<u>סיכום כללי לקורס: סיבוכיות חישוב</u>

שנה: תשפ"א 2021 (סמסטר א')
מרצה בקורס: ד"ר ערן עמרי
כתבה: הילה שושן

<u>תוכן:</u>

מקרא צבעים: שפות , מחלקות , הגדרות , משפטים חשובים

(1) מחלקות סיבוכיות בסיסיות, שפות וייצוגים:

P = מחלקת כל השפות שיש להן פתרון יעיל, כלומר אלגוריתם פולינומי שמכריע האם מילה בשפה או לא.
NP = מחלקת כל השפות שקל לוודא עבורן פתרון (יעיל), אך לא ידוע האם קיים פתרון יעיל עבורן.

מכונת טיורינג דטרמיניסטית (מ"ט): שלשה (Γ, Q, δ), כאשר:
$\Gamma \supseteq \Sigma \cup \{\Delta, b\}$ $(b = blank)$.
Q = קבוצת מצבים סופית לא ריקה. מכילה תמיד את q_{start}, q_{halt}.
δ = פונקציית המעברים. מגדירה איך לעבור מקונפיגורציה נוכחית לקונפיגורציה הבאה. כלומר: מה לכתוב על הסרט (איפה שמצביע הראש הקורא-כותב), לאן לזוז (ימינה/ שמאלה/ להישאר במקום) ולאיזה מצב בקרה לעבור.
Δ = סימן שנמצא תמיד בתחילת הרצה, בתא השמאלי של כל סרט אינסופי. מימינו תווי הקלט $x \in \Sigma^*$.
b = סימן שנמצא על כל שאר התאים בכל הסרטים במכונה.
חושבים על מ"ט כבעלת 3 סרטים: סרט קלט, סרט עבודה, וסרט פלט. יכולים להיות k סרטי עבודה.
הריצה נעצרת כשמגיעים ל-q_{halt}, והפלט של המכונה הוא המחרוזת שכתובה על סרט הפלט (מימין ל-Δ, ועד ה-b הראשון).

- ישנם מספר מודלים למ"ט. מבחינה חישובית - כולם שקולים. מבחינת סיבוכיות - לכל מ"ט הרצה בזמן $T(n)$, קיימת מ"ט המחשבת את אותה פונקציה ורצה בזמן $O(T(n)^2)$.
- קיימת מ"ט אוניברסלית U אשר לכל M ולכל $x \in \{0, 1\}^*$, כך ש-M רצה על x $T(n)$ צעדי חישוב, U מסמלצת את $M(x)$ בזמן $O(T(n)^2)$. למעשה ניתן גם $c \cdot T(n) \cdot log(T(n))$, c קבוע. (מסמלצת פירושה שהיא עונה כמו M על x).
- זמן ריצה של מ"ט = מספר הפעולות הבסיסיות שהיא מבצעת בעת ריצת האלגוריתם.

$DTIME$

תהי $T: N \rightarrow N$. שפה $L \subseteq \{0, 1\}^*$ היא ב-$DTIME(T(n))$ אם קיימת מ"ט דטרמיניסטית M, הרצה בזמן $O(T(n))$ (כלומר במקרה הגרוע), ומכריעה את L.

P

מחלקת כל השפות שקיימת עבורן מ"ט M הרצה בזמן פולינומיאלי באורך הקלט, עבור כל קלט.

במילים אחרות: $P = \bigcup_{i \in N} DTIME(n^i)$.

- נאמר שקיים אלגוריתם יעיל עבור שפה L אם $L \in P$.
- כל שפה ב-R היא ב-$DTIME$ של איזושהי פונקציה, אבל אנו לא תמיד יודעים לייצג את הפונקציה.

<u>הראינו בכיתה:</u> השפה st-con
$P \ni st - con = \{< G, s, t > | G = (V, E)\ undirected\ graph,\ s, t \in V \wedge \exists\ path\ from\ s\ to\ t\ in\ G\}$

<u>התזה של צ'רץ'-טיורינג:</u> כל מודל חישובי שניתן למימוש פיזי שקול למ"ט (יאפשר חישוב של אותו אוסף פונקציות).
<u>התזה המורחבת:</u> כל מודל חישובי שניתן למימוש פיזי, יגדיר את אותה מחלקה P.

(2) המחלקה NP ו-NP-Complete

NP

שפה $L \subseteq \{0,1\}^*$ היא במחלקה NP אם קיים פולינום $p(\cdot)$ וקיימת מ"ט (מוודאת) פולינומית, M_L, כך שלכל $x \in \{0,1\}^*$ מתקיים: $x \in L \Leftrightarrow \exists y \in \{0,1\}^{p(|x|)}\ s.t.\ M_L(x,y) = 1$.

- אם ניתן ל-M_L קלט x (ועד s) והיא תדחה, זה עדיין לא אומר ש-x לא בקבוצה. אלא זה רק אומר ש-s אינו עד עבור x. ייתכן שיש עד אחר s' עבורו $M_L(x,s') = 1$.

<u>ראינו בכיתה:</u>

- $NPC \ni INDSET = \{< G,k > | \exists S \subseteq V(G),\ |S| = k,\ s.t.\ S\ is\ an\ independent - set\}$
- בעיית הסוכן הנוסע היא $NP - Complete$.
- $GI \notin P,\ GI \notin NP - Complete$ אבל $GI \in NP$, כאשר:

$$GI = \{< G_1, G_2 > | G_1 = (V_1, E_1),\ G_2 = (V_2, E_2),$$
$$\exists\ permutation\ \pi\ s.t.\ (v_1, u_1) \in E_1 \Leftrightarrow (\pi(v_1), \pi(u_1)) \in E_2$$

הערה: קבוצת הגרפים האיזומורפיים לגרף מסוים מתקבלים ע"י כל הפרמוטציות שמשנות את שמות הקודקודים, כלומר $\pi: |V| \rightarrow |V|$.

EXP

מחלקת כל השפות שקיימת עבורן מ"ט M הרצה בזמן אקספוננציאלי באורך הקלט, עבור כל קלט.

$$EXP = \bigcup_{i \in N} DTIME(2^{n^i})$$

<u>טענה:</u> $P \subseteq NP \subseteq EXP$

מ"ט א"ד:

מוגדרת באופן כמעט שקול למ"ט רגילה, מלבד העובדה שיש לה שתי פונקציות מעברים (במקום אחת): δ_0, δ_1. בכל מעבר מקונפיגורציה c ניתן לעבור על פי δ_0 או על פי δ_1.

נאמר שמ"ט א"ד מקבלת את מילה x אם **קיים** חישוב מקבל של M על x, כלומר קיימת סדרת מעברים המוגדרת על פי הבחירות של δ_0, δ_1 בכל שלב, שבסופה מגיעים למצב מקבל ועל סרט הפלא מופיע 1.

- $T_{M,x}$= עץ הקונפיגורציות (של מכונה M על קלט x ספציפי). גרף מכוון, בו הקודקודים הם הקונפיגורציות, הצלעות הן מעברים בין קונפיגורציות על פי δ_0 או δ_1, והעלים הם מצבים סופיים. אם קיים עלה אחד שמתאר קונפיגורציה מקבלת, אז $x \in L(M)$.
- זמן הריצה מוגדר על פי החישוב הארוך ביותר על קלט x.
- נאמר שמ"ט א"ד M רצה זמן $T(n)$ אם לכל קלט $x \in \{0,1\}^*$ ולכל חישוב של M על x, החישוב מסתיים לאחר לכל היותר $T(|x|)$ צעדי חישוב.
- גודל העץ יכול להיות אקספוננציאלי בגודל x בזמן שעומקו פולינומי ב-$|x|$.

$NTIME$

תהי $T: N \rightarrow N$. שפה $L \subseteq \{0,1\}^*$ היא ב-$NTIME(T(n))$ אם קיימת מ"ט א"ד M, הרצה בזמן $O(T(n))$ (כלומר במקרה הגרוע), ו-$L = L(M)$.

- הגדרה חלופית ל-NP: $NP = \bigcup_{i \in N} NTIME(n^i)$

- במילים: יש מכונה מדוואת פולינומית אם קיים פולינום n^i וקיימת מ"ט א"ד M שעובדת בזמן $O(n^i)$

<u>הגדרה:</u> רדוקציה פולינומית/ רדוקציית $Karp$

שפה $L \subseteq \{0,1\}^*$ ניתנת לרדוקציית $Karp$ לשפה $L' \subseteq \{0,1\}^*$, אם קיימת פונקציה (שלוקחת מילה ומחזירה מילה) $f: \{0,1\}^* \rightarrow \{0,1\}^*$ המקיימת:

א. f ניתנת לחישוב בזמן פולינומי.

ב. לכל קלט $x \in \{0,1\}^*$: $x \in L \Leftrightarrow f(x) \in L'$

- נשתמש ברדוקציות כדי להראות "קושי" של שפות: אם $L \leq_p L'$ אז L' קשה לפחות כמו L.

  (לדוגמה: אם L אינה ניתנת לחישוב בזמן פולינומי, אז גם L').
- מכיוון ש- $f(x)$ נוצר בזמן פולינומי, $O(|x|^c)$, אז $|f(x)| \leq k \cdot |x|^c$, עבור k קבוע (כלומר גם פולינומי)

$NP - Hard$

שפה L תקרא NP-קשה אם מתקיים לכל שפה $\hat{L} \in NP$: $\hat{L} \leq_p L$.

$NP - Complete$

שפה L תקרא NP-שלמה אם היא NP-קשה וגם $L \in NP$.

<u>תכונות של רדוקציות:</u>

1. טרנזיטיביות: יהיו L_1, L_2, L_3 שפות. אזי $L_1 \leq_p L_2 \wedge L_2 \leq_p L_3 \Rightarrow L_1 \leq_p L_3$
2. אם $L \in NPH$ וגם $L \in P$ אזי $P = NP$
3. אם $L \in NPC$ אזי: $L \in P \Leftrightarrow P = NP$
4. אם קיימת f כך ש- $L_1 \leq_p L_2$ אז אותה f תראה $\overline{L_1} \leq_p \overline{L_2}$

<u>הגדרה:</u> השפות $SAT, 3SAT$

$SAT = \{\phi \mid \phi \ is\ a\ satisfiable\ CNF\ formula\}$

$3SAT = \{\phi \mid \phi \ is\ a\ satisfiable\ 3CNF\ formula\}$

<u>משפט:</u> קוק לוין

1. SAT היא NP-קשה
2. $3SAT$ היא NP-קשה

- מסקנה: $L \in NPC$: $\forall L \in NP\ s.t.\ SAT \leq_p L \ \vee \ 3SAT \leq_p L$
- דברים חשובים לזכור מההוכחה:
  - עבור השוואה בין שתי מחרוזות בגודל n: ניתן להגדיר נוסחת CNF באורך $2n$.
  - אוניברסליות של AND,OR,NOT: לכל פונקציה בוליאנית $f: \{0,1\}^l \rightarrow \{0,1\}$ קיימת נוסחת CNF, ϕ_f באורך $O(l \cdot 2^l)$ כך שמתקיים לכל $x \in \{0,1\}^l$: $f(x) = \phi_f(x)$.
  - מעבר בין קונפיגורציות: $z_{i-1}(a,b,q) = z_i(y_{inputpos(i)}, b(z_{prev(i)}), q')$

    כאשר: $prev(i)$ = השלב האחרון בחישוב שבו ביקרנו באותו תא עבודה כמו בשלב זה (i).
  - ניתן להגדיר פונקציה $F: \{0,1\}^{2c+1} \rightarrow \{0,1\}^c$ כאשר $0 < c = log(2 \cdot |\Gamma| + |Q|)$ קבוע.

    לפי הטענה הקודמת, ניתן ליצור נוסחה בגודל קבוע המגדירה את F.

    F מקבלת כקלט $inputpos(i)$, $z_{i-1}, z_{prev(i)}$, ומחזירה כפלט את z_i.

הגדרה: מ"ט $Oblivious$

לכל קלט $x \in \{0,1\}^n$, תנועת הראשים הקוראים במכונה לאורך כל החישוב על x זהה לתנועת הראשים הקוראים בחישוב של M על 0^n.
במילים אחרות: הראשים הקוראים נעים לאותו מקום בכל שלב i בחישוב, לכל קלט באורך n.

טענה: לכל מ"ט M העובדת בזמן $T(n)$, קיימת מ"ט $\hat{M}$ שהיא $Oblivious$ ורצה בזמן $O(T(n)^2)$.
[$T(\cdot)$ אינו חסם, אלא פונקציה שנותנת את אותו ערך לכל קלט $x \in \{0,1\}^*$].

בעיות חיפוש מול בעיות הכרעה:

למה: $SAT\ is\ self - reducible$

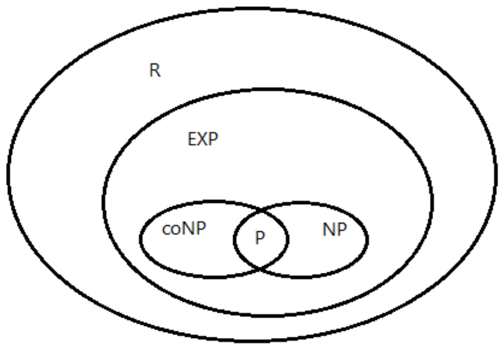
כלומר, בהינתן אלגוריתם יעיל המכריע את SAT (ומכונה מוודאת), ניתן למצוא באופן יעיל עד עבור קלט x.
נניח שקיים אלגוריתם יעיל A אשר בהינתן נוסחת CNF, ϕ, מחזיר האם $\phi \in SAT$, אזי ניתן לייצר אלגוריתם יעיל B אשר מוצא השמה מספקת עבור ϕ.

$coNP$

הגדרה 1: $coNP = \{L \mid \overline{L} \in NP$

הגדרה 2: $coNP = \{L \mid \exists p(\cdot), \exists M_L\ s.t.\ \forall x \in \{0,1\}^*: x \in L \Leftrightarrow \forall y \in \{0,1\}^{p(|x|)}\ M_L(x,y) = 1\}$

- לכל שפה $L \in P$ מתקיים גם: $L \in NP \land L \in coNP$ (כי אפשר לקחת את העד הריק).



הראינו בכיתה: השפה $Tautology$

$coNP \ni Tautology = \{\phi \mid \phi\ is\ a\ CNF\ formula\ for\ which\ any\ z \in \{0,1\}^n\ satisfies\ \phi(z) = 1\}$

$NEXP$

מחלקת כל השפות שקיימת עבורן מ"ט א"ד שמותר לה לעבוד בזמן אקספוננציאלי.

$$NEXP = \bigcup_{i \in N} NTIME(2^{n^i})$$

משפט: אם $EXP \neq NEXP$ אזי $P \neq NP$

- ההוכחה היא בעזרת $padding$: $L_{pad} = \{x \cdot 2^{|x|^c} \mid x \in L\}$ (כלומר שפה עם קלטים ארוכים).
- הרעיון הוא שאפשר להתעלם מ-$2^{|x|^c}$ הביטים שהוספנו, ולהתייחס רק לקלט x.
- ידוע $EXP \subseteq NEXP$

(3) לכסון ומשפטי היררכיה

<u>הגדרה:</u> פונקציות $time - constructible$

פונקציה $T: N \rightarrow N$ נקראת $time - constructible$ אם קיימת מ"ט M_T אשר בהינתן קלט 1^n רצה $O(T(n))$ $(T(n) \geq n)$ צעדים ומחזירה את הערך $T(n)$.

- כמעט כל פונקציה מקיימת זאת.
- $T(n) \geq n$ כדי שיהיה ניתן לעבור על כל הקלט (לפחות לינארי).

<u>משפט:</u> משפט ההיררכיה לסיבוכיות זמן דטרמיניסטית

תהיינה f, g פונקציות $time - constructible$ המקיימות: $f(n) \cdot log(f(n)) = o(g(n))$. אזי:

$DTIME(f(n)) \subset DTIME(g(n))$

כלומר הם מוכלים, וקיימת שפה $L \in DTIME(g(n))$ אבל לא $L \notin DTIME(f(n))$.

- הוכחה בעזרת לכסון, שמסתמכת על התכונות שלכל מ"ט יש אינסוף קידודים, ושכל מחרוזת היא קידוד של מ"ט כלשהי, ושניתן לסמלץ מ"ט בעזרת מ"ט אוניברסלית.

<u>משפט:</u> משפט ההיררכיה לסיבוכיות זמן אי דטרמיניסטית

תהיינה f, g פונקציות $time - constructible$ המקיימות: $f(n + 1) = o(g(n))$ אזי:

$NTIME(f(n)) \subset NTIME(g(n))$

<u>משפט:</u> משפט לנדר (שפה אמצעית ב-NP)

אם $P \neq NP$, אז קיימת שפה $L \in NP \backslash P$ שאינה NP-שלמה.

(מועמדים: $GI,\ Factorial$).

<u>הגדרה:</u> מ"ט עם גישת אורקל

מ"ט עם גישת אורקל לשפה L היא מ"ט רגילה עם סרט נוסף שבו ניתן לשאול שאילתות מהצורה האם $x \in L$, שנענות ב-$O(1)$.

כדי לעשות זאת, M תעבור למצב מיוחד, q_{query} באשר השאילתה x כתובה על הסרט הנוסף.

אם $x \in L \Leftarrow$ בתוך צעד אחד M תעבור למצב q_{yes}, ואם $x \notin L \Leftarrow$בתוך צעד אחד M תעבור למצב q_{no}.

- סימון: $M^O(x)$= החישוב של מ"ט M על קלט x, כאשר נתונה גישת אורקל לשפה O.
- לכל שפה O, נגדיר: P^O= מחלקת כל השפות L שקיימת עבורן מ"ט דטרמיניסטית פולינומית עם גישת אורקל ל-O.
- $coNP,\ NP \subseteq P^{SAT}$
- לכל $L \in P$: $P^L = P$, כי ניתן להחליף את האורקל במכונה המחשבת את L, ע"י החלפת כל שאילתה q בסימולציה של M_L על x.

$EXPCOM$

$$EXPCOM = \{< M, x, 1^n > |\ M(x) = 1\ in\ 2^n\ steps\}$$

- הערה: n הוא לא בהכרח $|x|$!
- לכל מכונה שעוצרת, לא משנה באיזה זמן, קיים לכל קלט n מספיק גדול כך ש-$< M, x, 1^n >$ בשפה.

<u>טענה:</u> $P^{EXPCOM} \subseteq NP^{EXPCOM} \subseteq EXP$

<u>טענה:</u> $EXP \subseteq P^{EXPCOM}$

<u>משפט:</u> קיימים אורקלים A, B כך שמתקיים: $P^A = NP^A,\ P^B \neq NP^B$

- מסקנה: לא ניתן להכריע את שאלת $P\ v.s.\ NP$ בעזרת הוכחות רלטיביות (לדוגמה לכסון).
הסיבה היא שהתכונות עליהן מסתמכות הוכחות של לכסון (הוזכרו למעלה), נכונות גם כאשר מדובר על מ"ט עם גישת אורקל. כלומר אם הייתה הוכחה שמשתמשת בתכונות האלו, אז המשפט הזה יסתור אותה. כי קיים אורקל שההוכחה הזו הייתה עובדת לגביו, אבל שלילתה לא (או להפך).
בפירוט:
  - אם ההוכחה הייתה אומרת $P = NP$, אזי לכל אורקל O היה מתקיים (ע"פ אותה הוכחה), ש-$P^O = NP^O$
  - ואם ההוכחה הייתה אומרת $P \neq NP$, אזי לכל אורקל O היה מתקיים (ע"פ אותה הוכחה), ש-$P^O \neq NP^O$

(4) סיבוכיות זכרון

$DSPACE$

תהא $S: N \rightarrow N$, ותהא שפה $L \in \{0,1\}^*$. נאמר שמתקיים $L \in DSPACE(S(n))$ אם קיים קבוע $c > 0$ וקיימת מ"ט דטרמיניסטית M המכריעה את L כך שלכל קלט $x \in \{0,1\}^*$, M בחישוב על x משתמשת בלכל היותר $c \cdot S(|x|)$ תאי זכרון.

$NSPACE$

אותו הדבר כמו $DSPACE$, רק עם מ"ט א"ד.

<u>הגדרה:</u> פונקציות $space - constructible$

פונקציה $S: N \rightarrow N$ תיקרא $space - constructible$ אם קיימת מ"ט דטרמיניסטית N אשר בהינתן קלט 1^n כותבת על סרט הפלט את $S(n)$ ורצה בסיבוכיות זכרון $O(S(n))$.

- כמעט כל פונקציה היא כזאת.
- תהי $S(n)$ פונקציה TC/SC, אזי בהכרח מתקיים: $DTIME(S(n)) \subseteq DSPACE(S(n))$ [כי אם מ"ט רצה בזמן k, היא אינה יכולה לגעת ביותר מאשר k תאים].

<u>משפט:</u> לכל פונקציה SC, $S: N \rightarrow N$ מתקיים:

$DTIME(S(n)) \subseteq DSPACE(S(n)) \subseteq NSPACE(S(n)) \subseteq DTIME(2^{O(S(n))})$

$PSPACE$

מחלקת כל השפות שקיימת עבורן מ"ט דטרמיניסטית המכריעה אותן, ומשתמשת בזכרון פולינומיאלי באורך הקלט.

$$PSPACE = \bigcup_{i \in N} DSPACE(n^i)$$

<u>הראינו בכיתה:</u> $3SAT \in PSPACE$

- מסקנה: $NP \subseteq PSPACE$, [כי אפשר להפעיל רדוקציה מכל שפה ב-NP ל-$3SAT$, ואז להפעיל את האלגוריתם שמראה שייכות של $3SAT$ ל-$PSPACE$].
- $coNP \subseteq PSPACE$ (אפשר לעבור על כל העדים ולחפש עד "רע").

<u>ידוע:</u> $PSPACE \subseteq EXP$

<u>לא ידוע:</u> $EXP \subseteq P,\ PSPACE \subseteq P,\ PSPACE \subseteq NP$

ממשפט ההיררכיה לסיבוכיות זמן, נוכל לבחור: $g(n) = 2^n$, $f(n) = n^{log(n)}$, ואז:
$P \subseteq DTIME(n^{log(n)}) \subset DTIME(2^n) \subseteq EXP$, ולכן: $P \subset EXP$.

$NPSPACE$

מחלקת כל השפות שקיימת עבורן מ"ט א"ד המכריעה אותן, ומשתמשת בזכרון פולינומיאלי באורך הקלט.

$$NPSPACE = \bigcup_{i \in N} NSPACE(n^i)$$

DL או L

מחלקת השפות שניתן לחשב אותן בזכרון לוגריתמי בעזרת מ"ט דטרמיניסטית.

$L = DL = DSPACE(log(n))$

- ידוע: $DL \subseteq P \subseteq NP$
- לא ידוע: $DL = P, NP, NL$
- מהמשפט: $DSPACE(log(n)) \subseteq DTIME(2^{O(log(n))}) = DTIME(n^{O(1)}) = P$

NL

מחלקת השפות שניתן לחשב אותן בזכרון לוגריתמי בעזרת מ"ט א"ד.

$NL = NSPACE(log(n))$

- ידוע: $DL \subseteq NL$

הראינו בכיתה: השפה $PATH$

$NL \ni PATH = \{< G, s, t > | G\ is\ a\ directed\ graph\ and\ \exists\ path\ from\ s\ to\ t\ in\ G\}$

- $PATH$ היא כמו $st - con$ רק בגרף מכוון.
- לא ידוע האם $PATH \in DL$, אך ידוע $st - con \in DL$

משפט: משפט ההיררכיה לסיבוכיות זכרון

תהיינה f, g פונקציות $space - constructible$, המקיימות $f(n) = o(g(n))$, אזי מתקיים:

$DSPACE(f(n)) \subset DSPACE(g(n))$

- אפשר להוכיח ע"י לכסון, בדומה למשפט ההיררכיה לסיבוכיות זמן.

משפט: משפט סביץ'

לכל פונקציה SC, $S: N \rightarrow N$, כך ש-$S(n) \geq log(n)$ מתקיים: $NSPACE(S(n)) \subseteq DSPACE(S(n)^2)$

- מסקנה: $NL \subseteq DSPACE(log(n)^2)$

$\Downarrow$

$(1)\ PATH \in DSPACE(log(n)^2)$

$(2)\ NPSPACE \subseteq PSPACE \Rightarrow PSPACE = NPSPACE$

הערות: עבור מ"ט M שרצה בסיבוכיות זכרון $S(n)$

- אורכה של קונפיגורציה כלשהי בחישוב $M(x)$: $log(|Q|) + |\Gamma| \cdot S(n) = O(S(n))$
- חסם על מספר הקונפיגורציות: $2^{O(S(n))}$

$PSPACE - Complete$

מחלקת כל השפות $L \in PSPACE$ כך ש: $\forall L' \in PSPACE:\ L' \leq_p L$ (רדוקציות פולינומיאליות בזמן ריצה).

הראינו בכיתה: השפה $SpaceTM$

$$PSPACE - Comp \ni SpaceTM = \{< M, w, 1^n > M\ is\ a\ deterministic\ TM\ s.t.\ M(w) = 1\ by\ using\ at\ most\ n\ cells\}$$

הגדרה: QBF

נוסחה בוליאנית עם כמתים (QBF) היא נוסחה מהצורה: $Q_1 x_1 Q_2 x_2 ... Q_n x_n\ \phi(x_1 ... x_n)$ כאשר: $Q_i \in \{\forall, \exists\}$,

ו-ϕ הינה נוסחה בוליאנית (לאו דווקא CNF), מעל משתנים $x_1, ..., x_n$.

- לנוסחה כזו יש תמיד ערך $True$ או $False$.
- ניסוח חדש ל-SAT: $SAT = \{\psi |\ \exists x_1 \exists x_2 ... \exists x_n\ \phi(x_1, ..., x_n)\ s.t.\ \phi\ is\ a\ CNF\ formula,\ \psi = T\}$

- ניסוח חדש ל-$Tautology$: $\{\psi |\ \forall x_1 \forall x_2 ... \forall x_n\ \phi(x_1,..., x_n)\ s.t.\ \ \phi\ is\ a\ CNF\ formula,\ \psi\ =\ T\}$

<u>הגדרה:</u> השפה $TQBF$

$TQBF\ =\ \{\psi |\ \psi\ is\ a\ QBF\ \ \wedge\ \ \psi\ =\ T\}$

<u>משפט:</u> $TQBF$ היא $PSPACE\ -\ Complete$

- קיימת נוסחה $\phi(c, c') \Leftrightarrow (c, c')\ \in\ E$ (בגרף הקונפיגורציות) שאורכה פולינומי ב-$|c, c'|$.

<u>הגדרה:</u> רדוקציות בזכרון לוגריתמי מובלע

פונקציה $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$ תיקרא חשיבה בזכרון לוגריתמי מובלע, אם f חסומה פולינומית, כלומר: $\exists p\ \forall x\ \in\ \{0, 1\}^*\ |f(x)|\ \leq\ O(|x|^c)$ עבור קבוע $c\ >\ 0$, וגם שתי הפונקציות הבאות שייכות ל-DL:

1) $L_f\ =\ \{< x, i >\ |\ f(x)[i]\ =\ 1\}$

כלומר בהינתן שהאינדקס בתוך $f(x)$, האם הביט ה-i של $f(x)$ שווה ל-1.

2) $L_f{}'\ =\ \{< x, i >\ |\ |f(x)|\ \geq\ i\}$

כלומר האם האינדקס (i) הוא בתוך $f(x)$.

שפה B ניתנת לרדוקציה בזכרון לוגריתמי מובלע לשפה A (סימון: $B \leq_l A$), אם קיימת רדוקציה f הניתנת לחישוב בזכרון לוגריתמי מובלע ומקיימת: $\forall x\ \in\ \{0, 1\}^*:\ x\ \in\ B\ \ \Leftrightarrow\ \ f(x)\ \in\ A$.

- הראינו הגדרה שקולה:

f רדוקציה אם קיימת מ"ט המחשבת אותה בזכרון לוגריתמי **בסרטי העבודה,** אך עם סרט פלט לכתיבה בלבד, שאינו מוגבל בזכרון, ומותר בו לנוע ימינה/ להישאר במקום בלבד (אסור לחזור).

$NL\ -\ Complete$

שפה $A\ \in\ NL$ שלמה ב-NL אם לכל שפה $B\ \in\ NL$ מתקיים $B \leq_l A$.

<u>למה:</u>

1. אם $B \leq_l C$ וגם $C \leq_l D$, אזי $B \leq_l D$ (טרנזיטיביות).
2. אם $B \leq_l C$ וגם $C\ \in\ DL$, אזי $B\ \in\ DL$.

- הערה לגבי הוכחת 1: צ"ל $x\ \in\ B \Leftrightarrow f(x)\ \in\ C \Leftrightarrow g(f(x))\ \in\ D$, אבל אי אפשר לעשות זאת באופן הרגיל, של שליחת $f(x)$ למכונה שמחשבת את g מכיוון שהוא יכול להיות באורך שאינו לוגריתמי, ואז לא נוכל לכתוב אותו כפלט באף מקום, כי המכונה חייבת לעבוד בזכרון לוגריתמי בסרטי העבודה.

לכן מה שעושים זה להשתמש ב-$L_f, L_f{}'$ ולחשב את הביט ה-i ב-f בכל פעם מחדש, וכך לסמלץ את קריאת f.

<u>הראינו בכיתה:</u> $PATH\ is\ NL\ -\ Complete$

<u>הגדרה אלטרנטיבית:</u> NL

שפה L נמצאת במחלקה NL אם קיימת מ"ט דטרמיניסטית M עם סרט קלט נוסף לקריאה יחידה (בו אסורה תנועה שמאלה), וקיים פולינום $p(\cdot)$ כך שלכל $x\ \in\ \{0, 1\}^n$ מתקיים:

$x\ \in\ L \Leftrightarrow \exists y\ \in\ \{0, 1\}^{p(n)},\ M(x, y)\ =\ 1$. [$y$ כתוב על סרט הקריאה היחידה בעת חישוב M על (x, y), ו-M משתמשת בזכרון לוגריתמי בכל סרטי העבודה].

$coNL$

$coNL = \{L \subseteq \{0,1\}^* \mid \overline{L} \in NL\}$

- בפרט $\overline{PATH} \in coNL$, ואפילו שלמה ב-$coNL$, כי זו אותה רדוקציה כמו של $PATH$.
- הראינו גם: $\overline{PATH} \in NL$.
- <u>מסקנה:</u> $NL = coNL$!

עד כה ידוע לנו:

$$\boxed{DL \subseteq NL = coNL \subseteq P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXP}$$

(5) ההיררכיה הפולינומית

נסתכל על הקטע בין NP ל-$PSPACE$ ולראות מה קורה ביניהם ביותר פירוט.
נשים לב: ממשפט ההיררכיה לסיבוכיות זכרון דטרמיניסטית: $DL \subset PSPACE$
וממשפט ההיררכיה לסיבוכיות זמן דטרמיניסטית: $P \subset EXP$
ובנוסף: $P \subset DTIME(n^{log(n)})$

$\sum_2^p$

אוסף כל השפות $L \subseteq \Sigma^*$ שקיימת עבורן מ"ט דטרמיניסטית M הרצה בזמן פולינומי, ופולינום $q(\cdot)$ כך שלכל $x \in \{0,1\}^n$ מתקיים: $x \in L \Leftrightarrow \exists u_1 \in \{0,1\}^{q(n)}\ \forall u_2 \in \{0,1\}^{q(n)}\ M(x, u_1, u_2) = 1$

- אבחנה 1: $NP \subseteq \sum_2^p$ (כי אפשר להתעלם מ-u_2).
- אבחנה 2: $coNP \subseteq \sum_2^p$ (כי אפשר להתעלם מ-u_1).

<u>הראינו בכיתה:</u> השפה $ExactINDSET$

$$\sum_2^p \ni ExactINDSET = \{< G, k > \mid G\ is\ an\ undirected\ graph\ with\ IS\ of\ size\ k,$$
$$but\ with\ no\ IS\ of\ size\ k + 1\}$$

- הערה: $ExactINDSET \notin NP$, כי אין עד מתאים (דורש גם עד שמעיד על קיום IS בגודל k, שזה כוח חישוב של NP, וגם עד שמעיד על אי קיום של IS בגודל $k + 1$, שזה $coNP$).

<u>משפט:</u> משפט Umans, השפה $MIN - EQ - DNF$

$$\sum_2^p - Comp \ni MIN - EQ - DNF = \{< \phi, k > \mid \phi\ is\ a\ DNF\ formula\ \wedge$$
$$\exists\ DNF\ formula\ \psi\ of\ size \leq k\ \ s.t.\ \phi \equiv \psi\}$$

(תחת רדוקציות פולינומיאליות).

$\prod_2^p$

אוסף כל השפות $L \subseteq \Sigma^*$ שקיימת עבורן מ"ט דטרמיניסטית N הרצה בזמן פולינומי, ופולינום $q(\cdot)$ כך שלכל $x \in \{0,1\}^n$ מתקיים: $x \in L \Leftrightarrow \forall u_1 \in \{0,1\}^{q(n)}\ \exists u_2 \in \{0,1\}^{q(n)}\ M(x, u_1, u_2) = 1$

- אבחנה 1: $NP \subseteq \prod_2^p$
- אבחנה 2: $coNP \subseteq \prod_2^p$
- אבחנה 3: $ExactINDSET \in \prod_2^p$
- אבחנה 4: $\overline{MIN - EQ - DNF} \in \prod_2^p - Comp$, כאשר:

$\overline{MIN-EQ-DNF} = \{<\phi, k> \mid \phi\ is\ a\ DNF\ formula\ \wedge \forall\ DNF\ formula\ \psi\ of\ size \leq k$
$\exists z\ s.t.\ \phi(z) = \psi(z)\}$

$\sum_i^p$

לכל $N \ni i > 0$, נגדיר את המחלקה $\sum_i^p$ להיות אוסף השפות L אשר קיימת עבורן מ"ט דטרמיניסטית פולינומית, M_L, וקיים פולינום $q(\cdot)$ כך שלכל $x \in \{0,1\}^*$:

$$x \in L \Leftrightarrow \exists u_1 \in \{0,1\}^{q(|x|)} \forall u_2 \in \{0,1\}^{q(|x|)} \ldots Q_i u_i \in \{0,1\}^{q(|x|)}\ M(x, u_1, u_2, \ldots, u_i) = 1$$

[עבור כל i אי זוגי: $Q_i = \exists$, ועבור כל i זוגי: $Q_i = \forall$].

$\prod_i^p$

לכל $N \ni i > 0$, נגדיר את המחלקה $\prod_i^p$ להיות אוסף השפות L אשר קיימת עבורן מ"ט דטרמיניסטית פולינומית, M_L, וקיים פולינום $q(\cdot)$ כך שלכל $x \in \{0,1\}^*$:

$$x \in L \Leftrightarrow \forall u_1 \in \{0,1\}^{q(|x|)} \exists u_2 \in \{0,1\}^{q(|x|)} \ldots Q_i u_i \in \{0,1\}^{q(|x|)}\ M(x, u_1, u_2, \ldots, u_i) = 1$$

[עבור כל i אי זוגי: $Q_i = \forall$, ועבור כל i זוגי: $Q_i = \exists$].

PH

$PH = \bigcup_{i=1}^{\infty} \sum_i^p$ או: $PH = \bigcup_{i=1}^{\infty} \prod_i^p$

<u>אבחנות:</u> 1) $NP = \sum_1^p$

2) $coNP = \prod_1^p$

3) $\prod_i^p = co\sum_i^p = \{L \mid \overline{L} \in \sum_i^p\}$

4) $\prod_i^p \subseteq \sum_{i+1}^p$

5) $\sum_i^p \subseteq \prod_{i+1}^p$

6) $PH \subseteq PSPACE$

- מאמינים כי $\sum_i^p \subset \sum_{i+1}^p$

<u>הגדרה:</u> קריסת PH

נאמר ש- PH קורסת ל-i אם מתקיים $PH = \sum_i^p$, או במילים אחרות: $\sum_i^p = \bigcup_{j\geq i} \sum_j^p$

<u>משפט:</u>

1. לכל $i \geq 1$, אם מתקיים $\prod_i^p = \sum_i^p$, אזי PH קורסת ל-i.
2. אם $P = NP$, אזי $PH = P$.

- הטענה הראשונה עבור $i = 1$ אומרת: אם $NP = coNP$, אז $PH = NP$.
  מוכיחים ע"י אינדוקציה על i שמראה $NP \subseteq \sum_i^p$ וגם $coNP \subseteq \prod_i^p$

<u>הראינו בכיתה:</u> השפה $\sum_i SAT$

$$\sum_i^p - Comp \ni \sum_i SAT = \{\psi = \exists u_1 \forall u_2 ... Q_i u_i\ \phi(u_1, ..., u_i) \mid \psi = T,\ u_1, ..., u_i\ are\ boolean\ vectors,$$
$$\phi\ is\ a\ CNF\ formula\}$$

<u>משפט:</u> אם קיימת שפה $L \subseteq \{0, 1\}^*$ שהיא שלמה ב-PH, אז קיים $i \in N$ כך שמתקיים $PH = \sum_i^p$ (PH קורסת לרמה ה-i).

<u>משפט:</u> לכל $i \geq 2$ מתקיים: $\sum_i^p = NP^{\sum_i SAT}$

- אפשר להרחיב את זה להגדרה חלופית ל-PH בעזרת אורקלים.

(6) מעגלים בוליאנים

הגדרה: מעגל בוליאני

לכל $n \in N$, מעגל בוליאני לקלטים $x \in \{0,1\}^n$ עם פלט יחיד, מיוצג ע"י גרף מכוון ללא מעגלים (DAG), באשר **הקלטים** הם $x_1, ..., x_n$ המיוצגים ע"י קודקודים עם דרגת כניסה 0 ודרגת יציאה לא חסומה.

הפלט מיוצג ע"י קודקוד עם דרגת יציאה 0.

כל קודקוד אחר בגרף הוא מאחת מ3 אפשרויות:

1. קודקוד AND: מייצג שער $\wedge$, דרגת כניסה 2, דרגת יציאה $\supseteq \{1, 2\}$.
2. קודקוד OR: מייצג שער $\vee$, דרגת כניסה 2, דרגת יציאה $\supseteq \{1, 2\}$.
3. קודקוד NOT: מייצג שער $\neg$, דרגת כניסה 1, דרגת יציאה $\supseteq \{1, 2\}$.

גודל המעגל יסומן ב-$|C|$, והוא מוגדר כמספר השערים בו.

הפלט של C על קלט x יסומן ב-$C(x)$, והוא מחושב ע"פ כללי הסמנטיקה של לוגיקה בוליאנית.

- הערה 1: נוסחאות בוליאניות הן למעשה מקרה פרטי של מעגלים בוליאנים עם דרגת יציאה 1 לכל קודקוד פנימי.
- הערה 2: מעגלים בוליאנים (וגם נוסחאות בוליאניות) נקראים מודל $Non - Uniform$, בניגוד למ"ט שהיא מודל $Uniform$. ההבדל הוא שמ"ט היא אובייקט יחיד בעל ייצוג סופי המטפל בקלטים מכל אורך, ומעגל/ נוסחה מטפלים רק בקלטים מאורך מסוים - n. (הם משפחה של אובייקטים, עם ייצוג אינסופי, שכל אובייקט מתאים לקלטים מאורך מסוים).

הערה: ייצוג מעגל עם k שערים

נדרשת מטריצת שכנויות בגודל $k \times k$, וגם מקום לקלטים. בנוסף, נדרש מערך של $k \cdot c$ ביטים ($c = 2$), שמספר מהי הפעולה הלוגית של השער ה-i במעגל.

הגדרה: סדרת/ משפחת מעגלים, זיהוי שפה

תהא $T: N \rightarrow N$ פונקציה. סדרת מעגלים מגודל $T(n)$ היא סדרה $\{C_n\}_{n \in N}$ של מעגלים בוליאנים, כאשר לכל n מתקיים ש-C_n הוא מעגל בוליאני עבור קלטים באורך n, ו-$|C_n| \leq T(n)$.

$SIZE(T(n))$

נאמר ששפה L היא ב-$SIZE(T(n))$ אם קיימת סדרת מעגלים $\{C_n\}_{n \in N}$ כך ש-C_n סדרת מעגלים מגודל $O(T(n))$, וגם לכל $x \in \{0,1\}^n$ מתקיים: $C_n(x) = 1 \Leftrightarrow x \in L$.

$P_{/poly}$

אוסף כל השפות L הניתנות להכרעה ע"י סדרת מעגלים $\{C_n\}_{n \in N}$ בגודל פולינומי.

$$P_{/poly} = \bigcup_{i \in N} SIZE(n^i)$$

טענה: $P_{/poly}$ לא מוכל ב-NP.

משפט: $P \subseteq P_{/poly}$ (כל מ"ט דטרמיניסטית פולינומית M, ניתנת לייצוג ע"י סדרת מעגלים $\{C_n\}_{n \in N}$ בגודל פולינומי, כך שלכל $x \in \{0,1\}^n$: $C_n(x) = M(x) = f_L(x)$).

הראינו בכיתה: השפה $CKT - True$

$P \ni CKT - True = \{< C_n, x > \mid x \in \{0,1\}^n,\ C_n \text{ is a boolean circle with n inputs},\ C_n(x) = 1\}$

- למעשה $CKT - True$ היא P-שלמה, כאשר הגדרת שלמות ב-P היא על בסיס רדוקציות בזכרון לוגריתמי מובלע.

השפה $CKT - SAT$:

$NPC \ni CKT - SAT = \{< C_n > \mid \exists\, x \in \{0,1\}^n,\ C_n \text{ is a boolean circle with n inputs},\ C_n(x) = 1\}$

<u>טענה:</u> בהינתן מ"ט דטרמיניסטית M העובדת בזמן $T(n)$, קיימת משפחת מעגלים $\{C_n\}_{n\in N}$ בגודל $O(T(n)^2)$, כך שלכל $x \in \{0,1\}^n$: $M(x) = C_n(x)$.

- ההוכחה היא בדומה למשפט קוק לוין - בונים מעגל המסמלץ את ריצת מ"ט ה-$oblivious$ של M.
- לנוסחה יש משתנים נוספים מלבד המשתנים המקוריים, וכאן אנו רוצים להשתמש רק ב-$x_1,\ldots,\ x_n$.

<u>הגדרה:</u> $P - uniform\ circuit\ families$

משפחת מעגלים $\{C_n\}_{n\in N}$ תיקרא P-יוניפורמית אם קיימת מ"ט דטרמיניסטית פולינומית, אשר בהינתן קלט 1^n מייצרת את המעגל C_n. (ייצוג סופי לסדרה אינסופית).

נסמן ב-χ את אוסף השפות L שיש עבורן משפחת מעגלים P-יוניפורמית המזהה אותן.

<u>טענה:</u> $\chi \subseteq P_{/poly}$ (מכיוון שאם יש משפחת מעגלים שניתן לייצר אותה בזמן פולינומי, אז הגודל של כל מעגל הוא פולינומי, ולכן יש סדרת מעגלים פולינומית).

<u>משפט:</u> $P = \chi$

<u>משפט:</u> לכל $n > 1$ קיימת פונקציה $f: \{0,1\}^n \to \{0,1\}$ שאינה ניתנת לחישוב/ הכרעה ע"י מעגל בגודל $\ge \frac{2^n}{10n}$.

מקביליות בחישוב

NC

עבור $d \in N$, שפה L הינה ב-NC^d אם L ניתנת להכרעה ע"י סדרת מעגלים $\{C_n\}_{n\in N}$, כאשר C_n בגודל פולינומי ב-n, ובעומק $O(log^d n)$.

$$NC = \bigcup_{j\in N} NC^j$$

AC

עבור $d \in N$, שפה L הינה ב-AC^d אם L ניתנת להכרעה ע"י סדרת מעגלים בגודל פולינומי, ובעומק $O(log^d n)$, אולם מותרת כל דרגת כניסה לשערים במעגל (AND ו-OR אינן בהכרח דו מקומיות).

$$AC = \bigcup_{j\in N} AC^j$$

- מעגל אריתמטי/ אלגברי:
רעיון מקביל למעגל לוגי, אלא שהשערים הם שערים של פעולות אלגבריות (כמו כפל, חיבור, חיסור).

(7) חישובים אקראיים

<u>למה:</u> $Schwartz - Zipple$

יהא פולינום מרובה משתנים $P: Z^n \rightarrow Z$, מדרגה לכל היותר d, שאינו פולינום ה-0. אזי לכל קבוצה סופית S של שלמים, אם נבחר n איברים $a_1, a_2, ..., a_n \in S$ באקראי, באופן ב"ת (עם חזרות), ובהתפלגות אחידה, אזי:

$Pr[P(a_1, ..., a_n) \neq 0] \geq 1 - \frac{d}{|S|}$

<u>הגדרה:</u> מ"ט מטילת מטבעות

מ"ט עם שתי פונקציות מעברים δ_0, δ_1. בהינתן קלט x, חישוב M על x מתבצע באופן הרגיל, מלבד העובדה שהבחירה בכל שלב האם להשתמש ב-δ_0 או ב-δ_1 מתבצעת באקראי - בהסתברות 0.5 (ב"ת במה שקרה עד כה).

- <u>הגדרה שקולה:</u>

מ"ט דטרמיניסטית עם קלט נוסף שעליו כתובה מחרוזת אינסופית של אפסים ואחדות, כאשר בכל תא i בסרט הקלט המיוחד - כל ביט נבחר בהסתברות ב"ת אחידה.

טעות חד צדדית

$RTIME$

תהא $T: N \rightarrow N$. המחלקה $RTIME(T(n))$ היא אוסף כל השפות L כך שקיימת עבורן מ"ט מטילת מטבעות, M_L, הרצה בזמן $O(T(n))$ ומקיימת לכל $x \in \{0,1\}^*$:

$x \in L \Rightarrow Pr[M_L(x) = 1] \geq \frac{2}{3}$ (צודקת בהסתברות $\frac{2}{3}$).

$x \notin L \Rightarrow Pr[M_L(x) = 1] = 0$ (צודקת בהסתברות 1).

RP

$$RP = \bigcup_{i \in N} RTIME(n^i)$$

<u>טענה:</u> $RP \subseteq NP$

$coRP$

$coRP = \{L | \overline{L} \in RP\}$

$coRP = \{L | \exists PPTM, M_L, s.t.\ \ x \in L \Rightarrow Pr[M_L(x) = 1] = 1 \ \wedge\ x \notin L \Rightarrow Pr[M_L(x) = 1] \leq \frac{1}{3}\}$

<u>ראינו בכיתה:</u> השפה $ZeroP$

$ZeroP \in coRP$ (הקלט הוא מעגל, והוא בשפה רק אם הוא מייצג את פולינום ה-0).

<u>טענה:</u>

לכל קבוע $c > 0$, אם קיימת מ"ט M עבור שפה L כך שמתקיים לכל $x \in \{0,1\}^*$:

$x \in L \Rightarrow Pr[M(x) = 1] = 1$, ו- $x \notin L \Rightarrow Pr[M(x) = 1] \leq 1 - \frac{1}{|x|^c}$ (טעות בהסתברות גדולה),

אזי קיימת מ"ט $\hat{M}$ שמקיימת את אותם תנאים עבור $c' > 0$, רק ש: $x \notin L \Rightarrow Pr[\hat{M}(x) = 1] \leq \frac{1}{c'|x|^c}$.

- כדי להגדיל את ההסתברות לתשובה נכונה, פשוט נעשה הרבה חזרות של אותו ניסוי עם אקראיות חדשה (ב"ת), ונקבל אמ"ם בכל הפעמים קיבלנו 1.
- אם חזרנו k פעמים על התהליך: $x \notin L \Rightarrow Pr[\hat{M}(x) = 1] \leq (1 - \frac{1}{c'|x|^c})^k$.

<u>מסקנה:</u> כאשר יש טעות חד צדדית, קל להגדיל את ההסתברות להיות צודקים.

ניתן להוריד את הטעות של coRP לקטנה מ-2^{-n}

טעות דו צדדית

$BPTIME$

עבור פונקציה $T: N \rightarrow N$, נאמר ששפה $L \subseteq \{0, 1\}^*$ היא ב-$BPTIME(T(n))$ אם קיימת מ"ט מטילת מטבעות M_L הרצה בזמן $O(T(n))$ ומקיימת לכל $x \in \{0, 1\}^*$:

$Pr[M_L(x) = f_L(x)] \geq \frac{2}{3}$

BPP

$$BPP = \bigcup_{j \in N} BPTIME(n^j)$$

<u>ידוע:</u> $coRP, RP \subseteq BPP$ (לכל מ"ט עם טעות חד צדדית, קיימת מ"ט עם טעות דו צדדית).

$P \subseteq BPP$

$BPP \subseteq EXP$, $BPP \subset NEXP$

<u>לא ידוע:</u> האם $BPP \subseteq P, NP$

<u>טענה:</u> הקטנת הטעות עבור BPP

תהא L שפה, ו-M מ"ט מטילת מטבעות הרצה בזמן פולינומי, ומקיימת לכל $x \in \{0, 1\}^*$, עבור $c > 0$ קבוע $Pr[M(x) = f_L(x)] \geq \frac{1}{2} + |x|^{-c}$.

אזי לכל $d > 0$ קיימת מ"ט M_d מטילת מטבעות, הרצה בזמן פולינומי כך ש:

$Pr[M_d(x) = f_L(x)] \geq 1 - 2^{-|x|^d}$

- לאחר החישוב יצא $k = 16|x|^{2c+d}$ (מספר הפעמים שמריצים את M).

<u>טענה:</u> $BPP \subseteq P_{/poly}$

<u>טענת עזר:</u> תהי שפה $L \subseteq \{0, 1\}^*$. אם קיימת מ"ט מטילת מטבעות M_L כך שלכל $x \in \{0, 1\}^*$: $Pr[M_L(x) \neq f_L(x)] \leq 2^{-|x|-1}$, אזי קיימת מחרוזת רנדומית r_n $(n \in N)$ כך שלכל x: $M_L(x, r_n) = f_L(x)$.

<u>הגדרה:</u> מ"ט עם עצה

מ"ט M היא מ"ט עם עצה אם קיימת מחרוזת a_n כך שלכל קלט $x \in \{0, 1\}^n$: $M(x, a_n) = f_L(x)$.

<u>הגדרה שקולה:</u> ל-$P_{/Poly}$

מחלקת כל השפות שקיימת עבורן מ"ט דטרמיניסטית עם עצה, שמכריעה אותן ורצה בזמן פולינומי.

<u>מסקנה מהטענה + הגדרה:</u>
אם $L \in BPP$ אזי קיימת עבורה מ"ט פולינומית דטרמיניסטית עם עצה המכריעה אותה (העצה היא אותה מחרוזת טובה r_n).

<u>ידוע:</u> $BPP \subseteq PSPACE \subseteq EXP \subseteq R$

- סדרת מעגלים לא יוניפורמיים לא מוכלת ב-PSPACE.
- בכללי: אי אפשר לקחת סדרת מעגלים ולהכניס אותה למחלקה של מ"ט! כי מ"ט זה עולם יוניפורמי ומעגלים זה עולם לא יוניפורמי. ואז זה אומר שאפשר לפתור ע"י סדרת המעגלים את בעיית העצירה, וב-BPP לא ניתן לפתור אותה!

(8) הוכחות אינטראקטיביות

<u>הגדרה:</u> הוכחה אינטראקטיבית

הוכחה שבה המוכיח ($Prover$) והמוודא ($Verifier$) מריצים יחד פרוטוקול שבו P מנסה לשכנע את V בנכונות טענה. בסוף האינטראקציה, V מחליט האם לקבל או לדחות.

מערכת הוכחה כזו צריכה לקיים 2 דרישות:

- נאותות ($Soundness$): המוודא "אינו" משתכנע בטענות שקריות.
- שלמות ($Completeness$): המוכיח ההגון מסוגל לשכנע את V באמיתות טענות נכונות.

<u>פרוטוקול דטרמיניסטי להוכחה:</u>

ישנו קלט משותף, x, וטענה ששניהם יודעים.

ל-V יש פונקציה g, ול-P יש פונקציה f.

הם מתקשרים ביניהם בהודעות, כך שכל אחד בתורו מפעיל את הפונקציה שלו על הקלט (x), ועל ההודעות שהיו עד עכשיו ($m_1,..., m_{k-1}$) ושולח זאת לשני כהודעה m_k.

בסופו של דבר V מחליט האם לקבל או לדחות: $g(x, m_1,... m_k) \in \{0, 1\}$.

- נרצה שהמוודא ירוץ בזמן פולינומיאלי, כלומר ש-g תהיה חשיבה בזמן פולינומיאלי.

<u>הגדרה:</u> מערכת הוכחה אינטראקטיבית דטרמיניסטית

לשפה L קיימת מערכת הוכחה אינטראקטיבית דטרמיניסטית ב-k סיבובים אם קיימת מ"ט דטרמיניסטית V הרצה בזמן פולינומי באורך x (בכל שלב בפרוטוקול), ויכולה לקיים אינטראקציה ב-k סיבובים עם **כל** פונקציה P (עם כוח חישוב שאינו חסום) כך שמתקיים:

1. שלמות: $x \in L \Rightarrow \exists P: \{0, 1\}^* \to \{0, 1\}^* \, Out_V < P, V > (x) = 1$
2. נאותות: $x \notin L \Rightarrow \forall P: \{0, 1\}^* \to \{0, 1\}^* \, Out_V < P, V > (x) = 0$

dIP

מכילה את כל השפות $L \subseteq \{0, 1\}^*$ שקיימת עבורן מערכת הוכחה אינטראקטיבית דטרמיניסטית ב-k סיבובים, עבור k שחסום ע"י פולינום.

<u>למה:</u> $dIP = NP$

IP

עבור $1 < k \in N$ (יכול להיות פונקציה של אורך הקלט x), נאמר ששפה L שייכת ל-$IP[K]$ אם קיימת מערכת הוכחה אינטראקטיבית שבה המוודא הינו $PPTM$,V, שיכולה לקיים אינטראקציה ב-$K(n)$ סיבובים על קלט $x \in \{0, 1\}^n$ עם כל פונקציה $P: \{0, 1\}^* \to \{0, 1\}^*$ כך שמתקיים:

- שלמות: $x \in L \Rightarrow \exists P \, Pr[Out_V < P, V > (x) = 1] \geq \frac{2}{3}$
- נאותות: $x \notin L \Rightarrow \forall P \, Pr[Out_V < P, V > (x) = 1] \leq \frac{1}{3}$
  - ההסתברות נלקחת מעל הטלות המטבעות של V.

$IP = \bigcup_{i \in N} IP[n^i]$ (כלומר מערכת הוכחה במספר פולינומי של סיבובים).

<u>למה:</u> ההגדרה הנ"ל הייתה שקולה גם אם היינו דורשים שלמות בהסתברות $1 - \frac{1}{2^{n^s}}$, ונאותות $\frac{1}{2^{n^s}}$ עבור כל $s > 0$ קבוע.

אבחנה: $dIP \subseteq IP$ ע"פ ההגדרה, ומכיוון שהוכחנו $dIP = NP$, נובע כי גם $NP \subseteq IP$.

ראינו בכיתה: $GNI \in IP$
- ומכיוון שלא ידוע האם $GNI \in NP$, אז מניחים שלא רק $NP \subseteq IP$, אלא יותר מזה.

הגדרה: הוכחות באפס ידע $(Zero - Knowledge\ Proofs)$
[הרעיון: P משכנע את V בנכונות טענה מסוימת מבלי ש-V ילמד כלום! המשמעות היא שכל דבר ש-V יכול לעשות לאחר האינטראקציה עם P הוא יכל לעשות גם בלעדיון.]
הוכחות כאלו הן הוכחות אינטראקטיביות שיש בהן שלמות, נאותות ואפס-ידע.
- ידוע שלכל שפה ב-NP קיימת מערכת הוכחה כזו.

הגדרה: $Public\ Coin\ Interactive\ Proofs\ (AM)$
A $(Arthur)$ הוא ה-$Verifier$, מי שמנסה לוודא טענה ע"י הצבת אתגרים גלויים, ו-M $(Merlin)$ הוא ה-$Prover$, מי שמנסה לשכנע את A באמיתות טענה מסוימת.
שפה תהיה ב-$AM[k]$ אם יש מערכת הוכחה אינטראקטיבית עבודה, ב-k סיבובים, בה המוודא שולח תמיד מחרוזות אקראיות (שהגריל): $r_1,..., r_k$, ובסוף התהליך מחליט באופן דטרמיניסטי האם לקבל או לדחות.
- A אינו יכול להסתיר את המטבעות שהגריל, ו-M אינו יכול לנחש אותם לפני ש-A שלח אותם.
- ברור כי $AM[k] \subseteq IP[k]$, כיוון שמערכת AM היא גם IP.

$AM = AM[2]$

משפט:
לכל $k \in N$: $IP[k] \subseteq AM[k + 2]$ (כלומר מספיקים עוד שני סיבובים נוספים כדי להפוך מערכת הוכחה IP למערכת AM).
- מהסתכלות בעץ עבור $AM[k]$ ניתן לראות כי $IP \subseteq PSPACE$.

משפט: $IP = PSPACE$
- כחלק מההוכחה הראינו כי $coNP \subseteq IP$, ע"י כך שהראינו מערכת הוכחה עבור $\overline{3SAT}$.
- לרעיון הזה קוראים **אריתמטיזציה**, כלומר בהינתן נוסחה, תרגמנו אותה לאובייקט מהעולם האריתמטי - פולינום.
- הפולינום שיצא גדול מידי על מנת ש-V יחשב אותו בעצמו, לכן הוא צריך את עזרת P.
- ראינו כי $\Phi(z) \leq 3^m$ לכל השמה $z \in \{0, 1\}^n$, לכן $\sum_{z_1}\sum_{z_2}...\sum_{z_n}\Phi(z_1,..., z_n) \leq 2^n \cdot 3^m$, ולכן ניתן לבצע את כל החישובים מודולו $q \geq 2^n3^m$ ראשוני, כדי להקטין את החישובים החלקיים, כדי שנוכל לייצג אותם, ולפשט את האנליזה.

(9) אלגוריתמי קירוב וקושי קירוב

מדברים כאן על בעיות חיפוש, ולא על בעיות הכרעה (שפות).

<u>הגדרה:</u> אלגוריתם $\frac{1}{\rho}$-מקרב

בעיית מינימום:

עבור $0 \leq \rho \leq 1$, נאמר שאלגוריתם A הוא $\frac{1}{\rho}$-מקרב לבעיית מינימום מסוימת, אם לכל קלט עבורו הפתרון האופטימלי הוא בגודל k, A מחזיר פתרון בגודל $k' \leq \frac{1}{\rho} \cdot k$ (פתרון קצת יותר גדול, אבל לכל היותר ב-$\frac{1}{\rho}$, כאשר $\frac{1}{\rho}$ הוא מספר גדול מ-1).

בעיית מקסימום:

עבור $\rho \geq 1$, נאמר שאלגוריתם A הוא ρ-מקרב לבעיית מקסימום מסוימת, אם לכל קלט עבורו הפתרון האופטימלי הוא בגודל k, A מחזיר פתרון בגודל $k' \geq \frac{1}{\rho} \cdot k$ (פתרון קצת יותר קטן, אבל לכל היותר ב-$\frac{1}{\rho}$, כאשר $\frac{1}{\rho}$ הוא שבר, קטן מ-1).

<u>הראינו בכיתה:</u>

- אלגוריתם 2-מקרב עבור MVC (כיסוי קודקודים מינימלי בגרף).
- אלגוריתם 2-מקרב עבור $MAX-3SAT$ (מספר הפסוקיות המקסימלי שניתן לספק בפסוק, ע"י השמה כלשהי). האלגוריתם הוא חמדן, ומקרב כל בעיית $MAX-SAT$ (לאו דווקא $3SAT$).
- אלגוריתם קירוב אקראי עבור $MAX-3SAT$: השמה אקראית מספקת בתוחלת $\frac{7m}{8}$ פסוקיות (כאשר m= מספר הפסוקיות ב-ϕ). כלומר האלגוריתם הוא $\frac{7}{8}$- מקרב **בתוחלת** (כי $\frac{7m}{8} \leq \frac{7k}{8}$).
  - האלגוריתם הזה הוא לא מקרב, כי הוא לא תמיד נותן את הקירוב הזה, אבל כמעט תמיד.
  - ניתן לקבל אלגוריתם דטרמיניסטי שמקרב את $MAX-3SAT$ עד כדי $\frac{7}{8}$ (תמיד), ע"י שימור התוחלת ובחירה של השמה לכל משתנה.

<u>הגדרה:</u> קושי קירוב

בעיות שאי אפשר לקרב אותן עבור אף ρ.

[אפשר להוכיח שבעיה קשה לקירוב ע"י כך שנניח בשלילה שקיים לה אלגוריתם ρ-מקרב, ואז נראה שבעזרתו ניתן לפתור בעיה אחרת שידוע שהיא NP-קשה, וכך נגיע לסתירה.]

- דוגמה: בעיית הסוכן הנוסע (TSP) היא קשה לקירוב, הראינו באמצעות בעיית מעגל המילטוני.

<u>הגדרה:</u> $Promise\ Problem$

ניתנת בעיית חיפוש/ אופטימיזציה/ הכרעה, אבל אנחנו צריכים להצליח רק על תת קבוצה של הקלטים האפשריים (ועל כל קלט שאינו בקבוצה הזו אין חשיבות לתשובה).

- ניתן לקרב את TSP כאשר "ההבטחה" היא שהגרף מקיים את א"ש המשולש (שזה מה שבאמת קורה בעולם האוקלידי, ולכן זה מה שרלוונטי על מנת לפתור את הבעיה בעולם האמיתי).
- הרדוקציה שהראינו קודם כדי להוכיח שלא קיים קירוב לבעיה, יכולה ליצור גרף שלא מקיים את א"ש המשולש. עכשיו כבר לא נוכל להשתמש בה.

<u>הראינו בכיתה:</u>

- אלגוריתם 2-מקרב לבעיית ה-TSP על גרפים המשמרים את א"ש המשולש (בעזרת עץ פורש מינימלי, מעגל "אוילר" וקיצורי דרך).

- הראינו שיפור לאלגוריתם: אלגוריתם 1.5-מקרב ל-TSP עם א"ש המשולש (ע"י הוספת צלעות לעץ הפורש, כך שנקבל גרף עם דרגה זוגית לכל קודקוד - בגרף כזה קיים מעגל אוילר, ע"י בחירת שידוך מלא במשקל מינימלי על קבוצת הקודקודים שב-T דרגתם אי זוגית).

<u>משפט:</u> משפט ה-PCP

אם $P \neq NP$, אזי אין אלגוריתם פולינומי אשר מקרב את $MAX - 3SAT$ בקירוב $\frac{8}{7} - \epsilon$ עבור $\epsilon > 0$ כלשהו

<u>הערה:</u> לא ניתן להשתמש בקושי קירוב של בעיה אחת כדי להראות קושי קירוב של בעיה אחרת, כי ישנן בעיות שקשה לקרב ב-ρ מסוים, אבל אחרות שיש עבורן קירוב כזה.

<u>הגדרה:</u> רדוקציה משמרת פער

נאמר שעבור בעיות p_1, p_2 קיימת רדוקציה משמרת פער מ-p_1 ל-p_2 אם קיימת פונקציה f חשיבה פולינומית, כך שלכל $x \in \{0,1\}^*$, בהינתן פתרון ρ-מקרב עבור $f(x)$ (שאילתה ל-p_2), ניתן להגדיר פתרון ρ-מקרב עבור x (שאילתה ל-p_1).

<u>הערה:</u> לא כל הרדוקציות הפולינומיות משמרות פער. רדוקציית קוק לוין למשל אינה משמרת פער, מכיוון שראינו שאפשר לקרב אותה עד כדי מספר קבוע של פסוקיות. היא אינה משמרת קושי קירוב.

<u>הראינו בכיתה:</u> הרדוקציה מ-$3SAT$ ל-$INDSET$ היא משמרת פער.

- מסקנה: אם קיים אלגוריתם A ρ-מקרב ל-$MAX - IS$, אזי בהינתן שאילתה ϕ עבור $MAX - 3SAT$, נוכל לחשב את G_ϕ, להריץ את A כדי לקבל קירוב, ואז לפי האמ"ם של הרדוקציה קיבלנו גם קירוב ל-ϕ. זאת משום שלפי הרדוקציה, ניתן לספק k פסוקיות ב-ϕ אמ"ם יש קבוצה ב"ת בגודל k ב-G_ϕ.
- מסקנה נוספת: לפי משפט ה-PCP, ולפי העובדה שהרדוקציה משמרת פער, ניתן להסיק כי אין קירוב של $\frac{8}{7+\epsilon}$ ל-$MAX - IS$.

<u>הגדרה:</u> רדוקציות מרחיבות פער

רדוקציה שבעזרתה ניתן לשפר את הקירוב של בעיה מסוימת.
לדוגמה, אם יש בעיית מקסימום, אז ככל ש-ρ קרוב יותר ל-1 הוא יותר טוב, אז בעזרת רדוקציה מרחיבת פער, נוכל למצוא קירוב ρ' שהוא קרוב יותר ל-1.

<u>טענה:</u> אם $P \neq NP$, אז לא קיים קירוב ρ ל-$MAX - IS$ עבור אף $\rho > 1$.

[מכיוון שהראינו שאם קיים קירוב כזה, אזי לכל קבוע k קיים קירוב $\sqrt[k]{\rho} = \rho^{\frac{1}{k}}$, וזו תהיה סתירה ל-$PCP$].

<u>הגדרה:</u> $PCP - Verifier$

תהא L שפה, ויהיו פונקציות $q, r: N \rightarrow N$.
ל-L יש מוודא-PCP-$(r(n), q(n))$ אם קיים אלגוריתם PPT, V, כך שמתקיים:

1. יעילות $(Efficiency)$: על קלט $x \in \{0,1\}^n$, ניתן ל-V גישת RAM להוכחה π כך ש: $|\pi| \leq q(n) \cdot 2^{r(n)}$. V רשאי להשתמש ב-$r(n)$ מטבעות אקראיים, ולשאול $q(n)$ שאילתות (גישות ל-π) לכל היותר. נסמן: $V^\pi(x)$= תשובתו של V על x עם הוכחה π.
2. שלמות $(Completeness)$: אם $x \in L$, אז קיים $\pi \in \{0,1\}^*$ כך ש: $Pr[V^\pi(x) = 1] = 1$.
3. נאותות $(Soundness)$: אם $x \notin L$, אז לכל $\pi \in \{0,1\}^*$: $Pr[V^\pi(x) = 1] \leq \frac{1}{2}$.

$PCP(r,q)$

נאמר ש-L היא ב-$PCP(r,q)$ אם קיים מוודא שהוא $PCP\ -\ (O(r(n)), O(q(n))$ עבור L.

<u>משפט:</u> משפט ה-PCP

עבור קלט באורך n, מתקיים: $NP\ =\ PCP(log(n), 1)$