חישוביות - שיעור 7

דוד קלוטה

17 במאי 2026

הקדמה

היום מתחילים ללמוד על סיבוכיות - אחד הנושאים המרגשים במדעי המחשב! בחלק השלישי של הקורס, נראה מה ניתן ולא ניתן לעשות בהינתן מגבלת משאבים. עד עכשיו דיברנו על מודלים פשוטים יותר, כמו אוטומט סופי, מכונת טיורינג, מכונת טיורינג לא דטרמיניסטית. מכאן, ראינו כי יש שפות חשיבות ושפות לא חשיבות. עכשיו, נרצה לראות מה ניתן לפתור בפועל. אם יש בעיית חישוב שכדי לפתור אותה צריך מספר זמן יותר ארוך מהצפי למות החום של היקום, מה בדיוק נעשה עם זה? נסווג את הבעיות לבעיות שאותן נוכל לפתור ריאלית ולבעיות שלא ניתן לפתור ריאלית

התחלנו!

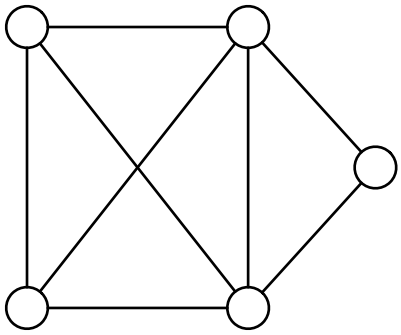
7.1 סיבוכיות

נרצה לסווג בעיות חישוב לפי מה ניתן / לא ניתן לפתור בהינתן כמות משאבים (זמן / מקום) מוגבלת.

דוגמה **7.1.1**. נתבונן בבעיות הבאות:

$$P_1 = \{G = (V, E) \mid G \text{ is an undirected graph and contains an Euler cycle}\}$$
$$P_2 = \{G = (V, E) \mid G \text{ is an undirected graph and contains an Hamilton cycle}\}$$

נזכור כי מעגל אוילר זה מעגל בו מבקרים בכל קשת בגרף בדיוק פעם אחת, ומעגל המילטון זה מעגל בו מבקרים בכל קודקוד בגרף בדיוק פעם אחת.



איור 1: דוגמה לגרף לא מכוון

קיים אלגוריתם יעיל אשר יודע לפתור את P_1:

Algorithm 1 Determine if a graph has an Euler cycle

```
if graph is not connected then
    return False
end if
for vertex v ∈ V do
    if deg v ≡ 1 mod 2 then
        return False
    end if
end for
return True
```

עם זאת, קיים אלגוריתם לא יעיל אשר יודע לפתור את P_2:

Algorithm 2 Determine if a graph has a Hamilton cycle

```
for each permutation p = (v_i)_{i∈ℕ} in G do
    if p has an edge going from each vertex to the vertex after it, and after the final vertex back to the start then
        return True
    end if
end for
return False
```

האם לבעיית מעגל המילטוני יש פתרון יעיל? כלומר פתרון בזמן ריצה שהוא פולינום כלשהו ב־n (מספר ההרצות?) זוהי בעיית P vs NP (נדבר רוב הקורס מעכשיו) - ואם תוכיחו או תשללו כי $\mathrm{P} = \mathrm{NP}$, ממתין לכם פרס של 1000000$!
להלן קישור, אם אתם אמיצים.

הגדרה 7.1.1 (זמן ריצה של מכונת החלטה). עבור מכונת החלטה M וקלט w נגדיר את זמן הריצה של $M(w)$ כמספר צעדי הריצה עד ש־M מגיעה ל־$q \in \{q_{\mathrm{acc}}, q_{\mathrm{rej}}\}$. אם $M(w)$ לא עוצרת, נגיד שזמן הריצה שלה הוא ∞. נסמן $\mathrm{time}(M, w)$.

הגדרה 7.1.2 (מקום ריצה של מכונת החלטה). עבור מכונת החלטה M וקלט w (כאשר M עם סרט חד־כיווני לימין) נגדיר את מקום הריצה של $M(w)$ להיות האינדקס הימני ביותר ש־M כותבת עליו בריצת $M(w)$. נסמן $\mathrm{space}(M, w)$.

זמן הריצה של M כפונקציה של n (אורך הקלט) יהיה

$$\max_{|w|=n} \{\mathrm{time}(M, w)\}$$

ובדומה עבור מקום

$$\max_{|w|=n} \{\mathrm{space}(M, w)\}$$

הגדרה 7.1.3 (מחלקת זמן ריצה). נגדיר $\mathrm{Time}(f(n))$ להיות מחלקת כל השפות L שיש עבורן מכונת טיורינג M שמכריעה את L בזמן $\mathcal{O}(f(n))$.

הגדרה 7.1.4 (מחלקת מקום ריצה). נגדיר $\mathrm{Space}(f(n))$ להיות מחלקת כל השפות L שיש עבורן מכונת טיורינג M שמכריעה את L במקום $\mathcal{O}(f(n))$.

דוגמה 7.1.2. נסתכל על השפה

$$L = \{0^n 1^n \mid n \in \mathbb{N}_0\}$$

ניתן להכריע את L בעזרת מכונת טיורינג M שפועלת באופן הבא:

Algorithm 3 Course of action for Machine M

```
repeat
    if there is no leading 0 or no trailing 1 then
        return False
    end if
    Remove the leading 0
    Move the head to the rightmost cell
    Remove the trailing 1
    Move to the leftmost cell
until the tape contains an empty word
return True
```

זמן הריצה על קלט באורך n יהיה $\Theta(n^2)$.
מצאנו כי

$$L \in \mathrm{Time}(n^2)$$

נשים לב שהפתרון דורש מקום n ולכן גם

$$L \in \mathrm{Space}(n)$$

הגדרה 7.1.5 (מחלקות סיבוכיות חשובות). נגדיר את המחלקות הבאות:

1. $\mathrm{P} = \mathrm{PTIME} \overset{\mathrm{def}}{=\!=} \bigcup_{k\in\mathbb{N}} \mathrm{Time}(n^k)$
2. $\mathrm{PSPACE} \overset{\mathrm{def}}{=\!=} \bigcup_{k\in\mathbb{N}} \mathrm{Space}(n^k)$

3. $\mathrm{EXP} \overset{\text{def}}{=\!=} \bigcup_{k\in\mathbb{N}} \mathrm{Time}(k^n)$

טענה 7.1.1. *מתקיים כי*

$$\mathrm{P} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXP}$$

הוכחה. נתחיל עם טענת העזר הבאה:

טענת עזר 7.1.1. *מתקיים כי* $\mathrm{Space}(f(n)) \supseteq \mathrm{Time}(f(n))$.

הוכחת טענת העזר. במספר צעדי ריצה $\mathcal{O}(f(n))$ ניתן להגיע לאינדקס מקסימלי לכל היותר ב־$\mathcal{O}(f(n))$. □

מסקנה 7.1.1. בזמן פולינומי $\mathcal{O}(n^k)$ ניתן לכתוב לכל היותר במספר פולינומי של תאים $\mathcal{O}(n^k)$.

ברור ש־$\mathrm{P} \subseteq \mathrm{EXP}$ - כי אם זמן הריצה החסום על ידי $\mathcal{O}(n^k)$ ברור שהוא חסום גם על ידי $\mathcal{O}\left(2^{(n^k)}\right)$.

טענת עזר 7.1.2. *מתקיים כי* $\mathrm{Space}(f(n)) \subseteq \mathrm{Time}\left(2^{\mathcal{O}(f(n))}\right)$

הוכחה. אם $L \in \mathrm{Space}(f(n))$, אז יש מכונת טיורינג דטרמיניסטית M שמכריעה את L תוך הסתפקות במקום $\mathcal{O}(f(n))$. המכונה עוצרת תמיד. בפרט זה אומר ש־M לא חוזרת על קונפיגורציות, אחרת הייתה בלולאה אינסופית.

מסקנה 7.1.2. מספר צעדי הריצה של M חסום מלעיל על ידי מספר הקונפיגורציות האפשריות שלה. כלומר כלל היותר

$$|Q| \times |\Gamma| \times \mathcal{O}(f(n)) = \mathcal{O}\left(2^{cf(n)}\right)$$

כלומר $\mathrm{Time}\left(2^{\mathcal{O}(f(n))}\right)$, כנדרש. □

נשתמש בטענת העזר עדי להוכיח כי $\mathrm{EXP} \supseteq \mathrm{PSPACE}$. נתון כי $L \in \mathrm{PSPACE}$, כלומר $\bigcup_{k\in\mathbb{N}} \mathrm{Space}(n^k)$. מכאן, קיים k כך ש־$L \in \mathrm{Space}(n^k)$. לפי טענת העזר, $L \in \mathrm{Time}\left(2^{\mathcal{O}(n^k)}\right)$ ולכן

$$L \in \bigcup_{k\in\mathbb{N}} \mathrm{Time}\left(2^{(n^k)}\right) = \mathrm{EXP}$$

כנדרש. □

טענה 7.1.2. *מתקיים כי*

$$\mathrm{P} \neq \mathrm{EXP}$$

הוכחה. תגיע ביום אחר. □

7.2 השלמה: מכונת טיורינג לא דטרמיניסטית

מכונה זו היא כמו מכונת טיורינג דטרמיניסטית מבחינת הכל מלבד כך שפונקציית המעברים שלה היא לא דטרמיניסטית (טיפה כמו באוטומט הסופי, אך מתקבלת קבוצה של כל השלשות האפשריות).

הגדרה 7.2.1. נגדיר מכונת טיורינג לא דטרמיניסטית להיות כמו מכונת טיורינג דטרמיניסטית, אך עם השינויים הבאים:

עבור מכונת טיורינג דטרמיניסטית, פונקציית המעברים $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$. במכונת טיורינג לא דטרמיניסטית, הפונקציה תהיה $\delta : Q \times \Gamma \to 2^{Q\times\Gamma\times\{L,R\}}$ - כלומר לכל זוג מתוך $Q \times \Gamma$ יכולה להיות יותר משלשה אחת אפשרית של $Q \times \Gamma \times \{L, R\}$.

לכן, לכל M מכונת טיורינג לא דטרמיניסטית וקלט w, תיתכן יותר מריצה אחת. נגיד ש־$w \in L(M)$ אמ"ם קיימת ריצה של $M(w)$ שמסתיימת ב־q_{acc}.

דוגמה 7.2.1. נרצה להראות מכונת טיורינג פולינומית לא דטרמיניסטית עבור שפת P_2 מדוגמה 7.1.1 - מכונה M תפעל באופן הבא:

Algorithm 4 Course of action for M

Require: Input is in the form of a neighborhood matrix of Graph G, which we will call N
Ensure: Whether G contains a Hamilton cycle
1: Let p be a nondeterministic numbering of each row of N
2: **return** whether p has an edge going from each vertex to the vertex after it, and after the final vertex back to the start

נוכיח נכונות: נניח כי $G \in P_2$. מכאן קיימת פרמוטציה שהיא מעגל המילטוני ב־G. לכן, במקרה כזה, אם בשלב הראשון של ריצת M המכונה תחשב את הפרמוטציה הנ"ל - אזי בשורה השנייה תוודה קיום קשת בין כל 2 צמתים עוקבים בפרמוטציה תעבור בשלום והאלגוריתם יחזיר True. נניח כעת כי $G \notin P_2$. לכל פרמוטציה יש לפחות זוג אחד של צמתים עוקבים בפרמוטציה שאין בינהם קשת, לכן האלגוריתם יחזיר False.

זמן הריצה של אלגוריתם זה - בשלב הראשון, רישום הפרמוטציה הוא ברור שפולינומי ב־n, והשלב השני, וידוא קיום קשת בין כל צומת לעוקבו, לינארי לכל קשת. סה"כ זמן פולינומי. מכאן, זמן הריצה של האלגוריתם הוא פולינומי תמיד - ו־$P_2 \in \mathrm{NP}$

7.3 המקרה הלא־דטרמיניסטי

נרצה להגדיר זמן ומקום עבור מכונת טיורינג לא דטרמיניסטית!

הגדרה 7.3.1. עבור מכונת החלטה לא דטרמיניסטית M וקלט w נגדיר את זמן הריצה של $M(w)$ להיות הזמן הארוך ביותר מכל ריצות $M(w)$.

באופן דומה, נגדיר גם עבור מקום.

הגדרה 7.3.2. עבור מכונה M נגדרי את זמן הריצה שלה על קלטים באורך n לפי המקסימום על כל המילים w באורך n.

באופן דומה, נגדיר גם עבור מקום.

הגדרה 7.3.3. נגדיר את $\mathrm{NTime}(f(n))$ להיות מחלקת כל השפות L כך שקיימת מכונת טיורינג לא דטרמיניסטית M שמכריעה את L בזמן $\mathcal{O}(f(n))$.

באופן דומה נגדיר את $\mathrm{NSpace}(f(n))$.

הגדרה 7.3.4. נגדיר את המחלקות הבאות:

1. $\mathrm{NP} \overset{\text{def}}{=\!=} \bigcup_{k\in\mathbb{N}} \mathrm{NTime}(n^k)$
2. $\mathrm{NPSPACE} \overset{\text{def}}{=\!=} \bigcup_{k\in\mathbb{N}} \mathrm{NSpace}(n^k)$

7.4 רדוקציות

הגדרה 7.4.1. עבור זוג שפות L_1, L_2, נגיד שיש רדוקציה פולינומית מ־L_1 ל־L_2 ונסמן $L_1 \leqslant_{\mathrm{P}} L_2$ אמ"ם יש רדוקציה $L_1 \leqslant_{\mathrm{M}} L_2$ שהיא חשיבה בזמן פולינומי. כלומר, קיימת מכונת טיורינג דטרמיניסטית R שמקבלת קלט $w \in \Sigma_1^*$ ומחזירה פלט $f(w) \in \Sigma_2^*$ במספר צעדי ריצה שהוא פולינום ב־$|w|$, מתקיים $w \in L_1 \iff f(w) \in L_2$.

משפט 7.4.1 (משפט הרדוקציה עבור P)**.** *תהיינה L_1, L_2 שפות. אם $L_1 \leqslant_{\mathrm{P}} L_2$ וגם $L_2 \in \mathrm{P}$ - אז $L_1 \in \mathrm{P}$.*

הוכחה. נניח כי $L_2 \in \mathrm{P}$. זה אומר שקיימת מכונת טיורינג M_2 שמכריעה את L_2 בזמן פולינומי. היות ש־$L_2 \leqslant_{\mathrm{P}} L_2$, קיימת מכונת טיורינג R שמחשבת פונקציית רדוקציה מ־A ל־B ומסתפקת בזמן ריצה פולינומי. נבנה בעזרתם מכונת טיורינג M_1 שמכריעה את A בזמן פולינומי.

Algorithm 5 Course of action for M_1

Require: $w \in \Sigma_1^*$
Ensure: $w \in L_1$
1: Let $y \leftarrow R(w)$
2: **return** $M_2(y)$

נוכיח זמן ריצה: נניח כי $w \in L_1$. זה קורה אמ"ם $y = f(w) \in L_2$ (מנכונות הרדוקציה R) שזה קורה אמ"ם $M_2(y) = \mathrm{True}$ (לפי נכונות M_2) שמתקיים אמ"ם $M_1(w) = \mathrm{True}$ □

זמן ריצת M_1 מורכב מ:

1. *זמן ריצת $R(w)$ - שהוא פולינומי ב־$|w|$. נאמר $y = R(w) = \mathcal{O}(|w|^c)$ בעבור $c \in [1, \infty)$ כלשהו.*
2. *זמן ריצת $M_2(y) = M_2 \circ R(w)$ - שהוא פולינומי ב־$|R(w)|$. נאמר $M_2(y) = \mathcal{O}(|y|^{c'})$ בעבור $c'' \in [1, \infty)$ כלשהו.*

הבעיה: תיאור זמן הריצה של M_1 תלוי גם באורך של y - אבל רצינו תיאור של זמן הריצה הנ"ל כפונקציה של $|w|$ בלבד. על מנת לפתור זאת, נשים לב ש־R היא מכונת טיורינג שרצה בזמן פולינומי, לכן הפלט שלה הוא באורך לכל היותר פולינומי ב־$|w|$. לכן קיים $c'' \in [1, \infty)$ כך ש־$|y| = \mathcal{O}(|w|^{c''})$. לכן, זמן ריצת $M_1(w)$ הוא

$$\mathcal{O}\left(|w|^c + |w|^{c' \cdot c''}\right) = \mathcal{O}\left(|w|^{c^{(3)}}\right)$$

בעבור $c^{(3)} \in [1, \infty)$ כלשהו.

משפט 7.4.2 (משפט הרדוקציה עבור NP)**.** *תהיינה L_1, L_2 שפות. אם $L_1 \leqslant_{\mathrm{P}} L_2$ ו־$L_2 \in \mathrm{NP}$, אז $L_1 \in \mathrm{NP}$.*

הוכחה. משום ש־$L_1 \leqslant_{\mathrm{P}} L_2$, קיימת מכונת טיורינג דטרמיניסטית המחשבת בזמן פולינומי רדוקציה R מ־L_1 ל־L_2. משום ש־$L_2 \in \mathrm{NP}$, קיימת מכונת טיורינג פולינומית לא דטרמיניסטית M_2 שמכריעה את L_2.

נראה בעזרת שתי המכונות הללו שיש מכונת טיורינג לא דטרמיניסטית שרצה בזמן פולינומי M_1 שמכריעה את L_1.

Algorithm 6 Course of action for M_1

Require: $w \in \Sigma_1^*$
Ensure: $w \in L_1$f
1: Let $y \leftarrow R(w)$
2: **return** $M_2(y)$

נראה נכונות וזמן ריצה:

נתחיל עם נכונות. אם $w \in L_1$, אז $y = R(w) \in L_2$ (מנכונות הרדוקציה) ומכאן יש ריצה של $M_2(y)$ שמחזירה True (מנכונות M_2) ולכן יש ריצה של $M_1(w)$ שמחזירה True. אם $w \notin L_1$, אז $y = R(w) \notin L_2$ (מנכונות הרדוקציה) ומכאן כל ריצה של $M_2(y)$ לא מסתיימת ב־True (מנכונות M_2) ולכן כל ריצה של $M_1(w)$ לא מסתיימת ב־True.

ונמשיך לזמן ריצה. זמן הריצה $M_1(w)$ מורכב משני חלקים:

1. ריצת $R(w)$ לפי הגדרת רדוקציה פולינומיאלית, לוקחת $\mathcal{O}(|w|^c)$ בהינתן $c \in [1, \infty)$.
2. ריצת $M_2(y)$ לוקחת, לפי הגדרת מכונת NP לוקחת $\mathcal{O}(|y|^{c'})$ בהינתן $c' \in [1, \infty)$

אבל הראינו כבר בהוכחת משפט **7.4.1** כי $|y| = \mathcal{O}(|w|^{c''})$ בהינתן $c'' \in [1, \infty)$. סה"כ זמן ריצת $M_1(w)$ חסום מלעיל על ידי פולינום ב־$|w|$. $\square$

תרגיל 7.4.1. תהיינה L_1, L_2 כך ש־$L_2 \notin \mathcal{C}$ ו־$L_1 \leqslant_{\mathrm{P}} L_2$ כאשר $\mathcal{C} \in \{\mathrm{P}, \mathrm{NP}\}$. הוכיחו כי $L_1 \in \mathcal{C}$.

תרגיל 7.4.2. הוכיחו כי $\leqslant_{\mathrm{P}}$ הוא יחס טרנזיטיבי.