חישוביות - שיעור 8

דוד קלוטה

24 במאי 2026

התחלנו!

8.1 מוודא פולינומי

הגדרה 8.1.1. מוודא פולינומי V עבור שפה L הוא מכונת טיורינג דטרמיניסטית שמקבלת קלט w ועֵד c (מחרוזות מעל Σ^*) רצה בזמן פולינומי ב־w, ומתקיים:

אם קיים $w \in L$ אז קיים c כך ש־$V(w,c) = q_{\text{acc}}$

ואם $w \notin L$ אז לכל c מתקיים $V(w,c) = q_{\text{rej}}$

מטרתנו היא בעצם לתת איפיון חלופי למחלקה NP - אחר כך נוכיח פורמלית.

דוגמה **8.1.1.** הרצאה קודמת ראינו כי השפה אשר מכריעה אם קיים מעגל המילטוני היא ב־NP.

$$\text{U} - \text{Hamilton} - \text{Cycle} = \{G \mid G \text{ contains a Hamilton cycle}\} \in \text{NP}$$

זה בעצם שקול לכך שקיים מוודא פולינומי V כך שבהינתן w המייצג גרף, קיים c המייצג מעגל המילטוני כך ש־$V(w,c) = q_{\text{acc}}$ והיא תגיע אליו בזמן פולינומי.

נראה אותו. V תפעל באופן הבא:

Algorithm 1 Course of action of V

Require: w is $\langle G \rangle$ such that $G = (V, E)$ is an undirected graph
Require: $c = (v_i)_{i \in [|V|]}$ is a certificate for the existence of a Hamilton cycle, meaning a permutation on V
Ensure: c is a Hamilton cycle of G

```
Let i ← 1
Let S ← ∅
while i ⩽ n − 1 do
    if v_i ∈ S then
        return False
    end if
    if (v_i, v_{i+1}) ∉ E then
        return False
    end if
    i ← i + 1
end while
return whether there exists (v_{|V|}, v_1) ∈ E
```

נראה נכונות וזמן ריצה:

זמן ריצה כל שלב ב־V לוקח זמן חסום בפולינום ב$|w| = |G|$

נכונות אם $w \in \text{U} - \text{Hamilton} - \text{Cycle}$ אז יש ב־G מעגל המילטוני. לכן, אם נבחר את c להיות המעגל ההמילטוני הנ"ל, אז ריצת V על (w,c) תסתיים ב־q_{acc}. אם $w \notin \text{U} - \text{Hamilton} - \text{Cycle}$ אז אין ב־G מעגל המילטוני, ולכן לכל c מתקיים $V(w,c) = q_{\text{rej}}$ כי כל אחת מהבדיקות תיכשל.

הגדרה 8.1.2 (המחלקה NP'). נגדיר את NP' להיות מחלקת כל השפות L שיש להן מוודא פולינומי. נזכיר ש־NP היא מחלקת כל השפות L שיש להן מכונת טיורינג פולינומית לא דטרמיניסטית.

טענה 8.1.1. *מתקיים כי* $\text{NP}' = \text{NP}$. *כלומר, לשפה L יש מוודא פולינומי V אמ"ם יש ל־L מכונת טיורינג M פולינומית לא דטרמיניסטית.*

הוכחה. נוכיח את שני הכיוונים:

כיוון $\Longleftarrow$ נראה שאם יש מוודא פולינומי V ל־L אז יש מכונת טיורינג פולינומית לא דטרמיניסטית M. נבנה את M באופן הבא:

Algorithm 2 Course of action for M

Require: $w \in \Sigma^*$
Ensure: $w \in L$
1: Create a certificate c nondeterministically letter by letter
2: **return** $V(w, c)$

נראה זמן ריצה ונכונות:

זמן ריצה משיקולי אורך שדיברנו עליהם הרצאה קודמת (בעיקר כי $\mathrm{P} \subseteq \mathrm{PSPACE}$), נקבל כי $|c|$ הוא פולינומי ב־$|w|$. מכאן, ייצור c לוקח כמספר האותיות ב־c, כלומר בזמן פולינומי ב־w. לאחר מכן, הרצת $V(w, c)$ היא בזמן פולינומי מההגדרה. סה"כ זמן הריצה פולינומי ב־$|w|$.

נכונות אם $w \in L$, אז קיים c כך ש־$V(w, c) = q_{\mathrm{acc}}$ מהגדרת V. לכן, בריצה שבה M תתחיל ביצירת c הנ"ל, M תקבל. אם $w \notin L$ אז לכל c נקבל כי $V(w, c) = q_{\mathrm{rej}}$ ולכן לא משנה מה M תנחש בשלב יצירת c, נקבל כי M תדחה.

שאלה. *איך מייצרים באופן לא דטרמיניסטי את c?*

תשובה. קל לבנות פונקציית δ לא דטרמיניסטית שמייצרת אות אחרי אות, כך שבריצות שונות אפשר לקבל כל מחרוזת.

שאלה. *איך עוצרים לאחר מספר פולינומי של אותיות?*

תשובה. אפשר להוסיף מונה (טיימר) שמאותחל לאורך הפולינומי ובכל צעד, מורידים אותו ב־1.

הצלחנו להוכיח את הכיוון.

כיוון $\Longrightarrow$ נניח כי יש ל־L מכונת טיורינג פולינומית לא דטרמיניסטית. נרצה להראות שקיים מוודא פולינומי V עבור L.
נזכור כי פונקציית המעברים הלא דטרמיניסטית היא

$$\delta : Q \times \Gamma \to 2^{Q \times \Gamma \times \{L,R\}}$$

דהיינו לכל זוג אנחנו מחזירים קבוצה.
הרעיון שלנו: δ' פונקציית המעברים של V תדמה את δ - הכחירות הלא דטרמיניסטיות של δ יהיו רשומות בעד c.
נשים לב שמספר השלשות האפשריות בכל צעד של δ הוא לכל היותר $|Q| \cdot |\Gamma| \cdot 2$, כלומר קבוע k. אותיות c יהיו בתחום $[k]$. במחרוזת c נוכל לקודד את כל הבחירות הלא דטרמיניסטיות של δ בריצה של M על w.
פונקציה δ' של V (דטרמיניסטית!) תאפשר בכל סימולציה צעד ריצה לקרוא אות חדשה מ־c ואז באופן דטרמיניסטי לפעול כמו הבחירה הלא דטרמיניסטית של δ.

זמן ריצה זמן הריצה של V על (w, c) יהיה כמו זמן הריצה של $M(w)$ עם הבחירות הלא דטרמיניסטיות שרשומות ב־c כפול הזמן הנדרש לסימולציית צעד אחד.

נכונות אם $w \in L$, אז יש ריצה מקבלת של $M(w)$ ולכן אם ב־c נרשום את הבחירות הלא דטרמיניסטיות בריצה הנ"ל אז $V(w, c)$ תסיים ב־q_{acc}. אם $w \notin L$, אז לכל ריצה של $M(w)$ לא נקבל q_{acc}, ולכן לכל c נקבל כי $V(w, c)$ לא נקבל q_{acc}.
הוכחנו את שני הכיוונים, ולכן הטענה נכונה. □

8.2 NP־קושי

שבוע שעבר הגדרנו רדוקציה פולינומית מ־L_1 ל־L_2 כך: $L_1 \leqslant_{\mathrm{P}} L_2$ אם קיימת רדוקציה מ־L_1 ל־L_2 שמסתפקת בזמן פולינומי.

הגדרה 8.2.1 (NP־קושי). נגיד ששפה L היא NP־קשה NP-hard אם **לכל** $A \in \mathrm{NP}$ מתקיים $A \leqslant_{\mathrm{P}} L$.

הגדרה 8.2.2 (NP־שלמות). נגיד ששפה היא NP־שלמה NP-complete אם L היא NP־קשה ו־$L \in \mathrm{NP}$

טענה 8.2.1. *תהיינה L_1, L_2 שפות. אם L_1 היא NP־קשה וגם $L_1 \leqslant_{\mathrm{P}} L_2$ אז L_2 גם היא NP־קשה.*

הוכחה. הראינו שבוע שעבר שרדוקציות פולינומיות הן טרנזיטיביות, כלומר סגורות להרכבה, כלומר אם $L_1 \leqslant_{\mathrm{P}} L_2$ וגם $L_2 \leqslant_{\mathrm{P}} L_3$, אז גם $L_1 \leqslant_{\mathrm{P}} L_3$.
נראה שהנ"ל מסייע להוכחת הטענה: נתון כי L_1 היא NP־קשה. כלומר, לכל $L \in \mathrm{NP}$ מתקיים כי $L \leqslant_{\mathrm{P}} L_1$. נתון גם $L_1 \leqslant_{\mathrm{P}} L_2$, לכן מהרכבת רדוקציות נקבל כי $L \leqslant_{\mathrm{P}} L_2$ לכל $L \in \mathrm{NP}$ - כלומר גם L_2 היא NP־קשה. □

טענה 8.2.2 (אופציה אחת). *אם L היא NP־שלמה וגם $A \in \mathrm{P}$ אז $\mathrm{P} = \mathrm{NP}$.*

הוכחה. לפי משפט הרדוקציה. תהי $L' \in \mathrm{NP}$. נתון כי L היא NP־שלמה ובפרט L היא NP־קשה. לכן $L' \leqslant_{\mathrm{P}} L$ (כי לכל שפה ב־NP יש רדוקציה פולינומית ל־L). נתון כי $L \in \mathrm{P}$. לכן לפי משפט הרדוקציה ל־P ומתקיים $L' \in \mathrm{P}$.

מסקנה 8.2.1. לכל $L' \in \mathrm{NP}$ מתקיים גם כי $L' \in \mathrm{P}$. כלומר $\mathrm{P} \supseteq \mathrm{NP}$ וידוע כי $\mathrm{P} \subseteq \mathrm{NP}$ ולכן $\mathrm{P} = \mathrm{NP}$.

□

טענה 8.2.3 (אופציה שנייה). *אם L היא NP־שלמה ומתקיים כי $A \notin \mathrm{P}$, אז* $\mathrm{P} \neq \mathrm{NP}$

הוכחה. נניח כי L היא NP־שלמה. בפרט, $L \in \mathrm{NP}$. ואם $L \notin \mathrm{P}$ אז מצאנו דוגמת $L \in \mathrm{NP}$ ו־$L \notin \mathrm{P}$. □

נמצא שפה שכזו - נקבל מיליון דולר! ראה הרצאה קודמת.
נתאר כמה שפות וננסה להוכיח שהן NP־שלמות.

דוגמה 8.2.1. נתבונן בשפה

$$\mathrm{CLIQUE} = \left\{ (G, K) \;\middle|\; \begin{array}{l} G \text{ is an undirected graph} \\ G \text{ contains a clique of cardinality } k \end{array} \right\}$$

טענה 8.2.4. *השפה* CLIQUE *היא* NP־*שלמה.*

הוכחה. נידרש לכמה טענות עזר:

טענת עזר 8.2.1. *מתקיים כי* $\mathrm{CLIQUE} \in \mathrm{NP}$

הוכחה. נבנה מוודא פולינומי בעבור CLIQUE:

Algorithm 3 Polynomial verifier for CLIQUE

Require: $G = (V, E)$ is an undirected graph
Require: $k \in \mathbb{N}$
Require: $c = (v_i)_{i \in [k]}$ is some permutation of vertices in the graph
Ensure: c is a clique in G

```
for each pair (v_i, v_j) ⊆ [k] where i ≠ j do
    if (i, j) ∉ E then
        return False
    end if
end for
return True
```

נוכיח נכונות וזמן ריצה:

נכונות אם $(G, k) \in \mathrm{CLIQUE}$ אז בחירה של c להיות קליקה בגודל k תגרום ל־V לקבל. אם יש c כך ש־$V((G, k), c)$ מקבל, אזי בהכרח c הוא קליקה בגודל k ב־G. לכן, $(G, k) \in \mathrm{CLIQUE}$.

זמן ריצה לספור את הצמתים ב־c ולהשוות ל־k אפשר בזמן פולינומיאלי. לבדוק כל קשת בין זוג צמתים ב־c לוקחת זמן פולינומיאלי ויש $n^2 \geqslant k^2$ קשתות לבדיקה, סה"כ זמן פולינומיאלי ב־$|(G, k)|$. □

טענת עזר 8.2.2. *מתקיים כי* $3 - \mathrm{SAT} \leqslant_{\mathrm{P}} \mathrm{CLIQUE}$.

הוכחה. נבנה פונקציית רדוקציה פולינומית P כך ש־$P(\phi) = (G, k)$ ומתקיים

$$\phi \in 3 - \mathrm{SAT} \iff P(\phi) = (G, k) \in \mathrm{CLIQUE}$$

לכל פסוקית ב־ϕ נגדיר 7 צמתים ב־G (שיתאימו ל־7 ההצבות המספקות את הפסוקית). בין כל 2 צמתים נמתח קשת אמ"ם אין ביניהם סתירה, כלומר אמ"ם כל משתנה משותף מסכים על ההצבה. נגדיר $k = M$ מספר הפסוקיות ב־ϕ.
נרצה להוכיח זמן ריצה ונכונות:

זמן ריצה ייצור הצמתים לינארי בזמן המשתנים, ייצור הקשתות הוא לכל היותר ריבועי בזמן הצמתים - סה"כ הכל פולינומי במספר הצמתים. חישוב k הוא לינארי במספר הפסוקיות.

נכונות נניח ש־$\phi \in 3-\text{SAT}$. בפרט, ϕ ספיקה, כלומר יש הצבה שנותנת ערך True לכל אחת מהפסוקיות. נבחר הצבה כזו, ונגדיר בעזרתה קבוצת צמתים: מכל שביעייה, הצומת שמסכים עם ההצבה. עוצמת הקבוצה היא k (אחד מכל שורה / שביעייה). לכל זוג פסוקיות ההצבה מסכימה על כל המשתנים המשותפים ביניהם , כלומר, במילים של הגרף, בין זוג צמתים בקבוצה יש קשת. מכאן, הקבוצה היא קליקה!

נניח כעת כי ב־G יש קליקה בגודל $m = k$. לכן, מכל שורה / שביעייה יש בדיוק צומת אחד בקליקה (כי צמתים באותה שביעייה לא מחוברים בקשת). נרצה לפרש את הקליקה כהצבה ל־ϕ. בכל פסוקית ניתן הצבה לפי הצומת שנבחר (אחת מ־7 ההשמות המספקות לפסוקית). נשים לב שההצבה מסכימה בערך שהיא נותנת לאותו המשתנה בפסוקיות שונות, כי בין כל שני צמתים בקליקה יש קשת.

מסקנה 8.2.2. קיבלנו השמה גלובלית לכל אחד מהמשתנים וכן היא השמה מספקת כל אחת מספוקיות ϕ - ומכאן ϕ ספיקה וסיימנו. □

משתי הטענות ראינו כי השפה אכן NP־שלמה. □

דוגמה 8.2.2. נתבונן בשפה

$$\text{SAT} = \left\{ \phi \;\middle|\; \begin{array}{l} \phi \text{ is a boolean formula, meaning a formula over } \{0,1\} \\ \text{separated by } \wedge, \vee \text{ signs. } \phi \text{ must be satisfiable, meaning there exists an assignment} \\ \text{such that } \phi \text{ of that assignment resolves to True} \end{array} \right\}$$

השפה SAT היא NP־שלמה.

הוכחה. נידרש לכמה טענות עזר:

טענת עזר 8.2.3. *מתקיים כי* $\text{SAT}, \text{CNF}-\text{SAT}, 3-\text{SAT} \in \text{NP}$.

הוכחה. נבנה מוודא פולינומי אשר יתאים לנו בעבור השפות הנ"ל - כאשר המוודא c הוא הצבה: □

Algorithm 4 Polynomial verifier for $\text{SAT}, \text{CNF}-\text{SAT}, 3-\text{SAT}$

Require: ϕ is of the correct format
Require: c is a set of $\{\text{True}, \text{False}\}$
Ensure: $\phi(c)$

```
if c does not fit φ then
    return False
end if
return φ(c)
```

תרגיל 8.2.1. הוכיחו כי שלוש השפות הללו NP־קשות.

משום ששתי הטענות נכונות, השפות הללו הן אכן NP־שלמות. □

דוגמה 8.2.3. נתבונן בשפה

$$\text{CNF}-\text{SAT} = \{\phi | \phi \text{ is a satisfiable CNF formula}\}$$

השפה $\text{CNF}-\text{SAT}$ היא NP־שלמה.

הוכחה. □

דוגמה 8.2.4. נתבונן בשפה

$$3-\text{SAT} = \{\phi | \phi \text{ is a satisfiable CNF formula, and each part has at most 3 variables (or their negations)}\}$$

השפה $3-\text{SAT}$ היא NP־שלמה.

הוכחה. □

משפט 8.2.1 (משפט קוק־לוין). *מתקיים כי* $\text{CNF}-\text{SAT}$ *היא* NP־*קשה.*

הוכחה. בשבוע הבא. □

דוגמה 8.2.5. נתבונן בשפה

$$\text{INDEPENDENT}-\text{SET} = \left\{ (G,k) \;\middle|\; \begin{array}{l} G \text{ is an undirected graph, which contains} \\ \text{an independent set of } k \text{ vertices} \end{array} \right\}$$

השפה $\text{INDEPENDENT}-\text{SET}$ היא NP־שלמה.

הוכחה. נידרש לשתי טענות עזר:

טענת עזר **8.2.4.** *מתקיים כי* $\text{INDEPENDENT} - \text{SET} \in \text{NP}$

הוכחה. העד יהיה קבוצה בגודל k שאמורה להיות ב"ת. הווידוא יבדוק בזמן פולינומי שגודל הקבוצה הוא k ושכל זוג בה לא מחובר. □

טענת עזר **8.2.5.** *מתקיים כי* $\text{INDEPENDENT} - \text{SET}$ *היא* NP-*קשה.*

הוכחה. נוכיח זאת עם רדוקציה $\text{CLIQUE} \leqslant_{\text{P}} \text{INDEPENDENT} - \text{SET}$. ברדוקציה זו, ניקח את הצלעות בקליקה, ונוריד אותן. קשת הופכת ללא קשת, ולא קשת הופכת לקשת.

תרגיל **8.2.2.** הוכיחו נכונות וזמן ריצה לרדוקציה זו.

□

וסיימנו.

מכאן הטענה נכונה. □