

רדוקציות

רדוקציה עצמית

$R \subseteq \{0, 1\}^* \times \{0, 1\}^*$ הינו יחס בעל רדוקציה עצמית אם קיימת רדוקציה R ל S_R (בעיית ההכרעה - $S_R = \{x | \exists y (x, y) \in R\}$) כלומר, אפשר לפתור את בעיית החיפוש ע"י רדוקציה לבעיית ההכרעה.

דוגמה בתרגיל: R-SAT הינו יחס בעל רדוקציה עצמית

היום נראה שכל יחס R ש S_R (בעיית ההכרעה המתאימה ל R) היא NP-שלמה הוא בעל רדוקציה עצמית.

נשים: תמיד קיימת רדוקציה מ S_R ל R .

NP-שלמות

$S \subseteq \{0, 1\}^*$, נאמר ש S היא NP-קשה אם עבור כל $S' \in NP$ קיימת רדוקציה פולינומיאלית מ S' ל S .

כלומר, S קשה יותר מכל בעיה ב NP .

$S \subseteq \{0, 1\}^*$ תיקרא NP-שלמה אם S היא NP-קשה וכן $S \in NP$.

נציין NP-שלמות מוגדר יחסית לרדוקציית קארפ, כלומר S תיקרא NP-שלמה אם $S \in NP$ וקיימת רדוקציית קארפ מ $S' \in NP$ ל S .

תזכורת: רדוקציית קארפ מ S' ל S הינה $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ חשיבה פולינומיאלית כך ש $f(x) \in S \iff x \in S'$

נציין, מלכתחילה לא ברור שישנן קבוצות S שהן NP-שלמות.

טענה

קיימות בעיות הכרעה שהן NP-שלמות.

הוכחה

נתבונן ביחס הבא:

$$R_U = \left\{ \left(\langle M, x, 1^t \rangle, y \right) \mid \begin{array}{l} M \text{ is a description of a Turing machine that} \\ \text{accepts the pair } (x, y) \text{ within } t \text{ steps, and } |y| \leq t \end{array} \right\}$$

ראשית נשים לב כי $R_U \in PC$. זאת משום שאפשר להפעיל את מכונת הטיורינג האוניברסלית שתריץ את המכונה M צעדים על הקלט (x, y) , תבדוק אם המכונה מקבלת ותענה בהתאם.

נשים לב שמשקול דומה, $S_{R_U} = \{z | \exists y (z, y) \in R_U\}$ מקיימת $S_{R_U} \in NP$.

נראה ש S_{R_U} היא NP-שלמה. ראינו ש $S_{R_U} \in NP$, ולכן נותר רק להראות שעבור כל $S \in NP$, קיימת רדוקציית קארפ מ S ל S_{R_U} .

תהי $S \in NP$, כלומר קיים מוודא פולינומי V ופולינום P כך ש

$$\exists_y |y| < P(|x|), V(x, y) = 1 \iff x \in S$$

נגדיר את היחס R :

$$R = \left\{ (x, y) \mid \begin{array}{l} V(x, y) = 1 \\ |y| < P(|x|) \end{array} \right\}$$

כלומר $R \in PC$, כלומר קיימת מכונת טיורינג M_R וקיים פולינום P_R כך שמכונת הטיורינג רצה בזמן פולינומי ב $|x|$ וב $|y| \leq P_R(|x|)$, כלומר רצה בזמן $t_R(|x| + P_R(|x|))$, ומכריעה האם $(x, y) \in R$.

כעת נראה רדוקציית קארפ מ S ל S_{R_U} . עבור $x \in \{0, 1\}^*$ נגדיר $f(x) = \langle M_R, x, 1^{t_R(|x| + P_R(|x|))} \rangle$ ונטען כי

$$f(x) \in S_{R_U} \iff x \in S$$

$$\langle M_{R_U}, x, 1^{t_R(|x| + P_R(|x|))} \rangle \in S_{R_U} \iff x \in S$$

כך $|y| < P_R(|x|)$, קיים y $\iff \exists_y |y| < P_R(|x|), (x, y) \in R \iff x \in S$
 ש M_R מקבלת את הקלט (x, y) תוך $t_R(|x| + P_R(|x|))$ צעדים $\iff \langle M_R, x, 1^{t_R(|x| + P_R(|x|))} \rangle \in S_{R_U}$.



משפט (Cook)

SAT היא NP-שלמה

טענה

יחס R כך שבעיית ההכרעה המתאימה לו S_R היא NP-שלמה ניתן לרדוקציה עצמית, כלומר קיימת רדוקציה מ R ל S_R .

הוכחה

נשתמש בטענת העזר הבאה שראינו בשיעור שעבר, כאשר הוכחנו ש

$$NP \subseteq P \implies PC \subseteq PF$$

¹ t_R הוא פולינום של זמן ריצת המכונה לפי אורך הקלט.

טענת עזר

לכל $R \in PC$ קיימת רדוקציה מ S'_R ל $S'_R \in NP$ וכן כאשר

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

$R \in PC$ (כי $S_R \in NP$), ולכן ע"פ טענת העזר לעיל קיימת רדוקציה מ S'_R ל S'_R .
קעת נשתמש בהנחה ש $S_R = \{x | \exists y (x, y) \in R\}$ היא NP-שלמה, ונסיק כי קיימת רדוקציה מ S'_R ל S'_R (כי $S'_R \in NP$). בשל טרנזיטיביות רדוקציה, Reduction from R to S'_R Reduction from S'_R to S'_R גוררים רדוקציה מ S'_R ל S'_R כנדרש.



לעיתים מתקבל הרושם כי כל בעיה בNP היא או NP-שלמה, או קלה, כלומר בP. נראה מיד שזה אינו המצב:

משפט (Lander)

אם $NP \neq P$ אזי קיימות קבוצות NP שאינן NP-שלמות ואינן בP.

הוכחה

רעיון ההוכחה יהיה שימוש בליכסון כפול. נגדיר את הקבוצה הבאה:

$$A = \{x \mid x \in SAT \wedge \hat{M}(1^{|x|}) = 0 \pmod{2}\}$$

יהיו $M_1^D, M_2^D, M_3^D, \dots$ סידור של כל המכונות להכרעה הפועלות בזמן פולינומי. יהיו $M_1^C, M_2^C, M_3^C, \dots$ סידור של כל המכונות המחשבות פונקציות (רדוקציות) בזמן פולינומי.

נגדיר את המכונה \hat{M} לפעול באופן הבא:
 $\hat{M}(1^n)$

1. אם $n = 0$ תחזיר $k = 1$, אחרת תקבע $k = \hat{M}(1)^{n-1}$.

2. אם k זוגי: נסמן $i = \frac{k}{2}$. במקרה זה ננסה לוודא ש A אינה מוכרעת ע"י המכונה M_i^D . עבור כל מחרוזת z באורך $\log n \geq$ נבדוק האם $M_i^D(z)$ מחזירה תשובה נכונה לגבי השייכות של z ל A . אם מצאנו z שבו M_i^D טועה נחזיר $k + 1$ אחרת נחזיר k .

אם k אי זוגי: נסמן $i = \frac{k+1}{2}$. במקרה זה ננסה לוודא שהרדוקציה שמוגדרת ע"י המכונה M_i^C לא מחשב רדוקציית קארפ נכונה מSAT ל A .

עבור כל מחרוזת z באורך $\log n \geq$ נבדוק האם $z \in SAT$ והאם ואם $M_i^C(z) \in A$. אם מצאנו z שבואין התאמה נחזיר $k + 1$, אחרת נחזיר k .

טענה: $A \notin P$ (אם $P \neq NP$)

הוכחה: נניח בשלילה ש $A \in P$. אזי קיימת מ"ט להכרעה פולינומית M_j^D המכריעה את A .

במקרה זה, נסמן $k = 2j$ ונשים לב ש \hat{M} לעולם לא תחזיר את הערך $k + 1$. ז"א שקיים n^* כך שעבור כל $n' \geq n^*$ ועבור כל $|x| = n'$, $\hat{M}(1^{|x|}) = k$ וכן k זוגי. מכאן נסיק ש A זהה ל-SAT - פרט אולי למספר סופי של קלטים - ולכן A כזה היא NP שלמה, ולכן אם $P \neq NP$ ו $A \in NPC$ אזי $A \notin P$.

טענה: $A \notin NPC$ (אם $P \neq NP$)

הוכחה: נניח בשלילה ש $A \in NPC$. אזי קיימת רדוקציה קארפ מ-SAT ל- A . תהי M_j^C המכונה הראשונה המחשבת רדוקציה זו. נסמן $k = 2j - 1$ (אי זוגי), ונשים לב ש \hat{M} לעולם לא תחזיר את הערך $k + 1$. כלומר קיים n^* כך שלכל $n' \geq n^*$ ולכל קלט x באורך $|x| = n'$ מתקיים ש $\hat{M}(1^{|x|}) = k$, ולכן התנאי $\hat{M}(1^{|x|}) = 0 \pmod{2}$ לא מתקיים החל מ- n^* . מכאן נסיק ש A היא קבוצה סופית, ולכן $A \in P$ ולכן אם $P \neq NP$, $A \notin NPC$.

נותר להראות ש $A \in NP$. כדי להראות זאת די להוכיח ש $\hat{M}(1^n)$ רצה בזמן פולינומיאלי ב- n , שכן כדי לוודא ש $x \in A$ די לוודא ש $x \in SAT$. זאת ניתן לעשות בזמן פולינומיאלי ע"י המוודא של SAT, וכן יש לוודא ש $\hat{M}(1^{|x|})$ מחזיר ערך זוגי, זאת ניתן לוודא בזמן פולינומיאלי ע"י הרצת \hat{M} (בהנחה ש \hat{M} עובדת בזמן פולינומיאלי).
כעת נראה כי $\hat{M}(1^n)$ רצה בזמן פולינומיאלי ב- n :

• בשלב 2:

- אם k זוגי: קיימים n קלטים באורך $\log n \geq$, ועבור כל קלט כזה, שנשמנו z , צריך להריץ את $M_i^D(z)$ (עלות פולינומיאלית ב- $\log n$), וכן לבדוק האם $z \in A$ (בעלות אקספוננציאלית באורך של z , ולכן פולינומיאלית ב- n).

- אם k אי-זוגי: באופן דומה צריך לעבור על n קלטים באורך $\log n$. צריך להריץ את $M_i^C(z)$ (עלות לוגריתמית ב- n , פולינומית ב- z) ולבדוק האם $z \in A$ (בעלות אקספוננציאלית ב- z פולינומית ב- n).

ולכן סה"כ עלות שלב 2 פולינומית. שלב 2 מורץ n פעמים, ולכן סה"כ קיבלנו ש $\hat{M}(1^n)$ עובדת בזמן פולינומי.

ולכן קיבלנו:

$$A \notin P \quad A \notin NPC \quad A \in NP$$



הערה 1

התבוננו בסידור של המכונות הפולינומיאליות $M_1^D, M_2^D, M_3^D, \dots$, אבל אוסף המכונות שרצות בזמן פולינומיאלי אינו כריע. לכן למעשה נתבונן בשלשות מהסוג $\{(M, c, d) \mid c, d \in \mathbb{N}\}$, ושלשה כזו תייצג מכונת טיורינג M שמורצת על קלט x , $|x|^c + d$ צעדים, וכל מכונה פולינומיאלית תיוצג ע"י שלשה מתאימה.
כנ"ל לגבי $M_1^C, M_2^C, M_3^C, \dots$.

הערה 2

אנחנו לא יודעים על שפות טבעיות שמקיימות את התנאים האלה. ישנה השערה ש Factory מקיימת את 3 התנאים ש A מקיימת.