שיעור 13
סיבוכיות מקום ושלמות ב PSPACE

13.1 הגדרה של סיבוכיות מקום

הגדרה 13.1 סיבוכיות מקום של מכונת טיורינג

הסיבוכיות מקום של מ"ט M על קלט w היא פונקציה $f(|w|)$ השווה למספר התאי סרט לכל היותר של המכונה M שבהם נעשה שימוש בחישוב של M על w.

הגדרה 13.2 המחלקה $SPACE(f(n))$

מחלקת $SPACE(f(n))$ היא אוסף כל השפות L עבורן קיימת מכונת טיורינג דטרמיניסטית M שמכריעה אותה כך ש:
על כל קלט w באורך $n=|w|$, המכונה M משתמשת לכל היותר $O(f(n))$ תאי סרט.

$$SPACE(f(n)) = \{L \mid \text{ מ"ט } M \text{ שמכריעה } L \text{ ומשתמשת לכל היותר ב- } O(f(n)) \text{ תאי סרט} \exists \} \ .$$

דוגמה 13.1

נראה כי ניתן לפתור את הבעיה SAT ע"י אלגוריתם שהוא רץ במקום ליניארי. כלומר:

$$SAT \in SPACE(n) \ .$$

תהי ϕ נוסחה בוליאנית כלשהי. נסמן $n=|\phi|$ ונסמן ב- m את מספר המשתנים ב- ϕ. נגדיר מכונה M שפועלת כך:

$M =$ על כל קלט $\langle\phi\rangle$:

1) M רושמת את המחרוזת $\langle\phi\rangle$ על סרט הקלט.

2) לכל השמה $a_1,a_2,\ldots,a_m$ (כאשר $a_i\in\{0,1\}$ הוא הערך הנוכחי של x_i):

א) M רושמת את מחרוזת של ההשמה $a_1a_2\cdots a_m$ על סרט העבודה.

ב) M מחשבת את הערך של ϕ עבור ההשמה הנוכחית $a_1,\ldots a_m$ ע"י סריקה של הקלט $\langle\phi\rangle$ שרשום על סרט הקלט.

ג) אם מתקבל $\phi(a_1,\ldots,a_m)=1$ אז M מקבלת.

3) אם עבור כל ההשמות התקבל $\phi(a_1,\ldots,a_m)=0$ אז M דוחה.

מכאן אנחנו רואים כי המכונה M_1 רצה במקום ליניארי. בפרט:

- M שומרת על סרט העבודה את ההשמה $a_1\ldots a_m$ וזה נדרש $O(m)$ תאים.
- המספר המשתנים, m הוא n לכל היותר.

- לכן M רצה במקום $O(n)$.

לפיכך:
$$SAT \in SPACE(n) \ .$$

הגדרה 13.3 המחלקה $NSPACE(f(n))$

מחלקת $NSPACE(f(n))$ היא אוסף כל השפות L עבורן קיימת מכונת טיורינג אי-דטרמיניסטית N שמכריעה אותה כך ש:
על כל קלט w באורך $n=|w|$ המכונה N משתמשת לכל היותר $O(f(n))$ תאי סרט מתוך כל המסלולי חישוב של N.

$$NSPACE(f(n)) = \{L \mid \exists \text{ מ"ט א"ד } N \text{ שמכריעה } L \text{ ומשתמשת לכל היותר ב- } O(f(n)) \text{ תאי סרט.}\} \ .$$

דוגמה 13.2

תהי ALL_{NFA} השפה הבאה:
$$ALL_{NFA} = \{\langle A\rangle \mid A \text{ הוא } NFA \text{ עבורו } L(A)=\Sigma^*\} \ .$$

הוכיחו כי $ALL_{NFA} \in NSPACE(n)$.

פתרון:

הפתרון מתבוסס על זה שזה פשוט יותר לבנות אלגוריתם המכריע את השפה המשלימה:
$$\overline{ALL_{NFA}} = \{\langle A\rangle \mid \text{קיימת } w\in\Sigma^* \text{ עבורה } A \text{ דוחה } w\} \ .$$

נשתמש במשפט מרכזי כדי לבנות אלגוריתם שמכריע את $\overline{ALL_{NFA}}$:

משפט 13.1

אם $M=(Q,\Sigma,\delta,q_0,F)$ הוא NFA וקיים מילה w שנדחת ע"י M אז האורך המילה $|w|\leqslant 2^q$ כאשר $q=|Q|$ הוא המספר המצבים של M, וקיימים אינסוף מלים שנדחות ע"י M.

לפני שנתאר את האלגוריתם עצמו, נגדיר סימון שנשתמש בו בבניית האלגוריתם. נניח ש- $M=(Q,\Sigma,\delta,q_0,F)$ היא מכונת NFA כלשהי. תהי $P(Q)$ הקבוצת החזקה של Q. עבור כל NFA הפונקצית המעברים היא מהצורה
$$\delta : Q\times\Sigma \to P(Q) \ .$$

בהינתן מילה $w=a_1a_2\cdots a_n$ כאשר $a_i\in\Sigma$ הוא התו ה- i של המילה, $1\leqslant i\leqslant n$. נגדיר את הסדרה הבאה:
$$S_0 \ , \ S_1 \ , \ S_2 \ , \quad \ldots \quad , \ S_n$$
כאשר
$$S_0 \triangleq \{q_0\} \ , \quad S_{i+1} \triangleq \delta(S_i,a_i) \ ,$$
כאשר $S_i\in P(Q)$ לכל $0\leqslant i\leqslant n$.

<u>בניית האלגוריתם</u>

נבנה אלגוריתם לא-דטרמיניסטי, N המכריע את $\overline{ALL_{NFA}}$ באופן הבא:

$N=$ "על כל קלט x:

1) בודקת אם $x = \langle M \rangle$, כאשר $M = (Q, \Sigma, \delta, q_0, F)$ היא מכונת NFA.

- אם לא $\Leftarrow$ N תדחה.

2) יהי $q = |Q|$ מספר המצבים של M. נגדיר $S_0 = \{q_0\}$.

3) N מבצעת את הלולאה הבאה:

לכל $0 \leqslant i \leqslant 2^q - 1$:

א) בוחרת באופן אי־דטרמיניסטי תו קלט $a_i \in \Sigma$.

ב) מחשבת

$$S_{i+1} = \delta\left(S_i, a_i\right).$$

ג) אם $S_{i+1} \cap F \neq \varnothing \Leftarrow N$ תדחה.

בפועל N בודקת את התנאי הזה ע"י לסמן את כל המצבים שב- S_{i+1}. אם אחד מהמצבים המסומנים הוא מצב קבלה אז N תדחה.

4) אם בסיום הלולאה לא היה מצב-קבלה באף אחת מן הקבוצות S_i אז N תקבל. "

אם $x \in \overline{ALL_{NFA}}$:

$\Leftarrow$ $x = \langle A \rangle$, כאשר A היא מכונת NFA. וקיימת מילה $w \in \Sigma^*$ כך ש־ A תדוחה.

$\Leftarrow$ קיימת מילה w' באורך לכל היותר 2^q ש־A תדחה.

$\Leftarrow$ קיימת ריצה של N שבה N בוחרת את התווים של w' בלולאה.

$\Leftarrow$ במהלך הריצה של A על w', אין מצב קבלה באף אחת מן הקבוצות S_i.

$\Leftarrow$ N לא דחתה עד סוף הלולאה.

$\Leftarrow$ בסופה N תקבל.

אם $x \notin \overline{ALL_{NFA}}$ אז שני מקרים:

מקרה 1) $x \neq \langle A \rangle \Leftarrow N$ תדחה בשלב 1.

מקרה 2) $x = \langle A \rangle$ ו־ $L(A) = \Sigma^*$.

$\Leftarrow$ לכל מילה $w \in \Sigma^*$, קיים שלב שבו A נמצא במצב קבלה.

$\Leftarrow$ בכל ריצה של N, קיימת איטרציה i עבורה $S_i \cap F \neq \varnothing$.

$\Leftarrow$ באיטרציה זו N תדחה.

$\Leftarrow$ בכל ריצה N תדחה.

$\Leftarrow$ N דוחה את x.

סיבוכיות מקום

- נסמן ב־ $n = |\langle M \rangle|$ את אורך הקלט, וב־ $q = |Q|$ את מספר המצבים של ה־ NFA.
- כל מצב וכל מעבר של M מופיעים בקידוד, מתקיים $q = O(n)$.
- במהלך כל ריצה, N שומרת רק את המידע הבא:

* הקבוצה הנוכחית $S_i \subseteq Q$ של מצבים אפשריים.
  לפועל N שומרת S_i בוקטור ביטים באורך q לכל היותר.
* מונה של האיטרציות הלולאה עד 2^q, המאוחסן בייצוג בינארי ודורש $O(q)$ ביטים.
* תו קלט אחד הנבחר באפן אי-דטרמיניסטי בכל איטרציה, ומשתני עזר לחישוב S_{i+1}, הדורשים מקום קבוע או לינארי ב־q.

לפיכך סיבוכיות המקום הכוללת של N היא

$$O(q) = O(n) \ .$$

לפיכך האלגוריתם N פועל במקום לינארי.

שימו לב: N לינארי במקום אף על פי שזמן הריצה שלו עלול להיות אקספוננציאלי.

13.2 משפט סביץ'

הגדרה 13.4 $CANYIELD$

בהינתן מכונת טיורנג אי-דטרמיניסטית N, מספר טבעי חיובי t, ושתי קונפיגורציות c_1, c_2 של N (ראו את ההגדרה של קונפיגורציה בהגדרה 1.3). האלגוריתם $CANYIELD$ הוא אלגוריתם דטרמיניסטי הבודק אם ניתן לעבור מ- c_1 ל- c_2 על ידי לכל היותר t צעדי חישוב של N.
התאור פסאודוקוד של $CANYIELD$ הוא כדלקמן:

$CANYIELD =$ על קלט $\langle N, c_1, c_2, t \rangle$:

1) רושם את c_1, c_2 ו- t על מחסנית.

2) בודק אם N היא מכונת טיורנג, c_1, c_2 קונפיגורציות ו- t מספר טבעי חיובי.

* אם לא אז הוא דוחה.

3) אם $t = 1$:

* אם $c_1 = c_2$ אז הוא מקבל.
* אחרת אם $c_1 \vdash_N c_2$ (אם אפשר לעבור מ- c_1 ל- c_2 בצעד אחד [ראו הגדרה 1.4]) מקבל.
* אחרת הוא דוחה.

4) אם $t > 1$, לכל קונפיגורציה c_k של הרצה של N על w אשר משתמשת במקום $f(n)$ (כאשר w היא המילה הנקראת של הקונפיגורציה c_k):

5) מריץ $CANYIELD\left(N, c_1, c_k, \left\lfloor \frac{t}{2} \right\rfloor\right)$
כאשר $\left\lfloor \frac{t}{2} \right\rfloor$ הוא השלם הכי הקרוב ל- $\frac{t}{2}$ וקטן מ- או שווה ל- $\frac{t}{2}$.

6) מריץ $CANYIELD\left(N, c_k, c_2, \left\lceil \frac{t}{2} \right\rceil\right)$
כאשר $\left\lceil \frac{t}{2} \right\rceil$ הוא השלם הכי הקרוב ל- $\frac{t}{2}$ וגדול מ- או שווה ל- $\frac{t}{2}$.

7) אם שתי ההרצות בשלבי 4) ו- 5) הסתיימו בקבלה $\Leftarrow$ מקבל.

8) אחרת אם לא התקבלה תשובת קבלה $\Leftarrow$ דוחה.

משפט 13.2 משפט סביץ'

לכל פונקציה $f:\mathbb{N}\to\mathbb{R}^+$, אם $f(n)\geqslant n$ אז

$$NSPACE\left(f(n)\right)\subseteq SPACE\left(f^2(n)\right)\ .$$

הוכחה:

הריעון של ההוכחה:

תהי N מ"ט אי-דטרמיניסטית שמכריעה את השפה A במקום $O\left(f(n)\right)$, כאשר n אורך הקלט w של N.
נבנה מכונת טיורינג דטרמיניסטית, M שמכריעה את A במקום $O\left(f^2(n)\right)$.
כלומר, בהינתן $N\in NSPACE\left(f(n)\right)$ המכריעה שפה A, נבנה $M\in SPACE\left(f^2(n)\right)$ המכריעה A.
כלומר, אנחנו נראה שלכל $N\in NSPACE\left(f(n)\right)$ קיימת $M\in SPACE\left(f^2(n)\right)$.
באופן הזה אנחנו נוכיח כי

$$NSPACE\left(f(n)\right)\subseteq SPACE\left(f^2(n)\right)\ .$$

בניית המכונה:

תהי N מכונת טיורינג אי-דטרמיניסטית שמכריעה השפה A.
תהי w מחרוזת שהיא הקלט של N.
בהינתן שתי קונפיגורציות c_1,c_2 של N ומספר טבעי חיובי t.

- אם ניתן לעבור מ- c_1 ל- c_2 בלכל היותר t צעדים $\Leftarrow$ $CANYIELD\left(N,c_1,c_2,t\right)$ מקבל.
- אחרת $\Leftarrow$ $CANYIELD\left(N,c_1,c_2,t\right)$ דוחה.

נגדיר מכונת טיורינג דטרמיניסטית M שמסמלצת את המכונה האי-דטרמיניסטית N באופן הבא.

ראשית נסמן ב- n את אורך הקלט w של N.

תהי c_0 הקונפיגורציה ההתחלתית.

נתקן את N כך שלאחר כל ריצה הראש חוזר לקצה השמאל של תוכן הסרט ו- N עוברת לקונפיגורציה c_{acc}.

נגדיר d כך ש- $2^{df(n)}$ הוא חסם עליון של מספר הקונפיגורציות שקיימות בריצות של N שדורשות $O\left(f(n)\right)$ מקום.

המכונת טיורינג הדטרמיניסטית M תוגדר כך:

$M=$ "על קלט w:

1) מריצה $CANYIELD$ על הקלט $\left\langle N,c_0,c_{\text{acc}},2^{df(n)}\right\rangle$ ועונה כמוה."

הוכחת הנכונות:

נניח $w\in L(N)$ ו- $N\in NSPACE\left(f(n)\right)$.

$\Leftarrow$ לפי ההגדרה של d, ל- N יש $2^{df(n)}$ לכל היותר.

$\Leftarrow$ קיים מסלול חישוב N על w מ- c_0 ל- c_{acc}.

$\Leftarrow$ האלגוריתם $CANYIELD$ על הקלט $\left\langle N,c_0,c_{\text{acc}},2^{df(n)}\right\rangle$ יקבל.

$\Leftarrow$ M יקבל w.

נניח $w \notin L(N)$ ו- $N \in NSPACE\left(f(n)\right)$.

$\Leftarrow$ לפי ההגדרה של d, ל- N יש $2^{df(n)}$ לכל היותר.

$\Leftarrow$ לא קיים מסלול חישוב של N על w מ- c_0 ל- c_{acc}.

$\Leftarrow$ האלגוריתם $CANYIELD$ על הקלט $\langle N, c_0, c_{\text{acc}}, 2^{df(n)}\rangle$ ידחה.

$\Leftarrow$ M ידחה w.

<u>סיבוכיות מקום:</u>

- כל פעם ש- $CANYIELD$ מפעיל את עצמו באופן רקורסיבי, הוא רושם את c_1, c_2 ו- t על מחסנית, כך שניתן יהיה לשחזר אותם לאחר הקריאה הרקורסיבית הבאה.
- בגלל ש- $N \in NSPACE\left(f(n)\right)$ אזי הכתיבה של c_1, c_2 ו- t על המחסנית דורשת $O\left(f(n)\right)$ מקום.
- בכל שלב של הרקורסיה, האלגוריתם $CANYIELD$ מחלק את t ב- 2.
- הערך ההתחלתי של t הוא $2^{df(n)}$ לכן העומק של הרקורסיה הוא
$$O\left(\log_2\left(2^{df(n)}\right)\right) = O\left(f(n)\right) \ .$$
- לכן המכום הכולל ש- M דורש הוא $O\left(f^2(n)\right)$.

לפיכך
$$M \in SPACE\left(f^2(n)\right) \ .$$

לסיכום: הוכחנו שבהינתן מכונת אי-דטרמיניסטית N כלשהי שמכריעה שפה A כלשהי עבורה
$$N \in NSPACE\left(f(n)\right) \ ,$$
קיימת מכונת טיורינג דטרמיניסטית M שמכריעה A במקום $O\left(f^2(n)\right)$, כלומר:
$$M \in SPACE\left(f^2(n)\right) \ .$$

לפיכך:
$$NSPACE\left(f(n)\right) \subseteq SPACE\left(f^2(n)\right) \ .$$

■

13.3 המחלקה PSPACE

ההגדרה הבאה היא האנלוג של ההגדרה 10.2 של אלגוריתם זמן פולינומיאלי.

הגדרה 13.5 אלגוריתם מקום פולינומיאלי

אומרים כי אלגוריתם A מכריע בעייה במקום פולינומיאלי אם קיים קבוע $c > 0$ כך שהמקום הריצה של A על קלט w חסום ע"י $O\left(|w|^c\right)$.

התזה של צרץ' טיורינג אומר שאם קיים אלגוריתם המכריע בעייה במקום פולינומיאלי, אזי קיימת מכונת טיורינג דטרמיניסטית המכריעה את השפה השקולה לבעייה זו במקום פולינומיאלי.

$$\text{אלגוריתם מכריע} \equiv \text{מכונת טיורינג דטרמיניסטית} \ .$$

הגדרה 13.6 המחלקה $PSPACE$

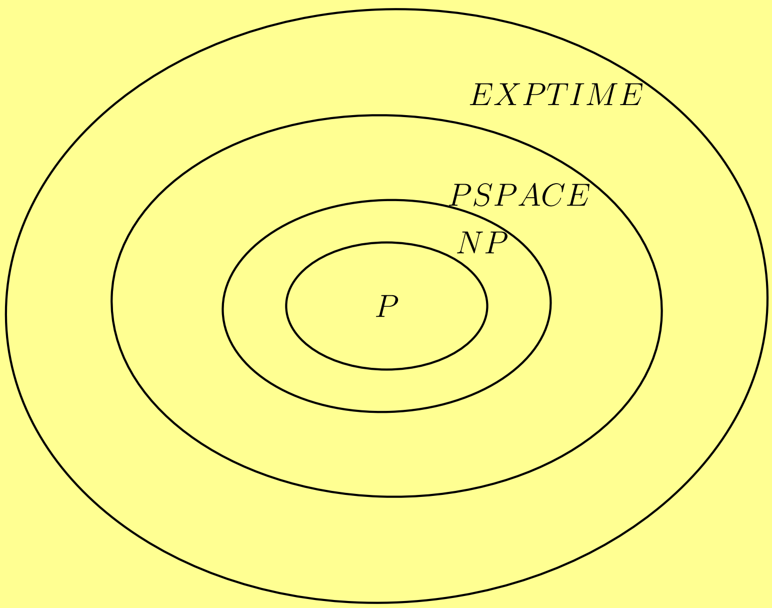
המחלקה $PSPACE$ היא אוסף כל הבעיות (השפות) שקיים עבורן אלגוריתם (מכונת טיורנג) דטרמיניסטי המכריע אותן במקום פולינומיאלי.

הגדרה 13.7 המחלקה $NPSPACE$

$NPSPACE$ היא אוסף כל הבעיות (השפות) שקיים עבורן אלגוריתם (מכונת טיורנג) אי-דטרמיניסטי המכריע אותן במקום פולינומיאלי.

משפט 13.3

$$P \subseteq NP \subseteq PSPACE \subseteq EXPTIME \ .$$



13.4 שלמות ב- $PSPACE$

הגדרה 13.8 $PSPACE$ קשה

בעייה B נקראת $PSPACE$ קשה אם לכל בעייה $A \in PSPACE$ קיימת רדוקציה זמן פולינומיאלית מ- A ל- B. כלומר: $A \leqslant_P B$.

הגדרה 13.9 $PSPACE$ שלמות

בעייה B נקראת $PSPACE$ שלמה אם השני התנאים הבאים מתקיימים:

1) $B \in PSPACE$

2) לכל בעייה $A \in PSPACE$ קיימת רדוקציה זמן פולינומיאלית מ- A ל- B. כלומר: $A \leqslant_p B$.

13.5 נוסחאות בוליאניות עם כמתים

בפרקים 11 ו- 12 הגדרנו נוסחה בוליאנית כביטוי מתמטי שבנוי מהמרכיבים הבאים:

- משתנים בוליניאיים, שמקבלים את הערכים 0 ו- 1 (לעתים מסומנים F ו- T).
- אופרטורים בוליאניים עיקריים

$\wedge$	וגם
$\vee$	או
$\neg$	לא

כעת נכליל את ההגדרה הזו לסוג היותר מורחב של נוסחה בוליאנית: נוסחה בוליאנית עם כמתים.

הגדרה 13.10 נוסחת בוליאנית עם כמתים - QBF

בנוסחת בוליאנית עם כמתים מופיעה אחת מהשני כמתים העיקריים:

$\forall$	"לכל" (נקרא גם "כמת כולל")
$\exists$	"קיים" (נקרא גם "כמת ישי")

דוגמה 13.3 דוגמאות של נוסחאותת בוליאניות עם כמתים

בדוגמאות הבאות x,y הם משתנים בוליאנים. כלומר $x,y\in\{0,1\}$.

(1

$$\phi=\forall\ x\ \exists\ y\ \ [(x\vee y)\wedge(\bar{x}\vee\bar{y})]\ .$$

בדוגמה זו $\phi=1$.

(2

$$\phi=\forall\ x\ \ [x\vee\bar{x}]\qquad\rightarrow\qquad\phi=1\ .$$

(3

$$\phi=\exists\ x\ \ (x\wedge\bar{x})\qquad\rightarrow\qquad\phi=0\ .$$

הגדרה 13.11 $TQBF$

$\langle\phi\rangle$ בשפה $TQBF$ אם ϕ נוסחת בוליאנית עם כמתים והנוסחה מעורכת לאמת.

$$TQBF=\left\{\langle\phi\rangle\ \middle|\ \phi\ \text{נוסחת בוליאנית עם כמתים ו-}\ \phi=1\ \right\}\ .$$

הערה 13.1

בניגוד ל- SAT עבורה השאלה היא האם קיימת הצבת אמת, ב- $TQBF$ לכל נוסחה יש ערך אמת או שקר יחיד, מכיוון שאין משתנים חופשיים.

משפט 13.4

$$SAT \subseteq TQBF \ .$$

הוכחה: תרגיל בית. ■

13.6 המחלקה L

13.7 המחלקה NL

13.8 שלמות ב- NL

13.9 שיוויון NL ו- coNL