סיבוכיות

הגדרות בסיסיות

<u>מחלקת P:</u>

[זו מחלקת הבעיות שקיים עבורן פתרון פולינומי (שרץ בזמן פולינומי ביחס לגודל הקלט).]

$L\in P$ [כלומר שפה L נמצאת ב- P] אם קיים אלגוריתם **המכריע** את L בזמן פולינומי.

[לרוב חושבים על שפות כעל אוסף מחרוזות בינאריות (ניתן כך לייצג כל מידע סופי – למשל לקודד גרף). למשל שפת הגרפים המכילים מסלול המילטוני מכילה קידודים של גרפים שכאלה.]

<u>מחלקת NP:</u>

[בעיקרון זו מחלקת הבעיות שקיים עבורן אלגוריתם פולינומי אי-דטרמיניסטי – אבל יש לנו הגדרות שקולות הרבה יותר נחמדות מזה.]

שפה $L\in NP$ אם קיים אלגוריתם אימות $V_L\left(x,y\right)$ ופולינום $p_L(x)$ כך ש:

- אם $x\in L$ אזי קיים **עֵד** y כך ש- $|y|\le p_L(x)$ ו- $V_L\left(x,y\right)=\mathcal{T}$.
- אם $x\notin L$ אזי לכל y, $V_L\left(x,y\right)=\mathcal{F}$.
- האלגוריתם $V_L\left(x,y\right)$ הוא פולינומי ב- $|x|+|y|$ – ולכן הוא פולינומי ב- $|x|$

  [[הרכבת פולינומים היא גם פולינום, לכן אם $|y|$ פולינום ב- $|x|$ $\left(|y|=p_L\left(|x|\right)\right)$ ו- $V_L\left(x,y\right)$ פולינום ב- $|y|$ $\left(V_L\left(x,y\right)=q\left(|y|\right)\right)$, אז זמן הריצה שלו הוא $V_L\left(x,y\right)=q\left(|y|\right)=q\left(p_L\left(|x|\right)\right)=\left(q\circ p_L\right)\left(|x|\right)$ לכן כיוון ש- $q\circ p_L$ פולינום, הוא פולינומי ב- $|x|$]].

[השאלה המרכזית של מדעי המחשב בימים אלה:

האם $P=NP$?

עדיין לא ידועה התשובה לשאלה זו (רק ידוע ש- $P\subseteq NP$).]

שאלה 1 [סגירות לאיחוד של NP]

נתונות $L_1,L_2\in NP$.

הוכיחו כי $L=L_1\cup L_2\in NP$.

<u>פתרון:</u>

משום ש- $L_1,L_2\in NP$ ידוע כי קיים אלגוריתם אימות וחסם פולינומי על גודל עד ל- L_1 ו- L_2 – נסמנם:

$$V_{L_1}(x,y) \qquad p_{L_1}(x,y)$$
$$V_{L_2}(x,y) \qquad p_{L_2}(x,y)$$

נדרש:

- אלגוריתם אימות $V_L(x,y)$
- פולינום $p_L(x)$
- עד

בהינתן מילה x, נדרוש עד שהיה מראה כי $x \in L_1$ או $x \in L_2$.
אלגוריתם האימות:

$$V_L(x,y) = V_{L_1}(x,y) \vee V_{L_2}(x,y)$$

כיוון ש-$V_{L_1}(x,y)$ וְ-$V_{L_2}(x,y)$ פולינומים ב-$|x|+|y|$, גם $V_L(x,y)$ פולינומי ב-$|x|+|y|$ [[זה בסה"כ הרצת שני האלגוריתמים, אז זמן הריצה הכולל יהיה סכום זמני הריצה, שכמובן גם פולינומי]].

אם $x \in L$, נניח בה"כ כי $x \in L_1$.
אזי קיים עד y כך ש-$V_{L_1}(x,y) = \mathcal{T}$ ולכן $V_L(x,y) = \mathcal{T}$ עבור אותו y.

אם $x \notin L$ אזי $x \notin L_1, x \notin L_2$ ולכן $V_{L_1}(x,y) = \mathcal{F}$ וְ- $V_{L_2}(x,y) = \mathcal{F}$ לכל y וכך גם $V_L(x,y)$.

גודל העד:

$$p_L(x) = \max\left(p_{L_1}(x), p_{L_2}(x)\right)$$

לכן הוא פולינומי ב-$|x|$ וסיימנו.

הערה לגבי שאר השאלות בתרגול זה

בשאר התרגול אנו נראה רק ראשי פרקים של ההוכחות (ונוכיח אותן בנפנוף ידיים אינטנסיבי), כי אנחנו רוצים להספיק להראות מגוון כמה שיותר גדול – זה **לא אומר** שניתן להוכיח בנפנוף ידיים בשיעורי הבית ובבחינות. יש לעבור על מסמך התרגול באתר כדי לראות את פרטי הפורמליזציה.

שאלה 2 [סגירות לחיתוך של NP]

עבור $L_1, L_2 \in NP$ יש להוכיח כי $L_1 \cap L_2 \in NP$.

אלגוריתם האימות יבצע:

$$V_L(x,y) = V_{L_1}(x,y_1) \wedge V_{L_2}(x,y_2)$$

<u>בעיה</u>: לאו דווקא קיים קשר בין y_1 ו- y_2.
למשל, L_1 יכולה להיות שפת הגרפים שמכילים מסלול המילטוני, ו- L_2 יכולה להיות שפת הגרפים שמכילים מס' אי-זוגי של קדקודים.
העד בראשון יהיה המסלול, והשני יהיה מס' הקדקודים – אין שום קשר בין שני סוגי המידע הללו.
איך צריך להיראות בעצם עד עבור החיתוך?
למשל בדוגמה הנ"ל – עלינו לתת גם את המסלול וגם את כמות הקדקודים.
לכן נדרוש שהעד שלנו יהיה פשוט שרשור של שני העדים – כלומר $y=\langle y_1,y_2\rangle$.

שאלה 3 [סגירות לכוכב קליני]

$$L=L_1^*\in NP$$

במקרה זה אנו מקבלים מילה ארוכה שמורכבת ממילים שונות (ויתכן גם באורך שונה).
לכן העד ראשית צריך להראות שזו בכלל מילה שמורכבת ממילים ב- L_1 – לכן הוא יכיל את המידע שייתן לנו לראות את הפירוק למילים של x. בנוסף, עליו להכיל את העדים עבור כל אחת מתתי-המילים.
כלומר העד מכיל:

- פירוק של x ל- n תתי-מילים, x_i.
- n עדים משורשרים לכל תת-מילה.
- בסה"כ: $y=\langle\langle x_1,x_2,\dots,x_n\rangle,\langle y_1,y_2,\dots,y_n\rangle\rangle$

אלגוריתם האימות:

- לכל תת-מילה x_i הרץ את $V_{L_1}(x_i,y_i)$.
    - אם קיבלת $\mathcal{F}$, החזר $\mathcal{F}$
- החזר $\mathcal{T}$ [[כלומר אם לא קיבלנו $\mathcal{F}$ בשום שלב]]

[אפקטיבית: $V_L(x,y)=V_{L_1}(x_1,y_1)\wedge V_{L_1}(x_2,y_2)\wedge\cdots$]

לכל היותר מבצעים $|x|$ הפעלות של $V_{L_1}(x_i,y_i)$, לכן האלגוריתם פולינומי ב- $|x|$.

גודל העד – יש לנו כמה אפשרויות לבצע זאת:

1. ניתן שכל x_i בעד יכיל ממש את המחרוזת x_i – כך גודל העד (לפחות החלק הראשון) הוא בוודאות $O(|x|)$.
2. ניתן לשמור בחלק הראשון של העד את **מיקומי הפיצול** (או את אורכי תתי-המחרוזות) – כל מספר שכזה ניתן לייצוג ב- $\log|x|$ סיביות, לכן הגודל הכולל הוא $O(|x|\log|x|)$.

שתי השיטות טובות לנו (שתיהן מניבות גודל פולינומי).
הראשונה עדיפה ב-worst case, אבל השנייה עדיפה ב-best case וב-average case [אבל שוב, עבורנו זה לא באמת רלוונטי].

רדוקציה פולינומית

נאמר כי שפה A ניתנת לרדוקציה פולינומית ל- B ונסמן $A \leq_p B$ אם:

- קיימת f כך ש- $x \in A \Leftrightarrow f(x) \in B$
- קיים אלגוריתם פולינומי לחישוב f.

[נקודה זו חשובה כי אחרת ניתן לבצע עבודה קשה (למשל עם זמן חישוב אקספוננציאלי) ברדוקציה עצמה ובכך לייצר מופע של בעיה קלה לפתרון – לדוגמה בשביל למצוא מסלול המילטוני בגרף, ברדוקציה נייצר רשימה של **כל** המסלולים בגרף ואז רק נצטרך לבדוק האם מסלולים הם המילטוניים או לא.]

שאלה 4 [מסלולי ומעגלי המילטון]

- $HamCycle(G)$ – קיים מעגל המילטוני ב- G.
- $HamPath(G,s,t)$ – קיים **מסלול** המילטוני מ- s ל- t ב- G.

<u>סעיף ראשון</u>: יש להוכיח: $HamPath \leq_p HamCycle$.

<u>רעיון ראשון</u>: נוסיף לגרף את הצלע (s,t).

[דוגמה למצב בו זה עובד]

כאן באמת אם $G \in HamPath$ אז $f(G) \in HamCycle$.

<u>בעיה</u>: יכול להיות G עבורו $G \notin HamPath$ אך $f(G) \in HamCycle$.

[דוגמה למצב בו זה לא עובד]

<u>רעיון שני</u>: נוסיף קדקוד בין s ל- t וקשתות ביניהם ($s \rightarrow \bullet \rightarrow t$).

<u>סעיף שני</u>: יש להוכיח: $HamPath_p \geq HamCycle$.

ניתן שלל רעיונות ובעיות בהם (או למה הם עובדים):

1. נבחר שני קדקודים שכנים ונקרא להם s וְ- t.
   ← לא יעבוד אם הם לא שכנים **על המעגל ההמילטוני**.
2. נוסיף קדקוד "מלך" s (שמחובר לכולם) וקדקוד "בור" t (שכולם מחוברים אליו).
   ← גרף עם שני קדקודים וקשת ביניהם סותר זאת.
3. נבצע את 1 על כל קשת בגרף ונבדוק OR ביניהם.
   ← מגעיל ממש אבל כנראה יעבוד.
4. נוסיף s שנחבר לקדקוד כלשהו v, נוסיף t, ולכל $u \in V$ כך ש- $(u,v) \in E$ נוסיף קשת (u,t).

← יכול להיות שיעבוד; אנחנו לא מצליחים לסתור זאת כרגע.

5. נפצל קדקוד כלשהו v לקדקוד נכנס v_{in} (כל הקשתות שנכנסו ל-v יכנסו אליו) וקדקוד יוצא v_{out} (כל הקשתות שיצאו מ-v יצאו ממנו) **ולא** נוסיף קשת ביניהם.

← זה יעבוד, וזה גם הפתרון הרשמי.

שאלה 5 [At Most 3SAT]

At Most 3SAT = קבוצת כל נוסחאות ה-CNF φ כך שבכל פסוקית של φ יש לכל היותר 3 ליטרלים.

[CNF = נוסחה של פסוקיות המכילות רק פעולות OR, שביניהן יש רק פעולות AND.]

הוכיחו:

$$\text{At Most 3SAT} \le_p \text{3SAT}$$

[3SAT = אותו הדבר, אבל שיש **בדיוק** 3 ליטרלים]

פתרון:

ניקח את הנוסחה ונשנה אותה כך:

- פסוקית עם 3 ליטרלים נשאיר כפי שהיא.
- פסוקית עם שני ליטרלים מהצורה $(a_1 \vee a_2)$ נהפוך להיות $(a_1 \vee a_2 \vee y_1) \wedge (a_1 \vee a_2 \vee \bar{y}_1)$.
- פסוקית עם ליטרל אחד מהצורה (a_3) נהפוך להיות $(a_3 \vee y_1 \vee y_2) \wedge (a_3 \vee \bar{y}_1 \vee y_2) \wedge (a_3 \vee y_1 \vee \bar{y}_2) \wedge (a_3 \vee \bar{y}_1 \vee \bar{y}_2)$.

מדוע זו רדוקציה פולינומית?
כי מס' הליטרלים (שזה גודל הקלט) בנוסחה גדל לכל היותר פי 12.

[אגב, אם קלט של בעיה הוא **מספר** n, ויש רדוקציה f שמקיימת כי $|f(n)| = n$ (הגודל של התוצאה של f הוא n) אז מה סדר גודל הרדוקציה? האם היא פולינימית?
תשובה: לא! כי קלט שהוא מספר n לוקח $\log n$ סיביות, לכן $n = 2^{\log n}$ – כלומר הרדוקציה אקספוננציאלית!]

משפט [שייכות ל-NP לפי רדוקציה]

תהי $L \in NP$.
אם $L' \le_p L$ אזי $L' \in NP$.

הוכחה:
אלגוריתם אימות:

נדרוש עד y המוכיח כי $f(x) \in L$ ונקרא לעד $V_L(f(x), y)$.

מנכונות הרדוקציה,:

$$x \in L' \Leftrightarrow f(x) \in L$$

ולכן $V_L\left(f(x), y\right) = \mathcal{T}$.

העד y פולינומי ב- $f(x)$ וְ- $V_L\left(f(x), y\right)$ גם פולינומי ב- $f(x)$.

מדוע y וְ- $f(x)$ פולינומים ב- x?

$f(x)$ פולינומי ב- x כי f פולינומית [ולכן גם היא **בכל צעד** בריצה שלה כותבת ערך לפלט, אורך הפלט שלה הוא לכל היותר פולינומי ב- x].

שאלה 6 [צביעה]

צביעה היא פונקציה $c: V \to \{1, \ldots, k\}$.

צביעה חוקית היא צביעה בה $c(u) \neq c(v)$ לכל $(u,v) \in E$.

$$4-Color = \{\ G : \text{גרף צביע ב-4 צבעים } G\ \}$$

הוכיחו כי $4-Color \in NP$.

<u>אלגוריתם אימות:</u>

- עד: צביעה חוקית
- בדוק כי הצביעה מכילה 4 צבעים לכל היותר ובדוק כי כל 2 קדקודים סמוכים שונים בצבעם.
זה לוקח $O\left(|E|\right)$ כי בדיקת 2 קדקודים סמוכים = בדיקת הקשת שביניהם.

<u>$3-Color \le_p 4-Color$:</u>

נגדיר שפונקציית הרדוקציה מוסיפה קדקוד חדש ומחברת אותו לכל הגרף.

- אם הגרף המקורי 3-צביע אז ניתן לצבוע את הקדקוד הרביעי בצבע נוסף ולכן הגרף המתקבל הוא 4-צביע.
- אם הגרף המקורי אינו 3-צביע אז חייבים לפחות 4 צבעים כדי לצבוע אותו.
כיוון שהקדקוד החדש מחובר לכל הקדקודים הקודמים, בכל צביעה ב-4 צבעים של הגרף המקורי הקדקוד החדש מחובר לקדקודים בכל ארבעת הצבעים הקיימים, לכן חייבים להשתמש בצבע חדש עבורו – כלומר הגרף החדש דורש לפחות 5 צבעים.