הגדרות

P: תהי L בעיית הכרעה/שפה. נאמר ש$L \in P$ אם קיים אלגוריתם פולינומי דטרמיניסטי שבהינתן קלט x, מחזיר 1 אם"ם $x \in L$.

NP: $L \in NP$ אם קיים אלגוריתם פולינומי V ופולינום $P\left(\cdot\right)$ כך ש:

1. **שלמות:** לכל $x \in L$, קיים y("עד") עם $|y| \leq P\left(|x|\right)$ כך ש$V\left(x,y\right)=1$.
2. **נאותות:** לכל $x \notin L$ ולכל y מתקיים $V\left(x,y\right)=0$.

השאלה הגדולה של מדעי המחשב - האם $P \equiv NP$? האם כל שפה שאפשר לוודא בזמן פולינומי אפשר גם להכריע בזמן פולינומי?

נרצה לדבר על קושי לפתור בעיות. בשביל זה נשתמש בכלי שנקרא רדוקציה:

הגדרה - רדוקציית many-to-one פולינומית(karp)

תהיינה A וB שתי שפות. נאמר שיש רדוקציית many-to-one פולינומית מB לA ונסמן $B \leq_{m}^{p} A$ אם קיימת פונקציה f הניתנת לחישוב בזמן פולינומי כך ש:

$$\forall_{x} x \in B \iff f\left(x\right) \in A$$

במילים אחרות: קיום של אלגוריתם יעיל עבור A גורר אלגוריתם יעיל עבור B.

הגדרה - NP-Hardness

תהי L שפה. נאמר שL היא NP-קשה אם

$$\forall_{L' \in NP} L' \leq_{m}^{p} L$$

כלומר שפה היא NP-קשה היא קשה לפחות כמו כל שפה אחרת בNP. במילים אחרות - אם יצליחו למצוא אלגוריתם פולינומי לשפה NP-קשה כלשהי, אז בעצם יהיה פתרון פולינומי לכל שפה בNP.

הגדרה - NP-Completeness

תהי L שפה. נאמר שL היא NP-שלמה ונסמן $L \in NPC$ אם:

1. $L \in NP$
2. L היא NP-קשה

משפט קוק

$$SAT \in NPC$$

הגדרה - CONP

$$CONP=\left\{L\middle|\overline{L}\in NP\right\}$$

דוגמה

$$\overline{SAT}=\{\varphi|\varphi \text{ is not satisfiable}\}\in CONP$$

שאלה פתוחה

$$NP\stackrel{?}{=}CONP$$

אבחנה

תהי $L\in NPC$. אם $L\notin P$ אזי $P=NP$.

תרגיל

תחת ההנחה ש$P\neq NP$, לכל אחד מהסעיפים הבאים קבעו האם הוא נכון, לא נכון, או שקול לשאלה פתוחה כלשהי.

א. תהי $L\in NPC$ ותהי L' שפה כך ש$L\subseteq L'$, אזי $L'\in NPC$

פתרון: לא נכון - נראה דוגמה נגדית לL,L' כך ש:

א. $L\in NPC$

ב. $L\subseteq L'$

ג. $L'\notin NPC$($L'\in P$)

לכן ניקח:

$$L=SAT\in NPC \qquad L'=\Sigma^*$$

$$L\subseteq L' \qquad L'\in P$$

ב. המחלקה NPC סגורה תחת איחוד. כלומר לכל $L_1,L_2\in NPC$ מתקיים $L_1,L_2\notin NPC$, מתקיים $L_1\cup L_2\in NPC$.

פתרון: לא נכון - נראה דוגמה נגדית ל$L_1, L_2 \in NPC$ כך ש$L_1 \cup L_2 \in P$:

$$L_1 = \{(\varphi, k) | \varphi \in SAT \vee k \text{ is even}\} \in NPC$$

$$L_2 = \{(\varphi, k) | \varphi \in SAT \vee k \text{ is odd}\} \in NPC$$

$$L_1 \cup L_2 = \{(\varphi, k) | \varphi \in SAT \vee k \text{ is even} \vee k \text{ is odd}\} = \{(\varphi, k)\} \in P$$

ג. תזכורת: $\text{Clique} = \{(G, k) | \text{Exists a } k \text{ sized clique in } G\} \in NPC$. נגדיר:

$$\text{Big-Clique} = \left\{ G \middle| \begin{array}{c} \text{Exists a } n-4 \text{ sized clique in } G \\ n = \text{number of nodes in } G \end{array} \right\}$$

האם $\text{Big-Clique} \in NPC$?

פתרון: לא נכון - נראה ש$\text{Big-Clique} \in P$. נבנה אלגוריתם פולינומי שעובר על כל תתי קבוצות הקודקודים בגודל $n-4$ ובודק אם אחת מהן מהווה קליק. מספר תתי הקבוצות בגול $n-4$ הוא:

$$\binom{n}{n-4} = \frac{n!}{4!\,(n-4)!} = \frac{n\,(n-1)\,(n-2)\,(n-3)}{4!} \in O\left(n^4\right)$$