

משפט: SAT היא NP-שלמה.

צריך להראות:

1. $SAT \in NP$.

2. SAT היא NP-קשה.

צריך להראות: לכל שפה $L \in NP$ מתקיים $L \leq_p SAT$.

נקבע $L \in NP$. יש מ"ט $M \in P$ (מ"ט דטרמיניסטית שרצה בזמן פולינומי), ו- $x \in L \Leftrightarrow \exists y. M(x, y) = 1, |y| = poly(|x|)$.

אנחנו צריכים להראות פונקציה $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$ כך ש:

1. $f(x) \in SAT \Leftrightarrow x \in L$.

2. $f \in P$.

הרדוקציה:

f ידעת את מ"ט M , עובדת על קלט $x \in \{0, 1\}^*$ ותבנה פסוק SAT, שנקרא לו

$\varphi_{M,x}$.

המשתנים של $\varphi_{M,x}$: $x_{i,j,s}$ לכל $1 \leq i \leq s(n)$ ו- $1 \leq j \leq T(n)$ (אנחנו נפרש: $x_{i,j,s}$ בתור בזמן j במקום i יש תוכן s). לכל i, j מוגדר s יחיד שרק עבורו $x_{i,j,s} = \text{true}$. s כולל גם אם הראש הקורא נמצא על התו i , אם כן, מה המצב של המכונה, ובכל מקרה, מה כתוב בסרט העבודה בתא. בסך הכל, $O(1)$ אפשרויות.

הפסוק: $\left(\bigvee_{s \in S} x_{i,j,s} \right) \wedge \left(\bigwedge_{\substack{s_1, s_2 \\ s_1 \neq s_2}} \neg (x_{i,j,s_1} \wedge x_{i,j,s_2}) \right)$

פסוק כזה לכל (i, j) – פולינומי.

מצב תחילי נכון: $x_{1,0,x_1} \vee x_{2,0,x_2} \vee \dots$.

מצב בסוף מקבל: $\bigvee_{i=1, \dots, s(n)} x_{i,T(n), s_{acc}}$.

מצב רגיל: $\bigvee_{(s_1, s_2, s_3, s_4) \in \text{Valid}} x_{i,j,s_1} \wedge x_{i-1,j-1,s_2} \wedge x_{i,j-1,s_3} \wedge x_{i+1,j-1,s_4}$ לכל i, j .

$\varphi_{M,x}$ של כל זה, זה הפסוק $\wedge$.

טענה א': אם $x \in L$ (כלומר $\exists w. M(x, w) = 1$) אזי $\varphi_{M,x}$ ספיק.

$x_{i,0,s}$ – השמה שמשקפת ראש קורא על התא השמאלי וקלט $x_1, \dots, x_n$ (עבור $1 \leq i \leq n$).

$x_{n+1,0,s}$ – השמה שמשקפת את העד.

עבור $x_{i,j,s}$ ($j > 0$), נבחר את ההשמה שמתאימה לערך של התא ה- i בזמן t בטבלת החישוב של M על הקלט ב- x_1 .

$$x_{i,j,s} = \begin{cases} \text{true} & \begin{array}{l} s \text{ משקף את הערך} \\ \text{של התא ה-} i \\ \text{בזמן } t \text{ בטבלת} \\ \text{החישוב} \end{array} \\ \text{false} & \text{אחרת} \end{cases}$$

בגלל ש- M מקבלת את x, w , אזי הפסוק שאומר שבזמן $t(n)$ עוצרים במצב מקבל – יסתפק, ולכן $\varphi_{M,x}$ ספיק (מצאנו לו השמה מספקת).

טענה ב': אם $\varphi_{M,x}$ ספיק, אזי $x \in L$.

נבחר את w להיות מה שיוצא מהמשתנים $x_{n+1,0}, \dots, x_{n+m,0}$.

טענה: (באינדוקציה) M מקבלת את (x, w) , כלומר $x_{i,j,s}$ מתאים למצב של המכונה M על הקלט ב- x_1 , בזמן j ובתא i .
לכן, M מקבלת את x, w , ולכן $x \in L$.

לכן: $x \in L \Leftrightarrow \exists w. M(x, w) = 1 \Leftrightarrow \varphi_{M,x}$ ספיק $\Leftrightarrow \varphi_{M,x} \in SAT$.
סיבוכיות מקום:

נכתוב אלגוריתם Log שבהינתן M, x מוציא את הפסוק $\varphi_{M,x}$:
לכל i, j , פשוט פולטים את הפסוקים למדפסת.
ביזרון יש רק את i, j שהם בגודל לוגריתמי.

מסקנה: לכל שפה $L \in \text{NP}$, $L \leq_{\text{Log}} \text{SAT}$.

פסוקית: $\ell_1 \vee \dots \vee \ell_k$.

פסוק CNF: $\bigwedge_{\text{פסוקית}} c_i$.

פסוק הוא 3SAT אם כל פסוקית v היא על לכל היותר 3 ליטרלים.

השפה 3SAT

קלט: פסוק 3SAT φ .

פלט: האם הפסוק ספיק?

ברור ש- $3\text{SAT} \in \text{NP}$.

אפשרות א': להראות שכל $L \in \text{NP}$ מקיימת $L \leq_{\text{Log}} 3\text{SAT}$.

אפשרות ב': נראה $\text{SAT} \leq_{\text{Log}} 3\text{SAT}$.

למה זה מספיק? לכל $A \in \text{NP}$: $A \leq_{\text{Log}} \text{SAT} \leq_{\text{Log}} 3\text{SAT}$.

מסקנה: אם $A \in \text{NP}$ ו- B היא NP-קשה, ו- $B \leq A$, אזי גם A היא NP-קשה.

רדוקציה $3\text{SAT} \leq_{\text{Log}} \text{SAT}$ בבית.

Clique

קלט: גרף $G=(V, E)$, ומספר k ($1 \leq k \leq |V|$).

פלט: האם יש ב- G קליקה בגודל k ?

נראה $3\text{SAT} \leq \text{Clique}$.

נבנה רדוקציה. $f: \{0,1\}^* \rightarrow \{0,1\}^*$. צריך: $x \in 3\text{SAT} \Leftrightarrow f(x) \in \text{Clique}$.

בהינתן קלט פסוק 3SAT $c_1 \wedge c_2 \wedge \dots \wedge c_m$ כאשר $c_i = \ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}$. נשים לב שלכל c_i יש רק השמה אחת לא מספקת.

הקודקודים:

לכל פסוקית c_i על b משתנים ($b \in \{1,2,3\}$) נשים $2^b - 1$ קודקודים בגרף,

ונשייך אותם ל- $2^b - 1$ ההשמות המספקות של c_i .

הצלעות:

נחבר 2 קודקודים בקשת, אם שתי תתי-ההשמות עקביות, כלומר לא סותרות.

$k=m$ - מספר הפסוקיות.

נכונות:

טענה א': אם $x \in 3\text{SAT}$ אזי ב- $G=(V, E)$ יש קליקה בגודל m .

$x \in 3\text{SAT}$. לכן, יש לו השמה מספקת $a_1, \dots, a_n$. מבין הקודקודים ששייכים ל- c_i

נבחר את הקודקוד שעקבי עם ההשמה a . קיבלנו m קודקודים (אחד לכל

פסוקית), והם כולם עקביים, ולכן קיבלנו קליקה בגודל m .

טענה ב': אם יש ב- $f(x)$ קליקה בגודל m , אזי x הוא ספיק.

נבחר ההשמה שמרחיבה (איחוד) את m תתי ההשמות של הקליקה בגודל m .

נקבל ההשמה שעקבית עם כל m תתי ההשמות, ולכן מספקת את כל m

הפסוקיות, ולכן x ספיק.

אלגוריתם Log למימוש הרדוקציה:

לכל $i, j=1, \dots, m$:

נכתוב את 7 ההשמות המספקות של i .

נכתוב את 7 ההשמות המספקות של j .

נבדוק מי מהן עקביות, ולפי זה נחליט אילו קשתות להוציא.

יש לנו $O(\log m)$ זיכרון, ופולטים את הגרף למדפסת.

יכול להיות ש- $\text{NP} = \text{Log}$.

מ"ט לא דטרמיניסטי

$$\forall q, \sigma. \delta(q, \sigma) \subseteq Q \times \{L, R\} \times \Sigma$$

למכונה יש סרט קלט, סרט עבודה וסרט פלט (כמו במקרה הדטרמיניסטי).

המכונה מקבלת קלט מסוים אם ורק אם יש מסלול ממצב תחילי למקבל.

סיבוכיות הזיכרון היא הסיבוכיות על המסלול הכי גרוע ממצב תחילי למסיום.

STCON

קלט: גרף מכוון $G=(V, E)$, $s, t \in V$.
 פלט: האם יש דרך מ- s ל- t ?
 בבירור $STCON \in P$ ו- $STCON \in NP$.
 נוכיח ש- $STCON \in NL$:

האלגוריתם:

נתחיל ב- s . כל פעם ננחש שכן לעבור אליו, ונספור את מספר הצעדים שניחשנו.
 נעצור:

- אם הגענו ל- t (ואז נקבל).
- אם הגענו ל- n צעדים (ואז נדחה).
- אם הגענו לקודקוד בלי יציאה (ואז נדחה).

נכונות:

אם $(G, s, t) \in STCON$, אזי יש מסלול בין s ו- t ב- G , ואורכו לכל היותר $n-1$. לכן, אם המכונה תנחש לפי המסלול, היא תקבל.
 אם $(G, s, t) \notin STCON$, אזי כל ניחוש מוביל רק לשכנים שקשורים ל- s , ולכן הוא לא מוביל ל- t , ולכן הוא יסתיים בדחייה (כל ניחוש יסתיים בדחייה).

סיבוכיות זיכרון:

אנחנו רק צריכים לשמור את מספר הצעדים שעשינו, ואת הצומת הנוכחי (שניהם ב- $O(\log n)$).

משפט: $STCON$ היא NL -קשה. כלומר, כל שפה $L \in NL$ מקיימת $L \leq_{\log} STCON$.
רדוקציה:

תהי $L \in NL$, הנפתרת על ידי מ"ט M , ונקבע קלט x .
 נבנה גרף $G_{M,x}=(V, E)$ וגדיר s, t כך ש- $x \in L \Leftrightarrow s$ קשורה ל- t ב- $G_{M,x}$.
 הגרף $G_{M,x}$ יהיה גרף הקונפיגורציות של המכונה M על הקלט x .
 s תהיה הקונפיגורציה התחילית של M .
 t היא הקונפיגורציה המקבלת של M .

נכונות:

$x \in L \Leftrightarrow$ מההגדרה יש מסלול ממצב תחילי למצב מקבל $s \Leftrightarrow$ יהיה קשיר ל- t
 $G_{M,x} \in STCON \Leftrightarrow$ ב- $G_{M,x}$.

סיבוכיות הזיכרון של הרדוקציה:
 ???

מסקנה 1: $NL \subseteq P$, בגלל ש- $STCON \in P$.

אם $A \in NL$, אזי $A \leq_{\log} STCON$ (כי $STCON$ שלמה ב- NL), ו- $STCON \in P$.

טענה 2: $N \log \subseteq DSpace(\log^2 n)$ (משפט Savitch משנת 1970). במילים: כל מה שאפשר לפתור עם מכונה לא דטרמיניסטית בזיכרון לוגריתמי, אפשר לפתור במכונה דטרמיניסטית בזיכרון $O(\log^2 n)$.

נראה כי $STCON \in Space(\log^2 n)$.

זה מספיק כי אם $A \in NL$, אזי $A \leq_L STCON$, ו- $STCON \in DSpace(\log^2 n)$, ולכן $A \in DSpace(\log^2 n)$.

נניח כי $A \leq_L STCON$ עם f , כלומר $x \in A \Leftrightarrow f(x) \in STCON$.
 בהינתן קלט $x \in A$, היינו רוצים לחשב את $f(x)$ (יקר מידי) ולהריץ את M על $f(x)$ (אבל $f(x)$ ארוך מידי). בפועל: מריצים את M כאילו יש לה קלט, וכשהיא מבקשת ביט מ- $f(x)$, מריצים את f , ונותנים ל- M את הביט ש- f פלטה.

ואכן: $Reach(s, t, n) \Leftrightarrow \exists v. Reach\left(s, v, \frac{n}{2}\right) \wedge Reach\left(v, t, \frac{n}{2}\right)$. עוברים על כל v אפשרי לקודקוד אמצע.

אם נסמן את סיבוכיות הזיכרון של $Reach(s, t, k)$ ב- $S(k)$, אזי $S(k) = \underbrace{\log k}_v + S\left(\frac{k}{2}\right)$.

$S(n) = O(\log^2 n)$, $S(1) = O(1)$.