

רשימת משפטים והגדרות בסיבוכיות

רשימה זו מכילה את ההגדרות והמשפטים החשובים שראינו במהלך השיעורים והתרגולים. היא לא מכילה את כל החומר, ולא ניתן להשתמש בכל משפט שנמצא פה כעובדה נתונה במבחן (סביר שאף צריך להכיר הוכחות של רבים מהם). לא מומלץ ללמוד את החומר מכאן, אך כן מומלץ לבדוק שאתם יודעים את כל המשפטים שנמצאים פה, וללכת לשיעור הרלוונטי בסיכומים אם אתם רוצים להכיר את ההוכחה.

שיעור 1:

הגדרה: בעיית הכרעה/החלטה - פונקציה בוליאנית $f: \{0,1\}^* \rightarrow \{0,1\}$

משפט טיורינג: קיימת אלגוריתם u , הנקרא האלגוריתם האוניברסאלי (או המכונה האוניברסאלי), כך שלכל $x, \alpha \in \{0,1\}^*$: $u(x, \alpha) = M_\alpha(x)$, כאשר אם $M_\alpha(x)$ נתקע אז $u(x, \alpha)$ נתקע.

משפט: קיימת פונקציה $UC: \{0,1\}^* \rightarrow \{0,1\}$ שלא ניתנת לחישוב (מטעמי עוצמות, ובאמצעות ליכסון)

הגדרה: סיבוכיות זמן ריצה – עבור פונקציה $T: \mathbb{N} \rightarrow \mathbb{N}$ נגדיר את $DTIME(T(n))$ בתור קבוצת כל השפות שיש אלגוריתם שפותר אותן בזמן $O(T(n))$, כלומר קיים קבוע $c > 0$ שזמן הריצה הוא לכל היותר $c \cdot T(n)$.

הגדרה: $P = \bigcup_{c>0} DTIME(n^c)$

הגדרה: NP – שפה $L \subseteq \{0,1\}^*$ שייכת ל-NP אם קיים אלגוריתם M שרץ בזמן פולינומיאלי ופוליוןם

$$\forall x. x \in L \leftrightarrow \exists w \in \{0,1\}^{p(|x|)}. M(x, w) = 1$$

תרגול 1:

הגדרה: $co-C = \{\bar{L} \mid L \in C\}$

שאלה פתוחה: $?NP = coNP$

שיעור 3:

הגדרה: $EXP = \bigcup_{c>0} DTIME(2^{n^c})$

טענה: $P \subseteq NP \subseteq EXP$

הגדרה חלופית ל-NP: $LENP$ אם קיימת מכונה אי-דטרמיניסטית אשר מכריעה את L .

הגדרה: רדוקצית Karp פולינומיאלית משפה A לשפה B היא פונקציה f שניתנת לחישוב בזמן פולינומיאלי כך שלכל x : $x \in A \Leftrightarrow f(x) \in B$ נסמן $A \leq_p B$.

הגדרה: שפה A היא NP-קשה אם לכל $B \in NP$ מתקיים $B \leq_p A$.

הגדרה: שפה A היא NP-שלמה אם $A \in NP$ וגם A NP-קשה.

תרגול 3:

הגדרה: רדוקציית Cook משפה A לשפה B היא אלגוריתם שמכריע את השפה A תוך שימוש באלג' שמכריע את השפה B (למעשה, בהינתן אורקל של B).

שיעור 4:

משפט קוק-לויין: SAT היא NP-שלמה.

הגדרה: $NEXP = \bigcup_{c>0} NTIME(2^{n^c})$

משפט: $NP \neq P \leftarrow NEXP \neq EXP$

שיעור 5:

הגדרה: Σ_2^P היא מחלקת כל השפות L שיש עבורן אלגוריתם פולינומיאלי M ופולינום p כך ש:

הגדרה: $\Pi_2^P = co\Sigma_2^P$

הגדרה: ההירארכיה הפולינומיאלית $PH = \bigcup_{i>0} \Sigma_i^P$

משפט (קריסת ההירארכיה הפולינומיאלית): $P = NP \rightarrow P = PH$

משפט הירארכיית הזמן: לכל פונקציות f, g כך ש- $g(n) = \omega(f(n)\log f(n))$ מתקיים:

הגדרה: אם $O \subseteq \{0,1\}^*$ שפה כלשהי, אז אלגוריתם עם גישה ל- O הוא אלגוריתם שיש לו פקודה מיוחדת שמאפשרת לו לחשב את $O(x)$ לכל x ביחידת זמן אחת. כך ניתן לסמן את המחלקות P^O ו- NP^O , וכד'.

משפט: קיימת שפה A כך ש- $NP^A \neq P^A$, וקיימת שפה אחרת B כך ש- $NP^B = P^B$.

שיעור 6:

הגדרה: מכונת טיורינג היא השישייה הסדורה: מספר סופי (וקבוע) של מצבים Q , א"ב סופי של הקלט Σ , א"ב סופי של המכונה $\Gamma \subseteq \{ _ \} \cup \Sigma$, פונקצית מעברים $Q \times R \rightarrow Q \times R \times \{L, R\}$, δ : מצב התחלתי q_0 , מצב מקבל q_{accept} ומצב דוחה q_{reject} .

הגדרה: $SPACE[t(n)] = \{L \mid L \text{ is decided by a } O(t(n))\text{-space TM}\}$

הגדרה: $NSPACE[t(n)] = \{L \mid L \text{ is decided by a } O(t(n))\text{-space non-deterministic TM}\}$

הגדרת מחלקות:

$$L = SPACE[\log(n)]$$

$$NL = NPSACE[\log(n)]$$

$$PSPACE = \bigcup_{k \geq 1} SPACE(n^k)$$

הגדרה: רדוקציית log-space היא פונקציה f ניתנת לחישוב בסיבוכיות מקום לוגריתמית, כך ש:
 $\forall x \in A \leftrightarrow f(x) \in B$. נסמן $A \leq_L B$.

משפט: המחלקות $L, NL, P, NP, PSPACE, EXP$ סגורות תחת רדוקציות log-space.

משפט: $CONN$ היא NL -שלמה.

שיעור 8:

משפט Savitch: $\forall S(n) \geq \log(n) : NSPACE[S(n)] \subseteq SPACE[S(n)^2]$

מסקנה: $PSPACE = NPSPACE$

משפט Immerman: $Non-CONN \in NL$

מסקנה: $!NL = coNL$

שיעור 9:

הגדרה: נוסחא מכומתת לחלוטין היא נוסחא מהצורה $\forall x_1 \forall x_2 \dots \forall x_n. \varphi(x_1, \dots, x_n)$ כאשר $\varphi(x_1, \dots, x_n)$ היא נוסחה ללא כמתים.

משפט: $TQBF \in PSPACE\text{-Complete}$

הגדרה: בהינתן בעיית אופטימיזציה, אלגוריתם A מספק c -קירוב אם לכל קלט x , ערך הפתרון A -ש- x מספק מקיים: $A(x) \geq \frac{OPT(x)}{c}$ או $A(x) \leq c \cdot OPT(x)$ (תלוי אם זו בעיית מינימיזציה או מקסימיזציה)

משפט: יש אלגוריתם פולי' לפתרון Linear Programming בשם Ellipsoid.

שיעור 10:

הגדרה: נאמר שאלגוריתם פותר את בעיית $gap-A[\alpha, \beta]$ (נניח ש- A בעיית מינימום) אם עבור כל קלט x עבורו $OPT(x) < \alpha$ הוא מחזיר 1 ועבור כל קלט y עבורו $OPT(y) > \beta$ הוא מחזיר 0. לא איכפת לנו מה הוא מחזיר עבור ערכים בין α ל- β (אם זאת בעיית מקסימיזציה אז צריך להחליף בין 0 ל-1).

משפט: לקרב את TSP זה NP-קשה מכל יחס קירוב.

תרגול 10:

משפט: אם $gap-A[\alpha, \beta]$ היא NP-קשה, אז זה NP-קשה לקרב את A מסדר $\frac{\beta}{\alpha} - \epsilon$ לכל $\epsilon > 0$.

משפט ה-PCP: בעיית $gap-3SAT[\frac{7}{8} + \epsilon, 1]$ היא NP-קשה (לכל $\epsilon > 0$)

שיעור 11:

הגדרה: גרף אילוצים הוא שלישייה: $G=(V,E)$, קבוצת "צבעים" לצביעת הקודקודים Σ , ופונקציית אילוצים מהקשתות אל זוגות הצבעים $\phi: E \rightarrow P[\Sigma^2]$.

טענה: $\text{gap-CSG}_V[7/8+\epsilon, 1]$ היא NP-קשה.

משפט (אמפליפיקציה): $\text{gap-kCSG}_V[\delta, 1] \leq_L \text{gap-k}^l\text{CSG}_V[\delta^l, 1]$

טענה: $\text{gap-kCSG}_V[\delta, 1] \leq_L \text{gap-IS}[\delta/k, 1/k]$

מסקנה: קשה לקרב את IS מכל סדר קבוע.

שיעור 12:

הגדרה: qCSG_Δ זה גרף אילוצים כאשר פונקציית האילוצים ϕ נקבעת באופן יחיד על ידי ההפרש המודולרי בין הצבעים.

משפט: $\text{gap-kCSG}_V[\delta, 1] \leq_L \text{gap-(nk)}^5\text{CSG}_\Delta[\delta, 1]$

מסקנה: $\text{gap-}\chi[q, q/\delta]$ היא NP-קשה לכל $\delta > 0$.

שיעור 13:

הגדרה: מכונת טיורינג רנדומית (הסתברותית) היא מכונת טיורינג עם סרט נוסף – סרט רנדומי. נגדיר את $\Pr_r[M(x,r)]$ בתור הסיכוי ש-M מקבלת את x אם בוחרים את r בהתפלגות אחידה על $\{0,1\}^n$.

הגדרה: שפה L שייכת למחלקה BPP אם קיימת מ"ט רנדומית פולינומיאלית אשר מקיימת:

טענה (אמפליפיקציה): אם $L \in \text{BPP}$ אז קיימת מ"ט הסתברותית פולי' M' ופולינום $p(n)$ כך ש-

משפט: $\text{BPP} \subseteq \Sigma_2^P$

משפט ההתפלגות הסטציונרית: אם $G=(V,E)$ גרף קשיר, אז לכל $i \in V$:

טענה: $\text{CONN} \in \text{BPP}$