סיכום הרצאות בסיבוכיות

(עידן אטיאס)

סמסטר ב' 2016, מרצה: פרופ' אמיר שפילקה

הרצאה 1:

- **בעיית חיפוש:** יהי $R \subseteq \{0,1\}^* \times \{0,1\}^*$ יחס. נסמן $R(x) = \{y \in \{0,1\}^* | (x,y) \in R\}$. $f : \{0,1\}^* \to \{0,1\}^* \cup \{\perp\}$ פותרת את בעית החיפוש עבור R אם לכל $x \in \{0,1\}^*$ מתקיים $f(x) \in R(x)$. אם $R(x) = \emptyset$ אז $f(x) = \perp$.

- **בעיית הכרעה:** תהי $S \subset \{0,1\}^*$, בהינתן x רוצים לדעת האם $x \in S$. פ' $f : \{0,1\}^* \to \{0,1\}$ פותרת את בעיית ההכרעה עבור S אם מתקיים $f(x) = 1$ $x \in S \Longleftrightarrow$.

- **מודל החישוב העיקרי:** מכונת טיורינג. מ"ט עם סרט אחד הינה : $(Q, \Gamma, \Sigma, \delta, q_0, q_a, q_r)$.

- מוסכמה: אם רוצים לחשב פונקציה, נאמר שמה שכתוב על הסרט מימין לראש בסוף החישוב זו תוצאת החישוב. וריאנט: להוסיף סרט פלט $write - only$.

- הערה: ראינו כי יש שפות לא כריעות (נניח $Halt$). הוכחה: יש $\aleph_0$ מ"ט , מספר השפות הוא $\aleph = 2^{\aleph_0}$. לכן רוב השפות לא כריעות.

- הערה: מ"ט הינה מודל חישוב יוניפורמי־ אלגוריתם העובד לכל אורך קלט.

- **הגדרה: מודל חישוב לא יוניפורמי:** תהי $A \in \{0,1\}^*$, נסמן $A_n = A \cap \{0,1\}^n$. נגדיר $f_n : \{0,1\}^n \to \{0,1\}$ ע"י: $f(x) = 1 \Longleftrightarrow x \in A_n$ (חישוב f_n יכול להיעשות ע"י CNF, DNF ונקבל מעגל בוליאני).

- **הגדרה: מעגל בוליאני** במשתנים $x_1, .., x_n$ הינו גרף מכוון חסר מעגלים. כל שער הוא: אם דרגת כניסה=0 אז זהו שער קלט ומסומן ע"י אחד המשתנים. אחרת: שער לוגי $\vee, \wedge$ (דרגת כניסה 2), $\neg$(דרגת כניסה 1). שער עם דרגת יציאה 0 נקרא פלט.

- חישוב: בהינתן $\bar{a} \in \{0,1\}^n$ שער x_i יחשב את a_i. כל שער לוגי מחשב את תוצאת החישוב של בניו. גודל מעגל = מס' שערים+מס' קשתות.

- נוסחה בוליאנית הינה מעגל בוליאני עם דרגת יציאה ≤ 1 בכל שער (גרף החישוב הוא עץ).

- **הגדרה:** משפחת מעגלים $\{C_n\}_{n=1}^{\infty}$ מכריעה שפה A אם לכל n, C_n(מעגל עם n משתנים) מחשב את A_n. כלומר עבור $x \in \{0,1\}^n$: $C_n(x) = 1 \Longleftrightarrow x \in A_n$.

- עובדה: לכל שפה יש משפחת מעגלים המחשבת אותה (כלומר במודל זה ניתן לחשב שפות לא כריעות).

- **משפט:** לכל $f:\{0,1\}^n \to \{0,1\}$ יש מעגל בגודל $O(n2^n)$ המחשב אותה.
- **משפט:** רוב הפונקציות הבוליאניות על n משתנים צריכות מעגל באורך $\Omega(\frac{2^n}{n})$.

__הרצאה 2:__

- עובדה: ניתן לסמלץ מ"ט עם k סרטים ע"י מ"ט עם סרט אחד. ניפול זמן הריצה ריבועי (הכרחי).
- **הגדרה:** מ"ט אוניברסלית U: מקבלת קלט $(<M>,x)$ ומריצה את M על x.
- עובדה: אם זמן הריצה של M על x הוא $t(n)$ אז U יכולה לסמלץ זאת בזמן $O(tlog(t))$.
- עובדה: לכל פ' t שהיא $t.c$, $n \le t(n)$, U יכולה לסמלץ את $t(|x|)$ הצעדים הראשונים של M על x בזמן ריצה $O(tlog(t))$.
- **משפט היררכיית הזמן:** לכל פונקציה t שהיא $t.c$ ולכל פונקציה T המקיימת $tlog(t) = o(T)$ מתקיים: $DTIME(t(n)) \subsetneq DTIME(T(n))$ (בד"כ כשרוצים להראות ששפה לא ניתנת לחישוב- ע"י לכסון).
- סימון: $DTIME(t(n))$: כל השפות $L \in \{0,1\}^*$ הניתנות לחישוב ע"י מ"ט דטרמינסטית בזמן $\le t(|x|)$ לכל קלט x.
- **הגדרה: מ"ט עם אורקל:** מ"ט עם תוספת של 3 מצבים מיוחדים: Q_{yes}, Q_{no}, Q_{query}. בנוסף ישנו סרט מיוחד עבור שאילתות לאורקל (שפה Bכלשהי). M רצה כמ"ט רגילה, כאשר מגיעה ל-Q_{query} תוך צעד חישוב אחד תעבור ל-Q_{yes} אם המחרוזת בסרט השאילתא שייכת ל-B או שתעבור ל-Q_{no} אם המחרוזת בסרט השאילתא לא שייכת ל-B.
- סימון: $DTIME(t(n))^B$ = אוסף השפות הניתנות לחישוב ע"י מ"ט עם אורקל לשפה B הרצה בזמן $t(|x|)$ לכל x.
- הערה: משפט היררכיית הזמן עובד גם למ"ט עם אורקל.
- הערה: יש שפות A,B כך ש- $P^A = NP^A$, $P^B \subsetneq NP^B$ (לא עוזר להכריע $P \stackrel{?}{=} NP$).
- **הגדרה: מ"ט מוגבלת זיכרון:** 3 סרטים- קלט($read\ only$), עבודה ($read\ write$), פלט ($write\ only$). נאמר שמכונה מכריעה בסיבוכיות זיכרון $s(n)$ אם משתמשת בלכל היותר $S(|x|)$ תאים בסרט עבודה לכל x.
- $DSPACE(s(n))$ = אוסף השפות שניתן להכריע ע"י מ"ט דטרמיניסטית המשתמשת ב- $s(n)$ תאי זיכרון.
- דוגמא: כפל מטריצות בוליאניות (משמש לחישוב קיום מסלולים בגרף)- ניתן לחשב בזיכרון $O(log(n))$.
- עובדה: $regular\ languages = DSPACE(0) = DSPACE(o(loglog(n)))$.
- $LogSpace = \mathbf{L} = \cup_{c=1}^{\infty} DSPACE(clog(n))$, $\mathbf{PSPACE} = \cup_{c=1}^{\infty} DSPACE(n^c)$

- **משפט:** $PSPACE \subseteq P \subseteq L$ ובנוסף $L \subsetneq PSPACE$ (לא יודעים להוכיח יותר מזה).

- **משפט:** אם $log(n) \leq s(n)$ אז $DSPACE(s(n)) \subseteq DTIME2^{O(s(n))}$ (מכך נובע $L \subseteq P$. הוכחה בעזרת גרף הקונפיגורציות). הערה: התנאי הכרחי, דוג'־ פונקציה שמחשבת את הביט האחרון ניתנת לחישוב בזיכרון קבוע אבל לא יכולה להיות סאב לינארית כי חייבת לעבור על כל הקלט)

- הערה: מ"ט מוגבלת זיכרון עם אורקל תהיה עם 4 סרטים כך שסרט העבודה וסרט האורקל נספרים בתור זיכרון.

<u>הרצאה 3:</u>

- סימון: $\mathbf{EXP} = \cup_{c=1}^{\infty} DTIME(2^{n^c})$.

- **משפט:** הרכבת פונקציות בזיכרון נמוך: פתרון נאיבי־ נחשב $f_1(x)$ ואז $f_2(f_1(x))$. זמן: $t_1(|x|) + t_2(|f_1(x)|)$. זיכרון: $s_1(x) + |f_1(x)| + s_2(|f_1(x)|)$. פתרון חסכוני בזיכרון־ נריץ את f_2 כך שכל פעם שמ"ט תרצה לקרוא סימבול מהקלט נריץ את f_1 ונחשב אותו. זיכרון: $m = s_1(x) + log(|f_1(x)|) + s_2(|f_1(x)|)$. זמן: 2^m או $t_1(|x|) \cdot t_2(|f_1(x)|)$.

- **מסקנה:** חישוב A^k: נאיבי־ זמן וזיכרון פולי'. חסכוני־ זיכרון $O(log(n) \cdot log(k))$, זמן: $n^{O(log(k))}$.

- יהי G גרף, A מטריצת שכינויות.

$$(A^k)_{i,j} = \begin{cases} 1 & there\ exists\ directed\ path\ of\ len\ k\ between\ i,j \\ 0 & else \end{cases}$$

נגדיר $A' = A \vee I$ (הוספת לולאות עצמיות) אזי:

$$(A'^k)_{i,j} = \begin{cases} 1 & there\ exists\ directed\ path\ of\ len\ at\ most\ k\ between\ i,j \\ 0 & else \end{cases}$$

- הראינו רדוקציה L(הוצאת A') בין $STCON$ל־$\{< A, s, t > |(A^n)_{s,t} = 1\}$.

- **מסקנה:** בהינתן גרף מכוון עם n קודקודים אפשר לבדוק האם קיים מסלול מכוון בין 2 קודקודים: בזמן וזיכרון פולי' או בזיכרון $O(log^2(n))$ וזמן $n^{O(log(n))}$.

- **הגדרה:**
$STCON = \{< G, s, t > |\text{G is a directed graph,there exists directed path between i,j}\}$.

- **משפט** $SAVITCH$: $STCON \in DSPACE(O(log^2(n))$.

- **רדוקציות קארפ:** $\varphi : \{0,1\}^* \rightarrow \{0,1\}^*$ רדוקציה משפה A ל־B אם מתקיים $x \in A \iff \varphi(x) \in B$. ($\varphi(A) \subseteq B$, $\varphi(A^c) \subseteq B^c$). רדוקציות יעילות־ זמן פולינומיאלי ($\leq_P$) , זיכרון לוגריתמי ($\leq_L$).

- הערה: רדוקציה ב־L זו מכונה שמקבלת x בסרט הקלט, משתמשת בזיכרון לוגריתמי על סרט העבודה ומדפיסה את הנוסחה המתאימה לסרט הפלט.

- **טענה:** אם $A \leq_P B \leq_P C$ אזי $A \leq_P C$ (אותה טענה נכונה גם עבור $\leq_L$).

- **הגדרה:** תהי C מחלקה של שפות ותהי A שפה, נאמר כי A קשה ל־C תחת רדוקציות L (P) אם לכל $B \in C$ יש רדוקציה $B \leq_L A$. נאמר ש־A שלמה ל־C תחת רדוקציות L (P) אם $A \in C$ וגם קשה ל־C תחת רדוקציות L (P).

- הערה: כל שפה ב־P היא שלמה ל־P תחת רדוקציות P.

- **הגדרה:**
$CVAL = \{< C, \bar{a} > |\text{C is a boolean circuit with n inputs, } a \in \{0,1\}^n, \text{C}(\bar{a}) = 1\}$.

- **משפט:** $CVAL$ שלמה ל־P תחת רדוקציות L (שייכות ל־P ע"י בדיקת ערך שערי המעגל מרמה 0 ועד רמת הפלט. קשה ל־P: לוקחים את טבלת החישוב של מ"ט הפולי' ומייצרים ממנה מעגל וקלט מתאימים).

- **הגדרה:** NC^k: אוסף השפות $A \subseteq \{0,1\}^*$ כך שיש משפחה C_n של מעגלים המחשבת את A (C_n מחשב את $A_n = A \cap \{0,1\}^n$), גודל המעגלים פולי' ועומק $O(log^k(n))$.

- **הגדרה:** $uniform - NC^k$: מחלקת השפות ב־NC^k כך שיש מ"ט ב־L שעל קלט 1^n מוציאה את C_n.

- **הגדרה:** $NC = \cup_k NC^k$, $uniform - NC = \cup_k uniform - NC^k$ (שפה ניתנת לחישוב מקבילי יעיל אם היא ב־$uniform - NC$).

- **הגדרה:** QBF: פסוקים (כל המשתנים מכומתים) מעל הקשרים $\neg, \vee, \wedge$ מהצורה: $\exists x_1 \forall x_2 \exists x_3 \cdots Q_n x_n \psi(x_1, .., x_n)$. $TQBF$ שפת כל הפסוקים ב־QBF שערך האמת שלהן הוא $true$.

- הערה: הסדר של $\exists, \forall$ לא משנה כי ניתן להוסיף משתני דמה.

- **משפט:** $TQBF$ שלמה ל־$PSPACE$ תחת רדוקציות L (שייכות ל־$PSPACE$: טריק מוכר: מגדירים $\varphi = \varphi_0 \vee \varphi_1$, מחשבים φ_0, אם קיבלנו 1 נחזיר 1 אחרת נחשב φ_1 וכן הלאה. בעצם מוצאים השמה ל־ψ. קשה ל־$PSPACE$: בעזרת גרף הקונפיגורציות ופ' $Reach(u, v, k)$ מתארים רדוקציה שמוציאה נוסחה שערך האמת שלה הוא $true$ אמ"מ יש מסלול מקונפי' התחלתית לקונפי' המקבלת [ניתן להניח בה"כ שהיא יחידה]).

- **הגדרה:** מ"ט לא דטרמינסטית: מ"ט עם ההבדל הבא: δ אינה פ' אלא יחס. $\delta : \{Q, \Gamma\} \to \mathbb{P}(Q \times \Gamma \times \{\to, \leftarrow, -\})$. מ"ט א"ד מקבלת קלט x אם יש מסלול חישוב המקבל את x. זמן ריצה: נאמר כי זמן הריצה של M על x הוא $t(n)$ אם כל מסלול חישוב של M על x נמשך לכל היותר $t(n)$ צעדים. סימון: $NTIME(t(n))$= כל השפות שניתן להכריע ע"י מ"ט א"ד הרצה על קלט x לכל היותר $t(|x|)$ זמן.

- **NP**: ניתן 2 הגדרות שקולות. **הגדרה 1:** אוסף השפות עבורן יש מוודא יעיל. כלומר, עבור בעיית הכרעה $S \subseteq \{0,1\}^*$ נאמר כי יש מוודא יעיל (זמן ריצה פולי') V, אם קיים c כך שלכל $x \in S$ קיים y כך ש־ $|y| \leq |x|^c$ (הוכחה קצרה) ומתקיים $V(x, y) = 1$. אם $x \notin S$ אז לכל $|y| \leq |x|^c$ מתקיים $V(x, y) = 0$.
הגדרה 2: $NP = \cup_c NTIME(n^c)$.

- **הגדרה:** $CSAT = \{C|\text{C is a boolean circuit,there exists input that satisfies it}\}$.

- **משפט:** $CSAT$ היא NP שלמה תחת רדוקציות L (לוקחים מ"ט עבור $A \in NP$, בה"כ מ"ט כותבת באופן לא דטרמיניסטי עד y (ייצור הקלט) ואז מריצה מוודא דטרמינסטי פולי'. כעת משתמשים ברדוקציה ל־$CVAL$).

- **הגדרה:** $SAT = \{\varphi|\varphi \textbf{ is a satisfiable CNF formula}\}$. $3-SAT$: אותו דבר כך שבכל פסוקית יש לכל היותר 3 ליטרלים.

- **משפט:** $3-SAT$ היא NP שלמה תחת רדוקציות L (מראים רדוקציה מ־$CSAT$).

__הרצאה 5:__

- בעיות NP שלמות שראינו במודלים: $Clique, Independent\ Set,$ $Hamiltonian\ Cycle, 3-Color, Subset\ Sum, Integer\ Programming$

- **הגדרה:** $Max\ 2-SAT$: קלט: נוסחת CNF כך שבכל פסוקית 2 ליטרלים לכל היותר ומספר t. קלט בשפה אם יש השמה המספקת לפחות t פסוקיות.

- **טענה:** $Max\ 2-SAT$ היא NP שלמה תחת רדוקציות L (רדוקציה מ־$3-SAT$).

- **הגדרה:** מ"ט א"ד מוגבלת זיכרון: δ אינה פ' אלא יחס. ישנם 3 סרטים (קלט,עבודה,פלט). סיבוכיות הזיכרון על קלט x: מס' מקסימלי של תאים בסרט העבודה ששימשו אותנו באיזשהו מסלול חישוב על x. זמן ריצה על קלט x הוא זמן הריצה המקסימלי באיזשהו מסלול חישוב.

- הערה: אין 2 הגדרות שקולות במקרה זה. בהגדרה עם עד צריך לרשום אותו ובמקרה זה אולי אין מספיק זיכרון. עם זאת, נראה וריאציה על מוודא: מוודא־NL.

- $NSPACE(s(n))$= כל השפות עבורן יש מ"ט א"ד מוגבלת זיכרון המכריעה בזיכרון $s(|x|)$ לכל היותר על כל קלט x.

- $\mathbf{NL} = \cup_c NSPACE(clog(n))$. $\mathbf{NPSPACE} = \cup_c NSPACE(n^c)$.

- **משפט:** $STCON$ שלמה ל־NL תחת רדוקציות L (שייכות ל־NL: מנחשים מסלול מ־s ל־t, בכל שלב מנחשים קודקוד חדש ובודקים האם הוא שכן של הקודקוד שבזיכרון וכן הלאה. מחזיקים מונה שיחשב את אורך המסלול. קשה ל־NL: בעזרת גרף הקונפיגורציות. בודקים קיום מסלול בין קונפי' התחלתית לקונפי' המקבלת [ניתן להניח שהיא יחידה]).

- **מסקנות:** $NL \subseteq P$ (כי $STCON \in P$). משפט $SAVITCH$:
$NL \subseteq DSPACE(O(log^2(n)))$ (כי $STCON \in DSPACE(O(log^2(n)))$).

- **משפט** $SAVITCH$ הכללי: לכל $log(n) \le s(n)$ מתקיים $NSPACE(s(n)) \subseteq DSPACE(s(n)^2)$.

- **מסקנה:** $NPSPACE = PSPACE$ (כיוון אחד ברור וכיוון שני נובע מהמשפט הקודם).

- תמונת מצב עד כה:

$$L \subseteq \overset{(STCON)}{NL} \subseteq \overset{(CVAL)}{P} \subseteq \overset{(3-SAT,SAT,CSAT)}{NP} \subseteq \overset{(TQBF)}{PSPACE} = NPSPACE \subseteq EXP$$

$$NL \subseteq DSPACE(O(log^2(n))$$

$$P \subsetneq EXP\ ,\ L \subsetneq PSPACE\ (Hierarchy\ Theorems)$$

- **הגדרה:** השפה המשלימה לשפה A הינה $\bar{A} = \{x | x \notin A\}$. נשים לב ש- $\bar{\bar{A}} = A$.
- **הגדרה:** C מחלקה של שפות. $co - C = \{\bar{A} | A \in C\}$.
- **טענות:** אם $A \subseteq B$ (שפות) אז $\bar{B} \subseteq \bar{A}$. אם $C_1 \subseteq C_2$ (מחלקות) אז $co - C_1 \subseteq co - C_2$.
- **טענה:** $co - P = P$ (נשים לב שההוכחה שמראה זאת לא עובדת על מ"ט א"ד. עם זאת, רעיון ההוכחה כן עובד על מ"ט א"ד המחשבות באפס שגיאה).
- **טענה:** אם שפה A שלמה למחלקה C (תחת $\leq_L$, $\leq_P$) אז $\bar{A}$ שלמה ל-$co - C$ (תחת $\leq_L$, $\leq_P$) [אותה הרדוקציה עובדת].
- אם $A \in NP \cap co - NP$ אזי אם $x \in A$ אז יש לכך הוכחה קצרה וגם אם $x \notin A \iff x \in \bar{A}$ אז יש לכך הוכחה קצרה.
- $P \subseteq NP \cap co - NP$.
- **הגדרה:** השפה $PRIMES$: קלט: מחרוזת בינארית באורך n. x בשפה אם הוא ייצוג בינארי של מספר ראשוני.
- **טענה:** $PRIMES \in NP \cap co - NP$. יותר מכך, $PRIMES \in P$. נשים לב שישנו אלגוריתם הסתברותי יעיל ושימושי (מילר רבין) המראה כי $PRIMES \in co - RP$. שני האלגוריתמים מתבססים על משפה פרמה הקטן.
- **הגדרה:** השפה $Integer\ Factor$: קלט: 2 מחרוזות בינאריות באורך n. (N, M) בשפה אם ל-N יש גורם $1 \neq$ הקטן או שווה ל-M.
- **טענה:** $Integer\ Factor \in NP \cap co - NP$.
- **הגדרה:** מ"ט א"ד מכריעה קלט x (או מחשבת פ' $f(x)$) <u>באפס שגיאה</u> אם לכל קלט x בשפה, בכל ריצה של המכונה היא מקבלת או מחזירה "אינני יודעת" וישנו לפחות מסלול חישוב אחד המקבל. אם x איננו בשפה אז ישנו לפחות מסלול אחד דוחה ובמסלולים שלא נדחו מוחזר "אינני יודעת".
- **טענה:** מחלקת השפות המוכרעות ע"י מ"ט א"ד באפס שגיאה הרצה בזמן פולינומיאלי היא $NP \cap co - NP$.
- הערה: ראינו ש-$STCON$ שלמה ל-NL ולכן $STCON^c$ שלמה ל-$co - NL$.

- **משפט:** $(Immerman - Szelepcseny)$ $NL = co - NL$ ובאופן כללי לכל פ' $log(n) \le s(n)$ מתקיים: $NSPACE(s(n)) = co-NSPACE(s(n))$ (משתמשים בהערה הקודמת ומראים אלגוריתם המחשב באפס שגיאה את $STCON$ וגם [ע"י היפוך מצבים מקבלים ודוחים] את $STCON^c$. נשים לב שצריך להתמודד עם הסוגיה שאין מסלול בין שני קודקודים ולכן צריך לבדוק את כל המסלולים האפשריים).

- הערה: צורת הסתכלות נוספת על NP: שפה $A \in NP$ אם קיים יחס $R(x,y)$ ב־P ($|y| \le |x|^c$ עבור c קבוע כלשהו) כך ש־$x \in A$ אמ"מ $\exists y R(x,y)$ נכונה. עבור $A \in co - NP$ מתקיים ש־$x \in A^c$אמ"מ $\neg \exists y R(x,y) = \forall y \neg R(x,y)$

- **הגדרה:** השפה $\Sigma_1 - SAT$ הינה כל הפסוקים הנכונים (נוסחאות מכומתות לגמרי המכומתים באמצעות הכמת $\exists$ בלבד. השפה $\Sigma_k - SAT$ באופן דומה כך שיש $k-1$ חילופי כמתים בנוסחא ומתחילה ב־$\exists$. באופן דומה $\Pi_1 - SAT$ ו־$\Pi_k - SAT$ (להחליף $\exists$ ב־ $\forall$).

- **טענה:** השפה $\Sigma_1 - SAT$ הינה NP שלמה (רדוקציה מ־SAT), השפה $\Pi_1 - SAT$ הינה $co - NP$ שלמה ($\Pi_1 - SAT = (\Sigma_1 - SAT)^c$).

- **הגדרה:** המחלקה Σ_k^p הינה מחלקת השפות שיש להן רדוקציה L ל־$\Sigma_k - SAT$. באופן דומה Π_k^p.

- **מסקנה:** $\Sigma_1^p = NP$, $\Pi_1^p = co - NP$.

- **הגדרה:** מחלקת ההיררכייה הפולינומית הינה: $\mathbf{PH} = \cup_k \Sigma_k^p$.

- **טענה:** $PH \subseteq PSPACE$ (לשים לב שנוסחת $\Sigma_k - SAT$ היא נוסחת $TQBF$). סבורים כי $PH \neq PSPACE$.

- **טענה:** לכל $k \in \mathbb{N}$ מתקיים: $\Pi_k^p \subseteq \Sigma_{k+1}^p$, $\Sigma_k^p \subseteq \Pi_{k+1}^p$.

- אם $PH = PSPACE$ אז קיים k כך ש־ $PH = \Sigma_k^p$ ונאמר שיש קריסה של ההירכייה.

- דוגמא: אם $\Sigma_k^p = \Pi_k^p$ עבור k כלשהו אז $PH = \Sigma_k^p$. הערה: אם יש שפה השלמה ל־PH אז ההירכייה קורסת.

- הערה: מוכיחים בתרגול כי $(\Sigma_k^p)^{\Sigma_l^p} = \Sigma_{k+l}^p$, $(\Pi_k^p)^{\Sigma_l^p} = \Pi_{k+l}^p$.

- **טענה:** $\Sigma_{k+1}^p = NP^{\Sigma_k^p}$.

- הערה: לכל שפה A, אורקל ל־A שקול לאורקל ל־$co - A$. לדוג' מתקיים $NP^{\Sigma_i^p} = NP^{\Pi_i^p}$.

- **הגדרה:** השפה $Exact - IS$ הינה שפת המילים $< G, k >$, מילה בשפה אם בגרף G יש קבוצה בלתי תלויה מגודל k בדיוק ואין קבוצה בלתי תלויה מגודל הגדול מ־k.

- **טענה:** $Exact - IS \in \Sigma_2^p$

- **הגדרה:** השפה $Optimal/min - Circuit$ הינה שפת כל המעגלים הבוליאניים כך שאין מעגל קטן יותר המחשב את אותה הפונקציה שהם מחשבים.

- **טענה:** $Optimal-Circuit \in \Pi_2^p$
- **הגדרה:** השפה $SAT-Solver$ הינה שפת כל המעגלים הבוליאניים כך שמעגל C בשפה אם מתקיים: לכל נוסחה φ, $C(\varphi)=1$ אמ"מ φ ספיקה (C מקבל קידוד של נוסחה).
- **טענה:** $SAT-Solver \in \Pi_2^p$. אפילו ישנה שייכות ל-$co-NP=\Pi_1^p$.

<u>הרצאה 7:</u>

- **טענה:** $SAT-Solver \in \Pi_1^p$ (הרעיון הוא שבהינתן מעגל הפותר את SAT ניתן לחלץ ממנו השמה מספקת ולהשתמש בה כדי לרשום פסוק Π_1-SAT).
- **הגדרה:** מ"ט טיורינג עם עצה (מודל לא יוניפורמי): עבור פ' $t,a:\mathbb{N}\to\mathbb{N}$ נסמן ב-$DTIME(t(n)/a(n))$ את מחלקת השפות שניתן להכריע ע"י מ"ט הרצות בזמן $t(n)$ ומקבלות $a(n)$ ביטים של עצה (<u>אורך העצה תלוי בגודל הקלט</u>) . בנוסף מגדירים: $P/Poly=\cup_{c,d}DTIME(n^d/n^c)$ (עצות באורך פולי' וזמן ריצה פולי').
- **טענה**(ש.ב): $A\in P/Poly$ אמ"מ ניתן לחשב את A בעזרת סדרת מעגלים באורך פולינומי.
- **משפט** ($Karp-Lipton$)**:** אם $NP\subseteq P/Poly$ אז $PH=\Sigma_2^p$ (לכן סביר שלא ניתן לחשב את NP בעזרת סדרת מעגלים בגודל פולינומי).
- הערה: בתרגיל בית ראינו שלכל $c\in\mathbb{N}$ יש שפה ב-PH עבורה אין סדרת מעגלים המחשבת אותה בגודל n^c. <u>זה לא מהווה הוכחה לשאלה האם $PH\not\subseteq P/Poly$</u> (זו שאלה פתוחה).
- **אלגו' הסתברותי:** עבור שלוש מטריצות A,B,C מעל F_2 רוצים לדעת האם $AB=C(mod2)$. מגרילים t וקטורים באקראי $v_1,..,v_t$ ובודקים $(AB-C)v_i=0$. אם זה מתקיים עבור כל הוקטורים אז מקבלים ואחרת דוחים. השגיאה היא חד כיוונית וניתן להקטין אותה כרצוננו (השפה ב-$co-RP$). זמן ריצה פולינומיאלי. אלגו' מסוג זה (מותר גם שגיאה דו כיוונית) נקראים אלגו' מונטה קרלו (בנוסף ניתן להראות כי השפה גם ב-P) .
- **אלגו' הסתברותי:** בהינתן n רוצים למצוא ראשוני p כך ש- $2^n\le p\le 2^{n+1}$. ננחש מספר בן $n+1$ ספרות ונבדוק ראשוניות (בזמן פולינומיאלי). אם כן נחזיר אותו, אחרת ננחש שוב (מתבסס על כך שההסת' למצוא ראשוני בקטע היא לפחות $\frac{1}{3n}$). נשים לב שהאלגו' לא שוגה אולם ייתכן ולא עוצר. עם זאת, תוחלת מס' ההרצות היא פולינומיאלית (מ"מ מקרי גיאומטרי עם הסתברות הצלחה $\frac{1}{3n}$ ולכן תוחלת מס' ההרצות היא $O(n)$). אלגו' מסוג זה נקראים אלגו' לאס וגאס.

<u>הרצאה 8:</u>

- **הגדרה:** כמה הגדרות שקולות ל**מ"ט הסתברותית. הגדרה 1:** מ"ט עם 2 (או יותר) פ' מעברים δ_0,δ_1 כשבכל צעד בוחרים אחת מהן בהסתב' $\frac{1}{2}$ (ההגרלות הן $i.i.d$). **הגדרה 2:** מגדירים פ' מעברים $\delta:[(Q\times\Gamma)\times(Q,\Gamma,\{\to,\leftarrow,-\})]\to[0,1]$ כך שמתקיים לכל q,σ: $\Sigma_{q',\sigma',head\ pointer}\delta(q,\sigma,q',\sigma',head\ pointer)=1$. בוחרים כל צעד לפי ההסתב' ש-δ משרה. **הגדרה 3:** מ"ט עם סרט נוסף עליו הראש יכול לזוז ימינה בלבד ובכל פעם שראש זה זז נכתב עליו באקראי 0 או 1 בהסת' $\frac{1}{2}$ ($i.i.d$).

- **הגדרה:** מט"ה מכריעה שפה A בזמן $t(n)$ אם לכל קלט x זמן הריצה הוא לכל היותר $t(|x|)$ ומתקיים: $\begin{cases} x \in A & Pr(M\ acc\ x) \geq \frac{2}{3} \\ x \notin A & Pr(M\ rej\ x) \geq \frac{2}{3} \end{cases}$ (כלומר הסת' שגיאה $\frac{1}{3}$).

- סימון: $BPTIME(t(n))$=אוסף השפות שניתן להכריע ע"י מט"ה הרצה הזמן $t(n)$. נגדיר $\mathbf{BPP} = \cup_c BPTIME(n^c)$ (=$BPP(\frac{1}{3}, \frac{1}{3})$).

- הערה: ניתן להגדיר את השגיאה באופן כללי: $A \in BPP(\alpha, \beta)$ אם מתקיים $\begin{cases} x \in A & Pr(M\ rej\ x) \leq \beta \\ x \notin A & Pr(M\ acc\ x) \leq \alpha \end{cases}$.

- **הגדרה** נוספת ל-$BPP(\alpha,\beta)$: כל השפות $A \subseteq \Sigma^*$ עבורן יש פולינומים t, r ומ"ט דטרמיניסטית $M(x,y)$ ($|y| \leq r(|x|)$) כך שלכל קלט (x,y) זמן הריצה הוא לכל היותר $t(|x|)$ ומתקיים: $\begin{cases} x \in A & Pr_{y \in_u \{0,1\}^{r(x)}}(M\ rej\ (x,y)) \leq \beta \\ x \notin A & Pr_{y \in_u \{0,1\}^{r(x)}}(M\ acc\ (x,y)) \leq \alpha \end{cases}$ (לחשוב על y כעל הסרט הרנדומי).

- הערה: כל ההגדרות הנ"ל מדברות על אלגו' מונטה קרלו. עבור אלגו' לאס וגאס מגדירים את המחלקה ZPP.

- **הגדרות:** $\mathbf{RP} = BPP(0, \frac{1}{2})$, $\textbf{co-RP} = BPP(\frac{1}{2}, 0)$.

- **הגדרה:** המחלקה **ZPP**: **הגדרה 1:** מחלקת השפות A כך שקיים קבוע $c \in \mathbb{N}$ ומט"ה M שמכריעה את A, כך ש<u>תוחלת</u> זמן הריצה של M הוא $O(n^c)$ והסתברות השגיאה של M היא 0 ("אלגוריתמי לאס וגאס").
  הגדרות שקולות: **הגדרה 2:** $ZPP = RP \cap co-RP$. **הגדרה 3:** $A \in ZPP$ אמ"מ קיימת מט"ה M <u>שתמיד</u> רצה בזמן פולינומי ומתקיימות התכונות: א. לכל $x \in A$, M מקבלת או מכריזה "אינני יודעת". ב. לכל $x \notin A$, M דוחה או מכריזה "אינני יודעת". ג. לכל x, M מכריזה "אינני יודעת" בהסתברות לכל היותר $\frac{1}{2}$.

- **משפט:** (לכל קבוע c) **א.** $RP = BPP(0, \frac{1}{2}) = BPP(0, \frac{1}{2^{n^c}})$.
  ב. $BPP = BPP(\frac{1}{3}, \frac{1}{3}) = BPP(\frac{1}{2^{n^c}}, \frac{1}{2^{n^c}})$.
  ג. לכל קבוע $\frac{1}{n^c} \leq \alpha \leq 1 - \frac{1}{n^c}$: $BPP = BPP(\alpha - \frac{1}{n^c}, 1 - \alpha - \frac{1}{n^c})$.

- תזכורת: א"ש צ'רנוף ($multiplicative\ form$): יהיו מ"מ ב"ת $X_1, .., X_k$ המתפלגים ברנולי כך ש- $Pr(x_i = 1) = p$, $PR(x_i = 0) = 1 - p$. נסמן $X = \Sigma_{i=1}^k X_i$, $\mu = E[X] = p \cdot k$ אזי לכל $0 \lneq \delta \lneq 1$ מתקיים: $Pr(X \geq (1+\delta)pk) \leq e^{\frac{-\delta^2 \cdot p \cdot k}{3}}$.

- **אלגו' הסתברותי:** חישוב השפה $2 - SAT$ (ראינו כבר שייכות ל-P). (נציב תחילה $\bar{x} = 0$. אם הנוסחה לא הסתפקה אז נבחר פסוקית לא מסופקת כלשהי ובאקראי נבחר את אחד הליטרלים בה ונשנה את ערך ההשמה עליו. נמשיך כך k צעדים. אם במהלך התהליך מצאנו השמה מספקת אז נגיד שהנוסחה ספיקה. אחרת נחזיר לא ספיקה. בחירת $k = \frac{n^2}{\epsilon}$ תבטיח שגיאה ϵ קטנה כרצוננו. ניתוח בעזרת "הילוך שיכור").

- **משפט (**$Adelman$**):** $BPP \subseteq P/Poly$ **(**מראים שרוב המחרוזות הרנדומיות y באורך פולינומי "טובות" לכל קלט x ($|x| = n$). ניקח אחת כזו והיא תשמש כעצה לכל הקלטים באורך n).

- **מסקנה:** אם $NP \subseteq BPP$ אז $PH = \Sigma_2^p$ (לפי אדלמן+קארפ־ליפטון).
- **טענה**(לא הוכחנו): $RP \subseteq NP$.

<u>הרצאה 9:</u>

- הערה: מתקיים כי $BPP \subseteq EXP, PSPACE$ (לעבור על 2^n מחרוזות רנדומיות).
- עובדה: $BPP = co-BPP$.
- **משפט(**$Sipser$**):** $BPP \subseteq \Sigma_2^p \cap \Pi_2^p$ **(**מראים ש־$BPP \subseteq \Sigma_2^p$ ולהשתמש בעובדה הקודמת. בנוסף, משתמשים ברעיון חדש: כשקבוצה מכסה חלק משמעותי מהמרחב בו היא נמצאת, מעט "הזזות" שלה מכסות את כל המרחב [השתמשנו בהוכחה הסתברותית להוכחת קיום הזזה כזאת]. לעומת זאת כשקבוצה קטנה לא ניתן לעשות זאת).
- $Polynomial\ Identity\ Testing$: יהי F שדה ($\mathbb{Q}$ או F_p, F_{p^n}[שדה סופי גדול מספיק]). נתונה מטריצה $X_{n\times n}$ ($n << |F|$) כך שכל איבר שלה הוא סקלר מהשדה או משתנה מבין $\{x_1, .., x_n\}$. השאלה היא האם $Det(X) = 0$, כלומר האם הפולינום המתקבל מחישוב הדטרמיננטה הוא פולינום האפס (כל מקדמיו 0). השאלה שקולה למציאת הצבה למשתנים במטריצה כך שדרגתה תהיה מלאה. מהלמה של שוורץ־ציפל מקבלים שעבור בחירה מקרית $\alpha \in F^n$: $Det(X(\bar{\alpha})) \neq 0$ בהסתב' גבוהה אם $Det(X) \neq 0$ (אלגוריתם $Black\ Box$).
- הערה: השאלה הזו במובן מסוים הינה הבעיה הכי כללית עבורה ידוע אלגו' הסתב' ולא ידוע אלגו' דטרמיניסטי בסיבוכיות פולינומיאלית. היינו רוצים למצוא קבוצה H (שגודלה פולינומיאלי ב־n) כך שאם $Det(X) \neq 0$ אז יש איבר $\bar{\alpha} \in H$ כך ש־$Det(X(\bar{\alpha})) \neq 0$. תהליך זה נקרא דה־רנדומיזציה. למקרים פשוטים יחסית, נניח דטרמיננטה לינארית, ניתן למצוא אלגו' $black\ box$ דטרמיניסטי שכזה (אולם במקרה הכללי לא ידוע איך לעשות זאת).
- **סיכומון מחלקות הסתברותיות:** ראינו $BPP, RP, co-RP, ZPP(RP \cap co-RP)$.
  ניתן להגדיר גם $RL(Randomized\ Logaritmic\ space)$, $co-RL$.
  $RP \subseteq NP$, $P \subseteq BPP \subseteq P/Poly$, $BPP \subseteq \Sigma_2^p \cap \Pi_2^p$, $BPP = co-BPP$.

<u>הרצאה 10:</u>

- **הוכחות אינטראקטיביות:** ראינו הוכחות בצורת "עד" לבעיית NP. כלומר־ מוכיח נותן הוכחה קצרה (פולי') והמוודא בודק באופן דטרמיניסטי בזמן פולי'. גם במצב בו המוכיח חזק חישובית, מוודא "מוגבל" (דטרמיניסטי) יכול לקבל רק שפות ב־NP. לכן נאפשר למוודא להיות הסתברותי.
- **הגדרה:** שני גרפים $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ איזומורפיים אם קיימת העתקה חח"ע ועל $\varphi : V_1 \to V_2$ כך ש:$(v, u) \in E_1 \iff (\varphi(v), \varphi(u)) \in E_2$.
- הערה: כדי להוכיח $G_1 \approx G_2$ מספיק לכתוב את φ(פולי') ולכן זה כמו במקרה של NP. אולם איך מוכיחים $G_1 \not\approx G_2$? קיים אלגו' דטרמיניסטי הרץ בזמן $n^{(logn)^c}$.
- $[GNISO]$ פרוטוקול עבור $G_1 \not\approx G_2$ (מפתחות פרטיים): המוודא(V) בוחר באקראי: $i \in \{1, 2\}$ ופרמוטציה מקרית $\sigma : [n] \to [n]$ ושולח למוכיח(P) $\sigma(G_i)$. המוכיח

שולח חזרה $j \in \{1,2\}$ (הגרף שממנו לטענתו הגיע הפרמוטציה). המוודא משתכנע (מקבל) אם $j = i$. ניתוח: אם הגרפים איזו' אז המוכיח צודק בהסתב' $\frac{1}{2}$ (לא יכול לדעת מאיזה גרף הגיע הפרמוטציה). אם הגרפים לא איזו' אז המוכיח צודק בהסת' 1.

- **הגדרה:** פרוטוקול כולל את הדברים הבאים: **א.** כמה סיבובי תקשורת יש. **ב.** מי מדבר ראשון. **ג.** כמה ביטים V צריך לוודא, איזו הודעה הוא שולח בהינתן הביטים והיסטוריית התקשורת. **ד.** מה אומר P בכל שלב בהסתמך על היסטוריית התקשורת. **ה.** בהינתן כל השיחה, האם V מחליט לקבל/לדחות.

- **הגדרה** (בשפה של מ"ט): הוכחה אינטראקטיבית עבור שפה A הינה פרוטוקול (P,V) כאשר P מ"ט ללא מגבלת זיכרון וזמן, V מט"ה הרצה בזמן פולינומיאלי. ל־P,V יש גישה בכל עת לקלט $x \in \{0,1\}^*$. P יכולה לכתוב הודעות על סרט ייעודי ש־V יכולה לקרוא ולהפך. בכל צעד בדיוק אחת המכונות "פעילה". נאמר כי ל־A ישנו פרוטוקול אינטרקטיבי המכריע אותה אם מתקיים: $\begin{cases} x \in A & \exists P : Pr(V\ acc\ x) \geq \frac{2}{3} \\ x \notin A & \forall P : Pr(V\ acc\ x) \leq \frac{1}{3} \end{cases}$.

- **הגדרה:** המחלקה **IP** מכילה את כל השפות עבורן קיימת הוכחה אינטרקטיבית (<u>מפתחות פרטיים</u>).

- **הוכחות באפס ידיעה:** מערכת הוכחה אינטראקטיבית בה המוכיח משכנע בהסתב' גבוהה את המוודא באמיתות הטענה, מבלי לשפוך אור על הטענה עצמה (על האופן בו יש להוכיח את הטענה).

- דוגמה: הוכחת אפס ידיעה לבעיית $3-Color$ (NP שלמה).
$3-Color = \{G | there\ exists\ a\ legal\ 3-vertex\ coloring\}$. פרוטוקול: P בוחר צביעה ב־3 צבעים ומכסה אותה, בנוסף בכל שלב של הפרוטוקול יעשה פרמוטציה מקרית על $\{1,2,3\}$. V בוחר קשת ומבקש מ־P לחשוף את צבעי הקודקודים. ניתוח: אם הגרף 3־צביע : המוודא רואה בכל שלב 2 צבעים שונים מקריים. אם לא 3־צביע: ההסת' שינחש קשת עם צביעה לא חוקית $\leq \frac{1}{\binom{n}{2}}$. עבור n^3 חזרות המוודא "תופס" את המוכיח בשקר בהסתברות הגדולה מ־$1 - \frac{1}{e^n}$.

- **הגדרה:** הוכחות אינטראקטיביות עם <u>מפתחות ציבוריים</u>: **פרוטוקול ארתור־מרלין:** $A = Arthur,\ M = Merlin$. מרלין יודע את הביטים המקריים של A. במקרה זה מדברים על מס' קבוע של סיבובים. $AM[k]$: k סיבובים בהם מדברים ארתור ומרלין לסירוגין ($AM[2] = AM$). מסמנים גם ב־AM את השפות עבורן יש פרוטוקול AM. הסימונים עבור MA הם בהתאמה.

- **הגדרה**(אלטרנטיבית):
y הינה ההודעה של מרלין, z הם הביטים המקריים בשלב החישוב של ארתור.

$$MA = \{L | there\ exists\ PMT\ M : \begin{cases} x \in L & \exists y Pr_z[M(x,y,z)\ acc] \geq \frac{2}{3} \\ x \notin L & \forall y Pr_z[M(x,y,z)\ acc] \leq \frac{1}{3} \end{cases}$$

z הם הביטים המקריים ששלח ארתור, y הינה ההודעה של מרלין.

$$AM = \{L | there\ exists\ PMT\ M : \begin{cases} x \in L & Pr_z[\exists y M(x,y,z)\ acc] \geq \frac{2}{3} \\ x \notin L & Pr_z[\exists y M(x,y,z)\ acc] \leq \frac{1}{3} \end{cases}$$

* z,y הינם בגודל פולי' באורך הקלט.

- הערה: לשים לב ש- $NP, BPP \subseteq MA, AM$.

- הערה: ניתן להניח בה"כ שהסת' הקבלה היא 1 עבור קלט בשפה. בנוסף, ניתן להקטין את השגיאה כרצוננו בלי להוסיף סיבובים (תרגיל).

- **טענה** (ש.ב): $AM \subseteq \Pi_2^p$ (ההוכחה נובעת מהגדרת AM [אם הקלט בשפה אז מקבלים בהסתברות 1] + רדוקציה L: $(\varphi(x) = \forall z \exists y (M(x,y,z) = 1)$).

- **משפט: א.** $MA \subseteq AM$.
ב. $AM[k] = AM$ ($k \geq 2$).
ג. $IP[k] \subseteq AM[k+2] = AM$, עבור קבוע k (כלומר כדי לעבור ממפתחות פרטיים לציבוריים נדרשים רק 2 סיבובים נוספים. מס' הסיבובים עם מפתחות ציבוריים לא משנה).
ד. **(**$Shamir$**)** $IP = PSPACE$**.** מס' הסיבובים ב-IP פולי' באורך הקלט ($\subseteq$: מכונת $PSPACE$ תעבור על כל השיחות האפשריות בין P ל-V. $\supseteq$: מוכיחים כי $\#3-SAT \subseteq IP$ ע"י אריתמטיזציה ותיאור פרוטוקול מתאים. בפועל בשביל הטענה המקורית מוכיחים $TQBF \in IP$ ורעיון ההוכחה דומה).

- **מסקנה:** ניתן להגביל את P להיות חזק כמו $PSPACE$ (הוא לא מסוגל לשכנע עבור שפות חזקות מכך).

__הרצאה 11:__

- הערה:
$\#3-SAT = \{(\varphi,k) | \varphi:\ 3-SAT\ formula, k:\ num\ of\ satisfying\ assignments\}$

- **בעיות אופטימיזציה:** נתון יחס $R(x,y)$ ($|y| \leq |x|^c$) כך שניתן לבדוק ביעילות האם $(x,y) \in R$. בנוסף ישנה פ' $Val_x : \{0,1\}^y \to \mathbb{R}_+$ ("כמה טוב הפתרון y"). הבעיה: בהינתן x למצוא y הממקסם את Val_x כלומר למצוא $argmax_y(Val_x(y))$ (לעיתים נרצה $argmin_y(Val_x(y))$).

- **הגדרה:** נאמר כי אלגו' A הוא c–קירוב עבור בעיית האופטימיזציה (בעית מקסימום) אם הוא מחשב ערך y_A כך ש: $\frac{1}{c} \cdot \overbrace{max_y(Val_x(y))}^{=OPT(x)} \leq Val_x(y_A) \leq \overbrace{max_y(Val_x(y))}^{=OPT(x)}$
(במקרה של מינימום: $min \leq \square \leq c \cdot min$).

- **הגדרה:** השפה $Vertex\ Cover$ (NP שלמה): נתון גרף G ורוצים למצוא קבוצת קודקודים מינימלית $S \subseteq V$ כך שלכל צלע $e = (u,v)$, אחד מקודקודיה ב-S (בעיית מינימזציה).

- **אלגו' קירוב** ל-VC: אלגו' חמדן לא מספיק טוב (פקטור לפחות $ln(n)$ מהפתרון, דוג' לכך תהיה גרף שכבות). אלגו' (הכי טוב שידוע): כל עוד יש צלע לא מכוסה, נוסיף את שני קודקודיה לקבוצה. זהו 2–קירוב ל-VC.

- **אלגו' נוסף:** נציג את הבעיה כבעיית תכנון בשלמים: לכל קודקוד v נגדיר משתנה $x_v \in \{0,1\}$. אז רוצים: $\begin{cases} OPT_{IP} = min(\Sigma_v x_v) \\ x_v + x_u \geq 1,\ \forall (u,v) \in E \end{cases}$. ערך אופטימלי בדיוק

פותר את VC.
אולם זו בעיה NPקשה. נעשה רלקסציה: נשנה את התנאי עבור x_v ל־$0 \leq x_v \leq 1$.
זאת כבר בעיית תכנון לינארי LP־(ב־P לפי אלגו' של $Karmarkar$). נפתור את LP ונחזיר את הקבוצה $S = \{v | x_v \geq \frac{1}{2}\}$ וזה גם כן ייתן $2-$קירוב ל־VC.

- **הגדרה:** $Set\ Cover$ (NP שלמה): קלט: יהי עולם בגודל n $\{1,..,n\}$ ותהיינה קב' $A_1,..,A_m \subseteq \{1,..,n\}$ כך ש־$\cup_{i=1}^m A_i = \{1,..,n\}$. רוצים למצוא קב' מינימלית $S \subseteq \{1,..,m\}$ כך ש: $\cup_{i \in S} A_i = \{1,..,n\}$.

- הערה: VC זהו מקרה פרטי של $Set\ Cover$ (בהינתן G מגדירים $A_v = \{e | v \in e\}$, העולם יהיה כל צלעות הגרף).

- **אלגו' קירוב (חמדן) ל־**$Set\ Cover$**(**הכי טוב שידוע): בכל שלב מוסיפים לכיסוי את הקב' המכסה הכי הרבה איברים לא מכוסים. ניתוח: [ראינו דוגמא בה פקטור הקירוב $\leq \Omega(ln(n))$] מראים כי פקטור הקירוב $\geq O(ln(n))$.

הרצאות 12+13:

- **בעיות הבטחה:** זוג $Y, N \subseteq \{0,1\}^*$ ($Y \cap N = \emptyset$). בהינתן קלט $x \in Y \cup N$ צר.ך להכריע האם $x \in Y$ או $x \in N$.

- **בעיית פער:** (דרך לגרום לבעיית אופטימיזציה להיראות כמו בעיית הכרעה) עבור יחס $R(x,y)$, מובטח שכל קלט x מקיים אחד מהשניים: **א.** יש y כך ש־ $Val_x(y) \geq \beta$. **ב.** לכל y : $Val_x(y) \leq \alpha$
צריך להכריע עבור קלט x איזה מהמקרים מתקיים עבורו.

- הערה: תיאור בעיית הפער כבעיית הבטחה:
$Y = \{x | \exists y, (x,y) \in R, Val_x(y) \geq \alpha\}$, $N = \{x | \forall y, Val_x(y) \leq \beta\}$.

- **טענה**: אם ליחס R יש אלגו' $c-$קירוב אז ניתן לפתור את בעיית הפער $gap - R[\alpha, c \cdot \alpha]$. (נריץ אותו, אם נקבל ערך $>\alpha$ אז נקבל ואחרת נדחה).

- **מסקנה:** אם $gap - A[\alpha, \beta]$ **(**$gap - A[\alpha, c \cdot \alpha]$**)** היא NP קשה אז גם קירוב $\frac{\alpha}{\beta}$ **(**c**)** של A היא NP קשה (כתיב בסוגריים שקול עבור c $\frac{\alpha}{\beta} \leq c$).

- **הגדרה:** רדוקציה בין בעיות פער: רדוקצית קארפ φ בין $gap - A[\alpha_A, \beta_A]$ ל־ $gap - B[\alpha_B, \beta_B]$ נראית כך: עבור קלט טוב $x \in A$, $\varphi(x)$ קלט טוב עבור B. עבור קלט רע $x \in A$, $\varphi(x)$ קלט רע עבור B (כשמגדירים זאת כך, ניתן לחשוב באותו אופן על מקסימיזציה ומינימיזציה).

- **טענה:** בהינתן נוסחת $3 - CNF$ עם בדיוק 3 ליטרלים שונים בכל פסוקית, קיימת השמה המספקת $\frac{7}{8}$ מהפסוקיות.

- **משפט ה־**PCP**:** לכל $0 < \epsilon < \frac{1}{8}$, $gap - 3ESAT[\frac{7}{8} + \epsilon, 1]$ היא NP קשה (כלומר קיימת רדוקציה $\leq_L$ מ־$3SAT$ לבעיית הפער).

- **מסקנה:** $gap - Clique[\frac{1}{3}(\frac{7}{8} + \epsilon), \frac{1}{3}]$ היא NP קשה.

- הערה: על מנת להוכיח שבעיית פער היא NP קשה כדאי להראות רדוקציה מבעיית פער אחרת (רדוקציה מבעיית NP רגילה שקולה להוכחת PCP).

- **הגדרה:** בעיית $Constraint\ Satisfaction\ Graph$: קלט: $U=(V,E,\Sigma,\Phi)$ כך ש: (V,E) ־גרף. Σ ־ קבוצת צבעים $\{1,..,k\}$. $\Phi:E\to\mathbb{P}(\Sigma^2)$ ־ הצבעים המותרים לצביעת קודקודים עבור כל קשת (כל קשת והאילוץ שלה). המטרה היא לצבוע את הקודקודים כך שכל האילוצים מתקיימים (מסמנים $kCSG$ עבור הבעיה עם k צבעים).

- **הגדרה:** 2 בעיות אופטימיזציה (נעבוד עם הראשונה):
א. MAX_V-CSG: למצוא צביעה חלקית $c:V\to\Sigma\cup\{\perp\}$ כך שבגרף המושרה על הקודקודים הצבועים כל האילוצים מסתפקים. מטרה: למקסם את אחוז הקודקודים הצבועים.
ב. MAX_E-CSG: למצוא צביעה מלאה $c:V\to\Sigma$, רוצים למקסם את מספר האילוצים שסופקו.

- **טענה:** $gap_V-3CSG[\frac{7}{8}+\epsilon,1]$ היא NP קשה (רדוקציה מ־$gap-3SAT[\frac{7}{8}+\epsilon,1]$).

- **טענה:** קיימת רדוקציה L כך ש: $gap_V-kCSG[\delta,1]\leq_L gap-IS[\frac{\delta}{k},\frac{1}{k}]$.

- **טענה:** $gap_V-kCSG[\delta,1]\leq_L gap_V-k^l[\delta^l,1]$ ($amplification$).

- **מסקנה:** לכל $\delta>0$ קיים k כך ש־$gap_V-IS[\frac{\delta}{k},\frac{1}{k}]$ היא NP קשה. כלומר, לא ניתן לקרב את IS ביעילות עבור כל פקטור קבוע δ .
($gap_V-3CSG[\frac{7}{8}+\epsilon,1]\overset{amplification}{\leq_L} gap_V-kCSG[\delta,1]\leq_L gap_V-IS[\frac{\delta}{k},\frac{1}{k}]$).