחה ומסלול מקבל משלבים (1), (2), מה שאומר שעבור מילים שכן בשפה נרצה 'להשלים' מסלול שלא עוצר, ועבור מילים אחרות לא.

אם $L \ni < M,w >$ אז M ירוץ על w וייתכנו שני מקרים:

1. M דוחה את w – אם כך ניכנס ללולאה אינסופית והמילה $< M,w >$ כולה באמת תהיה מילה רב מסלולית ב-U
2. M לא עוצרת על w – ושוב, גם U לא תעצור וככה המילה כולה תהיה מילה רב מסלולית ב-U

אחרת, אם $L \not\ni < M,w >$ אז M עוצרת ומקבלת את w וסה"כ יהיה חסר מסלול שלא עוצר.