

## סיבוכיות

סיבוכיות היא תחום העוסק בבעיות בעלות פתרון יעיל. אלגוריתם יקרא יעיל אם הוא עובד בזמן פולינומי באורך הקלט שלו. הנושא שנעסוק בו היום הוא שאלת P מול NP. נעסוק בגרסת החיפוש ובגרסת ההכרעה של השאלה.

## P מול NP בגרסת החיפוש

בעיית חיפוש ניתנת להתאמה ליחס  $R \subseteq \{0, 1\}^* \times \{0, 1\}^*$  (שקול ל- $R = \{(x, y) \mid (x, y) \in R\}$ ), בהנתן  $x$  צ"ל  $y$  כך  $(x, y) \in R$ . אין אין  $y$  כנדרש, צריך להחזיר  $\perp$  (סימון מיוחד). היחס  $R$  הוא חסוס פולינומיאלית אם קיים פולינום  $p$  כך שעבור כל  $(x, y) \in R$  מתקיים  $|y| \leq p(|x|)$ . מחלקת בעיות החיפוש הניתנות לפתרון בזמן פולינומיאלי תוגדר בשם  $Polinomial)PF$  (Find):

$$PF = \left\{ R \mid \begin{array}{l} \text{Exists polynomial algorithm } A \\ \text{that solves the search problem} \\ \text{that is represented by } R \end{array} \right\}$$

### אלגוריתם A פותר את בעיית החיפוש של R

בהנתן  $x$ , אלגוריתם  $A$  העובד בזמן פולינומיאלי ב- $|x|$  צריך להחזיר  $y$  כך  $(x, y) \in R$  אם  $y$  כזה קיים, ו- $\perp$  אחרת. נסמן  $R(x) = \{y \mid (x, y) \in R\}$ , אזי  $A(x)$  מחזיר  $y \in R(x)$  אם  $R(x) \neq \emptyset$ , אחרת מחזיר  $\perp$ .

### דוגמה

הבעיה של מציאת 2-צביעה חוקית של גרף שייכת ל- $PF$ . למשל הבעיה של מציאת 3-צביעה חוקית של גרף לא שייכת ל- $PF$ . להלן נתאר מחלקה טבעית של בעיות שלא ידועות להיות שייכות ל- $Polinomial)PC - PF$ .

:(Check

$$PC = \left\{ R \mid \begin{array}{l} R \text{ is polinomially bounded relation.} \\ \text{Exists polinomial algorithm that given} \\ \text{a pair } (x, y) \text{ determines if } (x, y) \in R, \\ \text{otherwise returns } \perp \end{array} \right\}$$

## שאלת $P$ מול $NP$ בגרסת החיפוש

שאלת  $P$  מול  $NP$  בגרסת החיפוש מתאימה לשאלה  $?PC \subseteq PF$   
במילים אחרות: האם כאשר ניתן להכריע בזמן יעיל האם פתרון לבעיית חיפוש הוא פתרון חוקי, האם במקרה זה ניתן גם למצוא פתרון בזמן יעיל?

## טענה

$$PF \subsetneq PC$$

## הוכחה

נראה יחס  $R$  שעבורו מתקיים  $R \in PF, R \notin PC$

$$R = \underbrace{\{(x, 1) \mid x \in \{0, 1\}^*\}}_{R_1} \cup \underbrace{\{(x, 0) \mid x \in S\}}_{R_2}$$

כאשר  $S$  תהיה קבוצה שאינה כריעה, כלומר לא קיימת מכונת טיורינג שיכולה להכריע האם  $x \in S$ .  
 $R \in PF$  כי עבור כל  $x \in \{0, 1\}^*$ , "1" הוא פתרון חוקי וניתן למצוא אותו, אבל  $R \notin PC$  כי בהנתן זוג  $(x, 0) \in R$  כדי להכריע אם  $(x, 0) \in R$  צריך להכריע אם  $x \in S$  וזו בעיה לא כריעה.

## בעיית $P$ מול $NP$ בגרסת ההכרעה

בעיות הכרעה ניתנות לתיאור בעזרת קבוצות:

$$S = \{x \mid x \text{ fulfills condition } \pi\}$$

בעיית ההכרעה המתאימה לקבוצה  $S$ : בהינתן  $x$  נחזיר "כן" אם  $x \in S$ , אחרת נחזיר "לא".  
קבוצת הבעיות הניתנות להכרעה בזמן פולינומיאלי היא  $P$ :

$$P = \{S \mid \text{Exists polinomial algorithm that determines } S\}$$

## דוגמה

קבוצת הגרפים ה-2-צביעים שייכת ל- $P$ .

## דוגמה

קבוצת הגרפים ה-3-צביעים לא ידועה להיות שייכת ל- $P$ .

---

קבוצה טבעית של בעיות שלא ידועה להיות שייכת ל- $P$ :

$$NP = \{S \mid S \text{ has a solving system of type } NP\}$$

## מערכת הוכחה מסוג $NP$

זוג  $V$ -מוודא  $P$ -פולינום המקיים:

1. שלמות - טענות נכונות ניתנות להוכחה:

$$\exists_y |y| < P(|x|), V(x, y) \iff x \in S$$

מוודא הנו אלגוריתם פולינומיאלי ב- $|x|$ .

2. נאותות - טענות לא נכונות לא ניתנות להוכחה:

$$\forall_y V(x, y) = 0 \iff x \notin S$$

## השאלה של $P$ מול $NP$ בגרסת ההכרעה

השאלה של  $P$  מול  $NP$  בגרסת ההכרעה היא האם  $P = NP$ ?  
אנחנו יודעים, בשונה מגרסת החיפוש, ש  $P \subseteq NP$ .

---

שאלת  $P$  מול  $NP$  היא למעשה השאלה "האם להוכחות יש כוח או שלמצוא הוכחה זו משימה לא הרבה יותר קשה מלוודא אותה?"

## טענה

$$PC \subseteq PF \iff P = NP$$

## הוכחה

נניח כי  $PC \subseteq PF$  ונראה כי  $P = NP$  (למעשה, נראה  $NP \subseteq P$ , הרי ידוע ש  $P \subseteq NP$ ).  $\implies$

תהי  $S \in NP$  [ונראה כי  $S \in P$ ]. אזי  $\exists y |y| < P(|x|), V(x, y) = 1 \iff x \in S$

נגדיר יחס  $R$  באופן הבא:  $R = \{(x, y) | V(x, y) = 1\}$ . מכיוון ש  $S \in NP$ , פועל בזמן פולינומיאלי ולכן  $R \in PC$ .

כעת נשתמש בהנחה  $PC \subseteq PF$ , ונקבל  $R \in PF$  - ז"א קיים אלגוריתם פולינומיאלי  $A(x)$  שבהנתן  $x$  מחזיר  $y \in R(x)$  אם  $R(x) \neq \emptyset$ , אחרת מחזיר  $\perp$ , ולכן נגדיבר בעזרת  $A(x)$  אלגוריתם פולינומיאלי שמכריע את  $S$ . האלגוריתם יפעל באופן הבא: בהנתן  $x$  יחזיר "כן" אם  $A(x) \neq \perp$ , אחרת יחזיר "לא".

נניח  $P = NP$  ונראה כי  $PC \subseteq PF$ .  $\impliedby$

עבור  $R \in PC$  צ"ל  $R \in PF$ . נגדיר  $S_R = \{x | R(x) \neq \emptyset\}$ . נגדיר כמו כן את הקבוצה הבאה:

$$S'_R = \{(x, y') | \exists y'' (x, y'y'') \in R\}$$

כלומר  $(x, y') \in S'_R$  אם  $x \in S_R$  - כלומר  $(x, y') \in S'_R$  אם  $y'$  הוא תחילית של פתרון ל- $x$ .

$$S'_R \in NP$$

$$\exists y' |y'| < P(|xy'|), V((x, y'), y'') = 1 \iff (x, y') \in S'_R$$

לעשות בזמן פולינומי כי  $R \in PC$ ,  $V((x, y'), y'') = 1 \iff (x, y'y'') \in R$  ואת הבדיקה ש  $(x, y'y'') \in R$  ניתן

כעת נשתמש בהנחה  $NP \subseteq P$  ונקבל ש  $S'_R \in P$ . מטרתנו היא להראות  $R \in PF$ . נראה אלגוריתם פולינומיאלי שבהנתן  $x$  מחזיר  $y \in R(x)$  אם  $R(x) \neq \emptyset$ , אחרת מחזיר  $\perp$ .

1.  $(x, y') \notin S'_R$  מחזיר  $\perp$  (אין פתרון)

2.  $y' \leftarrow y''$

3. כל עוד  $(x, y') \notin R$ , אם  $(x, y'0) \in S'_R$ , אחרת  $y'1 \leftarrow y'$

4. נחזיר את  $y'$

ולכן קיבלנו כי  $R \in PF$ .



למה שעשינו כאן קוראים רדוקציה. רצינו לפתור את בעיית החיפוש  $R$ , ופתרנו אותה ע"י זה שהשתמשנו באלגוריתם שפותר את בעיית ההכרעה  $S'_R$ .

## רדוקציה

### רדוקציה(קוק) מבעיה B לבעיה A

אלגוריתם יעיל הפותר את בעיה B תוך שימוש ב"קופסה שחורה" הפותרת את בעיה A. הראנו בטענה רדוקציה מ  $R$  ל  $S'_R$ . מקובל לתאר רדוקציית קוק כמכונת טיורינג בעלת גישת אורקל:

### הגדרה

נאמר כי למכונת טיורינג M יש גישת אורקל f ונסמן  $M^f$  אם M יכולה להפעיל שאילתות של f בעלות של פועלה בודדת.

### הגדרה - רדוקציית קוק מ B ל A

מכונת אורקל פולינומיאלית  $M^A$  הפותרת את B.

### רדוקציית(קארפ) (Karp)

רדוקציית קארפ היא רדוקציה בין שתי בעיות הכרעה. היא מקרה פרטי של רדוקציית קוק. בהינתן  $S_1, S_2 \in \{0, 1\}^*$ , רדוקציית קארפ מ  $S_1$  ל  $S_2$  היא פונקציה  $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$  כך ש

$$x \in S_1 \iff f(x) \in S_2$$

f צריכה להיות פונקציה חשיבה פולינומיאלית.

### דוגמה

$$R_{\text{Clique}} = \left\{ \langle (G, K), C \rangle \mid \begin{array}{l} C \text{ is a Clique} \\ |C| \geq K \end{array} \right\}$$

$$S_{\text{Clique}} = \{ \langle G, K \rangle \mid \exists C \subseteq G \text{ is Clique, } |C| \geq K \}$$

כך G גרף, K מספר, ו C הוא קליק - כלומר אוסף קודקודים שבין כל שניים יש קשת.

$$R_{\text{IS}} = \left\{ \langle (G, K), IS \rangle \mid \begin{array}{l} IS \text{ is independent set} \\ |IS| \geq K \end{array} \right\}$$

$$S_{\text{IS}} = \{ \langle G, K \rangle \mid \exists IS \subseteq G \text{ IS is independent set, } |IS| \geq K \}$$

IS הוא קבוצה בלתי תלויה - אוסף קודקודים שבין כל שניים אין קשת. רדוקציית קארפ מ  $S_{\text{Clique}}$  ל  $S_{\text{IS}}$ :  $f : \langle G, K \rangle \mapsto \langle \bar{G}, K \rangle$  - כלומר הופכים את G לגרף המשלים. אם בגרף המשלים יש Clique, אז בגרף המקורי יש קבוצה בלתי תלויה.

## רדוקציית Levin - רדוקציה בין שתי בעיות חיפוש

בהנתן  $R_1, R_2 \subseteq \{0, 1\}^* \times \{0, 1\}^*$ , רדוקציית לויין מ  $R_1$  ל  $R_2$  היא זוג פונקציות

$$f : \{0, 1\}^* \rightarrow \{0, 1\}^* \quad g : \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}^*$$

כך שמתקיימים שני הדברים הבאים:

$$R_1(x) \neq \emptyset \iff R_2(f(x)) \neq \emptyset \quad .1$$

$$g(x, y) \in R_1(x) \iff y \in R_2(f(x)) \quad .2$$

## דוגמה - רדוקציית לויין מ $R_{IS}$ ל $R_{Clique}$

[לפתור את  $R_{IS}$  - בהנתן  $\langle G, K \rangle$  צ"ל  $IS$  בגודל לפחות  $K$  ב  $G$ ]

$$f : \langle G, K \rangle \mapsto \langle \bar{G}, K \rangle$$

$$g : (\langle G, K \rangle, C) \mapsto C$$