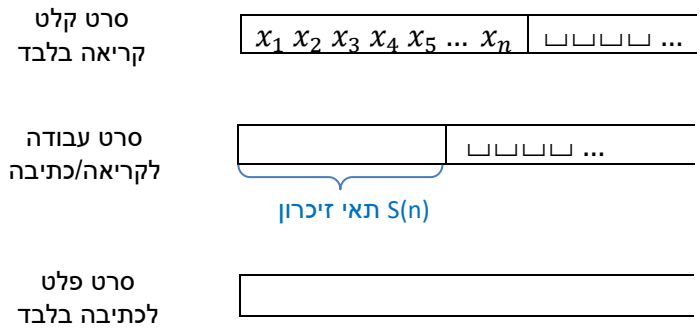
מודלים חישוביים	16.6.2025

הרצאה 12: סיבוכיות מקום

Based on previous iterations of this course, given by Nir Bitansky, Rotem Oshman, Iftach Haitner, and Omer Paneth.

מרצה: אורי שטמר

<u>המודל:</u> מ"ט עם 3 סרטים:



1. M משתמשת בסרט הקלט לקריאה בלבד. לא משנה בו תווים ולא חורגת מ- ␣ השמאלי ביותר.
2. בסרט העבודה M משתמשת בלכל היותר $S(n)$ תאי זיכרון ולא חורגת מתחום זה בזיכרון.
3. M משתמשת בסרט הפלט <u>לכתיבה בלבד</u> וכל פעם שהיא כותבת תו היא זז ימינה ואף פעם היא לא זזה בו שמאלה.

<u>הערה:</u> בסיבוכיות המקום רק סרט העבודה נספר. זה מאפשר לנו לדבר על מקרים בהם $S(n) < n$.

<u>דוגמה:</u> יהיו $A, B \in \{0,1\}^{n\times n}$ מטריצות ריבועיות. כמה מקום צריך בשביל לחשב $AB \bmod 2$?

<u>תשובה:</u> מתקיים $(AB)_{ij} = \sum_k A_{ik}\, B_{kj}$ לכן כדי לחשב את $(AB)_{ij}$ אנחנו צריכים לשמור רק את האינדקסים i, j, k על סרט העבודה. סה"כ זיכרון $O(\log n)$.

שימו לב כי במקרה זה $S(n) \ll$ input/output length

מוטיבציה למודל הזה: דמיינו שהקלט ענק ומאוחסן באינטרנט ואנחנו מריצים אלג' ממכשיר עם זיכרון מוגבל, למשל טלפון חכם.

<u>הגדרה:</u> תהי $S: \mathbb{N} \to \mathbb{N}$ ו- M מ"ט עם 3 סרטים: סרט קלט לקריאה בלבד, סרט עבודה, וסרט פלט לכתיבה חד פעמית. M רצה במקום $S(n)$ אם לכל $n \in \mathbb{N}$ ולכל קלט x באורך n, מתקיים ש- M משתמשת בלכל היותר $S(n)$ תאים בסרט העבודה בטרם עוצרת (בפרט תמיד עוצרת).

<u>הערה 1:</u> בחלק מהספרות לא דורשים שמכונה עם זיכרון מוגבל תמיד תעצור. כלומר תיתכן מכונה עם זיכרון קטן שנכנסת ללופ אינסופי. אבל כל עוד $S(n)$ היא מספיק "נחמדה" אז זה לא משנה כלום ואפשר להמיר מכונה עם זיכרון $S(n)$ שלא תמיד עוצרת למכונה עם זיכרון $O(S(n))$ שתמיד עוצרת. ראו הערה 2 בהמשך.

<u>הערה:</u> כשדיברנו על <u>זמן</u> ראינו שיכול להיות הבדל גדול בין זמן הריצה של מ"ט חד-סרטית לזמן הריצה של מ"ט דו-סרטית. מסתבר שלגבי זיכרון זה לא כ"כ משנה: ניתן לסמלץ מכונה k-סרטית הרצה במקום $S(n) \geq \log n$ בעזרת מכונה חד-סרטית עם מקום $O(S(n))$.
(הסיבה לדרישה $S(n) \geq \log n$ היא שכדי לבצע את הסימולציה המ"ט החד סרטית צריכה לזכור, בנוסף לתוכן של k הסרטים, גם את מיקום הראש בכל סרט. בשביל זה צריך $\log n$ לכל אינדקס כזה...)

<u>הגדרה:</u> תהי $S: \mathbb{N} \to \mathbb{N}$.
$$DSpace(S(n)) := \{L(M) : M \text{ מ"ט הרצה במקום } O(S(n))\}$$

מחלקות סיבוכיות מקום (דטרמיניסטיות):

$$PSPACE = \bigcup_{c \in \mathbb{N}} DSpace(n^c)$$

$$L = LOGSPACE = DSpace(\log(n))$$

$$DSpace(O(1)) = DSpace(o(\log\log(n)) = \{L : \text{רגולרית}\}$$

הערה: באופן אולי מפתיע, המחלקה L מכילה שפות לא טריוויאליות:
- **פלינדרומים:** $\{x : x = x^R\}$. זה מאוד פשוט להראות את זה: חשבו על מכונה שמחזיקה שני אינדקסים ("הקצוות הנוכחיים של הקלט") ובכל שלב משווה את הקצה הימני לשמאלי, דוחה אם הם שונים, ואחרת מקרבת את הקצוות בתא אחד מכל כיוון. צריך לזכור רק את שני האינדקסים.
- **קיום מסלול מ- s ל- t בגרף <u>לא</u> מכוון.** זה מאוד לא טריוויאלי להראות את זה, אבל זה נכון...

<u>מקום לעומת זמן:</u>
- $DTime(t(n)) \subseteq DSpace(t(n))$. מדוע? ב t צעדי ריצה אפשר להשתמש ב t תאי זיכרון לכל היותר
- $NP \subseteq PSPACE$. מדוע? במקום פולינומי אפשר לעבור על כל האפשרויות לעדים ולהריץ את אלג' הווידוא עם כל אחד מהם. בפרט $P \subseteq PSPACE$...

<u>טענה:</u> תהי $S(n) \geq \log(n)$. אזי $DSpace(S(n)) \subseteq DTime(2^{O(S(n))})$.

<u>מסקנות:</u>
- $L \subseteq P$
- $PSPACE \subseteq EXP$

<u>הוכחת הטענה:</u>
תהי $M = (Q, \Sigma, \Gamma, \dots)$ מ"ט הרצה במקום $S(n)$ ומכריעה שפה $A \subseteq \Sigma^*$.
בהינתן קלט $x \in \Sigma^{*n}$, נחסום את מספר הקונפיגורציות $c(n)$ אליהן $M(x)$ יכולה להגיע:
$$c(n) \leq \underbrace{n}_{\text{ראש קלט}} \times \underbrace{|\Gamma|^{S(n)}}_{\text{תוכן סרט עבודה}} \times \underbrace{S(n)}_{\text{ראש עבודה}} \times \underbrace{|Q|}_{\text{מצב}} \leq 2^{O(S(n))}$$
מכיוון ש- M תמיד עוצרת, מבצעת לכל היותר $c(n)$ צעדים. מ.ש.ל.

בדומה למשפט היררכיית הזמן, ניתן להוכיח משפט היררכיה למקום:

<u>הגדרה:</u> פונקציה $S:\mathbb{N}\to\mathbb{N}$ היא <u>חשיבה במקום</u> (space-constructible) אם קיימת מ"ט שבהינתן 1^n מחשבת את הקידוד הבינארי של $S(n)$ במקום $O(S(n))$.

<u>משפט:</u> תהי $S(n)\geq\log(n)$ פונק' חשיבה במקום. אזי
$$DSpace\left(o(S(n))\right)\subsetneq DSpace(S(n))$$

שימו לב שבניגוד להיררכיית הזמן, כאן אנחנו מקבלים הפרדה הדוקה. זה נובע מכך שיודעים לבצע סימולציה אוניברסלית עם תקורת מקום קבועה. מעבר לפרט זה ההוכחה דומה.

<u>מסקנה:</u> $L\subsetneq PSPACE$. לכן (מכיוון ש $L\subseteq P\subseteq PSPACE$), אז <u>לפחות</u> אחד מהתנאים הבאים נכון:

(א) $L\subsetneq P$ (ב) $P\subsetneq PSPACE$

מאמינים שגם (א) וגם (ב) נכונים.

<u>הערה 2 (המשך להערה 1):</u> נניח M מ"ט המשתמשת בזיכרון $S(n)\geq\log(n)$ אשר לא בהכרח עוצרת, עבור $S(n)$ פונק' חשיבה בזיכרון. אזי קיימת מ"ט M' המשתמשת בזיכרון $O(S(n))$ אשר תמיד עוצרת:

- כפי שראינו, עבור קלט x באורך n יש ל $M(x)$ לכל היותר $2^{O(S(n))}$ קונפ' אפשריות. לכן אפשר להוסיף למכונה "מונה צעדים" ואם המכונה ביצעה יותר צעדים מזה, אז אנחנו בלופ אינסופי ואפשר לדחות.
- תחזוק מונה כזה דורש $O(S)$ זיכרון (בהנחה ש- $S(n)$ בעצמה חשיבה בזיכרון).

<u>קשיות מקום:</u>
כפי שאמרנו, אנחנו יודעים שמתקיים $L\subseteq P$ ומאמינים ש- $L\subsetneq P$ אבל לא יודעים להוכיח את זה. במובן מסוים, בעיות ב L הן בעיות "קלות במיוחד" בתוך P מכיוון שבעיות אלו ניתן לפתור לא רק בזמן פולינומי אלא אפילו עם זיכרון לוגריתמי.
היינו רוצים לזהות בעיות "קשות" בתוך P שאותן אנחנו מאמינים שלא ניתן לפתור בזיכרון לוגריתמי.

נפעל במתודולוגיה דומה למה שעשינו עם NP: נזהה בעיות בתוך P שאם נדע לפתור אחת מהן במקום לוגריתמי אז נדע לפתור כל בעיה ב P במקום לוגריתמי. תחת האמונה ש $L\neq P$, זה יגרור שהשפות האלה לא ב L. לשם כך נרצה להגדיר מושג מתאים של רדוקציה:

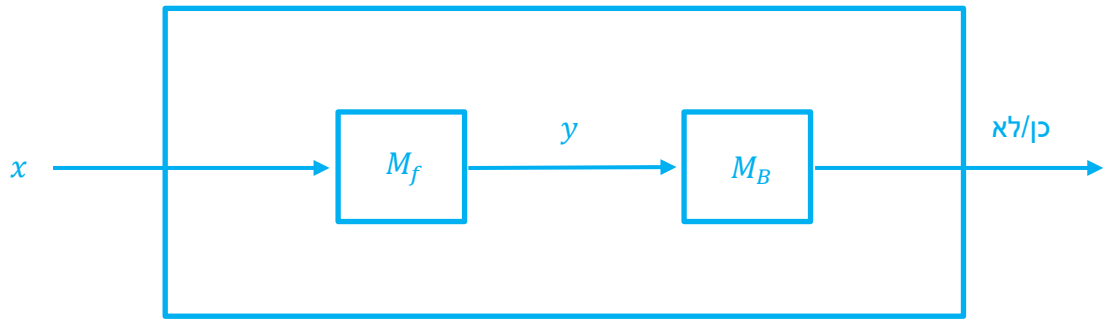
<u>הגדרה:</u> יהיו Σ_A,Σ_B אלפאבית ויהיו $A\subseteq\Sigma_A^*$, $B\subseteq\Sigma_B^*$ שפות. <u>רדוקצית מיפוי במקום לוגריתמי</u> מ- A ל- B היא פונקציה $f:\Sigma_A^*\to\Sigma_B^*$ חשיבה במקום לוגריתמי כך שלכל $x\in\Sigma_A^*$ מתקיים
$$x\in A\iff f(x)\in B$$

<u>סימון:</u> $A\leq_L B$

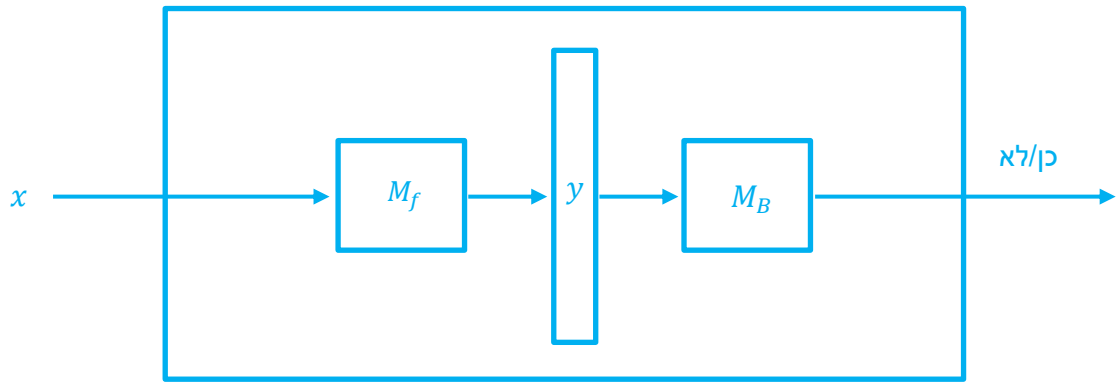
שימו לב: אם מתקיים $A\leq_L B$ אז גם מתקיים $A\leq_P B$.
מדוע? אם f חשיבה במקום לוגריתמי אז היא גם חשיבה בזמן פולינומי.

<u>טענה 3:</u> אם $A \leq_L B$ וגם $B \in L$ אז $A \in L$.

איך היינו רוצים להוכיח את הטענה הזאת? כשהוכחנו טענה דומה עבור P ו- NP עשינו משהו כזה:



הבעיה היא שכשאנחנו סופרים את כמות הזיכרון שמ"ט דורשת אנחנו לא סופרים את סרט הפלט וסרט הקלט. ברדוקציה הנ"ל עלינו לרשום את y איפשהו, אבל הוא עשוי ענק ביחס לכמות הזיכרון המותרת לנו (חשבו על בעיית הכפלת המטריצות כדוגמה). בציור:



רעיון הפתרון: נשים לב ש- M_B לא צריכה את כל y בבת אחת – הראש הקורא/כותב שלה "רואה" בכל שלב רק חלק קטן מ- y. לכן נבנה רדוקציה שתיתן ל- M_B כל פעם רק חלק מ y (ביט אחד או משהו כזה...).

<u>הוכחת הטענה:</u>
תהי M_B מ"ט המכריעה את B במקום לוגריתמי. תהי f רדוקצית מיפוי במקום לוגריתמי מ A ל- B. תהי M_f המ"ט המחשבת את f במקום לוגריתמי.

נגדיר מ"ט M'_f:
קלט: x ואינדקס i
פלט: הביט ה i ב- $f(x)$

M'_f מריצה את M_f ללא כתיבת הפלט. בכל שלב ש M_f כותבת ביט, M'_f מגדילה מונה ℓ.
כאשר $\ell = i$ היא מחזירה את הביט. אם M_f עצרה לפני שהגענו לביט ה i בפלט אז M'_f מחזירה $\sqcup$.

תיאור M_A על קלט x (זאת המ"ט המכריעה את A בזיכרון לוגריתמי):

- נריץ את M_B על $y = f(x)$ מבלי להחזיק את y בזיכרון באופן מפורש.
- בכל שלב נזכור את מיקום הראש של M_B בסרט הקלט שלה (בעזרת מונה i) ונחשב עבורה את ערך הביט שאמור להיות שם
  - בכל שלב ש M_B מזיזה את הראש ימינה נגדיל את i באחד ונריץ את M'_f על x ועל i (החדש) ונקבל y_i
  - כנ"ל אם M_B זזה שמאלה נבצע $i \leftarrow i-1$ ונחשב את הביט ה- i.

זיכרון M_A:

- סרטי העבודה של M_B: $\log|y|$.
  זכרו כי $L \subseteq P$ ולכן M_B רצה בזמן פולינומי ולכן $|y| \leq n^c$ ולכן $\log|y| = O(\log|x|)$.
- סרטי העבודה של M'_f (שהם סרטי העבודה של M_f): $\log|X|$
- המצביע i הוא מספר בין 1 לבין $|y|$ ולכן דורש $\log|y|$ ביטים.

סה"כ $O(\log|x|)$ מקום.

מ.ש.ל.

שאלה: מה זמן ריצת M_A?
תשובה: לכל ביט ב y אנחנו מריצים את M'_f.
כמה פעמים אנחנו ניגשים לביט מ y? כזמן הריצה של M_B. לכן:

$$[\text{זמן ריצת } M_A] = \underbrace{[\text{זמן ריצת } M_f]}_{\text{פולינומי}} \times \underbrace{[\text{זמן ריצת } M_B]}_{\text{פולינומי}} \quad \text{פולינומי}$$

טענה 4: אם $A \leq_L B$ וגם $B \leq_L C$ אז $A \leq_L C$. כלומר רדוקציות לוגריתמיות הן טרנזיטיביות!

הוכחת הטענה הזאת דומה להוכחת טענה 3. כלומר, כדי לחשב את $f_{AC}(x) = f_{BC}(f_{AB}(x))$ נריץ את M_{BC} על קלט $y = f_{AB}(x)$ מבלי להחזיק את y בזיכרון באופן מפורש.

הגדרה:

- שפה A_0 היא P-קשה (תחת רדוקציות מקום לוגריתמי) אם לכל $A \in P$ מתקיים $A \leq_L A_0$
- שפה A_0 היא P-שלמה אם היא P-קשה ובנוסף $A_0 \in P$

אבחנה: אם A_0 היא P-שלמה וגם $A_0 \in L$ אז $P = L$.

הסבר: תחת תנאי האבחנה, לכל $A \in P$ מתקיים $A \leq_L A_0$ ולכן, לפי טענה 3, מתקיים $A \in L$.

מסקנה: אם $P \neq L$ ו- A_0 היא P-קשה אזי $A_0 \notin L$.

דוגמה לבעיה P-שלמה:

הגדרה:

$CVAL = \{\langle c, x\rangle : \ c(x) = 1$ מעגל בוליאני כך ש $c\}$

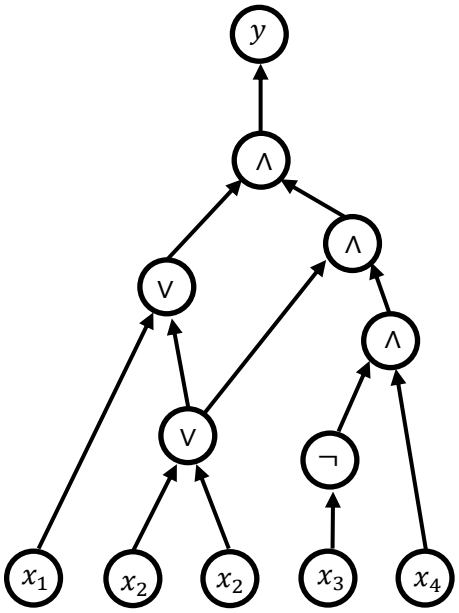
טענה: $CVAL$ היא P-שלמה.

דיי ברור ש- $CVAL \in P$ ולכן החלק המעניין בטענה הזאת הוא להראות ש $CVAL$ היא P-קשה.

איך רואים ש- $CVAL \in P$?
נמיין את צמתי הגרף במיון טופולוגי. כלומר אם יש קשת מצומת u לצומת v אז u יהיה לפני v במיון הטופולוגי. ניתן לעשות זאת בזמן פולינומי. לכל עלה ניתן ערך על פי ההצבה x. לכל שער על פי המיון הטופולוגי נחשב את ערכו על פי הערך שחושב כבר לצמתים עם קשתות נכנסות לשער.

האלג' שתיארנו פה הוא מאוד "מסודר" והוא שומר את כל הערכים שמחושבים לכל מוצא של שער לאורך המעגל. כלומר משתמש בזיכרון בערך לינארי. נראה שאין אלג' עם זיכרון לוגריתמי לבעייה הזאת...

תזכורת: מעגל בוליאני זה משהו שנראה ככה. זהו גרף מכוון חסר מעגלים מכוונים עם שערי AND,OR,NOT



נראה ש- $CVAL$ היא P-קשה. ההוכחה תהיה דומה להוכחת משפט קוק-לוין.

תהי $A \in P$ ונראה $A \leq_L CVAL$. לשם פשטות נניח כי $A \subseteq \{0,1\}^*$ כלומר ש- A היא שפה בינארית. זה בה"כ כי כל שפה נוכל לתרגם לשפה בינארית.

נתון כי $A \in P$ ולכן קיימת מ"ט פולינומית M שרצה בזמן n^k ומכריעה את A.

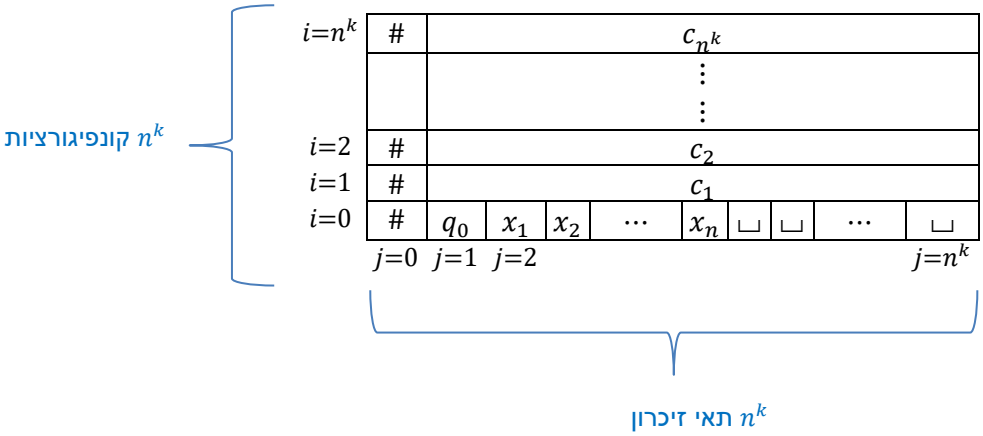
רעיון הרדוקציה: נראה רדוקציה לוגריתמית f אשר בהינתן קלט $x \in \{0,1\}^n$ מחזירה מעגל C_n מעל n משתנים ומחזירה את x (הקלט של הרדוקציה) בתור השמה לשערי הקלט של C_n. כלומר הרדוקציה מחזירה את הזוג (C_n, x).
אנחנו נבנה את C_n מתוך המ"ט M בצורה שמבטיחה שהשמה x מספקת את C_n אם"ם M מקבלת את x

לכן נקבל ש-

$$(C_n, x) \in CVAL \Leftrightarrow C_n(x) = 1 \Leftrightarrow M(x) \text{ מקבלת} \Leftrightarrow \ x \in A$$

שימו לב: בזמן חישוב הרדוקציה אנחנו לא יודעים אם $M(x)$ מקבלת ו/או אם $C_n(x) = 1$, כי יש לנו רק מקום לוגריתמי וכדי לחשב את הדברים האלה צריך (כנראה) מקום פולינומי.

בהינתן x באורך $|x| = n$ נסתכל על טבלת החישוב של M על x:



נסמן:

$$\Delta = \Gamma \cup Q \cup \{\#\}$$

התווים שיכולים להופיע בטבלה

במשפט קוק-לוין, לכל מקום בטבלה התאמנו $|\Delta| = O(1)$ משתנים. כאן נתאים $|\Delta| = O(1)$ שערים: לכל $0 \le i \le n^k$ לכל $0 \le j \le n^k$ ולכל $\sigma \in \Delta$ יהיה שער OR שנסמנו $u_{i,j,\sigma}$. השערים האלה יקודדו בשבילנו את תוכן התאים בטבלה.

בנוסף לשערי ה OR האלה, יהיו לנו שערי AND בין כל רמה i-1 לרמה i שידאגו לקונסיסטנטיות.

המעגל שנבנה C_n צריך להבטיח את הדברים הבאים. אם השמה x מספקת את C_n אז:

א. ההשמה הזאת <u>מגדירה טבלה</u>, במובן הזה שלכל i,j יהיה בדיוק σ אחד כך שהמוצא של השער $u_{i,j,\sigma}$ הוא True
ב. השורה הראשונה בטבלה היא הקונפ' ההתחלתית של M על אותו x
ג. לכל i, אם בשורה $i-1$ רשומה קונפ' c_{i-1} אז בשורה i רשומה הקונפ' c_i ש- M עוברת אליה בצעד אחד מ- c_{i-1}.
ד. המצב בקונפ' האחרונה הוא q_a

מכל אלה ביחד נקבל: אם השמה x מספקת את C_n אז M מקבלת את x.

הערה: את (א) נקבל בחינם משאר החלקים.

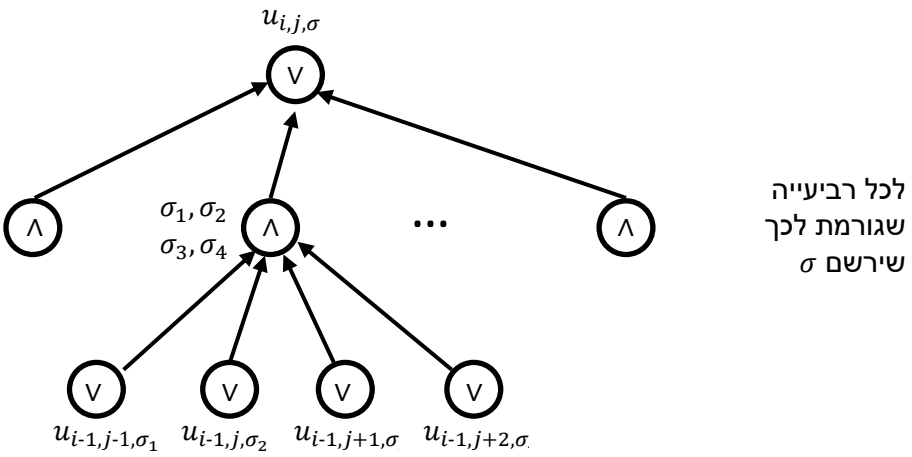
<u>חלק ג:</u> כפי שראינו בהוכחה של קוק-לוין, התו במקום ה (i,j) נקבע מתוך התווים במקומות $(i\text{-}1,j\text{-}1),(i\text{-}1,j),(i\text{-}1,j+1),(i\text{-}1,j+2)$. בציור:

	(i,j)		
$(i\text{-}1,j\text{-}1)$	$(i\text{-}1,j)$	$(i\text{-}1,j+1)$	$(i\text{-}1,j+2)$

כלומר, לכל רביעייה של תווים $\sigma_1,\sigma_2,\sigma_3,\sigma_3$ <u>קיים</u> σ <u>יחיד</u> שיהיה רשום במקום ה j בקונפ' ה i.

כלומר:

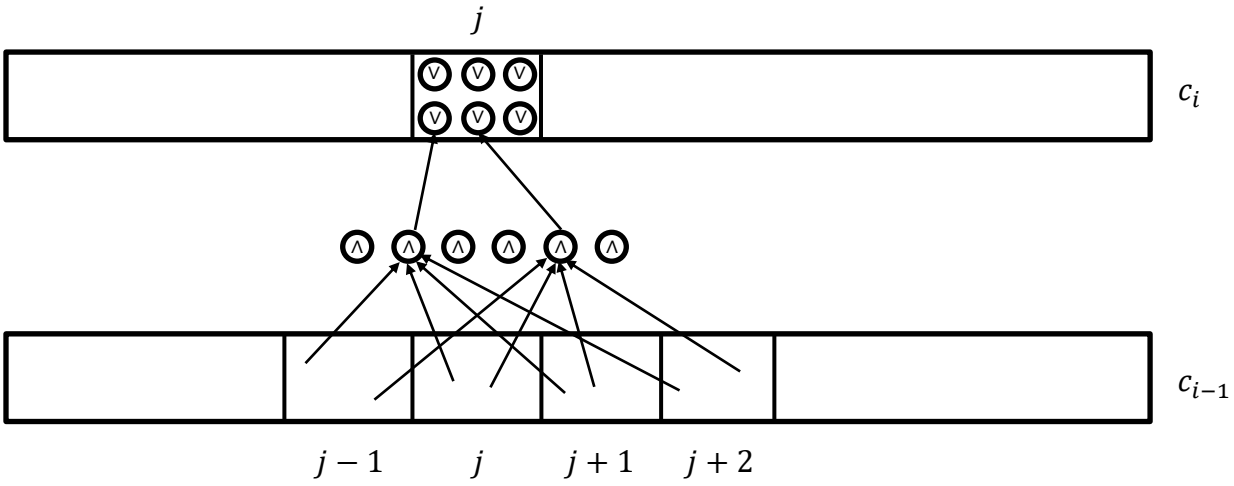
- ברגע שהשורה ה (i-1) נקבעה (כלומר לכל j קיים ויחיד σ כך שהמוצא של $u_{i,j,\sigma}$ הוא Ture)
- אז השורה ה i צריכה להיקבע מתוכה באופן חד משמעי. בנוסף, כדי לקבוע את תוכן התא (i,j) מספיק להסתכל ב 4 תאים שמתחתיו.
- לכן, כדי לאכוף את זה במעגל שאנחנו בונים, אנחנו רוצים לחבר את שערי ה OR שמתאימים לתא (i,j) לשערי ה OR שמתאימים לתאים (i-1,j+2),(i-1,j-1),...,(i-1,j-2) בצורה כזאת שתבטיח:
- אם $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ הם התווים כך ש- $u_{i\text{-}1,j\text{-}1,\sigma_1} = u_{i\text{-}1,j,\sigma_2} = u_{i\text{-}1,j+1,\sigma_3} = u_{i\text{-}1,j+2,\sigma_4} = True$
- אז $u_{i,j,\sigma} = True$ עבור התו σ ש- $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ גוררים, ולכל $\tilde{\sigma}$ אחר מתקיים $u_{i,j,\tilde{\sigma}} = True$



כלומר, בכל תא בטבלה יש לנו $|\Delta|$ שערי OR שכל אחד מהם מחובר לכמה שערי AND. דרגת הכניסה של כל שער AND היא 4. דרגת הכניסה של כל שער OR היא לכל היותר $|\Delta|^4 = O(1)$.

אז על כל שער OR יש לנו $O(1)$ שערי AND מתחתיו. יש לנו $O(n^{2k})$ שערי OR (מס' קבוע של שערי OR לכל תא בטבלה) ולכן סה"כ מס' השערים במעגל הוא $O(n^{2k})$.

ציור נוסף:



מה רואים בציור הזה? לכל תא בטבלה יש לנו |Δ| שערי OR (שער OR לכל תו אפשרי ב Δ). אנחנו נרצה לדאוג שלכל תא קיים ויחיד שער OR שהפלט שלו הוא 1.
נניח שהתכונה הזאת מתקיימת בשורה i-1. זה קובע את הערך של כל השורה הזאת.
אנחנו רוצים לדאוג שהקביעה הזאת גם קובעת את הערכים לשורה i בצורה נכונה.
איך אנחנו עושים את זה?
נסתכל על איזשהו שער OR בתא (i,j), למשל $u_{i,j,\sigma}$. לאן אנחנו מחברים אותו? זה מופיע בציור הראשון מבין שני הציורים הנ"ל. לכל רביעיית תווים שגוררים את σ אנחנו מחברים את $u_{i,j,\sigma}$ לשער AND שהכניסות שלו הם 4 שערי ה OR מהשורה i-1 שמתאימים לתווים הגוררים האלה.

- עכשיו ננית שבשורה i-1 במקומות (i-1,j+2),(i-1,j-1),...,(i-1,j) רשומים תווים a,b,c,d וננית שהם גוררים תו e. אנחנו רוצים להראות ש- $u_{i,j,e}$ "ידלק" ושהוא היחיד "שידלק" בתא (i,j):
- אנחנו יודעים שהשערים $u_{i\text{-}1,j\text{-}1,a}$, $u_{i\text{-}1,j,b}$, $u_{i\text{-}1,j+1,c}$, $u_{i\text{-}1,j+2,d}$ "דולקים"
- בנוסף, מכיוון שהם גוררים e אז אחת הכניסות לתוך $u_{i,j,e}$ היא משער $AND\left(u_{i\text{-}1,j\text{-}1,a}\ ,\ u_{i\text{-}1,j,b}\ ,\ u_{i\text{-}1,j+1,c}\ ,\ u_{i\text{-}1,j+2,d}\right)$

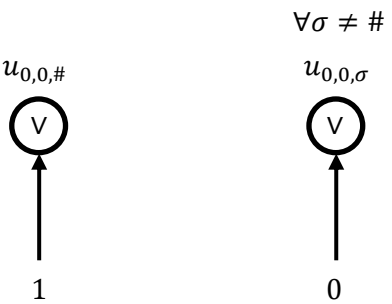
(כי לתוך $u_{i,j,e}$ יש כניסות משערי AND על כל רביעייה אפשרית שגוררת אותו)

- לכן גם $u_{i,j,e}$ "נדלק"
- ומדוע הוא יהיה היחיד "שנדלק"? אם נדלק איזשהו $u_{i,j,\gamma}$ אחר, אז אחד ה AND-ים שמחוברים אליו נדלק. ולכן ארבעת שערי ה OR מהשורה (i-1) שמחוברים לשער ה AND הזה נדלקים. אבל שערי ה OR היחידים שנדלקים פה הם $u_{i\text{-}1,j\text{-}1,a}$, $u_{i\text{-}1,j,b}$, $u_{i\text{-}1,j+1,c}$, $u_{i\text{-}1,j+2,d}$ והתו היחיד שהם גוררים הוא e ולכן $u_{i,j,\gamma}$ לא מחובר ל $AND\left(u_{i\text{-}1,j\text{-}1,a}\ ,\ u_{i\text{-}1,j,b}\ ,\ u_{i\text{-}1,j+1,c}\ ,\ u_{i\text{-}1,j+2,d}\right)$ ולכן הוא לא נדלק.

<u>טענה 3:</u> אם שערי ה OR ברמה (i-1) מתאימים לקונפ' חוקית של M אז שערי ה OR ברמה i מתאימים לְקונפ' ש- M עוברת אליה בצעד אחד ממנה.

<u>חלק ב:</u>
את הקונפ' ההתחלתית אנחנו יכולים לקודד בצורה מפורשת. כלומר לקבוע את הכניסות של השערים שמתאימים לשורה של הקונפ' c_0. זה יותר קל ממשפט קוק לוין כי אין לנו עד שאנחנו צריכים לקודד אותו והכל נתון לנו:

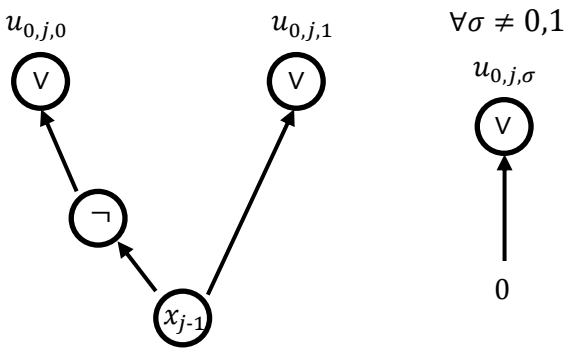
- אנחנו רוצים שבתא 0,0 יהיה רשום #:



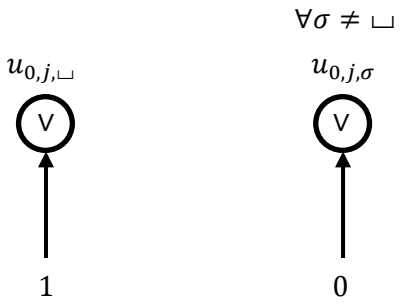
- כנ"ל לגבי התא 0,1 והתו q_0

- עבור $2 \leq j \leq n+1$:

הנחנו ש- A היא שפה בינארית,
כלומר ש- $A \subseteq \{0,1\}^*$



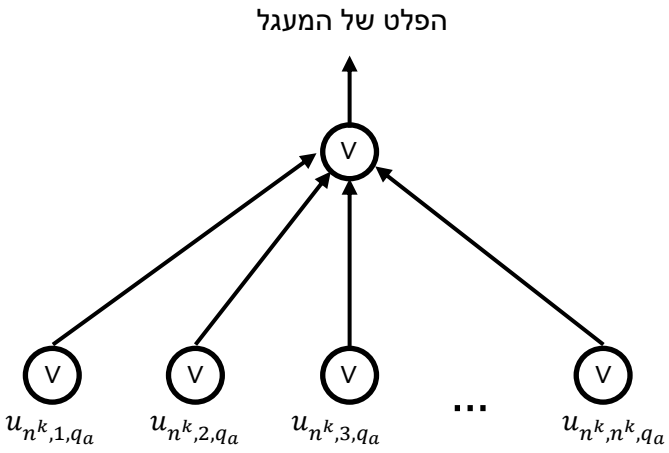
- עבור $n+2 \leq j \leq n^k$



<u>טענה 2:</u> כאשר משערכים את המעגל C_n על x אז שערי ה OR ברמה 0 מתאימים לְקונפ' ההתחלתית של M על x.

<u>מטענות 3+2 אנחנו מקבלים:</u> כאשר משערכים את המעגל C_n על x אז לכל i שערי ה OR ברמה i מתאימים לְקונפ' ה-i-ית בחישוב של M על x.

<u>חלק ד:</u>
באחד התאים בקונפ' האחרונה רשום q_a
(השער שמתאים לתו $\sigma = q_a$ מקבל True)



<u>סה"כ נקבל:</u> כאשר משערכים את המעגל C_n על x אז לכל i שערי ה OR ברמה i מתאימים לְקונפ' ה-i-ית בחישוב של M על x. אם $x \in A$ אז בקונפ' האחרונה מופיע מצב מקבל ולכן $C_n(x) = 1$. אם $x \notin A$ אז בקונפ' האחרונה לא מופיע מצב מקבל ולכן $C_n(x) = 0$.

סה"כ $x \in A$ אם"ם $(C_n, x) \in CVAL$.

<u>שאלה:</u> אז איך קיבלנו את (א) בחינם? בשורה הראשונה דאגנו "ידנית" שבדיוק תו אחד בכל תא יקבל T. הבניה שלנו בחלק ד דואגת שזה יפעפע גם לשורות הבאות.

<u>בניית המעגל מתבצעת בזיכרון לוגריתמי:</u>
עבור $1 \le i \le n^k$
עבור $1 \le j \le n^k$
עבור כל $\sigma \in \Delta$
בנה את השער עבור $u_{i,j,\sigma}$ כולל שערי AND

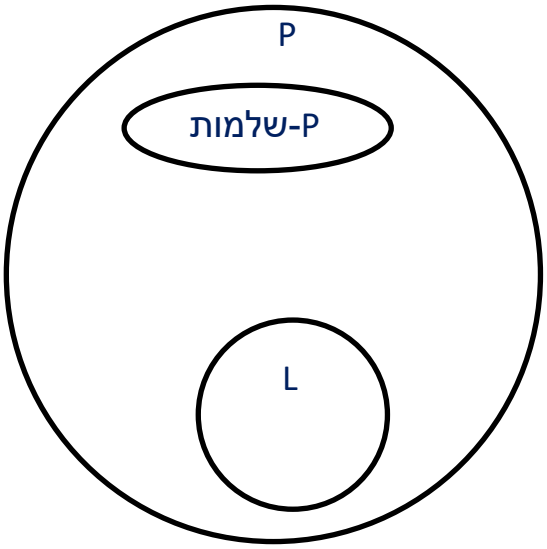
זיכרון נדרש: החזקת המונים i, j מה שדורש $O(\log(n^k)) = O(\log(n))$ זיכרון.

נקודה למחשבה: כפי שאמרנו, הרדוקציה שהראינו מקבלת קלט $x \in \{0,1\}^n$, בונה מעגל C_n מעל n משתנים ומחזירה (C_n, x) כך שיתקיים $x \in A$ אם"ם $C_n(x) = 1$.
שאלה: האם המעגל C_n תלוי בקלט x?
תשובה: לא! לכל קלט באורך n הרדוקציה בונה אותו מעגל C_n. כלומר הרדוקציה תלויה רק באורך של x

כלומר בעצם מה שהראינו זה דרך לקחת מ"ט M ולהפוך אותה <u>לסדרת מעגלים</u> (כי המעגל שבנינו תלוי רק באורך הקלט). פרטים נוספים באפנדיקס.

ישנן שפות P-שלמות נוספות מעניינות כגון תכנון לינארי (בגרסת הכרעה). שפות P-שלמות נותנות דוגמאות לשפות שלא סביר שניתן לפתור במקום קטן.

תמונה חלקית של העולם שלנו:



בהרצאה הבאה אנחנו ננסה להעשיר את התמונה הזאת בכך שנדבר על דברים שהם בין P-שלמות לבין L. אנחנו נגדיר מחלקה נוספת, NL, ונראה שהיא מוכלת בתוך P ומכילה את L.

אפנדיקס

הרדוקציה שראינו לוקחת מ"ט M שרצה בזמן $t = n^k$ ומחזירה מעגל C_n בגודל $O(t^2) = O(n^{2k})$ מעל n משתנים כך שלכל x באורך n מתקיים $C_n(x) = 1$ אם"ם $M(x)$ מקבלת. אותה בניה מראה את הלמה הבאה:

<u>למה:</u> תהי $t(n)$ חשיבה בזמן ותהי M מ"ט הרצה בזמן $t(n)$. קיימת פונק' חשיבה במקום לוגריתמי שבהינתן 1^n מחשבת (קידוד) מעגל $C_{M,n}: \{0,1\}^n \to \{0,1\}$ המקיים:

(1) לכל $x \in \{0,1\}^n$ מתקיים ש- $C_{M,n}(x) = 1$ אם"ם $M(x)$ מקבלת.

(2) $\left|C_{M,n}\right| \leq O\left(t^2(n)\right)$

<u>מסקנה:</u> תהי $t(n)$ חשיבה בזמן. אם $f: \{0,1\}^* \to \{0,1\}$ אינה ניתנת לחישוב ע"י משפחת מעגלים בגודל $O\left(t(n)\right)$ אזי f לא ניתנת לחישוב על ידי מ"ט שרצה בזמן $\sqrt{t(n)}$.

<u>הוכחת המסקנה:</u> אם קיימת מ"ט M שמחשבת את f בזמן $\sqrt{t(n)}$ אז משפחת המעגלים $\{C_{M,n}\}_{n\in\mathbb{N}}$ בגודל $O\left(t(n)\right)$ מחשבת את f. סתירה.

<u>Blast from the past:</u> המסקנה הזאת מפרמלת בשבילנו את המשפט החצי-פורמלי שראינו בסוף הרצאה 1, שאמר לנו שאם פונקציה מסוימת היא "קשה" למעגלים אז היא גם קשה למ"ט (ולכן גם קשה לפייתון).