

שיעור 9

מבוא לסיבוכיות זמן

9.1 הגדרה של סיבוכיות זמן

הגדרה 9.1 זמן הריצה של מכונת טיורינג

זמן הריצה של מ"ט M על קלט w היא פונקציה $f(|w|)$ השווה למספר הצעדים הנדרש בחישוב של M על w .

הערה 9.1

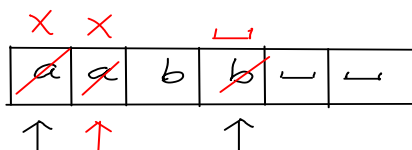
זמן הריצה של מ"ט M על קלט w , נמדד ביחס לגודל הקלט w , כלומר $f(|w|)$.

הגדרה 9.2 סיבוכיות זמן של בעיה/שפה

בהינתן קלט w באורך $n = |w|$. אומרים כי ניתן להכריעה שפה L בזמן $f(n)$ אם קיימת מ"ט M המכריעה את L כך שלכל $w \in \Sigma^*$, זמן הריצה של M על w חסום ע"י $f(|w|)$.

דוגמה 9.1 (דוגמה של סיבוכיות זמן של שפה)

נבנה מ"ט M עם סרט יחיד שמכריעה את השפה $L = \{a^n b^n \mid n \geq 0\}$.



התאור של M :

על קלט w :

(1) אם התו שמתחת לראש הוא $_$ \Leftarrow מקבלת.

(2) אם התו שמתחת לראש הוא b \Leftarrow דוחה.

(3) מוחקת את התו שמתחת לראש ע"י X .

(4) מזיזה את הראש ימינה עד התו הראשון משמאל ל- $_$.

• אם התו הוא a או X \Leftarrow דוחה.

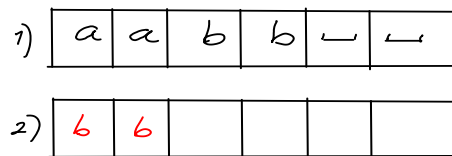
• מוחקת את התו שמתחת לראש ע"י $_$, מזיזה את הראש שמאלה עד התו הראשון מימין ל- X וחוזרת ל- (1).

זמן הריצה

- M מבצעת $\frac{|w|}{2}$ איטרציות.
- בכל איטרציה M סורקת את הסרט פעמיים וזה עולה $O(|w|)$.
- לכן סה"כ זמן הריצה של M חסום ע"י $\frac{|w|}{2} \cdot O(|w|) = O(|w|^2)$.

דוגמה 9.2 (דוגמה של סיבוכיות זמן של שפה)

נבנה מ"ט מרובת סרטים M' שמכריעה את השפה $L = \{a^n b^n \mid n \geq 0\}$.



התאור של M' :

על קלט w :

- (1) מעתיקה את ה- b -ים לסרט 2 (ותוך כדי בודקת האם w מהצורה a^*b^*).
 - (2) מזיזה את הראשים לתחילת הסרטים.
 - (3) אם שני הראשען מצביעים על $_$ \Leftarrow מקבלת.
 - (4) אם אחד הראשים מצביע על $_$ והשני לא \Leftarrow לא.
 - (5) מזיזה את שהע הראשים ימינה וחוזרת לשלב (3).
- שלב (3-5): $O(|w|)$.

זמן הריצה

זמן הריצה של M' הוא $O(|w|)$.

הגדרה 9.3 המחלקה $TIME$

תהי $f: \mathbb{N} \rightarrow \mathbb{R}^+$ פונקציה מהטבעים אל הממשיים האי-שליליים. המחלקה $TIME(f(n))$ היא האוסף של כל השפות שכריעות ע"י מכונת טיורינג $O(f(n))$.

דוגמה 9.3 (דוגמה של מחלקת-זמן של שפה)

עבור השפה L בדוגמה 9.1: $L \in TIME(n^2)$.

דוגמה 9.4 (דוגמה של מחלקת-זמן של שפה)

עבור השפה L בדוגמה 9.2: $L \in TIME(n)$.

9.2 יחס בין הסיבוכיות של מ"ט סרט יחיד ומטמ"ס

משפט 9.1

לכל מ"ט מרובת סרטים M הרצה בזמן $f(n)$ קיימת מ"ט סרט יחיד M' השקולה ל- M ורצה בזמן $O(f^2(n))$.

הוכחה:

בהינתן מ"ט מרובת סרטים M , הרצה בזמן $f(n)$, נבנה מ"ט עם סרט יחיד M' באותו אופן כמו בהוכחת השקילות במשפט 3.1.

כלומר, M' שומרת את התוכן של k סרטים של M על הסרט היחיד שלה (עם הפרדה ע"י #), ובכל צעד חישוב, M' סורקת את הסרט שלה כדי לזהות שת האותיות שמתחת לראשים (שמסומנות ב- $\hat{\alpha}$) ואחרי זה, משתמשת בפונקצית המעברים של M , וסורקת את הסרט פעם נוספת כדי לעדכן את התוכן בכל אחד מהסרטים ואת מיקום הראש בכל אחד מהסרטים.

1) 2)

⋮

 k)

#	$\hat{\alpha}_1$	#	$\hat{\alpha}_2$	#	$\hat{\alpha}_3$	#	
---	------------------	---	------------------	---	------------------	---	--

כמה לוקח ל- M' לסרוק את הסרט שלה? מכיוון שהסרט של M' מכיל את התוכן של k הסרטים של M , והגודל של כל אחד מהסרטים של M חסום ע"י $f(n)$, גודל הסרט של M' חסום ע"י

$$k \cdot f(n) = O(f(n)) .$$

העלות של הסריקה של M' לסרט שלה היא $O(f(n))$ וזה עלות של צעד חישוב בריצה של M' על הקלט.

מכיוון ש- M רצה בזמן $f(n)$, זמן היצרה של M' חסום ע"י

$$f(n) \cdot O(f(n)) = O(f^2(n)) .$$

■

9.3 יחס בין הסיבוכיות של מ"ט דטרמיניסטית ומ"ט א"ד

9.4 הגדרה

בהינתן מכונת טיורינג אי-דטרמיניסטית M , זמן הריצה של M על קלט w , היא פונקציה $f(|w|)$ השווה למספר הצעדים בחישוב המקסימלי של M על w .

9.2 משפט

לכל מכונת טיורינג אי-דטרמיניסטית N הרצה בזמן $f(n)$, קיימת מכונת טיורינג דטרמיניסטית D השקולה ל- N שרצה בזמן $2^{f(n)}$.

הוכחה:

בהינתן מכונת טיורינג אי-דטרמיניסטית N הרצה בזמן $f(n)$ מכונת טיורינג דטרמיניסטית D באותו אופן כמו בהוכחת השקילות במשפט 4.1.

כלומר, בהינתן קלט w , D תסרוק את עץ החישוב של N ו- w לרוחב ותקבל כל אחד החישובים של N המסתיים ב- q_{acc} .

בהינתן קלט w באורך n :

- כל מסלול בעץ החישוב של N על w חסום ע"י $f(n)$.
- מספר החישובים ש- D מבצעת חסום ע"י מספר הקודקודים בעץ החישוב של N ו- w .
- מכיוון שמספר הבנים של כל קודקוד בעץ החישוב חסום ע"י

$$C = 3|Q| \cdot |\Gamma|$$

מספר הקודקודים בעץ החישוב חסום ע"י

$$C^0 + C^2 + \dots C^{f(n)} \leq C^{f(n)+1} = C \cdot C^{f(n)} .$$

ולכן זמן הריצה של D חסום ע"י

$$f(n) \cdot C \cdot C^{f(n)} \leq C^{f(n)} \cdot C^{f(n)} = C^{2f(n)} = (C^2)^{f(n)} = 2^{C' \cdot f(n)} = 2^{O(f(n))} .$$

נתייחס כאן לשני החסמים הבאים:

■

(1) חסם פולינומיאלי הוא חסם מהצורה n^c עבור $c > 0$ כלשהו.

(2) חסם אקספוננציאלי הוא חסם מהצורה 2^{n^c} עבור $c > 0$ כלשהו.

הגדרה 9.5 בעיית הכרעה

בעיית הכרעה מוגדרת באופן הבא:

"בהינתן קלט כלשהו, האם הקלט מקיים תנאי מסוים"

דוגמה 9.5

בהינתן מספר n , האם n ראשוני?

כל בעיית הכרעה ניתן לתאר כשפה שקולה:

$$L_{\text{prime}} = \{ \langle n \rangle \mid n \text{ ראשוני} \}.$$

משפט 9.3

. שפה \equiv בעיית הכרעה

הגדרה 9.6 אלגוריתם זמן פולינומיאלי

אומרים כי אלגוריתם A מכריעה בעייה בזמן פולינומיאלי אם קיים קבוע $c > 0$ כך שזמן הריצה של A על כל קלט w חסום ע"י $O(|w|^c)$.

משפט 9.4 התזה של צירץ' (Church Thesis)

אם קיים אלגוריתם המכריע בעייה בזמן פולינומיאלי, אז קיימת מ"ט דטרמיניסטית המכריעה את השפה השקולה לבעייה זו בזמן פולינומיאלי.

. מכונת טיורינג \equiv אלגוריתם מכריעה

9.4 המחלקה P **הגדרה 9.7 המחלקה P**

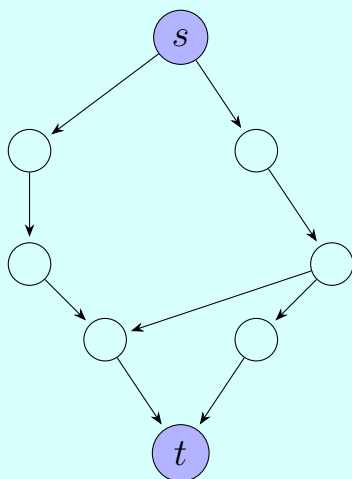
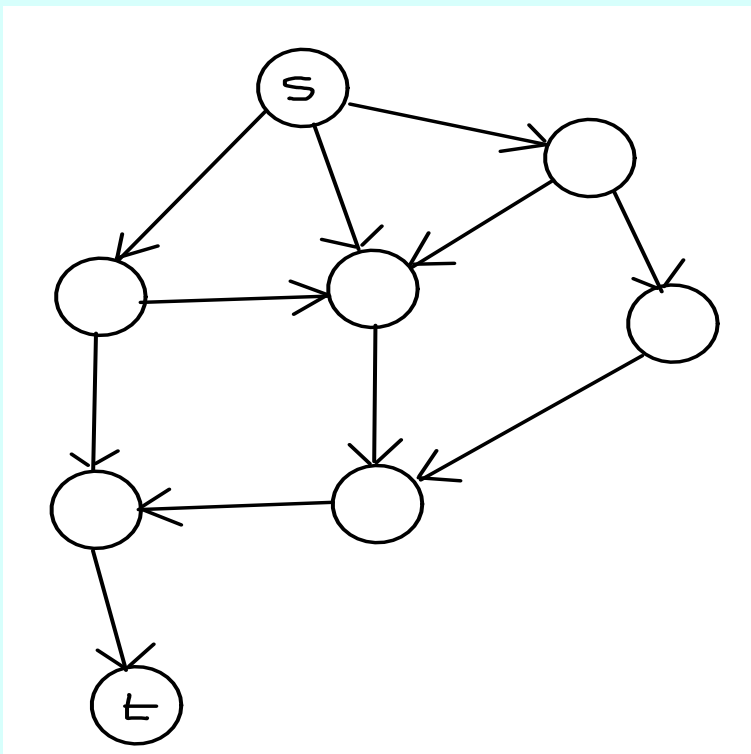
המחלקה P היא אוסף כל הבעיות (השפות) שקיים עבורן אלגוריתם (מכונת טיורינג דטרמיניסטית) המכריע אותן בזמן פולינומיאלי.

דוגמה 9.6

$$L = \{ a^n b^n \mid n \geq 0 \} \in P.$$

9.5 בעיית PATH

הגדרה 9.8 בעיית המסלול בגרף מכוון



קלט: גרף מכוון $G = (V, E)$ ושני קודקודים $s, t \in V$.

פלט: האם קיים מסלול ב- G מ- s ל- t ?

$$PATH = \{ \langle G, s, t \rangle \mid \text{קיים מסלול ב-} G \text{ מ-} s \text{ ל-} t \}$$

משפט 9.5

$$PATH \in P.$$

הוכחה: נבנה אלגוריתם A עבור הבעיה $PATH$.

$A = \langle G, s, t \rangle$ על קלט

(1) צובע את s .

(2) מבצע $|V| - 1$ פעמים:

• לכל צלע $(u, v) \in E$:

* אם u צבוע \Leftarrow צבע את v .

(3) • אם t צבוע \Leftarrow החזיר "כן".

• אחרת \Leftarrow החזיר "לא".

זמן ריצה של האלגוריתם הוא $O(|V| \cdot |E|)$ פולינומיאלי במספר הקודקודים $|V|$.

האם זה פולינומיאלי בגודל הקלט $|\langle G \rangle|$?

איך נקודד את G ?

• נניח כי $|V| = n$ ו- $V = \{1, 2, 3, \dots, n\}$.

• נניח כי הצלעות נתונות ע"י מטריצה M בגודל $n \times n$ כך ש-

$$M_{ij} = \begin{cases} 1 & (i, j) \in E \\ 0 & (i, j) \notin E \end{cases}.$$

• נניח כי מספרים מקודדים בבסיס ביניארי.

• אזי גודל הקידוד של G שווה $n^2 + n \log_2 n$, כלומר

$$|\langle G \rangle| = \Omega(|V|^2) \Rightarrow |V| = O(|\langle G \rangle|).$$

ולכן כל אלגוריתם הרץ בזמן פולינומיאלי במספר הקודקודים $|V|$ ירוץ בזמן פולינומיאלי בגודל הקידוד $|\langle G \rangle|$.

■

ולכן A רץ בזמן פולינומיאלי בגודל הקלט.

9.6 הבעיית RELPRIME

הגדרה 9.9 מספרים זרים (Relatively prime)

אומרים כי שני מספרים שלמים x, y הם זרים אם המחלק המשותף הגדול ביותר, מסומן $\gcd(x, y)$, שווה 1.

הגדרה 9.10 בעיית RELPRIME

קלט: שני מספרים x ו- y .

פלט: האם x ו- y זרים?

$$RELPRIME = \{ \langle x, y \rangle \mid \gcd(x, y) = 1 \} .$$

אנחנו נוכיח כי ניתן להכריע את $RELPRIME$ בזמן פולינומיאלי, כלומר נוכיח $RELPRIME \in P$ במשפט 9.8 למטה. לפניכן נסביר את האלגוריתם של אוקלידס למציאת ה- \gcd של שני שלמים, ומתוך זה נוכל לחשב את הסיבוכיות זמן של $RELPRIME$. ראשית נזכיר משפט שלמדנו בקורסים קודמים:

משפט 9.6 השמפט של האלגוריתם של אוקלידס

אם x, y שלמים אז

$$\gcd(x, y) = \begin{cases} x & y = 0 \\ \gcd(y, x \bmod y) & y \neq 0 \end{cases} .$$

■ **הוכחה:** ההוכחה היא לא חלק של הקורס ומופיע בסעיף האחרון "הוכחות של משפטים שימושיים" בדף 103.

האלגוריתם של אוקלידס הוא אלגוריתם, שמקבל כקלט שני מספרים x, y ופולט את $\gcd(x, y)$. הוא מתבוסס על המשפט 9.6. האלגוריתם עצמו הוא כדלקמן:

$EUCLID =$ על קלט $\langle x, y \rangle$ כאשר x, y מספרים שלמים בבסיס בינארי:

(1) כל עוד $y \neq 0$:

(2) $x \leftarrow x \bmod y$

(3) $\text{swap}(x, y)$

(כלומר מחליפים בין x ו- y).

(4) מחזירים את x .

לדוגמה:

$$\gcd(18, 32) = \gcd(32, 18) = \gcd(18, 14) = \gcd(14, 4) = \gcd(4, 2) = \gcd(2, 0) = 2 .$$

כדי להוכיח כי $RELPRIME \in P$ נצטרך למשפט עזר הבא:

משפט 9.7 (משפט עזר)

אם $x > y$ אז $x \bmod y < \frac{x}{2}$.

■ **הוכחה:** ההוכחה היא לא חלק של הקורס ומופיע בסעיף האחרון "הוכחות של משפטים שימושיים" בדף 104.

משפט 9.8

$$RELPRIME \in P .$$

הוכחה:

נבנה אלגוריתם A המכריע את $RELPRIME$ בזמן פולינומיאלי. $RELPRIME$ היא השפה של הבעיה, שמקבלת כקלט שני מספרים שלמים x, y ומחזירה תשובה לשאלה, האם x, y זרים. כלומר:

$$\langle x, y \rangle \in RELPRIME \iff \gcd(x, y) = 1.$$

לכן A משתמש בהאלגוריתם של אוקלידס $EUCLID(x, y)$ כדי לחשב $\gcd(x, y)$.

בניית האלגוריתם A המכריע $RELPRIME$:

$A =$ "על קלט $\langle x, y \rangle$ כאשר x, y שלמים בבסיס בינארי: A מריץ את $EUCLID$ על x ו- y .

• אם $EUCLID(x, y)$ מחזיר $\gcd(x, y) = 1$ אז A מקבל.

• אחרת A דוחה.

הוכחת הנכונות

הנכונות של A מנובעת ישר מהנכונות של האלגוריתם האוקלידס, $EUCLID$.

סיבוכיות זמן

נראה כי A רץ בזמן פולינומיאלי בגודל הקלט $\langle x, y \rangle$.

- לפי משפט עזר 9.7: $x \bmod y < \frac{x}{2}$.
- בכל איטרציה, בשלב (2) המשתנה x מקבל את הערך החדש $x \leftarrow x \bmod y$.
- לכן בכל איטרציה הערך החדש של x קטן ממש מחצי של הערך הקודם של x .
- לכן אחרי כל איטרציה, x קטן בלפחות חצי.
- בשלב (3), A מחליף בין x ו- y , אז אחרי כל 2 איטרציות, גם x קטן בלפחות חצי וגם y קטן בלפחות חצי.
- לכן המספר הפעמים המקסימלי שאפשר לבצע שלבי (2) ו-(3) היא $m = \min(2 \lfloor \log_2 x \rfloor, 2 \lfloor \log_2 y \rfloor)$.
- לכן המספר האיטרציות המקסימלי של $EUCLID$ הוא $m = \min(2 \lfloor \log_2 x \rfloor, 2 \lfloor \log_2 y \rfloor)$.
- יהי n האורך של הקלט. כלומר $n =$ האורך של המחרוזות של השלמים x ו- y בבסיס בינארי.

$$m \leq n$$

- לכן A דורש $O(n)$ צעדים.
- כל איטרציה של $EUCLID$ מתבצע בזמן פולינומיאלי.
- כלומר קיים טבעי k עבורו כל איטרציה של $EUCLID$ מתבצע בזמן $O(n^k)$.
- לכן $EUCLID \in TIME(n^{k+1})$ (ראו הגדרה 9.3) ולכן $A \in TIME(n^{k+1})$.

לכן A רץ בזמן פולינומיאלי בגודל הקלט. לכן

$$RELPRIME \in P.$$



9.7 *הוכחות של משפטים שימושיים

משפט 9.9 השמפט של האלגוריתם של אוקלידס

אם x, y שלמים אז

$$\gcd(x, y) = \begin{cases} x & y = 0 \\ \gcd(y, x \bmod y) & y \neq 0 \end{cases}.$$

הוכחה: (להעשרה בלבד)

המטרה של ההוכחה הזו היא רק להוסיף הבנה להוכחה של משפט 9.8 לסיבוכיות זמן של $RELPRIME$ למטה. היא לא הוכחה שאתם תיבחנו עליה ואפשר לדלג עליה.

נתחיל אם משפט החילוק של אוקלידס, שאומר שאם x, y שלמים אז קיימים שלמים q ו- $0 \leq r < y$ כך ש:

$$x = qy + r = \left\lfloor \frac{x}{y} \right\rfloor y + (x \bmod y). \quad (1*)$$

נגדיר $d \triangleq \gcd(x, y)$.

מכיוון ש- d הוא מחלק משותף של x ו- y אז $d \mid x$ וגם $d \mid y$. לכן בזכות משוואה (1*):

$$(d \mid x) \wedge (d \mid y) \xrightarrow{\text{משוואה (1*)}} d \mid (x \bmod y)$$

ז"א $d \mid y$ וגם $d \mid (x \bmod y)$ אז בהכרח:

$$d \mid \gcd(y, x \bmod y). \quad (2*)$$

כעת נגדיר $\bar{d} \triangleq \gcd(y, x \bmod y)$.

מכיוון ש- \bar{d} הוא מחלק משותף של y ו- $x \bmod y$ אז $\bar{d} \mid y$ וגם $\bar{d} \mid x \bmod y$. לכן בזכות משוואה (1*):

$$(\bar{d} \mid y) \wedge (\bar{d} \mid x \bmod y) \xrightarrow{\text{משוואה (1*)}} \bar{d} \mid x$$

ז"א $\bar{d} \mid y$ וגם $\bar{d} \mid x$ אז בהכרח:

$$\bar{d} \mid \gcd(x, y). \quad (3*)$$

לסיכום, לפי משוואות (2*) ו- (3*):

$$d \mid \bar{d} \quad \wedge \quad \bar{d} \mid d.$$

מכיוון ש- $d, \bar{d} > 0$ אז בהכרח $d = \bar{d}$, ז"א $\gcd(x, y) = \gcd(y, x \bmod y)$. ■

משפט 9.10 (משפט עזר)

אם $x > y$ אז $x \bmod y < \frac{x}{2}$.**הוכחה:** יש שני מקרים:

$$y \leq \frac{x}{2} \quad (1)$$

$$y > \frac{x}{2} \quad (2)$$

נוכיח את הטענה עבור שני המקרים.

מקרה 1: $y \leq \frac{x}{2}$.

לפי משפט החילוק של אוקלידס אם x, y שלמים עבורם $x > y$ אז קיימים $q = \left\lfloor \frac{x}{y} \right\rfloor$ ו- $r = x \bmod y$ כך ש $0 \leq r < y$ -

$$x = qy + r = \left\lfloor \frac{x}{y} \right\rfloor y + (x \bmod y) .$$

בפרט אם $r < y$ וגם $y \leq \frac{x}{2}$ לפיכך $x \bmod y < y \leq \frac{x}{2}$.

מקרה 2: $y > \frac{x}{2}$.

לפי משפט החילוק של אוקלידס אם x, y שלמים עבורם $x > y$ אז קיימים שלם $q = \left\lfloor \frac{x}{y} \right\rfloor$ ושלם $r = x \bmod y$ כך ש $0 \leq r < y$ -

$$x = qy + r = \left\lfloor \frac{x}{y} \right\rfloor y + (x \bmod y) .$$

בפרט אם $y > \frac{x}{2}$ אז $x < 2y$. אז בהכרח $q < 2$. מכיון ש- $x > y$ ו- $q = \left\lfloor \frac{x}{y} \right\rfloor$ אז הערך המינימלי של q הוא $q = 1$. לכן אם $q < 2$ בהכרח $q = 1$. לכן יש לנו

$$x = qy + r = (1)y + r + (x \bmod y) .$$

מכאן

$$x - y = x \bmod y .$$

כעת נציב את ההנחה ההתחלתית $y > \frac{x}{2} \Leftrightarrow x - y < \frac{x}{2}$ ונקבל

$$x \bmod y < \frac{x}{2} .$$

