19/10/20

omri

①

סיבוכיות חישוב / ערן עומרי

ספר הקורס: Computational Complexity - A Modern Approach / Sanjeev Arora, Boaz Barak

בקורס כה לא נלמד מה ניתן ומה לא ניתן לחשב. אלא נרצה לכמת את יעילות החישוב. הנושאים שנדון בהם בקורס הם:

- סיבוכיות זמן ריצה - מספר פעולות נדרשות לאלגוריתם. חלוקה למחלקות זמן (P, NP, וכו').
- סיבוכיות זיכרון - מספר תאי זיכרון הנדרש לאלגוריתם. חלוקה למחלקות זכרון.
- אלגוריתמים אקראיים
- מעגלים אריתמטיים ולוגיים.
- קירובים. קושי קירוב.
- PCP
- קריפטוגרפיה.

נושא 1 - סיבוכיות זמן

א) שפות וקידודים

א"ב Σ - קבוצה סופית לא ריקה של תווים. בד"כ $\Sigma=\{0,1\}$.

מילה - מחרוזת סופית של תווים מעל Σ.

שפה - תת קבוצה של Σ^*. יכולה להיות ריקה $\emptyset$, סופית, או אינסופית.

פונקציה - אנו נדבר על פונקציות מהצורה $f:\Sigma^*\to\Sigma^*$. כל פונקציה כזו מייצגת בעיה ופתרון, הפונקציה מקבלת מופע של בעיה ומחזירה פתרון. עבור בעיות הכרעה הפונקציה תהיה $f:\Sigma^*\to\{0,1\}$. נשים לב: שבעיות הכרעה ניתן לייצג כשפות. רק שאם התשובה לבעיה היא כן אזי המופע של הבעיה שייך לשפה.

ב) מכונת טיורינג

אנו מתייחסים למודל כללי שבו יש k סרטים עבור k קבוע. בד"כ יש סרט קלט, סרט פלט וסרט עבודה. מכונת טיורינג היא שלשה (Γ, Q, δ).

Γ - א"ב המכונה. בפרט $\Gamma=\Sigma\cup\{\triangleright,\sqcup\}$, כאשר Σ הוא א"ב הקלט, $\triangleright$ מציין את התא הראשון בסרט, ו-$\sqcup$ תא ריק.

Q - קבוצת מצבים סופית לא ריקה. בפרט $q_{start},q_{halt}\in Q$, כאשר q_{start} מצב התחלתי ו-q_{halt} מצב סופי.

δ - פונקציית מעברים. $\delta: Q\times\Gamma^k\to Q\times\Gamma^{k-1}\times\{L,S,R\}^k$, שאומרת בהינתן: מצב ותוכן ב-k הסרטים, לאיזה מצב לעבור ומה לכתוב ב-k הסרטים למעט סרט הקלט ולאן יזוזו k הראשים הקוראים/כותבים.

②

קונפיגורציה התחלתית: המכונה במצב q_{start}, בסרט הקלט יש $\rhd$ ולאחריו תווי הקלט המורכבים מ-Σ בלבד. בסרט העבודה והפלט יש $\rhd$ ולאחר מכן רצף אינסופי של $\sqcup$. כל צעד חישוב מוגדר באופן דטרמיניסטי ע"י δ.

קונפיגורציה סופית: כאשר המכונה מגיעה למצב q_{halt} היא עוצרת ומחזירה את כל המחרוזת בסרט הפלט מימין ל-$\rhd$ ועד ל-$\sqcup$ שאחריו יש רק $\sqcup$.

חישוב פונקציה: נאמר שמ"ט M מחשבת פונקציה $f:\Sigma^* \to \Sigma^*$ אם לכל קלט $x\in\Sigma^*$ M מריצה את x ולאחר מספר סופי של צעדי חישוב בסרט הפלט כתובה המחרוזת $f(x)$. מקובל לסמן את $f(x)$ גם ב-$M(x)$. אנו בקורס זה נתמקד במכונות טיורינג שמכריעות שפות L. מ"ט אלו מחשבות את הפונקציה $f:\Sigma^* \to \{0,1\}$, כך שעבור כל קלט $x\in\Sigma^*$ מתקיים $x\in L \iff f(x)=1$. כלומר המכונה תחזיר 1 אם ורק אם המילה x נמצאת ב-L, אחרת תחזיר 0. אנו בקורס זה מדברים רק על מ"ט שעוצרות תמיד!

זמן ריצה: נאמר שמ"ט M רצה בזמן $T(n)$ אם קיימת פונקציה $T:\mathbb{N}\to\mathbb{N}$ כך שלכל קלט $x\in\{0,1\}^n$ כאשר $n\in\mathbb{N}$, M רצה על x ומסיימת את הריצה שלה לכל היותר $T(n)$ צעדי חישוב. כלומר $T(n)$ זהו חסם עליון למספר צעדי החישוב ש-M עובדת על כל קלט באורך n.

באופן כללי אנו מגדירים חישוב יעיל עבור מכונות טיורינג שזמן הריצה שלהם הוא $O(n^i)$ עבור $i\in\mathbb{N}$ כלשהו. כלומר המ"ט רצה בזמן פולינומיאלי לאורך הקלט. כמובן שאם זמן הריצה הוא $O(n^{100})$ זהו זמן ריצה לא סביר ומניסיוננו בעולם לא קיים. אך הגדרה זו די טובה שכן כמעט תמיד חישוב פולינומיאלי לא יגיע לגדלים כאלו.

ג) שקילות מודלים

ישנם מספר מודלים השקולים למכונת טיורינג וכן ישנם המון סוגי מודלים של מכונת טיורינג. תזת צ'רץ'-טיורינג מחדשת לנו שכל מודל החזק לפחות כמו מכונת טיורינג שלא מכיל רכיבים אינסופיים שקול למכונת טיורינג. לכן אם יש במ"ט סרט אחד או k סרטים עדיין המודלים שקולים, אמנם יכול להיות שזמן הריצה יהיה עד $O(n^2)$ יותר.

ד) מכונת טיורינג אוניברסלית

קידוד מ"ט: כל מ"ט ניתן לייצג במחרוזת בינארית. נסמן קידוד של מ"ט M ב-$\langle M\rangle$. שתי טענות חשובות:

(1) כל מחרוזת בינארית $\alpha\in\{0,1\}^*$ מייצגת מכונת טיורינג כלשהי: M_α.

(2) לכל מכונת טיורינג M ישנן אינסוף מחרוזות α כך ש-$M=M_\alpha$.

אחר שסיימנו את הקידוד ניתן לרשום מה שרוצים זבל לא יהיה לכן יש ∞ קידודים.

ישנם כללים כיצד לקודד מ"ט. אם הקידוד עומד בכללים ניתן לפענח אותו בצורה יחידה. גם אם לא עומדת בהם נאמר שהיא מ"ט טריוויאלית.

בס"ד

21/10

<u>מכונת טיורינג אוניברסאלית U</u>: היא מ"ט שיודעת לקבל כקלט קידוד של מ"ט $\langle M\rangle$ ומילה x, ומריצה (מסמלצת) את M על x ועונה כמוה. אם זמן הריצה של M הוא $T(\cdot)$ זמן הריצה של M על x הוא $O(T(|x|)^2)$.
המחשב שלנו בעצם מהווה סוג של מכונת טיורינג אוניברסאלית שכן הוא יודע לקבל תוכנה וקלט ולהריץ את התוכנה על הקלט ולפלוט את מה שהתוכנה פלטה.

ה) המחלקה DTIME

הגדרה: עבור פונקציה $T:\mathbb{N}\to\mathbb{N}$ ושפה $L\subseteq\{0,1\}^*$, נאמר כי $L\in DTIME(T(n))$ אם קיימת מ"ט דטרמיניסטית M הרצה בזמן $O(T(n))$ ומכריעה את L. כלומר יש מחלקת DTIME עבור כל פונקציה $T(n)$ המכילה את כל השפות שקיימת עבורן מ"ט המכריעה אותן בזמן ריצה של $T(n)$.

ו) המחלקה P

המחלקה P מוגדרת כך: $P=\bigcup_{i\in\mathbb{N}} DTIME(n^i)$.
כלומר המחלקה P היא קבוצת כל השפות $L\subseteq\{0,1\}^*$ שקיימת עבורן מכונת טיורינג המכריעה אותן בזמן פולינומיאלי לגודל הקלט, המוגדר כ-$O(n^i)$ עבור קלט באורך $n\in\mathbb{N}$ ו-$i\in\mathbb{N}$ כלשהו.
המחלקה P מייצגת את כל השפות עבורן קיים חישוב יעיל. באופן כללי ניתן להרחיב הגדרה זו שלא תכיל רק שפות אלא את כל הפונקציות הניתנות לחישוב בזמן פולינומיאלי לגודל הקלט. בהמשך נלמד מדוע ניתן לעשות זאת (אם יש פתרון יעיל לבעיית הכרעה יש פתרון יעיל גם לבעיית חיפוש).
שאלה חשובה היא במקרה של שפה $L\in P$ אם זמן הריצה של המ"ט הטובה ביותר שמכריעה את L הוא $O(n^{500})$, האם זה עדיין נחשב חישוב יעיל? למרות שזמן ריצה זה לא מציאותי עדיין נאמר ש-L יש חישוב יעיל בגלל שהחלוקה של המחלקה P מאוד יציבה ומועילה לשימ"ם טכנולוגי"ם, וגם בגלל שכמעט תמיד הקבועים בחזקה בשפות השייכות ל-P הם מאוד קטנים.
⊗לפי התזה המורחבת של צ'רץ'-טיורינג כל מודל חישוב שניתן למימוש פיזי יגדיר את אותה מחלקה P.
<u>דוגמאות:</u> (1) $st\text{-}con=\{\langle G,s,t\rangle:$ קיים מסלול ב-G מ-s ל-$t\}\in P$. שכן קיימים אלגוריתמים לינאריים BFS ו-DFS שמכריעים את השפה $st\text{-}con$.
(2) תכנון ליניארי (LP): בהינתן M אי-שוויונים מעל המשתנים $x_1,x_2,\dots,x_n$, בעיית ההכרעה LP היא האם קיימת השמה מ-$\mathbb{R}$ ל-$x_1,x_2,\dots,x_n$ אשר מספקת את כל האי-שוויונים (אילוצים). $LP\in P$, שכן קיימים כל מיני אלגוריתמים פולינומיים שעושים זאת (סימפלקס, אליפסואיד, וכו').

26/10
④

בס"ד

ג) המחלקה NP

שפה $L\subseteq\{0,1\}^*$ שייכת ל-NP אם קיים פולינום $p(\cdot)$ וקיימת מ"ט M_L שזמן הריצה שלה פולינומי, כך שלכל $x\in\{0,1\}^*$ מתקיים: $x\in L \iff \exists y\in\{0,1\}^{p(|x|)}$ s.t $M_L(x,y)=1$.

כלומר, עבור כל קלט $x\in L$ יש עד y באורך פולינומי ל-$|x|$ כך שמכונת טיורינג פולינומית יכולה להכריע ש-y מעיד על כך שאכן $x\in L$. במילים אחרות, NP היא קבוצת כל השפות L כך שלכל מילה x ב-L ישנו y שניתן להוכיח באמצעותו בזמן פולינומי שאכן $x\in L$.

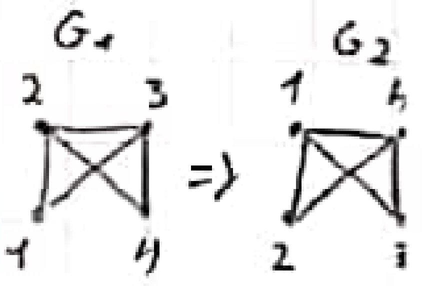
הגדרה נוספת היא באמצעות מ"ט א-דטרמיניסטית עליה נדבר בסעיפים הבאים.

<u>דוגמאות:</u> (1) $INDSET=\{\langle G,k\rangle :$ ב-G יש קבוצה בת"ל בגודל $k\}\in NP$. אם $\langle G,k\rangle\in INDSET$ אזי ניתן למצוא עד: y המכיל את k הצמתים ב-G שהם קבוצה בת"ל (יכול להיות גם וקטור שיש בו 1 באינדקס של וקטור השייך לקבוצה הבת"ל, או 0 אם לא שייך). מכונת טיורינג M המקבלת $\langle G,k,y\rangle$ יכולה לוודא בזמן פולינומי ש-y אכן קבוצה בת"ל בגודל k, ע"י מעבר על כל $\binom{k}{2}$ זוגות הצמתים ב-y ולוודא שאין ביניהם צלע.

(2) $HAM=\{\langle G\rangle :$ ב-G יש מסלול המילטון$\}\in NP$, נקראת גם בעיית הסוכן הנוסע. העד y הוא פרמוטציה של כל הצמתים ב-G. M צריכה לעבור בסדר הצמתים בפרמוטציה ולוודא שכל שני צמתים סמוכים מחוברים בצלע.

(3) תכנון בשלמים (IP): כמו LP רק ההשמה על $x_1,\dots,x_n$ היא מספרים שלמים. $IP\in NP$ שכן ההשמה יכולה להיות עבור מספרים גדולים מאוד כך שלהזין אותם זה כבר לא פולינומי.

(4) $GI=\{\langle G_1,G_2\rangle :$ G_1 ו-G_2 מייצגים אותו גרף עד כדי שינוי שמות הצמתים$\}\in NP$ (גרפים איזומורפים). העד y הוא פרמוטציה π של הצמתים ב-G_1 כך ששינוי סדר הצמתים לפי π יתן את G_2.



$\pi(1)=2$
$\pi(2)=1$
$\pi(3)=4$
$\pi(4)=3$

(5) $FACTORIZATION=\{\langle N,L,U\rangle :$ קיים גורם ל-N בתוך הקטע $[L,U]\}\in NP$. העד y הוא מספר בין L ל-U שהוא גורם של N. בעיה זו משמשת הרבה בניות קריפטוגרפיות. במחשב קוונטי קיים עבורה פתרון יעיל לכאורה.

ח) המחלקה EXP והיחס בין P ל-NP

המחלקה EXP מוגדרת כך: $EXP=\bigcup_{i\in\mathbb{N}} DTIME\left(2^{(n^i)}\right)$. כלומר EXP היא קבוצת כל השפות שקיימת* עבורן מ"ט דטרמיניסטית המכריעה אותן בזמן ריצה של $O\left(2^{(n^i)}\right)$ עבור כל $i\in\mathbb{N}$. מתקיים: <u>$P\subseteq NP\subseteq EXP$.</u>

והי' בעצם קבוצת
ל השפות שיש להם
ט שמכריעה אותם
זמן אקספוננציאלי.
אקספוננציאלי - EXP)

- $P\subseteq NP$: תהי שפה $L\in P$, צ"ל $L\in NP$. כיוון ש-$L\in P$ קיימת מ"ט M המכריעה את L בזמן פולינומי. אזי נגדיר $y=\varepsilon$ ונשים לב שאכן מתקיים: $x\in L\iff\exists y\in\{0,1\}^0\ M(x)=1$. ולכן $L\in NP$. מ.ש.ל.
- $NP\subseteq EXP$: רעיון להוכחה. יהי $L\in NP$, אזי קיימת מ"ט M ופולינום $p(\cdot)$ כך ש- $x\in L\iff\exists y\in\{0,1\}^{p(|x|)}$ s.t $M(x,y)=1$. ישנם $2^{p(|x|)}$ עדים אפשריים. נגדיר מ"ט M' שעל הקלט $x\in\{0,1\}^*$: תעבור על $2^{p(|x|)}$ העדים האפשריים ותכניס אותם ל-M, אם עבור אחד מהעדים M תחזיר 1 אזי M' גם תחזיר 1. ומכאן ש- $L\in EXP$.*

זמן הריצה של M'
אז: $O\left(q(x)\cdot 2^{p(|x|)}\right)$
$q(x)$ זמן הריצה של M

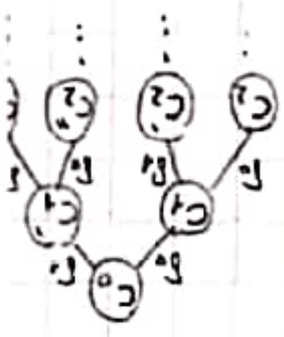
24/10 ⑤

בס"ד

שאלות פתוחות:
$P = NP$
$NP = EXP$

דוגמה: בהינתן לוח שחמט וסידור כלשהו של שחקנים, מהו הצעד הבא הטוב ביותר? הבעיה זו שייכת ל-EXP ולא ל-NP, שכן גם בהינתן צעד כלשהו לא נוכל לוודא שהוא הטוב ביותר בזמן פולינומי.

ט) מכונת טיורינג אי-דטרמיניסטית (NDTM)



מוגדרת כמו מ"ט דטרמיניסטית אלא שיש לה שתי פונקציות מעברים δ_0 ו-δ_1. בכל צעד יכולה לבחור באיזה פונקציה ללכת. ניתן לצייר את כל המסלולים של מ"ט א"ד על קלט x כעץ בינארי הנקרא $T_{M,x}$.

- נאמר שמ"ט א"ד מקבלת מילה x אם יש לפחות מסלול אחד בעץ החישוב שנגמר במצב מקבל.
- נאמר שמ"ט א"ד מכריעה שפה L אם עבור כל $x\in L$ יש לה מסלול מקבל ועבור $x\notin L$ כל המסלולים דוחים.
- שפה של מ"ט א"ד M מוגדרת: $L(M)=\{x\in\{0,1\}^*:$ קיים חישוב מקבל של M על x$\}$.
- נאמר שמ"ט א"ד רצה בזמן $T(n)$ אם לכל קלט $x\in\{0,1\}^*$, ולכל חישוב של M על x החישוב מסתיים לאחר לכל היותר $T(n)$ צעדים. כלומר, החישוב הארוך ביותר ב-M על x הוא $T(n)$.

י) המחלקה NTIME והגדרה נוספת ל-NP

תהא פונקציה $T:\mathbb{N}\to\mathbb{N}$ ותהא שפה $L\subseteq\{0,1\}^*$, נאמר ש-$L\in NTIME(T(n))$ אם קיימת מ"ט אי-דטרמיניסטית N הרצה בזמן $O(T(n))$ ומכריעה את L. כלומר, יש מחלקת NTIME עבור כל פונקציה $T(n)$ המכילה את כל השפות שקיימת עבורן מ"ט אי-דטרמיניסטית המכריעה אותן בזמן $T(n)$.

משפט: $NP=\bigcup_{i\in\mathbb{N}} NTIME(n^i)$. כלומר NP היא קבוצת כל השפות שקיימת עבורן מ"ט א"ד המכריעה אותן בזמן פולינומי ביחס לגודל הקלט, המוגדר כ-$O(n^i)$ עבור קלט באורך n ו-$i\in\mathbb{N}$ כלשהו.

הוכחה: כיוון 1 - תהי L שפה שקיימת לה מ"ט א"ד N_L המכריעה אותה בזמן פולינומי. אזי לכל $x\in L$ יש מסלול מקבל ב-N_L. נגדיר y להיות רצף בינארי של סדרת ההחלטות לפי מסלול זה (0 עבור δ_0 ו-1 עבור δ_1). נגדיר מ"ט דטרמיניסטית M_L שעבור כל קלט $\langle x,y\rangle$ תסמלץ את N_L על x לפי הסדרה y ותענה כמוה. M_L מכריעה את L שכן אם $x\in L$ אזי N_L תחזיר 1 ואם $x\notin L$ בכל מסלול N_L תחזיר 0. מסקנה $L\in NP$.

כיוון 2 - תהי $L\in NP$ אזי לכל $x\in L$ יש מ"ט דטרמיניסטית M_L ועד y כך ש-$M_L(\langle x,y\rangle)=1$. נגדיר מ"ט א"ד N_L שעבור כל קלט x, תנחש y (כלומר תגריל סדרת אפסים ואחדות הקובעת את y), ואז תריץ את M_L על $\langle x,y\rangle$ ותענה כמוה. N_L מכריעה את L שכן אם $x\in L$ יש מסלול שבו תנחש את y ואז M_L תחזיר 1, ואם $x\notin L$ אזי אין מסלול עבורו ינחש עד נכון ותמיד M_L תחזיר 0. מ.ש.ל.

יא) רדוקציות פולינומיות

הראנו גם רדוקציית Karp

שפה L_1 ניתנת לרדוקציה פולינומית לשפה L_2, ונסמן $L_1\le_p L_2$ אם קיימת פונקציה $F:\{0,1\}^*\to\{0,1\}^*$ המקיימת:

1) F ניתנת לחישוב בזמן פולינומי.
2) לכל קלט $x\in\{0,1\}^*$ מתקיים: $x\in L_1 \iff F(x)\in L_2$.

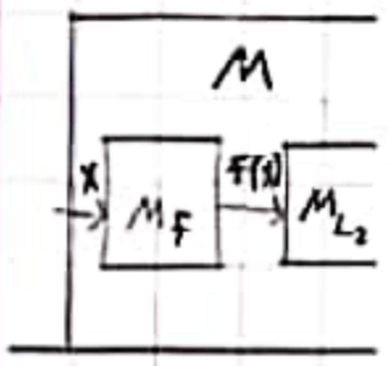
28/10

⑥

בס"ד

משפט הרדוקציה הפולינומית: אם מתק"ם: $L_1 \le_p L_2$ וגם $L_2 \in P$ אזי $L_1 \in P$.

הוכחה: כיוון ש-$L_2 \in P$ יש מ"ט M_{L_2} שמכריעה אותה בזמן פולינומי $P(\cdot)$, וכיוון ש-$L_1 \le_p L_2$ יש מ"ט M_F שמחשבת את פונקציית הרדוקציה בזמן פולינומי $q(\cdot)$. לכן נבנה מ"ט שתכריעה את L_1 בכך שתכניס כל קלט x ל-M_F ואז תכניס את $f(x)$ ל-M_{L_2} ותענה כמוה. זמן הריצה של M הוא: $T_M(|x|) = q(|x|) + P(|F(x)|)$. כיוון ש-f פולינומית גם הבדל האורך בין x ל-$f(x)$ פולינומי, סה"כ חיבור פונקציות פולינומיות הוא גם פולינומי.



תכונות רדוקציה פולינומית:

- הרדוקציה הפולינומית מקיימת רפלקסיביות, טרנזיטיביות ואי-סימטריות*.
- אם $L_1 \le_p L_2$ אזי גם $\overline{L_1} <_p \overline{L_2}$.

ו $L_1 \le_p L_2$ לא
כרח: $L_2 \le_p L_1$

יב) המחלקות NP-Hard ו-NP-Complete

- שפה $L \in NP\text{-}Hard$ אם כל שפה $L' \in NP$ ניתנת לרדוקציה פולינומית ל-L, $L' \le_p L$.
- שפה $L \in NP\text{-}Complete$ אם $L \in NP\text{-}Hard$ וגם $L \in NP$.

⊕ מקובל להתייחס למחלקות אלו בקיצור NPH ו-NPC.

משפטים: (1) אם $L \in NPH$ וגם $L \in P$ אזי $P=NP$.

(2) אם $L \in NPC$ אזי $P=NP \Leftrightarrow L \in P$.

(3) אם $P=NP$ אזי כל שפה ב-$P \setminus \{\Sigma^*, \Phi\}$ שייכת ל-NPC.*

ו,Σ^* אינן יכולות
ות NPC שכן כדי
תקיים $L' \le_p L$, פ-ו
ה להיות $x \in L$ ו-$y \notin L$

יג) השפה SAT

נוסחת CNF: זוהי נוסחה בוליאנית מעל משתנים $x_1, x_2, \ldots, x_n$. בנוסחה מופיעים ליטרלים של משתנים אלו. כל ליטרל של משתנה x_i הוא שווה למשתנה x_i או לשלילתו $\bar{x}_i$. נוסחת CNF מורכבת מפסוקיות שביניהם יש סימן $\cap$ (וגם) ובתוכן יש סימן $\cup$ (או) בין הליטרלים. נאמר שנוסחת CNF φ ספיקה אם יש השמה כלשהי של n המשתנים עבורם הנוסחה תקבל ערך 1 (true). השמה z לנוסחה φ היא וקטור $z \in \{0,1\}^n$ כאשר ההשמה למשתנה x_i להו רכיב ה-i בוקטור z.

דוגמה:
$(x_1 \cup \bar{x}_2 \cup x_4) \cap (x_3 \cup \bar{x}_4$

נוסחת k-CNF: זוהי נוסחה שבכל פסוקית בה יש בדיוק k ליטרלים.

הגדרת השפות:

$SAT = \{\varphi :$ φ נוסחת CNF שק"ימת עבורה השמה מספקת $\}$

$3\text{-}SAT = \{\varphi :$ φ נוסחת 3-CNF שק"ימת עבורה השמה מספקת $\}$

משפט קוק לוין:

(1) $SAT \in NPC$.

(2) $3\text{-}SAT \in NPC$.

בס״ד 2/11

⑦

משפט: $SAT \leq_p 3\text{-}SAT$.

הוכחה: סיכמתי בקורס אלגוריתמים 2 בפרק NPC סעיף ג׳.

הוכחת משפט קוק לוין

(1) $SAT, 3\text{-}SAT \in NP$: אם $\varphi \in SAT$ אזי נוכל להגדיר עד z שהוא ההשמה המספקת ל-φ. נגדיר מ״ט M המקבלת $\langle \varphi, z \rangle$ שתבדוק בכל הליטרלים את ההשמה שהמשתנה המתאים להם קיבל ב-z ותחזיר אם φ אכן מסופק. בדיקה זו היא בזמן פולינומי לאורך הקלט ח״ת כאשר k היא מספר הפסוקיות. באותו אופן נוכיח עבור 3-SAT.

(2) לכל $L \in NP$ מתקיים $L \leq_p SAT$:

- טענת עזר 1: ניתן לייצג השוואת מחרוזות X ו-Y באמצעות נוסחת CNF.

הוכחה: עבור $X = x_1x_2\ldots x_n$ ו- $Y = Y_1Y_2\ldots Y_n$ נגדיר נוסחת CNF: $(x_1 \cup \overline{Y_1}) \cap (\overline{x_1} \cup Y_1) \cap \ldots \cap (x_n \cup \overline{Y_n}) \cap (\overline{x_n} \cup Y_n)$

- טענת עזר 2: לכל פונקציה בוליאנית $F: \{0,1\}^n \to \{0,1\}$ קיימת נוסחת CNF: φ_F כך שלכל $x \in \{0,1\}^n$ $F(x) = \varphi_F(x)$.

הוכחה: נגדיר $A = \{x \in \{0,1\}^n : F(x) = 0\}$, לכל $x_j \in A$ נגדיר נוסחת CNF עם פסוקית אחת ו-n ליטרלים, רק שאם במקום ה-i ב-x_j יש 1 אזי ב-φ_j יהיה ליטרל $\overline{x_i}$, ואם יש 0 יהיה x_i.* נגדיר: $\varphi_F = \varphi_1 \cap \varphi_2 \cap \ldots \cap \varphi_{|A|}$.*

$x_j = 1011$

$\varphi_j = (\overline{x_1} \cup x_2 \cup \overline{x_3} \cup$

ים לב שכל פסוקית
תחזיר False רק
בור x_j.

- טענת עזר 3: לכל מ״ט M הרצה בזמן $T(\cdot)$ קיימת מ״ט $\hat{M}$ שהיא oblivious המחשבת את $L(M)$ בזמן ריצה $\underline{O(T(\cdot)^2)}$. מ״ט oblivious היא מכונה שבכל קלט תנועות הראש הקורא של סרט הקלט וסרט העבודה היא זהה.

נחזור להוכחת משפט קוק לוין. כיוון ש- $L \in NP$ אזי יש מ״ט M_L oblivious המוודאת $M_L(x,y) = 1$ אם $x \in L$ לפי y. בזמן ריצה $T(\cdot)$. נגדיר $z_i := (a, b, q)$ להיות קונפיגורציה של M_L בשלב i, כאשר a תו בסרט הקלט, b תו בסרט עבודה ו-q מצב במכונה. אורך הייצוג של z_i הוא: $|z_i| \underset{\vdots}{=} c = 2 \cdot \log|\Gamma| + \log|Q|$.

מטרתנו להגדיר פונקציה $f(x) = \varphi_{M_L,x}$ המקיימת: $\exists y \in \{0,1\}^{p(|x|)},\ M_L(x,y) = 1 \iff \exists \overset{\text{השמה}}{z} \in \{0,1\}^{\ell},\ \varphi_{M_L,x}(z) = 1$

כלומר יש השמה מספקת ל- $\varphi_{M_L,x}$ רק אם קיים עד y עבורו המכונה המוודאת מקבלת x.

נוכל להגדיר פונקציה $F(z_{i-1}, z_{prev(i)}, u_{inputPos(i)}) = z_i$, המחשבת z_i בהינתן: z_{i-1} - קונפיגורציה שהובילה ל-z_i.

$z_{prev(i)}$ - קונפיגורציה קודמת שבה הראש הקורא היה באותו מיקום כמו שלב i, $u_{inputPos(i)}$ - התו בקלט (x,y) במיקום בו נמצא הראש הקורא בשלב i. באמצעות נתונים אלו ניתן לחשב את z_i. לכל z_i נגדיר c פונקציות, כך שכל $F_j^{z_i}$ מחזיר את הביט ה-j ב-z_i כאשר הקלט שלו הוא $(z_{i-1}, z_{prev(i)}, u_{inputPos(i)})$. לפי טענת עזר 2, לכל $F_j^{z_i}$ נוכל להגדיר נוסחת CNF $\varphi_j^{z_i}$ כך שתמיד יחזירו אותו ערך. כעת נגדיר את $\varphi_{M_L,x}$ להיות

$$\varphi_{M_L,x} = \varphi_1^{z_1} \cap \varphi_2^{z_1} \cap \ldots \cap \varphi_c^{z_1} \cap \varphi_1^{z_2} \cap \ldots \cap \varphi_c^{z_2} \cap \ldots \cap \varphi_1^{z_{T(|x,y|)}} \cap \ldots \cap \varphi_c^{z_{T(|x,y|)}}$$

נוסחת CNF זו בודקת את רצף כל הקונפיגורציות ב-M_L על קלט (x,y), לכן אם M_L מחזירה 1 אז $\varphi_{M_L,x}$ תחזיר 1. אורך $\varphi_{M_L,x}$ הוא פולינומי שכן $T(\cdot)$ הוא פולינומי מהגדרת מ״ט מוודאת.

יג) הוכחת שייכות ל-NPC

משפט: אם קיימת רדוקציה פולינומית בין שפה $L^* \in NPC$ אל שפה $L \in NP$, $L^* \le_p L$, אזי גם $L \in NPC$.
ההוכחה נובעת מתכונת הטרנזיטיביות של הרדוקציה. שכן אם קיימת רדוקציה מכל שפה ב-NP אל L^*, אזי קיימת רדוקציה מכל שפה ב-NP אל L.

דוגמה: $INDSET \in NPC$. כבר הוכחנו $INDSET \in NP$, נוכיח $3\text{-}SAT \le_p INDSET$.
נגדיר פונקציה: $F(\langle\varphi\rangle) = \langle G,t\rangle$, כאשר t הוא מספר הפסוקיות ב-φ ו-$G=(V,E)$ גרף שבו:
V - יש צמתים ב-G עבור כל השמה אפשרית בכל פסוקית שלא נותנת לפסוקית ערך 0 (False). כיוון שבכל פסוקית יש 3 צמתים, יש $2^3=8$ השמות שונות לכל פסוקית ורק אחת נותנת לפסוקית ערך 0. סה"כ $7 \cdot t$ צמתים.
E - $(v,u) \in E$ רק אם בהשמה v והשמה u אינן מסכימות על משתנה כלשהו, כלומר יש משתנה x_i שהשמה v נתנה לו ערך והשמה u נתנה לו ערך נגדי. נשים לב שכל 7 הצמתים של פסוקית הם קליקה.
$\langle G,t\rangle \in INDSET \Rightarrow$ ב-G חייבים לבחור צומת אחת מכל קליקה כדי להגיע לקבוצה בת"ל בגודל t. t צמתים אלו מייצגים ביחד השמה ל-φ שבה כל t הפסוקיות מסופקות ואין סתירה ביניהם, שכן אחרת הם לא היו קבוצה בת"ל. לכן ל-φ יש השמה מספקת, ולכן $\langle\varphi\rangle \in 3\text{-}SAT$.
$\langle\varphi\rangle \in 3\text{-}SAT \Rightarrow$ יש השמה ב-φ שמספקת כל הפסוקיות. בפרט יש השמה עבור כל פסוקית כך שאין סתירה בין הפסוקיות וכל הפסוקיות מסופקות לפיהן. ניקח את t הצמתים המקבילים להשמות אלו בגרף G. t צמתים אלו הן קבוצה בת"ל שכן אחרת לא היו מספקים את φ. המסקנה ש-$\langle G,t\rangle \in INDSET$. מ.ש.ל.

כא) בין P ל-NPC

(1) משפט לדנר: אם $P \neq NP$ אזי קיימת שפה $L \in NP \setminus P$ שאינה NPC. L נקראת שפה אמצעית.
⊗ אנו מאמינים ששפות כמו גרפים איזומורפיים (GI) ופקטוריזציה הם שפות אמצעיות כאלו.

טז) בעיית חיפוש מול בעיית הכרעה

בפרק זה נוכיח באמצעות דוגמה שאם P=NP אזי בהינתן אלגוריתם פולינומי לבע"ת הכרעה ניתן למצוא אלגוריתם פולינומי לבע"ת חיפוש. עד כה כל הבעיות שראינו היו בעיות הכרעה שבהן נתון אובייקט (גרף נוסחת CNF. וכו') ושאלנו האם האובייקט הזה מקיים תכונה כלשהי. בעיות חיפוש אלו בעיות שבנוסף מחזירות את הקבוצה מתוך האובייקט המעידה את התשובה. נוכיח את המשפט עם דוגמה.

__דוגמה:__ אם P=NP רק ש- $SAT\in P$ אזי ישנו אלגוריתם פולינומי המתאים לבע"ת החיפוש של SAT.

הוכחה: אם $SAT\in P$ אזי ישנו אלגוריתם A פולינומי המכריע את SAT. בנוסף עבור נוסחת CNF φ נגדיר את φ_0 להיות הנוסחה המתקבלת מ-φ ע"י הצבת 0 במשתנה עם האינדקס הכי קטן, וכן את φ_1 להיות הנוסחה המתקבלת מ-φ ע"י הצבת 1 במשתנה עם האינדקס הכי קטן. נתאר אלגוריתם פולינומי לבע"ת החיפוש של SAT. __אלגוריתם כזה על קלט φ:__

- אם $A(\varphi)=0$ החזר "אין השמה מספקת".
- אתחל השמה $\alpha=\varepsilon$.
- כל עוד סכח (מספר המשתנים ב-φ)
  - אם $A(\varphi_0)=1$ בצע $\alpha=\alpha\circ 0$ $\varphi=\varphi_0$.
  - אחרת (כלומר $A(\varphi_1)=1$) בצע $\alpha=\alpha\circ 1$ $\varphi=\varphi_1$.
- החזר את α.

סיבוכיות: בכל שלב האלגוריתם נותן השמה למשתנה אחד לכן סיבוכיות היא $O(n)$.

נכונות: $\varphi\in SAT\Rightarrow$ בכל שלב משתנה אחד יקבל השמה כך שעדיין ניתן לספק את $\varphi\Rightarrow$ בסוף α יספק את כל φ.

$\varphi\notin SAT\Rightarrow$ כבר בשלב הראשון A יחזיר $0\Rightarrow$ יחזיר אין השמה מספקת.

* דוגמאות נוספות עבור clique ומסלול המילטון במצגת 11 שקופיות 11-14.*

ין דומה גם
ב' בקורס
לגוריתמים 2 בפרק
על NPC.

יז) המחלקה co-NP

הגדרת המחלקה היא $coNP=\{L\mid \bar{L}\in NP\}$, כל השפות שהמשלימים שלהם ב-NP. המאפיין של שפות-מהמחלקה זו הוא שאין עד פולינומי אחד באמצעותו ניתן לוודא שייכות לשפה, לדוגמה: SAT, $\overline{clique}$. כדי להוכיח ש- $L\in coNP$ צ"ל שק"ימת מ"ט דטרמיניסטית ופולינומית M ופולינום $p(\cdot)$, כך שלכל $x\in\{0,1\}^*$ מתקיים: $x\in L\Leftrightarrow\forall y\in\{0,1\}^{p(|x|)}: M(x,y)=1$

לדוגמה: M מוודאת לכל עד y שקלט $<G,K>$ שאין קליקה בגודל שווה ל-K. __דרך אחרת__ להוכיח ש- $\bar{L}\in NP$.

__תכונות:__
- $coNP\subseteq R$: שכן לכל שפה $L\in coNP$ ניתן לבנות מ"ט מכריעה הפוכה למכונה שמכריעה את $\bar{L}$.
- $P\subseteq NP\cap coNP$: ידוע ש- $P\subseteq NP$, נוכיח $P\subseteq coNP$ - $L\in P\Rightarrow\bar{L}\in P\Rightarrow\bar{L}\in NP\Rightarrow\bar{\bar{L}}\in coNP\Rightarrow L\in coNP$.
- האם $P=coNP$? או האם $NP=coNP$? הם גם כן שאלות פתוחות בנוסף ל- $P=NP$?

דוגמה: הוכח כי $coNP \in \{\varphi$ נוסחת CNF שכל השמה z מספקת את $\varphi \mid \varphi\} = $ Tautology

הוכחה: נוכיח כי $NP \in \{\varphi \mid$ קיימת השמה z עבורה $\varphi(z)=0\} = \overline{\text{Tautology}}$. נגדיר מ"ט מוודאת המקבלת (φ, z) ובודקת האם $\varphi(z)=0$. אם כן מקבלת, אחרת דוחה.

יח) המחלקה NEXP

המחלקה NEXP מוגדרת כך: $NEXP=\bigcup_{i\in\mathbb{N}} NTIME\left(2^{(n^i)}\right)$. כלומר NEXP היא קבוצת כל השפות שקיימת עבורן מ"ט א-דטרמיניסטית המכריעה אותן בזמן ריצה של $O\left(2^{(n^i)}\right)$ לכל $i\in\mathbb{N}$, זמן ריצה אקספוננציאלי.

כל המחלקות ה-A
P | NP
EXP | NEX
PTIME | NTI

משפט: אם $EXP \neq NEXP$ אזי $P \neq NP$.

הוכחה: נניח $P=NP$ ונוכיח $EXP = NEXP$. כיוון שידוע ש-$EXP\subseteq NEXP$, שהרי מ"ט דטרמיניסטית היא מקרה פרטי של מ"ט א-דטרמיניסטית, נותר להוכיח את הכיוון השני ש-$NEXP\subseteq EXP$. תהי $L\in NEXP$. אזי קיימת לה מ"ט א-דטרמיניסטית M_L המכריעה אותה בזמן של $O\left(2^{(|x|^c)}\right)$ לכל $x\in L$ ועבור c קבוע. נגדיר שפה $L_{PAD}=\{x\overset{\text{שרשור}}{\cdot}1^{\left(2^{(|x|^c)}\right)} : x\in L\}$. נגדיר מכונת טיורינג א-דטרמיניסטית M_{PAD} שתכריע את השפה L_{PAD}: M_{PAD} תחשב ב-$O(1)$ מהו החלק של x בקלט שקיבלה ואז תריץ על חלק זה את M_L ותענה כמוה. נמצא שסיבוכיות הזמן של M_{PAD} היא $O\left(2^{|x|^c}\right)$. אמנם עבור M_{PAD} סיבוכיות זו היא פולינומיאלית לגודל הקלט לכן $L_{PAD}\in NP$. מכך ש-$P=NP$ אזי גם $L_{PAD}\in P$. מכאן שיש מ"ט דטרמיניסטית $\hat{M}_{PAD}$ המכריעה את L_{PAD} בזמן פולינומי. כעת ניתן להגדיר מ"ט דטרמיניסטית ל-L. נגדיר מ"ט דטרמיניסטית $\hat{M}_L$ שעבור כל קלט x תשרשר לו $1^{\left(2^{(|x|^c)}\right)}$ ותפעיל על המחרוזת שיצאה את $\hat{M}_{PAD}$ ותענה כמוה. סיבוכיות $\hat{M}_L$ היא $O\left(2^{(|x|^c)}\right)$ ולכן $L\in EXP$. מ.ש.ל.

וגם $P\subseteq NP\subseteq EXP\subseteq NE$
$\subseteq DTIME = NTIME =$
{ מצה לא יודעים.

יט) משפטי ההירכ"ה לסיבוכיות זמן

__פונקציות Time-Constructible__: פונקציה $T:\mathbb{N}\to\mathbb{N}$ תקרא Time-Constructible אם קיימת מ"ט M_T אשר בהינתן קלט 1^n עבור $n\in\mathbb{N}^{*}$, רצה $O(T(n))$ צעדים ומחזירה את הערך $T(n)$. נדרוש גם $T(n)\geq n$. למעשה, אף שזו נראית הגדרה לא טריוויאלית כמעט כל הפונקציות הם Time-Constructible.

על כל קלט אחר
עבור n

__משפט ההיררכיה לסיבוכיות זמן דטרמיניסטית__ (Hartmanis-Streants): תהיינה f,g שתי פונקציות Time-Constructible המקיימות: $f(n)\cdot\log(f(n)) = O(g(n))$ אזי גם מתקיים: $DTIME(f(n))\subsetneq DTIME(g(n))$. כלומר, קבוצת השפות שקיימת עבורן מ"ט דטרמיניסטית המכריעה אותן בזמן $O(f(n))$ מוכלת ממש בקבוצת השפות שקיימת עבורן מ"ט דטרמיניסטית המכריעה אותן בזמן $O(g(n))$!

באמצעות משפט זה
נוכיח $EXP\nsubseteq P$
· הפונקציות
$f(n)=n^{\log(n)},\ g(n)=2^n$.
$n^{\log(n)}\cdot\log\left(n^{\log(n)}\right)=$
$n^{\log(n)}\cdot\log^2(n)=O(2^n$

__משפט ההיררכיה לסיבוכיות זמן א-דטרמיניסטית__: תהיינה f ו-g שתי פונקציות Time-Constructible המקיימות $f(n+1)=o(g(n))$ אזי גם מתקיים: $NTIME(f(n))\subsetneq NTIME(g(n))$

בס"ד

(11)

כ) מכונת טיורינג עם גישת אורקל

מ"ט עם גישת אורקל לשפה L כוהי מ"ט רגילה שיש לה סרט נוסף שבו ניתן לשאול שאלות כמו האם $x\in L$ עבור $x\in\{0,1\}^*$. כדי לעשות זאת המכונה תעבור למצב מיוחד q_{query} ותכתוב את השאילתא על הסרט הנוסף. אם $x\in L$ בתוך צעד אחד המכונה תעבור למצב q_{yes}, ואם $x\notin L$ תעבור למצב q_{no}.
באופן כללי, נסמן $M^L(x)$ להיות חישוב של מכונת טיורינג M על קלט x כאשר נתונה גישת אורקל לשפה L.
בנוסף, לכל שפה L נגדיר P^L להיות מחלקת כל השפות הניתנות להכרעה ע"י מכונת טיורינג דטרמיניסטית עם גישת אורקל לשפה L בזמן פולינומי. כמו כן NP^L עבור מכונת טיורינג א'-דטרמיניסטית.

משפטים:

(1) מתקיים: $NP, coNP\subseteq P^{SAT}$.

הוכחה: עבור כל שפה $L\in NP$ קיימת רדוקציה פולינומית $L\le SAT$ עם פונקציית רדוקציה f. לכן עבור כל $x\in L$ נוכל לחשב את $f(x)$ בזמן פולינומי ואז לשאול את האורקל, על $f(x)$ ולענות כמוה.
עבור $L\in coNP$ נענה הפוך מהאורקל.

(2) תהא $L\in P$. מתקיים $P^L=P$.

להשלים /

הוכחה:

(3) נגדיר את השפה $EXPCom=\{\langle M,x\rangle\cdot 1^n :$ M מקבלת את x תוך 2^n צעדי חישוב $\}$.

משפט: מתקיים $P^{EXPcom}\subseteq NP^{EXPcom}\subseteq EXP$

הוכחה: הכלה $P^{EXPcom}\subseteq NP^{EXPcom}$ נובעת טריוויאלית שכן כל מכונה דטרמיניסטית היא מקרה פרטי של מכונה א'-דטרמיניסטית. נוכיח $NP^{EXPcom}\subseteq EXP$. תהי $L\in NP^{EXPcom}$ אזי קיימת מכונת טיורינג א'-דטרמיניסטית N עם גישת אורקל ל-EXPcom הרצה בזמן $k\cdot n^c$ עבור k,c קבועים. נגדיר מכונת טיורינג דטרמיניסטית M הרצה בזמן אקספוננציאלי ומכריעה את L. M על קלט $x\in\{0,1\}^*$:

- מסמלצת את N על x לכל סדרת חישוב אפשרית ב-N. יש $2^{k\cdot n^c}$ מסלולי חישוב כאלו.
- לכל שאילתא של N לאורקל מהצורה $\langle\hat{M},y\rangle 1^m$ M תריץ את U (מ"ט אוניברסלית) על $\langle\hat{M},y\rangle$ למשך 2^m צעדים. התשובה שתחזיר ל-N (במקום האורקל) תהיה 0 אם $\hat{M}$ לא עצרה בתוך 2^n צעדים ואחרת, שכן עצרה בתוך 2^n צעדים תענה כמוה.

סיבוכיות של M היא $O\left(2^{k\cdot n^c}\right)$ וכיוון שמכריעה את L נובע כי $L\in EXP$. מ.ש.ל.

(4) מתקיים $EXP\subseteq P^{EXPcom}$. תהא $L\in EXP$ אזי קיימת מ"ט דטרמיניסטית M_L המכריעה את L בזמן $c_1\cdot 2^{n^{c_2}}$. נגדיר מ"ט דטרמיניסטית פולינומית M עם גישת אורקל ל-EXPcom שתכריע את L. M על קלט x תשלח $\langle M_L,x\rangle 1^{c_1|x|^{c_2}}$ ל-EXPcom ותענה כמוה.

טענה קודמת נובע
כי כל שלושת הקבוצות
ש"ת

Scanned with CamScanner

כא) קושי הוכחת $P=NP$ או $P\neq NP$

משפט Solovay, Gil, Baker: קיימים אורקלים A,B כך שמתקיים $P^A = NP^A$ וגם $P^B \neq NP^B$.

מסקנה: לא ניתן להכריע את השאלה $P\overset{?}{=}NP$ בעזרת הוכחות רלטיביות. שהן הוכחות המשתמשות בתכונות הבאות:

(1) כל מחרוזת מ-$\{0,1\}^*$ מייצגת מ"ט, ולכל מ"ט יש ∞ ייצוגים.

(2) ניתן לסמלץ ריצה של מ"ט על ידי מ"ט אוניברסלית.

הגיון למסקנה כי הוא שאם אכן הייתה הוכחה ש- $P=NP$ אז היא הייתה נכונה גם עבור כל אורקל C כך שמתקיים $P^C = NP^C$. אבל לפי המשפט לעיל תמיד יהיה אורקל B עבורו $P^B \neq NP^B$ בסתירה להוכחה.

וכן ההיפך, אם הייתה הוכחה ש- $P\neq NP$ אזי לכל אורקל C צריך להתקיים $P^C \neq NP^C$. אך שוב לפי המשפט יש אורקל A עבורו $P^A = NP^A$ (דוגמה ל-A כזה הוא EXPcom).