סיכומים לבוחן בסיבוכיות

פרופ' עודד רגב, 08/05/09 (סמסטר ב')

חישוביות:

משימה חישובית:

חישוב פונקציה $f:\{0,1\}^* \to \{0,1\}^*$, לרוב פונקציות בוליאניות (טווח $\{0,1\}$). בצורה שקולה נגדיר שפה: $L_f = \{x \in \{0,1\}^* | f(x) = 1\}$. ידוע גם בשם בעיות החלטה/הכרעה. מודל חישובי: אלגוריתמים בשפת תכנות כלשהי / מכונת טיורינג, אין מגבלת זיכרון.

מכונת החישוב האוניברסלית:

כל תוכנית בשפת תכנות כלשהי ניתן לייצג ע"י מחרוזת של $\{0,1\}$. תהי α תוכנית, נסמן ב-M_α את האלג' המייצג תוכנית זו.

משפט (טיורינג): **קיים** אלג' U שבהינתן $x, \alpha \in \{0,1\}^*$: $U(x,\alpha) = M_\alpha(x)$ - כלומר U מסמלץ את M_α על הקלט x. אם $M_\alpha(x)$ נתקע, גם U תתקע. רעיון ההוכחה: U עובר על התוכנית α ומסמלץ כל שורה בה. טיורינג הראה שניתן להריץ תוכנית על מחשב.

משפט טיורינג (לקיום פונ' לא חשיבה):

קיימת פונ' $UC:\{0,1\}^* \to \{0,1\}$ שלא ניתנת לחישוב ע"י אף אלגוריתם (פונ' חשיבה: פונ' f שקיים עבורה אלג' M כך שלכל קלט x: $M(x)$ עוצר ומחזיר את $f(x)$).

הוכחה:

דרך א': חישובי עוצמות – גודל קבוצת הפונקציות היא א לעומת גודל קבוצת התוכניות שהיא $\aleph_0$.

דרך ב': שיטת הלכסון: נציב בטבלה את כל האלג' $M_0, M_1, M_{00}, M_{01} \dots$ אל מול כל הקלטים $0,1,00,01, \dots$ ונגדיר את UC באופן הבא:

$UC(\alpha) = \begin{cases} 0, & M_\alpha(\alpha) = 1 \\ 1, & o/w \end{cases}$ - כלומר מחזיר תוצאה הפוכה לאלכסון, או תוצאה כלשהי (0 או 1) אם התוצאה על האלכסון מתבדרת. לפי בניה זו, אף אלג' לא יחשב נכון את UC לכל קלט. מניחים בשלילה שקיים קלט β עבורו M_β מחשב את UC, ומגיעים לסתירה מבניית UC על הקלט β.

משפט: $HALT$ אינה ניתנת לחישוב.

$HALT = \{< \alpha, x > | M_\alpha \ stops \ on \ x\}$. נניח בשלילה שקיים M_{HALT}, ונראה כיצד ניתן לחשב את UC: בהינתן קלט α, M_{UC} מריץ את $M_{HALT}(\alpha,\alpha)$. אם התוצאה לא עוצרת, נעצור ונוציא 1. אם התוצאה עוצרת, נחשבה ונחזיר את ההיפך.

פרק ראשון: סיבוכיות זמן ריצה:

נסתכל לרוב על מחלקות של שפות, כלומר של פונקציות בוליאניות.

הגדרה:

עבור פונ' $T:\mathbb{N} \to \mathbb{N}$ נגדיר $DTIME(T(n))$ בתור אוסף השפות שניתן לפתור אותן בעזרת אלגוריתם שרץ בזמן $c \cdot T(n)$ עבור קבוע c כלשהו.

המחלקה *P*:

- $P = \bigcup_{c \geq 1} DTIME(n^c)$: הגדרה זו תקפה לכל מודל חישובי, כגון שפת C, $JAVA$, TM וכו'.
- BPP: מחלקת אלגוריתמים הרצים בזמן פול' ומשתמשים באקראיות (שאלה פתוחה: $P = BPP$).
- מחלקת BQP (שימוש בפיסיקה קוונטית).

המחלקה *NP*:

מחלקת השפות שניתן לוודא שייכות אליהן בזמן פולינומיאלי. כלומר, שפה $L \subseteq \{0,1\}^*$ היא ב-NP אם קיים פולינום $p:\mathbb{N} \to \mathbb{N}$ ואלגוריתם M הרץ בזמן פולי' כך שלכל קלט $x \in \{0,1\}^*$: $x \in L \Leftrightarrow \exists w \in \{0,1\}^{p(|x|)}. M(x,w) = 1$. כלומר, האלג' בהינתן הקלט ועד באורך פולי' לאורך הקלט יחזיר t (1) אמ"מ x בשפה.

דוגמאות לבעיות:

- קבוצה ב"ת (*IS*) – בעיה ב-*NPC*: בהינתן גרף לא מכוון G ומספר טבעי k, האם יש ב-G קבוצה ב"ת בגודל לפחות k (קבוצת קודקודים ללא קשתות). העד יכול להיות רשימה של k קודקודים. האלג' המוודא M יעבור על העד ויבדוק את תקינותו.
- *HamPath* – בעיה ב-*NPC*: בהינתן גרף לא מכוון G, האם קיים בו מסלול המילטוני – מסלול המבקר בכל קודקוד בדיוק פעם אחת.
- *HamCycle* – בעיה ב-*NPC*: כנ"ל רק בודק קיום מעגל המילטוני.
- *MaxIS* – בעיה ב-Σ_2^P: בהינתן גרף G ומספר k, האם גודל הקבוצה הבלתי תלויה המקסימלית היא בגודל k.
- *Linear Programming* – בעיה ב-*P*: בהינתן אוסף אי-שוויונים, האם יש להם פתרון מעל הממשיים.
- *Integer Programming* – בעיה ב-*NPC*: כמו קודם, רק מעל השלמים.

- *Graph Isomorphism – בעיה ב-NP*: לא הצליחו להראות שבעיה זו ב-NPC או ב-P. בהינתן שני גרפים G, H, האם הם איזומורפיים, כלומר האם הם אותו גרף תחת שינוי שמות הקודקודים. העד יהיה פרמוטציה של שמות הקודקודים, והמוודא יבדוק את תקינות הפרמוטציה ונכונותה. אם בעיה זו ב-NPC, אז $NP = coNP$.
- *Compositness – בעיה ב-P*: בהינתן מספר N האם הוא פריק, כלומר לא ראשוני. גודל הקלט הוא $log\ N$, וביחס לגודל זה מוודאים יעילות. ב-NP בגלל שעד יכול להיות k מחלק כלשהו, שגודלו הוא $log\ N$, והמוודא בודק שהוא מחלק. ב-P בגלל:
- *Primality*: בהינתן מספר N, האם הוא ראשוני. ב-$coNP$, כי ניתן לתת מחלק k. עם הזמן גילו אלג' ב-BPP ולבסוף אלג' דטרמיניסטי ב-P.
- *Factoring – בעיה ב-NP וב-coNP*: בהינתן מספרים N,k, האם ל-N יש מחלק ראשוני בתחום $\{k, k+1, \dots, N-1\}$. ב-NP: ניתן לתת כעד מחלק ראשוני בתחום, ולבדוק בזמן פולי' שהוא בתחום, מחלק וראשוני. ב-$coNP$: ניתן לקבל רשימת מספרים $a_1, \dots, a_j$ ולוודא שמכפלתם היא N, שכל אחד מהם ראשוני ושכולם קטנים מ-k.

טענה: $P \subseteq NP \subseteq EXP = \bigcup_{c\geq 1} DTIME(2^{n^c})$

הוכחה:

- $P \subseteq NP$: בונים מוודא שמתעלם מהעד – המוודא יהיה האלג' המקורי שפותר בזמן פולי'.
- $NP \subseteq EXP$: תהי $L \in NP$ וכל המשתמע מכך. נבנה אלג' M' שמפעיל את האלג' המוודא M על כל $2^{p(|x|)}$ העדים האפשריים. אם לפחות אחת מהריצות קיבלה, נקבל. אחרת נדחה.

הגדרה חלופית ל-NP:

ההגדרה החילופית משתמשת באי-דטר'. נגיד ששפה $L \subseteq \{0,1\}^*$ שייכת ל-$NTIME(T(n))$ אם קיים $c > 0$ וקיים אלג' לא דטר' M כך שלכל x: קיימת בחירה אי דטרמיניסטית שגורמת ל-M לקבל $x \in L \Leftrightarrow$. בנוסף M עוצר אחרי $c \cdot T(n)$ צעדים עבור כל בחירה אי דטר'.

משפט: $NP = \bigcup_{c\geq 1} NTIME(n^c)$

הוכחה: עבור $L \in NTIME(n^c)$, נבנה אלג' מוודא: מריץ את האלג' האי דטר' המקורי, אך במקום פיצולים אי דטר', מתפצל לפי העד הניתן לו (אלג' זה דטר'). עבור $L \in NP$, נבנה אלג' לא דטר': מריץ את האלג' המוודא תוך כדי שבונה עד באופן לא דטרמיניסטי באורך $p(|x|)$.

תרגול:

סגירות לפעולות:

נגדיר $L^* = \{y \mid \exists k \in \mathbb{N}. y = y_1 \dots y_k, \forall i \in \{1, \dots, k\}. y_i \in L\}$.

תרגיל: האם NP סגורה ל-*?

פתרון: כן. עבור $L \in NP$, נגדיר ל-x קלט ל-L^* עד באופן הבא: $0 < i_1 < i_2 \dots < i_k = n$ חלוקה ל-x ו-$w_1, \dots, w_k$ עדים לכל אחד ממחרוזות בחלוקה של x. האלג' המוודא ישתמש במוודא של L לבדוק שכל קטע בחלוקה הוא מחרוזת ב-L. העד באורך $O(n \cdot p(n))$ (p אורך עד ב-L) וזמן הריצה גם כן. נכונות ברורה.

תרגיל: האם P סגורה ל-*?

פתרון: כן. עבור מילה $x = x_1 x_2 \dots x_n$ נבנה גרף בו הצמתים $0,1,2, \dots, n$ ולכל קטע בחלוקה מאינדקס i ל-j נעביר קשת מכוונת (i, j). נבדוק אם בגרף יש מסלול מכוון מ-0 ל-n. זמן ריצה $O(n^2 \cdot p(n))$ - הרצת האלג' המכריע את L על כל הקטעים האפשריים, בדיקת מסלול מכוון ע"י BFS. נכונות ברורה.

תרגיל: האם $coNP$ סגורה ל-*?

פתרון: בהינתן קלט x באורך n יש להוכיח כי ב-G_x אין מסלול מכוון מ-0 ל-n. העד: אוסף זוגות הצמתים שאין ביניהם קשת ועדות לכך שאין ביניהם קשת ($\bar{L} \in NP$ ולכן קיימים עדים כאלה). נקח את הגרף השלם ונסיר ממנו את אותן קשתות, ונבדוק שאכן אין עליו מסלול מכוון מ-0 ל-n.

רדוקציות ושלמות ב-NP:

רדוקציית *Karp*:

פולי' משפה A לשפה B היא פונ' f חשיבה בזמן פולי' (בגודל הקלט) כך שלכל x: $x \in A \Leftrightarrow f(x) \in B$. נסמן $A \leq_p B$.

- אם $A \leq_p B$ וגם $B \leq_p C$ אז $A \leq_p C$.
- אם $A \leq_p B$ ו-$B \in P$ אז $A \in P$ או $B \in NP$ אז $A \in NP$ בהתאם.

NP-hard: A תהיה NP-קשה אם לכל $L \in NP$: $L \leq_p A$.

NP-Complete: A תהיה NPC אם היא NP-קשה וגם $A \in NP$.

משפט: $TMSAT \in NPC$

$TMSAT = \{< \alpha, x, 1^x, 1^t > \mid t$ תוך זמן $< x, u >$ מוציא 1 על הקלט M_α כך שהאלגוריתם $u \in \{0,1\}^n$ קיים$\}$. הסיבה שמשתמשים בייצוג אונארי הוא כדי שהאלג' ירוץ בזמן פולי' ל-n ולא ל-$\log n$.

<u>הוכחה בקיצור</u>: $TMSAT \in NP$: בונים מוודא המקבל u כעד ומסמלץ את M_α על $< x, u >$ למשך t צעדים. גודל הקלט וזמן הריצה פולי' ($n + t \geq$). $TMSAT \in NP - hard$: תהי $L \in NP$ עם אלג' מוודא M_α הרץ בזמן $p(|x|)$ עם עד בגודל $q(|x|)$. הרדוקציה $L \leq_p TMSAT$: תתרגם קלט x ל-L ל: $< \alpha, x, 1^{q(|x|)}, 1^{p(|x|)} >$.

משפט *Cook-Levin*: $SAT \in NPC$

<u>טענה</u>: לכל פונק בוליאנית מעל l משתנים $f: \{0,1\}^l \to \{0,1\}$ יש נוסחת CNF שקולה בגודל $\geq l \cdot 2^l$.

<u>הוכחה</u>: $SAT \in NP$: ברור. $\forall L \in NP. L \leq_p SAT$: תהא M מ"ט לא דטר' המחשבת את L ורצה בזמן $p(|x|)$, עם שתי פוני' מעברים δ_0, δ_1. נתרגם קלט x ל-L לנוסחה φ_x כך ש-$x \in L \Leftrightarrow \varphi_x \in SAT$. לכל קלט x נבנה נוסחה המתארת טבלה בגודל $p(|x|) \times p(|x|)$ המתארת את ריצת המכונה M על x – כל שורה בטבלה היא שלב. לייצוג כל תא נדרש למקום קבוע: $log\,|Q| + log\,|\Sigma| + 1$. φ_x תורכב מתנאים המוודאים שכל השורות מורכבות באופן תקין (השורה הראשונה עם q_0 מעל התא הראשון, השורה האחרונה מייצגת מצב קבלה, מעברי השורות – מצבים – נכונים ע"פ δ_0 או δ_1). סה"כ יתקבל מס' פולי' של תנאים מעל מס' קבוע של משתנים – סה"כ נוסחת φ_x בגודל פולי' והרדוקציה פולי'. יש להוכיח אמ"מ.

טענה: $IS \in NPC$

<u>הוכחה</u>: $IS \in NP$: ברור. $NP - hard \ni E3SAT \leq_p IS$ ($E3SAT = exactly\ 3\ literals\ in\ each\ clause$): לכל φ עם m הסגרים ניצור גרף עם m משולשים שקודקודי כל משולש הם הליטרלים בהסגר. בנוסף כל ליטרל יחובר לשלילתו. אם הנוסחה ספיקה אז יש בגרף קבוצה ב"ת בגודל m – לוקחים את אחד המשתנים המספקים בכל הסגר. אם יש קבוצה ב"ת בגודל m, חייב להיות בה קודקוד אחד בכל שלישיה ולכן ישנה השמה מספקת ע"י סיפוק הליטרלים כפי שמופיעים בקבוצה הב"ת.

טענות נוספות: $Clique, VC \in NPC$.

<u>תרגול</u>:

רדוקציות *Cook / Karp*:

<u>תרגיל</u>: להראות רדוקציית $Karp$: $HamPath \leq_p HamCycle$.

<u>פתרון</u>: בהינתן G הרדוקציה תחזיר G' כך ש-$V(G') = V \uplus \{u\}$ ו-$E(G) = E \uplus (\{u\} \times V) \uplus (V \times \{u\})$ (מוסיפים קודקוד ומחברים אותו לכולם).

נכונות: אם ב-G מסלול המילטוני $v_1 \to \cdots \to v_n$, ב-G' מעגל המילטוני $u \to v_1 \to \cdots \to v_n \to u$. אם ב-$G'$ מעגל המילטוני, מסירים את u ומקבלים מסלול המילטוני.

טענה: NP סגורה לרדוקציות $Karp$.

<u>תרגיל</u>: להראות רדוקציית $Cook$: $HamPath \underset{cook}{\leq} HamCycle$.

<u>פתרון</u>: בהינתן G' קלט לבעיית מעגל המילטון, נבדוק האם $E' = \emptyset$. אם כן, נחזיר שאין מעגל המילטון. אם לא, נבחר קשת כלשהי (u, v) ונריץ את $HamPath(V \uplus \{w, z\}, E \uplus \{(z, v), (u, w)\})$. אם מחזיר כן, נחזיר כן – כי יחד עם הקשת (u, v) ניתן להשלים מעגל המילטון. אחרת, נסיר את הקשת הזו ונחזור לשלב ההתחלה.

טענה: NP סגורה לרדוקציית $Cook$ פולי' אמ"מ $NP = coNP$.

<u>הוכחה</u>: אם NP סגורה לרדוקציית $Cook$ פולי', והרי לכל $L: L \in NP$ $\bar{L} \underset{cook}{\leq} L$ ע"י החזרת התשובה ההפוכה לתשובת האלג' המכריע את L, נקבל כי $\bar{L} \in NP$ ולכן $NP \subseteq coNP$ וגם $coNP \subseteq NP$ ולכן $NP = coNP$. אם $NP = coNP$, תהיה $L \in NP = coNP$ ו-$L' \underset{cook}{\leq} L$, נראה $L' \in NP$. יהי A אלג' המכריע את L'. בהינתן קלט x לשפה L', יהי k מספר הקריאות של A לאלג' המכריע את L. העד שנצפה לו הוא התשובות $a_1, \dots, a_k \in \{0,1\}$ של האלג' המכריע את L, וכן עדים לנכונות תשובות $L: w_1, \dots, w_k$. כיוון ש-$L \in NP = coNP$, ניתן לוודא האם קלט שייך או לא שייך ל-L באמצעות עד. האלג' של L' יסמלץ את A ובכל קריאה לאלג' הפותר את L יוודא בעזרת w_i אם a_i התשובה הנכונה. אם לאורך כל החישוב העדים ענו על הדרישה ו-A קיבל, אז נקבל. אחרת נדחה.

<u>חיפוש לעומת החלטה</u>:

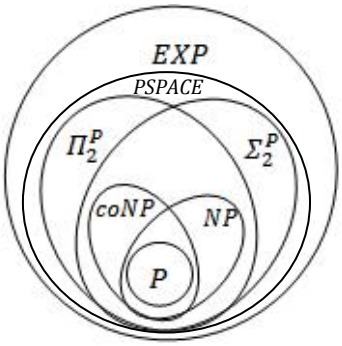
משפט: אם $P = NP$ אז לכל $L \in NP$ יש אלג' פולי' שבהינתן $x \in L$ מוציא עבורו עד.

<u>הוכחה על SAT</u>: נסמן את האלג' הפולי' הפותר את SAT ב-A. בהינתן φ נבדוק האם ספיקה. אם לא – נעצור. אם כן, נקבע את x_1 ל-t ע"י הוספת (x_1) לנוסחה ונבדוק האם ספיקה. אם כן, נמשיך עם $x_1 = t$, אחרת נמשיך עם $x_1 = f$ (נוסיף את $(\overline{x_1})$ במקום (x_1)). לבסוף מתקבלת השמה מספקת. נדרשת הוכחת נכונות.

<u>הוכחה כללית</u>: תהא $L \in NP$ עם אלג' מוודא M. נתרגם את M למ"ט דטר' M' המחשבת את L. לפי $C\text{-}L$ בהינתן x ניתן לבנות φ_x כך ש-$x \in L \Leftrightarrow \varphi_x \in SAT$, ולפי הוכחת $C\text{-}L$, בהינתן השמה מספקת ל-φ_x ניתן בזמן פולי' לשחזר את הבחירות הלא דטר' ומהן לבנות את העד שגורם ל-M לקבל.

<u>המחלקה $coNP$</u>:

$coNP = \{L \mid \bar{L} \in NP\}$. הגדרה חלופית: קיים פולינום p ומ"ט פולי' M כך שלכל $x \in \{01\}^*$: $x \notin L \Leftrightarrow \exists v \in \{0,1\}^{p(|x|)}. M(x,v) = 1$, או באופן שקול: $x \in L \Leftrightarrow \forall v \in \{0,1\}^{p(|x|)}. M(x,v) = 1$. NP ידועה גם כ-$\exists P$, ו-$coNP$ ידועה גם כ-$\forall P$.



דוגמא: $CNF - EQUIV = \{< \varphi, \psi > \mid \forall x \in \{0,1\}^*. \varphi(x) = \psi(x)\}$.

טענה: $\overline{SAT} \in coNPC$ (תחת רדוקציית $Karp$)

טענה: $CNF - EQUIV \in coNPC$

$\overline{SAT} \leq_p CNF - EQUIV$: $\varphi \rightarrow < \varphi, (x) \wedge (\bar{x}) >$ - הנוסחה השניה תמיד לא ספיקה. נכונות ברורה.

<u>המחלקות $EXP, NEXP$</u>:

$EXP = \bigcup_{c \geq 1} DTIME(2^{n^c})$, $NEXP = \bigcup_{c \geq 1} NTIME(2^{n^c})$ (ניתן להגדיר גם באמצעות עד).

ברור שמתקיים: $P \subseteq NP \subseteq EXP \subseteq NEXP$.

משפט: אם $EXP \neq NEXP$ אז $P \neq NP$

<u>הוכחה</u>: נניח $P = NP$. ברור כי $EXP \subseteq NEXP$, נראה $NEXP \subseteq EXP$. תהי $L \in NEXP$, נגדיר בשיטת ה-***Padding***: $L_{pad} = \{< x, 1^{2^{|x|^c}} > \mid x \in L\}$. $L_{pad} \in NP$: בהינתן קלט נבדוק שהוא מהצורה הנדרשת, ואם כן נפעיל עליו את האלג' הלא-דטר' הרץ בזמן $O(2^{|x|^c})$. זמן הריצה הוא פולי' בגודל הקלט ב-L_{pad}. אם כן, לפי הנחה, $L_{pad} \in P$, ומכאן קל לבנות אלג' אקספ' דטר' המכריע את L.

<u>המחלקות Σ_2^P, Π_2^P</u>:

<u>Σ_2^P</u>: מחלקת השפות שיש עבורן מ"ט פולי' M ופולינום p כך שלכל $x \in \{0,1\}^*$: $x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}. \forall v \in \{0,1\}^{p(|x|)}. M(x,u,v) = 1$.

דוגמא: $MIN - DNF$ - בהינתן נוסחת DNF φ ומספר k, האם קיימת נוסחת DNF ψ בגודל לכל היותר k כך ש-$\psi \equiv \varphi$: $\exists \psi. \forall x. \varphi(x) = \psi(x)$ (בעיה Σ_2^P-שלמה).

<u>Π_2^P</u>: מוגדרת בצורה דומה עם החלפת ה-$\forall$ עם ה-$\exists$. מתקיים: $L \in \Sigma_2^P \Leftrightarrow \bar{L} \in \Pi_2^P$.

דוגמא: $MAX - IS = \{< G, k > \mid G' s max. ind. set is exactly of size k\}$. ניתן להביע את התנאי בשתי הצורות הנ"ל, שייך ל-$\Sigma_2^P \cap \Pi_2^P$, הקרויה גם DP. בעיה זו היא DP-שלמה.

ניתן להמשיך ולהוסיף תנאי $\exists, \forall$ ולהגדיר Σ_3^P, Π_3^P וכן הלאה, והיררכיה זו קרויה ההיררכיה הפולינומיאלית – PH.

משפט: $P = NP \Leftrightarrow \Sigma_2^P = P$

<u>הוכחה</u>: כיוון אחד ברור. אם נניח ש-$P = NP$, ותהי $L \in \Sigma_2^P$ אז $x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}. \forall v \in \{0,1\}^{p(|x|)}. M(x,u,v) = 1$. נגדיר $L' = \{< x, u > | u \in \{0,1\}^{p(|x|)}, \forall v \in \{0,1\}^{p(|x|)}. M(x,u,v) = 1\}$. מהגדרה $L' \in coNP = P$, ונניח M' הוא האלג' המוודא של L'. מכאן: $x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}. M'(x,u) = 1$. באופן דומה זו שפה ב-$NP$ ולפי הנחה ב-P.

משפט הירכית הזמן:

לכל פונקציות f, g כך ש-(g גדולה משמעותית יותר מ-f) $g(n) = \omega(f(n) \cdot log\, f(n))$ מתקיים: $DTIME(f(n)) \subsetneq DTIME(g(n))$.

<u>הוכחה</u>: נגדיר בעית החלטה: בהינתן אלג' α האם הפעלת U (המ"ט האוניברסלית) לסמלץ את M_α על α למשך $g(|\alpha|)$ צעדים עוצרת? מהגדרה, הבעיה ב-$DTIME(g(n))$, נראה שאינה ב-$DTIME(f(n))$. חסר.

<u>תרגול</u>:

<u>3-צביעה</u>:

- בעיית הכרעה: בהינתן גרף G, האם ניתן לצבוע את צמתיו בשלושה צבעים $\{1,2,3\}$ כך שכל שני צמתים סמוכים צבועים שונה?
- בעיית חיפוש: בהינתן גרף G, מצא צביעה לקודקודיו בשלושה צבעים $\{1,2,3\}$ כך שכל שני צמתים סמוכים צבועים שונה.

תרגיל: למצוא רדוקצייה עצמית ($Cook$) מבעיית החיפוש לבעיית ההכרעה (כלומר: בהינתן אלג' מכריע ל-$3\text{-}col$, למצוא אלג' המוצא צביעה).

פתרון: בהינתן G תחילה נבדוק האם $G \in 3-col$, אם לא אז נדחה. אם כן, נוסיף לו שלושה צמתים חדשים $\{x,y,z\}$ שנחברם במשולש ונסמן גרף זה G'. לכל $v \in V$ נפעיל את אלג' ההכרעה על שלושה גרפים עם צמתי G' וקשתות G', כאשר לראשון נוסיף הקשתות $\{v,x\},\{v,y\}$, לשני $\{v,x\},\{v,z\}$ ולשלישי $\{v,y\},\{v,z\}$. מתוך הגרפים האלה נבחר גרף כלשהו G_i $(1 \le i \le 3)$ שהוא 3-צביע ונסמנו G'. הצבע של הצומת v יהיה i כאשר G_i הוא הגרף שנבחר מהאיטרציה על v. נכונות: G' 3-צביע ולכן ל-v צביעה בצבע מבין 1,2,3, ולכן הוספת 2 קשתות לצמתים מבין x,y,z שאינם צבועים בצבע זהה, תשאיר את הגרף 3-צביע לפי אותה צביעה.

תרגיל: אם קיימת שפה אונארית NP-שלמה אז $P=NP$.

הוכחה: תהא $L \subseteq \{1\}^* \ni NPC$, ולכן $SAT \le_p L$. תהא f הרדוקציה שלכל φ מחזירה $f(\varphi)=1^i$ כאשר $i \le p(n)$ כאשר p פולינום ו-n מספר משתני הנוסחה φ / אורך φ. נראה אלג' ל-SAT הרץ בזמן פולי' ומכך נסיק $P=NP$. נשתמש במערך A שכל ערכיו מאותחלים ל-$unknown$. נגדיר את $SAT(\varphi(x_1,\dots,x_n),A)$:

- אם $n=0$ החזר את φ (t או f).
- אם $A\left[|f(\varphi)|_{=i}\right] \neq unknown$, החזר את $A[|f(\varphi)|]$.
- אם $SAT(\varphi(t,x_2,\dots,x_n),A)$ או $SAT(\varphi(f,x_2,\dots,x_n),A)$ נשים t ב-$A[|f(\varphi)|]$ ונחזיר t, אחרת נשים f ונחזיר f.

הרעיון: עבור קלט בגודל n יתכנו 2^n קלטים אפשריים, אך f ממפה אותם לקבוצה קטנה יחסית – בגודל $p(n)$ (כי לכל $x \in L$ i היחיד באורך i הוא 1^i ואין עוד קלטים אחרים אפשריים באורך i). לכן שמירת הערכים ב-A תחסוך בדיקות בהמשך = תחסוך התפצלויות בעץ האלג' הנאיבי.

נכונות: נובעת מכך ש-$\varphi(x_1,\dots,x_n)$ ספיקה אמ"מ $\varphi(t/f,x_2,\dots,x_n)$ ספיקה. זמן ריצה: בכל שלב נבחר צומת שמתאים לקריאה רקורסיבית בעץ הרקורסיה (לא עלה) ונסיר אותו ואת המסלול המוביל אליו מהשורש: $O(n)$ צמתים. נעשה זאת עד שהעץ יתרוקן. כל קריאה רקורסיבית כזו הנמצאת בתחתית המסלול מתאימה לערך שונה מהמערך A, ולכן לכל היותר יוסרו $p(n)$ מסלולים. סה"כ: $O(n \cdot p(n))$. בכל קריאה כזו מפעילים את f ולכן סה"כ $O(p^2(n)\cdot n)$.

<u>אורקלים ומגבלת הלכסון</u>:

אורקל:

תהא $O \subseteq \{0,1\}^*$ שפה כלשהי. אלג' עם גישה לאורקל O הוא אלג' שיש לו פקודה המאפשרת לו לפתור את O בצעד אחד של קריאה לאורקל. אלג' זה יסומן M^O. בצורה דומה נגדיר אלג' לא דטר' עם גישה לאורקל ומכאן את המחלקות P^O, NP^O.

טענה: $\overline{SAT} \in P^{SAT}$. הוכחה: הפעלת האורקל על קלט φ והחזרת התשובה ההפוכה.

טענה: עבור $O \in P$ מתקיים $P^O = P$. הוכחה: $P \subseteq P^O$ - ברור. $P^O \subseteq P$ - ניתן לסמלץ את האורקל ע"י מכונה ב-P ולהישאר בזמן פולי'.

טענה: $NP^{EXPCOM} = P^{EXPCOM} = EXP$; $EXPCOM = \{< \alpha, x, 1^n > | M_\alpha(x)=1, runs\ in\ 2^n\ steps\}$

<u>הוכחה</u>:

1) <u>$EXP \subseteq P^{EXPCOM}$</u>: תהי $L \in EXP$ עם מכונה M_α הפותרת את L בזמן 2^{n^c} עבור c כלשהו. נבנה אלג' המשתמש ב-$EXPCOM$ הרץ בזמן פולי' באורך הקלט: על קלט x מחזיר את תשובת האורקל ל-$< \alpha, x, 1^{n^c} >$.

2) <u>$P^{EXPCOM} \subseteq NP^{EXPCOM}$</u>: ברור.

3) <u>$NP^{EXPCOM} \subseteq EXP$</u>: בזמן אקספ' ניתן לעבור על כל האפשרויות הלא-דטר' גם עם האורקל.

משפט *Baker-Gill-Solovay*: קיים אורקל A כך ש-$P^A \neq NP^A$

<u>הוכחה</u>: לכל שפה A נגדיר $U_A = \{1^n | \exists x \in A. |x|=n\}$. $U_A \in NP^A$ מתקיים כי העד יכול להגיד לנו איפה נמצא את המחרוזת $x \in A$. נותר לבנות A כך ש-$U_A \in P^A$.

<u>תרגול</u>:

<u>טענה</u>: $EXP^{EXP} \neq EXP$.

<u>הוכחה</u>: נראה ש-$DTIME\left(2^{2^n}\right) \subseteq EXP^{EXP}$ ואז לפי משפט היררכית הזמן: $EXP \subsetneq EXP^{EXP}$. תהא $L \in DTIME(2^{2^n})$ אז קיימת מ"ט M הרצה בזמן $c \cdot 2^{2^n}$ ומכריעה את L. נראה כי $L \in EXP^{EXPCOM} \subseteq EXP^{EXP}$: בהינתן קלט x ניצור את הקלט לאורקל $< M, x, 1^{2^{n+c'}} >$ ונחזיר את תשובת האורקל. נשים לב כי $\forall x.\ x \in L \Leftrightarrow < M, x, 1^{2^{n+c'}} > \in EXPCOM$. זמן הריצה של M חסום ע"י 2^{2^n} ולכן $EXP \subsetneq EXP^{EXP}$.

<u>סיבוכיות מקום</u>:

פרק שני: סיבוכיות מקום:

תהי $t: \mathbb{N} \to \mathbb{N}$ פונקצית סיבוכיות מקום (מונוטונית לא יורדת).

$SPACE(t(n))$: מחלקת השפות שניתנות להכרעה במקום $O(t(n))$ ע"י מ"ט דטרמניסטית.

$NSPACE(t(n))$: מחלקת השפות שניתנות להכרעה במקום $O(t(n))$ ע"י מ"ט לא דטרמיניסטית.

המחלקות *L*, *NL*:

$L = SPACE(log(n)), NL = NSPACE(log(n))$. נגדיר מ"ט בעלת שלושה סרטים באופן הבא:

1. סרט קלט: קריאה בלבד.
2. סרט עבודה: קריאה וכתיבה. גודל סרט זה קובע את סיבוכיות המקום.
3. סרט פלט: לכתיבה בלבד.

קונפיגורציות:

מספר הקונפיגורציות עבור קלט בגודל N וסרט עבודה בגודל S: $\underset{\text{המכונה מצב}}{|Q|} \times \underset{\substack{\text{מיקום הראש}\\ \text{בסרט העבודה}}}{S} \times \underset{\text{תוכן סרט העבודה}}{|\Gamma|^S} \times \underset{\substack{\text{מיקום הראש}\\ \text{בסרט הקלט}}}{N} \times \underset{\text{תוכן סרט הקלט}}{|\Sigma|^N}$

דוגמאות לשפות ב-L: ... $a^n b^n, a^n b^n a^n, a^n b^{2n} a^n, palindrome, a^n b^{2n} a^{4n} b^{8n}$.

שאלה פתוחה: האם קיימת שפה ב-$NL \backslash L$; האם $NL \subsetneq P$.

רדוקצית *log-space Karp*:

$A \leq_L B$ אם קיימת פונ' $f: \Sigma^* \to \Sigma^*$ הרצה במקום לוגריתמי כך שלכל w: $w \in A \Leftrightarrow f(w) \in B$.

משפט:

המחלקות $EXP, PSPACE, NP, P, NL, L$ סגורות תחת רדוקצית מקום לוגריתמי.

רדוקציה לוגריתמית במקום מ-A אל B: מסמלצים ריצה של f (סרט ה-out לא חסום במקום, לכן ניתן לסמלץ צעד צעד, ולשמור על שימוש במקום לוגריתמי).

בעיית *CONN*:

בהינתן גרף G וקודקודים s,t, האם קיים מסלול מ-s ל-t. $CONN \in NL$:

להלן אלג' במקום לוגריתמי לא דטר': יהי $u = s$ קודקוד התחלתי נבצע $|V|$ איטרציות: בכל שלב u בוחר שכן כלשהו וממשיך אליו. אם $u = t$, נקבל, אחרת נמשיך. אם לא הגענו ל-t, נדחה. את u ואת i (מספר הצעדים שביצענו עד כה, עד שנגיע ל-$|V|$) מחזיקים ב-$log\ |V|$ מקום.

NL-TM:

1. סרט קלט: קריאה בלבד.
2. סרט עבודה: קריאה וכתיבה.
3. סרט עדות: לקריאה בלבד, רק משמאל לימין, ללא אפשרות לחזור אחורה.

CONN* ∈ *NL* − *Complete:

בהינתן מכונה M וקלט x ניצור את G, s, t: נגדיר קונפיגורציה מקבלת: נמחק את סרט העבודה, ונגדיר קונ' זו כמקבלת. $G_{M,x}$ יהיה גרף הקונפיגורציות, שכל קודקוד בו יתאים לקונפיגורציה של M על הקלט x. מס' הקונ' הוא פולי' באורך x. קשת (u, v) תהיה ב-G אם יש מעבר $u \to v$ לפי δ (פונ' מעברים לא דטר') ב-M, כלומר מעבר קונפיגורציות חוקי. s יהיה הקונ' ההתחלתית ו-t תהיה הקונ' היחידה (כפי שהוגדרה לעיל) המקבלת.

$G_{M,x}$ ניתן ליצירה במקום לוגריתמי: נחזיק את הגרף במט' שכנויות התופסת מקום לינארי, אך נעבוד על קשת אחת בכל זמן נתון לפני הדפסתה לסרט הפלט – ולכן נזדקק לסרט עבודה לוגריתמי בלבד.

נותר להוכיח כי $\forall M, x: M\ accepts\ x \Leftrightarrow \exists s \to t\ in\ G_{M,x}$.

משפט *Savitch*:

$\forall S(n) \geq log(n): NSPACE\big(S(n)\big) \subseteq SPACE(S^2(n))$: כל מכונה המשתמשת במקום $S(n)$ לא דטר', ניתן לסמלץ ע"י מכונה דטר' עם $S^2(n)$ מקום. להלן אלג' דטר' ל-$CONN$ המשתמש ב-$log^2(n)$ מקום:

נראה אלג' הבודק האם בגרף מכוון G קיים מסלול מ-u אל v באורך לכל היותר d, ואז נבדוק על $s, t, d = |V|$: אם $(u, v) \in E$, נקבל. אחרת, אם $d = 1$, נדחה. אחרת, נבדוק האם לכל $w \in V$ מתקיים האלג' על $u, w, \left\lceil \frac{d}{2} \right\rceil$ ועל $w, v, \left\lfloor \frac{d}{2} \right\rfloor$. אם כן, נקבל. אחרת נדחה. עומק הרקורסיה באלג' הוא $log\ |V|$. בכל רמה של הרקורסיה צריך לשמור w – דורש מקום לוגריתמי בלבד. סה"כ: $log^2 |V|$ מקום.

<u>תרגול</u>:

$2SAT \in coNL - Complete$:

<u>$\overline{2SAT} \in NL$</u>:

בהינתן $\varphi \in 2CNF$ בעלת n משתנים, נבנה גרף עם $2n$ משתנים (לכל משתנה ושלילתו). נייצג את האילוצים בגרף ע"י קשתות מתאימות באופן הבא: כדי שהסגר כלשהו יסופק צריך להתקיים שאם שלילת הליטרל הראשון מקבל t, אז הליטרל השני צריך לקבל t. כמו כן אם שלילת הליטרל השני מקבלת t, הליטרל הראשון צריך לקבל t (כדי שיהיה לפחות ליטרל אחד מתוך ה-2 בהסגר שמקבל t). לכל הסגר $(x_1 \vee x_2)$ נסמן אם כן בגרף את הקשתות $(\overline{x_1}, x_2)$ ו-$(\overline{x_2}, x_1)$.

<u>טענה</u>: ל-φ **אין** השמה מספקת $\Leftrightarrow$ קיים משתנה x_i שיש מסלול ממנו ל-$\overline{x_i}$ (ומסלול הפוך גם כן). אם הטענה נכונה, קל להראות ש-$\overline{2SAT} \in NL$:
ננחש באופן אי דטר' את x_i, וננחש באופן לא דטר', כמו באלג' המכריע את $CONN$, מסלול מ-x_i אל $\overline{x_i}$ ולהיפך. אלג' זה עובד עם מקום $log\ (n)$ לא דטר'.

<u>הוכחת הטענה</u>:

אם קיים i כך שיש מסלול מ-x_i אל $\overline{x_i}$ ולהיפך, אז לפי הגדרת הגרף כל צומת שניתן להגיע אליו מ-x_i, אם $x_i = t$ אז גם אותו צומת מקבל t. לפיכך נקבל כי גם $\overline{x_i} = t$, וכמו כן אם $\overline{x_i} = t$ אז $x_i = t$, וזו סתירה. לפיכך ל-φ אין השמה מספקת.

נניח כעת כי לכל i לא קיים מסלול מ-x_i אל $\overline{x_i}$. כאמור, אם $x_i = t$, כל x_j שיש מסלול מ-x_i אליו מקבל t. נראה שלא יתכן שנקבל סתירה. נניח כי במתן ערכי אמת ל-x_i, x_k קיבלנו סתירה לערכו של x_j. מכאן שקיימים מסלולים מ-x_i ל-x_j ומ-x_k ל-$\overline{x_j}$. לפיכך, יש גם מסלול מ-x_j ל-$\overline{x_k}$, ולכן מסלול מ-x_i אל $\overline{x_k}$. לכן, כאשר קבענו את ערכו של x_i להיות t, קבענו בכך את ערכו של x_k להיות f, ומכאן שלא היתה סתירה בקביעת x_j.

<u>$CONN \leq_L \overline{2SAT}$</u>:

בהינתן G,s,t נגדיר את φ באופן הבא: $\varphi = (s \vee s) \wedge (\bar{t} \vee \bar{t}) \wedge_{(u,v) \in E} (\bar{u} \vee v)$. אם קיים מסלול בין s ל-t, אז כדי ש-φ תסופק, s חייב להיות t ו-t חייב להיות f וכל צומת שיש מסלול מ-s אליו חייב להיות t. לכן, לא יתכן שיש השמה מספקת ל-φ. אם אין ב-G מסלול מ-s ל-t, ניתן ערך t ל-s ולכל המשתנים שיש מסלול מ-s אליהם. נשים לב שאין קשת מצומת u שיש מסלול מ-s אליו לצומת v שאין מסלול מ-s אליו (אחרת היה מסלול $s \to u \to v$). לכן, כל ההסגרים מהצורה $(\bar{u} \vee v)$, אם $\bar{u} = f$ אז $u = t$, ומכיוון שיש מסלול מ-s אל v אז גם $v = t$. לכן כל ההסגרים יסתפקו.

<u>הרדוקציה במקום לוגריתמי</u>: -