הרצאה 9 – יום רביעי 20.8

סיבוכיות מקום

יהי M, DTM שעוצר על כל קלט, ותהי פונקציה $f:\mathbb{N}\to\mathbb{N}$. נאמר שיש ל-$M$ סיבוכיות מקום $f(n)$, או ש-M רצה במקום $f(n)$, אם מספר התאים ש-M קורא במהלך ריצה על כל קלט באורך n הוא לכל היותר $f(n)$.

עבור NTM, נגדיר את סיבוכיות המקום להיות המספר המקסימלי של תאים שנסרקים בכל ענף של עץ החישוב.

לגבי הקריאה של הקלט עצמו, לפעמים זה זניח (אם המקום הנדרש גדול מאורך הקלט) ולפעמים לא (אם המקום הנדרש קטן מאורך הקלט).

מחלקות סיבוכיות מקום, באופן כללי:

$$SPACE\big(f(n)\big) := \{L \mid L \text{ is a language decided by an } O\big(f(n)\big)\text{-space } DTM\}$$

$$NSPACE\big(f(n)\big) := \{L \mid L \text{ is a language decided by an } O\big(f(n)\big)\text{-space } NTM\}$$

דוגמה: $SAT \in SPACE(n)$

בעיית SAT דורשת זמן מעריכי, אבל מקום לינארי. כמו בסיבוכיות זמן, נראה את זה ע"י תיאור אלגוריתם שעובד במקום n:

נעבור על כל ההשמות וכל פעם נבדוק אם הנוסחה מסופקת. זה סיבוכיות זמן $EXPTIME$ אבל $O(n)$ בסיבוכיות מקום. בכל פעם צריך לשמור רק את ההשמה הנוכחית (אפשר בתור מחרוזת בוליאנית באורך מספר המשתנים) ולעדכן אותה (אפשר ע"י פעולת 1+, לא דורשת עוד מקום, ולינארית בזמן).

משפט סביץ' – *Savitch's theorem*

לכל פונקציה $f:\mathbb{N}\to\mathbb{R}^+$ כך ש- $f(n)\geq n$, מתקיים $NSPACE\big(f(n)\big)\subseteq SPACE\big(f^2(n)\big)$. כלומר, המעבר מ-$NTM$ ל-DTM הוא פולינומי (ריבועי).

<u>הוכחה</u>: יהי NTM N שרץ במקום $f(n)$ ותהי שפה $L := L(N)$.

נעדכן את N כך שבקבלה, הוא מוחק את הסרט ומחזיר את הראש להתחלה. זה נותן לנו שיש בפועל קונפיגורציה מקבלת יחידה($c_{accept}=q_Y$). זה לא משפיע על סיבוכיות המקום, כי זה רק צריך לעבור על המקסימום מבין גודל הקלט ומספר התאים שנכתבו.

נבנה DTM שמגדיר האם N יכול להגיע מ- c_1 ל-c_2 תוך t צעדים:

M_{yield} על קלט (c_1,c_2,t):

1. אם $t=1$ אז נבדוק ישירות האם $c_1\to c_2$. אם כן, נקבל. אחרת, נדחה.
2. אם $t>1$, אז לכל קונפיגורציה c_i:
   a. נריץ את $M_{yield}(c_1,c_i,t/2)$ ואת $M_{yield}(c_i,c_2,t/2)$. — כל פעם שמבצעים את שלב b זה דורש מקום $f(n)$.
   b. אם שניהם מקבלים, נקבל.
3. אחרת, נדחה.

עכשיו, כדי לסמלץ את $N(x)$, נריץ את $M_{yield}\big(c_{start},c_{accept},2^{df(n)}\big)$, כאשר $c_{start}=(q_{start},x)$ ו-d הוא קבוע כך ש- $2^{df(n)}$ הוא חסם עליון לזמן הריצה של N על כל ענף.

מספר הקריאות הרקורסיביות הוא $\log t$, אז עבור ההרצה שלנו זה $\log O\big(2^{df(n)}\big)=O\big(df(n)\big)=O\big(f(n)\big)$ קריאות, שכל אחת משתמשת ב- $f(n)$ מקום. בסה"כ סיבוכיות מקום $O\big(f^2(n)\big)$. כלומר, המעבר מ-NTM ל-DTM דורש במקרה הגרוע הגדלה ריבועית של מקום.

המחלקות *PSPACE, NPSPACE*

$$PSPACE := \bigcup_k SPACE(n^k),\qquad NPSPACE := \bigcup_k NSPACE(n^k)$$

$PSPACE$ זו מחלקת השפות שמוכרעות ע"י DTM במקום פולינומי. $NPSPACE$ זו מחלקת השפות שמוכרעות ע"י NTM במקום פולינומי. משפט סביץ' אומר לנו שהם שווים: $PSPACE=NPSPACE$, כי המעבר מ-NTM ל-DTM דורש רק שינוי ריבועי במקום (ולא מעריכי כמו שהיה בסיבוכיות זמן).

בגלל שכל מעבר על תא נחשב צעד, מתקיים $P\subseteq PSPACE$. אם מכונה פועלת ב-k צעדים, היא לא יכולה לבקר ביותר מ-k תאים.

באופן דומה, המקום של NTM חסום באורך הענף הארוך, אז $NP\subseteq NPSPACE$.

ואם TM משתמש במקום $f(n)$, אז מספר הקונפיגורציות הוא לכל היותר $f(n)2^{O(f(n))}$. בסה"כ:

$$P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME$$

שלמות *PSPACE – PSPACE-completeness*

שפה L תיקרא *PSPACE-hard* אם לכל $L' \in PSPACE$ מתקיים $L' \leq_p L$. (רדוקציה פולינומית בזמן! כי פולינומיות בזמן גוררת פולינומיות מקום).
והיא תיקרא *PSPACE-complete* אם בנוסף היא שייכת ל-*PSPACE*.

בעיית *TQBF – True Quantified Boolean Formula*

ניזכר **בכמתים** – *quantifiers*. $\forall$ – לכל, $\exists$ – קיים. המשתנה שמופיע יחד עם הכמת נקרא "קשור" (*bound*) לכמת, והנוסחה שיש בסוגריים אחרי הכמת נקראים התחום (*scope*) של הכמת. לדוגמה:

$$\varphi = \forall x\, \exists y\, \underbrace{[(x \vee y) \wedge (\neg x \vee \neg y)]}_{scope\ of\ \forall x, \exists y}$$

המשמעות של הנוסחה היא: **לכל** ערך שנבחר ל- x (T או F, 0 או 1), **קיים** ערך שאפשר לבחור ל- y, כך שהנוסחה φ תקבל ערך אמת.

אם כל משתנה בנוסחה מופיע בתחום של כמת כלשהו, נאמר שהנוסחה היא *fully quantified* (מכומתת).

$$TQBF \coloneqq \{\langle\varphi\rangle \mid \varphi\ is\ a\ true\ fully\ quantified\ boolean\ formula\}$$

לדוגמה: $\varphi_1 = \forall x\, \exists y\, [(x \vee y) \wedge (\neg x \vee \neg y)] \in TQBF$.

אם $x = 0$ אז נבחר $y = 1$ ואז נקבל $1 = (0 \vee 1) \wedge (1 \vee 0)$. אם $x = 1$ אז נבחר $y = 0$ ואז $1 = (1 \vee 0) \wedge (0 \vee 1)$.

מנגד, $\varphi_2 = \exists y\, \forall x\, [(x \vee y) \wedge (\neg x \vee \neg y)] \notin TQBF$.

כי אם $y = 0$ ונבחר $x = 0$, אז נקבל $0 = (0 \vee 0) \wedge (1 \vee 1)$. ואם $y = 1$ ונבחר $x = 1$ אז $0 = (1 \vee 1) \wedge (0 \vee 0)$.

הסדר של הכמתים משנה! (וואו, זוכרים לוגיקה? איכס).

TQBF היא *PSPACE-complete*

נוכיח בשני חלקים. ראשית, $TQBF \in PSPACE$. נוכיח ע"י תיאור *DTM* שמכריע את השפה בסיבוכיות מקום פולינומית:

M על קלט φ:

1. אם אין ל-φ כמתים (כלומר כל משתנה כבר קבוע), אז נחשב את הנוסחה ונחזיר את מה שיצא.
2. אם $\varphi = \exists x\, \varphi'$ (כלומר, הכמת הראשון הוא $\exists$. φ' זה המשך הנוסחה) אז נקרא ל- $M(\varphi')$ פעמיים: עם $x = 0$ ועם $x = 1$. אם אחד מהם מקבל אז נקבל. אחרת, נדחה.
3. אם $\varphi = \forall x\, \varphi'$ אז נקרא ל- $M(\varphi')$ פעמיים: עם $x = 0$ ועם $x = 1$. אם שניהם מקבלים אז נקבל. אחרת, נדחה.

עומק הרקורסיה הוא לכל היותר מספר המשתנים, ובכל פעם צריך לבצע את החישוב ($O(n)$) ולשמור רק את הערך שיצא – 0 או 1. אז:

$$TQBF \in SPACE(n) \subseteq PSPACE$$

שנית, נוכיח ש-*TQBF* היא *PSPACE-hard*. תהי שפה $L \in SPACE(n^k)$ שמוכרעת ע"י *DTM*, M. נתאר רדוקציה $L \leq_p TQBF$.

בהינתן x, נייצר φ_x: ראשית, נשים לב שמספר הקונפיגורציות וזמן הריצה חסומים ב- $2^{O(n^k)}$.

רעיון הרדוקציה: נייצר משתנה לכל קונפיגורציה, כך ש- $\varphi = T$ אמ"מ אפשר להגיע מהקונפיגורציה ההתחלתית לקונפיגורציה מקבלת ב-t צעדים. כלומר:

$\varphi_{c_1,c_2,1} = 1$ אם $c_1 \rightarrow c_2$, ואחרת 0. באופן כללי:

$$\varphi_{c_{start},c_{accept},t} = \exists c_i \left[\varphi_{c_{start},c_i,t/2} \wedge \varphi_{c_i,c_{accept},t/2}\right] = \exists c_1 \ldots \exists c_t \left[\varphi_{c_{start},c_1,1} \wedge \varphi_{c_1,c_2,1} \wedge \ldots \wedge \varphi_{c_t,c_{accept},1}\right]$$

הבעיה היא, שגודל הנוסחה הוא $2^{O(n^k)}$. ואנחנו צריכים רדוקציה פולינומית, ושהמקום יהיה פולינומי. אז נשנה קצת את הנוסחה שאנחנו בונים:

$$\varphi_{c_{start},c_{accept},t} = \exists c_i \left[\varphi_{c_{start},c_i,t/2} \land \varphi_{c_i,c_{accept},t/2}\right] = \exists c_i \; \forall (c_1,c_2) \in \{(c_{start},c_i),(c_i,c_{accept})\} \left[\varphi_{c_1,c_2,t/2}\right]$$

עומק הרקורסיה: $\log 2^{O(n^k)} = O(n^k)$. מקום בכל קריאה: 3 קונפיגורציות, $O(n^k)$. גודל הנוסחה: $O(n^k)O(n^k) = O(n^{2k})$, פולינומי.

שיטות משחק

אפשר לחשוב על הבעיה בתור משחק: שחקן A בוחר את ההשמות למשתנים עם כמת $\forall$, שחקן E בוחר את ההשמות למשתנים עם כמת $\exists$. סדר הבחירות הוא לפי סדר הופעת הכמתים בנוסחה. A רוצה שהנוסחה תצא 0, E רוצה שהיא תצא 1. בדוגמאות שראינו למעלה, E ניצח ב- φ_1, ו-A ניצח ב- φ_2. נאמר שיש לשחקן אסטרטגיה מנצחת (*winning strategy*) אם השחקן הזה מנצח בהנחה ששני השחקנים משחקים בצורה אופטימלית. נפרמל את זה ע"י השפה:

$$FORMULA\text{-}GAME \coloneqq \{\varphi \mid player\ E\ has\ a\ winning\ strategy\ for\ \varphi\}$$

<u>טענה</u>: $FORMULA\text{-}GAME \in PSPACE\text{-}complete$.

<u>הוכחה</u>: אם יש ל-E אסטרטגיה מנצחת, זה אומר שלא משנה מה A קובע למשתנים שלו, הנוסחה יוצאת אמת. כלומר:

$$E\ has\ a\ winning\ strategy\ in\ \varphi \Leftrightarrow \varphi \in TQBF$$

זו בעצם רדוקציה $FG \leq_p TQBF$, ומכיוון ש-$TQBF$ היא *PSPACE-complete*, גם FG היא *PSPACE-complete*.