NP- שלמות

- הצבענו על הדמיון שקיים בין המחלקה **RE** והמחלקה **NP**, ובין המחלקה **R** והמחלקה **P**.
- בהרצאה הזו נרחיב את הדמיון הזה גם למחלקה **RE-complete**, ונלמד על המקבילה שלה - שפות **NP**-שלמות. (אולי אחת ההגדרות הכי חשובות במדעי המחשב)

תזכורות מתורת החישוביות

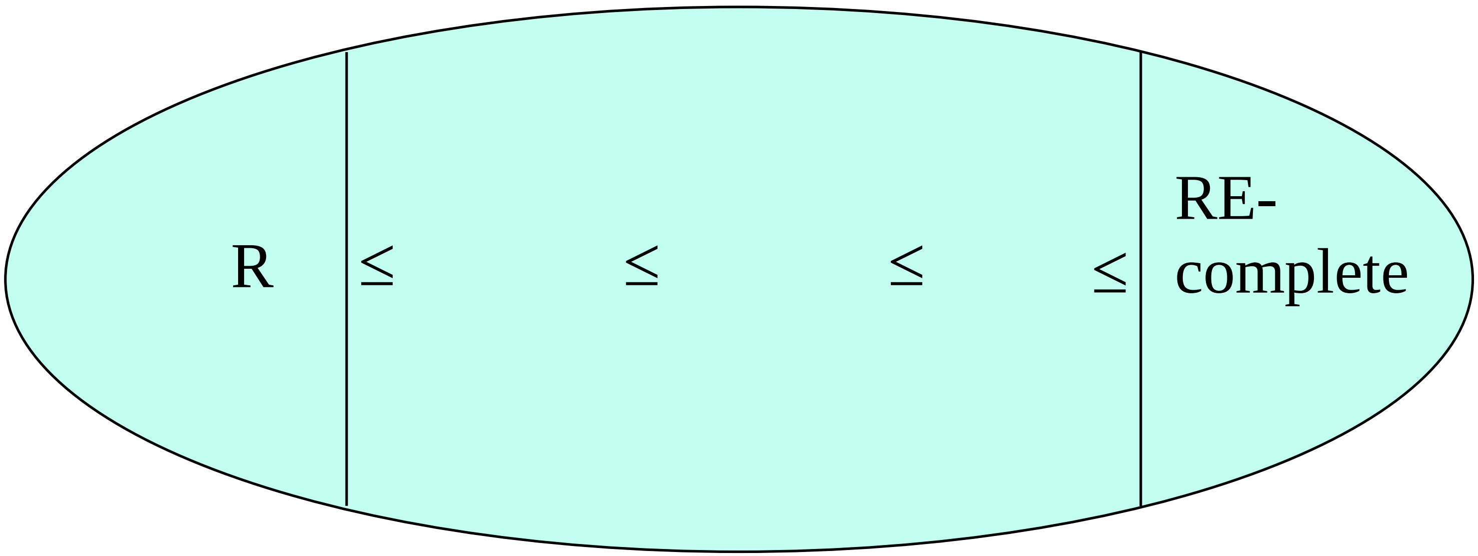
- נזכיר: השפות הכריעות (R) הן השפות הקלות במחלקת השפות RE.
  - יש רדוקציה מכל שפה כריעה לכל שפה לא טריוויאלית ב-RE.

תזכורות מתורת החישוביות

- נזכיר: השפות הכריעות (R) הן השפות הקלות במחלקת השפות RE.
  - יש רדוקציה מכל שפה כריעה לכל שפה לא טריוויאלית ב-RE.
- השפות הקשות במחלקה RE הן השפות השלמות ב-RE.
  - יש רדוקציה מכל שפה ב-RE לכל שפה שלמה ב-RE

תזכורת: איור של המחלקה

- המחלקה RE (מלבד השפות הטרוויאליות)



רדוקציות כלליות לא מתאימות

- אם נרצה לבנות משהו דומה במחלקה **NP**, לא נוכל להשתמש ברדוקציות לא מוגבלות
  - כל השפות ב-**NP** הן כריעות, וממילא ניתנות לרדוקציה לכל שפה ב-**NP** (פרט לטריוויאליות)

רדוקציות כלליות לא מתאימות

- אם נרצה לבנות משהו דומה במחלקה **NP**, לא נוכל להשתמש ברדוקציות לא מוגבלות
  – כל השפות ב-**NP** הן כריעות, וממילא ניתנות לרדוקציה לכל שפה ב-**NP** (פרט לטריוויאליות).
- לכן נגדיר סוג מיוחד של רדוקציות - רדוקציות שאפשר לחשב אותן בזמן פולינומיאלי.
  – אלה רדוקציות שיש מכונה שמחשבת את הרדוקציה בזמן פולינומיאלי בגודל הקלט.

רדוקציות בזמן פולינומיאלי

- **הגדרה**: תהיינה A ו-B שפות מעל האלפבתים Σ_A ו-Σ_B ($A \subseteq \Sigma_A^*$, $B \subseteq \Sigma_B^*$).

  רדוקציה בזמן פולינומיאלי של A ל-B היא פונקציה ניתנת לחישוב בזמן פולינומיאלי (בגודל הקלט) $f: \Sigma_A^* \to \Sigma_B^*$ שמקיימת:

  – אם $w \in A$, אז $f(w) \in B$.

  – אם $w \notin A$, אז $f(w) \notin B$.

רדוקציות בזמן פולינומיאלי

- **הגדרה**: תהיינה A ו-B שפות מעל האלפביתים Σ_A ו-Σ_B ($A \subseteq \Sigma_A{}^*$, $B \subseteq \Sigma_B{}^*$).

  רדוקציה בזמן פולינומיאלי של A ל-B היא פונקציה ניתנת לחישוב בזמן פולינומיאלי (בגודל הקלט) $f : \Sigma_A{}^* \rightarrow \Sigma_B{}^*$ שמקיימת:

  – אם $w \in A$, אז $f(w) \in B$

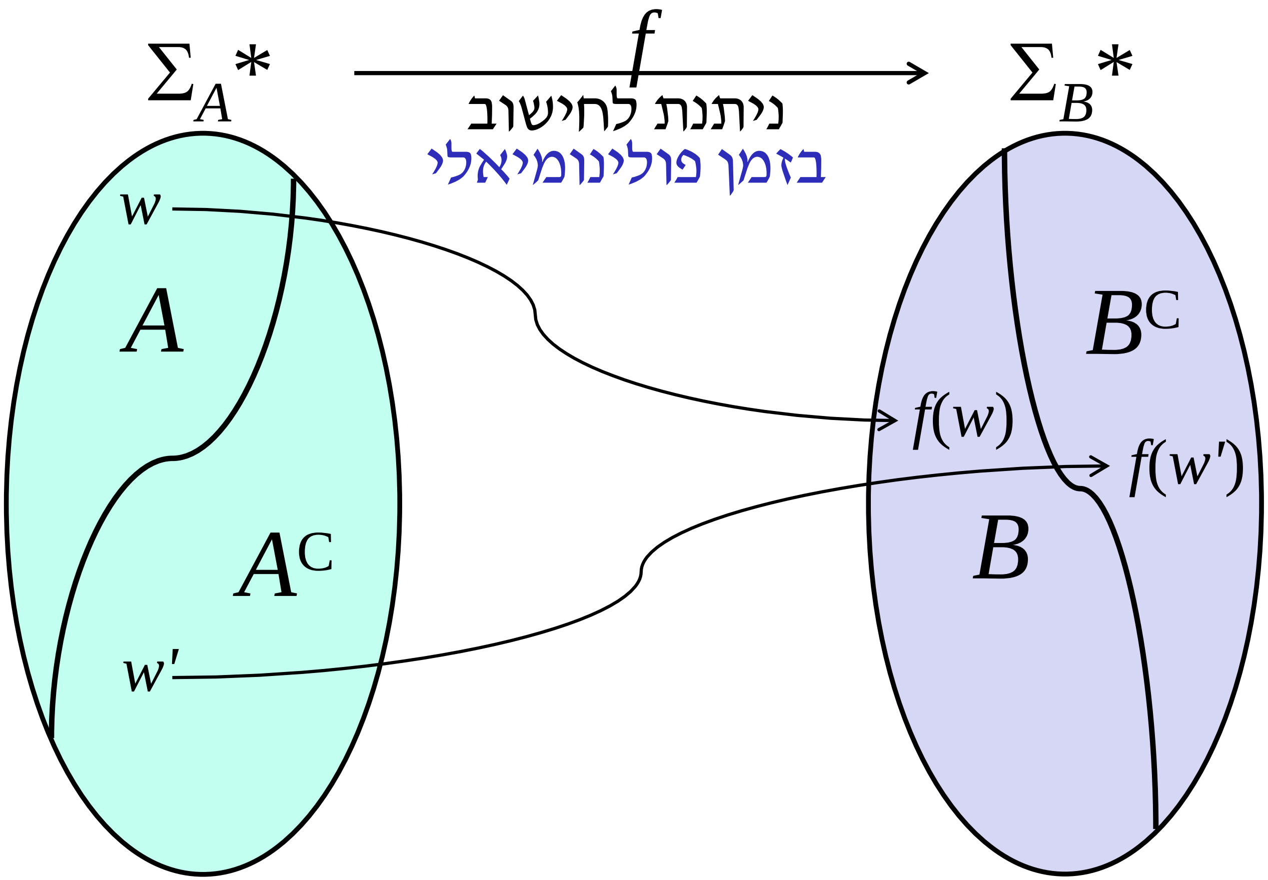
  – אם $w \notin A$, אז $f(w) \notin B$

רדוקציות בזמן פולינומיאלי

- **הגדרה**: תהיינה A ו-B שפות מעל האלפבתים Σ_A ו-Σ_B ($A \subseteq \Sigma_A^*$, $B \subseteq \Sigma_B^*$).

  רדוקציה בזמן פולינומיאלי של A ל-B היא פונקציה ניתנת לחישוב בזמן פולינומיאלי (בגודל הקלט) $f : \Sigma_A^* \to \Sigma_B^*$ שמקיימת:
  - אם $w \in A$, אז $f(w) \in B$.
  - אם $w \notin A$, אז $f(w) \notin B$.
  - **סימון**: אם יש רדוקציה בזמן פולינומיאלי של A ל-B, מסמנים $A \le_P B$.

ציור רדוקציוניסטי



רדוקציות בזמן פולינומיאלי

- אם שתי השפות מעל אותו א״ב נוכל להגדיר בצורה פשוטה יותר:

תהינה $L_1,L_2 \subseteq \Sigma^*$

נאמר כי $L_1 \leq_p L_2$ (במילים: L_1 ניתנת לרדוקציה פולינומית ל-L_2) אם:

קיימת פונקציה $f: \Sigma^* \rightarrow \Sigma^*$ המקיימת:

- f ניתנת לחישוב בזמן פולינומי
- f תקפה: $\forall x \in \Sigma^*: x \in L_1 \iff f(x) \in L_2$

דוגמאות לרדוקציות פולינומיות

• נתונות השפות הבאות:

$$Clique = \{< G, k > | G \text{ is a graph that contains a clique of size } k\}$$

$$IS = \{< G, k > | G \text{ is a graph that contains an independent set of size } k\}$$

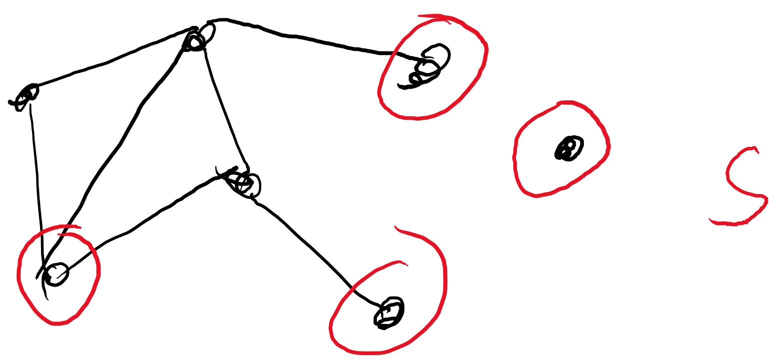
$$VC = \{< G, k > | G \text{ is a graph that contains a vertex cover of size } k\}$$

דוגמאות לרדוקציות פולינומיות

- **קבוצה בלתי תלויה, Independent Set:**

בהינתן גרף $G=(V,E)$, קבוצה בלתי תלויה ב-G היא תת קבוצה של קודקודים $S\subseteq V$ המקיימת שלא קיימות צלעות בגרף ששני הקודקודים החלים בהן שייכים ל-S.

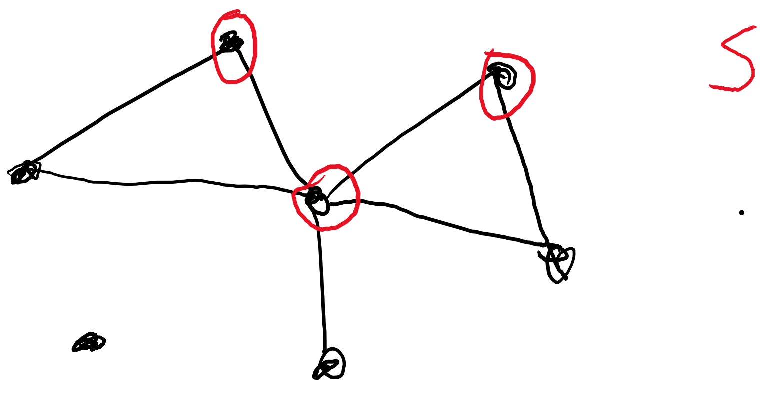
(במילים אחרות, לכל אחת מהצלעות ב-G יש לכל היותר קודקוד אחד ב-S.)



דוגמאות לרדוקציות פולינומיות

- **כיסוי קודקודים, Vertex cover**:

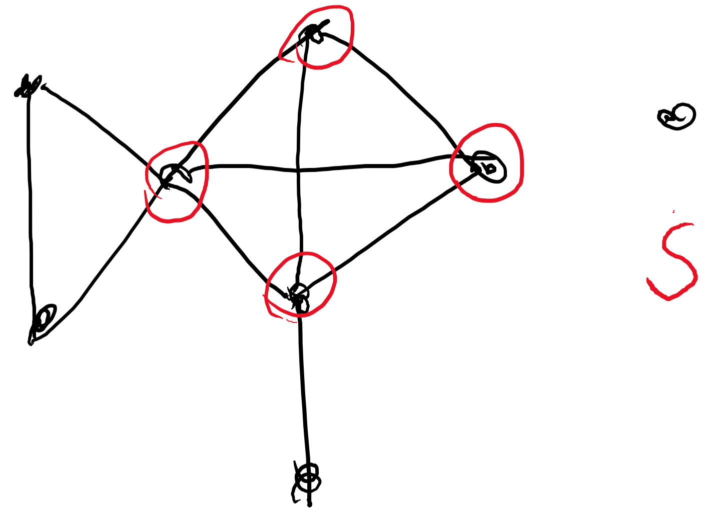
בהינתן גרף $G = (V,E)$, כיסוי קודקודים של G הוא תת קבוצה של קודקודים $S \subseteq V$ המקיימת שלכל צלע בגרף, לפחות אחד מהקודקודים החלים בצלע שייך ל-S.



דוגמאות לרדוקציות פולינומיות

- **קליקה, Clique**:

בהינתן גרף $G = (V,E)$, קליקה ב-G היא תת קבוצה של קודקודים $S \subseteq V$ המקיימת שכל זוג קודקודים ב-S הם שכנים בגרף.



$Clique \leq_p IS$

רעיונות לרדוקציה?

$Clique \leq_p IS$

הרדוקציה:

$$f(<G,k>) =< \bar{G},k>$$

הרדוקציה פולינומית: כדי לחשב את f יש לעבור על כל זוג קודקודים ב-G, אם הקודקודים שכנים ב-G, לא תהיה ביניהם צלע ב-$\bar{G}$, ואם הקודקודים לא שכנים ב-G תהיה ביניהם צלע ב-$\bar{G}$.

מה הסיבוכיות?

$Clique \leq_p IS$

הרדוקציה:

$$f(<G,k>) = <\bar{G},k>$$

הרדוקציה פולינומית: כדי לחשב את f יש לעבור על כל זוג קודקודים ב-G, אם הקודקודים שכנים ב-G, לא תהיה ביניהם צלע ב-$\bar{G}$, ואם הקודקודים לא שכנים ב-G תהיה ביניהם צלע ב-$\bar{G}$.

לכן הסיבוכיות היא $\binom{n}{2} = O(n^2)$, (כאשר n הוא מספר הקודקודים ב- G)- פולינומי. (בהנחה שניתן לבדוק האם זוג קודקודים הם שכנים בזמן קבוע)

(הכי קל: לעבור על מטריצת השכנויות ולהפוך כל ביט)

$Clique \leq_p IS$

תקפות הרדוקציה נובעת מהלמה הבאה:

<u>למה</u>: עבור גרף $G = (V,E)$ קבוצת קודקודים $S \in V(G)$ היא קליקה ב-G אס"ם S היא קבוצה בלתי תלויה ב-$\bar{G}$.

$Clique \leq_p IS$

תקפות הרדוקציה נובעת מהלמה הבאה:

<u>למה</u>: עבור גרף $G = (V,E)$ קבוצת קודקודים $S \in V(G)$ היא קליקה ב-G אם"ם S היא קבוצה בלתי תלויה ב-$\bar{G}$.

<u>הוכחה</u>:

- נניח ש-S היא קליקה ב-G, אזי כל הצלעות האפשריות של זוגות מ-S קיימות ב-G, לכן ב-$\bar{G}$ כל הצלעות הללו לא קיימות, מכאן ש-S היא קבוצה בלתי תלויה ב- $\bar{G}$.

$Clique \leq_p IS$

תקפות הרדוקציה נובעת מהלמה הבאה:

<u>למה</u>: עבור גרף $G = (V,E)$ קבוצת קודקודים $S \in V(G)$ היא קליקה ב-G אס"ם S היא קבוצה בלתי תלויה ב-$\bar{G}$.

<u>הוכחה</u>:

- נניח ש-S היא קליקה ב-G, אזי כל הצלעות האפשריות של זוגות מ-S קיימות ב-G, לכן ב-$\bar{G}$ כל הצלעות הללו לא קיימות, מכאן ש-S היא קבוצה בלתי תלויה ב- $\bar{G}$.
- נניח ש-S היא קבוצה בלתי תלויה ב- $\bar{G}$, אזי כל הצלעות האפשריות של זוגות מ-S לא קיימות ב- $\bar{G}$, לכן ב-G כל הצלעות הללו קיימות, מכאן ש-S היא קליקה ב-G.

מ.ש.ל

$IS \leq_p VC$

הרדוקציה:

$$f(< G, k >) =< G, n - k >$$

הרדוקציה פולינומית: כדי לחשב את f הרדוקציה רק מחשבת את מספר קודקודי הגרף ($n=$) ומפחיתה מ-n את k.

$IS \leq_p VC$

תקפות הרדוקציה נובעת מהלמה הבאה:

<u>למה</u>: עבור גרף $G = (V,E)$ קבוצת קודקודים $S \in V(G)$ היא בלתי תלויה ב-G אס"ם $V \backslash S$ היא כיסוי קודקודים של G.

$IS \leq_p VC$

תקפות הרדוקציה נובעת מהלמה הבאה:

<u>למה</u>: עבור גרף $G = (V,E)$ קבוצת קודקודים $S \in V(G)$ היא בלתי תלויה ב-G אס"ם $V\setminus S$ היא כיסוי קודקודים של G.

<u>הוכחה</u>:

נניח ש-S היא קבוצה בלתי תלויה ב-G, אזי אין צלעות המחברות שני קודקודים ב-S, כלומר, כל צלעות הגרף נוגעות בלכל היותר קודקוד אחד מ-S. אזי הקבוצה $V\setminus S$ נוגעת בלפחות קודקוד אחד מכל צלע בגרף, כלומר $V\setminus S$ היא כיסוי קודקודים.

נניח ש-$V\setminus S$ היא כיסוי קודקודים ב-G, אזי כל צלע ב-G היא בעלת לפחות קודקוד אחד בקבוצה $V\setminus S$. מכאן שאין בכלל צלעות שחלות על שני קודקודים מ-S, כלומר S היא קבוצה בלתי תלויה.

מ.ש.ל

דוגמאות לרדוקציות פולינומיות

- **תרגיל**: הוכיחו: אם $L_1 \leq_P L_2$, אז $\bar{L}_1 \leq_P \bar{L}_2$.

שייכות ל-P בעזרת רדוקציה

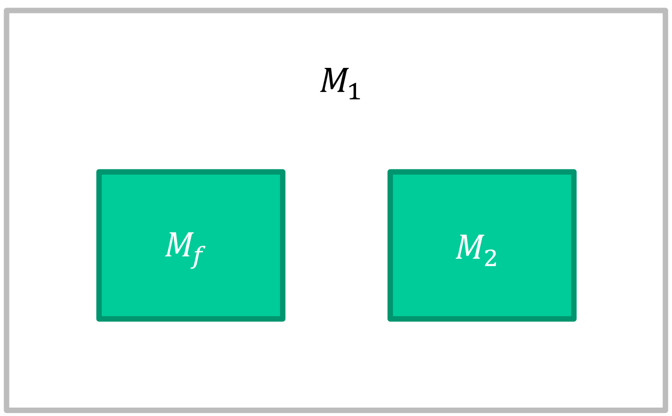
- **משפט**: אם $L_1 \leq_P L_2$ ו-$L_2 \in P$,
  אזי גם L_1 שייכת ל-P.

- **הערה**: לא נשתמש במסקנה הבאה (המקבילה למסקנה שבה השתמשנו בתורת החישוביות):
  אם $L_1 \leq_P L_2$ ו-L_1 לא שייכת ל-$\boldsymbol{P}$, אז גם L_2 לא שייכת ל-$\boldsymbol{P}$
  – לא נעסוק בשפות שהוכח עליהן שהן לא שייכות ל-$\boldsymbol{P}$.

משפט הרדוקציה עבור רדוקציה פולינומית

אם $L_1 \leq_p L_2$ אז:

- אם $L_2 \in P$ אז גם $L_1 \in P$.



רעיון ההוכחה (בציור)

משפט הרדוקציה עבור רדוקציה פולינומית- הוכחה

נניח כי $L_1 \leq_p L_2$ וגם $L_2 \in P$.

- תהי M_f מכונה דטר׳ פולינומית המחשבת את פונקציית הרדוקציה, ויהי p_1 הפולינום החוסם את זמן הריצה שלה.
- תהי M_2 מכונת טיורינג דטר׳ פולינומית המכריעה את L_2 ויהי p_2 חסם זמן הריצה שלה.

משפט הרדוקציה עבור רדוקציה פולינומית- הוכחה

נניח כי $L_1 \leq_p L_2$ וגם $L_2 \in P$.

- תהי M_f מכונה דטר׳ פולינומית המחשבת את פונקציית הרדוקציה, ויהי p_1 הפולינום החוסם את זמן הריצה שלה.
- תהי M_2 מכונת טיורינג דטר׳ פולינומית המכריעה את L_2 ויהי p_2 חסם זמן הריצה שלה.

נתאר מכונה דטר׳ פולינומית המכריעה את L_1:

M_1 על קלט x:

1. מחשבת את $f(x)$.
2. מריצה את M_2 על $f(x)$ ועונה כמוה.

משפט הרדוקציה - הוכחה

נכונות המכונה נובעת ישירות מהתקפות של פונקציית הרדוקציה.

בנוסף, זמן הריצה של M_1 הוא $O(p_1(|x|) + p_2(p_1(|x|)))$ -פולינומי.

מ.ש.ל

הפסקה

משפטים נוספים

- **משפט** : אם $L_2 \leq_P L_1$ ו-$L_2 \in NP$,
  אז גם L_1 שייכת ל-$\boldsymbol{NP}$

משפטים נוספים

- **משפט**: אם $L_1 \leq_P L_2$ ו-$L_2 \in NP$,
  אז גם L_1 שייכת ל-$\boldsymbol{NP}$
- **תרגיל**: הוכיחו את המשפט.

משפטים נוספים

- **משפט**: אם $L_1 \leq_P L_2$ ו-$L_2 \in NP$,
אז גם L_1 שייכת ל-$\boldsymbol{NP}$
- **תרגיל**: הוכיחו את המשפט.
- **הערה**: גם כאן לא נשתמש במסקנה מן המשפט
כדי להוכיח ששפות אינן שייכות ל-**NP**
  - לא נעסוק בשפות שהוכח עליהן שהן לא שייכות ל-**NP**

משפטים נוספים

- **משפט**: אם $L_2 \leq_P L_2$ ו-$L_1 \in NP$, L_2
  אז גם L_1 שייכת ל-$\boldsymbol{NP}$
- **תרגיל**: הוכיחו את המשפט.
- **הערה**: גם כאן לא נשתמש במסקנה מן המשפט כדי להוכיח ששפות אינן שייכות ל-**NP**
  - לא נעסוק בשפות שהוכח עליהן שהן לא שייכות ל-**NP**
- **משפט**: אם L_1 היא שפה ב-**P**, ו-L_2 היא שפה כלשהי שונה מ-ϕ ומ-Σ^*, אז $L_1 \leq_P L_2$.

משפטים נוספים

- **משפט**: אם $L_2 \leq_P L_2$ ו-$L_1 \leq_P L_2$, $L_2 \in NP$,
אז גם L_1 שייכת ל-$\boldsymbol{NP}$
- **תרגיל**: הוכיחו את המשפט.
- **הערה**: גם כאן לא נשתמש במסקנה מן המשפט
כדי להוכיח ששפות **אינן שייכות ל-NP**
  - לא נעסוק בשפות שהוכח עליהן שהן לא שייכות ל-**NP**
- **משפט**: אם L_1 היא שפה ב-**P**, ו-L_2 היא שפה כלשהי
שונה מ-ϕ ומ-Σ^*, אז $L_1 \leq_P L_2$.
- **תרגיל**: הוכיחו את המשפט.

היחס $\leq_P$ טרנזיטיבי

- **תרגיל**: הוכיחו שהיחס $\leq_P$ הוא טרנזיטיבי
  - אם $L_1 \leq_P L_2$ ו-$L_2 \leq_P L_3$, אז $L_1 \leq_P L_3$.
- **תרגיל**: האם היחס $\leq_P$ הוא רפלקסיבי? הוכיחו. האם הוא סימטרי? הוכיחו.

היחס $\leq_P$ טרנזיטיבי

- **תרגיל**: הוכיחו שהיחס $\leq_P$ הוא טרנזיטיבי
  - אם $L_1 \leq_P L_2$ ו-$L_2 \leq_P L_3$, אז $L_1 \leq_P L_3$.
- **תרגיל**: האם היחס $\leq_P$ הוא רפלקסיבי? הוכיחו. האם הוא סימטרי? הוכיחו.
- אינטואיטיבית, אם $L_2 \leq_P L_1$, אז L_2 קשה לפחות כמו L_1 (במובן של קיום אלגוריתם מהיר לשפה)
  - אם יש אלגוריתם מהיר לבדיקת השייכות ל-L_2, אז אפשר לבנות בעזרתו (ובעזרת הרדוקציה הפולינומיאלית) אלגוריתם מהיר לבדיקת השייכות ל-L_1.

השפות הקלות ב-NP

- **שאלה** : מיהן השפות הקלות ביותר במחלקה **NP** (ללא השפות טריוויאליות)?
  - לפי הסיווג של קל/קשה בעזרת היחס $\leq_P$

השפות הקלות ב-NP

- **שאלה**: מיהן השפות הקלות ביותר במחלקה **NP** (ללא השפות טריוויאליות)?
  - לפי הסיווג של קל/קשה בעזרת היחס $\leq_P$

תשובה: אלה השפות (הלא טריוויאליות) ב-**P**

- **הסבר**: אם L_1 היא שפה ב-**P**, אז $L_1 \leq_P L_2$ לכל L_2 במחלקה.
  - כל L_2 במחלקה קשה לפחות כמו L_1.
    - L_1 הקלה ביותר.

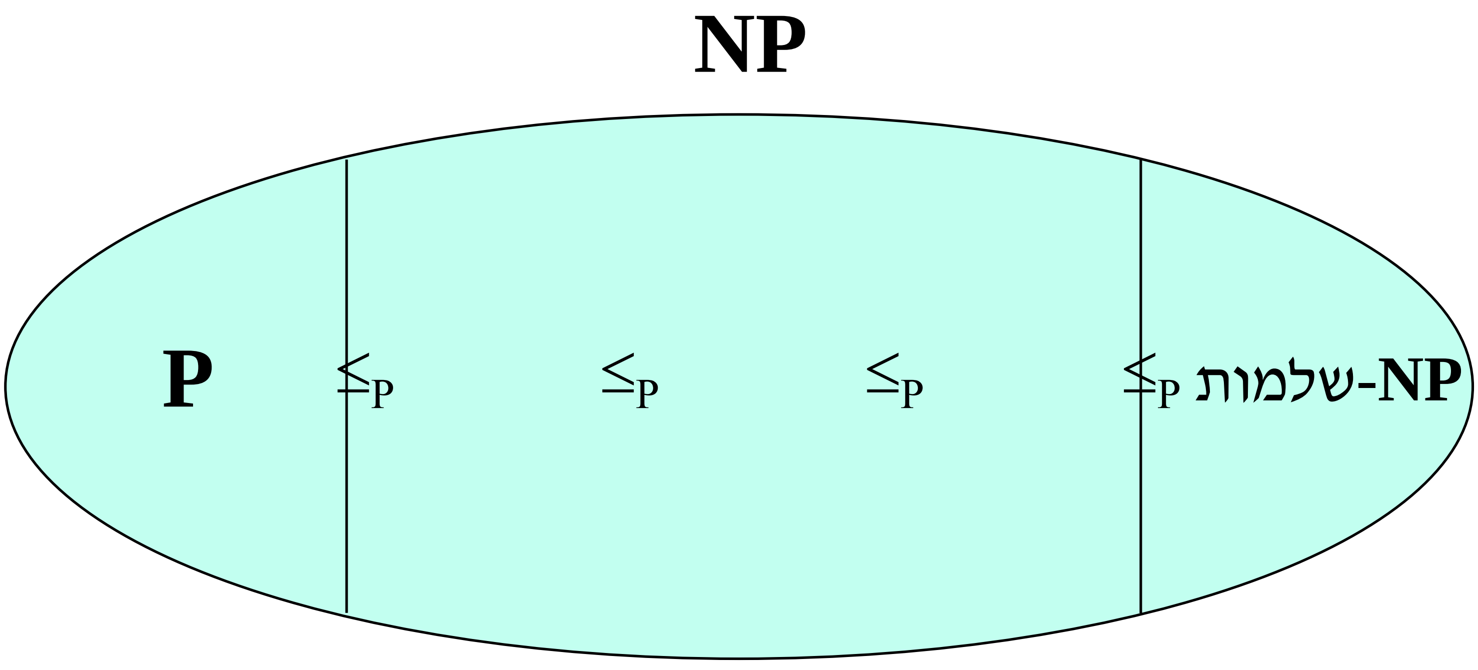
תזכורת: RE-complete

- **תזכורת**: שפה L_2 היא שלמה ב-RE אם
  - L_2 שייכת ל- RE (שייכת למחלקה)
  - לכל L_1 ב-RE, $L_1 \leq L_2$ (קשה ביותר במחלקה)
- השפות השלמות ב-RE הן הקשות ביותר במחלקה RE

שפות NP-שלמות

- **הגדרה** : שפה L_2 נקראת **NP**-שלמה אם
  - L_2 שייכת ל-**NP** (שייכת למחלקה)
  - לכל L_1 ב-**NP**, $L_2 \leq_P L_1$ (קשה ביותר במחלקה)
- מדובר על שפות ששייכות למחלקה, ולכל שפה במחלקה יש רדוקציה בזמן פולינומיאלי אליהן
  - הן קשות לפחות כמו כל שפה אחרת במחלקה

איור של המחלקה



איך נכריע האם P=NP

- **משפט** : אם יש שפה **NP**-שלמה ששייכת ל-**P**,
  אז **P = NP**

איך נכריע האם $P = NP$

- **משפט**: אם יש שפה $\boldsymbol{NP}$-שלמה ששייכת ל-$\boldsymbol{P}$, אז $\boldsymbol{P = NP}$.
- **תרגיל**: הוכיחו את המשפט.

איך נכריע האם $P = NP$

- **משפט**: אם יש שפה $\boldsymbol{NP}$-שלמה ששייכת ל-$\boldsymbol{P}$, אז $\boldsymbol{P = NP}$.
- **תרגיל**: הוכיחו את המשפט.
- **מסקנה**: אם רוצים להוכיח ש- $\boldsymbol{P = NP}$, די להראות מ״ט דטרמיניסטית בעלת זמן ריצה פולינומיאלי לאחת השפות ה-$\boldsymbol{NP}$ שלמות.
אם רוצים להוכיח ש- $\boldsymbol{P \neq NP}$, אפשר להוכיח על שפה L ב-$\boldsymbol{NP}$ שאין לה מ״ט דטרמיניסטית בעלת זמן ריצה פולינומיאלי. סביר ש-L תהיה $\boldsymbol{NP}$-שלמה (למה?)

יש שפות שלמות במחלקה

- נותרה השאלה האם יש בכלל שפות NP-שלמות?
  - האם יש שפות ב-NP כך שלכל שפה ב-NP יש רדוקציה בזמן פולינומיאלי אליהן?

יש שפות שלמות במחלקה

- נותרה השאלה האם יש בכלל שפות $\boldsymbol{NP}$-שלמות?
  - האם יש שפות ב-$\boldsymbol{NP}$ כך שלכל שפה ב-$\boldsymbol{NP}$ יש רדוקציה בזמן פולינומיאלי אליהן?
- **תשובה**: יש שפות כאלה. יש במחלקה $\boldsymbol{NP}$ שפות שלמות.
  - שפות קשות ביותר במחלקה.

יש שפות שלמות במחלקה

- נותרה השאלה האם יש בכלל שפות NP-שלמות?
  - האם יש שפות ב-NP כך שלכל שפה ב-NP יש רדוקציה בזמן פולינומיאלי אליהן?
- **תשובה** : יש שפות כאלה. יש במחלקה NP שפות שלמות.
  - שפות קשות ביותר במחלקה.
- השפה הראשונה שהוכח עליה שהיא NP-שלמה היא שפת הפסוקים הספיקים בתחשיב הפסוקים

תזכורות מהקורס בלוגיקה

- פסוק בתחשיב הפסוקים בנוי ממשתנים פסוקיים (אטומים) ומִקַּשְּׁרִים

תזכורות מהקורס בלוגיקה

- פסוק בתחשיב הפסוקים בנוי ממשתנים פסוקיים (אטומים) ומְקַשְּׁרִים
- המשתנים הפסוקיים (האטומים) יכולים לקבל ערך *true* (1) או *false* (0)

תזכורות מהקורס בלוגיקה

- פסוק בתחשיב הפסוקים בנוי ממשתנים פסוקיים (אטומים) ומְקַשְּׁרִים
- המשתנים הפסוקיים (האטומים) יכולים לקבל ערך *true* (1) או *false* (0)
- ערך האמת של הפסוק (*true* או *false*) נקבע לפי טבלאות האמת של הקשרים

תזכורות מהקורס בלוגיקה

- פסוק בתחשיב הפסוקים בנוי ממשתנים פסוקיים (אטומים) ומְקַשְּׁרִים
- המשתנים הפסוקיים (האטומים) יכולים לקבל ערך *true* (1) או *false* (0)
- ערך האמת של הפסוק (*true* או *false*) נקבע לפי טבלאות האמת של הקשרים
- אנחנו נסתפק בשלושת הקשרים $\neg$, $\wedge$, $\vee$
- שם הנוסחה הוא $\phi(x_1, x_2, \ldots, x_n)$

פסוקים ספיקים

- **דוגמה** : הפסוק
$\phi(x,y) = (x \vee \neg y) \wedge (\neg x \vee y)$
הוא פסוק מעל המשתנים x ו-y.

פסוקים ספיקים

- **דוגמה** : הפסוק
$$\phi(x,y) = (x \vee \neg y) \wedge (\neg x \vee y)$$
הוא פסוק מעל המשתנים x ו-y.

- **הגדרה** : פסוק נקרא ספיק אם יש לפחות השמה אחת של ערכי אמת למשתנים של הפסוק שבה ערך האמת של הפסוק הוא *true*.

פסוקים ספיקים

- **דוגמה**: הפסוק
$\phi(x,y) = (x \vee \neg y) \wedge (\neg x \vee y)$
הוא פסוק מעל המשתנים x ו-y.
- **הגדרה**: פסוק נקרא ספיק אם יש לפחות השמה אחת של ערכי אמת למשתנים של הפסוק שבה ערך האמת של הפסוק הוא *true*.
- **תרגיל**: האם הפסוק של הדוגמה ספיק?

פסוקים ספיקים

- **דוגמה**: הפסוק
$\phi(x,y) = (x \vee \neg y) \wedge (\neg x \vee y)$
הוא פסוק מעל המשתנים x ו-y.
- **הגדרה**: פסוק נקרא ספיק אם יש לפחות השמה אחת של ערכי אמת למשתנים של הפסוק שבה ערך האמת של הפסוק הוא *true*.
- **תרגיל**: האם הפסוק של הדוגמה ספיק?
- **תשובה**: בהחלט. הנוסחה מקבילה לשער *xnor* (מי שלא עשה מערכות ספרתיות – צר לי עליכם).

הפסקה

- תנוחו היטב, עוד מעט מתחילה אחת ההוכחות הכי ארוכות שראיתם בתואר.
- עם "שקפים מפוצלים" זה ייצא 32 שקפים!

השפה SAT

- השפה SAT היא שפת הפסוקים הספיקים בתחשיב הפסוקים:

$$SAT = \{<\phi> | \phi \text{ is a satisfiable Boolean formula}\}$$

- זוהי השפה הראשונה שהוכח עליה שהיא NP-שלמה.

SAT היא NP-שלמה

- **משפט** (Cook-Levin): SAT היא $\boldsymbol{NP}$-שלמה
- צריך להוכיח שני דברים:
  - SAT שייכת ל-$\boldsymbol{NP}$
  - לכל שפה $L \in NP$ מתקיים $L \leq_P SAT$

SAT היא NP-שלמה

- **משפט** (Cook-Levin): SAT היא $\boldsymbol{NP}$-שלמה
- צריך להוכיח שני דברים:
  - SAT שייכת ל-$\boldsymbol{NP}$
  - לכל שפה $L \in NP$ מתקיים $L \leq_P SAT$
- **תרגיל**: הוכיחו ש-SAT שייכת ל-$\boldsymbol{NP}$

SAT היא NP-שלמה

- **משפט** (Cook-Levin): SAT היא $\boldsymbol{NP}$-שלמה
- צריך להוכיח שני דברים:
  - SAT שייכת ל-$\boldsymbol{NP}$
  - לכל שפה $L \in NP$ מתקיים $L \leq_P SAT$
- **תרגיל**: הוכיחו ש-SAT שייכת ל-$\boldsymbol{NP}$
- כעת צריך להראות שלכל שפה L ב-$\boldsymbol{NP}$ יש רדוקציה בזמן פולינומיאלי ל-SAT

איך תיראה הרדוקציה?

- מה אנחנו יודעים על L?
  - לא הרבה. אנחנו יודעים שהיא ב-$\boldsymbol{NP}$.
  - כלומר, יש לה מכונה לא דטרמיניסטית מכריעה N שרצה בזמן $O(n^k)$ ל-k טבעי כלשהו.

איך תיראה הרדוקציה?

- מה אנחנו יודעים על L?
  - לא הרבה. אנחנו יודעים שהיא ב-$\boldsymbol{NP}$.
  - כלומר, יש לה מכונה לא דטרמיניסטית מכריעה N שרצה בזמן $O(n^k)$ ל-k טבעי כלשהו.
- איך תיראה הרדוקציה?
  - הרדוקציה תקבל כקלט מילה w מעל האלפבית של L.
  - הרדוקציה תבנה (בזמן פולינומיאלי ב-$|w|$) פסוק בתחשיב הפסוקים שיהיה ספיק אם ורק אם w שייכת ל-L.
  - כלומר, הפסוק ייבנה כך שהוא יהיה ספיק אם ורק אם יש ל-N חישוב מקבל על w.

תמונה של מסלול חישוב

- נניח אם כן שיש ל-L מכונה לא דטרמיניסטית מכריעה N שזמן הריצה שלה על מילה w חסום על-ידי n^k ($n = |w|$) ל-k טבעי כלשהו (קבוע).

תמונה של מסלול חישוב

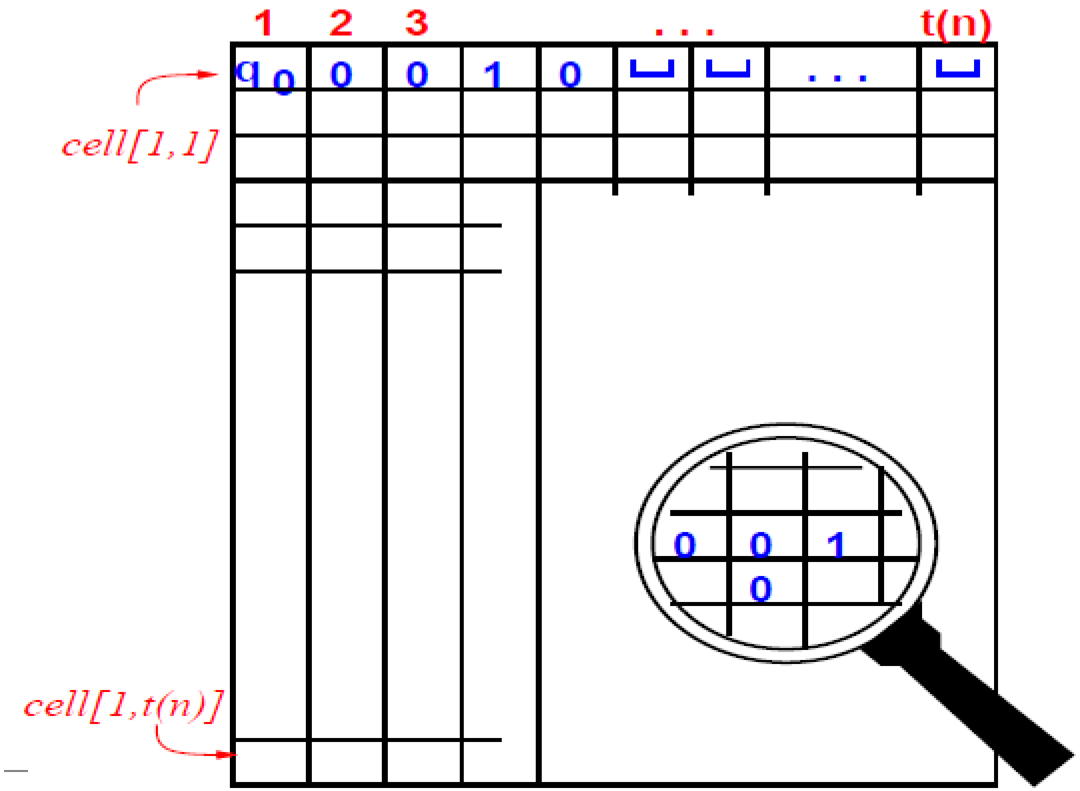
- נניח אם כן שיש ל-L מכונה לא דטרמיניסטית מכריעה N שזמן הריצה שלה על מילה w חסום על-ידי n^k ‏($n = |w|$) ל-k טבעי כלשהו (קבוע).
- בתוך n^k צעדים N יכולה להגיע לכל היותר לריבוע ה-$n^k + 1$ על הסרט שלה (מניחים של-N יש סרט יחיד)

תמונה של מסלול חישוב

- נניח אם כן שיש ל-L מכונה לא דטרמיניסטית מכריעה N שזמן הריצה שלה על מילה w חסום על-ידי n^k ($n = |w|$) ל-k טבעי כלשהו (קבוע).
- בתוך n^k צעדים N יכולה להגיע לכל היותר לריבוע ה-$n^k + 1$ על הסרט שלה (מניחים של-N יש סרט יחיד)
- תמונה של ריצת N על w באחד ממסלולי החישוב היא מטריצה (tableau) מסדר $(n^k + 2) \times \left(n^k + 1\right)$ -
  - השורה ה-i מתאימה לקונפיגורציה ה-i–1 בחישוב.
  - השורה הראשונה מתאימה לקונפיגורציה ההתחלתית.
  - השורה האחרונה (ה-$n^k + 1$) מתאימה לקונפיגורציה ה-n^k.

  - זה היה נראה לכם יותר הגיוני אם היינו עושים את Wang-Tiling. אבל החלטנו שהוא "אכזרי מדיי" לסמסטר קיץ שגם ככה דחוס מאוד...

Tableau



כמה הנחות

- לשם הנוחות, נניח שכל קונפיגורציה מתחילה ומסתיימת ב-#.
  - כלומר, בכל העמודה הראשונה ובכל העמודה האחרונה במטריצה מופיע הסמל #.
  - נרחיב את המטריצה לסדר $(n^k + 4) \times (n^k + 1)$.
    לשם הנוחות של האינדקסים בהמשך נניח שהיא מסדר $(n^k + 1) \times (n^k + 1)$.

כמה הנחות

- לשם הנוחות, נניח שכל קונפיגורציה מתחילה ומסתיימת ב-#.
  - כלומר, בכל העמודה הראשונה ובכל העמודה האחרונה במטריצה מופיע הסמל #.
  - נרחיב את המטריצה לסדר $(n^k+1)\times(n^k+4)$.
    לשם הנוחות של האינדקסים בהמשך נניח שהיא מסדר $(n^k+1)\times(n^k+1)$.
- גם נניח שאם מגיעים לקונפיגורציה עוצרת (קבלה או דחייה), מעתיקים אותה לשורות שתחתיה במטריצה.
  - כך נבטיח שתמיד המטריצה היא מסדר $(n^k+1)\times(n^k+1)$

מטריצה מקבלת

- המטריצה תיקרא מקבלת, אם אחת השורות שלה היא קונפיגורציה מקבלת
  - כלומר, המצב בקונפיגורציה הוא q_{accept}

מטריצה מקבלת

- המטריצה תיקרא מקבלת, אם אחת השורות שלה היא קונפיגורציה מקבלת
  - כלומר, המצב בקונפיגורציה הוא q_{accept}
- כל מטריצה מקבלת של N על w מתאימה לחישוב מקבל של N על w.

מטריצה מקבלת

- המטריצה תיקרא מקבלת, אם אחת השורות שלה היא קונפיגורציה מקבלת
  - כלומר, המצב בקונפיגורציה הוא q_{accept}
- כל מטריצה מקבלת של N על w מתאימה לחישוב מקבל של N על w.
- לכן השאלה האם N מקבלת את w שקולה לשאלה האם יש מטריצה מקבלת של N על w.

מה תעשה הרדוקציה

- הרדוקציה תבנה מ-w פסוק שערכו יהיה $true$ אם ורק אם יש מטריצה מקבלת של N על w.

מה תעשה הרדוקציה

- הרדוקציה תבנה מ-w פסוק שערכו יהיה *true* אם ורק אם יש מטריצה מקבלת של N על w.
- כעת נסביר איך ייראה הפסוק הזה.

מה תעשה הרדוקציה

- הרדוקציה תבנה מ-w פסוק שערכו יהיה $true$ אם ורק אם יש מטריצה מקבלת של N על w.
- כעת נסביר איך ייראה הפסוק הזה.
- מיהם הסמלים שיכולם להופיע במטריצה?
  - סמלים מן הקבוצה $C = Q \cup \Gamma \cup \{\#\}$.
  - שימו לב שגודל הקבוצה C איננו תלוי ב-w (אלא רק ב-N), ולכן הוא נחשב קבוע בחישוב זמן הריצה.

הפסוק שבונה הרדוקציה

- לכל i ו-j בין 1 ל-$n^k + 1$ ולכל s ב-C נגדיר משתנה $x_{i,j,s}$.
  - הכוונה היא שערכו יהיה $true$ אם ורק אם במקום ה-i,j במטריצה נמצא הסמל s.
- **תרגיל**: הראו שמספר המשתנים הוא $O(n^{2k})$
- הרדוקציה תבנה מ-w פסוק ϕ שהוא קוניונקציה (AND) של כמה פסוקים:

$$\phi = \phi_{cell} \wedge \phi_{start} \wedge \phi_{move} \wedge \phi_{accept}$$

ϕ_{cell}

- נפרט על כל אחד מן הרכיבים של הפסוק
- המשמעות של ϕ_{cell} היא שבכל תא במטריצה יש סמל אחד ויחיד מתוך C.
  - לכל i ולכל j יש s אחד ויחיד כך ש-$x_{i,j,s}$ הוא *true*.

    בניסוח של פסוק נכתוב:

$$\phi_{cell} = \wedge_{1 \le i,j \le n^k+1} \left[\left(\vee_{s \in C} x_{i,j,s} \right) \wedge \left(\wedge_{s,t \in C, s \ne t} \left(\neg x_{i,j,s} \vee \neg x_{i,j,t} \right) \right) \right]$$

ϕ_{start}

- המשמעות של ϕ_{start} היא שהשורה הראשונה של המטריצה היא הקונפיגורציה ההתחלתית של N על w.
  - בכל תא בשורה הראשונה במטריצה מופיע הסמל הנכון של הקונפיגורציה ההתחלתית. בניסוח של פסוק:

$$\phi_{start} = x_{1,1,\#} \land x_{1,2,q_0} \land x_{1,3,w_1} \land \cdots \land x_{1,n+2,w_n} \land x_{1,n+3,\_} \land \cdots \land x_{1,n^k,\_} \land x_{1,n^k+1,\#}$$

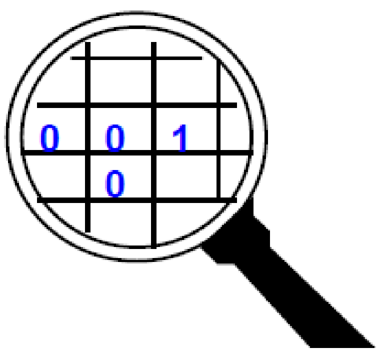
ϕ_{accept}

- המשמעות של ϕ_{accept} היא שיש במטריצה קונפיגורציה מקבלת.
  - הדרישה היא שהסמל q_{accept} מופיע בתא כלשהו של המטריצה. בניסוח של פסוק:

$$\phi_{accept} = \bigvee_{1 \le i,j \le n^k+1} x_{i,j,q_{accept}}$$

ϕ_{move}

- המשמעות של ϕ_{move} היא לוודא שכל שורה (קונפיגורציה) מאפשרת לעבור לשורה הבאה (קונפיגורציה עוקבת) דרך פונקציית המעברים של N.
  - הנקודה החשובה היא: ההבדל בין שתי קונפיגורציות עוקבות מתבטאת בלכל היותר בשלושה תאים רצופים. (מיקום הראש, והתאים שלימינו ולשמאלו). כל השאר זהה.
  - לכן די לוודא שכל "חלון" מסדר 3×2 הוא חוקי לפי N.



חלונות חוקיים ולא חוקיים

- **דוגמה**: נניח שב-N, $\delta(q_1,a) = \{(q_1,b,R)\}$
  $\delta(q_1,b) = \{(q_2,c,L),(q_2,a,R)\}$

חלונות חוקיים:

a	a	q_1
a	a	b

a	q_1	b
a	a	q_2

a	q_1	b
q_2	a	c

b	b	b
c	b	b

a	b	a
a	b	q_2

#	b	a
#	b	a

חלונות לא חוקיים:

b	q_1	b
q_1	b	q_2

a	q_1	b
q_1	a	a

a	b	a
a	a	a

הפסוק ϕ_{move}

- לכל חלון מסדר 2×3 יהיה ב-ϕ_{move} תת-פסוק שיאמר ששת הסמלים בחלון מהווים חלון חוקי

$$\phi_{move} = \bigwedge_{\substack{1 \le i < n^k+1 \\ 1 < j < n^k+1}} (the\ (i,j)\ window\ is\ legal)$$

הפסוק ϕ_{move}

- לכל חלון מסדר 3×2 יהיה ב-ϕ_{move} תת-פסוק שיאמר ששת הסמלים בחלון מהווים חלון חוקי

$$\phi_{move} = \bigwedge_{\substack{1 \le i < n^k+1 \\ 1 < j < n^k+1}} (the\ (i,j)\ window\ is\ legal)$$

כאשר הקביעה שהחלון הוא חוקי היא הפסוק הבא:

$$\bigvee_{a_1,\dots,a_6\ is\ a\ legal\ window} \left(x_{i,j-1,a_1} \wedge x_{i,j,a_2} \wedge x_{i,j+1,a_3} \wedge x_{i+1,j-1,a_4} \wedge x_{i+1,j,a_5} \wedge x_{i+1,j+1,a_6}\right)$$

(פסוקים כאלה מופיעים לכל i ולכל j ולכל חלון חוקי)

הפסוק ϕ_{move}

- לכל חלון מסדר 3×2 יהיה ב-ϕ_{move} תת-פסוק שיאמר ששת הסמלים בחלון מהווים חלון חוקי

$$\phi_{move} = \bigwedge_{\substack{1 \le i < n^k+1 \\ 1 < j < n^k+1}} (the\ (i,j)\ window\ is\ legal)$$

כאשר הקביעה שהחלון הוא חוקי היא הפסוק הבא:

$$\bigvee_{a_1,\dots,a_6\ is\ a\ legal\ window} (x_{i,j-1,a_1} \wedge x_{i,j,a_2} \wedge x_{i,j+1,a_3} \wedge x_{i+1,j-1,a_4} \wedge x_{i+1,j,a_5} \wedge x_{i+1,j+1,a_6})$$

(פסוקים כאלה מופיעים לכל i ולכל j ולכל חלון חוקי)

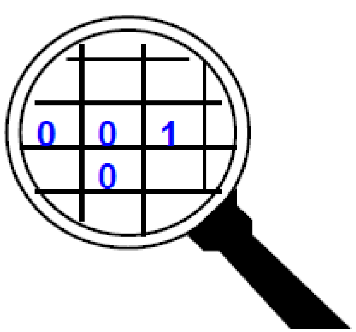
- שימו לב שמספר החלונות החוקיים איננו תלוי כלל ב-w (אלא רק במכונה N), ולכן הוא ייחשב לקבוע בחישוב זמן הריצה של הרדוקציה.

הנכונות של ϕ_{move}

- **תרגיל**: הוכיחו את הטענה הבאה:

  אם השורה הראשונה במטריצה היא הקונפיגורציה ההתחלתית של N על w, וכל חלון מסדר 2×3 במטריצה הוא חלון חוקי לפי N,
  אז כל שורה בטבלה היא קונפיגורציה עוקבת לקונפיגורציה של השורה שלפניה, בְּחישוב של N על w.

הדרכה: לכל סמל בקונפיגורציה שאיננו #, התבוננו בחלון שבו הסמל הזה הוא הסמל בשורה העליונה בחלון.



הרדוקציה תקפה

- כעת נוכיח שהרדוקציה תקפה
- **תרגיל**: הוכיחו:

$$\phi = \phi_{cell} \wedge \phi_{start} \wedge \phi_{move} \wedge \phi_{accept}$$

ספיק אם ורק אם w שייכת ל-L.

הרדוקציה תקפה

- כעת נוכיח שהרדוקציה תקפה
- **תרגיל**: הוכיחו:

$$\phi = \phi_{cell} \wedge \phi_{start} \wedge \phi_{move} \wedge \phi_{accept}$$

ספיק אם ורק אם w שייכת ל-L.

- תיארנו את הרדוקציה, והוכחנו שהיא תקפה (כלומר, הסטודנטים הרציניים ישבו בבית ויוכיחו). כעת נעבור לחישוב זמן הריצה של הרדוקציה, ונוכיח שהוא פולינומיאלי בגודל הקלט. זה יסיים את הוכחת משפט Cook-Levin.

הגודל של הפסוק ϕ

- נחשב את הגודל של ϕ
  - כמו שראינו, מספר המשתנים הוא $O(n^{2k})$
  - הגודל של ϕ_{cell} הוא $O(n^{2k})$
  - הגודל של ϕ_{start} הוא $O(n^{k})$
  - הגודל של ϕ_{move} הוא $O(n^{2k})$
  - הגודל של ϕ_{accept} הוא $O(n^{2k})$
- **מסקנה**: הגודל של ϕ הוא $O(n^{2k})$.

זמן הריצה של הרדוקציה

- אפשר לחשב את ϕ בזמן פולינומיאלי ב-$|w|$:
  - ϕ_{cell}, ϕ_{cell} ו-ϕ_{accept} לא תלויים ב-w עצמה, אלא רק ב-$|w|$ (כדי לקבוע את התחום של האינדקסים i ו-j).
  - הרכיבים של הפסוקים הללו חוזרים על עצמם, ולכן אפשר לחשב אותם בזמן פולינומיאלי.
  - את ϕ_{start} אפשר לחשב מתוך w בזמן פולינומיאלי.
- בזה תמה הוכחת משפט Cook-Levin:
  SAT היא $\boldsymbol{NP}$-שלמה!

Cook-Levin

- קצת ידע כללי:

א. יש עוד הוכחה למשפט, תפגשו אותה בקורס סיבוכיות (מי שימשיך לתואר שני).

ב. המשפט הוכח על ידי קוק בשנת 1971, ועל ידי לוין בשנת 1973.

ג. הם לא ידעו אחד על השני. בעוד שקוק היה אמריקאי, לוין היה מאחורי מסך הברזל.

ד. היות ושניהם הוכיחו את אותו המשפט, לחלוטין בנפרד, הוא נקרא על שם שניהם.

צורה קוניונקטיבית נורמלית

- **תזכורת**: פסוק בתחשיב הפסוקים הוא בצורה קוניונקטיבית נורמלית (CNF), אם
  - הפסוק בנוי מקבוצה של פסוקיות שביניהן יש הקשר $\wedge$
  - כל פסוקית היא קבוצה של ליטרלים שביניהם יש הקשר $\vee$
  - וכל ליטרל הוא משתנה או שלילה של משתנה
- **דוגמה**: $(x \vee y \vee \neg z) \wedge (x \vee z) \wedge (\neg y \vee \neg z)$
- **תזכורת**: לכל פסוק יש פסוק שקול ב-CNF.
- ליוצאי מערכות ספרתיות: $CNF \equiv POS$

CNF-SAT

- **תרגיל**: הראו שכל חלקי הפסוק ϕ בהוכחת משפט Cook-Levin, פרט ל-ϕ_{move}, הם בצורת CNF.
- אפשר להעביר גם את ϕ_{move} לצורת CNF.
  - זה יגדיל את הפסוק, אך מכיוון שהגודל של כל תת-פסוק של חלון ב-ϕ_{move} תלוי רק במכונה N (ולא ב-w), אפשר להתייחס להגדלה הזו כאל כפל בקבוע.

CNF-SAT

- **תרגיל**: הראו שכל חלקי הפסוק ϕ בהוכחת משפט Cook-Levin, פרט ל-ϕ_{move}, הם בצורת CNF.
- אפשר להעביר גם את ϕ_{move} לצורת CNF.
  – זה יגדיל את הפסוק, אך מכיוון שהגודל של כל תת-פסוק של חלון ב-ϕ_{move} תלוי רק במכונה N (ולא ב-w), אפשר להתייחס להגדלה הזו כאל כפל בקבוע.
- **מסקנה**: גם השפה $CNF - SAT$ היא $\boldsymbol{NPC}$:

$$CNF - SAT = \{< \phi > | \phi \ is \ a \ satisfiable \ CNF \ Boolean \ formula\}$$

DNF-SAT

- **תזכורת**: פסוק בתחשיב הפסוקים הוא בצורה דיסיונקטיבית נורמלית (DNF), אם
  - הפסוק בנוי מקבוצה של פסוקיות שביניהן יש הקשר $\vee$
  - כל פסוקית היא קבוצה של ליטרלים שביניהם יש $\wedge$
  - וכל ליטרל הוא משתנה או שלילה של משתנה

$$DNF - SAT \ = \ \{< \phi > \ | \phi \ is \ a \ satisfiable \ DNF \ Boolean \ formula\}$$

DNF-SAT

- **תזכורת**: פסוק בתחשיב הפסוקים הוא בצורה דיסיונקטיבית נורמלית (DNF), אם
  - הפסוק בנוי מקבוצה של פסוקיות שביניהן יש הקשר $\vee$
  - כל פסוקית היא קבוצה של ליטרלים שביניהם יש $\wedge$
  - וכל ליטרל הוא משתנה או שלילה של משתנה

$$DNF-SAT = \{<\phi> \mid \phi \; is \; a \; satisfiable \; DNF \; Boolean \; formula\}$$

- **תרגיל**: הוכיחו: $DNF-SAT$ שייכת ל-***P***
  - שימו לב שבין הפסוקיות יש הקשר $\vee$

האם גם $CNF\text{-}SAT$ ב- P ?

- **שאלה**: למה האלגוריתם הבא ל-SAT-$CNF-$ לא מוכיח שגם $CNF-SAT$ שייכת ל-$\boldsymbol{P}$?
  - "על קלט $<\phi>$ כאשר ϕ פסוק בצורת CNF:
    1. העבר את ϕ לפסוק שקול ϕ' בצורת DNF.
    2. בדוק האם ϕ' ספיק. אם כן, קבל. אם לא, דחה."

האם גם $CNF\text{-}SAT$ ב-P?

- **שאלה**: למה האלגוריתם הבא ל-$CNF - SAT$ לא מוכיח שגם $CNF - SAT$ שייכת ל-$\boldsymbol{P}$?
  - "על קלט $< \phi >$ כאשר ϕ פסוק בצורת CNF:
    1. העבר את ϕ לפסוק שקול ϕ' בצורת DNF.
    2. בדוק האם ϕ' ספיק. אם כן, קבל. אם לא, דחה."
- **תשובה**: המעבר מ-CNF ל-DNF עלול ליצור פסוק ϕ' שגודלו אקספוננציאלי בגודל של הפסוק ϕ
  –

לדוגמה

$$\phi = (x_1 \lor x_2) \land (\neg x_1 \lor \neg x_2) \land (x_3 \lor x_4) \dots \land (\neg x_{n-1} \lor \neg x_n)$$

כיצד ממירים את זה לצורת DNF?

- לעיון בדוגמה "קצרה":

https://math.stackexchange.com/questions/1710210/converting-formula-from-cnf-to-dnf