שיעור 13
סיבוכיות מקום ושלמות ב PSPACE

13.1 הגדרה של סיבוכיות מקום

המשאב המרכזי שדברנו עליו בקורס היה זמן החישוב שמדדנו באמצעות מספר הצעדים שמבצעת מכונת טיורינג. מדד חשוב נוסף הוא **כמות הזיכרון** שבה החישוב משתמש. לכאורה, המודל של מכונת טיורינג משתמש ב"אינסוף" זיכרון, מכיוון שהסרט שעליו כותבת המכונה הוא אינסופי. כזכור לנו הגדרנו מכונת טיורינג כך שבסרט יש רצף אינסופי של תווי רווח מצד שמאלי ומצד ימין של הקלט. אבל בפועל ניתן לדרוש שהמכונה לא תגיע לתאים שמימין לתא ספציפי כלשהו, שמהווה את חסם הזיכרון של המכונה.

הגדרה 13.1 סיבוכיות מקום של מכונת טיורינג

הסיבוכיות מקום של מכונת טיורינג M על קלט w היא פונקציה $f(|w|)$ שבהם M עושה שימוש בחישוב על w.

אם הסיבוכיות מקום של מכונה M היא $f(n)$ אז אומרים כי M רצה סמקום $f(n)$.

הגדרה 13.2 המחלקה $SPACE(f(n))$

אומרים כי שפה L שייכת למחלקה $SPACE(f(n))$ אם קיימת מכונת טיורינג דטרמיניסטית שמכריעה את L במקום $O(f(n))$. פורמלי:

$$SPACE(f(n)) = \{L \mid \text{מ"ט דטרמיניסטית } M \text{ שמכריעה } L \text{ במקום } O(f(n)) \ \exists\} \ .$$

דוגמה 13.1

נראה כי ניתן לפתור את הבעיה SAT ע"י אלגוריתם שרץ 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 $a_1 a_2 \cdots a_m$ 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."

M 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 $NSPACE\,(f(n))$

אומרים כי שפה L שייכת למחלקה $NSPACE\,(f(n))$ אם קיימת מכונת טיורינג אי-דטרמיניסטית שמכריעה את L במקום $O(f(n))$. פורמלי:

$$NSPACE\,(f(n)) = \left\{ L \;\middle|\; \exists \text{ מ"ט אי-דטרמיניסטית } M \text{ שמכריעה } L \text{ במקום } O\,(f(n)) \right\} \ .$$

דוגמה 13.2

תהי ALL_{NFA} השפה הבאה:

$$ALL_{NFA} = \left\{ \langle A \rangle \;\middle|\; A \text{ הוא } NFA \text{ עבורו } L(A) = \Sigma^* \right\} \ .$$

הוכיחו כי $ALL_{NFA} \in NSPACE\,(n)$.

פתרון:

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

$$\overline{ALL_{NFA}} = \left\{ \langle A \rangle \;\middle|\; \text{קיימת } w \in \Sigma^* \text{ עבורה } A \text{ דוחה } w \right\} \ .$$

נשתמש במשפט עזר הבא 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 $|w| \leqslant 2^q$, כאשר $q = |Q|$ הוא המספר המצבים של M, וקיימים אינסוף מלים שנדחות ע"י M.

כעת נגדיר סימון חדש שיעזור לנו לבנות את 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 $w = a_1 a_2 \cdots a_n$ כאשר $a_i \in \Sigma$ הוא התו ה- i של המילה, $1 \leqslant i \leqslant n$. נגדיר את הסדרה הבאה:

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

כאשר

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

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־ $n=\left|\langle M\rangle\right|$ 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 טיורינג, 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 $\langle N,c_0,c_{\text{acc}},2^{df(n)}\rangle$ 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 $\langle N, c_0, c_{\text{acc}}, 2^{df(n)}\rangle$ יקבל.

$\Leftarrow$ M יקבל w.

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

$\Leftarrow$ M ידחה w.

סיבוכיות מקום:

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

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

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.

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

הגדרה 13.6 המחלקה $PSPACE$

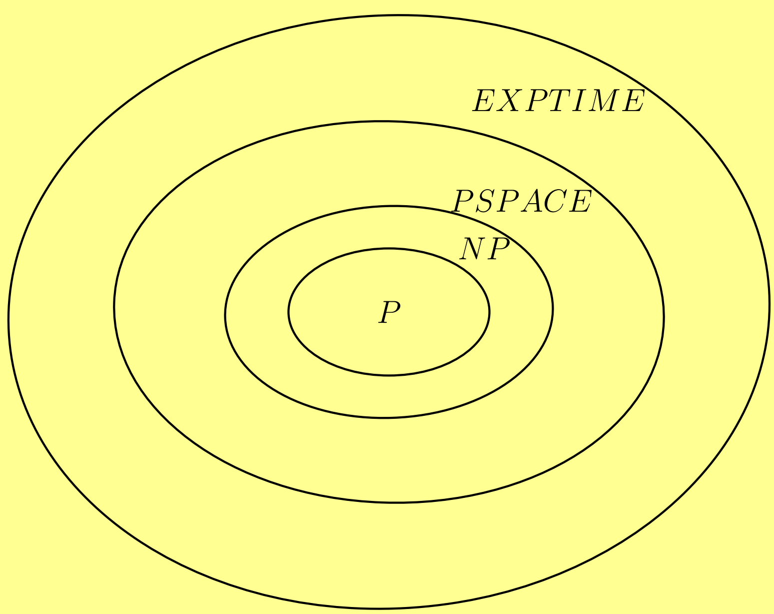
אומרים כי שפה L שייכת ל- $PSPACE$ אם קיימת מכונת טיורנג דטרמיניסטית שמכריעה אותה במקום פולינומיאלי.

הגדרה 13.7 המחלקה $NPSPACE$

אומרים כי שפה L שייכת ל- $NPSPACE$ אם קיימת מכונת טיורנג אי-דטרמיניסטית שמכריעה אותה 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

אפשר לחשוב על בעיות NP בתור סוג של "משחק" לשחקן יחיד: בהינתן שפה $L \in NP$ קיים לה יחס R_L, ואפשר לחשוב ה"המשחק" שבו בהינתן x מטרת השחקן היא למצוא y כך ש- $(x,y) \in R_L$. למשל, בהינתן לוח סודוקו x, מטרת השחקן היא למצוא מילוי y של הלוח. דוגמה נוספת, בהינתן אוסף x של צורות, מטרת השחקן היא למצוא דרך y לסדר אותן כך שירכיבו ריבוע. עוד דוגמה היא, בהינתן קוביה הונגרית במצב x, מטרת השחקן היא למצוא סדרת מהלכים y שתחזיר אותה למצב מסודר, וכן הלאה. כל המשחקים הללו מאופיינים בכך שהם משחקים לשחקן **יחיד** ועם ידע מלא (אין אלמנט של אקראיות, מרגע השחקן קיבל את הקלט).

הבעיה ה- NP- שלמה המייצגת ביותר היא SAT. אפשר לחשוב על SAT בתור בעיה של בדיקת מחרוזת בייצוג בינארי: השמה היא מילה $w \in \{0,1\}^n$ ופסוק ה- SAT ϕ בודק האם המילה w עומה על קריטריון מסוים ש- ϕ "מקודדת". כפי שראינו, לכל המשחקים שלעיל אנחנו יכולים לבנות נוסחה ϕ שתקודד את המשחק עצמו, ותבדוק האם w היא פתרון של המשחק.

אפשר להכליל את הרעיונות הללו למשחקים בשני שחקנים. המחלקה המתקבלת היא $PSPACE$ והשפה המקבילה ל- SAT במחלקה זו הוא $TQBF$.

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 בוליאניים עיקריים

$\wedge$	וגם
$\vee$	או
$\neg$	לא

אפשר להכליל את ההגדרה הזו לנוסחה בוליאנית **עם כמתים**. כמתים הם סימנים המוצמדים לפסוק ומוסיפים לו משממעות של **לכל** או של **קיים**. בהקשר שלנו "פסוק" יכלול משתנים, קשרים (אופרטורים בוליאנים עיקריים), **קבועים** (0 ו- 1) וכמתים, כאשר כמתים מתווספיםבצורה הבאה:

הגדרה 13.10 כמתים

אם ϕ הוא פסוק, אז $\forall x\phi$ (קרי "לכל x ϕ") הוא פסוק.

אם ϕ הוא פסוק, אז $\exists x\phi$ (קרי "קיים x כך ש- ϕ") הוא פסוק.

הסימנים $\exists, \forall$ נקראים **כמתים** ונשתמש בסימון Q כדי לציין כמת מבין $\forall, \exists$ באופן כללי.

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

- אמרים שבפסוק $Qx\phi$, כל מופע של x בתוך ϕ **נופל תחת הכמת** Q.
- משתנה שנופל תחת כמת כלשהו נקרא **משתנה קשור** ומשתנה שלא נופל תחת אף כמת נקרא **משתנה**

חופשי.

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 $\phi=T$.

2)

$$\phi=\forall\ x\ \ [x\vee\bar{x}]\qquad\to\qquad \phi=T\ .$$

3)

$$\phi=\exists\ x\ \ (x\wedge\bar{x})\qquad\to\qquad \phi=F\ .$$

אם בפסוק אין משתנים חופשיים, ערך האמת שלו נקבע באופן חד משמעי, בצורה הבאה:

הגדרה 13.11 ערך אמת של פסוק שלא כולל משתנים חופשיים

ערך האמת של פסוק שכולל רק קבועים וקשרים נקבע בהתאם לטבלאות האמת של הקשרים.
נסמן ב- $\phi(x)$ פסוק עם המשתנה חופשי x ונסמן ב- $\phi(a)$ את מה שמתקבל מהפסוק כשמחליפים כל מופע של x בקבוע a ($a\in\{0,1\}$).
אז:

- ערך האמת של $\forall x\phi(x)$ הוא T אם ורק אם ערך האמת של $\phi(0)$ **וגם** של $\phi(1)$ הוא T.
- ערך האמת של $\exists x\phi(x)$ הוא T אם ורק אם ערך האמת של $\phi(0)$ **או** של $\phi(1)$ הוא T.

הגדרה 13.12 QBF

אומרים שפסוק הוא QBF (Quantified Boolean Formula) אם הוא מהצורה

$$Qx_1Qx_2\ldots Qx_n\phi\left(x_1,x_2,\ldots,x_n\right)$$

כך ש- ϕ הוא פסוק ההמשתנים היחידים שלו הם $x_1,\ldots,x_n$.
לפסוק כזה יש ערך אמת מוגדר: או T או F.

הגדרה 13.13 $TQBF$

השפה $\langle\phi\rangle$ $TQBF$ כוללת את כל פסוקי ה- QBF שערך האמת שלהם הוא T.

$$TQBF=\left\{\langle\phi\rangle\ \middle|\ \phi \text{ QBF כך ש: } \phi=T\ \right\}\ .$$

הערה 13.1

נשים לב שאפשר לישוב על SAT בתור מעין כמקרה פרטי של $TQBF$ שבו כל ה- Q- ים הם $\exists$, ואז השאלה האם קיימת לפסוק השמה מספקת היא למעשה השאלה האם הפסוק עם כמתים הוא T.

משפט 13.4

$$SAT \subseteq TQBF\ .$$

הוכחה: תרגיל בית. ■

משפט 13.5 $TQBF \in PSPACE$

השפה

$$TQBF \in PSPACE\ .$$

הוכחה:
נראה אלגוריתם רקורסיבי שבכל קריאה רקורסיבית מוריד את אחד מהכמתים של הנוסחה. תנאי העצירה הוא נוסחה ϕ ללא כמתים. בנוסחה כזו אין משתנים כלל (כי ב- QBF אין משתנים חופשיים) אלא רק קבועים, וחישוב ערך האמת של הפסוק הוא פולינומיאלי בגודל הפסוק.

<u>בניית האלגוריתם</u>

$A =$ "על קלט $\langle\psi\rangle$ כאשר ψ QBF:

$$\psi = Q_1x_1Q_2x_2\ldots Q_nx_n\phi\left(x_1,\ldots,x_n\right)$$

A מבצע כך:

1) <u>מקרה הבסיס:</u> אם $n=0$ אז ψ הוא קבוע.

- אם $\psi=1$ מקבל.
- אם $\psi=0$ דוחה.

2) <u>מקרה הרקורסיבי:</u> אם $n>0$ אז A מבצע את הרקורסיה הבאה.

- אם $Q_1=\exists$:
  - מריץ A על $\psi'=Q_2x_2\ldots Q_nx_n\phi(0,x_2,\ldots,x_n)$.
  - אם התקבל שערך האמת של ψ' הוא T אז מקבל.
  - אחרת מריץ A על $Q_2x_2\ldots Q_nx_n\phi(1,x_2,\ldots,x_n)$ ועונה כמוה.
- אם $Q_1=\forall$:
  - מריץ A על $\psi'=Q_2x_2\ldots Q_nx_n\phi(0,x_2,\ldots,x_n)$.
  - אם התקבל שערך האמת של ψ' הוא F אז דוחה.
  - אחרת מריץ A על $Q_2x_2\ldots Q_nx_n\phi(1,x_2,\ldots,x_n)$ ועונה כמוה.

<u>הוכחת הנכונות</u>

ניתן להוכיח הנכונות של A ע"י אינדוקציה.

<u>שלב הבסיס: $n=0$.</u>

אם $\psi=1 \Leftarrow$ בשלב (1) A מקבל.

אם $\psi=0 \Leftarrow$ בשלב (1) A דוחה.

<u>שלב המעבר: $n=k$.</u>

נניח ש- A מכריע QBF באורך k: $\psi=Q_1x_1Q_2x_2\ldots Q_kx_k\phi\left(x_1,\ldots,x_n\right)$. זוהי ההנחת האינדוקציה שלנו.
נוכיח כי A גם מכריע QBF באורך $k+1$ באופן הבא:

אם $T=\left(x_1,\ldots,x_k,x_{k+1}\right)\phi Q_1x_1Q_2x_2\ldots Q_kx_kQ_{k+1}x_{k+1}$ $\Leftarrow$ שני מקרים:

מקרה 1: $Q_1=\exists$

$\Leftarrow$ $Q_2x_2\ldots Q_kx_kQ_{k+1}x_{k+1}\phi\left(0,\ldots,x_k,x_{k+1}\right)=T$
או $Q_2x_2\ldots Q_kx_kQ_{k+1}x_{k+1}\phi\left(1,\ldots,x_k,x_{k+1}\right)=T$
$\Leftarrow$ מכיוון ש- A מכריע QBF באורך k אז A מקבל.

מקרה 2: $Q_1=\forall$

$\Leftarrow$ $Q_2x_2\ldots Q_kx_kQ_{k+1}x_{k+1}\phi\left(0,\ldots,x_k,x_{k+1}\right)=T$
וגם $Q_2x_2\ldots Q_kx_kQ_{k+1}x_{k+1}\phi\left(1,\ldots,x_k,x_{k+1}\right)=T$
$\Leftarrow$ מכיוון ש- A מכריע QBF באורך k אז A מקבל.

אם $F=\left(x_1,\ldots,x_k,x_{k+1}\right)\phi Q_1x_1Q_2x_2\ldots Q_kx_kQ_{k+1}x_{k+1}$ $\Leftarrow$ שני מקרים:

מקרה 1: $Q_1=\exists$

$\Leftarrow$ $Q_2x_2\ldots Q_kx_kQ_{k+1}x_{k+1}\phi\left(0,\ldots,x_k,x_{k+1}\right)=F$
או $Q_2x_2\ldots Q_kx_kQ_{k+1}x_{k+1}\phi\left(1,\ldots,x_k,x_{k+1}\right)=F$
$\Leftarrow$ מכיוון ש- A מכריע QBF באורך k אז A דוחה.

מקרה 2: $Q_1=\forall$

$\Leftarrow$ $Q_2x_2\ldots Q_kx_kQ_{k+1}x_{k+1}\phi\left(0,\ldots,x_k,x_{k+1}\right)=F$
וגם $Q_2x_2\ldots Q_kx_kQ_{k+1}x_{k+1}\phi\left(1,\ldots,x_k,x_{k+1}\right)=F$
$\Leftarrow$ מכיוון ש- A מכריע QBF באורך k אז A דוחה.

<u>סיבוכיות מקום של האלגוריתם</u>

- עומק הרקורסיה בהרצה של האלגוריתם על פסוק עם n כמתים הוא n.
- כל רמה של הרקורסיה דורשת תא בודד של מקום למעט שלב תנאי העצירה שדורש מקום פולינומיאלי.
- לכן האלגוריתם כולו דורש רק מקום פולינומיאלי.

■

משפט 13.6 $TQBF$ היא $PSPACE$ שלמה.

השפה $TQBF$ היא NP שלמה.

הוכחה: נראה כי

1) $TQBF\in PSPACE$

2) לכל $A\in PSPACE$ מקתיים $A\leqslant_P TQBF$.

הוכחנו כי $TQBF\in PSPACE$ במשפט 13.5. נשאר להוכיח כי לכל $A\in PSPACE$ מתקיים $A\leqslant_p TQBF$.

בניית הרדוקציה $L\leqslant_P TQBF$.

נראה כי לכל שפה $L\in PSPACE$ מתקיים

$$L\leqslant_P TQBF\ .$$

תהי L שפה השייכת ל- $PSPACE$ ותהי M מכונת טיורינג המכריעה את L במקום פולינומיאלי $S(n)$. נוכיח שקיימת פונקציה f כך ש:

$$x\in L \iff f(x)\in TQBF\ .$$

ז"א $f(x)=\psi$ כאשר ψ היא נוסחה בוליאנית עם כמתים כך ש:

$$\begin{aligned} x\in L &\quad\Rightarrow\quad \psi=1\ , \\ x\notin L &\quad\Rightarrow\quad \psi=0\ . \end{aligned}$$

כדי להגדיר את הפונקצית הרדוקציה, קודם נגדיר את הנוסחאה הבוליאנית הבאה. יהיו c,c' שתי קונפיגורציות של המכונה M ויהי t מספר שלם חיובי.

$$\phi\left(c,c',t\right)\triangleq\begin{cases}1 & : \text{אפשר לעבור מ- } c \text{ ל- } c' \text{ עם } t \text{ צעדים לכל היותר} \\ 0 & : \text{אחרת}\end{cases}\ .$$

הצורה המפורשת של הנוסחה $\phi\left(c,c',t\right)$ עצמה בנויה רקורסיבי באופן הבא.

<u>המקרה $t>1$</u>

לכל $t>1$ ולכל קונפיגורציות c,c':

$$\phi\left(c,c',t\right)=\exists w\ \forall x,y\quad (x=c\wedge y=w)\vee(x=w\wedge y=c')\quad\Rightarrow\quad\phi\left(x,y,\left\lfloor\frac{t}{2}\right\rfloor\right)\ .$$

כאן w מסמן קונפיגורציה ביניים בין הקונפיגורציה c לקונפיגורציה c'. היחס הרקורסיבי הזה של $\phi\left(c,c',t\right)$ אומר ש- M יכול לעבור מ- c ל- c' ב- t צעדים לכל היותר אם קיימת קונפיגורציה ביניים w כך ש- M יכול לעבור מ- c ל- w בכל היותר $\lfloor\frac{t}{2}\rfloor$ צעדים, ואחר כך M יכול לעבור מ- w ל- c' בכל היותר $\lfloor\frac{t}{2}\rfloor$ צעדים. הנוסחאות $\phi\left(c,w,\lfloor\frac{t}{2}\rfloor\right)$ ו- $\phi\left(w,c',\lfloor\frac{t}{2}\rfloor\right)$ הן נבנות בשלב הבא של הרקורסיה באותו אופן. התתליך הזה ממשיך עד שנגיע לנוסחה $\phi(x,y,t)$ שבה $t=0$ או $t=1$.

<u>המקרה $t=1$</u>

$$\phi(c,c',t=1)=\begin{cases}1 & : \text{אם } M \text{ יכול לעבור מ- } c \text{ ל- } c' \text{ בצעד חישוב בודד} \\ 0 & : \text{אחרת}\end{cases}$$

הצורה המפורשת של $\phi(c,c',t=1)$ היא כדלקמן. תהיינה c,c' שתי קונפיגורציות כלשהן. נגדיר טבלת הקונפיגורציות:

#	c_1	c_2	$\ldots$	c_{N-1}	c_N	#
#	c'_1	c'_2	$\ldots$	c'_{N-1}	c'_N	#

נניח כי $M=(Q,\Sigma,\Gamma,\delta,q_0,q_{\text{acc}},q_{\text{rej}})$ ותהי $C=Q\cup\Gamma\cup\{\#\}$. עבור כל תא ה- (i,j) של הטבלת הקונפיגורציות נגדיר משתנה בוליאני $x_{i,j,s}$ לכל $1\leqslant i\leqslant 2$ ולכל $1\leqslant j\leqslant N$ כך ש:

$$x_{ijs}=\begin{cases}1 & :ij \text{ הסימן } s\in C \text{ כתוב בתא }\\ 0 & :ij \text{ הסימן } s\in C \text{ לא כתוב בתא }\end{cases}$$

למשל אם בתא $(2,5)$ כתוה a אז $x_{2,5,\text{a}}=1$ בעוד $x_{2,5,\text{b}}=0$. הנוסחה $\phi\left(c,c',t=1\right)$ מוגדרת:

$$\phi\left(c,c',t=1\right)=\phi_{\text{cell}}\wedge\phi_c\wedge\phi_{\text{move}}\wedge\phi_{c'}\ .$$

- $\phi_{\text{cell}}=1$ אם ורק אם יש רק סימן אחד בדיוק שכתוב בכל תא ו- 0 אחרת. בפרט:

$$\phi_{\text{cell}}=\bigwedge_{\substack{1\leqslant j\leqslant 2\\ 1\leqslant j\leqslant n^k}}\left[\left(\bigvee_{s\in C}x_{i,j,s}\right)\wedge\left(\bigwedge_{\substack{s,t\in C\\ s\neq t}}\left(\overline{x_{i,j,s}}\vee\overline{x_{i,j,t}}\right)\right)\right] \tag{13.1}$$

  * $\bigvee_{s\in C}x_{i,j,s}$ מבטיח שלכל תא לפחות משתנה אחד דולק.
  * $\bigwedge_{\substack{s,t\in C\\ s\neq t}}\left(\overline{x_{i,j,s}}\vee\overline{x_{i,j,t}}\right)$ מבטיח שלכל תא משתנה אחד לכל היותר דולק.

לפיכך ϕ_{cell} מסופקת אם ורק אם יש בדיוק סימן אחד כתוב בכל תא. בניגוד זה אם שני משתנים של אותו תא דולקים בגלל שהסימן s וגם הסימן t כתובים בו זמנית בתא ה- ij אז $x_{i,j,s}=1$ וגם $x_{i,j,t}=1$ ואז ϕ_{cell} תהיה שווה ל- 0.

- הנוסחה ϕ_c ($\phi_{c'}$) קובעת כי המשתנים הספציפים של הקונפיגורציות c (c') הם דולקים:

$$\begin{aligned}\phi_c=&x_{1,0,\#}\wedge x_{1,1,c_1}\wedge x_{1,2,c_2}\wedge\ldots\wedge x_{1,N-1,c_{N-1}}\wedge x_{1,N,c_N}\wedge x_{1,N+1,\#}\ ,\\ \phi_{c'}=&x_{2,0,\#}\wedge x_{2,1,c'_1}\wedge x_{2,2,c'_2}\wedge\ldots\wedge x_{2,N-1,c'_{N-1}}\wedge x_{2,N,c'_N}\wedge x_{2,N+1,\#}\ .\end{aligned} \tag{13.2}$$

- $\phi_{\text{move}}=1$ אם אפשר להגיע מ- c ל- c' על ידי תזוזה חוקית אחת של M ו- 0 אחרת. תזוזה חוקית בין כל שתי קונפיגורציות מוגדרת על פי סהפונקצית המעברים של M. במונחי הטבלה, אפשר להגדיר תזוזה חוקית בין השתי שורות על ידי תת-טבלה מסדר 3×2 שמכילה 3 עמודות.

  נקרא תת-טבלה כזה "חלון". אלה הם דוגמאות של חלונות חוקיים:

a	q_1	b
q_2	a	c

a	q_1	b
a	a	q_2

a	a	q_1
a	a	b

#	b	a
#	b	a

a	b	a
a	b	q_2

b	b	b
c	b	b

  בעוד אלה הם דוגמאות של חלונות לא חוקיים:

a	b	a
a	a	a

a	q_1	b
q_1	a	a

b	q_1	b
q_2	b	q_2

$\phi_{\text{move}}=1$ אם כל חלון של הטבלה חוקי ו- 0 אם לא. בפרט, כל חלון מכיל 6 תאים. החלון המכיל את ה- 3 עמודות i, $i+1$ ו- $i+2$ תקרא החלון-i. לכן:

$$\begin{aligned}\phi_{\text{move}}=&\bigwedge_{1\leqslant i\leqslant N-2}\left(\text{חלון-} i \text{ חוקי}\right)\\ =&\bigwedge_{1\leqslant i\leqslant N-2}\left(x_{1,i,c_i}\wedge x_{1,i+1,c_{i+1}}\wedge x_{1,i+2,c_{i+2}}\wedge x_{2,i,c'_i}\wedge x_{2,i+1,c'_{i+1}}\wedge x_{2,i+2,c'_{i+2}}\right)\end{aligned} \tag{13.3}$$

המקרה $t=0$

אם $t=0$ אז

$$\phi\left(c,c',t=0\right)=\begin{cases}1 & : c=c' \\ 0 & : c\neq c'\end{cases} \ .$$

כעת נחזור להגדרה עצמה של הרדוקציה פולינומיאלית. תהי L שפה כריעה ע"י מכונת טיורינג M במקום $O\left(n^k\right)$, אז קיימת רדוקציה $f(x)=\psi$ כך ש:

$$\psi=\phi\left(c_{\text{start}},c_{\text{acc}},h\right)$$

כאשר $h=2^{df(n)}$ ו- $d>0$ מספר ממשי חיובי הנבחר כך של- M יש לכל היותר $2^{df(n)}$ קונפיגורציות בהינתן קלט של אורך n, ו- $f(n)=n^k$.

הוכחת הנכונות

אם $x\in L$

$\Leftarrow$ M תקבל x

$\Leftarrow$ M יכולה לעבור מ- c_{start} ל- c_{acc} במספר צעדים פחות מ- או שווה ל- h

$\Leftarrow$ $\psi=1$

$\Leftarrow$ $\psi\in TQBF$.

אם $x\notin L$

$\Leftarrow$ M תדחה x

$\Leftarrow$ לא קיימים צעדים של M מ- c_{start} ל- c_{acc}.

$\Leftarrow$ $\psi=0$

$\Leftarrow$ $\psi\notin TQBF$.

סיבוכיות זמן של הרדוקציה

בפונקצית הרדוקציה מחשבת $f(x)=\psi=\phi\left(c_{\text{start}},c_{\text{acc}},h\right)$ כאשר $h=2^{df(n)}$ ו- $f(n)=n^k$, באופן רקורסיבי. לרקורסיה יש מספר שלבים השווה ל:

$$\log_2\left(2^{df(n)}\right)=df(n)=dn^k \ .$$

- הנוסחה (13.1) של ϕ_{cell} מכילה n^{2k} נוסחאות עם 3 ליטרלים. לכן הסיבוכיות זמן של ϕ_{cell} היא:
$$O\left(n^{2k}\right) \ .$$
- הנוסחאות ϕ_c ו- $\phi_{c'}$ במשואה (13.2) מכילות בדיוק n^k ליטרלים. לכן הסיבוכיות זמן של ϕ_c ושל $\phi_{c'}$ הן:
$$O\left(n^{k}\right) \ .$$
- הנוסחה (13.3) של ϕ_{move} מכילה n^{2k} נוסחאות עם 6 ליטרלים. לכן הסיבוכיות זמן של ϕ_{move} היא
$$O\left(n^{2k}\right) \ .$$

לפיכך בסה"כ הרדוקציה ניתנת לחישוב בזמן $O\left(n^{2k}\right)$ ולכן $f\in TIME\left(n^{2k}\right)$ ולכן הרדוקציה היא רדוקציה פולינומיאלית.

■

13.6 אסטרטגיות מנצחות במשחקים

משחק בערך מוגדר תחרות בו שני שחקנים מתחרים להשיג מטרה כלשהי לפי כללים השמחק. משחקים באים במגוון צורות, כולל משחקי לוח, כגון שחמט, משחקים כלכליים, משחקי מלחמה, משחקים שמודלים תחרות עסקית ותחרות חברתית.

יש קשר בין משחקים וכמתים. לכל פסוק עם כמתים קיים משחק מתאים, ולהפך כל משחק ניתן לתאר ע"י נוסחה עם כמתים. ראשון תיאור פסוק עם כמתים כמשחק נותן הבנה אינטואטיבית של המשמעות של הפסוק. מצד שני ביטוי של משחק במונחי פסוק עם כמתים עוזר לנו להבין את האינטרקציות בין השחקנים, ההשפעה של התור של שחקן אחד על השני והתוצאה של המשחק.

למטה נגדיר משחק כזה שנרא **משחק נוסחה בוליאנית** $FORMULAGAME$

הגדרה 13.14 $FORMULAGAME$

תהי

$$\psi = \exists x_1 \forall x_2 \exists x_2 \ldots Q x_k \phi\left(x_1, \ldots, x_k\right)$$

נוסחה בוליאנית עם כמתים בצורת PNF כאשר Q מסמן $\forall$ או $\exists$. נגדיר משחק $FORMULAGAME$ באופן הבא:

- יש שני שחקנים: אליס (A) ובוב (B).
- אליס ובוב בוחרים בתורם ערך של אחד המשתנים $x_1, \ldots, x_k$.
    - שחקן A בוחר את הערכים של אילו משתנים הנופלים תחת הכמתים $\forall$.
    - שחקן B בוחר את הערכים של אילו משתנים הנופלים תחת הכמתים $\exists$.
- התורים של אליס ובוב בתורי המשחק הם לפי סדר הופעת הכמתים בנוסחה ψ.
- כיוון שלא נשארים משתנים בלי ערכים, בודקים את הערך אמת של הנוסחה ψ ללא הכמתים בתחילת הנוסחה. כלומר בודקים את הערך של $\phi(x_1, \ldots, x_k)$ עם הערכים של המשתנים שאליס ובוב בחרו.
    - אם $\phi = 1$ אז בוב (שקחן B) מנצח.
    - אם $\phi = 0$ אז אליס (שקחן A) מנצחת.

13.7 המחלקה L

13.8 המחלקה NL

13.9 שלמות ב- NL

13.10 שיוויון NL ו- coNL