משפט קוק-לוין ('71)

[["מוכנים? אנחנו הולכים להבין למשפט הזה את הצורה."]]

המשפט

SAT היא NP-קשה.

כלומר: לכל שפה $A \in NP$ מתקיים $A \leq_p SAT$.

[בניגוד לפעמים קודמים, בהן ידענו מה שתי השפות ביניהן עשינו רדוקציה, הפעם עלינו לעשות זאת עבור $A \in NP$ כללית.]

משמעות הנתון $A \in NP$ היא שיש אלגוריתם אימות M וקבועים $c_2 \geq c_1 \geq 1$ כך שלכל מחרוזת x:

1. אם $x \in A$ אזי קיים עד w בגודל $\underbrace{|x|}_{\substack{\text{length}\\ \text{of } x}}^{c_1}$ כך ש- M מקבלת את (x,w) (כלומר $M(x,w)$ מחזיר "כן").
2. אם $x \notin A$ אזי לכל עד w, $M(x,w)$ מחזיר "לא".

בנוסף, לכל w בגודל $|x|^{c_1}$ רץ בזמן $|x|^{c_2}$.

הרדוקציה

לכל $x = x_1 x_2 \cdots x_n$ [זו מחרוזת של התווים $x_1, \ldots, x_n$ בזה אחר זה] נבנה פסוק φ (בגודל $poly(x)$) שמקודד את הטענה "ריצה של $M(x,w)$ מחזירה 'כן'."

הסבר בראשי פרקים

φ מורכב מ-4 חלקים:

$$\varphi = \varphi_{input} \wedge \varphi_{witness} \wedge \varphi_{alg} \wedge \varphi_{accept}$$

משמעותיהם:

- φ_{input}: מקודד את הטענה "החלק הראשון של הקלט ל- M הוא "$x_1 \cdots x_n$,"."
- $\varphi_{witness}$: השמה מספקת $\equiv$ בחירה של עד w.
- φ_{alg}: השמה מספקת $\equiv$ ריצה נכונה של $M(x,w)$.
- φ_{accept}: מקודד את הטענה "המכונה הגיעה למצב מקבל."

ביחד: יש השמה מספקת ל- φ
$\Leftrightarrow$ יש השמה ש:

1) מספקת את $\varphi_{witness}$ (דהיינו, בוחרת עד w)
2) מתארת ריצה נכונה של $M(x,w)$ שמחזירה "כן".

$\Leftrightarrow x \in A$.

[ע"מ לדבר על ריצת אלגוריתם, יש לקחת בחשבון את המודל החישובי בו מדובר.
אנו נשתמש במכונת טיורינג – לא נתעמק בעניין זה יותר מדי; רק נסביר מה צריך בשביל שהרדוקציה תעבוד.]

מכונת טיורינג

- מכילה סרט אינסופי: [ציור]
עם ראש קורא/כותב/זז.

- מס' סופי של פקודות מהסוג:
"אם המכונה במצב q,
והראש קורא σ בתא הנוכחי,
אז:
  - נכתוב σ'
  - נעבור למצב q'
  - נזיז את הראש ימינה/שמאלה/נשאיר במקום"

דוגמה: "אם מצב = 20 וגם הראש קרא 0, אזי נכתוב 0, נעבור למצב 60 ונשאיר את הראש במקום.
אם אתם מכירים את שפת התכנות העתיקה BASIC, אז אם נתבונן לדוגמה בתכנית הבאה:

```
10.  For i = 1 To 10
20.       If A(i) = 0 Then GoTo 60
30.  Next i
40.  Print "No 0 found"
50.  End
60.  Print i
70.  End
```

אז הרעיון הוא כמו מה שמתבצע בשורה המסומנת.

הגדרת המשתנים

נגדיר משתני עזר שיתארו ריצה שלמה של $M(x,w)$.

לכל נקודת זמן $t = 0, \ldots, n^{c_2}$ נגדיר משתנים שמתאים:

- המצב בזמן t
- מיקום הראש בזמן t
- מה כתוב בכל תא בסרט (בזמן t) מ-1 ועד n^{c_2}.

[ציור]

[למה אנחנו לא מסתכלים על הערכים בסרט במקום שמימין לתא מס' n^{c_2}?
כיוון שאנו רצים רק n^{c_2} צעדים, ובכל צעד יכולים להזיז את הראש במיקום אחד בלבד (או כלל לא)]]
וכיוון שהוא מתחיל בקצה השמאלי]], לא ייתכן שהאלגוריתם יכתוב ערך במקומות שמימין לתא זה.]

<u>המשתנים</u>:

- $z_{i,\sigma}^{t} = \begin{cases} \mathcal{T}, & \text{אם בזמן } t\text{, במקום ה-}i\text{, בריצה של } M(x,w) \text{ כתוב } \sigma \\ \mathcal{F}, & \text{אחרת} \end{cases}$ → מה כתוב בסרט (לכל תו σ, זמן t ומקום i)
- $h_{i}^{t} = \begin{cases} \mathcal{T}, & \text{אם בזמן } t\text{, הראש נמצא בתא } i \\ \mathcal{F}, & \text{אחרת} \end{cases}$ → מיקום הראש בזמן t
- $s_{q}^{t} = \begin{cases} \mathcal{T}, & \text{אם המצב בזמן } t \text{ הוא } q \\ \mathcal{F}, & \text{אחרת} \end{cases}$ → מצב המכונה בזמן t

<u>טענה</u>: השמה מספקת את $\varphi \Leftrightarrow$ היא "בוחרת את w" ומקיימת את כל הנ"ל.

בחירת העד ($\varphi_{witness}$)

$$\varphi_{witness} = \bigwedge_{i=1}^{n^{c_1}} \left(z_{n+1+i,\sigma_1}^{0} \vee z_{n+1+i,\sigma_2}^{0} \vee \cdots \vee z_{n+1+i,\sigma_{|\Sigma|}}^{0} \right)$$

]] זה מאפשר לנו לבחור כל מילה [להשלים]]]

מצב מקבל (φ_{accept})

$$\varphi_{accept} = \left(s_{q_A}^{n^{c_2}} \right)$$

כאשר q_A הוא המצב המקבל של מכונת הטיורינג.

<u>משמעות הנ"ל</u>: "בזמן סוף האלגוריתם]] כלומר $t = n^{c_2}$]], המצב הוא המצב המקבל."

ריצה נכונה של האלגוריתם (φ_{alg})

φ_{alg} – מאלץ ריצה נכונה של M – לדוגמה מוודע מעבר תקין מזמן t לזמן $t+1$ במיקום הראש.
]] זה למעשה מה שמוודא שהמשתנים מתארים את ריצת המכונה M לפי איך ש- M בנויה.]]

<u>דוגמה</u>:
אם יש לנו את הפקודה:

"אם הראש קורא 2 והמצב = 6
אזי נכתוב 7,
נעבור למצב 3
ונזוז אחד ימינה."

[ציור]

בדוגמה זו, הפסוקית שתתאר את מעבר זה (בכל זמן שהוא, t, ובכל מקום שבו נמצא הראש, i) היא:

$$\bigwedge_{i,t}\begin{pmatrix}\left(\left(z_{i,2}^{t}\wedge s_{6}^{t}\wedge h_{i}^{t}\right)\to z_{i,7}^{t+1}\right)\wedge\\ \left(\left(z_{i,2}^{t}\wedge s_{6}^{t}\wedge h_{i}^{t}\right)\to s_{3}^{t+1}\right)\wedge\\ \left(\left(z_{i,2}^{t}\wedge s_{6}^{t}\wedge h_{i}^{t}\right)\to h_{i+1}^{t+1}\right)\end{pmatrix}$$

[[זוהי פעולת AND בין כל הפסוקיות הרשומות עבור כל ערכי ה- i וה- t האפשריים.]]

ניזכר מלוגיקה בסיסית שהפסוק $A\to B$ שקול לפסוק $\neg A\vee B$ לכל A,B, לכן הפסוקית השלישית בביטוי הנ"ל, למשל, שקול לפסוקית:

$$\left(\neg\left(z_{i,2}^{t}\wedge s_{6}^{t}\wedge h_{i}^{t}\right)\vee h_{i+1}^{t+1}\right)$$

ולפי חוקי דה-מורגן:

$$\left(\neg\left(z_{i,2}^{t}\wedge s_{6}^{t}\wedge h_{i}^{t}\right)\vee h_{i+1}^{t+1}\right)\equiv\left(\neg z_{i,2}^{t}\vee\neg s_{6}^{t}\vee\neg h_{i}^{t}\vee h_{i+1}^{t+1}\right)$$

[שימו לב שכל הפסוקיות הללו הן מצורה בה כל הליטרלים מופיעים עם שלילה פרט לאחד.
למעשה, **כל** הפסוקיות שיצרנו **בכל** הנוסחה כולה הן כאלה **פרט** לאלה שב- $\varphi_{witness}$.
פסוקית מהצורה הזו נקראת "פסוקית קרן" (פסוקיות בהן יש לכל היותר ליטרל חיובי אחד), ומסתבר שבעיית SAT עם נוסחאות שכולן כאלה נקראת $Horn-SAT$, והיא נמצאת ב- P.]

עוד דברים ש- φ_{alg} צריך לקודד:

- "אין יותר מתו אחד באותו תא בו זמנית": $\bigwedge_{\sigma\neq\sigma'}\bigwedge_{i,t}\left(z_{i,\sigma}^{t}\to\neg z_{i,\sigma'}^{t}\right)$

פולינומיות

הטבלה שלנו ([להשלים]) היא בגודל פולינומי: $n^{c_2}\times n^{c_2}=O\left(n^{2c_2}\right)$.