

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.

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

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:

$$M = \text{על כל קלט } \langle \phi \rangle$$

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

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

(א) M רושמת את מחרוזת של ההשמה $a_1 a_2 \dots a_m$ על סרט העבודה.

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

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

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

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

• M שומרת על סרט העבודה את ההשמה $a_1 \dots a_m$ 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 .

$$NSPACE(f(n)) = \{L \mid \exists \text{ מכריעה } N \text{ ומשתמשת לכל היותר ב- } O(f(n)) \text{ תאי סרט.}\}$$

דוגמה 13.2

תהי ALL_{NFA} השפה הבאה:

$$ALL_{NFA} = \{\langle A \rangle \mid L(A) = \Sigma^* \text{ עבור } NFA \text{ } A\} .$$

הוכיחו כי $ALL_{NFA} \in NSPACE(n)$.

פתרון:

הפתרון מתבוסס על זה שזה פשוט יותר לבנות אלגוריתם המכריע את השפה המשלימה:

$$\overline{ALL_{NFA}} = \{\langle A \rangle \mid w \in \Sigma^* \text{ עבור } A \text{ דוחה } w\} .$$

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 שנדחה ע"י M אז האורך המילה $|w| \leq 2^q$ 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 הפונקציה המעברים היא מהצורה

$$\delta : Q \times \Sigma \rightarrow P(Q) .$$

בהינתן מילה $w = a_1 a_2 \dots a_n$ כאשר $a_i \in \Sigma$ הוא התו ה- i של המילה, $1 \leq i \leq n$. נגדיר את הסדרה הבאה:

$$S_0, S_1, S_2, \dots, S_n$$

כאשר

$$S_0 \triangleq \{q_0\}, \quad S_{i+1} \triangleq \delta(S_i, a_i),$$

כאשר $S_i \in P(Q)$ לכל $0 \leq i \leq n$.

בניית האלגוריתם

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 $N \Leftarrow$ תדחה.

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

(3) N מבצעת את הלולאה הבאה:

לכל $0 \leq i \leq 2^q - 1$

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

(ב) מחשבת

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

(ג) אם $N \Leftarrow S_{i+1} \cap F \neq \emptyset$ 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 $x \in \overline{ALL_{NFA}}$

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

\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) $N \Leftarrow x \neq \langle A \rangle$ 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 $S_i \cap F \neq \emptyset$.

\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 באופן 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:

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

- (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 $CANYIELD \left(N, c_k, c_2, \left\lfloor \frac{t}{2} \right\rfloor \right)$

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 $f: \mathbb{N} \rightarrow \mathbb{R}^+$, אם $f(n) \geq n$ אז

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

הוכחה:

הריעון של ההוכחה:

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

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

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 בכלל היותר t צעדים $\Leftrightarrow CANYIELD(N, c_1, c_2, t)$ מקבל.

• אחרת $\Leftrightarrow CANYIELD(N, c_1, c_2, t)$ 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 $O(f(n))$ מקום.

המכונת טיורינג הדטרמיניסטית M תוגדר כך:

$$M = \text{"על קלט } w$$

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

הוכחת הנכונות:

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

\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 $\langle N, c_0, c_{acc}, 2^{df(n)} \rangle$ יקבל.

M יקבל w . \Leftarrow

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

\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 $\langle N, c_0, c_{acc}, 2^{df(n)} \rangle$ ידחה.

$\Leftarrow M$ ידחה w .

סיבוכיות מקום:

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

לפיכך

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

לסיכום: הוכחנו שבהינתן מכונת אי-דטרמיניסטית N כלשהי שמכריעה שפה A כלשהי עבודה

$$N \in NSPACE(f(n)) ,$$

קיימת מכונת טיורינג דטרמיניסטית M שמכריעה A במקום $O(f^2(n))$, כלומר:

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

לפיכך:

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



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 $O(|w|^c)$.

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

. אלגוריתם מכריע \equiv מכונת טיורינג דטרמיניסטית

הגדרה 13.6 המחלקה $PSPACE$

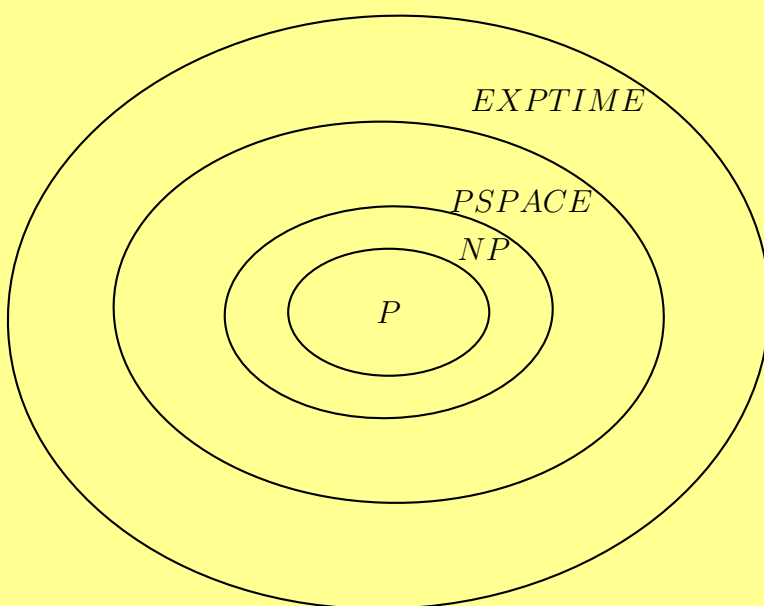
המחלקה $PSPACE$ היא אוסף כל הבעיות (השפות) שקיים עבורן אלגוריתם (מכונת טיורנג) דטרמיניסטי המכריע אותן במקום פולינומיאלי.

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: $A \leq_P B$.

הגדרה 13.9 $PSPACE$ שלמות

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

$$B \in PSPACE \quad (1)$$

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

13.5 נוסחאות בוליאניות עם כמתים

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

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

| | |
|-----|----------|
| וגם | \wedge |
| או | \vee |
| לא | \neg |

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 נוסחאות 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)

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

(3)

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

הגדרה 13.11 $TQBF$

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

$$TQBF = \{ \langle \phi \rangle \mid \phi \text{ נוסחת בוליאנית עם כמתים ו-} \phi = 1 \} .$$

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 שיויון NL ו- coNL