

## שיעור 9

# מבוא לסיבוכיות זמן

### 9.1 הגדרה של סיבוכיות זמן

עד כה כל הבעיות החישוביות שעשכנו בהן הניחו שהמשאים שעומדים לרשות מכונת הטיורינג שפותרת אותן הם **בלתי מוגבלים**. כתע נ עבור לעסוק בשאלת מה קורה כאשר אנחנו מוגבלים חלק ממשאים אלו. יש סוגים רבים של ממשאים שנייתן לעסוק בהם, אבל שני הנפוצים ביותר בתיאוריה של מדעי המחשב הם

- **זמן חישוב**,

- **הזיכרון שנדרש לצורך החישוב**.

את מה בעיות שבוחן נתקלים:  
 כשמעוניינים למדוד את צרכית המשאים הללו של אלגוריתם מסוים היא שלא ברור כיצד למדוד אותם.  
 האם זמן חישוב נמדד בשניות?  
 אם כן, כיצד ניתן לחשב את זמן החישוב עבור אלגוריתם נתון?  
 האם علينا לקודד ולהריץ אותו על מחשב מסוים?

אבל במחשבים שונים האלגוריתם ירוץ זמן זנים שונים בשל

- **יעילות המעבד**,

- **אופטימיזציות שמתבצעות ברמת המעבד**,

- **אופטימיזציות בזמן הקומפליציה**,

ובווצה בוחן.

אפילו תנאים חיצוניים כמו החום בסביבת המעבד עשויים להשפיע על זמן הריצה. מכאן הרצון למצוא הגדרה **תיאורטיבית** של זמן ריצה, שאינה תלולה בחומרה זו או אחרת.

#### הערה 9.1

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

#### הגדרה 9.1 זמן הריצה של מכונת טיורינג

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

#### הגדרה 9.2 סיבוכיות זמן של בעיה/שפה

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

**דוגמה 9.1 סיבוכיות זמן של השפה של מחרוזות האוניריות**

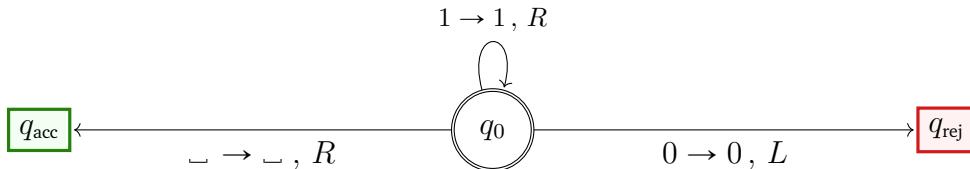
נתבונן על השפה של מחרוזות האוניריות הבאה:

$$L = \{1^n \mid n \geq 1\}.$$

נבנה מכונת טיורינג הבאה שמכריעה אותה:

$$M = (Q, q_0, \Sigma, \Gamma, \delta, q_{acc}, q_{rej})$$

כאשר  $\Sigma = \{0, 1\}$  ו-  $\Gamma = \{0, 1\}$ . המ מצבים והמעברים מתוארים בהתרשים מצבים של מטפה:



בכדי לחשב את הסיבוכיות זמן של  $L$  בהתאם למוגנה בפסאודוקוד. על כל קלט  $w$ :

- (1) • אם המילה היא מילת הריקה תדחה.
- אחרת ממשיכה לשלב (2).
- (2) • אם התו הנקרא 0 תדחה.
- אחרת אם התו הנקרא הוא ⊢ מקבל.
- אחרת חוזרת לשלב (2).

כל שהקלט ארוך יותר, כך  $M$  תבצע צעדי חישוב נוספים יותר. בפרט המספר הצעדים המקסימלי הוא במקרה שבקלט  $w$  יש רק תווי 1 ולכן המ"ט תבצע  $|w| = n$  צעדי חישוב. בכל מקרה אחר היא תבצע פחות מ-  $n$  צעדים.

$$\begin{aligned} &\Leftarrow \text{חסם העליון של מספר צעדי חישוב של } M \text{ על קלט } w \text{ הוא } n \text{ כאשר } |w| = n. \\ &\Leftarrow M \text{ עוצרת בזמן } (n) \\ &\Leftarrow L \text{ כריעה בזמן } (n).O \\ &\Leftarrow L \in TIME(n) \end{aligned}$$

באופן כללי, אם מכונה על קלט  $w$  מבצעת פחות מ-  $|w|$  צעדי חישוב, המשמעות היא המכונה כלל לא קראה את כל הקלט, וזה אינו נדיר במיוחד (אם כי הוא בחילוט קיים). אם כן ברור שمدידות זמן חריצת היא תמיד **ביחס לאורך הקלט**.

**דוגמה 9.2**

בדוגמה זו נבנה מכונת טיורינג  $M$  עם סרט ייחיד שמכריעת את השפה  $L = \{a^n b^n \mid n \geq 0\}$  ואז נחשב את הסיבוכיות זמן שלה.

**בנייה המכונת טיורינג**

"על קלט  $w$ :

- (1) אם התו הנקרא הוא ⊢ מקבל.

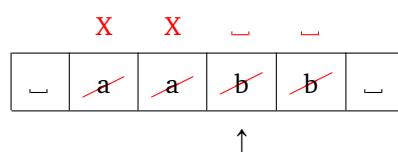
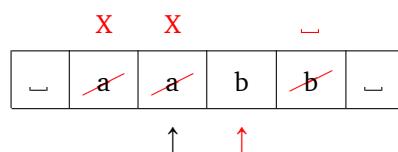
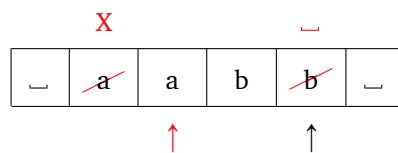
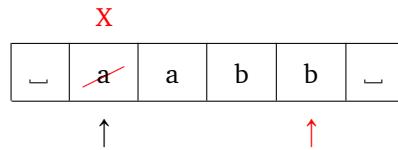
(2) אחרת אם הtau הנקרא הוא  $M \Leftarrow b$  דוחה.

(3) אחרת מוחקמת את הtau הנקרא ע"י  $X$ .

(4) מזיהה את הראש ימינה עד הtau הראשון משמאלו ל-  $\sqsubset$  בקצבה הימין של הקלט.

- אם הtau הוא  $a$  או  $M \Leftarrow X$  דוחה.

• אחרת מוחקמת את הtau הנקרא ע"י  $\sqsubset$ , מזיהה את הראש שמאלה עד הtau הראשון מימין ל-  $\sqsubset X$  וחוזרת לשלב (1).



### זמן הריצה

נסמן את אורך הקלט ב-  $n = |w|$ .

•  $M$  מבצעת  $\left\lfloor \frac{n}{2} \right\rfloor$  איטרציות.

• בכל איטרציה  $M$  סורקת את הסרט פעמיים וזה עולה ( $O(n)$  צעדים).

• לכן סה"כ הזמן הריצה של  $M$  חסום ע"י

$$\frac{n}{2} \cdot O(n) = O(n^2).$$

### דוגמה 9.3

כעת נבנה מכונת טיורינג מרובת סרטים'  $M'$  שמכריעת השפה  $L = \{a^n b^n \mid n \geq 0\}$  ונראה שהסיבוכיות זמן שונה מזו של המכונת טיורינג עם סרט יחיד שראינו בדוגמה הקודמת.

### בנייה המכונה

:*w* על קלט" = *M'*

- .*O(n)* (1) מעתיקת את ה-  $b$ -ים לסרטט 2 (ותוך כדי לבדוק האם  $w$  מהצורה  $a^*b^*$ ).

.*O(n)* (2) מזיהה את הראשים לתחילת הסרטים.

.*O(n)* (3) אם שני הראשים מצביעים על  $\_ \Leftarrow$  מקבלת.

.*O(n)* (4) אם אחד הראשים מצביע על  $\_ \Leftarrow$  והשני לא  $\Leftarrow$  תדחה.

.*O(n)* (5) מזיהה את שני הראשים ימינה וחוזרת לשלב (3). “

**שלבים (3-5):**

. $O(n)$  : (3-5)

(1) סרט \_\_\_\_\_ a a b b \_\_\_\_\_

סרט (2) \_\_\_\_\_ b b \_\_\_\_\_

סיבות זמן

נסמן את אורך הקלט ב-  $n$ .

זמן הריצה של  $M'$  הוא  $O(n)$

**הגדלה TIME ( $f(n)$ ) 9.3**

נגדיר הקבוצת השפות  $TIME(f(n))$  כך שלכל שפה  $L \in TIME(f(n))$  קיימת מכונת טיריניג דטרמיניסטית שמכריעה את  $L$  בזמן לכל היותר  $f(n)$ , כאשר  $n$  הוא האורך של הקלט:

$$TIME(f(n)) = \{L \mid O(f(n)) \text{ בזמן } L\}.$$

#### 9.4 דוגמה

### عبر השפה בדוגמה 9.2:

$$L \in TIME(n^2) .$$

## 9.5 דוגמה

### عبرור השפה בדוגמה 9.3:

$$L \in TIME(n) \enspace .$$

## 9.6 דוגמה

הדוגמה זו ראינו בדוגמה 1.5. נבנה מכונת טיריניג  $M$  עם סרט ייחד שמכריעת את השפה  $L = \{a^{2^n} \mid n \geq 0\}$  וזו נחשב את הסיבוכיות זמן שלה.

הרעין

המכונה תסrox את הסרט שלא משMAL לימין ובכל איטרציה תמתק חצי מה- a -ים שנשארו.

בנייה המכונה

" על קלט  $w = M$

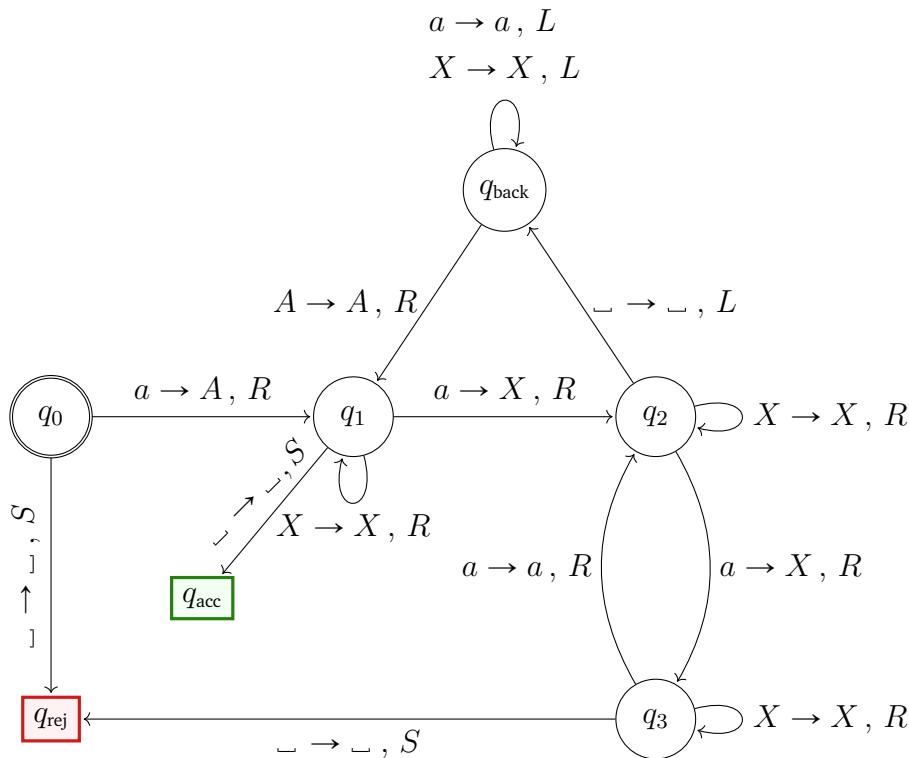
(1) אם התו הנكرة הוא  $\_ \Leftarrow \text{דוחה.}$

(2) אחרת מחליפה את ה-  $a$  הראשון ב-  $A$  (כדי לסמן את תחילת הסרט).

(3) סורקת את הסרט משמאלה לימין ומוחקמת כל  $a$  שני:

- אם הסרט מכיל  $a$  יחיד  $\Leftarrow \text{מקבלת.}$
- אם הסרט מכיל מספר אי-זוגי של  $a$  -ים  $\Leftarrow \text{דוחה.}$
- מזיהה את הראש שמאליה לתחילת הסרט וחוזרת לשלב (3)."

התרשימים מצבאים של המוכנה מתואר באירור למטה:



### סיבוכיות זמן

- שלב (3) דורש  $n$  צעדים לכל היותר.
- המוכנה חוזרת על שלב (3) לכל היותר  $n$  פעמים.
- לפיכך הסיבוכיות זמן של השפה  $L$  היא  $O(n^2)$ . קלומר:

$$L \in TIME(n^2)$$

### **דוגמה 9.7 חיסור בינארי**

בדוגמה זו נבנה מכונת טירינג מ- 3 סרטים שמקבלת קלט שני מספרים שלמים  $y > x$  (בבסיס בינארי ומחשבת את החיסור  $y - x$ ).

### בנייה המכונה

$x - y$  הנקרא  $SUBTRACT$  "על קלט  $\langle x, y \rangle$  כאשר  $x$  שלמים בסיס בינארי ו-  $y > x$ :

(1) רושמת את  $x$  ב סרט 1 ואת  $y$  ב סרט 2 משמאלי לימין כך שהתאים עם הקצוות הימניות של  $x$  ו-  $y$  מיושרים זה מתחת זהה. בתחילת הסרט 3 ריק.

(2) מעמידה את ראש הסרטים 3, 2, 1 על התאים המיושרים של הקצוות הימניות של הקלטים. בפרט, ראש הסרטים 1 ו- 2 נמצאים על הספרות הפחות משמעותיות של  $x$  ו-  $y$ , וראש הסרט 3 נמצא בתא שמתוחם.

(3) שלב זה מבצע חישור ללא חוב.

יהי  $\sigma_1 \in \{0, 1\}^*$ תו הנקרא הסרט 1 והוא  $\sigma_2 \in \{0, 1\}^*$ תו הנקרא הסרט 2.

- אם  $(\sigma_1, \sigma_2) = (\_, \_)$  אז  $M$  מקבלת.
- אם  $(0, 0, 0)$ , כותבת  $(L, L, L)$ , מזינה את הראשים  $(\sigma_1, \sigma_2) = (0, 0)$  וחזרת על שלב זה.
- אם  $(1, 0, 1)$ , כותבת  $(L, L, L)$ , מזינה את הראשים  $(\sigma_1, \sigma_2) = (1, 0)$  וחזרת על שלב זה.
- אם  $(1, 1, 0)$ , כותבת  $(L, L, L)$ , מזינה את הראשים  $(\sigma_1, \sigma_2) = (1, 1)$  וחזרת על שלב זה.
- אם  $(\_, 0, 0)$ , כותבת  $(S, L, L)$ , מזינה את הראשים  $(\sigma_1, \sigma_2) = (\_, 0)$  וחזרת על שלב זה.
- אם  $(\_, 1, 1)$ , כותבת  $(S, L, L)$ , מזינה את הראשים  $(\sigma_1, \sigma_2) = (\_, 1)$  וועברת לשלב הבא.
- אם  $(0, \_, 0)$ , כותבת  $(L, S, L)$ , מזינה את הראשים  $(\sigma_1, \sigma_2) = (0, \_)$  וחזרת על שלב זה.
- אם  $(1, \_, 1)$ , כותבת  $(L, S, L)$ , מזינה את הראשים  $(\sigma_1, \sigma_2) = (1, \_)$  וחזרת על שלב זה.
- אם  $(0, 1, 1)$ , כותבת  $(L, L, L)$ , מזינה את הראשים  $(\sigma_1, \sigma_2) = (0, 1)$  וועברת לשלב הבא.

(4) שלב זה מבצע חישור כאשר קיים חוב.

- אם  $(1, 0, 0)$ , כותבת  $(L, L, L)$ , מזינה את הראשים  $(\sigma_1, \sigma_2) = (1, 0)$  וועברת לשלב הקודם.
- אם  $(0, 1, 1)$ , כותבת  $(L, L, L)$ , מזינה את הראשים  $(\sigma_1, \sigma_2) = (0, 1)$  וחזרת על שלב זה.
- אם  $(1, 1, 1)$ , כותבת  $(L, L, L)$ , מזינה את הראשים  $(\sigma_1, \sigma_2) = (1, 1)$  וחזרת על שלב זה.
- אם  $(0, 0, 1)$ , כותבת  $(L, L, L)$ , מזינה את הראשים  $(\sigma_1, \sigma_2) = (0, 0)$  וחזרת על שלב זה.
- אם  $(\_, 0, 1)$ , כותבת  $(S, L, L)$ , מזינה את הראשים  $(\sigma_1, \sigma_2) = (\_, 0)$  וחזרת על שלב זה.
- אם  $(\_, 1, 0)$ , כותבת  $(S, L, L)$ , מזינה את הראשים  $(\sigma_1, \sigma_2) = (\_, 1)$  וחזרת על שלב זה.
- אם  $(0, \_, 1)$ , כותבת  $(L, S, L)$ , מזינה את הראשים  $(\sigma_1, \sigma_2) = (0, \_)$  וחזרת על שלב זה.
- אם  $(1, \_, 1)$ , כותבת  $(L, S, L)$ , מזינה את הראשים  $(\sigma_1, \sigma_2) = (1, \_)$  וועברת לשלב הקודם.

### סיבוכיות זמן

יהי  $|y| = n$ . המכונה  $SUBTRACT$  תורק את שני הסרטים שביהם כתובים המספרים  $x$  ו-  $y$  במקביל, ותוך כדי רושמת את הפלט על סרט 3. לכן  $SUBTRACT$  מבצעת ( $O(n)$  פעדים).

לכן הסיבוכיות זמן של  $SUBTRACT$  היא ( $O(n)$  ליניארי).

## דוגמה 9.8 האלגוריתם החילוק של אוקלידס

המשפט החילוק של אוקלידס אומר שבහינתנו שני מספרים שלמים  $x, y$ , אזי קיימים שלמים  $q, r$  כך ש

$$x = qy + r, \quad 0 \leq r < |y|.$$

במקרה הפרטי ש-  $x < y < 0$  אז:

$$q = \left\lfloor \frac{x}{y} \right\rfloor, \quad r = x \bmod y.$$

קיים אלגוריתם שמקבל כקלט  $y, x$  ונותן כפלט  $q$  ו-  $r$ . האלגוריתם עובד לכל שלמים  $y, x$  ( בלי קשר לסימן שלהם ). אנחנו נסתכל על האלגוריתם במקרה הפרטי ש-  $x < y < 0$ , כמוותואר למטה .

### בנייה האלגוריתם

:  $0 < y < x$  כאשר  $y, x$  שלמים בבסיס בינארי ו-  $x = DIVISION$

$$\left. \begin{array}{l} q \leftarrow 0 \\ r \leftarrow x \end{array} \right\} \text{1) מAtlantic}$$

(2) כל עוד ש  $r \geq y$

$$r \leftarrow r - y \quad \text{3}$$

$$q \leftarrow q + 1 \quad \text{4}$$

(5) פלט:  $q, r$ .

### סיבוכיות זמן

נסמן  $| \langle x, y \rangle | = n$  אורך הקלט.

- $DIVISION$  מבצע  $r$  איטרציות לכל היותר.
- $y < r$  לכן  $y$  הוא חסם עליון של המספר האיטרציות המקסימלי של  $DIVISION$ .
- לכן  $DIVISION$  מבצע  $O(n)$  איטרציות לכל היותר.
- בכל איטרציה  $DIVISION$  מבצע חיסור בינארי עם  $SUBTRACT$  אשר (כפי שראינו בדוגמה 9.7) הוא  $O(n)$ .
- לכן הסיבוכיות זמן של  $DIVISION$  היא  $O(n^2)$ .

## 9.2 יחס בין הסיבוכיות של מכונת טיורינג עם סרט יחיד ומכונת טיורינג מרובת סרטים

### משפט 9.1

לכל מכונת טיורינג מרובה סרטים  $M$  הרצה בזמן  $t(n)$  קיימת מכונת טיורינג סרט יחיד  $M'$  השköלה ל- $M$  ורצה בזמן  $O(t^2(n))$ .

**הוכחה:**

תהי  $M$  מכונת טיורינג כלשהי עם  $k$  סרטים הרצה בזמן  $O(t(n))$ .

نبנה מכונת טיורינג  $S$  עם סרט אחד שרצה בזמן  $O(t^2(n))$ .

- ראשית נרשום את התוכן של  $k$  הסרטים של  $M$  על הסרט היחיד של  $S$ .
- התכנים של הסרטים מופרדים על ידי תו '#' על הסרט היחיד.
- המיקומים של הראשים של כל הסרטים של  $M$  מסומנים על הסרט היחיד על ידי חצים.

להלן יש דוגמה לכך של מכונת טיורינג עם 3 סרטים:

המכונת טיורינג  $M$

0	1	0	1	0	-	-
↑						

a	a	a	-	-	-	-
↑						

a	b	a	b	-	-	-	-
↑							

המכונת טיורינג  $S$

#	0	1	0	1	0	#	a	a	a	#	a	b	b
↑				↑			↑				↑		

אפשר לסמלו צעד חישוב אחד של  $M$  במכונת טיורינג  $S$  באופן הבא:

בנויות המכונה  $S$

- 1) תחילת  $S$  מתחילה את הסרט שלה בלבסוף את התכנים של  $k$  הסרטים על הסרט היחיד שלה, עם תו '#' להפריד בין סרטים שונים של  $M$ .
- 2) בנוסף  $S$  רושמת תו '#' בקצה השמאלי כדי לסמנו את התחלת הקלט ותו '#' בקצת הימין כדי לסמנו את סוף הקלט.

3) ב כדי לסייע צעד חישוב אחד של  $M$  בהמוניה  $S$ , המכונה  $S$  سورקת את הסרט מ- # הראשון ל- # האחרון. ב巡视ה זו  $S$  זכרת את המיקומים של ה-  $k$  ראשים על פי התאים שמסומנים עם חצים, באמצעות  $k$  תא זכרון.

4) אחר כך  $S$  מבצעת סיריקה שנייה של הסרט. ב巡视ה זו, לפי הפונקציית המעברים של  $M$ , המכונה  $S$  מבצעת, לכל  $1 \leq i \leq k$

- כתיבה של הסימן החדש בסרט ה- $i$  במקום הנוכחי של הראש של סרט ה- $i$ .
- תזוזה של הראש של סרט ה- $i$ .

5) במצב שהראש של אחד הסרטים של  $M$  זו ימינה לתו רוח מצד ימין של סוף הקלט, אז  $S$  תוסיף תא אחד עםתו רוח מצד שמאל של ה- # המפריד בין סרטים, ולאחר כך היא תזוז את כל התאים שבצד ימין של התא המוסף מקום אחד ימינה.

### סיבוכיות זמן של המכונה

יהי  $n$  האורך של התוכן הכי ארוך מthan כל התכנים של ה-  $k$  סרטים של  $M$ .

• האורך של התוכן של  $S$  שווה לסכום הארכים של התכנים של ה-  $k$  סרטים של  $M$ .

• נתון שהזמן הריצה של  $M$  הוא  $O(t(n))$ .

$\Leftarrow M =$  עשה שימוש של  $O(t(n))$  תאים.

$\Leftarrow$  אורך הקלט של  $S$  חסום מלמעלה ע"י  $O(t(n))$ .

$\Leftarrow$  סיריקה שלמה של הקלט של  $S$  מתבצעת ב-  $O(t(n))$  פעמים.

$\Leftarrow$  מכיוון שכל סיריקה דורשת זמן  $O(t(n))$  וכי לדמות צעד אחד של  $M$ , המכונה  $S$  מבצעת שתי סיריקות, אז  $S$  לוקחת זמן  $O(t(n))$  לבצע צעד חישוב אחד של  $M$ .

$\Leftarrow$  הסיבוכיות זמן של  $S$  היא:

$$O(t(n)) O(t(n)) = O(t^2(n))$$

## 9.3 יחס בין הסיבוכיות של מכונת טירינג דטרמיניסטיבית ומכונת טירינג אי דטרמיניסטיבית

### משפט 9.2

לכל מכונת טירינג אי-דטרמיניסטיבית  $N$  הרצה בזמן  $f(n)$ , קיימת מכונת טירינג דטרמיניסטיבית  $D$  השකולה לה-  $N$  ורצה בזמן  $2^{f(n)}$ .

הוכחה:  
בhinintן מכונת טירינג אי-דטרמיניסטיבית  $N$  הרצה בזמן  $f(n)$  מכונת טירינג דטרמיניסטיבית  $D$  באותו אופן כמו בהוכחת השקילות במשפט 4.1.

כלומר, בהינתן קלט  $w$ ,  $D$  תסrox את עץ החישוב של  $N$  ו-  $w$  לרוחב ותקבל כל אחד החישובים של  $N$  המסתויים ב-  $q_{\text{acc}}$ .

בhinintן 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

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

נתיחס כאן לשני החסמים הבאים:



1) חסם פולינומיAli הוא חסם מהצורה  $n^c$  עבור  $0 < c$  כלשהו.

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

## 9.4 המחלקה $P$

### הגדרה 9.4 בעיית הכרעה

בעיית הכרעה מוגדרת באופן הבא:

"בhinintן קלט כלשהו, האם הקלט מקיים תנאי מסוים"

### דוגמה 9.9

בhinintן מספר  $n$ , האם  $n$  ראשוני?

כל בעיית הכרעה ניתנת לתאר כשפה שקולה:

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

### משפט 9.3

. שפה  $\equiv$  בעיית הכרעה .

### הגדרה 9.5 אלגוריתם זמן פולינומיAli

אומרים כי אלגוריתם  $A$  מכריעה בעיה בזמן פולינומיAli אם קיימים קבוע  $0 < c$  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 בעיה בזמן פולינומי, אז קיימת מכונת טיורינג דטרמיניסטיבית המכריע את השפה השקולה לבעיה זו בזמן פולינומי.

. מכונת טיורינג ≡ אלגוריתם מכריע .

**הגדרה 9.6 המחלקה  $P$** 

המחלקה  $P$  היא אוסף כל הבעיות (השפות) שקיימים עבורן אלגוריתם (מכונת טיורינג דטרמיניסטיבית) המכריע אותן בזמן פולינומי.

**דוגמה 9.10**

בדוגמה 9.2 הוכחנו כי קיימת מכונת טיורינג דטרמיניסטיבית שמכריעת עת השפה

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

בזמן  $O(n^2)$ . לכן  $L \in P$ .

**דוגמה 9.11**

תהי  $L_{\text{prime}}$  השפה הבאה:

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

הוכיחו:  $L_{\text{prime}} \in P$

**פתרונות:**

כדי להוכיח כי  $L_{\text{prime}} \in P$ , נבנה אלגוריתם שמכריע את  $L_{\text{prime}}$  ולאחר כך נראה כי האלגוריתם רץ בזמן פולינומי.

האלגוריתם עצמו עוזה שימוש של המשפט הבא:

**משפט: למספר פריך יש מחלק הקטן מ- או שווה לשורשו**

יהי  $n$  מספרשלם. אם  $n$  פריך אז קיים שלם  $a \leq \sqrt{n}$  שמחלק את  $n$ .

**הוכחה:**

- נניח בsvilleה כי  $n$  פריך ולא קיים מחלק  $d$  של  $n$  כך ש-  $\sqrt{n} < d \leq n$ .
- $n$  פריך אז קיימים שלמים  $a, b$  כך ש-  $n = ab$ , כאשר  $1 < a, b < n$ .
- נניח כי  $n < a \leq b$ .

• על פי ההנחה שלנו, אם  $a$  מחלק  $n$  אז  $\sqrt{n} > a$ . בנוסף:  $b \geq a > \sqrt{n}$ .

$$n = ab > (\sqrt{n})(\sqrt{n}) = n ,$$

ז"א  $n > n$  וזה סתירה!

על סמך המשפט הזה נבנה אלגוריתם  $PRIME$  המカリע את  $L_{\text{prime}}$  בזמן פולינומיائي. הרעיון של האלגוריתם הוא, בהינתן קלט  $n$ , לבדוק אם קיים מחלק של  $n$  אשר קטן מ-  $\sqrt{n}$ . אם כן אז  $n$  פריק ואם לא אז  $n$  ראשוני.

### בנייה האלגוריתם

: $x = PRIME$  על קלט

**1)** בודק אם  $x = \langle n \rangle$  ו-  $n$  מספרשלם חיובי.

- אם לא  $\Leftarrow$  דוחה.

**2)** אם  $n = 1 \Leftarrow$  דוחה.

**3)** אם  $n = 2 \Leftarrow$  מקבל.

**4)** אחרת לכל  $3 \leq d \leq \lfloor \sqrt{n} \rfloor$

- מחשב  $n \bmod d$

$\Leftarrow n \bmod d = 0$  • מקבל.

**5)** מקבל.

### סיבוכיות זמן

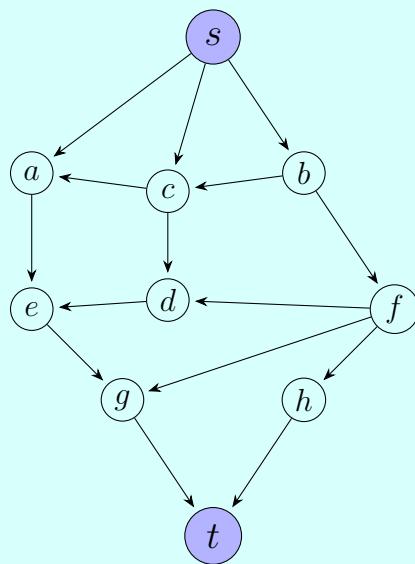
- $PRIME$  מבצע  $O(n)$  איטרציות.
- בכל איטרציה  $PRIME$  מחשב  $n \bmod d$  באמצעות האלגוריתם  $DIVISION$ , אשר הוכחנו בדוגמה 9.8 שהוא רץ בזמן  $O(n^2)$ .
- לכן הסיבוכיות זמן של  $PRIME$  היא  $O(n^3)$ .

לפיכך:

$$L_{\text{prime}} \in P .$$

## 9.5 בעיית PATH

### הגדלה 9.7 בעיית המסלול בגרף מכוון



קלט: גראף 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  $.PATH$

$$:\langle G, s, t \rangle = A$$

**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  $O(|V| \cdot |E|)$  פולינומיAli, במספר הקודקודים  $|V|$ .

אם זה פולינומיAli בגודל הקלט ?

איך נקודד את  $G$  ?

- ניתן כי  $n = |V|$

- 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 בסיס ביניארי.

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

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

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

■ ולכן  $A$  רץ בזמן פולינומיائي בגודל הקלט.

## RELPRIME 9.6 הביעית

### הגדרה 9.8 מספרים זרים (Relatively prime)

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

### הגדרה 9.9 ביעית RELPRIME

קלט: שני מספרים  $x$  ו-  $y$ .

פלט: האם  $x$  ו-  $y$  זרים?

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

אנחנו נוכיח כי ניתן להכrie את  $\text{RELPRIME}$  בזמן פולינומיائي, כלומר נוכיח  $\text{RELPRIME} \in P$  במשפט 9.8 למטה. לפניכן נסביר את האלגוריתם של אוקליזד למציאת ה-  $\gcd$  של שני שלמים, ומכיון זה נוכל לחשב את הסיבוכיות זמן של  $\text{RELPRIME}$ . ראשית נזכיר משפט שלמדו בקורסים קודמים:

### משפט 9.6 המשפט של האלגוריתם של אוקליזד

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 משפטי שימושיים" בדף 110.

האלגוריתם של אוקליזד הוא אלגוריתם, שמקבל קלט שני מספרים  $y, x$  וpollט את  $\gcd(x, y)$ . הוא מתבסס על המשפט 9.6. האלגוריתם עצמו הוא כדלקמן:

: $x, y$  על קלט = EUCLID

(1) כל עוד  $0 \neq y$ :

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

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

(כלומר מחליפים בין  $x$  ו-  $y$ ).

(4) מוחזירים את  $x$ .

לדוגמא:

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

כדי להוכיח כי  $\text{RELPRIME} \in P$  נדרש למשפט עזר הבא:

### משפט 9.7 (משפט עזר)

$$\text{אם } x \bmod y < \frac{x}{2} \text{ או } x > y$$

**הוכחה:** ההוכחה היא לא חלק של הקורס ומופיע בסעיף האחרון "הוכחות של משפטי שימושיים" בדף 111.

### משפט 9.8

$$\text{RELPRIME} \in P .$$

**הוכחה:**

בננה אלגוריתם  $A$  המכريع את  $\text{RELPRIME}$  בזמן פולינומילי.  $\text{RELPRIME}$  היא השפה של הבעיה, שמקבלת כקלט שני מספרים שלמים  $y, x$  ומחזירה תשובה לשאלת, האם  $y, x$  זרים. כלומר:

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

לכן  $A$  משתמש באלגוריתם של אוקלידס ( $\text{EUCLID}(x, y)$ ) כדי לחשב

#### בנייה האלגוריתם $A$ המכريع

"על קלט  $\langle x, y \rangle$ :  $A$  מרים את  $\text{EUCLID}(x, y)$  על  $x$  ו-  $y$ .  $= A$

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

• אחרת  $A$  דוחה."

#### הוכחת הנכונות

הנכונות של  $A$  מנובעת ישיר מהנכונות של האלגוריתם האוקלידס,  $\text{EUCLID}$ .

#### סיבוכיות זמן

נראה כי  $A$  רץ בזמן פולינומילי בגודל הקלט  $\langle x, y \rangle$ . נסמן את אורך הקלט  $n = |\langle x, y \rangle|$ .

• לפי משפט עזר 9.7:  $x \bmod y < \frac{x}{2}$ .

• בכל איטרציה, בשלב (2) המשתנה  $x$  מקבל את הערך החדש  $x \bmod y$ . ניתן לחשב את  $x \bmod y$  בעזרת האלגוריתם החילוק של אוקלידס  $\text{DIVISION}$  שרץ בזמן  $O(n^2)$  (ראו דוגמה 9.8).

• לכן בכל איטרציה הערך החדש של  $x$  קטן ממש מחצי של הערך הקודם של  $x$ .

- לכן אחרי כל איטרציה,  $x$  קטן לפחות חצי.
- בשלב (3),  $A$  מחליף בין  $x$  ו-  $y$ , אז אחרי כל 2 איטרציות, גם  $x$  קטן לפחות חצי וגם  $y$  קטן לפחות חצי.
- $m \stackrel{\Delta}{=} \min(2\lfloor \log_2 x \rfloor, 2\lfloor \log_2 y \rfloor)$  הוא מספר הפעמים המקסימלי שאפשר לבצע שלבים (2) ו- (3) היא  $O(n^2)$ .
- לכן  $m$  הוא חסם עליון של מספר האיטרציות ש-  $EUCLID$  מבצע.
- $n \leq m$  כאשר  $n$  הוא האורך של הקלט  $\langle x, y \rangle$  לכט  $EUCLID$  מבצע  $O(n)$  איטרציות.
- כל איטרציה מבצעת  $DIVISION$  כדי לחשב  $x \bmod y$  אשר רץ בזמן  $O(n^2)$ .
- לכן  $A$  רץ בזמן  $O(n^3)$ . כלומר:

$$RELPRIME \in TIME(n^3).$$

לכן:

$$RELPRIME \in P.$$

■

## 9.7 \*הוכחות של משפטי שימושים

### משפט 9.9 השפט 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 זו היא רק להוסיף הבנה להוכחה של משפט 9.8 לסיבוכיות זמן של  $RELPRIME$  למקרה שהוא לא הוכח שאתם תיבחנו עליה ואפשר לדלג עליה.

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

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

נגידיר  $\bar{d} \stackrel{\Delta}{=} \gcd(x, y)$ . מכיוון ש-  $d$  הוא מחלק משותף של  $x$  ו-  $y$  אז  $d \mid x$  ו-  $d \mid y$ . לכן בזכות משווואה (1\*):

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

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

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

כעת נגדיר  $\bar{d} \stackrel{\Delta}{=} \gcd(y, x \bmod y)$ . מכיוון ש-  $\bar{d}$  הוא מחלק משותף של  $y$  ו-  $x \bmod y$  אז  $\bar{d} \mid y$  ו-  $\bar{d} \mid x \bmod y$ . לכן בזכות משווואה (1\*):

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

ז"א וגם  $\bar{d} \mid y$  אז בהכרח:

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

לסיכום, לפי משוואות (2\*) ו-(3\*):

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

מכיוון ש-  $\gcd(x, y) = \gcd(y, x \bmod y)$ , אז  $d = \bar{d}$  כיון ש-  $d, \bar{d} > 0$ .

### משפט 9.10 (משפט עזר)

$$\text{אם } x \bmod y < \frac{x}{2} \text{ או } x > y$$

**הוכחה:** יש שני מקרים:

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

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

נווכיח את הטענה עבור שני המקרים.

$$\text{מקרה 1: } y \leq \frac{x}{2}$$

לפי משפט החילוק של אוקלידס אם  $x, y$  שלמיםubeiros  $y > x$  אז קיימים  $r = q = \left\lfloor \frac{x}{y} \right\rfloor$  כך  $0 \leq r < y$ .

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

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

$$\text{מקרה 2: } y > \frac{x}{2}$$

לפי משפט החילוק של אוקלידס אם  $x, y$  שלמיםubeiros  $y > x$  אז קיימים  $r = q = \left\lfloor \frac{x}{y} \right\rfloor$  כך  $0 \leq r < y$ .

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

בפרט אם  $q = \left\lfloor \frac{x}{y} \right\rfloor$  אז  $x > y$  או  $x < 2y$ . מכיוון ש-  $q < 2$ . מכיוון ש-  $x > y$  אז  $x > 2y$ . לכן  $q = 1$ .

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

מכאן

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

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

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