סיכום חישוביות

17 בינואר 2022

1 שבוע 1:

1.1 הרצאה 1:

- **אוטומט:** זהו גרף שמגדיר סדרת פקודות חוקיות. יש לו מצב התחלתי, מצב מקבל (סיום), וא"ב - אותיות שכתובות על הקשתות.

- **הגדרות:**
  א"ב: יסומן באות $\sum$ והוא קבוצה סופית של אותיות.
  מילה: סדרה סופית של אותיות.
  המילה הריקה: מילה ללא אותיות תסומן ב ε
  $\sum^*$**:** כל המילים הסופיות מעל ה א"ב $\sum$
  שפה - L**:** קבוצה של מילים. למעשה היא תת קבוצה של $\sum^*$.

- **הגדרה - אוטומט דטרמיניסטי** DFA**:** האוטומטים אותם נגדיר הם דטרמיניסטים (אנו יודעים בדיוק לאן כל פעולה תביא אותנו בהינתן מצב מסויים). ומסומן כך: $A =< Q, \sum, \delta, q_0, F >$
  Q: קבוצה סופית של מצבים - קדקודי הגרף.
  $\sum$: א"ב.
  δ: פונקציית מעברים ממפה מצב ואות למצב הבא - $\delta : Q \times \sum \Rightarrow Q$. q_0: מצב התחלתי.
  $F \subseteq Q$: קבוצת מצבים מקבלים (סופיים).

- **הגדרה - ריצה:** בהינתן מילה $W = w_1 \cdot w_2 \cdot .... \cdot w_n \in \Sigma^*$ ריצה של האוטומט A על W היא סדרה $R = q_0, \cdot q \cdot ... \cdot q_n$ (יש $n+1$ מצבים) כך ש R מתחילה במצב ההתחלתי q_0. וגם לכל $0 \leq i < n$ מתקיים: $q_{i+1} = \delta(q_i, w_{i+1})$.
  הגדרה - ריצה מקבלת: R היא ריצה מקבלת אם $q_n \in F$.
  הגדרה - אוטומט מקבל: נאמר שהאוטומט A מקבל את W אם הריצה של A על W מקבלת.
  הגדרה - השפה של A**:** $L(A) = \{w : A \ accept \ w\}$.

- **הגדרה - שפה רגולרית:** שפה $L \subseteq \Sigma^*$ היא רגולרית אם קיים אוטומט A כך ש $A(L) = L$ (האוטומט מזהה אותה).

1.2 הרצאה 2 - תכונות סגור של שפות רגולריות:

- **איחוד:** $L_1 \cup L_2 = \{w \in L_2 \vee w \in L_1\}$.
- **חיתוך:** $L_1 \cap L_2 = \{w \in L_2 \wedge w \in L_1\}$
- **השלמה:** $L^- = \Sigma^* - L$
- **שרשור:** $L_1 \cdot L_2 = \{w_1 \cdot w_2 : w_1 \in L_1, w_2 \in L_2\}$
- **כוכב:** $L^* = \{w_1 \cdot w_2 \cdot ... \cdot w_k : k > 0, w_i \in L \ \forall 1 \leq i < k\}$
- **הערה:** L^* סופית, $L^* = \{\varepsilon\}$ עבור: $L = \emptyset$ ועבור $L = \{\varepsilon\}$. אחרת היא אינסופית.
- **משפט - רגולריות האיחוד:** עבור שפות L_1, L_2 רגולריות, אזי $L_1 \cup L_2$ רגולרית.
- **משפט - רגולריות החיתוך:** עבור שפהL_1 רגולרית, אזי $L_1 \cap L_2$ רגולרית.
- **משפט - רגולריות ההשלמה:** עבור שפות L_1, L_2 רגולריות, אזי אותו אוטומט עם $F^- = Q \backslash F$ רגולרית.
- **אוטומט המכפלה:** עבור שני אוטומטים A_1, A_2אוטומט המכפלה הינו מכפלה קרטזית של מצבי שני האוטומטים.

1.3 תרגול 1:

- **למה:** $A \backslash B = A \cap B^-$.
- **יחס שקילות:** R הוא יחס מעל קבוצה A אם: $R \subseteq A \times A$
- **יחס רפלקסיבי:** אם מתקיים $\forall a \in A : (a, a) \in R$ כלומר - A ביחס עם עצמה.
- **יחס סימטרי:** אם מתקיים $\forall a, b \in A : (a, b) \in R \Rightarrow (b, a) \in R$
- **יחס טרנזטיבי:** אם מתקיים $\forall a, b, c \in A : (a, b) \in R \wedge (b, c) \in R \Rightarrow (a, c) \in R$
- **יחס שקילות:** יחס המקיים רפלקסיביות, סימטריות וטרנזטביות נקרא יחס שקילות.
- **משפט:** יחס שקילות R משרה חלוקה של A. כלומר לכל שני אברים $a, b \in A$ או שהמחלקות שלהם זהות או זרות,
  כלומר - לא יתכן שהן שונות ויש בניהם חיתוך לא ריק.
  בנוסף - איחוד כל מחלקות השקילות מכסה את כל הקבוצה A.
- **גודל של קבוצות:** יהיו A, B קבוצות. נאמר ש $|A| \leq |B|$ אם קיימת פונק' $f : A \Rightarrow B$ חח"ע.
  נאמר ש $|A| < |B|$ אם קיימת פונק' $f : A \Rightarrow B$ חח"ע **ולא** על.
  נאמר ש $|A| = |B|$ אם קיימת פונק' $f : A \Rightarrow B$ חח"ע ועל.

- **עוצמה של קבוצות:** נגדיר את העוצמה של קבוצת הטבעיים להיות $\aleph_0$, קבוצת הטבעיים שווה בעצמתה לקבוצת השלמים $\mathbb{Z}$ והרציונלים $\mathbb{Q}$. כלומר ־ כל קבוצה שניתן לספור את אבריה (בת מניה) יש העתקה חח"ע ועל מהטבעים אליה ולכן היא נקראת **בת מניה.**

  עצמת הממשיים $\mathbb{R}$**:** נקראת $\aleph = 2^{\aleph_0}$.

- **הוכחה אוטומט ושפה:** בכדי להוכיח ששפה ואוטומט תואמים, צ"ל דו כיווניות. גם שהאוטומט מוציא מילים חוקיות בשפה, וגם שכל מילה חוקית לשפה יכולה להיות באוטומט.

- δ^*: הפונקציה : $\delta^*(S,w)=S^|$ היא פונקציה שמעבירה אותנו מהמצב S למצבים $S^|$ ע"י קריאת המילה w.

2 שבוע 2:

2.1 הרצאה 3 ־ אוטומטים אי דטרמיניסטים ־ NFA:

- **משפט ־ רגולריות הגרירה:** עבור שתי שפות רגולריות A_1, A_2 מתקיים: $A_1 \Rightarrow A_2 = \neg A_1 \vee A_2$ כלומר השפה L תהיה רגולרית:

$$L=\{w : if\ w \in L(A_1)\ so\ w \in L(A_2)\}$$

  נייצר את DFA (אוטומט דטרמיניסטי) באופן הבא:

  1: איחוד של $A_1^{\sim}$ ו A_2.

  2: אוטומט המכפלה עם $F=(Q_1 \cdot F_1) \times Q_2) \cup (Q_1 \times F_2)$

- **משפט ־ רגולריות השרשור:** עבור שתי שפות רגולריות L_1, L_2 שמקיימות : $L_1=\{w : |w| - 0 \bmod(2)\}, L_2=\{w :$ $|w| = 0\ mod(5)\}$

  אזי השרשור של L_1עם L_2 יהיה: כל המילים שאחת שייכת לשפה L_1 והשניה שייכת לשפה L_2.

- **הגדרה ־ אטומט אי דטרמיניסטי ־** NFA**:**

  $A=<Q, \sum, \delta, Q_0, F>$

  בדומה לאוטומט דטרמיניסטי, אך הוא שונה בהגדרות הבאות:

  $Q_0 \subseteq Q$: קבוצה של מצבים התחלתיים.

  δ: פונקציית מעברים ־ ממפה מצב ואות איחוד ε, לקבוצת מצבים ־ $\delta : Q \times (\sum \cup \{\varepsilon\}) \Rightarrow 2^Q$.

  במילים: האוטומט יכול לבחור האם ללכת למצב מקבל או לא. כלומר ־ אותה המילה יכולה להוביל למצב מקבל או לא.

  בנוסף ־ אפשר להוסיף צעד אפסילון, כלומר ־ אפשר לעבור ממצב למצב בלי אף אות, אלא עם האות אפסילון.

- **הגדרה ־ ריצה עם אוטומט אי דטרמיניסטי:** ריצה של A על מילה $|w|=n$ היא סדרת מצבים $R=R_0, R_1, ...R_m$ עבור $n \leq m$. כך ש:

  1: ניתן לכתוב את w כ ־ $y_1 \cdot y_2 .... \cdot y_m$ עבור $y_i \in \sum \cup \{\varepsilon\}$ (ניתן לשים ε בין שתי אותיות).

  2: $R_0 \in Q_0$

3: לכל $0 \le i < m$ מתקיים $R_{i+1} \in \delta(R_i, y_{i+1})$
נאמר ש A מקבלת את w - אם קיימת ריצה מקבלת ש A על w.

- **משפט:** לכל NFA (לא דט') יש DFA שקול (מזהה את אותה השפה).
בהינתן אוטומט לא דטרמיניסטי ניתן לבנות לו אוטומט דטרמיניסטי שקול. נעשה זאת על ידי בניית אוטומט דטרמיניסטי, ע"י יצירת קבוצת מצבים ומעברים דטרמניסטים בניהם.

- **טענה:** (להכחת המשפט לעיל) לכל $w \in \sum^*$ מתקיים: $\rho^*\left(Q_0, w\right) = \delta^*(q_0, w)$.

- **הגדרה רקורסיבית - ביטויים רגולריים:**
ביטוי רגולרי מעל א"ב Σ:
1: $a \in \Sigma$, ε, ו $\emptyset$ הם ביטויים רגולרים (ב"ר).
2: אם R_1, R_2 הם ב"ר אז כך גם: $R_1 \cup R_2$ וגם R_1^* וגם $R_1 \cdot R_2$הם ב"ר.
כל ביטוי רגולרי R מגדיר את השפה $L(R)$.

2.2 הרצאה 4:

- הוכחת הבניה של אוטומט דטרמניסטי מאוטומט אי דטרמניסטי.

2.3 תרגול 2:

- **משפט:** לכל $NFA - \varepsilon$ יש NFA שקול ללא מעברי ε. נעשה זאת ע"י הורדת מעברי ה ε וחיבור המעברים הישיגים ע"י מעבר ε למצב הרשון ע"י האות המתאימה.

- **הערה:** ניתן לקחת בעיית אוטומטים ולתרגם אותה לבעית גרף, להריץ עליה חיפוש עומק או רוחב ולבצע פעולות על האוטומט.

- **טענה:** לכל שתי שפות רגולריות מעל א"ב Σ, איחוד השפות הינו רגולרי ג"כ. ניתן להוכיח את הטענה ע"י בניית אוטומט אי דטרמניסטי NFA המזהה את האיחוד. (בהרצאה הוכחנו ע"י אוטומט המכפלה).

- **הערה:** כשמוכיחים ששפה היא רגולרית ע"י NFA צריך להוכיח שוויון על ידי הכלה דו"כ. גם שמילה בשפה נמצאת באוטומט וגם שהאוטומט מוציא לנו מילים בשפה.
אם הרכבנו אוטומט חדש משני אוטומטים אחרים אזי צריך להוכיח כי ריצה מקבלת על אוטומט המקור היא גם מקבלת באוטומט החדש. וכן להיפך - אם ריצה מקבלת באוטומט החדש אזי היא ריצה מקבלת על אוטומטי המקור (אחד מהם, או שניהם יחד).

3 שבוע 3:

3.1 הרצאה 5:

- **למת הניפוח:** אם L שפה רגולרית אזי יש $1 \leq p$ כך שלכל מילה $w \in L$ אם $|w| \geq p$ אזי יש חלוקה $w = xyz$ כך ש $|y| \geq 0$ וגם $|xy| \leq p$ וגם לכל $0 \leq i$ המילה $xy^iz \in L$.

- **הוכחה ששפה לא רגולרית:** נראה כי לכל $p \geq 1$ קיימת מילה $w \in L$ כך ש $|w| \geq p$ ולכל חלוקה של $w = xyz$ אם $|y| \geq 0$ וגם $|xy| \leq p$ אך קיים $i \geq 0$ כך ש $xy^iz \notin L$.
כלומר למת הניפוח לא מתקיימת ולכן השפה לא רגולרית.

- **אפיון של שפות רגולריות:** עבור שפה $L \in \Sigma^*$ נגדיר יחס $\sim_L \subseteq \Sigma^* \times \Sigma^*$, עבור $x, y \in \Sigma^*$ מתקיים ש $x \sim_L y$ אם לכל $z \in \Sigma^*$ מתקיים $yz \in L \iff xz \in L$ (כלומר ־ אם אין ל z, y זנב שמפריד אותן כך שמילה אחת שייכת לשפה והאחרת לא).

- **למה:** היחס $\sim_L$ הוא יחס שקילות.

- **משפט** $Myhill - Nored$: לכל שפה $L \subseteq \Sigma^*$ התנאים הבאים שקולים:
  1: L רגולרית.
  2: יש ליחס $\sim_L$ מספר סופי של מחלקות שקילות.
  בנוסף: גודל ה DFA המינימי שווה למספר מחלקות השקילות.

3.2 הרצאה 6:

- חזרה על הרצאה קודמת

3.3 תרגול 3:

- **ביטוי רגולרי ־ הגדרה:** r הוא ביטוי רגולרי מעל א"ב Σ אם הוא אחד מהבאים:
  1: $r = \emptyset$
  2: $r = \varepsilon$
  3: $r = a$ עבור $a \in \Sigma$
  4: $r = s \cup t$ כאשר s, t ביטויים רגולרים שונים מ t
  5: $r = s \cdot t$ כאשר s, t ביטויים רגולרים שונים מ t
  6: $r = t^*$ כאשר t ביטויי רגולרי שונה מ t.

- **הגדרה ־ שפה של ביטוי רגולרי:** $L(r)$ נגדיר באנדוקציה על המבנה של r.
  1: $L(\emptyset) = \emptyset$
  2: $L(\varepsilon) = \varepsilon$
  3: $L(a) = a$ עבור $a \in \Sigma$

4: $L(s \cup t) = L(s) \cup L(t)$ כאשר s, t ביטויים רגולרים שונים מ t
5: $L(s \cdot t) = L(s) \cdot L(t)$ כאשר s, t ביטויים רגולרים שונים מ t
6: $L(t^*) = L(t)^*$ כאשר t ביטויי רגולרי שונה מ t.

- **משפט:** L שפה רגולרית $\iff$ קיים ביטוי רגולרי r כך ש $L(r) = L$.

- **$GNFA$:** אוטומט כללי, המעברים שלו יכולים להכיל ביטויים רגולרים (ולא רק אותיות כמו NFA רגיל).
בנוסף נניח בה"כ שבכל $GNFA$:
1: יש מצב התחלתי יחיד q_{start} שאין קשתות שנכנסות אליו.
2: יש מצב מקבל יחיד q_{accept} שאין קשתות שיוצאות ממנו.
3: $q_{start} \neq q_{accept}$

כיצד נבנה: אם יש לאוטומט יותר ממצב התחלתי 1, ניצור מצב התחלתי חדש עם מעברי אפסילון למעברים ההתחלתיים המקוריים ונבטל אותם. אותו הדבר נעשה עם המצבים המקבלים.

- **אלגוריתם למעבר מאוטומט לביטוי רגולרי:** ניצור $GNFA$ ע"י מעברי אפסילון, כמו שכתוב למעלה. לאחר מכן נעבור על כל מצב $q_i \in Q$ שמקיים $q_i \neq q_{start} \vee q_{accept}$. ונבטל אותו. במקומו נשים מעבר ישיר לכל הקודקודים שעברו דרכו, ועל המעברים בניהם נכתוב את הביטוי הרגולרי הקיים $\cup$ הביטוי שהעביר אותנו לקודקוד המבוטל.
לבסוף נישאר עם שני מצבים $q_{start}, q\ \mathrm{accept}$.

4 שבוע 4:

4.1 הרצאה 7 - מזעור אוטומטים:

- **מזעור אוטומטים:** (שקול למציאת מחלקות שקילות של יחס) - מציאת DFA מינימלי לשפה.
אלגוריתם: נגדיר סדרת מחלקות שקילות על המצבים. נגדיר סדרה $(=_i) \subseteq Q \times Q$ לכל $i \in [\mathbb{N}]$ של יחסים מעל Q, ומתקיים $q =_i q'$ אמ"מ לכל מילה w אם $|w| \leq i$ אז $\delta^*(q, w) \in F \leftrightarrow \delta^*(q', \omega) \in F$.
כלומר - כל שני מצבים שיסכימו על מילים עד אורך מסויים, יהיו ביחס.

נגדיר:
$q =_0 q'$ אמ"מ: $q \in F \leftrightarrow q' \in F$. (לשקול 0 יש רק שתי מחלקות שקילות)
$q =_{i+1} q'$ אמ"מ: $q =_i q'$ וגם לכל אות $\sigma \in \Sigma$ מתקיים $\delta(q, \sigma) \leftrightarrow \delta(q, \sigma)$.

- **טענה:** לכל $i \geq 0$ $q =_i q'$ אמ"מ לכל w אם $|w| \leq i$ אז $\delta^*(q, w) \in F \leftrightarrow \delta^*(q', \omega) \in F$.
לסדרה $=_i$ יש נקודת שבת. קיים $i \leq |Q|$ כל ש $=_i$זהה ל $=_{i+1}$ משום שבכל איטרציה שאיננה נקודת שבת, מתפצלת לפחות מחלקה אחת.
נאמר ש $q =_a q'$ אמ"מ $q =_i q'$ עבור האיטרציה i שבה הגענו לנקודת שבת.

- **האוטומט המינימלי יהיה:**
Q' הן מחלקות השקילות של $=_a$.
$\delta([q],a) = [\delta(q,a)]$
F' תהיה $\{[q] \cdot q \in F\}$

4.1.1 שפות חסרות הקשר:

- **דקדוק חסר הקשר - הגדרה:** רביעה$G =< V, \Sigma, R, S >$
V: משתנים.
Σ: א"ב.
R: חוקי גזירה מהצורה - $V \Rightarrow (V \cup \Sigma)^*$.
$S \in V$: משתנה התחלתי.

- **שפות חסרות הקשר** CFL**:** שפה של דקדוק ח"ה.
1: אם $w, u, v \in (V \cup \Sigma)^*$ ו $A \Rightarrow w$ הוא חוק. אז נאמר ש uAv מייצר את uwv.
2: אם $u, v \in (V \cup \Sigma)^*$ אז $u \Rightarrow^* V$ אם יש סדרה $u = u_1 \Rightarrow u_2 \Rightarrow u_3 \Rightarrow ... \Rightarrow u_k = V$ עבור $1 \leq k$.
3: השפה של דקדוק G היא: $L(G) = \{w \in \Sigma^* : S \Rightarrow^* w\}$.

4.2 הרצאה 8:

- **שפת הסוגריים המקוננים באופן חוקי:** $S \Rightarrow aSb|SS|\varepsilon$ שהיא שפה לא רגולרית וחסרת הקשר.

- **משפט:** כל שפה רגולרית היא חסרת הקשר. $REG \subseteq CFL$
נוכיח בעזרת אוטומט של השפה הרגולרית שממנו נייצר שפה שתשווה לשפה ח"ה.

- **כיצד ניצור את הדקדוק:** יהי A אוטומט של שפה רגולרית. נגדיר דקדוק $G =< V, \Sigma, R, S >$ כך:

$$V = \{V_q : q \in Q\}, S = q_0$$

הרעיון הוא ש G ממשתנה V_qגוזר מילים שמתקבלות מהמצב q.
נגדיר את R: לכל מעבר המקיים $\delta(q,a) = q'$ נוסיף את הכלל $V_q \to aV_{q'}$
ואם $q \in F$ אז $V_q \Rightarrow \varepsilon$.
הערה: נקבל דקדוק מסוג דקדוק לינארי ימני (כל משתנה הולך לאות ומשתנה).

- **למת הניפוח לשפות ח"ה:** אם L חסרת קשר אזי קיים $p \geq 1$ כל שלכל מילה w אם $|w| \geq p$ אזי: יש חלוקה של $w = uvxyz$ עבורה $|vxy| \leq p$ וגם $|vy| > 0$ אזי $uv^ixy^iz \in L$ לכל $i \leq 0$.

- **אוטומט מחסנית:** אוטומט שמחזיק מחסנית וכך הוא יכול לבדוק האם האוטומט מקבל את המילה או לא.

4.3 תרגול 4:

- **טענה:** תהי $f:\mathbb{N}\Rightarrow\mathbb{N}$ פונקציה מונוטונית עולה ממש המקיימת $f=\omega(n)$. אזי השפה $L_f=\left\{a^{f(n)}|n\in N\right\}$ אינה רגולרית.

- **למה:** תהי $f:\mathbb{N}\Rightarrow\mathbb{N}$ פונקציה מונוטונית עולה ממש המקיימת $f=\omega(n)$. אזי סדרת ההפרשים אינסופית.

5 שבוע 5 - תורת החישוביות:

5.1 הרצאה 9 - מכונת טיורינג:

- **מכונת טיורינג:** מקבלת סרט המכיל מילים כך שבסוף כל מילה יש את התו רווח.

ההבדלים בין מכונת טיורינג לאוטומט:

1: הסרט אינסופי.

2: יכולה לכתוב על הסרט.

3: יכולה לזוז שמאלה וימינה.

4: מצבי קבלה ודחיה, כך אפשר דעת מתי לסיים את הריצה.

- **מכונת טיורינג - פורמלית:**

$$M=\{Q,\Sigma,\Gamma,\delta,q_0,q_{accept},q_{reject}\}$$

Q: קבוצת מצבים סופית.

$q_0\in Q$: מצב התחלתי.

Σ: א"ב הקלט. "רווח" לא שייך ל Σ

Γ: א"ב העבודה - אותיות שנכוב על הסרט במהלך העבודה, רווח שייך ל Γ.

$q_{accept}\in Q$: מצב מקבל.

$q_{reject}\in Q$: מצב דוחה.

δ: $\delta:Q\times\Gamma\Rightarrow Q\times\Gamma\times\{L,R\}$ (מציעה לנו: מצב,אות וכיוון).

- **קונפיגורציה של מ"ט:** מציגה - מצב, תוכן הסרט ואת מיקום הראש הקורא.

- **תיאור קונפיגורציה C:** עבור $v,u\in\Gamma^*$ ומצב $q\in Q$, אזי vqu היא קונפיגורציה שבה: המצב הוא q, תוכן הסרט הוא $v\cdot u$, והראש מצביע על האות הראשונה ב u.

- **הקונפיגורציה ההתחלתית:** של M על מילה $w\in\Sigma^*$ היא q_0w.

- **קונפיגורציות עוקבות:** יהיו $a,b,c\in\Gamma$, ו $v,u\in\Gamma^*$, ו $q,q'\in Q$. וקונפ' $uaqbv$. הקונפ' העוקבת נמצאת ב:$\delta(q,b)$. נחלק למקרים:

1: אם $q\in q_{exp}\vee q_{rej}$: אין קונפ' עוקבת, והריצה מסתיימת.

2: אם $\delta(q,b)=(q',c,L)$ אזי הקונפ' העוקבת היא: $uq'acv$.

3: אם $\delta(q,b)=(q',c,R)$ אזי הקונפ' העוקבת היא: $uacq'v$.

4: אם הקונפ, היא: qav ו $f(q,a)=(q',b,L)$ (כלומר ־ אנו עומדים בקצה והמצב הבא הוא ללכת שמאלה), אזי הקונפ העוקבת היא: $q'bv$ (דורכים במקום כדי לא ליפול).

- **ריצה סופית של מכונת טיורינג:** ריצה של M על מילה $w=w_0w_1...w_n\in\Sigma^*$, סדרה $R=C_0,C_1,....C_m$ של קונפיגורציות כך ש $C_0=q_0w$היא הקונפ' ההתחלתית של M על w, ולכל $0\le i<m$:C_{i+1} עוקבת ל C_i, ו C_m היא קונפ' עוצרת כלומר ־ המצב שלה הוא $q_{exp}\vee q_{rwj}$.

- **שפה של מכונת טיורינג:**

$$L(M)=\{w: there\ is\ an\ accepting\ run\ of\ M\ on\ w\}$$

- **נשים לב** שיש שלש אופציות לריצה של Mעל w :

1: עוצרת ומקבלת.

2: עוצרת ודוחה.

3: לא עוצרת. (לא בהכרח שיש קונפ' שחוזרת על עצמה, אך הכיוון ההפוך כן נכון).

- **זיהוי ־ הגדרה:** נאמר במכונת טיורינג M מזהה את השפה $L\subseteq\Sigma^*$ אם $L(M)=L$.
כלומר ־ אם היא מקבלת עבור כל מילה שבשפה, אך היא יכולה להיכנס ללולאה אינסופית על כל מילה שלא בשפה.

- **ניתנת למניה רקורסיביות** RE **־ הגדרה:** נאמר ששפה היא ניתנת למניה רקורסיביות (או ב RE), אם קיימת מ"ט M שמזהה את L.

- **הכרעה ־ הגדרה:** נאמר שמ"ט מכריעה את השפה L, אם M מזהה את L וגם M עוצרת על כל קלט .

- **רקורסיבית** R **־ הגדרה:** נאמר ששפה היא רקורסיבית (או ב R), אם קיימת מ"ט שמכריעה אותה. (יש מ"ט שמזהה ועוצרת תמיד).

- **הערה:** נשים לב כי $R\subseteq RE$, (משום שהכרעה חזקה יותר מזיהוי) אך ההפך לא נכון.

- **המחלקה המשלימה** $co-RE$ **הגדרה:**

$$L^-\in RE \quad\Leftrightarrow\quad L\in co-RE$$

כלומר יש מ"ט עבור L^- כזו שלכל $w\in\Sigma^*$: אם $\omega\notin L$ אז M עוצרת ומקבלת. אחרת ־ M דוחה או לא עוצרת.

5.2 הרצאה 10:

- **משפט:** $R = RE \cap co - RE$. כלומר ־ אם ניתן לזהות שפה ואת שלילתה אזי ניתן ליצור מ"ט שתכריע את השפה.

- **טענה:** אם $L \in R$ אזי גם $L^- \in R$ (ניתן ליצור מכונה M^- ע"י החלפת q_{exp} ו q_{rej} שתכריע את L^-).

- **מודל ספרן** $Enumerator$**:** מכונת טיורינג ללא קלט אך עם מדפסת. היא מדפיסה מילים ב Σ^*. השפה של ספרן E הן המילים שמודפסות בסופו של דבר. (יתכן שמילה תודפס אינסוף פעמים).

- **משפט:** $L \in RE$ אמ"מ יש ספרן E כך ש $L(E) = L$.

- **הערה:** אם מכונה היא רק מזהה (RE), **אי אפשר** להגדיר חוסר זיהוי. משום שמצב זה יכול להיכנס ללולאה אינסופית, ואין לנו את היכול לדעת מתי זה קורה.
הפתרון הוא: להריץ במקביל.

5.3 תרגול 5 ־ שפות ח"ה ומכונות טיורינג:

- **תכונות סגור של שפות חסרי הקשר:**
איחוד: יהיו L_1, L_2 שפות ח"ה, אזי השפה$L_1 \cup L_2$ ח"ה.
שרשור: יהיו L_1, L_2 שפות ח"ה, אזי השפה$L_1 \cdot L_2$ ח"ה.

ההוכחה נעשית באמצעות יצירת דקדוק חדש שהוא שילוב של שני הדקדוקים.

- **דקדוק בצורה נורמלית:** דקדוק G בצורה נורמלית של חומסקי, אם כל כלל ב G הוא אחד מהצורות הבאות:
1: $S \Rightarrow \varepsilon$.
2: $A \Rightarrow BC$ עבור $A, B \notin \{S\}$
3: $A \Rightarrow \sigma$ עבור $\sigma \in \Sigma$.

- **משפט:** ניתן להפוך כל דקדוק ח"ה לדקדוק בצורה נורמלית.
נעשה זאת כך:
1: נוסיף משתנה התחלתי חדש S_0, ואת הכלל $S_0 \Rightarrow S$.
2: נוריד כללים מהצורה $A \Rightarrow \varepsilon$ עבור $A \neq S$. (בכל מקום שמופיע A מצד ימין, נחליף אותה בכל מה שהיא יכולה לגזור. אם יופיע כלל חדש שלא הורדנו קודם והוא מהצורה $R \Rightarrow \varepsilon$ נוריד גם אותו).
3: נוריד כללים מהצורה $A \Rightarrow B$. (נחליף את B במה שהוא יכול לגזור).
4: נוריד כללים מהצורה $A \Rightarrow v_1, v_2 ... v_k$ עבור $3 \leq k$. (נייצר משתנים $u_2 u_3 ... u_{k-1}$ ואת הכללים $A \Rightarrow v_1, u_2$ ו $u_2 \Rightarrow v_2 u_3$ נמשיך כך באופן רקורסיבי עד הכלל $u_{k-1} \Rightarrow v_{k-1} v_k$).
5: נוריד אותיות. (עבור כל אות נוסיף משתנה חדש $X \Rightarrow \sigma$).

- **מודל שקול:** שתי מכונות טיורינג (לאו דווקא מאותו מודל) הן שקולות, אם הן מזהות את אותה השפה וגם עוצרות על אותן המילים.

- **טענה:** לכל מכונה עם סרט אחד, יש מכונה עם שני סרטים ששקולה לה, ולהיפך.

6 שבוע 6:

6.1 הרצאה 11:

- **התזה של צ'רץ' וטיורינג:** אלגוריתם = הכרעה ע"י מ"ט.
- **שלש רמות לתיאור אלגוריתם:**
  1: על ידי יצירת מכונת טיורינג $M = \langle\rangle$.
  2: על ידי תיאור הפעולה של מכונת טיורינג.
  3: פסואודו קוד בשפה עילית.
- **קידוד במכונת טיורינג:** נסמן ב $\langle A\rangle$ את הקידוד של האיבר A (מטריצה, פולינום וכו).
בהינתן קידוד של אובייקט מסויים, מ"ט יכולה להכריע עבורו (לדוגמה - להכריע אם גרף קשיר) האם הוא ב R.
- **עבודה עם קידוד:**
  1: ניתן לעבור על הקלט ולסמן קלטים שכבר עברנו עליהם (לדוגמה אם עברנו על קורקוד v נסמן אותו ב v_c)
  2: ניתן ליצור מכונה בעלת כמה סרטים, ולכתוב את הקודקוד הנוכחי עליו אנו עובדים בסרט הנוסף.
- **אי כריעות:**
טענה: יש שפה L שהיא לא כריעה.
טענה: מכונות טיורינג לא יכולות להכריע האם מ"ט אחרת היא כריעה. כלומר - השפה $A_{TM} = \{\langle M, w\rangle : M\ accept\ w\}$ לא שייכת ל R

6.2 הרצאה 12:

- **השפה** A_{TM}:

$$A_{TM} = \{\langle M, w\rangle : M\ accept\ w\}$$

טענה: $A_{TM} \in RE\backslash coRE$.

- **השפה** $\overline{A_{TM}}$:

$$\overline{A_{TM}} = \{\langle M, w\rangle : M\ not\ accept\ w\}$$

טענה $\overline{A_{TM}} \notin RE$: הוכחנו ש $A_{TM} \notin R$ ו $A_{TM} \in RE$ ולכן $A_{TM} \notin coRE$. א"כ קיבלנו ש $\overline{A_{TM}} \notin RE$.

- **נסתכל על שפה נוספת (בעיית העצירה):**

$$HALT_{TM} = \{\langle M, w\rangle : M\ stop\ on\ w\}$$

טענה: $HALT_{TM} \in RE$
טענה: $HALT_{TM} \notin R$, כי אם כן היינו יכולים לבנות מכונה שמכריעה את A_{TM} בסתירה, (רדוקציה).

- **הגדרה - פונקציה ניתנת לחישוב:** עבור א"ב Σ נאמר ש $f:\Sigma^*\Rightarrow\Sigma^*$ **ניתנת לחישוב** אם קיימת מ"ט M_f שבהינתן קלט $x\in\Sigma^*$ היא תמיד עוצרת עם $f(x)$ על הסרט. (לדוגמא מכונת טיורינג שמקבלת n ומחזירה $n+1$ עבור הפונקציה $f(n)=n+1$)

- **הגדרה - ניתנת לרדוקציית מיפוי:** עבור שתי שפות $A,B\subseteq\Sigma^*$ נאמר ש A **ניתנת לרדוקציית מיפוי** ל B ונסמן $A\leq_m B$ (כלומר - A קלה יותר מ B), אם קיימת פונקציה ניתנת לחישוב $f:\Sigma^*\Rightarrow\Sigma^*$ כך שלכל $x\in f$:
$x\in A\iff f(x)\in B$ (f ממפה קלטים מ A לקלטים צ B)
למעשה נוכל להגיד דברים מסויימים על B ולהשליך אותם על A.

6.3 תרגול 6:

- **טענה:** זמן ריצה של מכונה בעלת סרט 1, ומכונה בעלת k סרטים, שווה עד כדי פולינום (בשתי המכונות זמן הריצה יהיה פולנומי).

- **הרצה במקביל:** ניקח שתי מכונות או יותר, וניצור מכונה חדשה M', נקבע אינדקס רץ i, ונאמר שעבור כל i נריץ את שתי המכונות, אם אחת מהן קיבלה/דחתה - נפסיק. אחרת - נקדם את i.

- **תכונות סגור של RE:**
סגירות לאיחוד: יהיו $L_1,L_2\in RE$ אזי $L_1\cup L_2\in RE$ (ניתן ליצור מ"ט שמזהה את האיחוד, ע"י הרצה במקביל).
סגירות לשרשור: יהיו $L_1,L_2\in RE$ אזי $L_1\cdot L_2\in RE$ (ניתן ליצור מ"ט שמזהה את השרשור, ע"י הרצה במקביל על כל החלוקות האפשריות של w).

- **הגדרה - מכונה אוניברסלית:** U היא מכונה המסוגלת לקבל בתור קלט תיאור של מכונה M ומילה w (מעל א"ב של M), ולסמלץ את הריצה של M על w, כך ש U תקבל תדחה או לא תעצור אמ"מ M תקבל תדחה או לא תעצור על w.
כיצד ניצור מכונה אוניברסלית: ניצור מכונה עם כמה סרטים, כאשר סרט אחד מחזיק את הקידוד של M, סרט נוסף חזיק את w הנוכחית, וסרט נוסף יחזיק את המצב הנוכחי. בכל פעם נעבור על פונקציית המעברים ונמצא את המצב הבא לפי המילה והמצב הנתון.

7 שבוע 7:

7.1 הרצאה 13:

- **משפט הרדוקציה:** אם $A\leq_m B$ ו $B\in R$ אזי גם $A\in R$.
ואם $A\notin R$ אזי גם $B\notin R$.

- **שפה נוספת - $HALT_{TM}^{\varepsilon}$:**

$$HALT_{TM}^{\varepsilon}=\{\langle M\rangle : if\ M\ stop\ on\ \varepsilon\}$$

טענה: $HALT_{TM}^{\varepsilon}\notin R$ אך $HALT_{TM}^{\varepsilon}\in RE$.

- **שפה נוספת** REG_{TM}**:** מכונה המכריעה האם שפה של מכונת טיורינג אחרת היא רגולרית.

$$REG_{TM}=\{\langle M\rangle : L(M)\in REG\}$$

טענה: $REG_{TM}\notin R$, וגם $REG_{TM}\notin coRE$, וגם $REG_{TM}\notin RE$, אך $REG_{TM}\in\overline{RE\cup coRE}$

- **משפט הרדוקציה לשפות ב** RE**:** אם $A\leq_m B$ ו $B\in RE$ אזי גם $A\in RE$.
ואם $A\notin RE$ אזי גם $B\notin RE$.

- **משפט הרדוקציה לשפות ב** $coRE$**:** אם $A\leq_m B$ ו $B\in coRE$ אזי גם $A\in coRE$.
ואם $A\notin coRE$ אזי גם $B\notin coRE$.

- **משפט:** $A\leq_m B$ אמ"מ $A^-\leq_m B^-$.

- **משפט:** $A^-\leq_m B$ אמ"מ $A\leq_m B^-$

7.2 הרצאה 14:

- **שפה נוספת** INF_{TM}**:**

$$INF_{TM}=\{\langle M\rangle : L(n)\ is\ infinity\}$$

טענה: $INF_{TM}\notin RE$ וגם $INF_{TM}\notin coRE$.

7.3 תרגול 7:

- **כיצד נוכיח ברדוקציה:**

1: נתאר את הרדוקציה: בהינתן קלט $\langle M,w\rangle$ אזי קיימת מכונה M' שמקבלת קלט x ומבצעת עליו פעולות (או מתעלמת ממנו) בהתאם למה ש M עונה על w, כלומר נצטרך לתאר מה M' תעשה על x ומה היא תחזיר והאם תזהה או תכריע.

2: נוכיח שהיא נכונה: אם M מקבלת את w אזי $L(M')$ היא השפה שרצינו, ואם M לא מקבלת את w מתקבל ההפך.

3: נוכיח שהפונקציה ניתנת לחישוב: נבנה פונקציה ומכונה כך שתבצע את מה שאנו רוצים שיתקיים.

- **שפה נוספת** All_{TM}**:**

$$All_{TM}=\{\langle M\rangle : L\left(M\right)=\Sigma^*\}$$

טענה: $All_{TM}\notin coRE$ וגם $All_{TM}\notin RE$ ולכן $All_{TM}\notin R$. ומתקיים $All_{TM}\in\overline{RE\cup coRE}$

- **הגדרה - לולאה:** כאשר מכונה לא עוצרת ניתן לראות אם היא חוזרת על קונפיגורציה פעמיים וכך לזהות אם קיימת לולאה.

באופן פורמלי: אם עבור קונפיגורציות c_i,c_j כאשר $i\neq j$ מתקיים $c_i=c_j$ (וגם $c_{i+1}=c_{j+1}$) אזי קיימת לולאה והמכונה לא תיעצר.

- **שפה נוספת** $Repeat_{TM}$:

$$Repeat_{TM} = \{\langle M, w\rangle : ✡\}$$

עבור ✡: M חוזרת על קונפיגורציה בריצתה על w (ולכן M אינה עוצרת על).
טענה: $Repeat_{TM} \in RE$ אך $Repeat_{TM} \notin coRE$ ולכן $Repeat_{TM} \notin R$

- **כיצד ניצור מכונה שלא נכנסת ללולאה:** ניתן לאתחל מונה שהערך שלו תמיד יעלה ב 1. כך בכל קונפיגורציה הסרט ישתנה, ולכן לא נכנס ללולאה אף פעם.

- **שפה נוספת** $Usles_{TM}$:

$$Usles_{TM} = \{\langle M\rangle : ✡\}$$

עבור ✡: קיים מצב q_{usles} שהוא לא מקבל או דוחה, ועבור כל קלט w, M לא עוברת ב q_{usles}.
טענה: $Usles_{TM} \in coRE$ אך $Usles_{TM} \notin RE$

8 שבוע 8:

8.1 הרצאה 15:

- **בעיית הריצוף:** מקבלים כקלט קבוצה סופית T של אריחים, ותנאי שכנות במאוזן ומאונך $V, H \subseteq T \times T$ ואריח התחלתי $T_{init} \in T$.
האריחים הם ריבועים, אשר לכל אחת מצלעותיהם יש אפשרות להיצבע בצבע שונה. האוריינטציה המקורית של כל אריח נשמרת, ולא ניתן לסובב אריחים. הצבת האריחים בזה לצד זה אפשרית אך ורק אם לצלעותיהם המשיקות יש צבע זהה.
השאלה היא האם ניתן לרצף ריבוע $n\times n$ ריצוף חוקי (כשבפינה השמאלית למטה יש את T_{init} ויחסי השכנות נשמרים).

- **נגדיר את בעיית הריצוף כשפה:**

$$TILE = \{\langle T, V, H, T_{init}\rangle : ✡\}$$

עבור ✡: האם יש ריצוף חוקי $n \times n$ לכל $n \geq 1$.
כאשר ריצוף חוקי מוגדר כך:

$$f : \{1, n\} \times \{1, n\} \Rightarrow T$$

כך שמתקיים: $f(1, 1) = T_{init}$
וגם לכל $1 \leq j \leq n$ ו $1 \leq i < n$ מתקיים $H(f(i, j), f(i + 1, j))$
ולכל $1 \leq i \leq n$ ו $1 \leq j < n$ מתקיים $V(f(i, j), f(i, j + 1))$

- **נשים לב:** כי יש ריצוף חוקי $n \times n$ לכל n אמ"מ יש ריצוף חוקי (אינסופי) לרבע המישור החיובי.

- **הלמה של קניג:** בכל עץ אינסופי עם דרגת פיצול סופית לכל קדקוד, קיים מסלול אינסופי.

- **טענה:** $TILE \in coRE$.
נבדוק את כל הריצופים עבור $n = 1, 2, 3...$ אם נמצא ריצוף ל n הנוכחי - נתקדם עם n, אחרת - נדחה.
נראה ברדוקציה כי $TILE \notin RE$**:** ברדוקציה מ $\overline{HALT^{\varepsilon}_{TM}} \leq_m TILE$

- **בעיית ההתאמה של פוסט** PCP**:** מקבלים כקלט אוסף סופי של אבני דומינו e_i כאשר החלק העליון של האבן מסומן ב u_i והתחתון ב d_i כאשר $u_i, d_i \in \Sigma^*$.
נשאל: האם יש סדרה של אבנים כך שהמילה שתהיה כתובה למעלה תהיה שווה למילה שכתובה למטה.

- **טענה:** $PCP \in RE$. נעבור על כל האסדרות ואם קימצ התאמה נמצא אותה ונקבל.

- **טענה:** $PCP \notin R$ נוכיח ברדוקציה מ $A_{TM} \leq_m PCP$ או רדוקציה מ $\overline{TILE}$.

- **נגדיר את** PCP **כשפה:**

$$PCP = \{\langle e_1, e_2, ...e_m \rangle : there\ is\ a\ match\}$$

- **בעיית אוטומטים ממושקלים:** בהינתן אוטומט אי דטריניסטי ממושקל, השאלה האם ניתן למצוא לו אוטומט דטרמניסטי שקול **לא הוכרעה**.

8.2 הרצאה 16 - תורת הסיבוכיות:

- **אפיון של שפות כריעות:** נרצה לאפיין מה המשאבים (זמן ומקום) שידרשו בכדי להכריע את השפות.
ראינו כבר אפיון לפי $REG, CFL \subseteq R$.

- **סיבוכיות זמן:** נתכל על השפה $L = \{0^n 1^n : n \geq 0\}$, נשאל מה מספר הצעדים שהמכונה צריכה לעשות בכדי להכריע אם מילה $w \in L$.

- **הגדרות:**
חסם עליון: אלגוריתם.
חסם תחתון: נראה כי אי אפשר לפתור בדרך מהירה יותר.
חסם הדוק: כאשר חסם עליון = חסם תחתון.

- **מחלקת סיבוכיות זמן:** עבור פונקציה $t : \mathbb{N} \Rightarrow \mathbb{N}$ נגדיר את מחלקת הסיבוכיות

$$TIME(t(n)) = \{L : \davidsstar\}$$

עבור $\davidsstar$: כך ש L ניתנת להכרעה ע"י מ"ט **דטרמיניסטית** עם **סרט יחיד**, הרצה על מילה w באורך n לכל היותר $O(t(n))$ צעדים.

- **טענה:**

$$L=\{0^n1^n : n\geq 0\}\in TIME(n\cdot log(n))$$

- **משפט:** לכל מ"ט מרובת סרטים שעובדת בזמן $O\left(t(n)\right)$ יש מ"ט שקולה בעלת סרט יחיד שעובדת בזמן $O\left(t^2(n)\right)$.

- **משפט:** אם L ניתנת להכרעה בזמן $o(n\cdot log(n))$ אזי L רגולרית.

8.2.1 מ"ט אי דטרמניסטית:

- **הגדרה - מ"ט אי דטרמניסטית:** ההבדל היחיד הוא בפונקציית המעברים.

$$M=\langle Q,\Sigma,\Gamma,q_0,\delta,q_{acc},q_{rej}\rangle$$

δ**:** הפונקציה לא דטרמניסטית ולכן היא שולחת לקבוצת קונפיגורציות עוקבות. $\delta : Q\times\Gamma\Rightarrow 2^{Q\times\Gamma\times(R,L)}$.

- **עץ ריצות:** נתאר את ריצת המכונה על המילה באמצעות עץ, כאשר הקונפיגורציות העוקבות הן הבנים של כל קודקוד (פירוט פורמלי בתרגול).

- **הגדרת מ"ט מכריעה:** נאמר ש M אי דטרמניסטית מכריעה את L אם עבור כל מילה w, M עוצרת על **כל** הריצות של w.

- **הגדרת מ"ט מקבלת:** נאמר ש M אי דטרמניסטית מקבלת את w אם **קיימת** ריצה מקבלת של M עבור w.

- **השפה של מכונה M אי דטרמניסטית:** כל המילים $w\in\Sigma^*$ כך ש M מקבלת את w.

- **מחלקת סיבוכיות זמן למכונות אי דטרמניסטיות:**

$$NTIMETIME(t(n))=\{L : ✡\}$$

עבור ✡: כך ש L ניתנת להכרעה ע"י מ"ט **אי דטרמיניסטית** עם **סרט יחיד**, הרצה על מילה w באורך n לכל היותר $O(t(n))$ צעדים, **בכל** ריצותיה.

- **המחלקות** P,NP**:** אנו נתעניין במחלקות הבאות - P,NP (מ $PTIME,NPTIME$).
כאשר:
P**:** שפות שניתן להכריע אותם ע"י מ"ט **דטרמניסטית** שרצה בזמן פולינומיאלי

$$P=\cup_k TIME\left(n^k\right)$$

NP**:** שפות שניתן להכריע אותם ע"י מ"ט **אי דטרמניסטית** שרצה בזמן פולינומיאלי

$$NP=\cup_k NTIME\left(n^k\right)$$

- **מחלקת $EXPTIME$:** כל מה שניתן להכריע ע"י מ"ט דטרמיניסטית בזמן אקספוננציאלי:

$$EXPTIME = \cup_k TIME\left(2^{n^k}\right)$$

- **טענה:**

$$P \subseteq NP \subseteq EXPTIME$$

- **משפט:** אם L ניתנת להכרעה ע"י מ"ט אי דטרמניסטית בזמן $t(n)$, אזי L ניתנת להכרעה ע"י מ"ט דטרמניסטית בזמן $2^{O(t(n))}$.

8.3 תרגול 8:

- **הגדרות:**

תכונה: היא קבוצה או אוסף של מכונות טיורינג.

תכונה סמנטית: תכונה p היא סמנטית אם לכל שתי מכונות M_1, M_2 עם אותה השפה $L(M_1) = L(M_2)$ מתקיים ש $M_2 \in p \iff M_1 \in p$ ואם $M_2 \notin p \iff M_1 \notin p$.

תכונה לא טרוויאלית: תכונה p היא לא טרוויאלית אם יש מ"ט $M_1 \in p$ ו $M_2 \notin p$.

- **אבחנה:** לכל תכונה p מתקיים - p היא סמנטית לא טרוויאלית אמ"מ $\overline{p}$ היא סמנטית לא טרוויאלית.

- **משפט רייס:** תהי p תכונה סמנטית לא טרוויאלית, אזי השפה $L_p = \{\langle M\rangle : M \in P\}$ (שפת כל המכונות שמקיימות את התכונה p) אינה ב R.

- **למה:** תהי p תכונה סמנטית לא טרוויאלית עם $T\emptyset \notin p$ (מכונה עם שפה ריקה המקיימת $L(T\emptyset) = \emptyset$) אזי $A_{TM} \leq_m L_p$.

8.3.1 מכונות אי דטרמניסטיות:

- **הגדרה - עץ ריצות:** עבור כל מכונה M וקלט w - $T_{M,w} = \langle V, E\rangle$. נסמן ב C את קבוצת הקונפיגורציות של M.

קבוצת הקודקודים V מקיימת: $V \subseteq C \times \{\mathbb{N} \cup \{0\}\}$.

השורש של $T_{M,w}$ הוא: $\langle q_o, w, 0\rangle$.

קשתות העץ: כל קשת מצביעה לרמה הבאה ולכן $\cup_{i\geq 0} (C \times \{i\}) \times (C \times \{i+1\})$. עבור שתי קונפ' c, d יתקיים כי קשת היא מהצורה $E \supseteq (\langle c, i\rangle, \langle d, i+1\rangle) = e$, לכל i מתקיים כי e היא קשת אמ"מ d היא קונפ' עוקבת של c.

- **אבחנות:**

אבחנה 1: יש קבוע k (שאינו תלוי בקלטים של המכונה, אלא במכונה עצמה) שחוסם את מספר הקונפיגורציות העוקבות לכל קונפ' של M. (חוסם את דרגת הפיצול המקסימלית בעץ עבור כל קודקוד).

אבחנה 2: לכל w יש בעץ $T_{M,w}$ מספר סופי של קדקודים אמ"מ M מכריעה.

- **משפט:** לכל מ"ט אי דטרמניסטית N, קיימת מכונה דטרמניסטית D עם $L(N) = L(D)$, כך ש:
אם N מכריעה אזי D מכריעה.
לכל $w \notin L(N)$ - D עוצרת על w אמ"מ כל הריצות של N על w עוצרות. (אחרת - אם N לא מכריעה גם D לא בהכרח תעצור.)

- **טענה:** בהינתן מילה u מעל Σ_k, המכונה D יכולה לבדוק אם u היא כתובת חוקית של קונפ' מקבלת.

- **הערה:** ניתן ליצור מכונה אי דטרמניסטית שתנחש לנו מילים או מספרים, ע"י יצירת מצב חדש שינחש אות מסויימת, ובסוף ינחש אם להפסיק או להתקדם.

9 שבוע 9:

9.1 הרצאה 17 - המחלקה NP:

- **מציאת מסלול המילטון בגרף:** מסלול המילטון הוא מסלול שעובר בכל הקודקודים, ועובר בכל קדקוד פעם אחת בלבד. המטרה היא למצוא מסלול המילטון בין שני קודקודים.

- **השפה** $D-ST$-**HAMPAT H**:

$$D-ST\text{-HAMPAT H} = \{\langle G,s,t\rangle : \text{✡}\}$$

עבור ✡: G גרף מכוון, וקיים מסלול המילטון בגרף מ s ל t.

- **טענה:** $D-ST\text{-HAMPAT H} \in EXPTIME$.
ניתן לבנות מ"ט שמכריעה את השפה בזמן אקספוננציאלי, היא תעבור על כל הסדרות ב $|V|^n$ ואם יש סדרה המתחילה ב s ומסתיימת ב t, פרמוטציה של V - מקבלת, אחרת - דוחה.
נשים לב: קשה (אקספוננציאלי) לבדוק האם יש מסלול המילטון, אבל קל לבדוק שסדרה של קודקודים היא מסלול המילטון.

- **טענה:** $D-ST\text{-HAMPAT H} \in NP$.
ניתן לבנות מ"ט א"ד שרצה בזמן פולינומיאלי. היא תנחש סדרה $v_1...v_n$ של קדקודים ותדחה אם מתקיימים אחד או יותר מהתנאים הבאים:
1: $s \neq v_1$. **2:** $t \neq v_n$. **3:** יש קדקוד שחוזר על עצמו פעמיים. **4:** אין קשת בין v_i ל v_{i+1}.
אחרת - תקבל.

- **אפיון אלטרנטיבי של NP בעזרת מאמת:**
אינטואיציה: בעיות שקשה לפתור אותן (אין להן אלגוריתם פולינומיאלי) אך קל לאמת אותן.

הגדרה פורמלית: עבור שפה L, מוודא עבור L היא מ"ט דטרמניסטית V כך ש:

$$L = \{w : ✡\}$$

עבור ✡: קיים $c \in \Sigma^*$ כך ש V מקבלת את $\langle w, c\rangle$. c היא למעשה שכנוע לכך ש $w \in L$.

- השפה $COMPOSIME$:

$$COMPOSIME = \{x : p, q \neq 1, p, q, x \in \mathbb{N}, p \cdot q = x\}$$

הערה: אם x נתון בבינארית - האלגוריתם יעבור על כל המספרים עד $\sqrt{x}$, אורך הקלט הוא $log_2(x)$ ולכן האלגוריתם אקספוננציאלי. (החישוב תלוי בערך הקלט, ולא בגודל שלו).
טענה: $COMPOSIME \in EXPTIME$.
נשים לב: קשה להכריע האם x פריק, אך בהינתן p קל לבנות מאמת V שיבדוק האם p מקיים את הדרוש. עבור:

$$V = \{\langle x, p\rangle \quad x = 0mod(p)\}$$

- **סיבוכיות המוודא:** נמדדת ביחס למילה w.
מוודא פולינומיאלי רץ על הזוג $\langle w, c\rangle$ בזמן פולינומיאלי ב $|w|$. כלומר - אם קיים c פולינומאלי ב w, כך ש V מקבלת בזמן פולינומיאלי את $\langle w, c\rangle$.

- **משפט:** $L \in NP$ אמ"מ יש ל L מוודא פולינומיאלי.

- **נשים לב:** לא ברור שכל בעיה ב $EXPTIME$ היא ב NP.

- **הגדרה - פונקציה ניתנת לחישוב בזמן פולינומיאלי:** יש מ"ט דרמיניסטית M_f(שמחשבת את הרדוקציה) שעוצרת על קלט w אחרי $t(w)$ צעדים, עם $f(w)$ על הסרט ו $t(w)$ הוא פולינום.

- **הגדרה - רדוקציות פוליומיאליות:** $A \leq_p B$ אם יש פונקציה ניתנת לחישוב בזמן פולינומיאלי $f : \Sigma^* \Rightarrow \Sigma^*$, כך שלכל $w \in \Sigma^*$, מתקיים $f(w) \in B \iff w \in A$.

- **משפט הרדוקציה עם סיבוכיות:** אם $A \leq_p B$
אם $B \in P$ אזי $A \in P$.
אם $B \in NP$ אזי $A \in NP$.
אם $B \in coNP$ אזי $A \in coNP$.

- **הגדרה -** $NP - hard$**:** נאמר ששפה L היא $NP - hard$ אם לכל $k \in NP$ מתקיים $L \leq_p k$.

- **הגדרה-** $NP - complete$**:** שפה היא NP שלמה, אם:
1: $L \in NP$. (באופן שקול - ניתן להראות שיש מוודא פולינומי).
2 - אם L **היא** NP **קשה:** אם $L \in P$ אזי ינבע מזה ש $P = NP$.
או באופן שקול: אם לכל $L' \in NP$ מתקיים $L' \leq_p L$.
או באופן שקול: יש שפה L'' כך שהיא $NP - hard$ ו $L'' \leq_p L$.

9.2 הרצאה 18:

- **השפה ־ SAT:** סיפוק $(satisfied)$ של נוסחאות בתחשיב הפסוקים.

$$SAT = \{\langle\varphi\rangle : \varphi\ is\ satisfied\}$$

- **בעיית $3-SAT$:** ספיקות של נסחאות בתחשיב הפסוקים שנתונות בצורה נורמלית $3-CNF$.
  ליטרל ־ משתנה או שלילתו: $x, \sim x$.
  פסוקית: $\vee$ על ליטרלים.
  נוסחה ב CNF: $\wedge$ של פסוקיות.
  נוסחה ב 3־CNF: $\wedge$ של פסוקיות שכולן באורך 3.

- **משפט קוק לוין:** $SAT \in NP-hard$ ולכן גם $3-SAT \in NP-hard$

- **בעיית הקליקה:** נתון גרף $G = \langle V, E\rangle$ לא מכוון, ואנו רוצים לדעת אם יש קליקה בגרף. קליקה ־ קבוצת קדקודים שיש בין כולם קשתות.
  טענה: $CLIQUE \in NP$.

- **כיצד נעשה רדוקציה פולינומיאלית:**
  1: נמצא פונקציה f פולינומיאלית שתמיר לנו את הבעיה מ A ל B.
  2: נוכיח שהרדוקציה פולינומיאלית.
  3: נוכיח נכונות של הרדוקציה.

9.3 תרגול 9:

- **טענה:** אם יש שפה L שהיא $NP-complete$ וגם $L \in P$ אזי $P = NP$.

- **בעיית כיסוי קודקודים:** עבור גרף $G = \langle V, E\rangle$ כיסוי קדקודים היא תת קבוצה $C \subseteq V$ כך שלכל קשת $\{x, y\} \in E$ מתקיים $x \in C$ או $y \in C$.
  כלומר ־ אנו רוצים קבוצה מינימלית של קודקודים שנוגעת בכל קשתות הגרף.

$$VC = \{\langle G, k\rangle : \text{✡}\}$$

  עבור ✡: יש ב G כיסוי קדקודים בגודל k.
  טענה: בעיית כיסוי קדקודים היא $NP-complete$.

- **הגדרה ־ קבוצה שולטת:** יהי $G = \langle V, E\rangle$ גרף לא מכוון, קבוצה שולטת ב G היא תת קבוצה $C \subseteq V$ כך שלכל $v \in V$ מתקיים: $v \in C$ או קיים $u \in C$ כך ש $\{u, v\} \in E$.
  כלומר ־ כל קדקוד נמצא ב C או ששכן שלו נמצא ב C.

$$DS = \{\langle G, k\rangle : \text{✡}\}$$

עבור ✡: יש ב G קבוצה שולטת בגודל k.

טענה: $DS \in NP-complete$

10 שבוע 10:

10.1 הרצאה 19:

- **משפט קוק לוין:** אם $SAT \in P$ אזי $P = MP$.
כלומר $SAT \in NP-complete$ לכל $L \in NP$ מתקיים $L \leq_p SAT$. (למעשה המשפט מדבר על $3-SAT$ שממנו נובע המקרה הכללי SAT שהוא קל יותר)

- **הוכחה ששפה היא** $NP-hard$**:** עבור שפה L נוכל להוכיח כי היא $NP-hard$ על ידי $3-SAT \leq_p L$.

- **תרגום מטריצה ל** SAT**:** ניתן לתרגם מטריצה לנוסחה בצורה הבאה ־ עבור כל תא במטריצה $A_{i,j}$ ניצור משתנה $X_{i,j,s}$ שיחזיר $true$ אם האות $s \in S$ נמצאת בתא ה $A_{i,j}$, ו $false$ אחרת.

- **מעבר מ** CNF **ל** $3-CNF$**:** בהינתן נוסחה φ ב CNF ניתן לבנות בזמן פולינומיאלי נוסחה φ' ב $3-CNF$ כך ש φ ספיקה אמ"מ φ' ספיקה.
בהינתן פסוקית C_j:
1: אם $|C_j| = 3$ ־ נשאיר אותה כמו שהיא.
2: אם $|C_j| < 3$ ־ נכפיל ליטרלים כך: $(x_1 \vee x_2) \Rightarrow (x_1 \vee x_2 \vee x_2)$.
3: אם $|C_j| = n_j > 3$ ־ נוסיף משתני עזר. פסוקית עם n_j ליטרלים תוחלף ב $n_j - 2$ פסוקיות, עם $n_j - 3$ משתנים חדשים. כך:

$$(a_1 \vee a_2 \vee a_2 \vee ... \vee a_l) \Rightarrow (a_1 \vee a_2 \vee z_1) \wedge (\bar{z}_1 \vee a_3 \vee z_2) \wedge (\bar{z}_2 \vee a_4 \vee z_3) \wedge \ldots (\bar{z}_{l-3} \vee a_{l-1} \vee a_l)$$

למעשה כל ליטרל מוקף בשלילה של משתנה או במופע חיובי של משתנה חדש עם אינדקס אחריו.

- **שפה** IS **־ קבוצה ב"ת בגרף:** עבור גרף $G = \langle V, E \rangle$ לא מכוון, קבוצה $S \subseteq V$ היא בלתי תלויה ־ אם לכל $v_1, v_2 \in S$ מתקיים ש $(v_1, v_2) \notin E$. כלומר ־ אין קשת בין קדקודי הקבוצה.

$$IS = \{\langle G, k \rangle : ✡\}$$

עבור ✡: G גרף לא מכוון, $1 \leq k \in \mathbb{N}$, ויש ב G קבוצה ב"ת בגודל k.

טענה: $IS \in NP-complete$.

10.2 הרצאה 20:

- **המחלקה** $coNP$**:**

$$L \in NP \leftrightarrow L \in co-NP$$

- **השפה** $CONTRADICTIONS=\overline{SAT}$:

$$\overline{SAT}=\{\langle\varphi\rangle:\varphi\ is\ not\ safisticated\}$$

טענה: $\overline{SAT}\in coNP-complete$.

- **השפה** $VALIDITY\ /\ TAUTOLOGY$:

$$VAL=\{\langle\varphi\rangle:Every\ placement\ is\ satisfying\}$$

φ, כך ש כל ההשמות מספקות את φ.
טענה: $VAL\in coNP-hard$.

10.3 תרגול 10:

- **טענה:** $D-ST-HAMPATH\in NP-hard$. הוכחה ברדוקציה מ $3SAT$.
- **השפה** $U-ST-HAMPATH$:

$$U-ST-HAMPATH=\{\langle G,s,t\rangle:undirected\ G\ with\ Hamilton\ path\}$$

טענה: $U-ST-HAMPATH\in NP-complete$. הוכחה ברדוקציה מ $D-ST-HAMPATH\in NP-hard$

- **הפיכת גרף מכוון ללא מכוון:** נשכפל כל קדקוד לשלשה קדקודים $v\Rightarrow v_{in},v_{mid},v_{out}$ ועבור כל צלע (u,v) נמתח צלע (u_{out},v_{in}) ו נמתח צלעות בין v_{in},v_{mid},v_{out} לכל $v\in V$.
v_{mid} קדקוד שיתפקד כבודק ־ יבדוק שהתנאי הנדרש מתקיים (לדוגמה מסלול המילטון).
- **הגדרה** $coNP-hard$: שפה L היא $coNP-hard$ אם לכל $k\in coNP$ מתקיים $k\leq_p L$. ואם בנוסף $L\in coNP$ אזי $L\in coNP-complete$.
- **משפט:** $L\in NP-hard\iff\overline{L}\in coNP-hard$.
- **משפט:** $L\in NP-complete\iff\overline{L}\in coNP-complete$.

11 שבוע 11:

11.1 הרצאה 21 ־ סיבוכיות זכרון:

- **בעיית** $SUBSET-SUM$: בקלט יש קבוצה A של מספרי טבעיים. ומספר יעד $S\in\mathbb{N}$. הפלט הוא ־ האם יש $B\subseteq A$ כך ש $\Sigma_{b\in B}b=S$.

השפה:

$$SUBSET-SUM=\left\{\langle A,S\rangle:\begin{array}{c}B\subseteq A,\\ \sum B=S\end{array}\right\}$$

טענה: $SUBSET-SUM\in NP-complete$.

11.1.1 סיבוכיות זכרון:

- **פונקציית הזכרון של המכונה** M**:** בהינתן מ"ט בעלת סרט אחד העוצרת על כל קלט, נאמר שסיבוכיות הזכרון של M היא פונקציה $s:\mathbb{N}\Rightarrow\mathbb{N}$ כך ש $s(n)$ הוא חסם על מספר התאים ש M משתמשת בהם בריצתה על מילה באורך n.

- **מחלקות הסיבוכיות:** עבור $n\leq s(n)$ נגדיר:

$$SPACE\left(s(n)\right)=\{L:\text{✡}\}$$

עבור ✡: יש מכונה דט' שמכריעה את L בסיבוכיות זכרון $O(s(n))$.

$$NSPACE\left(s(n)\right)=\{L:\text{✡}\}$$

עבור ✡: יש מכונה (יתכן א"ד) שמכריעה את L בסיבוכיות זכרון $O(s(n))$.

- **קשרים בין סיבוכיות זמן ומקום:**

טענה 1: לכל $f(n)$ מתקיים $TIME(f(n))\subseteq SPACE(f(n))$. כי מכונה שעוצרת תוך $f(n)$ צעדים, לא יכולה להשתמש ביותר מ $f(n)$ תאים.

טענה 2: לכל $f(n)$ מתקיים $SPACE(f(n))\subseteq TIME\left(2^{O(f(n))}\right)$. כי מספר הקונפיגורציות חסום ע"י $|Q|\cdot|\Gamma|^{s(n)}\cdot s(n)$.

- **טענה:** השפה SAT ניתנת להכרעה בזמן לינארי.

הרעיון: נעבור על כל ההשמות האמת האפשריות, אם נגיע להשמה מספקת - נעצור ונקבל. אם נסיים את המעבר - נעצור ונדחה.

11.2 הרצאה 22:

- **הגדרה** $PSPACE$**:** $PSPACE=\bigcup_k SPACE\left(n^k\right)$.
- **הגדרה** $NPSPACE$**:** $NPSPACE=\bigcup_k NSPACE\left(n^k\right)$
- **טענה:** $SAT\in PSPACE$.
- **טענה:** $NP\subseteq PSPACE\subseteq EXPTIME$.
- **משפט:** $NP\subseteq PSPACE$.
- **טענה:** $NPSPACE=PSPACE$.
- **בעיית** $EMPTY_{NFA}$**:**

$$EMPTY_{NFA}=\{\langle A\rangle : A\ is\ NFA\ and\ L(A)=\emptyset\}$$

$$\overline{EMPTY_{NFA}} = \{\langle A\rangle : A \ is \ NFA \ and \ L(A) \neq \emptyset\}$$

טענה: $\overline{EMPTY_{NFA}} \in NP$ וגם $\overline{EMPTY_{NFA}} \in PTIME$.

- **בעיית** ALL_{NFA}:

$$ALL_{NFA} = \{\langle A\rangle : A \ is \ NFA \ and \ L(A) = \Sigma^*\}$$

$$\overline{ALL_{NFA}} = \{\langle A\rangle : A \ is \ NFA \ and \ L(A) \neq \Sigma^*\}$$

נשים לב: $L(\bar{A}) = \phi \leftrightarrow L(A) = \sum^*$. לכן קיימת רדוקציה (לא פולינומיאלית) $ALL_{MFA} \leqslant EMPTY_{NFA}$

טענה: $ALL_{NFA} \in EXPTIME$.

11.3 תרגול 11:

- **הערה:** אם נתונים לנו n משתנים מתוכם אנו צריכים לבחור קבוצה בגודל k:
אם k **נתון עם הקלט -** נצטרך לעשות $\binom{n}{k}$ וזו בעיה קשה (NP) משום שיכול להיות $k = \frac{n}{2}$.
אם k **ב"ת בקלט -** אזי הגודל של k הוא קבוע ו $\binom{n}{k}$ הוא פולינומי ולכן הבעיה ב P.

- **הגדרה - נוסחה מאוזנת:** נאמר שנוסחה בוליאנית φ מצורת CNF היא **מאוזנת** אם יש השמה שמקיימת:
1: σ מספקת את φ.
2: σ נותנת ללפחות שליש מהמשתנים ערך T, וגם נותנת ללפחות שליש מהמשתנים ערך F.

12 שבוע 12:

12.1 הרצאה 23:

- **טענה:** $\overline{ALL_{NFA}} \in NPSPACE$

- **טענה:** $ALL_{NFA} \in PSPACE$ (שיעור הבא נוכיח שהיא קשה).

- **משפט סביץ':** לכל פונקציה $S(n) \geq n$ מתקיים: $\text{NSPACE(S(n))} \subseteq \text{SPACE}\ (S^2\text{(n)})$
כלומר - בהינתן מ"ט א"ד שעובדת בזכרון $S(n)$, יש מ"ט דט' שקולה שעובדת בזכרון $S^2(n)$.

- **מסקנה ממשפט סביץ':** $\overline{NPSPACE} = NPSPACE = PSPACE = \overline{PSPACE}$.

- **טענה:** מחלקות דטרמנסטיות סגורות לשלילה.

- **שלמות ב** $PSPACE$**:** נאמר ש L היא $PSPACE-complete$ אם:
1: $L \in PSPACE$.
2: $L \in PSPACE-hard$ כלומר לכל $L' \in PSPACE$ מתקיים $L' \leq_p L$.

- **ההירכיה:** $\text{PTIME} \subseteq NP \subseteq \text{PSPACE} = \text{NPSRACE} \subseteq \text{EXPTIME}$.

12.2 הרצאה 24:

- **השפה** $CONT_{NFA}$:

$$CONT_{NFA} = \{\langle A_1, A_2\rangle : L(A_1) \subseteq L(A_2).\ \ A_1, A_2 \in NFA\}$$

טענה: $CONT_{NFA} \in PSPACE - hard$ (ע"י רדוקציה מ ALL_{NFA}).

- **טענה:** $ALL_{NFA} \in PSPACE - hard$.

12.3 תרגול 12:

- **השפה** $TQBF$:

$$TQBF = \{\langle\varphi\rangle : ✡\}$$

עבור ✡: φ נוסחה בוליאנית כך שהיא מכומתת לחילוטין (אין משתנים חופשיים וכל הכמתים נמצאים בתחילת הנוסחה) ונכונה (מקבלת ערך T לפי ערכים בוליאנים ולפי הכמתים הנתונים בנוסחה).

הגדרה רקורסיבית: תהי $\varphi = \square x_1, \square x_2 \ldots \square x_m \psi\left(x_1, \ldots, x_m\right)$ נוסחה ($\square$ מייצג כמתי לכל או קיים), ערך האמת שלה מוגדר רקורסיבית.

1 - אם φ **היא חסרת כמתים:** φ שווה להצבה ספציפית של משתנים ב ψ.

2 - אם $\varphi = \exists x....$: הערך של φ הוא T אם הערך של לפחות אחת מהנוסחאות הבאות הוא T.

הנוסחאות:

$$\square x_2 \ldots \square x_m \psi\left(F, \ldots, x_m\right), \quad \square x_2 \ldots \square x_m \psi\left(T, \ldots, x_m\right)$$

3 - אם $\varphi = \forall x....$: הערך של φ הוא T אם הערך של שתי הנוסחאות הבאות הוא T.

הנוסחאות:

$$\square x_2 \ldots \square x_m \psi\left(F, \ldots, x_m\right), \quad \square x_2 \ldots \square x_m \psi\left(T, \ldots, x_m\right)$$

כיצד נייצג את הנוסחה: נייצג בעץ רקורסיה בינארי כאשר לכל קדקוד יש שני בנים כשאחד מייצג ערך T והשני ערך F.

בכל פעם האלגוריתם ישמור רק את תת העץ הנוכחי עליו הוא עובד, כך נשתמש במקום פולינומי כי נשמור לכל היותר את עומק העץ (כמו DFS).

- **טענה:** $TQBF \in PSPACE - hard$.

13 שבוע 13:

13.1 הרצאה 25 - המחלקות L, NL:

- **המחלקות** $NLOGSPASE = NL$ **ו** $LOGSPASE = L$**:** סיבוכיות זכרון תת לינארית, המכונה תשתמש בשטח לוגוריתמי בקלט (חוסמים את שטח העבודה).
  הערה: זמן תת לינארי לא מעניין, כי זה אומר שלא יהיה לנו מספיק זמן לקרוא את כל הקלט.
- **כיצד תעבוד מכונה עם מיקום לוגוריתמי:** מכונה עם שני סרטים, מילת הקלט כתובה בסרט הקלט - שרט לקריעובד, כך נשתמש במקום פולינומי כי נשמור לכל היותר את עומק העץ (כמו DFS).
- **טענה:** $TQBF \in PSPACE - hard$.

14 שבוע 13:

14.1 הרצאה 25 - המחלקות L, NL:

- **המחלקות** $NLOGSPASE = NL$ **ו** $LOGSPASE = L$**:** סיבוכיות זכרון תת לינארית, המכונה תשתמש בשטח לוגוריתמי בקלט (חוסמים את שטח העבודה).
  הערה: זמן תת לינארי לא מעניין, כי זה אומר שלא יהיה לנו מספיק זמן לקרוא את כל הקלט.
- **כיצד תעבוד מכונה עם מיקום לוגוריתמי:** מכונה עם שני סרטים, מילת הקלט כתובה בסרט הקלט - שרט לקריאב בלבד.
  בנוסף יש סרט עבודה שקטן יותר מגודול הקלט, והוא סרט לקריאה וכתיבה.
- **המחלקה** $LOGSPASE$:

$$LOGSPASE = \{L : ✡\}$$

עבור ✡: יש מ"ט דט' שמכריעה את L עם סרט עבודה שמשתמש ב$O\left(log(n)\right)$ תאים על מילה באורך n.

- **המחלקה** $NLOGSPASE$:

$$NLOGSPASE = \{L : ✡\}$$

עבור ✡: יש מ"ט א"ד שמכריעה את L עם סרט עבודה שמשתמש ב$O\left(log(n)\right)$ תאים על מילה באורך n.

- **בעיית** $PATH$:

$$PATH = \{\langle G, s, t\rangle : G \ directed \ and \ thare \ is \ a \ path \ from \ s \ to \ t\}$$

אנו יודעים כי $PATH \in NP$.

נראה כי ניתן לחשב את בעיית $PATH$ **בזמן לוגריתמי:**

טענה: $PATH \in NL-complete$

המכונה זוכרת כל פעם את הקדקוד הנוכחי v, ומונה צעדים i.

באתחול $v=s, i=0$.

כל עוד $i \leq |V|$: אם $v=t$ - נעצור ונקבל, אחרת - נעדכן את v להיותו שכן של v, ונעדכן $i=i++$.

אחרת - נדחה.

סיבוכיות זכרון: $log(|V|)$ תאים ל v וגם ל i. לכם סה"כ $O\left(log(||V|)\right)$.

נכונות: אם יש מסלול אזי יש מסלול פשוט שאורכו לכל היותר $|V|$ ולכן נמצא אותו.

- **נשים לב:** ממשפט סביץ אנו יודעים כי $LOGSPACE \neq SPACE\left(\log^2 n\right) \subseteq$ NLOGSACE, לא ניתן להסיק מסביץ כי $NL \subseteq L$.
- **מספר הקונפ' שיש למכונה עם** $s(n)=O(log(n))$: קונפ' היא מצב, תוכן סרט העבודה, מיקום הראש הקורא וכותב ומיקום הראש הקורא.

  לכן: יש קבוע d כך שמספר הקונפ' חסום ע"י $2^{d \cdot log(n)}$.
- **שלמות ב** NL: נאמר ששפה L היא $NL-complete$ אם:

  1: $L \in NL$.

  2: לכל שפה $L' \in NL$ מתקיים $L' \leq_{logspace} L$.
- **הגדרה - משרן** $(transducer)$: מכונה שמחשבת את פונקציית הרוקציה, בהינתן w תחשב לנו את $f(w)$.

מ"ט דט' עם לשה סרטים: קלט - קריאה. עבודה (גודל לוגריתמי) - קריאה וכתיבה. פלט (לא מוגבל בגודל) - כתיבה. עבור קלט w המכונה משתמשת ב $O(log(|w|))$ תאים בסרט השני, כדי לחשב את הקלט, וכותבת אותו על הסרט השלישי.

- **פונקציה** $f:\Sigma^* \Rightarrow \Sigma^*$ **ניתנת לחישוב בשטח לוגריתמי:** אם קיים משרן בשטח לוגוריתמי שעל קלט w עוצר עם $f(w)$ בסרט הפלט, לכל $w \in \Sigma^*$.
- **רדוקציית** $logspace$: עבור $A, B \subseteq \Sigma^*$, נאמר ש $A \leq_{logspace} B$ אם יש פונקציה ניתנת לחישוב בשטח לוגריתמי f כך שלכל $w \in \Sigma^*$ מתקיים $w \in A \leftrightarrow f(w) \in B$.
- **משפט הרדוקציה ל** $logspace$: אם $A \leq_{logspace} B$ ו $B \in L$ אזי גם $A \in L$.

הוכחה: המכונה M_A לא מחשבת את $f(w)$, אלא בכל פעם ש M_B רוצה לקרוא את האות ה i-ית ב $f(w)$ היא מריצה את המשרן M_f ומריצה את M_B צעד אחד על האות $f(w)[i]$.

כלומר - בכל פעם נחזיר רק אות אחת, כך הזכרון יהיה לוגריתמי.

14.2 הרצאה 26:

- **הערה:** $NL \subseteq P$.

- **ההירככיה:**

$$\text{L} \subseteq \text{NL} \subseteq \text{P} \subseteq \text{NP} \ \subseteq \ \text{PSPACE} = \text{NPSPACE} \ \subseteq \ \text{EXPTIME}$$

באופן דומה:

$$\text{L} \subseteq \text{NL} \subseteq \text{P} \subseteq \text{co-NP} \ \subseteq \ \text{PSPACE} = \text{NPSPACE} \ \subseteq \ \text{EXPTIME}$$

14.2.1 שפות על גרפים ממשוקלים: $G = \langle V, E, w \rangle$ עבור $w : E \Rightarrow N^+$

- **השפה BAR (ישיגות חסומה מלמעלה):**

$$BAR = \{\langle G, s, t, b \rangle : \text{✡}\}$$

עבור ✡:אב בלבד.

בנוסף יש סרט עבודה שקטן יותר מגודול הקלט, והוא סרט לקריאה וכתיבה.

- **המחלקה $LOGSPASE$:**

$$LOGSPASE = \{L : \text{✡}\}$$

עבור ✡: יש מ"ט דט' שמכריעה את L עם סרט עבודה שמשתמש ב$O\left(log(n)\right)$ תאים על מילה באורך n.

- **המחלקה $NLOGSPASE$:**

$$NLOGSPASE = \{L : \text{✡}\}$$

עבור ✡: יש מ"ט א"ד שמכריעה את L עם סרט עבודה שמשתמש ב$O\left(log(n)\right)$ תאים על מילה באורך n.

- **בעיית $PATH$:** עובד, כך נשתמש במקום פולינומי כי נשמור לכל היותר את עומק העץ (כמו DFS).
- **טענה:** $TQBF \in PSPACE - hard$.

15 שבוע 13:

15.1 הרצאה 25 - המחלקות L, NL:

- **המחלקות $NLOGSPASE = NL$ ו $LOGSPASE = L$:** סיבוכיות זכרון תת לינארית, המכונה תשתמש בשטח לוגוריתמי בקלט (חוסמים את שטח העבודה).

  הערה: זמן תת לינארי לא מעניין, כי זה אומר שלא יהיה לנו מספיק זמן לקרוא את כל הקלט.

- **כיצד תעבוד מכונה עם מיקום לוגריתמי:** מכונה עם שני סרטים, מילת הקלט כתובה בסרט הקלט - שרט לקריאב בלבד.

  בנוסף יש סרט עבודה שקטן יותר מגודול הקלט, והוא סרט לקריאה וכתיבה.

- **המחלקה** $LOGSPASE$:

$$LOGSPASE = \{L : \text{✡}\}$$

עבור ✡: יש מ"ט דט' שמכריעה את L עם סרט עבודה שמשתמש ב$O\left(log(n)\right)$ תאים על מילה באורך n.

- **המחלקה** $NLOGSPASE$:

$$NLOGSPASE = \{L : \text{✡}\}$$

עבור ✡: יש מ"ט א"ד שמכריעה את L עם סרט עבודה שמשתמש ב$O\left(log(n)\right)$ תאים על מילה באורך n.

- **בעיית** $PATH$:

$$PATH = \{\langle G, s, t\rangle : G \ directed \ and \ thare \ is \ a \ path \ from \ s \ to \ t\}$$

אנו יודעים כי $PATH \in NP$.

נראה כי ניתן לחשב את בעיית $PATH$ **בזמן לוגריתמי:**

טענה: $PATH \in NL - complete$

המכונה זוכרת כל פעם את הקדקוד הנוכחי v, ומונה צעדים i.

באתחול $v = s, i = 0$.

כל עוד $i \leq |V|$: אם $v = t$ - נעצור ונקבל, אחרת - נעדכן את v להיתו שכן של v, ונעדכן $i = i++$.

אחרת - נדחה.

סיבוכיות זכרון: $log(|V|)$ תאים ל v וגם ל i. לכם סה"כ $O\left(log(||V|)\right)$.

נכונות: אם יש מסלול אזי יש מסלול פשוט שאורכו לכל היותר $|V|$ ולכן נמצא אותו.

- **נשים לב:** ממשפט סביץ אנו יודעים כי $\text{NLOGSACE} \subseteq SPACE\left(\log^2 n\right) \neq LOGSPACE$, לא ניתן להסיק מסביץ כי $NL \subseteq L$.

- **מספר הקונפ' שיש למכונה עם** $s(n) = O(log(n))$: קונפ' היא מצב, תוכן סרט העבודה, מיקום הראש הקורא וכותב ומיקום הראש הקורא.

  לכן: יש קבוע d כך שמספר הקונפ' חסום ע"י $2^{d \cdot log(n)}$.

- **שלמות ב** NL: נאמר ששפה L היא $NL - complete$ אם:

  1: $L \in NL$.

  2: לכל שפה $L' \in NL$ מתקיים $L' \leq_{logspace} L$.

- **הגדרה - משרן** $(transducer)$: מכונה שמחשבת את פונקציית הרוקציה, בהינתן w תחשב לנו את $f(w)$.
מ"ט דט' עם לשה סרטים: קלט - קריאה. עבודה (גודל לוגריתמי) - קריאה וכתיבה. פלט (לא מוגבל בגודל) - כתיבה. עבור קלט w המכונה משתמשת ב $O(log(|w|))$ תאים בסרט השני, כדי לחשב את הקלט, וכותבת אותו על הסרט השלישי.

- **פונקציה** $f : \Sigma^* \Rightarrow \Sigma^*$ **ניתנת לחישוב בשטח לוגריתמי:** אם קיים משרן בשטח לוגוריתמי שעל קלט w עוצר עם $f(w)$ בסרט הפלט, לכל $w \in \Sigma^*$.

- **רדוקציית** $logspace$: עבור $A, B \subseteq \Sigma^*$, נאמר ש $A \leq_{logspace} B$ אם יש פונקציה ניתנת לחישוב בשטח לוגריתמי f כך שלכל $w \in \Sigma^*$ מתקיים $w \in A \leftrightarrow f(w) \in B$.

- **משפט הרדוקציה ל** $logspace$: אם $A \leq_{logspace} B$ ו $B \in L$ אזי גם $A \in L$.
הוכחה: המכונה M_A לא מחשבת את $f(w)$, אלא בכל פעם ש M_B רוצה לקרוא את האות ה i-ית ב $f(w)$ היא מריצה את המשרן M_f ומריצה את M_B צעד אחד על האות $f(w)[i]$.
כלומר - בכל פעם נחזיר רק אות אחת, כך הזכרון יהיה לוגריתמי.

15.2 הרצאה 26:

- **הערה:** $NL \subseteq P$.

- **ההירכיה:**

$$\text{L} \subseteq \text{NL} \subseteq \text{P} \subseteq \text{NP} \ \subseteq \ \text{PSPACE} = \text{NPSPACE} \ \subseteq \ \text{EXPTIME}$$

באופן דומה:

$$\text{L} \subseteq \text{NL} \subseteq \text{P} \subseteq \text{co-NP} \ \subseteq \ \text{PSPACE} = \text{NPSPACE} \ \subseteq \ \text{EXPTIME}$$

15.2.1 שפות על גרפים ממשוקלים: $G = \langle V, E, w \rangle$ עבור $w : E \Rightarrow N^+$

- **השפה** BAR **(ישיגות חסומה מלמעלה):**

$$BAR = \{\langle G, s, t, b \rangle : ✡\}$$

עבור ✡:

$$PATH = \{\langle G, s, t \rangle : G \ directed \ and \ thare \ is \ a \ path \ from \ s \ to \ t\}$$

אנו יודעים כי $PATH \in NP$.

נראה כי ניתן לחשב את בעיית $PATH$ **בזמן לוגריתמי:**

טענה: $PATH \in NL - complete$

המכונה זוכרת כל פעם את הקדקוד הנוכחי v, ומונה צעדים i.

באתחול $v = s, i = 0$.

כל עוד $i \leq |V|$: אם $v = t$ ־ נעצור ונקבל, אחרת ־ נעדכן את v להיותו שכן של v, ונעדכן $i = i++$.

אחרת ־ נדחה.

סיבוכיות זכרון: $log(|V|)$ תאים ל v וגם ל i. לכם סה"כ $O\left(log(||V|)\right)$.

נכונות: אם יש מסלול אזי יש מסלול פשוט שאורכו לכל היותר $|V|$ ולכן נמצא אותו.

- **נשים לב:** ממשפט סביץ אנו יודעים כי $\text{NLOGSACE} \subseteq SPACE\left(\log^2 n\right) \neq LOGSPACE$, לא ניתן להסיק מסביץ כי $NL \subseteq L$.

- **מספר הקונפ' שיש למכונה עם** $s(n) = O(log(n))$: קונפ' היא מצב, תוכן סרט העבודה, מיקום הראש הקורא וכותב ומיקום הראש הקורא.

  לכן: יש קבוע d כך שמספר הקונפ' חסום ע"י $2^{d \cdot log(n)}$.

- **שלמות ב** NL: נאמר ששפה L היא $NL - complete$ אם:

  1: $L \in NL$.

  2: לכל שפה $L' \in NL$ מתקיים $L' \leq_{logspace} L$.

- **הגדרה ־ משרן** $(transducer)$: מכונה שמחשבת את פונקציית הרוקציה, בהינתן w תחשב לנו את $f(w)$.

מ"ט דט' עם לשה סרטים: קלט ־ קריאה. עבודה (גודל לוגריתמי) ־ קריאה וכתיבה. פלט (לא מוגבל בגודל) ־ כתיבה.
עבור קלט w המכונה משתמשת ב $O(log(|w|))$ תאים בסרט השני, כדי לחשב את הקלט, וכותבת אותו על הסרט השלישי.

- **פונקציה** $f : \Sigma^* \Rightarrow \Sigma^*$ **ניתנת לחישוב בשטח לוגריתמי:** אם קיים משרן בשטח לוגוריתמי שעל קלט w עוצר עם $f(w)$ בסרט הפלט, לכל $w \in \Sigma^*$.

- **רדוקציית** $logspace$: עבור $A, B \subseteq \Sigma^*$, נאמר ש $A \leq_{logspace} B$ אם יש פונקציה ניתנת לחישוב בשטח לוגריתמי f כך שלכל $w \in \Sigma^*$ מתקיים $w \in A \leftrightarrow f(w) \in B$.

- **משפט הרדוקציה ל** $logspace$: אם $A \leq_{logspace} B$ ו $B \in L$ אזי גם $A \in L$.

הוכחה: המכונה M_A לא מחשבת את $f(w)$, אלא בכל פעם ש M_B רוצה לקרוא את האות ה i־ית ב $f(w)$ היא מריצה את המשרן M_f ומריצה את M_B צעד אחד על האות $f(w)[i]$.

כלומר ־ בכל פעם נחזיר רק אות אחת, כך הזכרון יהיה לוגריתמי.

15.3 הרצאה 26:

- **הערה:** $NL \subseteq P$.

- **ההירכיה:**

$$L \subseteq NL \subseteq P \subseteq NP \ \subseteq \ PSPACE = NPSPACE \ \subseteq \ EXPTIME$$

באופן דומה:

$$L \subseteq NL \subseteq P \subseteq \text{co-}NP \ \subseteq \ PSPACE = NPSPACE \ \subseteq \ EXPTIME$$

15.3.1 שפות על גרפים ממשוקלים: $G = \langle V, E, w \rangle$ עבור $w : E \Rightarrow N^+$

- **השפה BAR (ישיגות חסומה מלמעלה):**

$$BAR = \{\langle G, s, t, b \rangle : ✡\}$$

עבור ✡: G גרף מכוון ממושקל עם משקולות חיוביים נתונים באונארית ו $s, t \in V$, ו $b \geq 0$ נתון באונארית. אנו רוצים לדעת אם יש מסלול מ $s - t$ במשקל $\geq b$.

טענה: $BAR \in NL - hard$.
בניה: תנחש מסלול ותשמור בכל רגע רק קודקוד נוכחי, ומונה ששומר את סכום המשקלים עד כה.

- **השפה BBR (ישיגות חסומה מלמטה):**

$$BBR = \{\langle G, s, t, b \rangle : ✡\}$$

עבור ✡: G גרף מכוון ממושקל עם משקולות חיוביים, ו $s, t \in V$, ו $b \geq 0$ נתון באונארית. אנו רוצים לדעת אם יש מסלול מ $s - t$ במשקל $\leq b$.

טענה: $BBR \in NL - hard$.
בניה: מנחשת מסלול שמשקלו $\leq b$ עד קדקוד v כלשהו, ואחכ מוצאת מסלול מ v ל t. (בכל פעם תשמור קדקוד נוכחי ומונה)

- **השפה $SBBR$:**

$$SBBR = \{\langle G, s, t, b \rangle : ✡\}$$

עבור ✡: G גרף מכוון ממושקל עם משקולות חיוביים, ו $s, t \in V$, ו $b \geq 0$ נתון באונארית. אנו רוצים לדעת אם יש מסלול **פשוט** מ $s - t$ במשקל $\leq b$.

טענה: $SBBR \in NP - hard$. (רדוקציה מ $HAMPATH$)

15.4 תרגול 13:

- **הגדרה ־ חשיבה בזמן:** פונקציה t **חשיבה בזמן** אם יש מ"ט כך שבהינתן $1^{t(n)}$ באונארית ומחשבת את $t(n)$ ביצוג בינארי בזמן של $O(t(n))$.

  הערה: כל פולינום עם מקדמים אי שליליים ואקספוננט הם חשיבים בזמן.

- **משפט ההירכיה בזמן:** עבור $t : N \Rightarrow N$ אם $t = \Omega(n \cdot log(n))$ וגם חשיבה בזמן, אזי קיימת שפה L שניתנת להכרעה בזמן $O(t(n))$ אבל לא ניתנת להכרעה בזמן $o\left(\frac{t(n)}{log(t(n))}\right)$.

- **מסקנות מהמשפט:**

  1: לכל $c_1 > c_2 \geq 2$ מתקיים $TIME\left(n^{c_2}\right) \subsetneq TIME\left(n^{c_1}\right)$.

  2: $P \subsetneq EXPTIME$.

- **טענה:** יש מ"ט S כך שבהינתן קלט $\langle M, w, t\rangle$, S מסמלצת את ריצת M על w במשך t צעדים, לוקח ל S לסמלץ את הריצה של M ־ $t \cdot log(t) \cdot p\left(|\left\langle M\right\rangle|\right)$ צעדים. (עבור p פולינום קבוע כלשהו)

- **הגדרה ־ חשיבה במקום:** פונקציה t **חשיבה במקום** אם יש מ"ט כך שבהינתן $1^{t(n)}$ באונארית ומחשבת את $t(n)$ ביצוג בינארי במקום של $O(t(n))$.

- **משפט ההירכיה במקום:** אם $t = \Omega(log(n))$ וגם t חשיבה במקום,יש שפה L כך ש $L \in SPACE(t(n)) \backslash SPACE(o(t(n)))$.

- **השפה** $2 - SAT$:

$$2 - SAT = \{\langle\varphi\rangle : \varphi \; satisfiable\}$$

  טענה: $2 - SAT \in NL$.

  בניה:

  עבור φ ־ נתבונן בגרף שקדקודיו הם הליטרלים ושלילתם. עבור פסוקית (a, b) נמתח צלע $(\overline{a}, b)$ ו $\left(\overline{b}, a\right)$. כעת אם יש מסלול מליטרל לשלילתו ולהיפך אזי הנוסחה לא ספיקה. אחרת, אם אין צלע בין משתנה לשלילתו ־ ניתן למשתנה את הערך T ולכל הקדקודים הישיגים ממנו גם כן. אם יש צלע בין משתנה לשלילתו ־ ניתן למשתנה ערך F וללילתו ולישיגים ממנו ערך T.

16 שבוע 14:

16.1 הרצאה 27:

- **הערה:** בכל המחלקות הדט' $P, PSPACE, L$ מתקיים כי המחלקה שווה ל $co-$.

- **משפט:** $NL = co - NL$.

- **משפט:** $\overline{PATH} \in NL$

- **השפה** A_{DFA}:

$$A_{DFA} = \{\langle A \rangle : A \ is \ DFA \ and \ L(A) = \Sigma^*\}$$

טענה: $A_{DFA} \in NL$.

- **השפה** INF_{DFA}:

$$INF_{DFA} = \{\langle A \rangle : A \ is \ DFA \ and \ L(A) \ is \ infinity\}$$

טענה: $INF_{DFA} \in NL$.
טענה: אותו הדבר נכון לגבי NFA

- **השפה** MIN_{DFA}:

$$MIN_{DFA} = \{\langle A, k \rangle : ✡\}$$

עבור ✡: A הוא DFA וקיים לו אוטומט שקול B כך שמספר המצבים ב B שווים ל k.
טענה: $MIN_{DFA} \in PTIME - complete$.

- **שלמות ב** $PTIME$**:** שפה $A \in PTIME - complete$ אם:
1: $A \in PTIMR$.
2: לכל שפה $B \in PTIME$ מתקיים $B \leq_{logspace} A$

- **דוגמאות לשפות** $PTIME - complete$:
1: שערוך מעגלים בוליאנים. (מעגל עם שערי or, and ואנו רוצים לדעת אם חוזר לנו T בסוף)
2: ישיגות מתחלפת בגרף AR. (גרף עם שני סוגי קדקודים שונים, שמייצגים שני שחקנים שונים והשאלה היא אם יש לשחקן א אפשרות להגיע מ $s - t$)

- **טענה:** כל שפה שהיא ב P ניתנת לרדוקציה בזמן פולינומיאלי לכל שפה לא טרוויאלית.

16.2 תרגול 14:

- **טענה:** $L \subseteq P$.

- **טענה:** $NL \subseteq NP$.

- **השפה** $SC - strongly \ connected \ component$:

$$SC = \{\langle G \rangle : ✡\}$$

עבור ✡: G גרף מכוון קשיר חזק, כלומר - עבור כל שני קדקודים $x, y \in V$ יש מסלול $y - x$ וגם מסלול $x - y$.

טענה: $SC \in NL - complete$.

- **השפה** $2-SC$**:**

$$2-SC=\{\langle G\rangle : ✡\}$$

עבור ✡: G גרף מכוון שיש בו בדיוק שני רכיבי קשירות חזקה.

טענה: $2-SC \in NL-complete$.

- **השפה** $MAX2SAT$**:**

$$MAX2SAT=\{\langle \varphi, k\rangle : ✡\}$$

עבור ✡: φ היא מצורת $2CNF$, ויש השמה שמספקת לפחות k מהפסוקיות של φ.

טענה: $MAX2SAT \in NP-complete$.