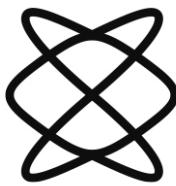


החולג למדעי המחשב (0368)
מודלים חישוביים (2200)
(גרסה ארוכה)

מרצה: ניר בטנסקי
מתרגלים: גל מאור, שמעאל יוסף אמוניאל
תשפ"ד, סמסטר א' (2024)

מסכם: רועי מעין



The Raymond and
Beverly Sackler Faculty
of Exact Sciences
Tel Aviv University



פרק 1 – מודלים בסיסיים

3	מעגלים בוליאניים.....
7	אוטומטים סופיים.....
15	שפות רגולריות.....

פרק 2 – תורת החישוביות

20	מכונות טיורינג.....
29	מחלקות חישוביות.....
35	רדווקציות.....

פרק 3 – תורת הסיבוכיות

43	סיבוכיות זמן.....
51	בעיות ספריקות.....

1 – מודלים בסיסיים

מעגלים בוליאניים

מבוא

מושגים בסיסיים:

- **אלפבית (א"ב)**: קבוצה סופית של תווים. נסמן Σ .
 - $\{0,1\} = \Sigma$ קלט בינארי.
 - $\{z, a, b, \dots\} = \Sigma$ אותיות השפה האנגלית.
- **מילה**: מילה מעל אלפבית Σ היא שרשרת של מספר סופי של תווים מ- Σ .
 - נסמן Σ^* את אוסף כל המילים מעל Σ . כך נקבל למשל $\{\dots, 0, 0, 1, 0, 0, 0, 1, \dots, \varepsilon\} = \Sigma^*$.
 - נסמן Σ^n את אוסף כל המילים מעל Σ באורך n .
 - נסמן Σ^{-} את המילה הריקה.
- **שפה**: שפה מעל א"ב Σ היא תת-קבוצה של Σ^* , יכולה להיות סופית או אינסופית.
 - לדוגמה, השפה האנגלית.
 - $\{p \in \mathbb{N} : p \text{ is prime}\}$
 - $\{n \in \mathbb{N} : n^n \text{ מחזורות שמקבילות אפסים ואז אחדים בכמויות שווה}\}$.
 - נסמן \emptyset את השפה הריקה.
- **בעיית הכרעה**: "האם קלט $\Sigma \in x$ שייך לשפה L ?" נאמר כי אלגוריתם מכיריע שפה L אם לכל $x \in L$ הוא מקבל (true) ולכל $x \notin L$ הוא דוחה (false).

מעגל בוליאני

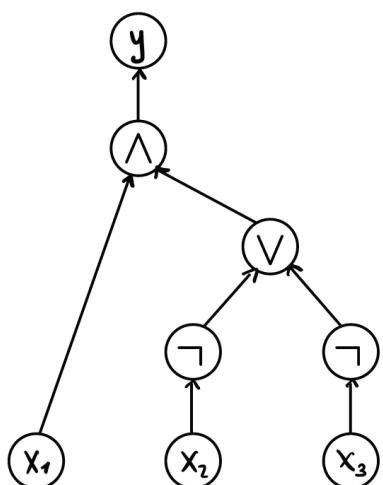
מדובר במודל חישוב לא-ווניפורמי (גודל המעגל תלוי בגודל הקלט, כלומר לכל גודל קלט צריך מעגל נפרד, אלגוריתם שונה), אשר דומה לחומרה שוכלונו מכירים (כמו שלמדו בקורס מבנה מחשבים): רצף של טעירים לוגיים שמחוברים אחד לשני. זאת לעומת מודל חישוב ווניפורמי (אלגוריתם ייחיד שעובר לכל אורך קלט).

מעגל בוליאני C מחשב פונקציה $m: \{0,1\}^n \rightarrow \{0,1\}^m$. לפונקציה יש n ביטי קלט ו- m ביטי פלט. הוצאה שבה החישוב מתבצע, היא על ידי הריבבה של פעולות לוגיות פשוטות על בסיס דה-מורגן (\neg : NOT; AND: \wedge ; OR: \vee).

x_1	x_2	$x_1 \wedge x_2$	$x_1 \vee x_2$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

דוגמה לפונקציה: $(x_1 \wedge x_2) \vee x_3 = x_1 x_2 + x_3$, ניתן לתאר אותה גם בתרשימים הבא. ביטי הקלט מסומנים למטה, החוקים נעים לתוך השערים הלוגיים, עד שהם מתכנסים לפלט y . בibold;ן זהן אינו המעגל היחיד שמחשב את הפונקציה זו, אפשר באמצעות דה-מורגן למצאו מעגל אחר.

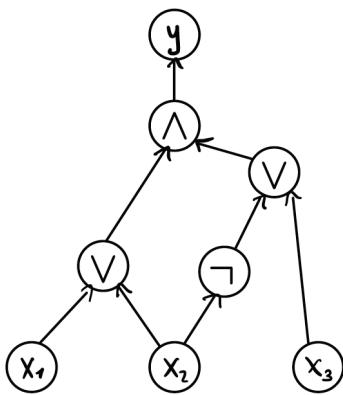
מעגל בוליאני: תהיו B קבוצה של פונקציות בוליאניות, עליה נחשב בתור הבסיס שלהם. מעגל בוליאני מעל B עם ביטי קלט $x = x_1, \dots, x_n$, ביטי פלט $y = y_1, \dots, y_m$ הוא גראף **מכוון חסר מעגלים** (כדי שהחישוב תמיד יסתתיים) המקיים את התכונות הבאות:



- כל צומת מסומן ע"י בית קלט x_i , בית פלט y_j , או שער $B \in g$.
- לכל בית פלט y_j , **בדוק צומת אחד** מסומן ע"י y_j עם דרגת **בנisa 1** ודרגת **יציאה 0** (הוא מסיים את החישוב שלו).
- לכל בית קלט x_i , דרגת **הבנייה** שלו היא **0** (הם מתחילה את החישוב שלנו).
- בנגדוד לביטי הפלט, אין מוגבלה לצומת אחד עבור כל בית קלט, יכולם להיות יותר ואפשר לשכפל אותם.
- אין מוגבלה על דרגת היציאה.
- לכל צומת המסומן בפונקציה $B \in g$, אם g מוגדרת על k ביטים $\{0,1\}^k$, אז דרגת הבניה שלה היא k . כל קשת הנכנסת לצומת מקבלת אינדקס $[k] \in i$ (בין 1 ל- k).
- בדה-מורגן אין חישבות לסדר המשתנים, הפעולות הן קומוטטיביות: $x_1 \wedge x_2 = x_2 \wedge x_1$.

מונחים:

- גודל המעלג: מספר השערים בו.
 - שער (gate): צומת שמסמן בפונקציה $B \in g$.
 - חוט (wire): קשת בין צמתים.
 - **fanout**: עבר צומת מודבר על דרגת היציאה של הצומת (כמה קשיות יוצאות ממנו). ה-
 - **fanout** של מעגל הוא דרגת היציאה המקסימלית של צומת בו.
 - **נוסחאות**: מעיגלים עם fanout שווה ל-1 (ambil צומת יוצאה רק קשת אחת).
- דוגמה לפונקציה נוספת: $(x_1 \vee x_2) \wedge (x_3 \vee x_4)$. המעגל זהה אינו נסחה, אפשר להפוך אותו בקלה לנוסחה באשר נרשום שני עותקים של x_2 , ובכ"ה fanout יהיה 1.

חישוב בمعالג:

בהינתן מעגל C עם n ביט קלט, וקלט $\{0,1\}^n \rightarrow u$, נרצה לשערר את המעלג. מציבים ערכים לחוטים באופן איטרטיבי:

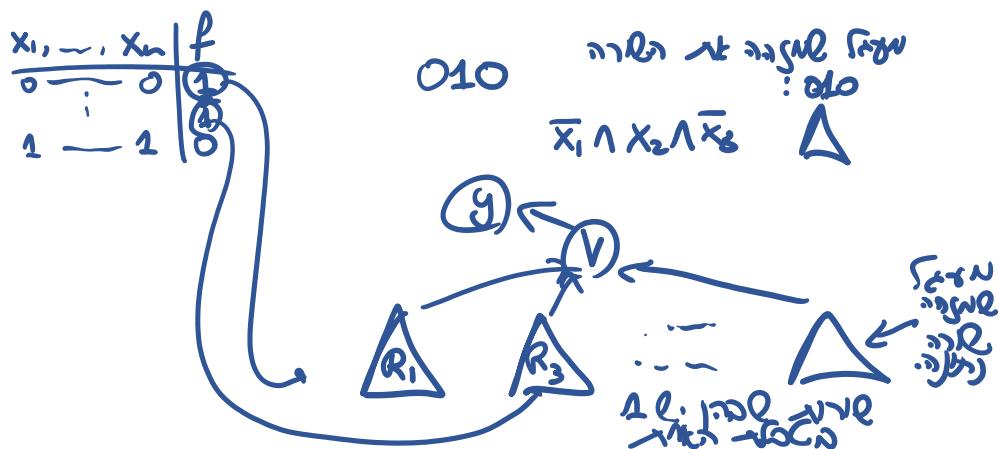
- מציב כל ביט קלט על החוט שיזוא מהצמת המתאים לו i (אם יש ביט קלט שמיוצג על ידי מספר צמותים, מציב בכלום).
- כל עוד אפשר: נחפש שער שהכניות שלו בבר נקבעו והפלט עוד לא נקבע, ואז נשערר אותן. כך הלאה עד שנתקבל את הערכים עבור החוטים שנקבעו לצמת פלט y .
- הפלט (u) הוא הערכים שהצבענו לבניוסות של $y_m \dots y_1$.

טענה: התהיליך מוגדר היטב, נתונים ערך ייחיד לכל חוט בمعالג. بيان שהגרף הוא חסר מעיגלים, לא ניתקל במצב שבו אנחנו צריכים ביט של קלט שטרם חושב.

האוניירסאלות של דה-מורגן: כל $\{0,1\}^n \rightarrow f: f(x) = \bigvee_{x \in f^{-1}(1)} x = \bigwedge_{x \in f^{-1}(0)} \neg x$

הוכחה:

- נסתכל על טבלת האמת של הפונקציה: התוצאה של הפונקציה עברור כל קומבינציה של ביטי הקלט. נזהה את כל השורות $\{0,1\}^n \rightarrow u$ בטבלה שבהן הפונקציה מוחזרת 1 לפחות 1 פעמיים.
- נבנה מעגל I_u שמצויה את הקלט $u = x = 1 \Leftrightarrow f(x) = 1$. למשל עבור I_{001} נמשך $x_3 \wedge x_2 \wedge x_1$.
- נבנה מעגל שמחשב את f : ניקח את כל המעלגים שמצוים את השורות שמניבו 1, ובבצע OR ביניהם. לכל $(1) \in f^{-1}(1)$ ניקח את $(x) \in I_{v \in f^{-1}(1)}$.



הערה: אפשר למשתמש $7,8$ על יותר מ-2 ארגומנטים באמצעות הפעלה חוזרת של השערים על 2 ארגומנטים ואז על התוצאה שלהם והשלישי וכן הלאה.

מסקנה: לכל פונקציה $m: \{0,1\}^n \rightarrow \{0,1\}$ יש מעגל מעל בסיס דה-מורגן שמחשב את f .

משפחת מעיגלים (circuit ensemble): אוסף אינסופי של מעיגלים $\mathcal{C} = \{C_n\}_{n \in \mathbb{N}}$ המוגדר על קלטים באורך n .

- $C(u)$ היא תוצאה חישוב של המעלג C על הקלט u .
- $L(C)$ היא השפה שمعالג C מקבל. מעגל C מקבל שפה L אם מתקיים $L = C(L)$.
- נאמר ש- C מבreira שפה $*: \{0,1\}^n \rightarrow L$ אם לכל $\mathbf{A} \in \{0,1\}^n$ ולכל $x \in L$ אם מתקיים $x \in C_n(x) \Leftrightarrow x \in A$.
- עברור קלטים באורך 0 = a (המילה ריקה), נרצה גם שערים קבועים – ONE, ZERO.



סיבוכיות מעגלים

שאלות מרכזיות בקורס: בהינתן פונקציה או שפה, כמה "קשה" לחשב או להכريع אותה? מה הופך פונקציה או שפה ל"קשה" עבור מעגליםبولיאניים?

משפט (חץ-פורמלי): אם $\{0,1\}^n \rightarrow \{0,1\}$: f לא ניתן לחישוב ע"י משפחת מעגלים $\{C_n\} = \mathcal{C}$ כך שביל מעגל בגודל 2^{n^0} (קטן מאוד זמן אקספוננציאלי), אז f לא ניתן לחישוב ע"י תכנית מחשב שרצה בזמן $(n^0)^2$.

גודל מעגל: גודל מעגל \mathcal{C} הוא מספר השערים בו. נסמן זאת $|\mathcal{C}|$. אינטואיטיבית, גודל המעגל מייצג את מספר הצעדים שהאלגוריתם מבצע.

טענה (חסם עליון): כל פונקציה $\{0,1\}^n \rightarrow \{0,1\}$: f ניתן לחישוב ע"י מעגל בגודל $2^{n^0} = 2^n \cdot n$.

הערה: זה לא נכון עבור פיתוחו למשל, יש פונקציות שלא ניתן לחשב בזמן אקספוננציאלי.

הוכחה: ראינו כי ניתן לייצג פונקציה כזו על ידי פירוק למעגלים נפרדים שמהווים כל אחד שורה שמניבת 1, וביצוע OR בין כל שורות האפשרויות היא 2^n בטלת האמת. כל מעגל שמהווה שורה מסוימת מורכב מכל היותר n שעריו ו-OR בין כל ביטי הקלט (או גם שערי $-$), כלומר הוא מגודל $(n) \cdot 0$. על ידי שרשור כל המעגלים באמצעות OR קיבל מעגל מגודל לכל היותר $2^n \cdot n$.

משפט שאנון (חסם תחתון): עבור n גדול מספיק, קיימות פונקציות שלא ניתן לחישוב ע"י מעגלים בגודל $\frac{2^n}{10n} < s$.

הוכחה (טייעון ספירה): נראה **שיש** הרבה יותר פונקציות מאשר מעגלים בגודל הנ"ל $\frac{2^n}{10n}$. זה אומר שיש פונקציות שאף מעגל בגודל זהה לא מחשב. מספר הפונקציות $\{0,1\}^n \rightarrow \{0,1\}$ הוא 2^{n^2} (לכל קלט צריך לבחור אחת ממשתי תוצאות).

חסם עליון על מספר המעגלים בגודל s :

- יש לנו s שערים (לכל שער 3 אפשרויות), לכן יש לנו 3^s אפשרויות עבור בחירת סוג השערים.
- לכל שער נתון יש לכל היותר שני ילדים (או שער קודם, או בית קלט), לכן עבור בחירת הבניות אליו יש לכל היותר $(s+1)^2$ אפשרויות.
- עבור בחירת הבנייה לפולט יש לנו $(s+1)^2$ אפשרויות.

החסם הגס שאנו מכבלים הוא $2^{n^2} < 3^s (s+1)^2 (s+1)^2 = \frac{2^n}{10n}$. נטען כי עבור s הביטוי שקיבلونו קטן מ- 2^{n^2} . נראה שבעמיט כל הפונקציות דורשות מעגלים ענקיים, בהתאם תוכניות שרצות הרבה זמן. אולם, הפונקציות שמעניינות אותנו הן לרוב מאד מסוימות, והחסם הזה לא אומר עליהן שום דבר. האם יכול להיות שאות כל הפונקציות "המעניינות" אפשר לחשב בזמן לינארי? אנחנו לא יודעים.

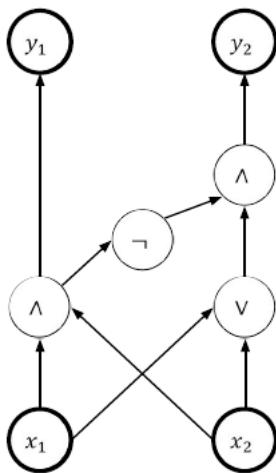
תרגול 1 (מעגליםبولיאניים)

פעולות על מילים – עברו מילה $w_1 \dots w_n$ ל- w :

- היפוך: $w^R = w_n \dots w_1$
- שרשור: $ww' = w_n \dots w_1 w'_1 \dots w'_n$
- חזקה: $w^0 = \varepsilon$ כאשר $\varepsilon = w \dots w$

פעולות על שפות:

- שורש: $w' \in L_2 \mid w \in L_1 \wedge w' \in L_1$. שרשור של כל שתי מילים כאשר הראשונה מהשפה הראשונה והשנייה מהשפה השנייה.
- מהשפה הראשונה והשנייה מהשפה השנייה:
- חזקקה: $\{w_i | i \in [k]\} \in L$, $w_1 w_2 \dots w_k \in L^k$. רצף של k מילים כאשר כל מילה היא בשפה, כלומר שרשור שפה L עם עצמאו k פעמים. נגידו $\{\varepsilon\}^L$.
- סגור קלייבן: $\{w_i | i \in L\} \in \mathbb{N}$, $w_1 w_2 \dots w_k \in L$, $w_k | k \in \mathbb{N}$. זו הכללה של * Sigma, כל המילים שניתן להרכיב משרשור بما מילימ שנרצה משפה L (נוכל לשגרר 10 מילים, או 3 מילים, כל מילה חדשה כזו היא $-L$).
- לכל L מתקיים ${}^*L \in \varepsilon$. למשל $\{\varepsilon\} = \emptyset$.

מעגלים בוליאניים:

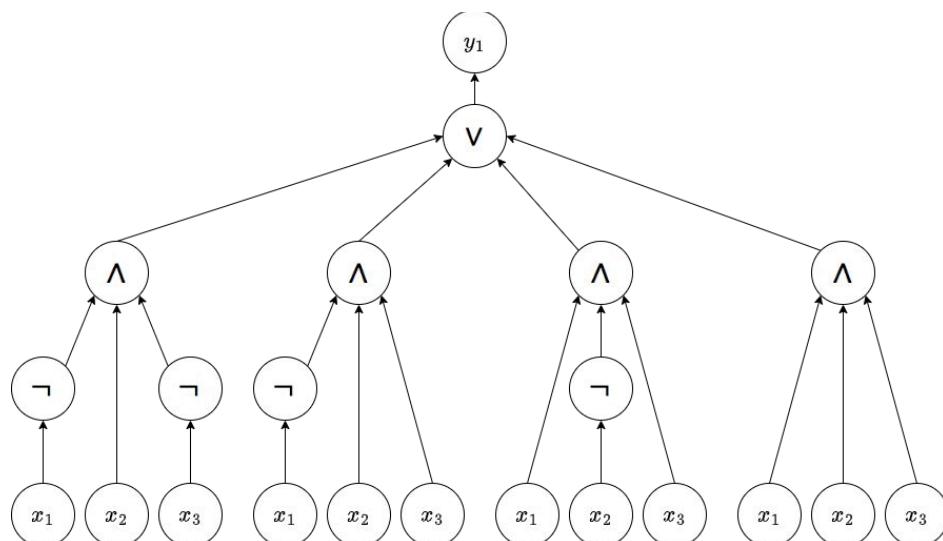
נסתכל לדוגמה על המעגל הבא הכלול 2 ביטים של קלט 2-ביטים של פלט. יש כאן 4 שערים אוז גודל המעגל הוא 4. המעגל מוחזר את סכום הקלטים בייצוג בינארי. נסתכל על כל בית בנפרד:

- הבית השמאלי – יהיה 1 אם "מ שני ביטי הקלט הם 1. לכן נבצע AND ביניהם.
- הבית הימני – יהיה 1 אם "מ רק אחד מהביטים הוא 1, לכן נבצע XOR ביניהם. אין לנו שער XOR ולכן נבנה אותו באמצעות ($NOT(AND(x_1, x_2)), OR(x_1, x_2)$) בולם שלפחות אחד מהם מתקיים אבל לא שניהם ביחד.

תרגיל 1: בנו מעגל שמקבל 3 ביטים, ומחזיר 1 אם "מ המספר הבינארי הוא ראשוני.

פתרון: נבדוק האם המספר הוא 2, 3, 5, 7 (המספרים הראשוניים בין 0 ל-7), כלומר המספרים הם 0,010, 0,111, 1,011, 1,101, 1,010. לבנייה לפי תתי-מעגלים עבור כל מספר. נשים לב כי יש לנו שניים עם 3 ו-4 קלטים (בשננתה את גודל המעגל צריכים להיות זהווים, כי יש כאן הרכבה של שערים).

לבניה מהסוג הזה קוראים DNF (כasher לוקחים OR על הרובה AND), בהמשך הקורס נעסק הרבה בצורה CNF (כasher ניקח AND על הרובה OR). אפשר גם להגיד כי השפה של המעגל היא $\{L = \{111, 101, 011, 010\}\}$.

**טענה 1:** לכל $\{0,1\}^n \subseteq L$ קיים מעגל המקבל אותה בגודל $(2^n \cdot n) \cdot O$.

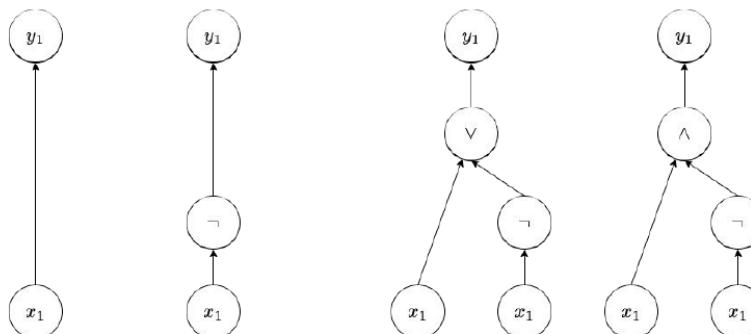
הוכחה (ראינו גם בשיעור): צריך להבין מה הגודל של המעגל שבנינו בצורה DNF בדוגמה כפונקציה של זה:

- כמה תת-מעגלים יש לנו? לכל היוטר 2^n , ככמות המיללים האפשרות בשפה.
- מה הגודל של כל תת-מעגל? בערך 2^n , לכל היוטר n פעמיים של NOT, ואפשר לשרשר באמצעות AND את כלם.
- כמה עולה לנו לחבר אותם ביחד? לכל היוטר 2^n .

טענה 2: לכל $\{0,1\}^n \subseteq L$ קיים מעגל המקבל אותה בגודל $(2^n \cdot n) \cdot O$.

הוכחה: נוכיח שלכל $\{0,1\}^n \subseteq L$ קיים מעגל C בגודל לפחות $10 \cdot 2^n - 10$ המקבל אותה, באינדוקציה על זה:

- בסיס: עבור $1 = n$ נראה שלכל $\{0,1\}^1 \subseteq L$ קיים מעגל בגודל לפחות $10 = 10 - 2^1 \cdot 10$ המקבל אותה. יש כאן 4 שפות אפשריות.

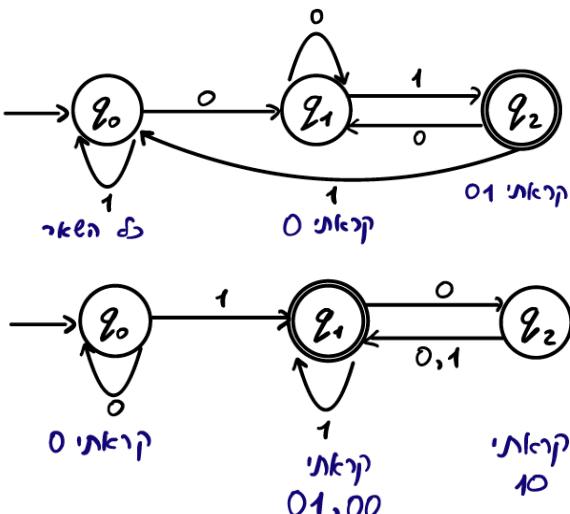


- צעד: נניח שקיימים מעגלים עבור שפות באורךים $1 - n$ ונשתמש בהם כדי לבנות מעגל מאורך n .
- עבור $\{0,1\}^n \subseteq L$ נגדיר את השפטות הבאות:
 - $\{w \in \{0,1\}^{n-1} : w \in L\} \subseteq L$ כל המילאים שאם נוסיף להן 0 הן יהיו ב- L .
 - $\{L_1 \in \{0,1\}^{n-1} : w \in L_1 \subseteq L\}$ כל המילאים שאם נוסיף להן 1 הן יהיו ב- L .
 - לפי ה"א קיימים מעגלים C_0, C_1 , בגודל לפחות 10 – $2^{n-1} \cdot 10$ המקבלים את L_1 בהתאם.
 - בעת נוכל לומר כי מתקיים $(L_1 ||| \{0\}) \subseteq L$.
 - אם $0 = x_n$ אז התוצאה שמענינו היא של C_0 , באופן דומה עבור C_1 ולכן בוצע OR על שתי האפשרויות.
 - כלומר המעגל שמחשב את הביטוי** $(x_{n-1}, \dots, x_1 \wedge C_1(x_1, \dots, x_{n-1}) \vee C_0(x_1, \dots, x_{n-1}))$ מקבל את L .
 - המעגל הוא מוגדל $5 - 2^n \cdot 10 < 4 + 2^{n-1} \cdot (10 \cdot 2)$ וכך הוכחנו.

אוטומטים סופיים

אוטומט סופי דטרמיניסטי (אס"ד - DFA)

דוגמאות ורקע:



דיברנו על המשימה של הכרעת שפה – האם קלט נתון יש תכמה בלשוי או לא. נסתכל על $\{0,1\}^*$: $w = 011$: שפת כל המילאים הבינאריות שמסתיימות ב-01. במודול הבא שלנו (DFA), נבעור על הקלט בצורה streaming ולא נחזור אחרת. נתאר את מה שהמודול "זוכר" בעזרת המצבים השונים. נסמן מצב תחيلي לריצה, ומצב מקבל שם נסימן בו קיבל את המילה.

למשל על הקלט 101 הריצה תהיה: $q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_0$ ולבן המילה בשפה. עבור 010: $q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_0$ לא סיימנו במצב מקבל ולבן המילה לא בשפה. ניתן לפרש את האוטומט לפי מה שקרה לנו ולאן זה מוביל אותנו (סמנטיקה של המצבים, מה כל מצב צריך לבצע).

דוגמה נוספת – האוטומט הבא מקבל את 1101: $q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_1$. לעומת זאת הוא לא מקבל את 0010.

אס"ד (DFA): אוטומט סופי דטרמיניסטי הוא חמישייה $(Q, \Sigma, \delta, q_0, F)$:

- Q – קבוצה סופית של מצבים.
- Σ – אלפבית.
- $\Sigma \times Q \times Q$: δ – פונקציית מעברים.
- $q_0 \in Q$ – מצב התחלתי.
- $F \subseteq Q$ – קבוצה סופית של מצבים מקבלים.

למשל עבור הדוגמה השנייה: $\{q_1\} = q_1, \Sigma = \{0,1\}, q_0, F = \{q_0, q_1, q_2\}$, $\delta(q_0, 1) = q_0, \delta(q_0, 0) = q_1, \delta(q_1, 1) = q_1, \delta(q_1, 0) = q_2, \delta(q_2, 0) = q_0$. פונקציית המעברים הבסיסית קוראת אחת ושולחת אותנו למצב חדש. נרצה להגיד **פונקציית מעברים מוחבנת יותר שתוכל גם לקבל מילה** (מפעילים את פונקציית המעברים הבסיסית על כל אות).

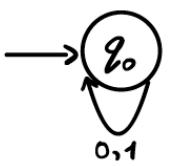
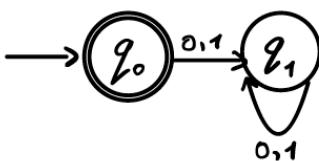
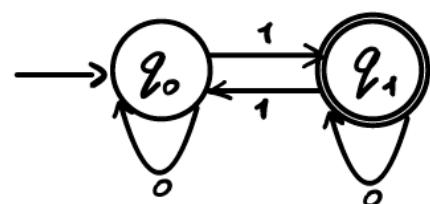
פונקציית מעברים מוחבנת: בהינתן אס"ד $(Q, \Sigma, \delta, q_0, F)$ = A , פונקציית המעברים המוחבנת $\hat{\delta}$: $\hat{\delta} : Q \times \Sigma^* \rightarrow F$ מוגדרת באינדוקציה באופן הבא:

- עבור המילה הריקה: $\hat{\delta}(q, \epsilon) = q$.
- עבור מילה $x \in \Sigma^*$: $\hat{\delta}(q, x) = \hat{\delta}(\hat{\delta}(q, x_1 \dots x_n), x_{n-1} \dots x_1)$. כלומר, אנחנו כל פעם מבצעים צעד אחד לפי פונקציית המעברים המקורי שלנו, δ , על כל אות בנפרד, החל מ- x_1 ועד x_n .

שפה של אוטומט:

- קובלה:** אס"ד A מקבל מילה $x \in \Sigma^*$ אם $x \in F$ או $x \in (\hat{\delta}(q_0, \hat{\delta}(q_0, \dots, \hat{\delta}(q_0, x_1))))$, כלומר הפעלת פונקציית המעברים המוחבנת על המילה מביאה למצב מקבל. באופן שקול, A מקבל את המילה \Leftrightarrow קיימת סדרה של מצבים $q_n, q_{n-1}, \dots, q_1 \in Q$ כך שלכל $[n] \in i$ מתקיים $q_i = \hat{\delta}(q_{i-1}, x_i)$ וכן המצב האחרון מקבל, $F \in q_n$.
- שפה האוטומט:** אס"ד A מכיר/**מקבל שפה** L אם L היא אוסף המילאים שהוא מקבל. נסמן זאת (A, L) .

דוגמאות לאוטומטיים:

 $: A_1$  $: A_2$  $: A_3$ 

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

1. לא מקבלים אף מילה, $\emptyset = L(A_1)$.2. מקבלים רק את המילה הריקה, $\{\epsilon\} = L(A_2)$.3. מקבלים מילים שמכילות מספר אי-זוגי של אחדות, $= L(A_3)$.

$1 \equiv x \in \{0,1\}^*$. ניתן גם לרשום את התנאי בצורה הבאה:
 $1 = x \oplus \#_1(x) \mod 2$ (הזוגיות של כמהות האחדות בקלט א, 0 עבור זוגי, 1 עבור אי-זוגי), כלומר ביצוע XOR על כל ספרות המילה אחריו השניה ($x_n \oplus \dots \oplus x_1 \oplus x$) יניב את התוצאה 1. כך נאמר שיש מספר אי-זוגי של אחדות.

4. האוטומט הינו מורכב הבא: $\{0\} \subseteq \{0,1\}^*, k \geq 0 : y = y10^{2k} = \{y10^{2k}\} = L(A_4)$. מילה בינארית כלשהי שמסיימת ב-1 ולאחר מכן מספר זוגי של אפסים (0, 2, 4 וכו').

נוכחות פורמלית עבור אס"ד 3:

- אבחנה: $q_{b \oplus b'} = q_b = \{b', b\} \in \{0,1\}$. כלומר לכל מצב

וקלט, המצב שאליו עברו הוא XOR שלם. אם אנחנו קוראים 0

אנחנו נשארים באותו המצב, ואם קוראים 1 אז הופכים את המצב.

נוכחים אינדוקציה על אורך הקלט $|w|$ כי מתקיים: $q = (\hat{\delta}(q_0, \hat{\delta}(x, \hat{\delta}(q_0, y, b))))$

כלומר התוצאה תלויה בהזוגיות של x. אם המספר האחדות ב-x הוא זוגי

(תוצאה ה-XOR היא 0) נסימן $b = q_0$ ואם הוא אי-זוגי (תוצאה ה-XOR

היא 1) נסימן $b = q_1$.

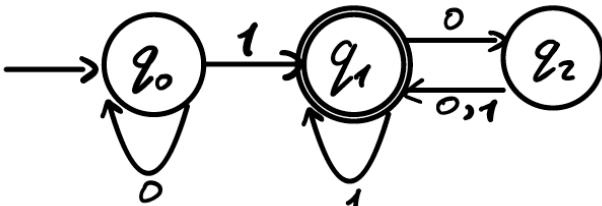
בבסיס: $q_0 = q_0 = (\epsilon, \hat{\delta}(q_0, \hat{\delta}))$, ביוון שבקלט הריק יש מספר זוגי של אחדות, לכן $\epsilon = \hat{\delta}(q_0, \hat{\delta})$.

צעד: נניח את נכונות הטענה עבור קלטים באורך n, ונוכחים עבור $n+1$. יהיו $y, yb, yb' \in \{0,1\}^n$.

המעבר האחרון נקבע מכך $q_{\oplus yb} = \hat{\delta}(q_{\oplus y}, b) = \hat{\delta}(\hat{\delta}(q_0, y), b) = \hat{\delta}(q_0, \hat{\delta}(y, b))$ אבחנה הנחת האינדוקציה

שהזוגיות של המחרוזות y זה הזוגיות של מספר האחדות ב-n הביטים הראשונים ואז הביטים האחרונים.

נוכחות פורמלית עבור אס"ד 4:



- נוכחים אינדוקציה על אורך הקלט x כי מתקיים:

$y10^{2k} = x \Leftrightarrow x = (\hat{\delta}(q_0, y), \hat{\delta})$. הבעייה בהוכחת הטענה

באינדוקציה היא שאין לנו מידע על חלק גדול מהמקרים. נניח

יש לנו קלט שhabi'a אותו ל- q_2 , אין לנו מידע על המבנה שלו.

נצטרך טענת אינדוקציה חזקה יותר.

- נחלק למקרים ונפרוש את כל האפשרויות של מחרוזות בינאריות:

או שאין בהן אחדות, או שיש בהן (נלק לאחד האחרו), וונתפצל לשני מקרים – מספר זוגי של אפסים, ומספר אי-זוגי).

מכל אלן נובע מקרה 2 בצורת האמ"מ. ככלומר אם מילה היא לא במקרה 2, היא במקרה 1 או 3 ולבן עוברת למקרים אחרים

שहם אינם מקבלים. אם מילה היא במקרה 2 היא עוברת למצב מקבל.

בבסיס: $\epsilon = x$ ובפרט $0 = x$. אכן לפי הגדרה מתקיים $q_0 = (\epsilon, \hat{\delta}(q_0, \hat{\delta}))$.

צעד: נניח נכונות עבור אורך n. יהי $y1^{n+1} \in \{0,1\}^{n+1}$. נחלק למקרים:

מקרה	טענה	הוכחה
1 – רק אפסים	$x \in \{0^k\}$ אם $q_0 = (\hat{\delta}(q_0, x), \hat{\delta})$.	בפרט $0 = x$ עבור $1 \leq k$. מתקיים: $\hat{\delta}(q_0, 0) = q_0 = \hat{\delta}(q_0, 0^{k-1}), 0 = \hat{\delta}(q_0, 0) = q_0$
2 – יש אחד ועוד אפסים	$x \in \{y10^{2k}\}$ אם $q_1 = (\hat{\delta}(q_0, x), \hat{\delta})$.	מקרה א: $y1 = x \Rightarrow x = y1$. אנחנו ידועים שהמילה מסתיימת ב-1. לא משנה באיזה מצב אנחנו, 1 מעביר אותנו תמיד ל- q_1 . מקרה ב: $y10^{2k'+1} = x \Rightarrow k' \geq 1 \Rightarrow y10^{2k'+1} \geq 1$. יש לנו לפחות שני אפסים בסופו. אם נוריד את האחרון נישאר עם מספר אי-זוגי של אפסים. מתקיים: $\hat{\delta}(q_0, x) = \hat{\delta}(\hat{\delta}(q_0, y10^{2k'+1}), 0) = \hat{\delta}(q_2, 0) = q_1$
3 – יש אחד ועוד אפסים	$x \in \{y10^{2k+1}\}$ אם $q_2 = (\hat{\delta}(q_0, x), \hat{\delta})$.	בפרט $0 = x$ ומתקיים: $\hat{\delta}(q_0, x) = \hat{\delta}(\hat{\delta}(q_0, y10^{2k}), 0) = \hat{\delta}(q_1, 0) = q_2$



สภาพ רגולריות

שפה רגולרית: שפה נקראת רגולרית אם קיים אס"ד שמקבל אותה. בולם, שפה שניית להכريع בעזרת אס"ד.

פעולות עלสภาพ רגולריות:

אפשרות ראשונה להוכיח שפה L היא רגולרית היא להציג אס"ד M שמקבל אותה, $(M) = L$. אפשרות שנייה היא בעזרת תכונות סגירות שלสภาพ רגולריות. יהיו $\Sigma \subseteq \{w : w \in L_1 \wedge w \in L_2\}$. נגיד את הפעולות הבאות:

- איחוד: $L_1 \cup L_2 = \{w : w \in L_1 \vee w \in L_2\}$
- חיתוך: $L_1 \cap L_2 = \{w : w \in L_1 \wedge w \in L_2\}$
- משלימים: $\bar{L} = \{w : w \notin L\}$
- שרשו: $L_1 || L_2 = \{xy : x \in L_1, y \in L_2\}$
- חזקה: $L^i = \{w : w \in L^{i-1}\}$, $L^0 = \{\epsilon\}$
- סגור קליבי: $L^* = \bigcup_{i \geq 0} L^i$. שרשורים באורך כלשהו של מילים מ- L .

סגורות שלสภาพ רגולריות: השפות הרגולריות סגורות לכל הפעולות הנ"ל.

דוגמאות לשפות רגולריות ואי-ז' פעליה טובה כדי להוכיח זאת:

- השפה $\{0^n 1^m : n, m \geq 0\} = L_1$ – רגולרית, בעזרת שרשו. אפשר לבנות אס"דים לשתי השפות בנפרד.
- כל שפה סופית היא רגולרית, בעזרת איחוד. קל לבנות אס"ד שמקבל מילה מסוימת בשפה.
- השפה $\{0 \#_1 (x) \#_2 0 : x \in \Sigma\}$ היא רגולרית, בעזרת משלימים לשפה שראינו קודם (אס"ד 4).

סגורות תחת איחוד: אם $\Sigma \subseteq L_1, L_2$ רגולריות אז גם $L_1 \cup L_2$ רגולרית.

ברגע שהרכינו את האס"ד הראשון נגמר הקטל, לא נוכל להריץ את האס"ד השני ולבדק האם המילה בשפה השנייה. הרעיון הוא שנרצה להריץ אותם במקביל, נüber על הקטל פעמי אחת, ובכל פעם שאנו קוראים אותן, נרצה להריץ את שני האס"דים (נשמר מצב בפועל, לכל אחד מהאס"דים). אם נסיים במצב מסוים במהלך מבחן – נקלבל, אחרת נדחה.

רעיון: יהיו $A_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$, $A_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ אס"דים עבור L_1, L_2 בהתאם. נגיד אס"ד חדש שיבריע את האיחוד של השפות הללו: $(Q_U, \Sigma, \delta_U, q_U, F_U) = A_U$ באשרו:

- $Q_U = Q_1 \times Q_2$
- לכל $(q_1, q_2) \in Q_1 \times Q_2$ ולבכל $\sigma \in \Sigma$: $\delta_U((q_1, q_2), \sigma) = (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma))$
- $q_U = (q_1, q_2)$
- $F_U = \{(q_1, q_2) : q_1 \in F_1 \vee q_2 \in F_2\}$

הוכחה: על מנת להוכיח נכונות נראה באינדוקציה (טענה כללית יותר):

- בסיס: $\epsilon = x$. מתקיים: $\delta_U((q_1, q_2), \epsilon) = (q_1, q_2) = (\delta_1(q_1, \epsilon), \delta_2(q_2, \epsilon))$
- צעד: נניח נכונות עבור קלט באורך n ונווכיח עבור $\sigma = x$. מתקיים:

$$\begin{aligned} \hat{\delta}_U((q_1, q_2), w\sigma) &= \delta_U(\hat{\delta}_U((q_1, q_2), w), \sigma) = \delta_U\left(\left(\hat{\delta}_1(q_1, w), \hat{\delta}_2(q_2, w)\right), \sigma\right) = \\ &= \left(\delta_1(\hat{\delta}_1(q_1, w), \sigma), \delta_2(\hat{\delta}_2(q_2, w), \sigma)\right) = \left(\hat{\delta}_1(q_1, w\sigma), \hat{\delta}_2(q_2, w\sigma)\right) \end{aligned}$$

סגורות תחת שרשו: אם $\Sigma \subseteq L_1, L_2$ רגולריות אז גם $L_1 || L_2$ רגולרית.

היינו רוצים לקפוץ מהרצאה של אס"ד 1 על א' להרצאה של אס"ד 2 על ב'. אנחנו לא יודעים מתי צריך לבצע את הקפיצה זו, מתי סימנו לקרוא את החלק ששיך ל- L_1 ונרצה להריץ את A_2 . זה מוביל אותנו ל-NFA.



תרגיל 2 (אס"ד)

תרגיל 1: בנו אוטומט שמקבל את שפת כל המילים המסתויימות ב-00, כלומר $\{w00 \mid w \in \{0,1\}^*\} = L(M)$.

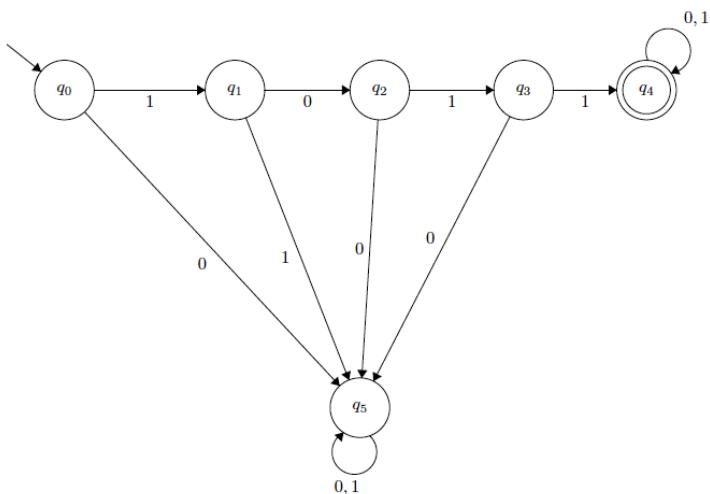
משמעותו הוא אם שתי האותיות האחרונות היו 00 (ואז נctrיך מספר מקובל), ומשמעותו הוא אם היה לנו 0 לפני או לא.

$$\begin{aligned} M &= (Q, \Sigma, \delta, q_0, F) \\ &= (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\}) \end{aligned}$$

כך ש- δ מוגדרת כך:

$$\begin{aligned} \delta(q_0, 0) &= q_1 & \delta(q_0, 1) &= q_0 \\ \delta(q_1, 0) &= q_2 & \delta(q_1, 1) &= q_0 \\ \delta(q_2, 0) &= q_2 & \delta(q_2, 1) &= q_0 \end{aligned}$$

- נגידור את המצב q_2 אחרי שריאנו 2 אפסים. אם נקבל עוד אפס נישאר במצב.
- נגידור את המצב q_0 כאשר 0 אפסים. בעבר אליו אחריו שריאנו 1 מ- q_2 .
- נגידור עוד מצב בין q_1 כאשר ראיינו בבדיקה פעם 1 אפס.



תרגיל 2: בנו אוטומט עבור שפת כל המילים שמתחלות ב-1011, כלומר $\{1011w \mid w \in \{0,1\}^*\} = L$.

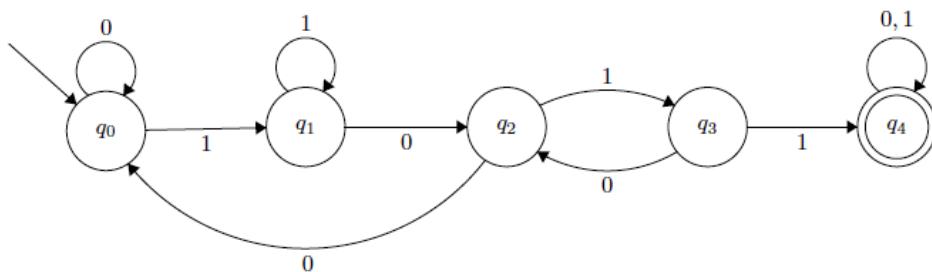
הדרישה לגבי מה שקרה בסוף מחזורות יותר מורכבות. באן אנחנו דורשים על תחילת המחזורות. נוכל לבנות מצבים שעוברים ביניהם עם כל אות קלט שימושית אותן-1,0-1-0-1, וכל דבר אחר מעביר אותנו למצב "בישולון". מהמצב של הבישולון לא נוכל לחזור חזרה, כי בבר לא נרצה לקבל את המילה.

אחרי שמדוברים אס"ד, תמיד לחשב על מקרים קצה מוזרים כמו המילה הריקה, ולראות שהאס"ד מתנהג כמו שאנו מכפים.

תרגיל 3: הוכיחו כי שפת כל המילים שמכילות את 1011 היא רגולרית, כלומר $\{1011w_1 w_2 \mid w_1, w_2 \in \{0,1\}^*\} = L$.

בצורה ישירה – בניית אס"ד שמקבל אותה. מעניינות אותנו מילים שפותנציאלית יכולות להפוך ל-1011, נעקוב אחרי התחליות השונות של 1011. נסתכל על האס"ד הנוכחי ונחשב עלי' בצורה זו:

- q_3 – עבשו היה 101 (אבל עוד לא היה 1011 בשום מקום). אם נקבל 0 לא נרצה ללבת ל"בור" שלנו, יש פוטנציאל שהוא יתחל מה-1 שיברנו. כי **צברנו ש-1011**
- q_2 – עבשו היה 10 (אבל עוד לא היה 1011...). אם נקבל 0 אז צברנו 100 או 00 וזה לא עוזר לנו. נחזור להתחלה.
- q_1 – עבשו היה 1, כל עוד יש 1 נישאר באותו מצב, עד שנראה 0 ראשון.



אם נרצה למצאו את שפת כל המילים **שלא** מכילות 1011, כלומר \bar{L} ? כל מצב שלא הגיענו בו- q_4 הוא טוב לנו. נהפוך את המצביעים המקבלים ונקבל אס"ד שמקבל את השפה המשילמה.

משפט 1: שפות רגולריות סגורות תחת משלימים.

הוכחה: תהא L שפה רגולרית. נוכיח כי \bar{L} רגולרית. לכן קיימים אס"דים M ו- M' כך ש- $L = L(M)$ ו- $\bar{L} = L(M')$. נגידור אוטומט $(Q \setminus F, \Sigma, \delta, q_0, Q \setminus F) = M'$. נוכיח כי M' מקבל את \bar{L} :

$$w \in \bar{L} \Leftrightarrow w \notin L \Leftrightarrow \hat{\delta}(q_0, w) \notin F \Leftrightarrow \hat{\delta}(q_0, w) \in Q \setminus F \Leftrightarrow w \in L(M')$$

משפט 2: שפות רגולריות סגורות תחת חיתוך.

הוכחה: יהיו L_1, L_2 שפות רגולריות. אז קיימים אס"דים $M_1 = (Q^1, \Sigma, \delta^1, q_0^1, F^1)$ ו- $M_2 = (Q^2, \Sigma, \delta^2, q_0^2, F^2)$ באופן דומה, אשר מקבלים את L_1, L_2 בהתאמה. נרצה לבנות אס"ד $M = L_1 \cap L_2 = (Q, \Sigma, \delta, q_0, F)$ (נעקוב אחר החישובים במקביל):



- $.Q = Q^1 \times Q^2$
- $\delta((q_1, q_2), \sigma) = (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma)) : \sigma \in Q^1 \times Q^2$ ולבלי $(q_1, q_2) \in Q^1 \times Q^2$
- $.q_0 = (q_0^1, q_0^2)$
- $.F = F^1 \times F^2 = \{(q_1, q_2) : q_1 \in F^1 \wedge q_2 \in F^2\}$

הראנו בשיעור כשהוכחנו סגירות עבור איחוד כי מתקיים: $\hat{\delta}((q_1, q_2), x) = (\hat{\delta}_1(q_1, x), \hat{\delta}_2(q_2, x))$. ניעזר בכך גם כאן ונסגור את ההוכחה:

$$w \in L(M) \Leftrightarrow \hat{\delta}(q_0, w) \in F \Leftrightarrow \hat{\delta}\left((q_0^1, q_0^2), w\right) \in F \stackrel{\text{טענה 10}}{\Leftrightarrow} \left(\hat{\delta}^1(q_0^1, w), \hat{\delta}^2(q_0^2, w)\right) \in F^1 \times F^2$$

$$\Leftrightarrow \hat{\delta}^1(q_0^1, w) \in F^1 \wedge \hat{\delta}^2(q_0^2, w) \in F^2 \Leftrightarrow w \in L(M_1) \cap w \in L(M_2) \Leftrightarrow w \in L(M_1) \cap L_1 \cap L_2$$

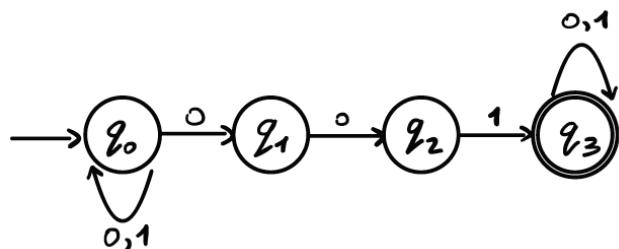
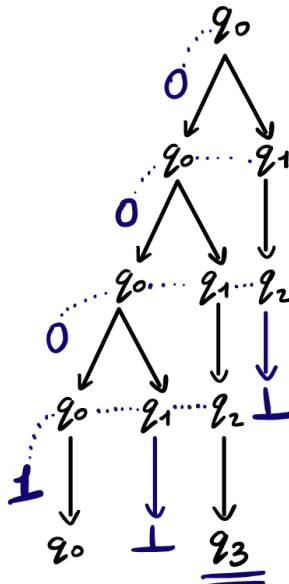
הוכחה חלופית: תוק שימוש בסגירות לאיחוד ומושלים, שימוש בהה-מורגן: $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$

אוטומט סופי א-דטרמיניסטי (אסל"ד - NFA)

בקע: באסל"ד שראינו עד כה, לכל מצב שאנו חווים ולבלי אותו מוגדר בדיקת מצב אחד בלבד שעוברים.

- באסל"ד יתכן עבור אותו ומצב שיש במאה מעברים אפשריים.
- בנוסף, ניתן לעצמנו גם לא להגדיר אף מעבר עבור אותו ומצב נתונים ("האוטומט נתקע").
- ניתן מעברים ספונטניים, שבהם אנחנו קופצים "בחינם" בלי לקרוא אות קלט (ε).

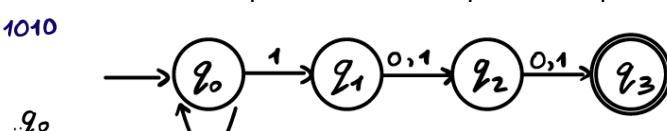
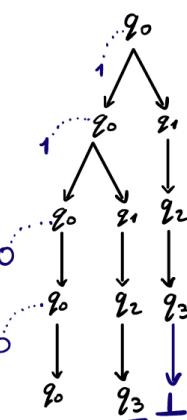
ריצה של אסל"ד:



נחשב עליה במשמעות ריצה במקביל על כל האפשרויות, בכל מצב אפשרי לעבור למספר מסוימלנטי. התהילה נמצאים בכולם סימולטנית. זהה יכול להתפשט בעז. מ מצבים מסוימים אין מעברים, וזה הענף נעצר שם.

לדוגמה עבור האסל"ד הבא, נרצה על הקלט 1001, ועבור כל מצב ואות נشرط את כל המעברים האפשריים הבאים. נאמר שהאסל"ד מקבל אם קיימים ענף כלשהו בעז שמסתויים במצב מקבל.

1100



דוגמא נוספת: בנו אסל"ד לשפה $\{0,1\}^*\{1,0,1\}^2$, כל המחרוזות הביניאריות שהתחוללו לפני הסוף הוא 1. היא רגולרית כי היא שרשרת של שפות רגולריות.

אסל"ד (NFA): אוטומט סופי א-דטרמיניסטי הוא חמישייה $(Q, \Sigma, \delta, S, F)$:

- Q – קבוצה סופית של מצבים.
- Σ – אלפבית.

$\Sigma \times Q \rightarrow P(Q)$: δ – פונקציית מעברים, שולחת אותנו לאוסף של מצבים (תת-קובוצה כלשהי של Q).

$S \subseteq Q$ – קבוצה סופית של מצבים התחלהיים.

$F \subseteq Q$ – קבוצה סופית של מצבים מקבלים.

עבור הדוגמה الأخيرة שלנו אפשר לכתוב במפורש את δ :

δ	0	1
q_0	$\{q_0\}$	$\{q_0, q_2\}$
q_1	$\{q_2\}$	$\{q_2\}$
q_2	$\{q_3\}$	$\{q_3\}$
q_3	\emptyset	\emptyset

ריצה של NFA: הגדירנו את פונקציית המעברים $P(Q) \rightarrow \Sigma \times Q$. נגידר את פונקציית המעברים המורחבת באופן אינדוקטיבי:
 $\hat{\delta}_N: P(Q) \rightarrow \Sigma^* \times P(Q)$

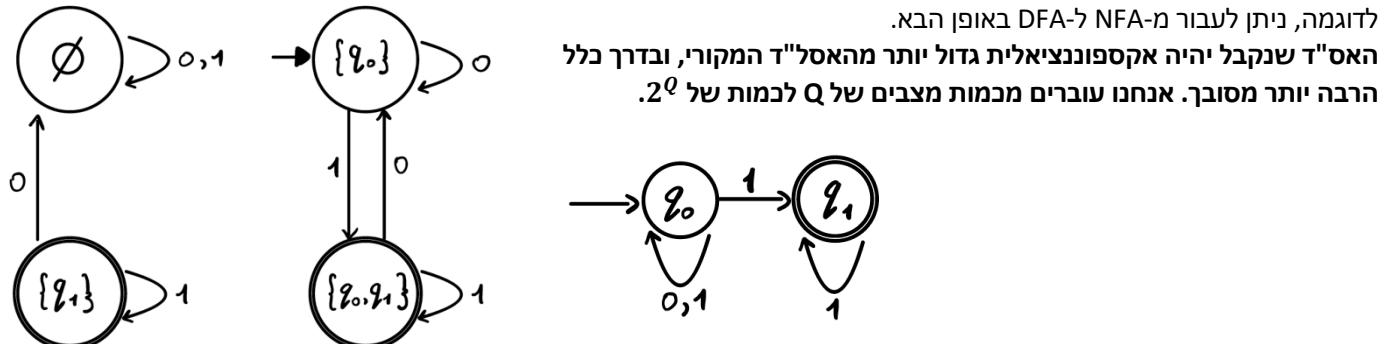
- בסיס: עבור הקלט הריק, לכל $Q \subseteq T$ מתקיים $T = \hat{\delta}(T, \epsilon) = \{q \in Q \mid \delta(q, \epsilon) = q\}$.
- צעד: עבור הקלט $\Sigma^n \in \Sigma$ ואות נוספת $x \in \Sigma$, לכל $Q \subseteq T$ מתקיים: $(\sigma, x) \hat{\delta}(T, x) = \bigcup_{q \in \hat{\delta}(T, x)} \delta(q, x) = \{q \in Q \mid \delta(q, x) = q\}$. כלומר, נסתכל על אוסף המצבים שנitin להגעה אליהם על ידי קריאת x , ומם נבצע איחוד של כל המצבים שנitin להגעה אליהם ע"י קריאת x .

קבלה ב-NFA: נאמר ש- N מקבל את המילה Σ^* אם $\emptyset \neq F \cap \{x \in \Sigma^* \mid \text{קיים } q \in Q \text{ כך ש } \hat{\delta}^*(q, x) = q\}$. באופן שקול: אם קיימים $q_0 \in S$ וمتsequים $(x_i, q_i) \hat{\delta}^* = q_0$ עבור $i = 1, \dots, n$, אז $q_0 \in F$.

טענה: הכוח החישובי של DFA ושל NFA הוא זהה. לכל שפה L : קיים DFA שמקבל את $L \Leftrightarrow$ קיים NFA שמקבל את L .

הוכחה:

- $\text{DFA} \Leftarrow \text{NFA}$: כי אס"ד הוא "מקרה פרטי" של אס"ל.
- $\text{NFA} \Leftarrow \text{DFA}$: נראה כי ניתן לבנות DFA שקול לאס"ל. המצבים של DFA יהיו תת-קבוצות הקבוצות האפשריות של מצבים. נאמר שיש קשת מחת-קבוצה 1 לתת-קבוצה 2 אם ניתן לעבור מ מצב כלשהו בתחום התת-קבוצה 1 למצב אחר בתחום התת-קבוצה 2.



אוטומט חזק: זה אס"ל $M = (Q_M, \Sigma, \delta_M, s_M, F_M) = (Q, \Sigma, \delta_N, S, F)$. נגידר אס"ד M כאשר:

- קבוצת מצבים המבוססת על תת-קבוצות של מצבים מ- Q : $Q_M = \{[R] \mid R \subseteq Q\}$.
- מצב תחيلي המוגדר להיות תת-הקבוצה של המצבים המתקבלים: $s_M = [S]$.
- קבוצת מצבים מקבלים: אוסף תת-קבוצות שמכילות מצב מקבל מ- F : $F_M = \{[R] \mid R \cap F \neq \emptyset\}$.
- δ_M – פונקציית מעברים שלוקחת לכל מצב $[R]$ את אוסף כל המצבים האפשריים לפי N ומאחדת ביניהם: $\delta_M([R], \sigma) = [\bigcup_{q \in R} \delta_N(q, \sigma)]$.

נשים לב כי הסימון $[R]$ לוקח קבוצה של מעברים $\{q_1, q_2, \dots, q_n\}$ מה-NFA ומתייחס אליהם במצב אוטומי יחיד ב-DFA. נוכיח כי $L(N) = L(M)$.

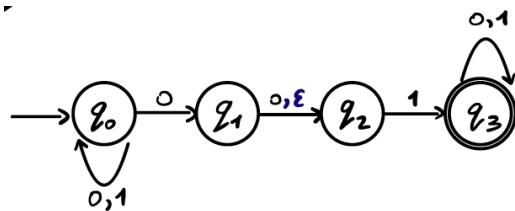
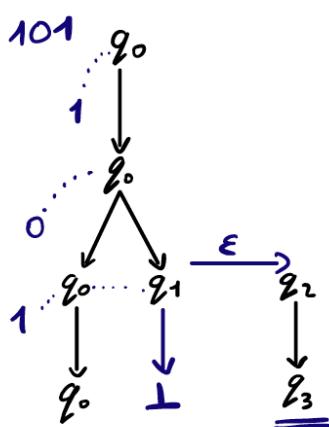
טענת עד: לכל מילה Σ^* w מתקיים: $[\hat{\delta}_M([S], w)] = [\hat{\delta}_N(S, w)]$. מהטענה נובע בפרט שלכל קלט Σ^* w :

$$w \in L(M) \Leftrightarrow \hat{\delta}_M([S], w) \in F_M \Leftrightarrow [\hat{\delta}_N(S, w)] \in F_M \Leftrightarrow \hat{\delta}_N(S, w) \cap F \neq \emptyset \Leftrightarrow w \in L(N)$$

הוכחה: באינדוקציה על אורך הקלט w :

- בסיס: עבור $\epsilon = w$ מתקיים $[\hat{\delta}_N(S, \epsilon)] = [\hat{\delta}_N(S)] = [S] = [\hat{\delta}_M([S], \epsilon)]$.
- צעד: עבור $\Sigma^n \in \Sigma$ וקלט נוסף $S \in \sigma$ מתקיים:

$$\hat{\delta}_M([S], x\sigma) = \delta_M(\hat{\delta}_M([S], x), \sigma) = \delta_M([\hat{\delta}_N(S, x)], \sigma) \stackrel{\text{by def}}{=} \delta_M \left[\bigcup_{q \in \hat{\delta}_N(S, x)} \delta_N(q, \sigma) \right] \stackrel{\text{by def}}{=} \hat{\delta}_N(S, x\sigma)$$

NFA עם מעברי ε

נרצה להרשות ממעריבי ϵ , בהן ניתן לעבור מ מצב אחד לאחר בילוי קרטיס. נסמן זאת במקבוק אחר ריצה של ה-NFA, כאשר אנחנו "עוברים הצדה" לעוד מצב חדש (או יותר, אם יש עוד מעבר ϵ) ללא עולות – לא מתקדים בזמן הזמן. זה פותח בפנים אופציה נוספת לסדרת מצבים.

נסמן $\{\epsilon\} \cup \Sigma = \Sigma_\epsilon$, וזה האסל"ד שלנו יוגדר $\delta: Q \times \Sigma_\epsilon \rightarrow P(Q)$ במשמעותו:

δ	0	1	ϵ
q_0	$\{q_0, q_1\}$	$\{q_0\}$	\emptyset
q_1	$\{q_2\}$	\emptyset	$\{q_2\}$
q_2	\emptyset	$\{q_3\}$	\emptyset
q_3	$\{q_3\}$	$\{q_3\}$	\emptyset

סבירה: נגדיר (q) להיות ה- ϵ -סיביה של מצב $Q \in q$, כל המצבים שאפשר להגיע אליהם ע"י מסע ϵ מ- q . עברו קבוצת מצבים $Q' \subseteq Q$ נגדיר את סביבת ה- ϵ להיות איחוד כל המצבים שנitin להגיע אליהם: $E(Q') = \bigcup_{q \in Q'} E(q)$.

ריצה של NFA עם מעבר ϵ : נגדיר $(Q, \Sigma^* \times P(Q) \rightarrow \hat{\delta})$ באופן אינדוקטיבי:

- בסיס: עברו הקלט הריך, לכל $Q \subseteq T$ מתקיים $(T, \epsilon) = E(T)$.
- צעד: עברו הקלט $x \in \Sigma$ ואות נוספת $\sigma \in \Sigma$, לכל $Q \subseteq T$ מתקיים: $(\sigma, \hat{\delta}(T, x)) = E\left(\bigcup_{q \in \hat{\delta}(T, x)} \delta(q, \sigma)\right)$. אנחנו תמיד מחושבים איך אפשר להסתכל "הצדה" בעז ולהרחיב את המסלול שלנו לעוד מצבים ע"י מעבר ϵ .

קבלה ב-NFA עם מעבר ϵ : נאמר ש- N מקבל את המילה $* \Sigma^* x$ אם $\emptyset \neq F \cap \hat{\delta}_N(S, x) \neq \emptyset$. בדיק בmeno קודם. באופן שקול (כآن בוצע במאשנויים): N מקבל את המילה $* \Sigma^* x \in \Sigma_\epsilon^k \bar{x}$ אם $\exists n \geq k$ כך ש- x מתקבל מ- \bar{x} על ידי $q_k = \delta(q_{i-1}, \bar{x})$, ומתקיים $q_i = \delta(q_{i-1}, x)$, ו- $q_0 \in S$, ו- $q_k \in F$.

טענה (סילוק מעבר ϵ): לכל NFA עם מעבר ϵ נקיים N לא מعتبر ϵ נקי שמתקיים $L(N) = L(N')$.

הוכחה: עברו $(Q, \Sigma, \delta, S, F) = (Q', \Sigma, \delta', S', F')$ באשר:

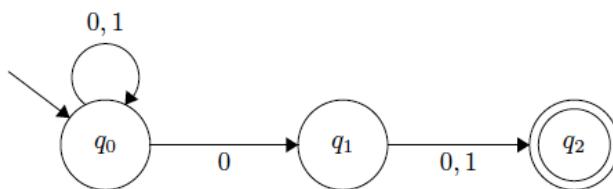
- $S' = E(S)$ – אנחנו מוכרים מראש את כל המצבים ההתחלתיים האופציונליים.
- $\delta' = E(\delta(q, \sigma))$ – כל צעד שנעשה יוביל אותנו לכל המצבים האפשריים כולל מעבר ϵ .

צריך להוכיח (באינדוקציה) שאנחנו משמרים את התנהגות של פונקציית המעברים המורחבת: $\delta'_N(Q', w) = \hat{\delta}'_N(Q', w)$.



תרגול 3 (اسل"ד וביטויים וגולריים)

תרגול 1: באס"ד אנחנו לא חייבים להגדיר מכל מצב מעבר עבור כל אות – implicitly זה אומר שאנו צריכים לקבועה הריקה. השפה של האס"ד זהה היא כל המילים באורך לפחות 2 מעל $\{0,1\}$ כאשר 0 היא הספרה הלווייתית האחרונה.



$$\begin{aligned} A &= (\{q_0, q_1, q_2\}, \{0,1\}, \delta, \{q_0\}) \\ \delta(q_0, 0) &= \{q_0, q_1\} \\ \delta(q_0, 1) &= \{q_0\} \\ \delta(q_1, 0) &= \delta(q_1, 1) = \{q_2\} \\ \delta(q_2, 0) &= \delta(q_2, 1) = \emptyset \end{aligned}$$

הוכחנו בשיעור כי אס"ד שקול לאס"ד, ונitin לעבור לאס"ד באמצעות מעבר לאוטומט החזקה. ביוון שיש לנו 3 מצבים באס"ד, יהיו לנו 8 מצבים בס"ד. המקבילים יהיו הקבוצות שמקילות מצבים מקבלים מהאסל"ד. נעבור על כל המעברים האפשריים:

- q_0 - אם קוראים 1 במצב q_0 בלבד זה מקרה קל.
- לעומת זאת, אם קוראים 0 אפשר לעבור או $-q_0$ או $-q_1$ וכן
- עבור מצב באס"ד של $\{q_0, q_1\}$.
- עבשו אם אנחנו קוראים 0 ואנו נמצאים או $-q_0$ או $-q_1$,
- נוכל להגיע גם $-q_2$ וכן עבור $\{-q_2\}$.

אין צורך להגדיר עברו שאר המצבים, כיון שלא ניתן הגיעו אליהם מ- q_0 בכלל. ביטוי רגולרי עבור השפה: $(1 \cup 0^*)^*$ (1 ס 0).

תרגול 2: עבור שפה L נגידר את השפה ההופכית: $L^R = \{w^R \mid w \in L\}$. הוכיחו כי התשיפות הרגולריות סגורות תחת rev. רגולריות ולבן קיימים אס"ד המקבל אותה: $L(A) = L(Q, \Sigma, \delta, q_0, F) = L'$. האינטואיציה שלם היא להפוך את כל החיצים כדי ליצור אוטומט חדש המקבל את המילים ההופכיות. אם היה מצב שבננסו אליו שתי קשותות, בעת יצאו ממנו שתי קשותות ולא נוכל להגדיר אס"ד, אלא אסל"ד. נגידר את $(Q, \Sigma, \delta', S, F') = A'$ באשר:

- $S = F$: המצבים ההתחלתיים הם המקבילים היישנים.
- $\{q_0\} = F'$: המצב המתקבל מה המצב ההתחומי הישן בלבד.
- $\{q\} = Q \mid \delta(r, q) = r \in \{(\sigma, q) \mid \delta'(r, \sigma) = r\}$: פונקציית המעברים תיקח אותנו לאוסף כל המצבים המקוריים r שהביאו אותנו ל- q לפי פונקציית המעברים החדשה – אנחנו הופכים את כל המעברים.

נוכיח כי $L(A') = rev(L)$ רגולריות כיון ש- A' מקבל אותה: $L(A') = rev(L)$

$$\begin{aligned} w = \sigma_n \dots \sigma_1 \in rev(L) &\Leftrightarrow w^R = \sigma_1 \dots \sigma_n \in L \\ &\Leftrightarrow \exists_{A'} q_1, \dots, q_n \in Q. \delta(q_{i-1}, \sigma_i) = q_i \wedge q_n \in F \\ &\Leftrightarrow \exists_{A'} q_1, \dots, q_n \in Q. q_n \in S \wedge q_{i-1} \in \delta'(q_i, \sigma_i) \wedge q_0 \in F' \Leftrightarrow q_0 \in \hat{\delta}'(S, w) \Leftrightarrow w \in L(A') \end{aligned}$$

תרגול 3: הוכיחו ששפת המילים שלא מסתיימת ב-01 או שמספר האחדות זוגי, היא רגולרית. במבנה ביטוי רגולרי: $((0^*10^*)^*00 \cup 11 \cup 10^*10 \cup 10^*11)$.

- ביוון שהשפה הנדרשת מכילה "או", זה מתורגם לאיחוד בביטוי הרגולרי. נטפל בכל חלק בנפרד.
- מצד שמאל – מילים שלא מסתיימות ב-01. או שמדובר במילה היריקה, או שמדובר במילה באורך 1 (כל אות בא"ב), או שמדובר בשרשור של כל דברים שבסיסו 10,11,00 – העיקר לא-ב-01.
- מצד ימין – במות בלחשי של 0, ניקח את ה-1 בזוגות, ונשים כובית מעל (במotaות זוגיות בלחש של 1, יכולה להיות אפס, שתים, ארבע וכו'), ובוין במות בלחשי של 0 שלא משפיעה לנו על התנאי.

תרגול 4: נכון/לא נכון:

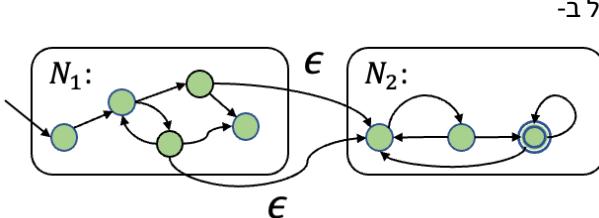
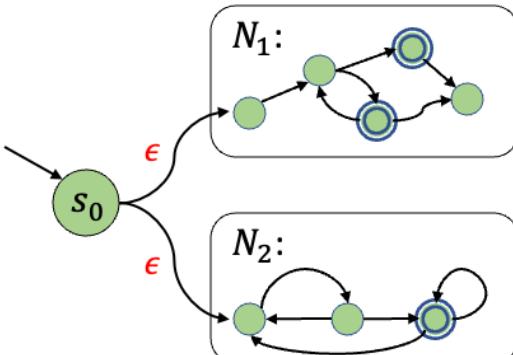
- $L(r) = ((\epsilon + r))^*$: נכון. יכול להיות שב- r אין את המילה ϵ ולכן הביטויים אינם שווים. אולם, ביוון שהבול תחת סגור קליני, זה מכניס את ϵ באופן אוטומטי כי ניתן לחתך רצף באורך 0. לכן הביטויים שווים.
- $(r^*s + s^*r)^* = L$: לא נכון. מצד ימין, לוקחים רצף מילים בלחשו מ- r ואז רצף מילים מ- s . מצד שמאל, כשאנו לוקחים מילים מ- s חיברים לשרשרא מילה מ- s , וזה עבר **איחוד עם מילים מ-s**. עבור $s = 1, r = r$ (בלומר השפות $\{0\}^*, \{1\}^*$) המילה 100 קיימת בצד ימין 10^2 אבל לא ניתן ליצור מצד שמאל 10^1 יותר מ-0 אחד. גם המילה 1 נמצאת מצד ימין, אבל לא מצד שמאל.



שפות רגולריות

סגירות פעולות רגולריות

הפעולות הרגולריות בהן נדון: איחוד, שרשור, סגור קליני. נרצה להוכיח סגירות של הפעולות הללו באמצעות DFA.



סגירות תחת איחוד: הוכחנו באמצעות DFA על ידי ייצור אוטומט המכפלת. בנתן נרצה להוכיח באמצעות DFA. יש לנו שתי שפות, ואנו משתמשים על האיחוד שלהם $L_1 \cup L_2$. נניח בה"כ $\emptyset = Q_1 \cap Q_2$.

בדרך הפשטונה נגדיר $N = (Q, \Sigma, \delta, S, F)$ שבו האיחוד בהינתן N_1, N_2 :

$$\begin{aligned} Q &= Q_1 \cup Q_2, F = F_1 \cup F_2, S = S_1 \cup S_2 \\ \delta|_{Q_i \times \Sigma_\epsilon} &= \delta_i \end{aligned} \quad \bullet$$

דרך נוספת היא עם מעברי ϵ (בתמונה).

סגירות תחת שרשור: אם משתמשים על שרטור $L_1 || L_2$. נשרשור כל מצב מקבל ב-

N_1 למקומות התחלתיים ב- N_2 עם מעברי ϵ . נניח בה"כ $\emptyset = Q_1 \cap Q_2$, ושאין מעברי ϵ בין N_1, N_2 .

נגדיר $N = (Q, \Sigma, \delta, S, F)$ שבו השרטור בהינתן N_1, N_2 :

$$\begin{aligned} Q &= Q_1 \cup Q_2 \\ F &= F_2, S = S_1 \end{aligned} \quad \bullet$$

$$\delta(q, a) = \begin{cases} S_2 & a = \epsilon \wedge q \in F_1 \\ \delta_1(q, a) & a \neq \epsilon \wedge q \in Q_1 \\ \delta_2(q, a) & a \neq \epsilon \wedge q \in Q_2 \end{cases}$$

הוכחת נכונות:

$w \in L_1 || L_2 \Rightarrow w \in L(N)$ •

- יהו $w_1 \in L_1, w_2 \in L_2$ ונראה כי $w = w_1 w_2 \in L(N)$. קיימים מצבים $Q_1 \in L_1, Q_2 \in L_2$ כאשר $a = |w_1|$, המצביעו על ראשון התחלתי, האחרון מקבל, וכל מצב מתתקבל מהקדם לו ע"י פונקציית המעברים. כמובן, קיים מסלול לקבלת המילה w_1 בא- N_1 .

באופן דומה, קיים מסלול דומה לקבלת המילה w_2 בא- N_2 .

- נגדיר $z = w_1 \epsilon w_2 = z_1 \dots z_k$ וסדרת מצבים $b_n, b_0, \dots, a_n, a_0, \dots, a_n, b_0, \dots, a_n, b_n$ כר' ש: $q_0 \in F_1$ וקיים מעבר ϵ בין המצביעים a_n, b_0 לפי הבניה שלנו. לכן מתקיים $z_i = \delta(q_{i-1}, z_i)$ ולכן $w_1 w_2 \in L(N)$.

$w \in L_1 || L_2 \Leftarrow w \in L(N)$ •

- יהי $w \in L(N)$ אז יש $1 \leq k \leq n$ עבורו $\sum_{\epsilon}^k \in \bar{z}$ כך ש- z מתקיים מ- \bar{z} ע"י השמטת ϵ , וקיים q_k, \dots, q_0 כך שמתקיים $q_i \in F_2$.

- טענה: קיים $0 \leq j \leq k-1$ כך ש: $z_0 \dots z_{j-1} z_j \epsilon z_{j+1} \dots z_{k-1} = \bar{z}$ כך ש: $q_0, \dots, q_j \in Q_1, q_{j+1} \in S_2, q_{j+2} \in F_1$ למסובב את התחלתי של N_2 .

כלומר עברנו ממצב מקבל של N_1 למסובב התחלתי של N_2 .

מכאן נובע שנתקבל את הדרישה של $w \in L(N)$ כאשר החלק הראשון הוא בא- L_1 והשני בא- L_2 .

סגירות תחת סגור קליני: נזכר כי L הוא פשוט שרוורי של שרורים של מילים מ- L באורך כלשהו. נרצה ליצור מעברי ϵ מכל המצביעים המתקבלים למקומות התחלתיים. הבעיה – אנחנו לא מקבלים את המילה הריקה! אנחנו לא נרצה לגרום להתחלה להיות גם מקבל (לא נרצה לכפות על דבריהם שעבורים במצב ההתחלה להתקבל – לא רצוי). לכן, נוסיף מצב התחלתי מיוחד שהוא יהיה מקבל.

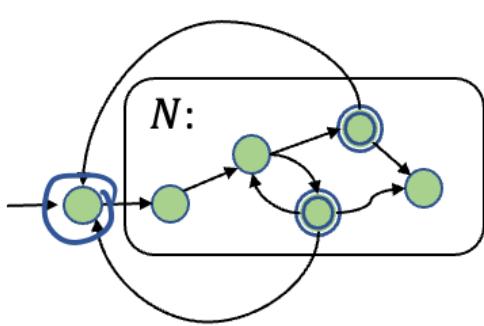
נגדיר $(F', \Sigma, S', \delta', F') = N'$ שבו סגור קליני בהינתן N :

$Q' = Q \cup \{q_0\}$ – הוספנו מצב התחלתי ייעודי. •

$S' = \{q_0\}$ •

$F' = F \cup \{q_0\}$ •

$$\delta(q, a) = \begin{cases} S & a = \epsilon \wedge q = q_0 \\ S' & a = \epsilon \wedge q \in F \\ \delta(q, a) & a \neq \epsilon \wedge q \in Q \end{cases}$$



**ביטויים רגולריים**

ביטויים רגולריים (ב"ר): נרצה לתאר שפה רגולרית באמצעות ביטויים שגדיר אותה. במקרה שלנו, נגדיר אבני בינו בסיסיות: השפות מהצורה $\{s\}$, השפה הריקה, והשפה $\{\epsilon\}$. פועלות היצירה יהיו איחוד, שרשור ואייטרציה. נגדיר באינדוקציה באופן הבא:

אורך	ביטוי רגולרי R מייצג	שפה $L(R)$ שהביטוי מייצג
1	\emptyset	\emptyset
	$\{\epsilon\}$	ϵ
	$\{a\}$	$a. a \in \Sigma$
2	$L(R_1) \cup L(R_2)$	$(R_1 \cup R_2)$
	$L(R_1)L(R_2)$	(R_1R_2)
	$L(R)^*$	(R^*)

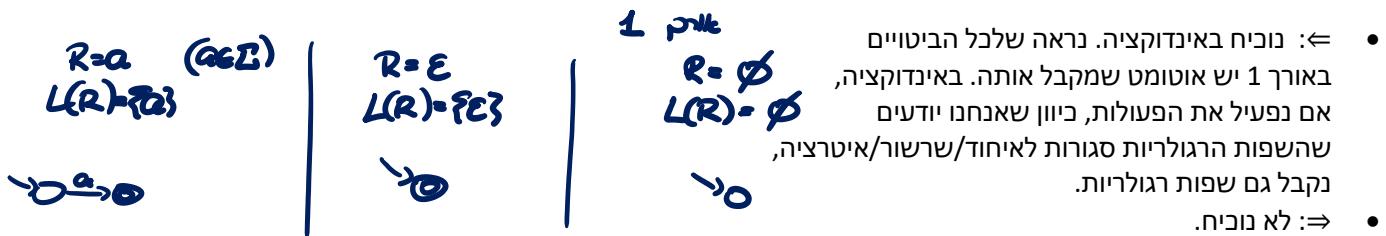
הערות:

- נסמן ב- (Σ) את אוסף הביטויים הרגולריים מעל Σ .
- אפשר לתאר הכל עם סוגרים, אבל הרבה פעמים נשמש את הסוגרים לפי סדר הקידימות הבא:
 - * קודם לכל (חזקת).
 - שרשור (כפל).
 - איחוד (חיבור).

דוגמאות:

- $(ab)^* \neq ab^*$: בכך יмин יש את כל השרשוריים האפשריים של ab , ובצד שמאל יש לנו קודם כל את b^* , אך קיבל a בתחילתיה ועוד אוסף של b -ים.
- $0^* \cup 1^* \cup 0 \cup 1 = (\Sigma)$: נתחיל מכל מחרוזת שניתן להרכיב מ-0 או 1, שרשור של 1, ואז שני תווים מהם או 0 או 1.
- $0^* 10^* \cup 10^* 0^* = \{0,1\}^*$: כל המילימ מעל $\{0,1\}$ שבו יש מספר זוגי של אחדות.
- $0^* 10^* 10^* = \{0,1\}^*$: כל המילימ מעל $\{0,1\}$ שבו יש מספר אי-זוגי של אחדות.

משפט: שפה ניתנת לתיאור ע"י ביטוי רגולרי \Leftrightarrow השפה רגולרית.

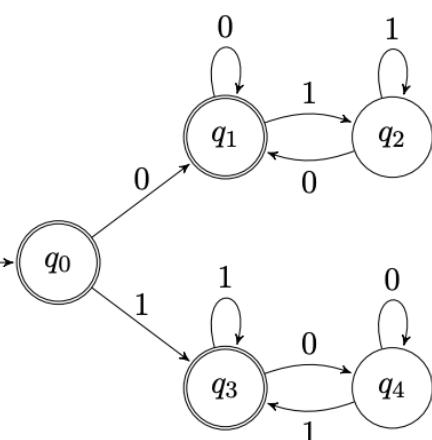
הוכחה:

:= נוכיח באינדוקציה. נראה שלכל הביטויים באורך 1 יש אוטומט שמקבל אותו. באינדוקציה, אם נפעיל את הפעולות, ביוון שאנו יונדים שהשפות הרגולריות סגורות לאיחוד/שרשור/אייטרציה, נקבל גם שפות רגולריות.

⇒ לא נוכיח.

למה הניפוינתבונן בשפות הבאות:

- $\{0^n 1^n\} \geq n = L_1$ – זו אינה שפה רגולרית, אין אוטומט שיכל להוות אותה. צריך לזכור בדיקון כמה אפסים ראיינו, כדי שטול להשוות במספר האחדות ולבודא שהיא אותו מספר. לאוטומט יש מספר סופי של מצבים – **זה לא יכול ללווה כמה אפסים הוא וראה**. הוא יכול ללווה מספר האפסים $2 \bmod 3$ (הזוגיות שלהם), אבל לא בדיק את המספר. לכן, אפשר לבלב אותו עם שתי מילימ בעלות מספר שונה של אפסים והוא הגיע אליו מצב. לאחר מכן, לא ידע בהינתן סדרת אחדות האם לקבל או לא.
- $\{w\} = \#_0(w) = L_2$ – גם זו אינה רגולרית, זה מצב אפלו יותר מורכב מהשפה הקודמת, כי האפסים והאחדות אפלו לא מגיעים ביחס.
- $\{w\} = \#_{01}(w) = \#_{10}(w) = L_3$ – זו דוקא שפה כן רגולרית.



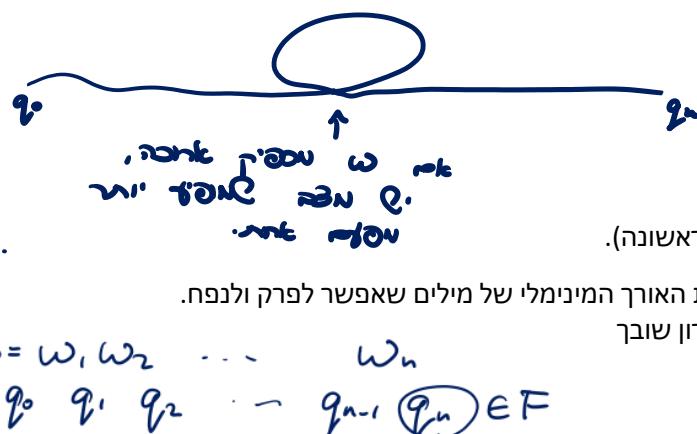
כדי להוכיח שפה היא רגולרית – נראה אוטומט שמצויה אותה, או ביטוי רגולרי מתאים. אבל אנחנו צריכים כל שיאפשר לנו להוכיח שפה אינה רגולרית, אפין מסוימים של שפות רגולריות. הכל שראנו נראה **למה הניפוי (pumping lemma)**: אם לשפה יש אוטומט סופי שמצויה אותה יש תכונה שמקבלים מותן האוטומט, וכדי להוכיח שאין אוטומט נראה שהתכונה לא מתקינה.



אינטואיציה: נתבונן ביריצה של אוטומט עם מספר מצבים נתון, על מילה מאוד ארוכה ($M \in L$) $w = q_m q_{m-1} \dots q_1 q_0$. אם ניקח מילה מסווג ארוכה, בריצה של האוטומט נצטרך לחזור על אותו מצב פעמיים – לפחות מקום אחד שיש בו לולאה. האוטומט לא מסוגל לזכור שהוא עדין בלולאה, או שהוא היה וכמה פעמיים.

ניקח את מילת הקטלן, ונמצא את החלק שבו האוטומט בתור הלולאה. יצא את החלק הזה מהמיליה, והאוטומט עדין קיבל את המיליה. או שניקח את החלק הזה ונחזיר עליי כמה פעמיים, והוא יסתובב בלולאה כמה פעמיים ועודין יקבל את המיליה. אנחנו ניקח חלק של המיליה וננפח את המיליה – ועדין זה יתקיים.

למה הנימוק: לכל שפה רגולרית L , **קיים אורך $\ell \geq 0$** , כך שכל $L \in w$ באורך $\ell \geq |w|$, ניתן כתוב $zyx = w$ nasuch:



1. לכל $0 \leq i$ מתקיים $L \in zy^i z$.
2. $0 > |y|$ (פירוק לא טריויאלי – לא ריק).
3. $|y| \leq |xy|$ (שהלולאה לא תקרה מאוחר מדי, ניקח את הראשונה).

הוכחה: יהיו M אס"ד כך $S(M) = L$. ניקח $1 + |Q| = \ell$, להיות האורך המינימלי של מילים שאפשר לפרק ולנפח. תהיו מילה $L \in w$ באורך $\ell \geq |w|$. האוטומט M רץ על w . לפ"ז עקרון שובר היונים יש מצב שמוופיע פעמיים, קיימים $k \neq j$ כך $q_j = q_k$. נבחר $k < j$ מינימליים ככלא ואז $\ell \leq j, k \leq \ell$.

נגדיר: $w_n w_{n+1} \dots w_k, z = w_{k+1} \dots w_\ell, y = w_{\ell+1} \dots w_j, x = w_1 \dots w_j$.

- תנאי 2 מתקיים כי $k \leq j$: אפילו אם $k = j + 1$ ניקח את w_{j+1} .
- תנאי 3 מתקיים כי $\ell \leq j, k = |xy|$.

תנאי 1: טענת עזר – לכל i מתקיים $y^i = q_j = q_k = q^{i\delta}$, ביוון שאנוינו רצים בלולאה (הוכחה באינדוקציה על i).
נשתמש בה כדי לאפיין את מה שהאוטומט עשווה על $z^{i\delta}$:

$$\hat{\delta}(q_0, xy^{i\delta}) = \hat{\delta}(\hat{\delta}(q_0, x), y^{i\delta}) = \hat{\delta}(\hat{\delta}(q_j, y^i), z) = \hat{\delta}(q_k, z) = q_n \in F$$

oir משתמשים בلمת הנפקות: כדי להוכיח $S-L$ אינה רגולרית. נניח בשילילה $S-L$ כן רגולרית. אז מקיימת את למת הנפקות. נרצה להוכיח שזה לא מתקיים (להוכיח את ההיפך): **לכל ℓ , קיימת** w כך **שלכל פירוק** $zyx = w$ **לא מתקיים** $1+2+3$. נניח $3+2+1$ ונסטור את 1 על ידי כך שהמיליה לא תהיה בשפה. כדאי לבחור בתור w מילים שונות בשפה אבל ממש "על הקשகש" כדי להיות בשפה.

הדוגמה	השפה L	הוכחה
1	$\{0^n 1^n : n \geq 0\}$	<p>נניח בשילילה $S-L$ רגולרית. יהיו ℓ אורך הנפקות. נבחר $1^\ell 0^\ell = w$. יהיו פירוק $zyx = w$ שמקיים את תנאים 1 ו-2. נחלק למקרים:</p> <ol style="list-style-type: none"> 1. ב-y יש רק אפסים. ניקח $i = \ell$ ומתקיים: $\#_0(xy^2 z) > \#_0(xyz) = \#_1(xy^2 z) = \#_0(xyz)$. 2. ב-$y$ יש רק אחדות. באופן דומה. 3. ב-y יש גם 0 וגם 1. במליה $z^2 ux$ יש אפסים אחרים ולכן לא באפשרות.
2	$\{w : \#_0(w) = \#_1(w)\}$	<p>באנו ביעזר בתנאי 3, ונאמר ש-y מופיע בהתחלה ולבן מכיל רק אפסים. בראנו נסיק שב-xy ובפרט ב-y יש רק אפסים.</p> <p>נבחר $i = \ell$: לכן $\#_1(xy^2 z) > \#_1(xyz)$ במו קדם.</p> <p>(בעצם, זה בדוגמה הקודמת לא נעזרנו בתנאי 3, גרמה לנו לחלק ליווור מקרים.)</p>
3	$\{j > i : 1^j 0^i\}$	<p>נניח בשילילה $S-L$ רגולרית. יהיו ℓ אורך הנפקות. נבחר $1^{\ell-1} 0^\ell = w$. יהיו פירוק $zyx = w$ שמקיים את כל התנאים. כמו בדוגמה הקודמת, ב-y יש רק אפסים: $1 \geq (\#_0(y))$.</p> <p>נבחר $0 = i$ ונקבל כי אין יותר אפסים מחודדים:</p> $\#_0(xy^0 z) < \#_0(xyz) = \ell \Rightarrow \#_0(xy^0 z) = \ell - 1$ <p>בנוסף: $1 = \#_1(xy^0 z) = \#_1(xyz) = \ell - 1$</p> <p>לכן מתקיים $L \notin z^0 y^0 z = xy^0 z \notin \#_1(xy^0 z) = \#_0(xy^0 z)$</p>
4	$\{w : w \text{ is prime}\}$	<p>נניח בשילילה $S-L$ רגולרית. יהיו ℓ אורך הנפקות. נבחר $1^p = w$ עבור $p \geq \max\{\ell, 2\}$. יהיו פירוק $zyx = w$ שמקיים את תנאי 2.</p> <p>כדי שזה לא יהיה ראשוני בהכרח נבחר את $1 + p = i$ ונקבל כי $p = y + z$.</p> <p>מאורך לא ראשוני ולבן $L \notin z^i$.</p>

הכח של למת הניפוי: אם L רגולרית \Leftarrow היא מקיימת את תנאי למת הניפוי. הכוון ההпро אינטנסיבי האם לכל שפה לא רגולרית, לא יתקיימו תנאי למת הניפוי ונגיע למסקנה שהיא רגולרית? לctrano לא. הולמה זו היא לא שלמה.

נסתכל לדוגמה על השפה הבאה (שאינה רגולרית): $\{ab^n c^n\}^*$. המילים בשפה מתחילה במעין סימן. אם יש a בודד המילה צריכה להיות בחלק הראשון עם מספר דומה של b ו- c . אם אין a או יותר מ-1, אנחנו בחלק השני שבו יכולים להיות b ו- c באופן שרירותי.

למת הניפוי מתקיימת **כאן**: אפשר לנפח בקבוקות את a - a . אם ניקח מילה בחלק הראשון, לנפח את a - a וזה יbia אותנו לחלק הימני של השפה, וזה עדין יהיה בשפה. החלק השני הוא שפה רגולרית שתמיד אפשר לנפח אותה.

נבחר $2 = \ell$. תהיו $L \in \Sigma$. נראה שקיים פירוק $xyz = w$ שמקיים את תנאי הניפוי.

- אם w מתחילה ב- a בודד: $b^n c^n = a, z = \epsilon, y = a, x = \epsilon$. מתקיים $0 > |y|, \ell \leq |yx|$. יהי i .
- אם $1 \neq i$ המילה $z^i xy$ מתחילה ב- $1 \neq k$ פעמים a ואחרי b רצף כלשהו של c , b ולכן היא בחלק הימני.
- אם $1 = i$ לא שיכינו את המילה והוא עדין בשפה: $L \in \{xyz = xy^i z\}$.
- אם w מתחילה ב- a^k .
- אם $0 = k$ נבחר $\epsilon = x$ ו- y להיות האות הראשונה, ניפוי לא יצא אותנו מהשפה.
- אם $2 \geq k$ נבחר $aa = x$ ו- y להיות האות השלישי. לכל i המילה $z^i xy$ מתחילה ב- aa ואחרי b או יותר a -ים, ולבסוף סיפא מ-* $\{b, c\}$. לכן $L \in \{xyz = xy^i z\}$.

כליים אחרים להוכחת א-רגולריות:

פעולות רגולריות:

נניח בשילוליה שהשפה L רגולרית.

או לבב שפה רגולרית אחרת L' השפה $L' \cap L$ רגולרית (כאן לקחנו חיתוך). ניקח את השפה הבאה: $\{aw: w \in \{b, c\}^*\} = L'$. ניתן לרשום לה ביטוי רגולרי: $L' = L(a(b \cup c)^*)$. החיתוך אותה ייעף לנו את החלק הימני. נקבל $\{ab^n c^n: n \geq 1\} \subseteq L' \cap L$. קיבלונו שפה לא רגולרית (ניתן להוכיח עם למת הניפוי) בסתירה, ולכן השפה המקורית שלנו L לא רגולרית.

מחלקות שקולות:

אופן הדוק של השפות הרגולריות. האינטואיציה היא שאוטומט M "זכור" רק את המצב הנוכחי. אם שתי מילims מגיעות באותו מצב, האוטומט לא מביחס ביניהן. אם שפה L היא רגולרית, היא משרה מספר סופי של מחלקות שקולות בר ש: $u \equiv_L v \Leftrightarrow \exists z \in \Sigma^* u z \in L(M) \Leftrightarrow v z \in L(M)$.

- L-שקלות: תהיו $\Sigma \subseteq L$, נאמר כי Σ $\subseteq L$ שקולות אם $\forall x \in \Sigma \exists y \in \Sigma$ $x \in L \Leftrightarrow y \in L$. נסמן כי Σ / \sim_L הוא שקולות \sim_L , ונסמן ב- \sim_L את מחלוקת השקלות של L . נסמן את קבוצת המנה $\{\Sigma^*/x: x \in \Sigma\}$.
- A-שקלות: יהי A אס"ד, נאמר כי $\Sigma \subseteq A$ שקולות אם האוטומט מביא אותו לאותו מצב: $(q_0, x) = \hat{d}(q_0, y)$.
- נשים לב כי מתקיים $|Q| \leq |\Sigma^*/\sim_L|$, כלומר מטען לנו מחלוקת שקולות.

טענה: נניח $u \sim_L v$, אז מתקיים $u \sim_L x$ בולם $\hat{d}(u, x) = \hat{d}(v, x)$ הוא עידן של \sim_L . הוכחה: פורמלית באינדוקציה. אם $u \sim_L v$ אז $\exists z \in \Sigma^* u z \in L \Leftrightarrow v z \in L$.

משפט מייל-נרד (Myhill-Nerode): אופן הדוק של השפות הרגולריות. L רגולרית $\Leftrightarrow |\Sigma^*/\sim_L|$ סופית.

מסקנה 1: $|Q| \leq |\Sigma^*/\sim_L|$. שימוש:

- נוכחים כי התוקורה האקספוננציאלית במובן אס"ד לאס"ד הכרחית. יהי $\mathbb{N} \in a, \dots, n$ ותהיו $\{1, \dots, n\} \subseteq x$. עבור $\Sigma \in x$ נסמן S_x את קבוצת התווים המשתתפים ב- x : $S_x = \{s \in \Sigma: s \in x\}$. נגדיר שפה $\Sigma \neq S_x = L$. בולם, שפה מכילה את כל המילים שבחן חסר לפחות אחד מהא"ב.
- קיימים אס"ד בין n מצבים Über L , אך כל אס"ד המדזה את L הוא בעל n^2 מצבים. נראה כי לכל $\Sigma \in y$, אם $S_y \neq S_x$ (ושלחותתו אחת שמיופיע באחת אבל לא בשניה), אז מתקיים $L[y] \neq L[x]$. זה מספיק כי מספר תתי הקבוצות S_x הוא n^2 .
- נניח בה"כ כי $\overline{S_x} \not\subseteq \overline{S_y}$. ניקח $\Sigma \in z$ שיפריד בין המילים, כך ש- $\overline{S_x} = S_z$ או $\overline{S_x} \subseteq S_z$. אם $S_{xz} = S_x$ ו- $S_{yz} = S_y$, חסר לנו תו כלשהו של $\overline{S_y}$ ולכן $L \in \Sigma$. נסתכל על $\overline{S_y} \not\subseteq \overline{S_x}$, $S_{yz} = S_y$ ו- $S_x \subseteq \overline{S_y}$.

מסקנה 2: אם L רגולרית אז $|\Sigma^*/\sim_L|$ סופית. שימוש:

- נראה כי $\{0 \geq n, m \geq 0, i \geq 1\} = L$ אינה רגולרית ע"י שנראה כי $|\Sigma^*/\sim_L|$ אינסופית.
- לכל $m \neq n$ מתקיים $L[m] \neq L[n]$. מה שיכניס את ab^2 לשפה הוא c^2 והוא לא יכניס את ab^3 לשפה...



שאלות אלגוריתמיות:

1) בהינתן אסל"ד N ומילה w , האם $w \in L(N)$?

- אפשרות אחרת: נתונים אס"ד M שקיים $L(M) = L(N)$ ואז אפשר להריץ את M על w ובודקים אם הגיעו למצב מקובל. יש כאן חוסר יעילות בגלל שהאס"ד דורש בינוי אקספוננציאלית.
- אפשרות שנייה: אם נריץ ישירות את האסל"ד w עץ של אפשרויות, עם מעברי עץ בערך בכלל לא חסום.

(2) האם $\emptyset = ?L(N)$ (ישיגות, reachability)

- בහינת הגרף המבוקן שמייצג את האוטומט, האם קיים מסלול שמתחלם במצב תחيلي ומסיים במצב מתקבל?

(3) האם $\Sigma^* \subseteq ?L(N)$

- בצע דוקציה לבעה שאנו בבר יודעים לפתור. השאלה כאן שקולה להאם המשלים של השפה הוא ריק.
- בבנה אס"ד M כר-ש- (N) , $L(M) = L(N)$ שמחה את המשלים $\overline{L(M)}$ ונבדוק האם $\emptyset = ?L(M)$.

(4) בהינתן N_1, N_2 האם $?L(N_1) \subseteq L(N_2)$

- תנאי שקול הוא $\emptyset = ?L(M) = L(N_1) \cap \overline{L(N_2)} = L(N_1) \cap L(N_2)$. בונים M כר-ש- (N_1) ובודקים האם $\emptyset = ?L(M)$.

תרגול 4 (שפות רגולריות)

כלים להוכחה שפות אין רגולריות:

- למת הניפוי.
- משפט מייהיל-נרווד.
- תכונות סגור.

הדוגמה	השפה	הוכחה
1	$\Sigma = \{0,1\}^n$	<p><u>למת הניפוי:</u> נניח בשיליה ש-L רגולרית. יהיו ℓ אורך הניפוי. נבחר $z = xy$ כך ש-$\ell \leq xy$ גם x וגם y מכילות רק אפסים. נסמן $k = y$. נבחר $i = 2$ ונקבע: את המילה $L \notin 0^{\ell+k} 10^\ell z = 0^{\ell+k} 10^\ell xy$.</p> <p><u>למת הניפוי:</u> נניח בשיליה ש-L רגולרית. יהיו ℓ אורך הניפוי. נבחר $z = xy$ כך ש-$\ell \leq xy$, גם x וגם y מכילות רק אפסים. נסמן $\ell \leq k = y$. נבחר $i = 2$ ונקבע: $z = xy^2z = 0^{\ell^2+k} 10^\ell xy^2z = 0^{\ell^2+k} 10^\ell z$ ולכן $n^2 < n^2 + k < n < \ell$ בסתרה.</p>
2	$\{w: \#_0(w) = \#_1(w)\}$	<p><u>הוכחנו את הטענה בשיעור לפי למת הניפוי.</u></p> <p><u>תכונות סגור:</u> נניח בשיליה כי L רגולריות ונשים לב כי: $\{0^i 1^{n-i} : i \geq 0\} \cap L(0^* 1^*) = \{0^i 1^{n-i} : i \geq 0\} \cap L$</p> <p>ביוון שגם L וגם $L(0^* 1^*)$ רגולריות (מכיוון שהיא מוגדרת ע"י ביטוי רגולרי), החיתוך שליהן הוא רגולרי. אמנם, השפה שקיבלנו אינה רגולרית (הראנו באמצעות למת הניפוי בהרצתה), בסתרה. לכן קיבלנו כי L המקורית שלנו אינה רגולרית.</p> <p><u>מייהיל-נרווד:</u> נראה כי יש אינסוף מחלקות שקולות ונסיק כי L לא רגולרית. נמצא אינסוף מילים כאשר כל אחת נמצאת במחלקה שקולות מילה. נסתבל על המילים מהצורה $0^j z$ עבור $j \neq i$ נטען כי $0^j z \sim 0^i z$ כי עבור $i > j$ מתקיים $0^j z \notin L$, $0^i z \in L$.</p>

טענה על למת הניפוי: תהינה A ו-B שפות המקיימות את למת הניפוי. הוכח/הפרך: השפה B ע"א מקיימת את למת הניפוי.

הוכחה: הטענה נכונה. יהיו ℓ_A, ℓ_B הקבועים המתאימים בלמת הניפוי. נבחר $\max\{\ell_A, \ell_B\} = \ell$ ונראה כי L ע"א מקיימת את למת הניפוי בעבורו. לכל מילה $B \in A$ ע"א $\#_0(w) \geq \#_1(w)$:

- אם $A \in w$ אז ביוון ש- $\ell_A \geq |w|$ מתקיימים תנאי למת הניפוי: קיים פירוק $z = xy$ ע"א $0 \leq |x| \leq \ell_A$ ו- $|y| \leq \ell_A - \ell_A \leq \ell$.
- ולכל $0 \leq i \leq \ell$ מתקיים $z \in A \subseteq B$ במדדש (המילה שיכת לשפה A ולכן שיכת גם לאיחוד של A ו-B).
- אם $w \in B$ אז הטענה סימטרית לחולטיין.

2 – תורת החישוביות

מבנה טיריניג

מבנה טיריניג (מ"ט)

מבנה טיריניג אוטומט סופי בשילוב עם דיבורן לא חסום. ראיינו שאוטומט סופי בלבד זה לא מספיק לחשב הכלול. ראיינו דוגמאות לשפوت שעם הוכנת מחשב קל להזות אותן, אבל עם אוטומט סופי אי אפשר להזות אותן. אי אפשר להסתפק בדיבורן חסום מראש.

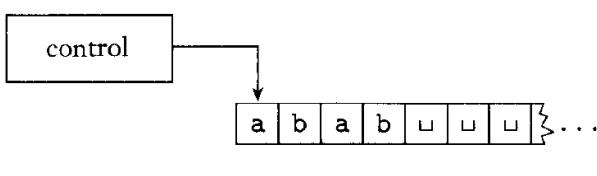
- הצורה שהדיבורן מקבל במבנה טיריניג הוא **סרט (tape)**: בכך שמאלו הוא חסום, ובצד ימין הוא נمشך לנצח. הוא מחולק לתאים, כאשר בכל תא אפשר כתוב סימן אחד מתוך אוסף סופי של סימנים (אלף בית). תאים שעוד לא כתבנו בהם שום דבר, האתחול שלהם הוא להיות "blank": ב.
- בתחלת החישוב: הקלט כתוב בתחילת הסרט וראשו נמצוא בתא מסויים של הסרט. אפשר הכתוב/לקרוא ולזרע אחד ימינה או **שמאלה**. במקביל להתנהלות מול הסרט, המכונה יכולה גם לשנות **מצב פנימי** שלו, והוא קובע את התנהלותה שלא (כמו אוטומט).
- השפעת צעד אחד של המכונה:
 - שינוי **המצב הפנימי** של המכונה.
 - שינוי **תוכן הסרט** במיקום הנוכחי של הראש (רק במקום אחד).
 - שינוי **מיקום הראש** (רק באחד, ימינה או שמאלה).

מתי החישוב מסתיים? באוטומט החישוב הסטיים בשיסייננו לקרוא את הקלט. מבנה טיריניג יכול להיעשות לפני הסרט ולקרוא אותו מספר פעמים, לבתו עלי, לבתו את הקלט, והוא מסיימת את החישוב כשהיא מחייב (מקביל ל-*return* בקוו). למונט טיריניג **יהיה מצב מקבל ומצב דוחה**. כל עוד לא הגיעו לאחד המצביעים האלה החישוב לא הסטיים, יוכל גם לא לעצור.

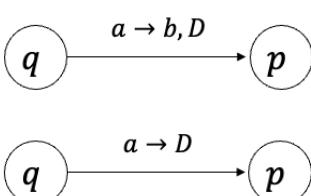
דוגמה: מבנה טיריניג עבור השפה $\{w \in \Sigma^* \mid w \# w = L\}$.

- לשם פשוטות נניח שיש רק # אחת, ומתקיים $\Sigma \neq \#$.
- **מוצאים את האות השמאלית ביותר שאינה X:** התכנית הכללית היא להתחיל מהאות הראשונה, ולזוזה שמה שMOVIPU אחר # אותו דבר, למחוק את שני הסימנים הללו. כך נמשיך עבר האות השנייה ואילך עד שנשווה את כל האותיות. אורך נמדד את ההשוואה בין אות ? לבין אות ? אחריו #?
- **זכרים את האות במצב ומחליפים אותה ב-X:** אין לשמור את התו הראשון לטובה ההשוואה מול המקביל לו? בעזרה מצב מיוחד עבר התו זהה (במקום משתנה מקומי ב-*python*). יש מספר סופי של אפשרויות/ מצבים – כי האלף בית תמיד סופי.
- אחריו שנסתכל על התו w_1 (נמחק אותו – נשים במקומותו מיוחד X) המכונה תעבור למצב שזכור אליותו זו היה, ושובכשו היא צריכה ללבת ימינה לאחרי ה-, והיא תשווה את מה שאחרי ה-#. עם מה שהיא בזיכרון. אם יש התאמה – **נחלה ב-X, אחרת – דוחה.**
- עבשו נחזור אחריה עד שיש לנו את האחרון שנמחק (על ידי התו X). משם נקרה את התו שלו, ואז נמשיך עד ל-#.
- מימינה נמשיך לקרוא את כל ה-X-ים, ואז נשווה את התו הבא באותו אופן.
- אם במעבר האחרון אנחנו רואים שאחרי ה-# לא מחקנו הכלול: המילים לא באותו אורך – דוחה. אם להיפך, נקרה אות w ואחרי ה-# אנחנו מוצאים רק X-ים – דוחה. המצב מקבל היחיד הוא שהצלחנו למחוק את כל מה שלפני #, ואת כל מה שאחרי #.
- נשים לב שם שמסמן לנו את סוף הקלט הוא ב.
- נגידר א"ב סרט: $\{p, X, q\} = \Gamma$.

הגדרה פורמלית של מבנה טיריניג: מבנה טיריניג מוגדרת $\langle Q, \Sigma, \Gamma, \delta, q_0, q_r, q_a \rangle = M$ כאשר:



- Q היא קבוצת מצבים סופית.
- Σ הוא א"ב קלט סופי שאינו מכיל את הסימן ב (blank).
- Γ הוא א"ב סרט סופי ברשותה: $\Gamma \subseteq \Sigma \cup \{\#\}$.
- δ היא פונקציית מעברים: $\{L, R\} \times \Gamma \times \Gamma \rightarrow Q \times \{L, R\}$.
- מצבים מיוחדים: q_0 תחורי, q_a מקבל, q_r דוחה ($q_a \neq q_r$).



איך מציררים מבנה טיריניג?

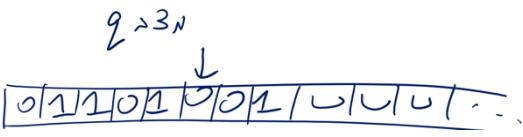
- $(p, b, D) = \delta(q, a)$.
- קיצור עבר $(p, a, D) = \delta(q, a)$ – לא משנהם את תוכן הסרט.
- קיצור עבר $(\dots, a, D) = \delta(q, a)$ – מעבר למצב הדוחה, לא נציג חץ עבר $\dots \rightarrow a$.



חישוב של מכונת טיריניג

חישוב של מילט Σ^* : החישוב מורכב מסדרה של snapshots/configurations, בכל נקודה אנחנו יודעים באיזה מצב המכונה, מיקום הראש, תוכן הסרט. כך אפשר לדעת מה עבר על המכונה במהלך החישוב.

- בתחילת החישוב: הקלט Σ^* הוא מופיע בתחילת הסרט, שאר הסרט מכיל סימני \square , הראש בתחילת הסרט (צד שמאל), המצב הוא q_0 . כדי לתאר את הקונפיגורציה זו, מבינות תוכן הסרט Γ כמיקום הנוכחי של הראש.
- צעד החישוב: מצב $\{q_a, q_r\} \in Q^*$, קוראים את תוכן הסרט Γ כמיקום הראש הנוכחי של הראש.
- אם $(X, b', q') = (q, a)$ אז דורסים את תוכן הסרט על ידי b , עוברים למצב q' , ואם $R = X$ ימינה (אחרת שמאל). בכל מספר סופי של צעדים המכונה יכולה לבתוב מספר סופי של סימנים, לבן יוכל גם אז להציג את תוכן הסרט המעניין בזורה סופית.



קונפיגורציה: מחרוזת ב- $Q^*\Gamma$, מברך נסיק את המצב הנוכחי, את תוכן הסרט, ואת מיקום הראש (התו שעלי ממקום הראש הוא מימין לאות q שמייצגת את המצב הנוכחי). המחרוזת בנוייה שלפני הראש על הסרט, המצב, ומה שמהරאש והלאה על הסרט. לדוגמה: 01101q001.

הערות לגבי תחילת הסרט:

FIGURE 3.4
A Turing machine with configuration 101101111

- אם הראש מנסה לו זוז שמאלה מתחילת הסרט? ישן מספר אפשרויות: המכונה קורסת (segmentation fault), המכונה מקבלת חוויה שאין לה لأن זוז והוא לא זהה, או שלא קורה כלום (האפשרות הכי מינימלית שאנו נבחר בה).

- נשים לב שהמכונה לא יודעת מתי היא בתחילת הסרט – אחרי שהתחיל החישוב, והראש זו ימינה ואז שמאלה אין דרך לדעת מתי חזרנו לתחילת הסרט, אלא אם נכתב בו משהו מיוחד, נסמן אותו.

מעבר בין קונפיגורציות: אם יש לנו שתי קונפיגורציות C, C' , מתי המכונה תעבור מ- C בצעד הבא ל- C' ? היא מפעילה את פונקציית המעברים δ , ומה שיצא לנו זה הקונפיגורציה C' . בדוגמאות הבאות: $\Gamma \in Q^*, a, b \in \Sigma, u, v \in \{q, p\}$.

דוגמאות:

- אם הראש נמצא במקום ימני ביותר (אחרי כל המידע), הסימן שהוא קורא הוא p . למשל עבור $p'v = uaqbv, C' = upab'v$. כדי שזה יהיה צעד החישוב, נגדיר $(p, b', L) = (q, b, R)$. הראש הולך שמאלה.
 - נגדיר: $C = uaqbv, C' = uab'pv$. הראש הולך ימינה.
- "מרקם מיוחד":

- אם הראש נמצא במקום ימני ביותר (אחרי כל המידע), הסימן שהוא קורא הוא p . למשל עבור $p'v = uaqbv, C' = uab'v$. נגדיר $(p, b', R) = (p, q, \delta)$.
- המצב המקורי, אם הראש נמצא במקום השמאלי ביותר (בתחילת הסרט). למשל עבור $pbun = qau, C' = pbun$. נגדיר $(q, a, L) = (p, b, R)$. אין לנו لأن זוז שמאלה, ולכן נשארנו במקום.

הגדרה פורמלית: נסמן $head(C)$ את מיקום הראש בקונפיגורציה C . מתי $xqax' \in Q^*\Gamma$ עוברת ל- $yqay'$ אם $D = ypy' = head(C) = head(C')$.

$p' = q'$

המחרוזות $(xx') - (yy')$ זהות פרט למיקום ה- d .

$d = head(C) - head(C')$ במקומות החדש כתוב התו d .

- אם $head(D) = head(C) + 1$ אז $Z = R$ (המיקום קטן ב-1), אם $head(D) = head(C) - 1$ אז $Z = L$ (המיקום גדול ב-1).

יצאי דוף:

- תחילת הסרט: אם $1 = head(C) = head(C') = (q, x'_1, \delta)$ אז $D = (q, x'_1, L)$.
- סוף הסרט: אם $C = xqax' \Leftrightarrow D = xq$ לפי הכללים הרגילים של δ עוברת ל- S .

קבלה או דחיה של מילת קלט:

- רצף של קונפיגורציות C_k, \dots, C_1 הוא חוקי עבור מ"ט M ומילת קלט Σ^* אם: C_1 היא קונפיגורציה תחילית עבור M , $w, q_0, C_1 = C_1$. לכל $i = 1, \dots, k$, הקונפיגורציה שנמצאים בה C_{i+1} התקבלה מהקדמת i לפני הכללים שהגדנו.
- רצף של קונפיגורציות הוא מקבל אם הוא מסוים בקונפיגורציה מקבלת, ודוחה אם מסוים בקונפיגורציה דוחה.
- המ"ט M מקבלת את Σ^* אם קיימים רצף קונפיגורציה חוקי ומתקבל עבור w .
- קונפיגורציה מקבלת – המצב הוא a .
- קונפיגורציה דוחה – המצב הוא r .



המחלקות R ו-RE

עבור מילה $\Sigma \in \Delta$ ומ"ט M , יש שלוש אפשרויות:

1. M מקבלת את Δ .
2. M דוחה את Δ .
3. **M לא עוצרת...**

השפה של M : השפה ($L(M)$) היא קבוצת המילים ש- M מקבלת, כלומר עצמה עליהן וקיבלה אותן. אם המכונה לא עצמה על המילה, המילה לא בשפה.

אם נסתכל על מ"ט עם שפה נתונה, זה לא בהכרח אומר שהמכונה הזאת היא מאוד שימושית – נניח שביקשו מ"ט שהשפה שלה היא **המספריים הראשוניים**. כשמרכיבים את המ"ט על מספר, מובטח שאם המספר הוא ראשוני המ"ט תעוצר ותקבל אותו, וגם להיפך כל מה שהוא מ"ט עוצרת ומתקבלת הוא מספר ראשוני. אבל, מה קורה אם נתונים בקלט מספר לא ראשוני? אולי היא לא תעוצר. לא ניתן לדעת את זה מראש, ובאופן נקודה בזמן החישוב שלה. יכול להיות שהמ"ט תעוצר ותקבל, יוכל להיות שלא. לא בהכרח אפשר להשתמש במכונה הזאת כדי לדעת אם מספר נתון הוא ראשוני או לא.

המחלקה RE ([recursively-enumerable, Turing-recognizable](#)): מחלקת השפות L עבורן קיימת מ"ט M שהשפה שלה היא $L = L(M)$, כלומר M מזזה/מקבלת את השפה ($M(L)$). במחלוקת זו ישן מ"ט שאומරת כן רק בשערין, ולא בהכרח עוצרות כשהתשובה היא לא.

אינטרואיציה: זה אומר שאפשר ליצור את המילים בשפה (כל מילה שהיא מייצרת היא בשפה, וכל מילה בשפה היא גם תייצר), אבל בהינתן מילה לא נוכל בהכרח להגיד האם היא בשפה או לא. למשל: **שפט כל המשפטים המתמטיים הנכונים**. איך ניתן את המילים בשפה? נNUMBER על כל הטענות באורך הולך ועולה (שורה אחת, שתי שורות...). אם ההוכחה תקינה (МОודאים שככל שורה נובעת לפיה כללים לוגיים מהשורה הקודמת), נפלוט אותה. אם ניתן משפט לא נבעז? המ"ט לא בהכרח מדחה אותו, יכול להיות שלא תעוצר...

המחלקה R ([recursive, Turing-decidable](#)): מחלקת השפות L עבורן קיימת מ"ט M שעוצרת על כל קלט, שהשפה שלה היא $L = L(M)$. במחלוקת זו ישן מ"ט שתרמיד יודעת לעזר בצורה נחרצת ולהגיד כן או לא – להכריע את השפה.

אינטרואיציה: על כל קלט המ"ט תמיד עוצרת, מקבלת/דוחה. שפה שניתן להכריע אותה – משפטים שהם נכונים (כמו קודם) ויש להם הוכחה באורך 10 צעדים. נבדוק את כל הטענות באורך עד 10, ואם מצאנו הוכחה תקינה למשפט נקבל, אחרת נדחה.

הערות:

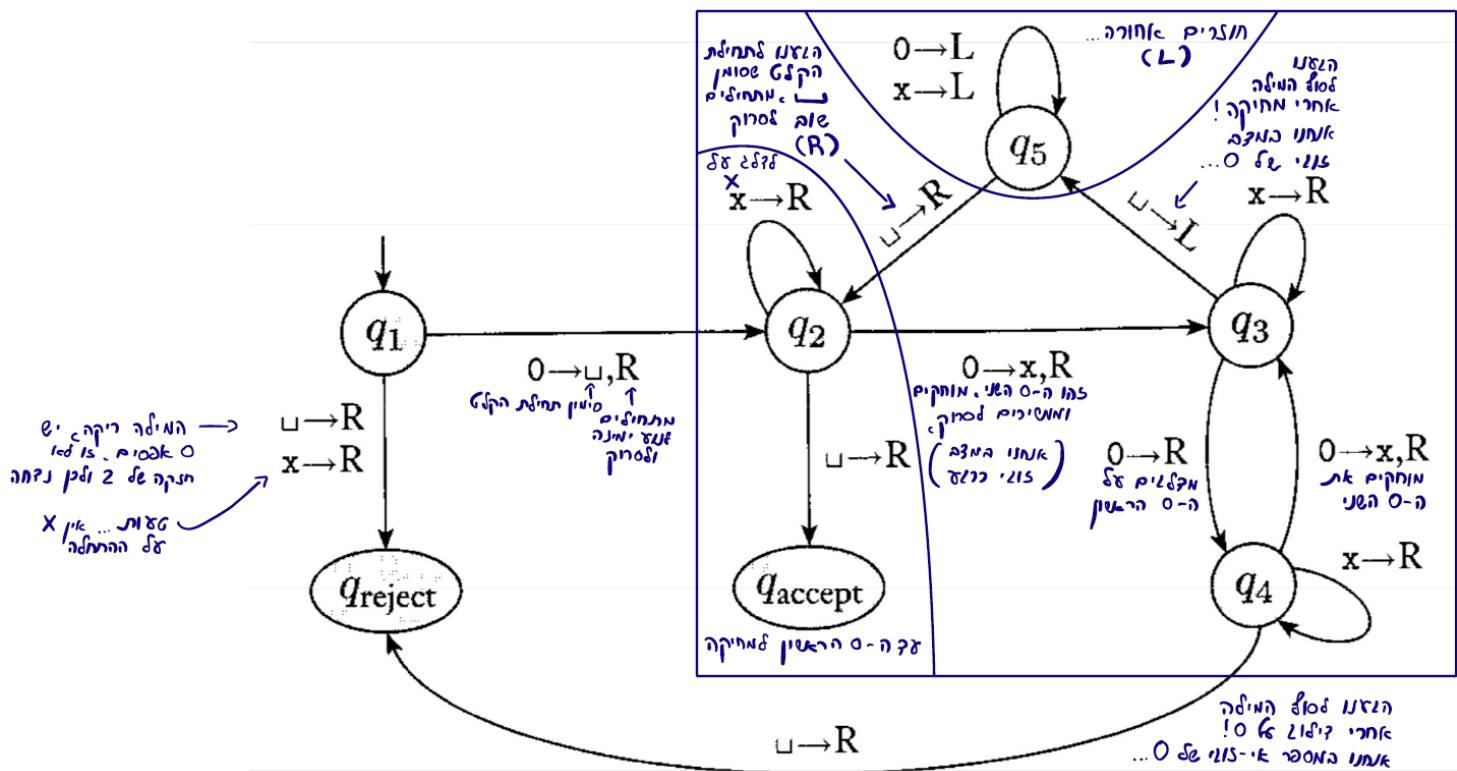
- נשים לב כי מתקיים $RE \subset R$, נוכיח זאת בהמשך.
- מאיפה באו השמות R ו- RE ? לוגיים בעבר רצוי להימנע מהגדירות מזוירה: מ הוא מספר זוגי אם קיימים $N \in m$ כך $2m = n$. במקום זאת, ישנה הגדרה וקורסיבית: 0 מסטר זוגי, $0 > n$ הוא מסטר זוגי $\Leftrightarrow 2 - n$ הוא מסטר זוגי.
- RE : קיימים אלגוריתם וקורסיבי שמייצר את המילים את המילים בשפה אחת אחרי השניה.
- R : קיימים אלגוריתם וקורסיבי שבהינתן מילה, תמיד עוצר, ומכריע אם המילה בשפה.



דוגמאות

דוגמה 1: השפה $\{0^n : n \geq 0\}$ (זוגמה לשפה שאינה רגולרית וניתן להכריע אותה ע"י מ"ט)

- נסמן את תחילת הקלט ב-L שנדע לאן לחזור (די שגרתי). נשים לב כי **ה-0 הראשון מתחלף ב-L**.
- נניח (על ידי החלפה בתו X) כל 0 שניי (אחרי ה-X-ים שכבר קיימים) לסירוגין, כדי לחלק את אורך המחרוזת ב-2. למשל אם נתחליל עם 8 אפסים, בריצה הריאשונה על הקלט משמאלו לימין נישאר עם 4, לאחר מכן עם 2, ולבסוף עם 1.
- אם נתחליל עם 6 אפסים, נישאר עם 3 ועוד עם 2 כי לא נמצא 0 למחוק בזוג האחרון של האפסים. ככלומר השאייה שלנו היא להגיע ל-0 אחד בודד (המסומן על ידי L בהתחלה), באשר כל שאר הקלט יהיה מחוק (X).

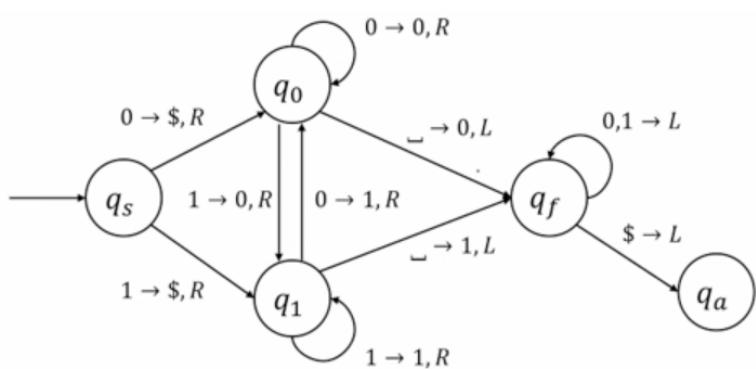
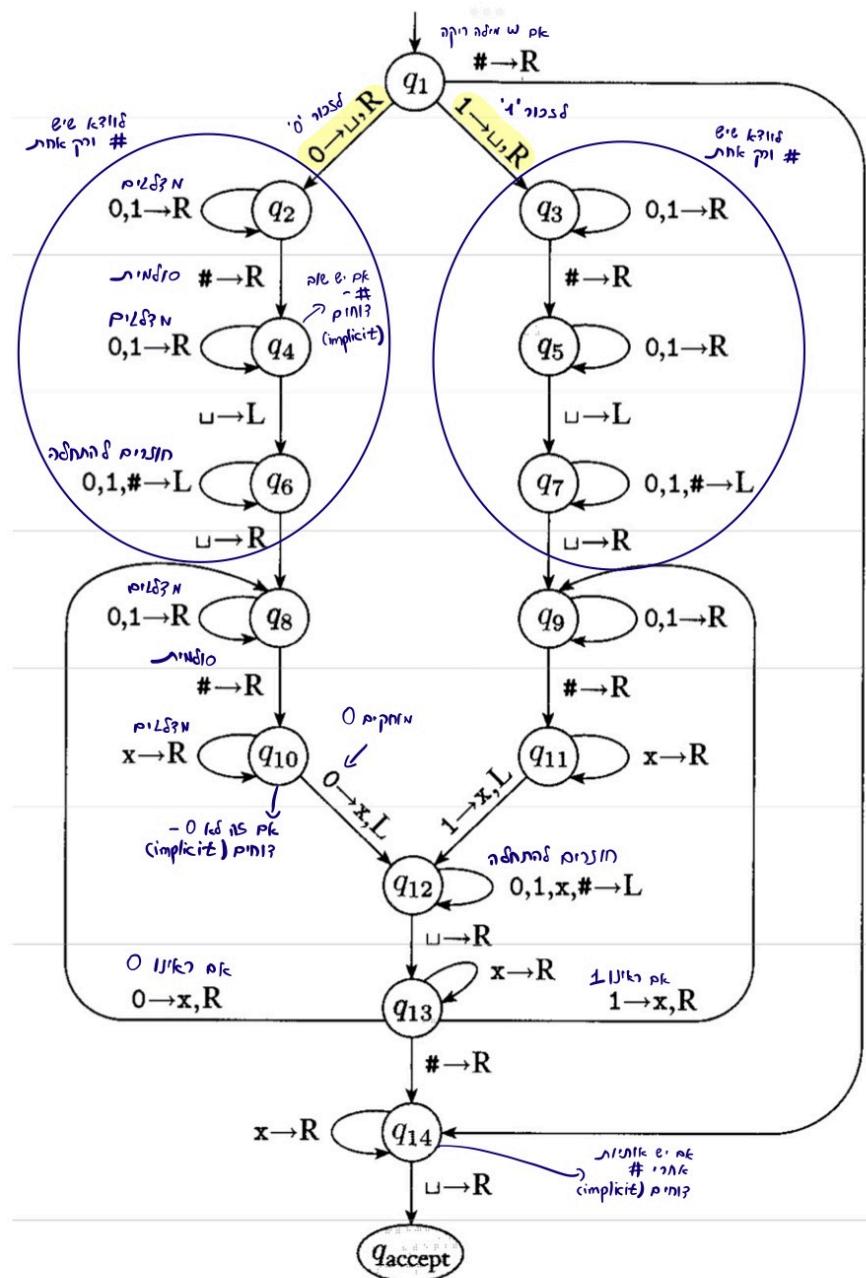
ריצה על $:0^8$:ריצה על $:0^6$:

$q_1 00000000$ $q_2 0000000$ $x q_3 000000$ $x 0 q_4 00000$ $x 0 x q_3 0000$ $x 0 x 0 q_4 000$ $x 0 x 0 x q_3 00$ $x 0 x 0 x 0 q_4 0$ $x 0 x 0 x 0 x q_3$ $x 0 x 0 x 0 q_5 x$ \vdots $q_5 x 0 x 0 x 0 x$ $q_2 x 0 x 0 x 0 x$ $x q_2 0 x 0 x 0 x$ $xx q_3 0 x 0 x 0 x$ $xxx q_3 0 x 0 x 0 x$ $xxx 0 q_4 x 0 x 0 x$ $xxx 0 x q_4 0 x 0 x$ $xxx 0 xx q_3 x 0 x 0 x$ $xxx 0 xxx q_3 x 0 x 0 x$	8 4 2	$xxx 0 xx q_5 x$ \vdots $q_5 xxx 0 xxx$ $q_2 xxx 0 xxxx$ $x q_2 xx 0 xxxx$ $xx q_2 x 0 xxxx$ $xxx q_2 0 xxxx$ $xxxx q_3 xxxx$ $xxxxx q_3 xx$ $xxxxxx q_3 x$ $xxxxxxxx q_3$ $xxxxxx q_5 x$ \vdots $q_5 xxxxxxxx$ $q_2 xxxxxxxx$ \vdots $xxxxxxxx q_2$ $xxxxxxxx q_{acc}$	$q_1 0000000$ $q_2 000000$ $x q_3 00000$ $x 0 q_4 0000$ $x 0 x q_3 00$ $x 0 x 0 q_4 0$ $x 0 x 0 x q_3$ $x 0 x 0 q_5 x$ $x 0 x q_5 0 x$ $x q_5 0 x 0 x$ $q_5 x 0 x 0 x$ $q_5 x 0 x 0 x$	6 3 2
---	-------------	--	---	-------------



דוגמה 2: השפה $\{w\#w : w \in \{0,1\}^*\}$

ראינו את השפה זו קודם, ואת הרעיון לבדוק עבור מילה האם היא בשפה.



טיפול בתחילת הסרטן: נניח $\{0,1\}^* = \Sigma$. בניית מ"ט שדוחפת תו מייחד (\$) בתחילת הסרטן, מזיהה את הקלט ימינה, וחזרה לתחילת הסרטן.

הזכירון שלמו יהיה בדמות המ מצבים השונים. כאשר נראה את התו 0 נעבור למצב q_0 , ובשנരאה 1 נעבור ל- q_1 . כלומר,esanchnu במאובט q_0 אנחנו "זוכרים" שהנחנו צריכים לבחון 0 במקום התו הנוכחית, ולפי ערך התו הנוכחי נעבור למצב q_1 (אם קראנו 1) או נישאר במקום אחר (קראנו 0). לבסוף כאשר נקרא את ט' נכתוב את התו לפי המצב המתאים ונתחיל לנעו שמאללה.

ביצוע פעולה בפועל: נסתכל על השפה $\{k \cdot i \cdot j : i, j \geq 1 \wedge k \geq i \cdot j\}$. הרעיון המרכזי: על כל a שאחננו רואים, נרצה למחוק c בנגד כל b , כיון שלא יוכל לעמוד במה b ראיינו. לאחר מכן נשחרר את כל ה- b - ונבצע זאת שוב עבור a . מתי נדע שסימנו? בשאலות הוא סימנים מחוקים ב巡视ה של הסרטן.



הגדרות שקולות למוכנות טירוניג

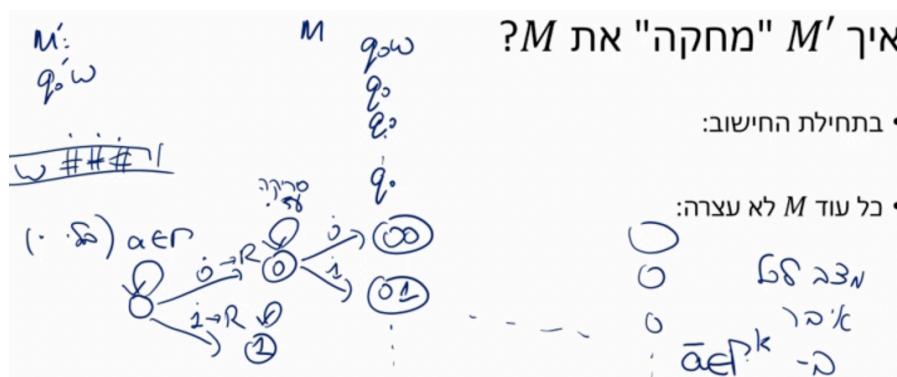
ויראציה 1 – הרוש יוביל להישאר במקום:

- בנוסף לתזוזה ימינה/שמאלה, הרוש יכול להישאר במקום. נגדיר פונקציית מצבים חדשה: $\hat{\delta}: Q \times \Gamma \times \{L, R, S\} \rightarrow Q$
- דוגמה למעבר בו הרוש נשאר במקום: $(q, b, S) = (\hat{q}, b)$.
- שיקולות להגדרה המקורית:
 - לכל $Q \in q$ אנחנו צריכים לעבור, ניצור מצב ביןים $Q \in \hat{q}$ שאליו נבצע תנועה ימינה (תוך כתיבת האות הרצiosa אם יש כזו) ואז שמאלה בחזרה ל-q (לא כתיבה בלבד).
 - לא נעשה R-L כי אם אנחנו בתחלת הסרט לא יוכל לווד שמאלה. משם ממשיר את החישוב שלו ברגיל.
 - כמובן, המעבר החדש שקול לשני מעברים בפונקציית המעברים היישנה:

$$\begin{aligned}\delta(p, a) &= (\hat{q}, b, R) \\ \delta(\hat{q}, \sigma) &= (q, \sigma, L)\end{aligned}$$

ויראציה 2 – מוכנה עם הרובה סרטים:

- למוכנה יש מספר קבוע k של סרטים. לכל סרט יש ראש ונוף.
- בתחלת החישוב: הקלט בתוכו הסרט הראשון, אחר הסרטים מלאים ט.
- המוכנה קוראת בו זמניota א סימנים, וצריך להגד לכל אחד מ-k הראשונים מה לבתוב ואיך לווד. על כן, פונקציית המעברים מוגדרת: $\{L, R\}^k \times \Gamma^k \rightarrow Q \times \Gamma^k: Q \rightarrow \{L, R\}^k$.
- שיקולות להגדרה המקורית:
 - מה שמעוניין אותנו על הסרט הוא תמיד סופי.
 - בסרט בודד יהיו k **תכנים של סרטים**, מופרים באמצעות \$ והוא מצב q: $q u_k ... u_1 \$$.
 - נפצל את פונקציית המעברים ל-k פונקציות מעברים $\delta_1, \dots, \delta_k$.
 - משפט: לכל מ"ט מרובת-סרט M קיימת מ"ט חדת-סרט 'M' כך שמתאים $'M = L(M)$.
- הוכחה:
 - 'M' שומרת את תוכן k הסרטים על הסרט היחיד שלו, מופרדים ב-#.
 - ניקח סימן מיוחד שמייצג את מקום הראש, על ידי הוספת סימן נקודה לכל אות: $\{ \dot{a}: a \in \Gamma \} \cup \Gamma = \Gamma'$.
 - בתחלת החישוב: נאתחל את k הסרטים הפיקטיביים שלו (ריקים), לנשימים רק סולמית, עם נקודה מעל כדי שנזכור שהראש נמצא בהתחלה: # ... #.a.
 - כל עוד M לא עצרה: נסורך כל עוד נראתה $\dot{a} \in a$ (בלי נקודה). ברגע שנראה לדוגמה \dot{a}_1 נדע שריאנו a בראש של הסרט הראש. נניח שהמוכנה הגדולה רצתה לווד R וככתו b, אך ככתו b₁ ונוסף נקודה על התו הבא מיomin. נחפש את התו הבא עם נקודה מעלי, ונטפל בהתחאם. כך נקבל מצב לכל k-יה.
 - מקרה קצה – תזוזה ימינה בסוף אחד מהסרטים (לפני #), נרחיב את גודל הסרט וניאלץ לבצע shift ימינה.



ויראציה 3 – RAM:

זה מודל דומה למוחשב מודרני. הוא מורכב מhardware הבאים: CPU שמבצע פעולות שונות כמו אРИטמטיקה ו קופיצה לבתובות. גיסטרים: IR – הפקודה הנוכחית לביצוע, PC – כתובות בזיכרון של הפקודה הבאה, ACC – "שטח עבודה". זיכרון RAM, ניתן לגשת לכל כתובות. איך המוכנה רצתה?

1. מקדמים את PC באחד.
2. קוראים את [PC]MEM לתוכן IR.
3. מבצעים את הפקודה ב-IR.
4. חוזרים ל-1.



משמעות: לכל M יש מכונת RAM 'M' כך שמתקיים ($'M = L(M)$) ולהיפך.

- נשותמש ב-4 סרטים: סרט לכלי רגיטר (IR, PC) וסרט עבור היזיכרונות: ...<address><contents>...>.
- כאשר נרצה לבצע פקודה, נצטרך קודם את ה-PC, לטען את הפוקודה שבכנתות PC מהיזיכרונות, לבתור אותה ב-IR ולבצע אותה.
- ניתן לסרט שמייצג את ה-PC, בתוכו שם מספר בינארי, נעשה לו $+1$, כלומר את הספרה האחורונה הופכים מ-0 ל-1 והולכים שמאליה עם ה-*carry* וכו'.
- ובשים נרצה לבצע את הפוקודה שבכנתות PC מהיזיכרונות. נרצה לבצע מעין pattern matching אל מול ה-address שמנצט בסרט היזיכרונות, וזהណ שמה שמוופיע אחר כך היא הפוקודה עצמה. נצטרך להפריד Σ עבור התוכן ועבור הכתובות (כינוח לשימוש נקודה מעל כל אות באלפבית).
- בוצע את מה שכתבנו לנו ב-IR.

טיריניג-שלמות (Turing Completeness): מודל חישובי נקרא טיריניג-שלם, אם הוא יכול לחשב כל מה שמכonta טיריניג יכול לחשב. דוגמאות: שפות תכנות כמו C, Python, PostScript (תכנית שמדפסה), משחקים מסויימים. **התזה:** מכונת טיריניג תופסת את כל מה שניתן לחשב באמצעות מודלים חישוביים.

מכונת טיריניג אי-דטרמיניסטיבית (טט"ז)

נזכיר בהגדירה המקורית: מכונת טיריניג מוגדרת $(Q, \Sigma, \Gamma, \delta, q_0, q_r) = M$ כאשר:

- Q היא קבוצת מצבים סופית.
- Σ הוא א"ב קלט סופי שאינו מכיל את הסימן \sqcup (blank).
- Γ הוא א"ב סרט סופי ברשותם: $\Gamma \subseteq \Sigma \cup \{\sqcup\}$.
- δ היא פונקציית מעברים: $\delta: (Q \setminus \{q_a, q_r\}) \times \Gamma \times Q \rightarrow \{L, R\} \times \Gamma \times Q$.
- מצבים מיוחדים: q_0 תחילי, q_a מקבל, q_r דוחה ($q_a \neq q_r$).

בעת, נרצה לאפשר לבונה לבחור בחירה אי-דטרמיניסטיבית מבין כמה אפשרויות. כלומר מה ש- δ תחזיר לנו היא לא אפשרות אחת אלא במא: $(Q \setminus \{q_a, q_r\}) \times \Gamma \rightarrow P(Q \setminus \{q_a, q_r\})$:

עד חישוב: קונפיגורציה C עוברת לkonfiguracija C' אם C מתקבלת מ- C על ידי בתיבת תו על הסרט, הזאת הראש ומ עבר מצב לפי **אחד האפשרויות ב- δ .**

- במ"ט דטרמיניסטיבית, konfiguracija $un = (q, \gamma, D, \delta)$ היא: $C' = (q', \gamma', D')$ עוברת ל- $\sqcup' q' un$ אם: $(q, \gamma, D, \delta) \xrightarrow{un} (q', \gamma', D')$.
- מתקובל מ- un ע"י החלפת המיקום ה-(C) $head(C)$ ב- γ .
- במ"ט אי-דטרמיניסטיבית C עוברת ל- C' אם קיימת אפשרות: $\delta(q, \gamma, D) \in \{L, R\}$ ב- C מתקבלת מ- C על ידי (D, γ, q') לפי ההגדירה הדטרמיניסטיבית.

עד החישוב: חישוב אי-דטרמיניסטי הוא עץ של כל האפשרויות, כל המסלולים שהמ"ט בודקת עם כל האפשרויות שהיא יכולה לעבור בהן לפי δ . זה יהיה העץ של קונפיגורציות:

- כל צומת מסומן בkonfiguracija.
- שורש העץ יהיה konfiguracija תחילית: w_0q_0 .
- הבנים של C הם כל הקונפיגורציות C' ש- C אפשר לעבור אליהן.

אם זהו עץ סופי? ברגע לאוטומט, מ"ט יכול להיכנס לולאה אינסופית, לחזור לתחילת הקלט – לא חיבת לעצור. לכן **העץ הזה לא בהכרח סופי**. יתכנו ענפים אינסופיים. **הרוחב של העץ בכל konfiguracija** הוא סופי: לאחר שלכל konfiguracija יש מספר סופי של konfiguraziotn שהיא עוברת אליהן. אבל העומק של העץ יכול להיות אינסופי: מיצג מסלול חישוב אינסופי.

השפה של מ"ט אי-דטרמיניסטיבית: בהינתן מ"ט אי-דטרמיניסטיבית N וקלט $w \in \Sigma^*$.

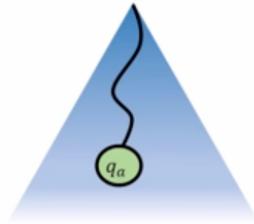
- אם בכל העץ **יש מסלול כלשהו שmagiu לkonfiguracija מקבל** – ואם כן המכונה מקבלת. כמובן, אם קיים עלה מקבל.
- N דוחה אם היא עצרת (העץ סופי) ולא מקבלת את w (אין עליים מקבלים).
- N עצרת (halts) אם היא מקבלת את w (מקבלת), או אם כל המסלולים בעץ החישוב על w הם סופיים (דוחה).
- N לא עצרת** אם יש לפחות מסלול חישוב אחד אינסופי, ואין אף konfiguracija מקבלת.

מ"ט אי-דטרמיניסטיבית מול דטרמיניסטיבית: יש סיטואציות שבהן אי-דטרמיניסטים יכולים להגדיל את הכוח החישובי (כמו באוטומט מחסנית). לעומת זאת, באן אנחנו נראה שהכוח החישובי שקול.

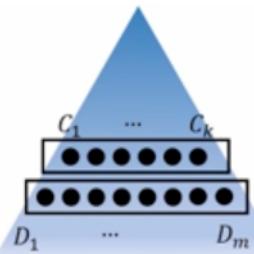


משפט: לכל מ"ט אי-דטרמיניסטי N , קיימת מ"ט דטרמיניסטיבית M שמחקה אותה, כלומר לכל קלט $* \Sigma \in w$:

- M מקבלת את $w \Leftrightarrow N$ מקבלת את w .
- M דוחה את $w \Leftrightarrow N$ דוחה את w .
- M לא עוצרת על $w \Leftrightarrow N$ לא עוצרת על w .



N מקבלת את w אם מסלול חישוב מקבל של N על w . איך נמצא אותו? יש שתי אסטרטגיות לחיפוש בעץ: DFS ו-BFS. האסטרטגיה שלא טוביה לנו היא DFS, אם נתקל במסלול אין סוף נמשיך לעומק ולעולם לא נעצור. לכן, **בחירה-BFS, נתתייל מהקונפיגורציה ההתחלתית (השושט)**, נפתח את כל הבנים ונבדוק אם יש שם קונפיגורציה מקבלת. אם לא, נמשיך עם הבנים שלהם וכן הלאה – חיפוש לרוחב. לשם נוחיות, משתמש בשני סרטים:



סרט ראשון, השבבה הנוכחית בעץ: $\$C_1 \# C_2 \# \dots \# C_k$. אנחנו יודעים שכמויות הקונפיגורציות היא סופית – k בלבד.

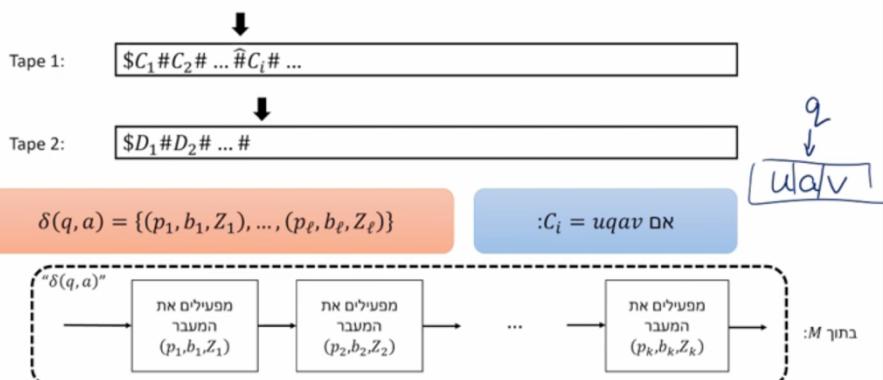
סרט שני, השבבה הבאה בעץ: $\$D_1 \# D_2 \# \dots \# D_m$. זה סרט עבודה, הבנים של קונפיגורציות בשבבה הקודמת. לבסוף, ניקח את הסרט השני ונכתבו אותו על גבי הסרט הראשון. # מפרידה בין קונפיגורציות.

אתחול M : נאותחל את הסרט הראשון להיות $\#q_0w$, הסרט השני יהיה ריק $\$$.

צעד של M :

- על הסרט הראשון נכתב: $\$C_1 \# C_2 \# \dots \# C_k$.
- נעבור על כל אחת מהן C_i . אם זו קו"ן מקבלת, המכונה מקבלת. אם היא דוחה, לא נעשה כלום (אין לה בנים בעץ).
- אם היא לא סופית, ניצור את כל הקונפיגורציות שהיא עוברת אליהן (הבנייה) ונכתב על הסרט השני.
- לבסוף נעתיק את הסרט השני לראשון.

чисוב הקונפ' העוקבות ל- i



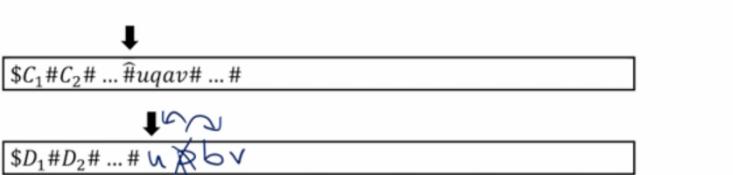
איך נחשב קונפיגורציות עוקבות (בנייה)?

- אם $taq = C_i$ (הראש נמצא מעל a במצב q). אנחנו יודעים את δ , לכן יהיה לנו חלק ב- M שמטטרתו להעתיק קונפיגורציה מהסרט הראשון לשני ולהפעיל את המעבר הראשון, להעתיק ולהפעיל מעבר שני וכו', עד שניצור על הסרט השני את כל הבנים של C_i .
- איך לאחרינו מפרק שלם "ט" רק מספר סופי של מצבים? יש מספר סופי של מעברים אפשריים. אנחנו נתבנתה כל מעבר.

איך מפעילים מעבר ספציפי על קונפיגורציה?

1. הראשים נמצאים מעל סולמיות, כל עוד לא הגיעו לסולמיה בסרט הראשון, מעתיקים אותן ווזים עם שני הראשים ימינה. سورקים חוזרת עד הסולמיה הקודמת.
2. הסרט השני, נסורך חוזרת להתחלה (סולמיה). سورקים ימינה עד שמצאים את $M-Q$, זה המצב.
3. בותבים במקום q את q , זו ימינה ובותב במקום a את b , ואז מבצעים את ההחלפה.

את כל השלבים האלה אפשר למשם עם מספר סופי של מצבים. זה ריבוע בודד בתרשים הקודם. לכן, כיוון שיש מספר סופי של ריבועים, לכל המכונה שלנו M יש מספר סופי של מצבים. מתי M עצרת:



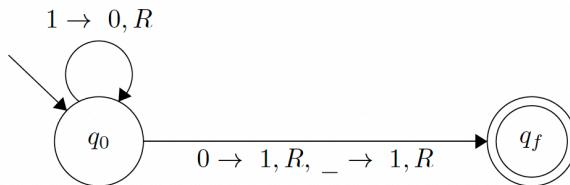
- מקבלת: אם מצאנו קונפיגורציה מקבלת.
- דוחה: אם כל המסלולים סופיים, יתרחקן לנו הסרט השני בסיום כתיבת השבבה הבאה (לא יהיו בנים), וזה נעצור.



תרגול 5 (מבנה טירונג)

תרגיל 1: בנו מ"ט עבור הפונקציה $f(x) = x + 1$ המקיימת קלט x ביצוג בינארי (LSB בתחילת הסרט), ומסימת באשר (x) רשום על הסרט ולאחריו \underline{L} . במקרה זה נגד ש-M מחשבת את f .

$$M = (\{q_0, q_f\}, \{0,1\}, \{0,1, \underline{L}\}, \delta, q_0, q_f)$$



- כל עוד אנחנו רואים 1 נהפוך ל-0 ונמשיך לנوع ימינה. ברגע שנראה 0 (או \underline{L}) נהפוך אותו ל-1 ונקבל.
- עבור הקלט 0 בלבד, נכתב במקומו 1. אז גם כאשר יש 1 לפניו ואז ראים 0 זה יעבד, וגם עבור 0 בלבד.

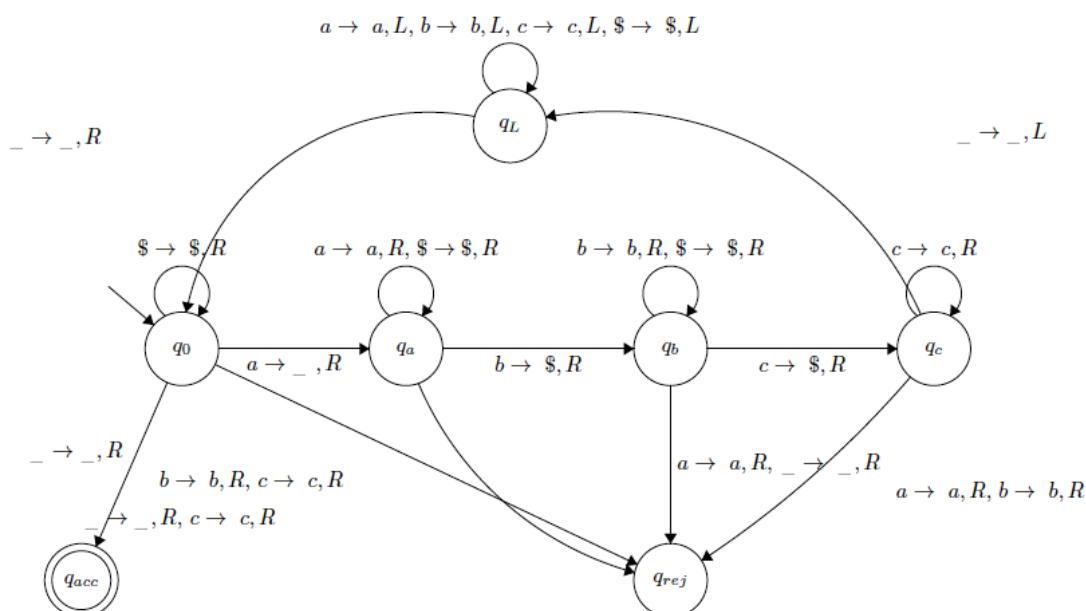
תרגיל 2: בנו מ"ט המכירעה את השפה $0^n c^n b^n a = L$. (בתרגול 5 של גל השנה – תשפ"ד – יש מ"ט עבור $a^n b^n c^n$).

נשתמש בתו מיוחד $\$$ כדי לצייןתו מחוקק, פרט עבור ה- a שבתחלת הקלט אותו נסמן ב- \underline{L} . הרעיון הוא לעבור שוב ושוב אל הקלט מההתחלת ועד הסוף, ובכל פעם למחוקק c, b, a בודדים. קיבל אם"מ נוכל לעשות זאת ולסימן עם מחרוזת שכולה $\$$.

1. (המקרה שאחננו שוואפים אלו)
 - כל עוד התו הנוכחי הוא \underline{L} , המשך ימינה (אלו תווים מחוקקים של c/b).
 - אם התו הנוכחי הוא \underline{L} , קבל. אם התו הנוכחי הוא c, b , דחה.
2. (מחיקת התווים)
 - החלף את a עם \underline{L} וודז ימינה.
 - כל עוד התו הנוכחי הוא $\$/a$ זוד ימינה. אם ראיית $\underline{L}/c -$ דחה, אם ראיית $b -$ החלף עם $\$$ וודז ימינה.
 - כל עוד התו הנוכחי הוא $\$/b$ זוד ימינה. אם ראיית $\underline{L}/a -$ דחה, אם ראיית $c/a -$ החלף עם $\$$ וודז ימינה.
 - כל עוד התו הנוכחי הוא c זוד ימינה. אם ראיית $b/a -$ דחה. אם ראיית \underline{L} חוזר שמאליה עד ל- \underline{L} הראשון שתראה. זוד ימינה חוזר לסעיף 1.

נשים לב כי:

- עברו מילה בשפה, תוכן סרט העבודה הוא תמיד מהצורה $a^j \$^i b^j \$^i c^j$.
- ניתן להגיע למצב מתקבל אם כל ה- a -ים שנמחקו (ע"י \underline{L}) שווה למספר ה- b -ים שנמחקו (ע"י $\$$), שווה למספר ה- c -ים שנמחקו (ע"י $\$$).
- לא ניתן לקבל מילים שאין בשפה.
- כל מילה שאינה בשפה מובילה למצב דוחה.





מחלקות חישוביות

מונחים (enumerators)

מונה (enumerator): מ"ט M היא מונה עבור L אם M מייצרת את כל המילים ב- L (ורק אותן) בזו אחר זו, ככלומר מונה אותן. באופן יותר פורמלי נגיד כי M מונה עבור $\Sigma \subseteq L$ אם:

- $L \in M$ יש סרט פלאט.
- A^B סרט הפלט הוא $\{\$ \}^A \Sigma = \Gamma$ (נכיה כי $\Sigma \neq \$$).
- סרט הפלט הוא "write-only": M עלולם לא משנה תא שביבר בתבה אליו.
- בשרמיצים את M על קלט ריק הוא $\$ w_2 w_1$ כאשר:

 - **כל מילה ב- L מופיעה על סרט הפלט** לפחות פעם אחת.
 - **הדרישה המשלימה: מילים שאין ב- L אינם לא מופיעות על סרט הפלט.**

משפט: $RE \in L \Leftrightarrow L \in RE$.

הכוון הטוריוני: אם $L \in M$ יש מונה, אז $RE \in L$.

- נכיה ש- N הוא מונה עבור L . בבנה מ"ט M שמריצה את N . אם מוצאים את הקלט w על גבי סרט הפלט של N , אז עוצרים ומקבלים (אחרת נרים את N أولי לנוכח).
- אויר M "מריצה את N ?" נחשוב על N בתור מ"ט נטענה, נלביש על גבי זה עוד מצבים. למשל, כל פעם ש- N כתבה $\$$ (סימנה לבתוב מילה), נעבור למצב שבודק האם הקלט w מופיע בסרט.
- אויר נבדוק שהקלט מופיע בסרט? ניעזר בסרטן נוסף של גבי כתובה w , נחזיר הראש של סרט הפלט ל-\$-האחרון, ונשורך ביחס עם הראשים כל עוד רואים את אותה אות. אם סימנו לקרוא את הקלט וכותב $\$$ על הפלט סימנו (יש התאמה) אחרת, נמשיך הלאה.

הכוון השני: אם $RE \in L$ אז $L \in M$ מונה.

- נכיה ש- $(M)L = L$, כאשר M מ"ט דטרמיניסטי. נקבע מכיה של Σ (כל המילים האפשריות) לפי סדר לקסיקוגרפי.
- "הינו רצים" לבצע לכל $i = 1, 2, 3, \dots$: להרים את M על s_i . אם M מקבלת: נכתבו את s_i בסרט הפלט. אבל, אם M לא עוצרת על s_i ו- $L \in s_i \Rightarrow s_i$ לא נפלוט את j !
- אויר נתכן את זה? באמצעות **dovetailing** (במקור סגן של חיבור בngeroot). לא נרצה להשקייע אינסוף זמן באף מילה נטענה, לא נרדוף לנץ אחרי אף מילה. **נקזה בהדרגה יותר זמן לבדיקת כל מילה.**
- עברו ... $i = 1, 2, 3, \dots$
- נרים את M במשך ? צעדי חישוב על כל המילים s_i, \dots, s_1 (מספר סופי של מילים, אחרת גם לבדוק עד אינסוף).
- אם M מקבלת את s_i תוך ? צעדים: בוטבים את s_i על סרט הפלט.
- הערכה: ברגע שקיבלו מילה, נקבל אותה שוב אינסופי פעמים (אם קיבלנו תוך צעד אחד נקבל גם תוך שניות...).
- **מתי מילה מסוימת $L \in \Sigma$ נכתבת על סרט הפלט?** אם $s_i = w$ ו- w מתקבלת על ידי M תוך ? צעדים, נכתבו את w באיטרציה שהיא $\max\{i, k\}$. לא ניתן להבטיח מראש מילה נתונה תיכתב על סרט הפלט, כמו כן סדר המילים.

משפט: $R \in L \Leftrightarrow L \in M$ מונה שמייצר את המילים ב- L לפי סדר לקסיקוגרפי.

הכוון הטוריוני: אם $L \in M$ יש מונה לקסיקוגרפי, אז $R \in L$. נכיה ש- N הוא מונה לקסיקוגרפי עבור L . על קלט w נרים את N עד ש:

- w מופיעה על סרט הפלט: מקבלים.
- מופיעה מילה 'w' שהיא גדולה מ-w בסדר לקסיקוגרפי: דוחים.
- אם המונה עצה (בmarker של שפה סופית): דוחים.

הכוון השני: אם $R \in L$ אז $L \in M$ מונה לקסיקוגרפי. בהינתן M מ"ט דטרמיניסטי שمبرיעה את L בבנה את המונה הבא:
לכל $i = 1, 2, 3, \dots$ נרים את M על s_i עד שהיא עצה. אם M קיבלה: בעתק את s_i לפט. אחרת: נמשיך.

מחלקות חישוביות:

L ∈ RE: אם קיימת מ"ט M כך ש- $L = L(M)$, מקבלת אך לא בהכרח עוצרת.

- אוסף השפות בהן אנו יודעים להגיד כן שצריך.

למשל, **משפטים שאפשר להוכיח**. בהינתן משפט, אפשר להתחיל לעבור על כל ההוכחות האפשריות, ובכל הוכחה נבדוק אם היא הוכחה של המשפט הנתון – נעצור ונקבל. אם המשפט לא ניתן להוכחה, לא בהכרח נגלה את זה, כי אנחנו בודקים רק מה כן אפשר.

אם $\bar{L} \in coRE$:

-

▫ אוסף השפות שהמשילמה של השפה היא ב- \bar{RE} . אלה שפות בהן אנו יודעים להגיד לא בשצריך.

למשל, **משפטים שיש להם דוגמה נגדית**. אם הטענה אכן לא נכונה, נמצא, במקרה בסופו של דבר דוגמה נגדית. אם הטענה נכונה, לעומת זאת, לא ניתן להפריך אותה ולא נמצא דוגמה נגדית, לעומת זאת לא נגיד כן.

אם קיימת מ"ט M שתמיד עוצרת לכל קלט כך ש- $L = L(M)$:

-

решפט: $R = RE \cap coRE$

הכיוון הראשון: $R \subseteq RE \cap coRE$

אם יש מ"ט M שמכריעה את L אז $L = L(M)$ וכן $L \in RE$.

אם ניקח את M' שהוא M שבזה החלפנו את המ מצבים q_r, q_a אז $\bar{L} = L(M')$ וכן $\bar{L} \in coRE$.

באופן כללי, אם אנחנו יודעים תמיד מה התשובה, אין לנו בעיה להפוך את התשובה.

הכיוון השני: $RE \cap coRE \subseteq R$

תהי $M_0 \in RE \cap coRE$. לכן קיימות מ"ט M_0, M_1 כך ש- $L(M_0) = \bar{L}, L(M_1) = L$.

בננה מ"ט M באופן הבא: מרכיבים צעד אחד אחד של M_0 על הקלט, ואז צעד אחד אחד של M_1 לסירוגין.

▫ אפשר למשם באמצעות שני סרטים. המבונה הראשונה תעבור על הסרט הראשון, והשנייה על השני. יהיו לנו שני עותקים, ונחנכו מרכיבים אותן במקביל על שני הסרטים. ברגע שאחת מקבלת נחילט לפיה המבונה קיבלת.

אם M_0 עצרה וקיבלה – נדחה (כי זו מכונה שמצויה את המ מצבים של השפה).

אם M_1 עצרה וקיבלה – נתקבל.

נשים לב כי מובטח לנו שלפחות אחת מהמבנהות תעצור, אך לא יכול להיות שתיהן תעוצר. המילה שלנו או בשפה (המבנה M_0 לא יכולה לקבל אותה) או לא בשפה (המבנה M_1 לא יכולה לקבל אותה).

מבנה טיריניג אוניברסלי

מ"ט אוניברסלי: מקבלת קלט מ"ט M (בקייםו בלבד) ומילת קלט, ומריצה את M על הקלט.

קידוד סטנדרטי למ"ט $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r \rangle$

▫ $Q = \{q_1, \dots, q_m\}$ כאשר q_0, q_a, q_r הם q_1, \dots, q_m בהתאם.

▫ $\Sigma = \{\sigma_1, \dots, \sigma_k\}$

▫ $\Gamma_s = \{\gamma_1, \dots, \gamma_k\}$ כאשר $\gamma_1 = \sigma_1, \dots, \gamma_k = \sigma_k$ (בולם k אותיות הסרט הראשונות הן אותן האותיות הקלט).

▫ ניצג את $\{\Gamma, \Sigma\}$ ע"י $\{1, 2\}$ ע"י $\{L, R\}$ בהתאם.

▫ מספיק לקודד רק את δ !

דוגמאות:

למשל, את המעבר $(b, \gamma_s, \delta, q_i, q_j) = (q_t, \gamma, \delta, q_i, q_j)$ נציג ע"י $0^i 10^j 10^t 10^s 10^b$.

את δ כולה נציג ע"י רשימת מעברים, מופרדים ב-11.

כדי לקודד זוג (w, M) כאשר M מ"ט ו- w קלט: נפריד ע"י 111.

решפט: קיימת מ"ט U שמקבלת קידוד $* \in \{0, 1\}^M$, מודדת שזהו קידוד חוקי של מ"ט + קלט, ומחקה את (w, M) .

▫ אם M מקבלת את w אז U מגיעה למצב מקבל.

▫ אם M דוחה את w אז U מגיעה למצב דוחה.

▫ אם M לא עוצרת על w , אז U לא עוצרת.

איך לבצע את הסימולציה הזו? U מחזיקה סרטן שלה את הקונפיגורציה הנוכחיית של M , ומפעילה צעד חישוב לפי הקידוד של δ . כדי למצוא את הצעד הרלוונטי, נזהה את המצב הנוכחי וביצע pattern matching עם המעברים האפשריים ואת הסרטן, כדי למצוא אותו.



ל-U יש מספר קבוע של מצבים (ngeid 100), אך היא יכולה לבצעות עם הרבה יותר מצבים ממנה. באופן דומה לכך שkompilear (בהתור תכנית עם אורך מסוים), יכול לкомפלט תוכניות מאורך גדול כרצוננו. טירינג דיזה בבר במאמר שלו שיש בכך ממשמעות. הוא כתוב כי החשיבות של מ"ט האוניברסלית היא ברורה: אין צורך באינסוף מכונות שככל אחת מהן מבצעת משימה אחרת. בעיתם הנדסה של לייצר מכונות שונות למטרות שונות, נחלף אותה "בעבודה משדרית" של תכנות המכונה האוניברסלית המסוגלת לעשות כל דבר שמחשב יכול לעשות.

אי-כריעות

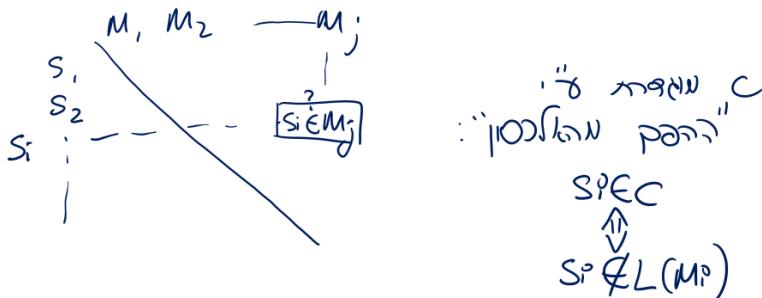
בעיה שאינה ב-RE:

בעיה כך שלא רק שלא ניתן להכريع אותה, לא ניתן להזהה מילים בשפה שעלהן צריך להגיד כן. נגידו:

- סידור לקסיקוגרפי של כל המחרוזות ב- Σ^* : s_1, s_2, \dots . כל מילה תופיע באיזשהו מספר סופי במניה.
- סידור לקסיקוגרפי של כל המ"ט מעל ס (לפי הקידוד שלו): M_1, M_2, \dots, M . כל מ"ט תופיע במספר סופי במניה.

טענה: השפה $\{s_i : M_i(s_i) \neq C\}$, אין אף מ"ט שהוא שלה.

הוכחה: נניח בשליליה שמתקיים $C \in RE$, וכן קיימת מ"ט שמצויה אותה, היא מופיעה במניה במקום כלשהו i : $M_i = C$. ניתן למוכנה i את s_i . מתקיים $s_i \in L(M_i) \Leftrightarrow s_i \notin C \Leftrightarrow s_i \notin L(M_i)$. לכן לא קיימת מ"ט בזו, ולכן השפה היא לא ב-RE.



- אם $s_i \in C$ נובע כי M_i לא מקבלת אותה, בסתיוher.
- לבסוף M_i מקבלת מילים בשפה C .
- אם $s_i \notin C$ נובע כי M_i מקבלת אותה, בסתיוher לבסוף M_i לא מקבלת מילים שאין בשפה C .

הוכחה דומה ללבסן:

- עצמת קבוצת כל השפות מעל ס היא 2^{\aleph_0} .
- $A = |2^{\{0,1\}^*}| = |\{0,1\}^*| = |\{L: L \subseteq \{0,1\}^*\}|$
- עצמת קבוצת כל מ"ט מעל ס היא \aleph_0 .
- $|L: L \in RE| \leq |\{M \text{ is a TM}\}| \leq |\{< M > \in \{0,1\}^* \text{ MT coding}\}| \leq \aleph_0$

בעיה ב-RE שאין כריעות (לא ב-R):

- שפת הקבלה: $L_{ACC} = \{(M, w) : M \text{ is a TM that accepts } w\}$ – השפה הזאת ב-RE כי המכונה שמקבלת אותה היא מ"ט האוניברסלית.
- שפת העצירה: $L_{HALT} = \{(M, w) : M \text{ is a TM that halts on } w\}$; גם כאן אפשר להריץ את המכונה על הקלט בעדרות מ"ט האוניברסלית, אם היא עצירה אנחנו מקבלים (לא משנה אם היא קיבלה או דחתה).
- שפת כל המשפטים שאפשר להוכיח: אפשר לעבור על כל ההוכחות ופושט לחפש הוכחה.

בעיית הקבלה (ACC): $R \in RE, L_{ACC} \notin R$. הקושי נובע מכך שאם המכונה שלנו M לא מקבלת, אנחנו לא יודעים שהיא לא קיבל, אין שום נקודה שבה אנחנו יודעים שהוא הולכת לדוחה אם היא לא עשתה זאת עד כה.

$L_{ACC} \in RE$ כי $L(U) = L$ כאשר U היא מ"ט האוניברסלית.

$L_{ACC} \notin R$: נניח בשליליה כי $R \in L_{ACC}$ בולם יש מ"ט A שمبرיעעה את L_{ACC} .

- A תמיד עצרת, היא מקבלת בשנותנים לה M שמקבלת את הקלט שלה, והיא דוחה כאשר M דוחה.
- כלומר נגדיר: $A(M, w) = \begin{cases} \text{accept, } M \text{ is a TM that accepts } w \\ \text{reject, otherwise} \end{cases}$
- נגדיר מ"ט Flip – הופכת את מה ש-A עשו. על קלט M (קידוד חוקי של מ"ט): נרץ את $(A)(M, M)$ ונשאל האם המכונה M מקבלת את הקידוד של עצמה? **לפי התשובה של A נעשה ההיפך**:
- אם A דוחה – מקבל.
- אם A מקבל – דוחה.
- הסתירה: מה קורא כאשר מרים את Flip על הקידוד של עצמה? **שואלים את A האם Flip מקבלת את ?Flip**?
- אם A אומרת כן (Flip מקבלת את Flip) – אז Flip מחזירה ההיפך ודוחה את flip.
- אם A אומרת לא (Flip דוחה את Flip) – אז Flip מחזירה ההיפך ומתקבלת את flip.

אם יש בעיה בהוכחה שהרכינו תוכנית על עצמה? לא! זה כמו להריץ קומPILEAR על עצמו ☺



ב夷ית העצירה (HALT): נוכיח כי $R \in RE$, $L_{HALT} \in RE$, $L_{HALT} \notin RE$. כלומר יש מ"ט A שمبرיעה את L_{HALT} .

- נגיד מ"ט 'Flip' – אנחנו שואלים האם M עוצרת על w, ולכן בעשה ההפך מבחינת עצירה. על קלט M: נרים את (M, M)) ונסאל האם המוכנה M עוצרת על הקידוד של עצמה? **לפי התשובה של A עשויה ההפך:**
 - אם A דוחה (לא עוצרת) – נעצור (נקבל או נדחה, לא משנה).
 - אם A מקבלת (עוצרת) – ניכנס לולאה אינסופית.
- הסתירה: מה קורה באשר מרים את Flip על הקידוד של עצמה $\langle \text{Flip} \rangle$? שואלים את A האם Flip עוצרת על Flip?
 - אם A אומרת בן (Flip עוצרת על flip) – אז Flip מחרירה ההפך ולא עוצרת.
 - אם A אומרת לא (flip לא עוצרת על flip) – אז Flip מחרירה ההפך ועוצרת.

חישוב יוניפורמי מול לא יוניפורמי האם יש משפחת מעגלים שהשפה שהיא מזזה היא L_{HALT} ? ניתן למוגל בקלט קידוד ביביאורי של מ"ט ושל קלט, ומוגל צריך להוציא 1 אם זה עוצר על הקלט ו-0 אחרת.

- **התשובה היא כן** – כי הוכחנו שלכל שפה יש משפחת מעגלים שמזזה אותה. אנחנו בונים מוגל אחר לכל גודל קלט.
- **נכיה שמעניינות אותן רק מ"ט** וקלטים שהקידוד שלהם הוא באורך 50, אז יש מספר סופי של קידודים באורך 50 והוא²⁵ – כל אחד מהקידודים יש עליו תשובה כן/לא – לכן בונה מוגל שמזזה רק את הקלטים באורך 50 שהתשובה עליהם היא כן – המוגל הזה קיים.

עד בעיות לא בריאות:

- האם CFG נתון מייצר את כל המילים ב- Σ^* ? מספר הפעמים שאנו שפט מפעלים כללים הוא לא חסום. נכיה שבדקנו את כל הניסיונות לגזר מילה מסוימת עם מיליון כללים ולא מצאנו דרך. זה לא אומר שאין דרך לגזר אותה. לא נדע מתי לעצור ולהגיד לא.
- בהינתן אוסף סופי של מטריצות A מעל השלים – האם ניתן לכפול אותן בסדר כלשהו (עם חזנות) ולקבל את מטריצת האפס? אפשר להסתכל על כפלים של המטריצות כולל חזנות, אין חסם על הסדרים האפשריים.
- האם ניתן לרוץ את המשיר עם אוסף נתון של מרצפות Wang? (מוחתר להניח רק אם הצבע מתאים) – אין אוסף רציפים.
- הבעיה העשירה של היילברט: האם למשווה מהצורה $0 = (x_1, \dots, x_n)P$ יש פתרון בשלמים, באשר $(x_1, \dots, x_n)P$ פולינום עם מקדמים שלמים?
- האם קיים מסלול טישה מיעד A ליעד B במחיר C, תחת כללי תמחור אמיתיים של חברות תעופה? מרוב שכלי חברות התעופה מסובבים, לטיסות הלוך-חזור לעומת טיסות חד-כוויניות.

תרגול 6 (מחלקות חישוביות)

מחלקות R ו-RE:

נזכר בהגדרות הבסיסיות:

- מ"ט M **מקבלת** (accepts) שפה L אם על קלט w: אם $L \in w$ היא מקבלת, ואם $L \notin w$ היא דוחה/לא עוצרת.
- מ"ט M **մכרצה** (decides) שפה L אם על קלט w: אם $L \in w$ היא מקבלת, ואם $L \notin w$ היא דוחה. **תמיד עוצרת!**

מחלקות חישוביות שראינו:

- $L \in RE$ – קיימת מ"ט M **המקבלת את L** (יכולת גם לא לעוצר): דברים שקל לנו למצואו.
- $L \in coRE$ – קיימת מ"ט M **המקבלת את \bar{L}** (יכולת גם לא לעוצר): דברים שקל לנו לפוטול.
- בולם $RE \subseteq \bar{L}$.
- נשים לב לא להתבלבל, קבוצה זו אינה \overline{RE} ! (המשלים של RE).
- $L \in R$ – קיימת מ"ט M **המכריעה את L** (עוצרת תמיד).
- הוכחנו כי $coRE \cap R = RE$.
- המחלקה R סגורה תחת משלים, איחוד וחיתוך.

תרגיל 1: הוכחו כי RE סגורה תחת איחוד.

יהיו $L_1, L_2 \in RE$. צריך להוכיח כי $L_1 \cup L_2 \in RE$. מהגדרת RE קיימות M_1, M_2 כך ש- $L_i = L(M_i)$ עבור $i \in \{1, 2\}$. אם מילה נמצאת בשפת האיחוד, היא שייכת לפחות מהשפות L_i . נרים על המוכנה הראשונה, נראה אם הצלח, אם לא ננסה על השנייה. הבעיה – המוכנה לא חייבת לעוצר. לכן, נרים במקביל על שתי המוכנות. נגדיר מכונה M, שעבור קלט w:

1. מסמלצת את M_2 על w באופן הבא: בכל שלב מבצעים צעד אחד בכל אחת מהמכונות.
2. אם אחת מהן קיבלה – נקבל.
3. אם שתיהן דחו – נדחה.



נכונות:

- אם $w \in L_1 \cup L_2 \in \omega$ אז אחת מההרצות תעוצר ותקבל, ומכאן ש- M מקבל ולכן $L(M) \in \omega$.
- אם $w \notin L_1 \cup L_2 \in \omega$ אז אף אחת מההרצות לא תעוצר ותקבל, ולכן M תדחה או לא תעוצר לעולם (זה תקין כיון שאנו חסרים ששותת האיחוד היא ב- RE) ולכן $L(M) \notin \omega$.

תרגיל 2: נגידור את השפה $\{\} = \{M : M \text{ is a TM s.t. } L(M) = \emptyset\} \in \text{coRE}$.

למה זה לא ב- RE ? אם רצחה להריץ על קלטיים שונים באופן מבוקר, לא יוכל להשתכנע שאין מילה בעתיד שהיא עלולה לקבל. אם יש מילה שהיא קיבלה אנחנו יודעים שזה לא נכון (אינטרואיציה לכך שהיא ב- coRE). אך נסתכל על השפה המשלימה: $\{\} \neq \{M : M \text{ is a TM s.t. } L(M) \neq \emptyset\}$. זה אנחנו יודעים לעשות ב- RE , ונסה מספיק מילים בקלט ל- M עד שהיא תקבל מילה כלשהי ועוד דעת שהמוכנה M בשפה. לכן, נוכיח כי $\overline{L_{\text{EMPTY}}} \in \text{RE}$:

פתרון א (בעזרת מ"ט): בבנה מ"ט A שמקבלת את $\overline{L_{\text{EMPTY}}}$, כך שעבור קלט $\{M\}$:

1. לכל $i = 1, 2, \dots$
2. לכל $j = 1, 2, \dots$
3. הרץ את M על x^j (המילה ה- j -בסדר הלקסיקוגרפי) למשך i צעדים.
4. אם M מקבלת – נקבע.

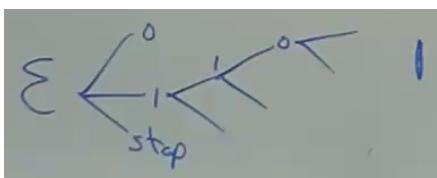
1	0
2	1
3	00
4	01
5	10
6	11

נכונות:

- אם $\{M\} \in \omega$ אז קיימים j, i , כך ש- M מקבלת את x^j לאחר i צעדים, ומכאן ש- A מקבל ולכן $L(A) \in \omega$.
- אם $\{M\} \notin \omega$ אז ככל j , M תדחה את x^j (לא שמנה לאחר כמה צעדים), ומכאן ש- A לא מקבל ולכן $L(A) \notin \omega$.

פתרון ב (בעזרת מטל"ד): בבנה מטל"ד A שמקבלת את $\overline{L_{\text{EMPTY}}}$, כך שעבור קלט $\{M\}$:

1. נחש מילה Σ^* $\in \omega$ (בכל רגע נחליט אם מפסיקים או כתובים 0 או 1 ונגיע לכל מילה אפשרית).
2. הרץ את M על ω .
3. אם M קיבלה – נקבע. אם M דחתה – נדחה.



נכונות:

- אם $\{M\} \in \omega$ אז קיימת מילה w ש- M מקבלת, ומכאן ש- A מקבל (w ניחוש w שעבורו A מקבל) ולכן $L(A) \in \omega$.
- אם $\{M\} \notin \omega$ אז רצף ריצה של A על M שמסתיימת ב- q_{acc} .
- כל ניחוש של w יוביל לדחיה/אי-עכירה (או ריצה של A על M שמסתיימת ב- q_{acc}).

מונה (Enumerators):

מונה (enumerator) עברור שפה L הוא פונקציה $f_L : L \rightarrow \mathbb{N}$: f_L שהוא על, וחשיבה (יש מ"ט שבהינתן $N \in \omega$ תפלות $L \in \omega$).

תרגיל 3: מונה מונוטוני עברור שפה L הוא מונה כך שאם $j < i$ אז $f_L(j) < f_L(i)$. בולם, הוא מונה את המילים בשפה בסדר לקסיקוגרפי. הוכחו: לכל שפה אינסופית $L, R \in \omega$ $\Rightarrow L \in \omega \Leftrightarrow R \in \omega$ \Leftrightarrow מונה מונוטוני.

\Leftarrow : נניח כי $R \in \omega$, שכן קיימת מ"ט M המכrica את L . בבנה מונה מונוטוני f_R : על קלט x , נרץ את M על המילים בסדר L ונקリスト את המילה ה- n -ית שהתקבלה. כיון ש- M מכrica את L היא בפרט **עצרת על כל קלט**. f_R חישיבה, מחזירה את כל המילים ב- L ומונוטונית.

\Rightarrow : נניח כי L שפה אינסופית ו- f_L הוא מונה מונוטוני עבורה. בבנה מ"ט M המכrica את L . על קלט x :

1. M מחשב את $f_L(1), f_L(2), \dots$.
2. אם היא הגיעו ל- x כך ש- $f_L(i) = x$ – M מקבל.
3. אם היא הגיעו ל- x וכך ש- $f_L(i) > x$ – M תדחה.

נכונות:

- אם $L \in \omega$ אז קיימים i, j כך $f_L(i) = x$ וכל $i < j$ מתקיים $x < f_L(j)$ ולכן M מקבל.
- אם $L \notin \omega$ אז לפחות מתקיים $x \neq f_L(i)$ ולכן M לא מקבל. מכיוון ש- L אינסופית, קיימים $x > y$ ו- $f_L(y) > f_L(x)$. לכן בשניהם לא- x , M תדחה.



תרגיל 4: הוכיחו כי לכל שפה אינסופית $RE \in L$, קיימת שפה אינסופית $L' \subseteq L$ כך ש- $R \in L'$.

$R \in L$ שכן קיים מונה f_L עבור L , מונה זה אינו מונוטוני.

אינטואיציה: נגדיר (1) $g(1) = f_L(k) = w_1$. קיימים k כך $w_1 > f_L(k)$: כיון ש- L אינסופית, בmoות המילים שקטנות או שווה לה היא סופית. אם נריץ את המונה על מספיק מספרים, בסוף נמצא מילה כלשהי שגדולה מ- w_1 . אז נקבל $f_L(k) = g(2)$.

באופן פורמלי: נסתכל על הסדרה ... $(2), f_L(2), f_L(1)$ ונגדיר תת-סדרה מונוטונית ... $(1), f_L(i_1), f_L(i_2), \dots$. נגידר למ' מונה מונוטוני חדש $i_{j+1} = i_j + 1$ והוא המספר הקטן ביותר שגדל מ- i_j כך שמתקיים $f_L(i_j) > f_L(i_{j+1})$. נגידר למ' מונה מונוטוני חדש $(j), g(j) = f_L(i_j)$.

נ קיבל $\{A \in L | g(j) = A\}$. לפי הבנייה שלנו, $L \subseteq L'$, היא אינסופית ו- g הוא מונה מונוטוני עבורה. g חשיבה שכן אנחנו צריכים לחשב מספר סופי של מילים כדי להגיע ל- (j) . מהתרגיל הקודם, $L' \in R$.

תרגיל 5: הוכיחו כי לכל שפה אינסופית L , קיימת $L \subseteq L'$ שאינה- RE .

L אינסופית (בmoות המילים בה בת-מניה), ולכן $\{L' | L' \subseteq L\} = 2^{|L|} = 2^{\aleph_0} = |\{M \text{ is a TM}\}|$. אולם $|\Sigma^*| \leq |\{M \text{ is a TM}\}|$. לכן זו מתקיים משיקולי ספירה.



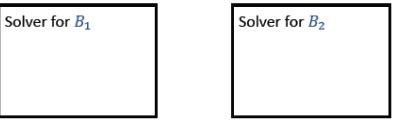
דоказיות

דоказיות

בעית העצירה (HALT): נחזר לבועית העצירה ונרצה להוכיח כי $R \notin L_{HALT}$ באמצעות טכnika אחרת בשם **דоказיה**. הרעיון הוא להשתמש בעובדה שאנו יודעים כי $R \notin L_{ACC}$, ונאמר שאם הינו יכולים להוכיח את בעית העצירה L_{HALT} הינו יכולים גם להוכיח את בעית הקבלה L_{ACC} – אבל אנו יודעים שבעה זו אינה ברעה. **נניח בשלילה** שיש מ"ט H שمبرיעה את בעית העצירה L_{HALT} . בונה מ"ט A שمبرיעה את L_{ACC} בינהן קלט $\langle M, x \rangle$:

- מריצה את $\langle x, H(M) \rangle$.
- אם H דוחה – M לא עוצרת ולכן נדחה.
- אם H מקבלת – M עוצרת, לכן נרץ את $\langle x, M \rangle$ עד לעצירה. נקבע $\Leftrightarrow M$ מקבלת.

קיבלנו סתייה לכך ש- L_{ACC} אינה ברעה, ולכן גם L_{HALT} אינה ברעה. הראנו **דоказיה מ- L_{ACC} ל- L_{HALT}** .

Solver for A 	<p>דоказיה: קיימת דоказיה מבועית הברעה A לבועית הברעה B, אם ניתן להשתמש בכל מכريع עבור B על מנת להוכיח את A.</p> <p>נסמן $B \leq A$, כלומר להוכיח את A לא יותר קשה מלהוכיח את B – אם מצליחים להוכיח את B בפרט נצליח להוכיח גם את A.</p> <p>שימוש: נתחיל מ-A שהוא בוטה, ונראה דоказיה מ-A ל-B, ונסיק ש-B בוטה.</p> <p>ברעה. אפשר לחשב על זה כפתרון ל-A שימושה בפתרון ל-B בתור <code>sub-routine</code>.</p> <p>שפה הריקה (EMPTY): נגיד $\{\} = \{M : M \text{ is a TM s.t. } L(M) = \emptyset\}$ – שפת כל המ"ט שלא מקבלות אף קלט, בולם השפה שלهن ריקה. נרצה להוכיח $R \notin L_{EMPTY}$. נניח בשלילה שיש מ"ט E שمبرיעת את בעית העצירה L_{EMPTY}. בונה מ"ט A שمبرיעת את L_{ACC} בינהן קלט $\langle M, x \rangle$:</p>
--	---

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">• תבנה תכנית חדשה $\langle M_x \rangle$ עם התוכנה: <ul style="list-style-type: none"> ◦ אם $x \in L(M)$ מקבלת $\Leftrightarrow \emptyset \neq L(M_x) \neq \emptyset$ בולם יש מילה כלשהי ש-M_x מקבלת. ◦ אם $x \notin L(M)$ מקבלת $\Leftrightarrow \emptyset = L(M_x)$ לא מקבלת אף מילה. • מריצה את $\langle M_x \rangle$. ◦ אם E דוחה – השפה של M_x לא ריקה ולכן נקבע. ◦ אם E מקבלת – השפה של M_x ריקה ולכן נדחה. </td><td> <p>מכريع ל- L_{ACC}</p> </td></tr> <tr> <td style="vertical-align: top; padding-right: 20px;"> <p>איך נבנה את $\langle M_x \rangle$?</p> <ul style="list-style-type: none"> • בהינתן קלט u, היא מתעלמת ממנו. • מריצה את $\langle x, M_x \rangle$. M_x מקבלת $\Leftrightarrow M$ מקבלת. בפרט: <ul style="list-style-type: none"> ◦ אם $L(X) \neq \emptyset \neq \Sigma^*$ מקבלת $\Leftrightarrow L(M_x) \neq \emptyset \neq \Sigma^*$ כי לא משנה מה u, היא מקבלת. ◦ אם $L(x) \neq \emptyset \neq \Sigma^*$ מקבלת $\Leftrightarrow L(M_x) \neq \emptyset \neq \Sigma^*$ כי לא משנה מה u, היא דוחה. </td><td> <p>תכנית עזר</p> </td></tr> </table>	• תבנה תכנית חדשה $\langle M_x \rangle$ עם התוכנה: <ul style="list-style-type: none"> ◦ אם $x \in L(M)$ מקבלת $\Leftrightarrow \emptyset \neq L(M_x) \neq \emptyset$ בולם יש מילה כלשהי ש-M_x מקבלת. ◦ אם $x \notin L(M)$ מקבלת $\Leftrightarrow \emptyset = L(M_x)$ לא מקבלת אף מילה. • מריצה את $\langle M_x \rangle$. ◦ אם E דוחה – השפה של M_x לא ריקה ולכן נקבע. ◦ אם E מקבלת – השפה של M_x ריקה ולכן נדחה.	<p>מכريع ל- L_{ACC}</p>	<p>איך נבנה את $\langle M_x \rangle$?</p> <ul style="list-style-type: none"> • בהינתן קלט u, היא מתעלמת ממנו. • מריצה את $\langle x, M_x \rangle$. M_x מקבלת $\Leftrightarrow M$ מקבלת. בפרט: <ul style="list-style-type: none"> ◦ אם $L(X) \neq \emptyset \neq \Sigma^*$ מקבלת $\Leftrightarrow L(M_x) \neq \emptyset \neq \Sigma^*$ כי לא משנה מה u, היא מקבלת. ◦ אם $L(x) \neq \emptyset \neq \Sigma^*$ מקבלת $\Leftrightarrow L(M_x) \neq \emptyset \neq \Sigma^*$ כי לא משנה מה u, היא דוחה. 	<p>תכנית עזר</p>
• תבנה תכנית חדשה $\langle M_x \rangle$ עם התוכנה: <ul style="list-style-type: none"> ◦ אם $x \in L(M)$ מקבלת $\Leftrightarrow \emptyset \neq L(M_x) \neq \emptyset$ בולם יש מילה כלשהי ש-M_x מקבלת. ◦ אם $x \notin L(M)$ מקבלת $\Leftrightarrow \emptyset = L(M_x)$ לא מקבלת אף מילה. • מריצה את $\langle M_x \rangle$. ◦ אם E דוחה – השפה של M_x לא ריקה ולכן נקבע. ◦ אם E מקבלת – השפה של M_x ריקה ולכן נדחה.	<p>מכريع ל- L_{ACC}</p>			
<p>איך נבנה את $\langle M_x \rangle$?</p> <ul style="list-style-type: none"> • בהינתן קלט u, היא מתעלמת ממנו. • מריצה את $\langle x, M_x \rangle$. M_x מקבלת $\Leftrightarrow M$ מקבלת. בפרט: <ul style="list-style-type: none"> ◦ אם $L(X) \neq \emptyset \neq \Sigma^*$ מקבלת $\Leftrightarrow L(M_x) \neq \emptyset \neq \Sigma^*$ כי לא משנה מה u, היא מקבלת. ◦ אם $L(x) \neq \emptyset \neq \Sigma^*$ מקבלת $\Leftrightarrow L(M_x) \neq \emptyset \neq \Sigma^*$ כי לא משנה מה u, היא דוחה. 	<p>תכנית עזר</p>			

<p>קיבלנו סתייה לכך ש- L_{ACC} אינה ברעה, ולכן גם L_{EMPTY} אינה ברעה. הראנו דоказיה מ-L_{ACC} ל-L_{EMPTY}.</p> <p>שפות רגולריות (REG): נגיד $\{\} = \{M : M \text{ is a TM s.t. } L(M) \text{ is regular}\}$, שפת כל המ"ט שהשפה שלhn רגולרית.</p> <p>נרצה להוכיח $R \notin L_{REG}$. נניח בשלילה שיש מ"ט G שمبرיעת את L_{REG}. בונה מ"ט A שمبرיעת את L_{ACC} בינהן קלט $\langle M, x \rangle$:</p>	
---	--

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">• תבנה תכנית חדשה $\langle M_x^{01} \rangle$ עם התוכנה: <ul style="list-style-type: none"> ◦ אם $x \in L(M)$ מקבלת $\Leftrightarrow \Sigma^* = L(M_x^{01})$ בום יש מילה כלשהי ש-M_x^{01} מקבלת. ◦ אם $x \notin L(M)$ מקבלת $\Leftrightarrow \{0^n 1^n n \geq 0\} = L(M_x^{01})$ שהוא אינה רגולרית. • אחריה, מריצה את $\langle x, M_x^{01} \rangle$. M_x^{01} מקבלת $\Leftrightarrow M$ מקבלת. בפרט: <ul style="list-style-type: none"> ◦ אם $(x, M_x^{01}) \neq \emptyset \neq \Sigma^* = L(M_x^{01})$ כי לא משנה מה u, היא מקבלת. ◦ אם $(x, M_x^{01}) \neq \emptyset \neq \{0^n 1^n n \geq 0\} = L(M_x^{01})$ כי היא מקבלת ורק את u ודווחה בלבור אחר. </td><td> <p>תכנית עזר</p> </td></tr> </table>	• תבנה תכנית חדשה $\langle M_x^{01} \rangle$ עם התוכנה: <ul style="list-style-type: none"> ◦ אם $x \in L(M)$ מקבלת $\Leftrightarrow \Sigma^* = L(M_x^{01})$ בום יש מילה כלשהי ש-M_x^{01} מקבלת. ◦ אם $x \notin L(M)$ מקבלת $\Leftrightarrow \{0^n 1^n n \geq 0\} = L(M_x^{01})$ שהוא אינה רגולרית. • אחריה, מריצה את $\langle x, M_x^{01} \rangle$. M_x^{01} מקבלת $\Leftrightarrow M$ מקבלת. בפרט: <ul style="list-style-type: none"> ◦ אם $(x, M_x^{01}) \neq \emptyset \neq \Sigma^* = L(M_x^{01})$ כי לא משנה מה u, היא מקבלת. ◦ אם $(x, M_x^{01}) \neq \emptyset \neq \{0^n 1^n n \geq 0\} = L(M_x^{01})$ כי היא מקבלת ורק את u ודווחה בלבור אחר. 	<p>תכנית עזר</p>
• תבנה תכנית חדשה $\langle M_x^{01} \rangle$ עם התוכנה: <ul style="list-style-type: none"> ◦ אם $x \in L(M)$ מקבלת $\Leftrightarrow \Sigma^* = L(M_x^{01})$ בום יש מילה כלשהי ש-M_x^{01} מקבלת. ◦ אם $x \notin L(M)$ מקבלת $\Leftrightarrow \{0^n 1^n n \geq 0\} = L(M_x^{01})$ שהוא אינה רגולרית. • אחריה, מריצה את $\langle x, M_x^{01} \rangle$. M_x^{01} מקבלת $\Leftrightarrow M$ מקבלת. בפרט: <ul style="list-style-type: none"> ◦ אם $(x, M_x^{01}) \neq \emptyset \neq \Sigma^* = L(M_x^{01})$ כי לא משנה מה u, היא מקבלת. ◦ אם $(x, M_x^{01}) \neq \emptyset \neq \{0^n 1^n n \geq 0\} = L(M_x^{01})$ כי היא מקבלת ורק את u ודווחה בלבור אחר. 	<p>תכנית עזר</p>	

שפטות שווות (EQ): נגדיר $\{M_1, M_2 : L(M_1) = L(M_2)\}$, שפט כל זוגות המ"ט שמקבלות את אותה השפה. נרצה להוכיח $R \notin L_{EQ}$. נניח בשלילה שיש מ"ט Q שמכריעה את L_{EQ} . נבנה מ"ט E שמכריעה את L_{EMPTY} : E בהינתן קלט $\langle M \rangle$:

- תבנה מ"ט A_ϕ שמקיימת $\emptyset = L(A_\phi)$.
- מרים את $(\langle M \rangle, A_\phi) \in Q$.
- Q מקבלת \Leftrightarrow אנחנו מקבלים.

בדוגמה זו ביצענו את הדרוזקציה הבאה $EQ \leq EMPTY$

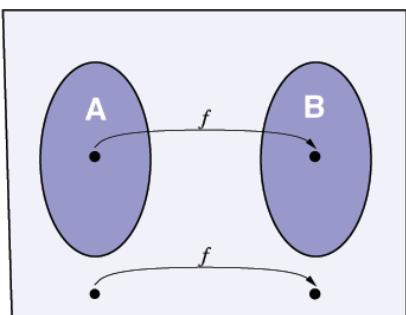
דוקציית מיפוי

רקע: עד כה, לפקטו שאלת מסויימת (האם מכונה מקבלת את השפה הריקה) ותרגמוני/מייפינו אותה לשאלת אחרת (האם שתי מכונות מקבלות את אותה השפה): $L_{EQ} \leq L_{EMPTY}$. במקרה אחר, קיבלנו שאלת (האם מכונה מקבלת קלט) ומייפינו אותה לשאלת (האם השפה שהמכונה מקבלת רגולרית או לא): $L_{ACC} \leq L_{REG}$. נשתמש בReLU זה עבור דוקציות מיפוי, נמפה קלט מבעה אחת לקלט לבעה אחרת בר שקלט שהוא עבר לקלט בשפה, וקלט שאינו בשפה עבר לקלט שאינו בשפה.

פונקציה חשיבתית: תהיו $(q_r, q_a, \delta, \Gamma, \Sigma) = M$ מ"ט, דומין $\Sigma \subseteq D$ ופונקציה $\{\Gamma^*\} \rightarrow f: D \rightarrow f(x)$. נאמר M - f מחשבת את f אם לכל קלט $x \in D$ M עוצרת ובסופו הרכזה בתובע על הסרט את התוצאה $\dots, \Gamma, f(x), \Sigma$. נאמר כי f חשיבתית אם קיימת מ"ט שמחשבת אותה.

האם כל הפונקציות מעלה מחרוזות ביןאריות הן חשיבותן לא! למשל $\{0,1\} \rightarrow \{0,1\}$: f של בעיית הקבלה, לוקחת קוד וקלט $\langle x, M \rangle$ ומחזירה 1 אם M מקבלת את x , 0 אם לא. הוכחנו שהבעיה הזאת היא לא חשיבתית. אם היינו יכולים לחשב אותה, היינו יכולים להכריע את בעיית הקבלה.

דוקציית מיפוי: יהיו Σ_B, Σ_A , $\Sigma_A \subseteq \Sigma_B^*$, $B \subseteq \Sigma_A^*$. דוקציית מיפוי מ- A -ל- B היא פונקציה $x \in A \mapsto f(x) \in B \Leftrightarrow x \in \Sigma_A^* \rightarrow f(x) \in \Sigma_B^*$. בפרט, כדי לדעת האם $x \in A$ מספיק לבדוק את $f(x) \in B$. נסמן $B \leq_m A$ אם $f(x) \in B \Leftrightarrow x \in A$.



טענה: אם $B \leq_m A$ בריעה, אז A בריעה.
הוכחה: יהיו M_B מכירע ל- B , אז $(x) \in M_B$: תחשב את $(x) f$ כי דוקציית המיפוי מ- A -ל- B , ואז תריע את $(x) f$.

נשתמש בכך בעיקר כדי להוכיח אי-בריאות. בפרט מהטענה הנ"ל, אם $B \leq_m A$, אז A לא בריעה, אז גם B לא בריעה.

בעיית העצירה (HALT) באמצעות דוקציית מיפוי: נרצה להוכיח כי $R \notin L_{HALT}$ באמצעות דוקציית מיפוי $ACC \leq_m HALT$

פונקציה חשיבתית	מבנה (w) :
<ul style="list-style-type: none"> • אם w אינו קידוד של מ"ט וקלט, נרצה למפותו למשהו שאינו בשפה ונחזר $w \notin L_{HALT}$ (למשל $w = y$). • אם $\langle x, M \rangle = w$ קידוד חוקי, נחזר קידוד $\langle M_\infty, x \rangle$. <p>באשר המכונה M_∞ מוגדרת בר על קלט y:</p> <ul style="list-style-type: none"> • מרץ את y. • אם M מקבלת – נקבע. • אם M דוחה – ניכנס לוולאה אינסופית (לא נעצור). 	פונקציה חשיבתית
<ul style="list-style-type: none"> • מתקיים כי f חשיבתית, ומתקיים $w \in L_{ACC} \Leftrightarrow w \in L_{HALT}$. • אם w לא קידוד חוקי, גם המיפוי שלו לא יהיה בשפה. • אם $w = \langle M, x \rangle$ <p>$f(w) \in L_{HALT}$ מ"ט מקבלת, $(x) M_\infty$ מקבלת ובפרט עוצרת, ולכן $f(w) \in L_{HALT}$</p> <p>$f(w) \notin L_{HALT}$ מ"ט דוחה/לא עוצרת, $(x) M_\infty$ בהכרח לא עוצרת, ולכן $f(w) \notin L_{HALT}$</p>	דוקציית מיפוי



$\text{HALT} \leq_m \text{HALT}_\epsilon$: נרצה להוכיח כי $R \in L_{\text{HALT}_\epsilon}$ באמצעות רזוקציית מיפוי $\cdot L_{\text{HALT}_\epsilon} = \{\langle M \rangle : M \text{ is a TM that halts on } \epsilon\}$

<p>בננה (w):</p> <ul style="list-style-type: none"> אם w אינם קיודד חוקי של מ"ט וקלט, נחיזר $L_{\text{HALT}_\epsilon} \notin w$ (קוד שאינו עוצר על הקלט הריק). אם $\langle x, M \rangle = w$ קיודד חוקי, נחיזר קיודד $\langle M_x \rangle$. <p>המוכנה M_x מוגדרת על קלט u (במו' קודם – היא פשוט מהקה את (x, M) לחלווטן): בהינתן קלט u, היא מתעלמת ממנו. מריצה את (x, M). M_x מקבלת $\Leftrightarrow M$ מקבלת.</p>	פונקציה חשיבה
<p>מתקיים כי f חשיבה, ומתקיים $w \in L_{\text{HALT}_\epsilon} \Leftrightarrow f(w) \in L_{\text{HALT}}$:</p> <ul style="list-style-type: none"> אם w לא קיודד חוקי, גם המיפוי שלו לא יהיה בשפה. אם $\langle x, M \rangle = w$ ○ אם $M(x)$ עצרת, $\langle x, M_x \rangle$ עצרת, ולכן M_x לא עצרת, ולכן ○ אם $M(x)$ לא עצרת, $\langle x, M_x \rangle$ לא עצרת, ולכן ○ $f(w) \notin L_{\text{HALT}_\epsilon}$ 	רזוקציית מיפוי

חומר סימטריון: אם $R \in A$ ברעה ו- $R \notin B$ אינה ברעה, אז $A \leq_m B \nleq_m A$ הוכח:

$B \nleq_m A$: כיוון שאם הייתה רזוקציה בזאת, $-A$ ברעה, אז גם B הייתה ברעה – אבל היא לא.
 $A \leq_m B$: נראה רזוקציה מיפוי M -ל- B . מכיוון ש- B לא ברעה, קיימים $b, \bar{b} \in B$, $b \neq \bar{b}$, לא יכולה להיות שפת כל המחרוזות, או שפה של אף מחרוזת (השפה הריקה אינה ברעה). נגידר בהתאם את פונקציית המיפוי הבאה:
 \circ $f(w) = \begin{cases} b & w \in A \\ \bar{b} & w \notin A \end{cases}$ ברעה ולכן ככל מילה נוכל לדעת האם היא ב- A או לא.

רזוקציות מיפוי ו- RE :

$$\text{טענה: } \text{הו } \Sigma \text{ א"ב ויהי } \Sigma^* \subseteq A, B \subseteq_m A, B \text{ כך שמתקיים } .A \leq_m B \Rightarrow A \in RE \quad (1)$$

$$.B \in coRE \Rightarrow A \in coRE \quad (2)$$

הוכחה:

- (1) אותה הוכחה רק ש- M_B מקבל ולא בהכרח מבירע, וה- M_A היה מקבל ולא בהכרח מבירע.
- (2) נרצה להראות: אם קיימת $M_{\bar{B}}$ המקבלת את \bar{B} אז קיימת $M_{\bar{A}}$, זה נובע מהטענה הבאה: **אם $B \leq_m A$ אז מתקיים $\bar{B} \leq_m \bar{A}$ עם אותה f (פונקציית מיפוי)** בדיק.

טענה: $\overline{L_{ACC}} \notin RE$

הוכחה: אם $L_{ACC} \in RE \cap coRE = R$ ובפרט $\overline{L_{ACC}} \in RE$ אז $\overline{L_{ACC}} \in RE$

בולם, בעזרת רזוקציות מיפוי נוכל לעשות יותר מלהראות א-בריעות, ובעור שפות מסוימות נוכל להראות שהן לא- RE . ניעזר במה שראהנו ונפתח את השיטה הבאה: **אם $B \leq_m \overline{L_{ACC}}$ אז נוכל להסיק ש- B לא- RE . אם היא הייתה ב- RE אז גם $\overline{L_{ACC}}$ – בסתריה.**

שפות שוות (EQ) באמצעות רזוקציית מיפוי: נרצה להוכיח כי EQ בולם $R = RE \cap coRE$ באמצעות רזוקציה מיפוי f :

$$. \overline{L_{ACC}} \leq_m \overline{L_{EQ}}$$

<p>בננה (w):</p> <ul style="list-style-type: none"> אם w אינם קיודד חוקי של מ"ט וקלט, נחיזר $L_{EQ} \in w$ (קיודדים שהשפות שלהם שוות). אם $\langle x, M \rangle = w$ קיודד חוקי, נחיזר קיודדים $\langle M_x, A_{\Sigma^*}, A_{\Sigma^*} \rangle$. <p>המוכנה M_x מקבלת $\Leftrightarrow M$ מקבלת, והמוכנה A_{Σ^*} מקבלת הכל.</p>	פונקציה חשיבה
<p>מתקיים כי f חשיבה, ומתקיים $w \in \overline{L_{ACC}} \Leftrightarrow f(w) \in \overline{L_{EQ}}$:</p> <ul style="list-style-type: none"> אם w לא קיודד חוקי, גם המיפוי שלו לא יהיה בשפה. אם $\langle x, M \rangle = w$ ○ אם $M(x) \in \overline{L_{ACC}}$, $w \notin \overline{L_{EQ}}$, ולכן ○ המוכנות $\langle M_x, A_{\Sigma^*} \rangle$ שוות. ○ אם $M(x) \notin \overline{L_{ACC}}$, $w \in \overline{L_{EQ}}$, ולכן ○ המוכנות $\langle M_x, A_{\Sigma^*} \rangle$ שוות. של המוכנות $\langle M_x, A_{\Sigma^*} \rangle$ שוות זו מזו. 	רזוקציית מיפוי

בעור $\overline{L_{ACC}} \leq_m L_{EQ}$ רק נחליף את $*_{\Sigma}$ ב- ϕ .



תרגיל 7 (דווקציות)

(בכל התרגילים נתעלם מהמקרה של קידוד לא חוקי).

תרגיל 1: הוכחו/הפריכו: $L_1 = \{\langle M \rangle : M \text{ is a TM that visits one state at least twice when running on } \epsilon\} \in R$

פתרונות: נשים לב כי מתקיים $RE \in L_1$ (נרייך ונבדוק עם מכונה שמחזיקה counter). נוביח כי $\epsilon \in L_1$. נראה מ"ט A שمبرיעה את L_1 .

1. מוצאת בקידוד את $|Q_M|$.
 2. מרים את $(\epsilon)M$ במשר כלבי יותר $+ |Q_M|$ צעדים (או עד שהוא עוצרת).
 3. נרשום בסרטן נפרד את כל המ מצבים בהם M בקרה.
- a. אם מצב מופיע פעמיים – נקבל (אם הגענו לכמה המצבים הזה, או שהוא הגיע פעמיים לאוטו מצב לפחות
策דים נמצאה אותה. אם M לא עצרת במצב הפעמיים היא בודאות עוצרת תוך פחות צעדים מ- $|Q_M|$, לא יהיה
מצב זה שיפוריע בעתיד, כי הריצה הסתיימה).
- b. אחרת – נדחה (אם M לא הגעה לאוטו מצב הפעמיים היא בודאות עוצרת תוך פחות צעדים מ- $|Q_M|$, לא יהיה
מצב זה שיפוריע בעתיד, כי הריצה הסתיימה).

תרגיל 2: הוכחו/הפריכו: $L_2 = \{\langle M, q \rangle : M \text{ is a TM that visits } q \text{ at least twice when running on } \epsilon\} \in R$

פתרונות: גם כאן $RE \in L_2$ (נרייך ונבדוק). נרצה להוכיח כי $\epsilon \notin L_2$ (באנו אין לנו שליטה מה המצב הגרפי שעוברים בו פעמיים)
 $HALT_\epsilon \leq_m L_2$.

<p>בבנה (w): עבור $\langle M \rangle = w$ נחיזיר קידוד $\langle q', M' \rangle$. המכונה M':</p> <ul style="list-style-type: none"> • מוגדרת בדומה ל-M רק עם מצב חדש נוסף q. • מרים את $(\epsilon)M$. אם M עצרת (מקבלת או דוחה) – M' עוברת ל-q ונשארת שם לנצח. • אין אף מעבר אחר ל-q. 	<p>פונקציה חשיבה</p>
<p>מתקיים ב-f חשיבה, ומתקיים $f(w) = \langle M', q \rangle \in L_2 \Leftrightarrow w = \langle M \rangle \in L_{HALT_\epsilon}$:</p> <ul style="list-style-type: none"> • אם M עצרת על המילה הריקה, אז M' מבקרת ב-q על המילה הריקה לפחות פעמיים (בפועלן אנסוף פעמיים), ולכן $f(w) \in L_2$. • אם M לא עצרת על המילה הריקה, אז M' לא תגיע ל-q לעולם, ולכן $f(w) \notin L_2$. 	<p>דווקציות מיפוי</p>

תרגיל 3: הוכחו כי $L_\infty = \{\langle M \rangle : M \text{ is a TM and } |L(M)| = \infty\} \notin RE \cup coRE$

פתרונות: נראה כי $\epsilon \notin L_\infty$ ו- $L_\infty \notin coRE$.

נוכיח $\epsilon \notin coRE$: נבצע דווקצית מיפוי $L_\infty \leq_m HALT$ כי ידוע לנו $L_\infty \notin coRE$ ובכך נקבל $HALT \leq_m L_\infty$.

<p>בבנה (w): עבור $\langle x \rangle = w$ נחיזיר קידוד $\langle M_x \rangle$. המכונה M_x:</p> <ul style="list-style-type: none"> • בהינתן קלט z, מתעלמת ממנו. • מרים את $\langle x \rangle M$ ומתקבלת M מקבלת. 	<p>פונקציה חשיבה</p>
<p>מתקיים ב-f חשיבה, ומתקיים $f(w) = \langle M_x \rangle \in L_\infty \Leftrightarrow w = \langle M, x \rangle \in L_{HALT}$:</p> <ul style="list-style-type: none"> • אם M עצרת אז M_x עצרת עבור כל הקלטים, ולכן $f(w) \in L_\infty$. • אם M לא עצרת, אז M_x גם לא עצרת ובפרט $\emptyset = L(M_x)$, ולכן $f(w) \notin L_\infty$. 	<p>דווקציות מיפוי</p>

נוכיח $\epsilon \notin L_\infty$: נבצע דווקצית מיפוי $L_\infty \leq_m HALT$ כי ידוע לנו $L_\infty \notin RE$ ובכך נקבל $RE \notin L_\infty$.

<p>בבנה (w): עבור $\langle x, M \rangle = w$ נחיזיר קידוד $\langle B_{M,x} \rangle$. המכונה $B_{M,x}$ מוגדרת על קלט z:</p> <ul style="list-style-type: none"> • מרים את $\langle x \rangle M$ במשר z צעדים. • אם M לא עצרת (לא משינה כמה צעדים נרייך היא לא תעוצר) – נקבל. • אם M עצרת – נדחה. 	<p>פונקציה חשיבה</p>
<p>מתקיים ב-f חשיבה, ומתקיים $f(w) = \langle B_{M,x} \rangle \in L_\infty \Leftrightarrow w = \langle M, x \rangle \in L_{HALT}$:</p> <ul style="list-style-type: none"> • אם $\langle x, M \rangle$ לא עצרת אז $\langle B_{M,x} \rangle = \Sigma^*$ כי קיבל הכלול, ולכן $f(w) \in L_\infty$. • אם $\langle x, M \rangle$ עצרת אז $\langle B_{M,x} \rangle = \emptyset$ כי נקבע \emptyset, ולכן $f(w) \notin L_\infty$. 	<p>דווקציות מיפוי</p>

הערות:

- יכולנו גם להראות בחלק השני: $\overline{L}_\infty \leq_m HALT$.
- בעת נוכל להשתמש בעובדה שאם עברו שפה A מתקיים שאם $A \leq_m L_\infty \text{ ו } A \notin RE \cup coRE$.

תרגיל 4: הוכחו/הפריבו: $L_{100} = \{\langle M \rangle : M \text{ on } \epsilon \text{ does not use more than 100 places on the tape}\} \in R$

פתרון: ברור לנו $coRE \in L_{100}$, כי קל לנו לפסול ברגע שהוא עוברת יותר מ-100 תאים על הסרט. באופן כללי, על מוכנה, לא נוכל לדעת האם היא נמצאת בלולאה אינסופית. אבל, אם הגבילנו את במות המקומות בסרט (לא עוברים את תא 100), כן נוכל לדעת.

הטענה נוכנה. נזכיר כי קונפיגורציה של מ"ט היא מהירות sqn בך-ש- $Q \in q -^* \Gamma, n$, ומיקומו של q ב מהירות הוא במיומו של הראש הקורא. במו כן, אם מ"ט **חודרת על אותה קונפיגורציה פעמיים, היא נכנסת לlolאה אינסופית**: כי הקונפיגורציה מחזיקה את כל המידע על מצב העולם מבחינת המוכנה, והיא דטרמיניסטית. אם $\langle M \rangle$, כמה קונפיגורציות אפשריות יש ל- M ? צריך לבחור את המצב הנוכחי, מיקום הראש הקורא ותוכן הסרט. לכן, נגידו $a \in Q_M$. נרץ את המוכנה $1 + (M) a$ צעדים ונבדוק האם היא עברה את תא 100 או לא.

בננה מ"ט שתכבריע את L_{100} . המוכנה $'M$ על קלט $\langle M \rangle$:

1. מחשבת את $|\Gamma|, |Q|$ עברו M .
2. נרץ $(\epsilon) M$ במשר $+ (M) a$ צעדים (או עד ש- M עצרת).
3. תוך כדי הריצה, נרשום על סרט נוסף את מספר התאים בהם M משתמשת.
 - אם M השתמשה ביוטר מ-100 תאים – דחה.
 - אחרת – קבל.

בנייה: ברור כי $'M$ תמיד עצרת. בעת:

- אם $'M$ מקבלת, אז מתקיים אחד מהשנים:
 - $(\epsilon) M$ עצרה ולא השתמשה ביוטר מ-100 תאים.
 - $(\epsilon) M$ לא עצרה, תוך $1 + (M) a$ לא השתמשה ביוטר מ-100 תאים, ולכן יכולות להיות קומות חדשות, בפרט מקום חדש תא חדש בסרט שהוא לא בקרה בו.
- אם $'M$ דחתה, אז M השתמשה ביוטר מ-100 תאים.

משפט ריס

עד ראיינו מספר תוכנות או-ברישות פרטניות. בעת ניתן אפיון של משפחה רחבה של בעיות שאין ניתן להכרעה. בהינתן $\langle M \rangle$ האם ניתן להכריע את בעיות עסקת ב"הבנייה קוד של תוכנית". בהינתן $\langle M \rangle$ האם מקבלת מה שראינו, משפחה זו של בעיות הבאות: האם $L(M)$ יותר מ-10 מצבים? האם M מקבלת ϵ תוך מאה צעדים? האם M מקבלת את ϵ בכל?^ל האם M מקבלת מספר סופי של קלטים? משפט ריס אומר שבבניהן קוד של תוכנית, לא ניתן להכריע בעיות התלוויות אך ורק בשפה $L(M)$ שהתוכנית מקבלת (אלא אם הן טריויאליות). בעיות אלו נקראות **סמנטיות**, ולא ניתן להכריע בעיות לא טריויאליות בכלל.

משפט ריס: $\text{ההשפות } C \subseteq RE \text{ אוסף שפות ב-RE (תכמה סמנטית), כך שמתקיים }$
 $L_C = \{\langle M \rangle : M \text{ is a TM and } L(M) \in C\} \notin R$

הערות:

- אם יש תוכנה של שפות, ונותנים לנו קוד של מוכנה, לא נוכל לקבוע אם השפה שהמוכנה מקבלת מקיימת את התכמה.
- עברו $\emptyset = C$ זה אכן לא מתקיים: $L_\emptyset, L_{RE} \in R$.

דוגמה: נגדיר $\{Primes\} = \{p \in \mathbb{N} : p \text{ is prime}\}$. נגדיר את השאלה $\{Primes\} = \{M \in TMs : L(M) = Primes\}$. נוכיח את השאלה $\{Primes\} \in C$ כאשר האוסף אינו טריואלי, מתקיים $C = \{Primes\} \subseteq L(M)$ ולכן משפט ריס נסיק כי $EQPrimes \notin R$.

$$L_C = \{\langle M \rangle : M \text{ is a TM and } L(M) \in C\} \notin R$$

הערות:

- אם יש תוכנה של שפות, ונותנים לנו קוד של מוכנה, לא נוכל לקבוע אם השפה שהמוכנה מקבלת מקיימת את התכמה.
- עברו $\emptyset = C$ זה אכן לא מתקיים: $L_\emptyset, L_{RE} \in R$.

דוגמה: נגדיר $\{Primes\} = \{p \in \mathbb{N} : p \text{ is prime}\}$. נגדיר את השאלה $\{Primes\} = \{M \in TMs : L(M) = Primes\}$. נוכיח את השאלה $\{Primes\} \in C$ כאשר האוסף אינו טריואלי, מתקיים $C = \{Primes\} \subseteq L(M)$ ולכן משפט ריס נסיק כי $EQPrimes \notin R$.



הוכחת המשפט: יהי $RE \subset C \subset \emptyset$. נחלה לשני מקרים:

מקרה א: $C \notin \emptyset$. נראה רצוקיות מיפוי $L_C \leq_m HALT$. תהי $A \in C \in RE$, בפרט M_A כר ש- A .

<p>בננה (w):</p> <ul style="list-style-type: none"> אם w אינו קידוד חוקי של מ"ט וקלט, נחזר $L_{HALT_\varepsilon} \notin y$ (קוד שאינו עוצר על הקלט הריך). אם $\langle M, x \rangle = w$ קידוד חוקי, נחזר קידוד $\langle M_x^C \rangle$. <p>המכונה M_x^C מוגדרת על קלט y:</p> <ul style="list-style-type: none"> מricane את (x) M. מricane את (y) M_A ומתקבלת $\Leftrightarrow M_A$ מקבלת. 	<p>פונקציה חשיבה</p>
<p>מתקיים כי f חסיבה, ומתקיים $:f(w) \in L_C \Leftrightarrow w \in L_{HALT}$</p> <ul style="list-style-type: none"> אם w לא קידוד חוקי, נחזר $L_C \notin y$ אם $\langle M, x \rangle = w$ <ul style="list-style-type: none"> אם (x) M עצרת, $L(M_x^C) = L(M_A) = A \in C$, ולכן $L(M_x^C) \in L_C$ אם (x) M לא עצרת ולכן $L(M_x^C) = \emptyset \notin C$, ולכן $L(M_x^C) \notin L_C$ 	<p>רצוקיות מיפוי</p>

הוכחנו כי $R \notin L_C$. למעשה הוכחנו כי $R \in coRE$ כי $L_C \notin coRE$ כי $L_{HALT} \in RE \setminus R$ ולכן $L_{HALT} \notin coRE$.

מקרה ב: $C \in \emptyset$. נוכיח כי $L_C \notin RE$ ובפרט יתקיים $R \notin L_C$.

- נשים לב כי $\bar{C} = RE \setminus C$ גם אינה טריוויאלית ומתקיים $\bar{C} \notin \emptyset$.
- לפי מקרה א' מתקיים $HALT \leq_m L_{\bar{C}}$ ובפרט $HALT \notin L_{\bar{C}} \subseteq RE$.
- מכאן נובע כי $HALT \notin L_{\bar{C}} \subseteq RE$.
- כמו כן, $\{w : w \text{ is not a TM}\} = \bar{L}_{\bar{C}}$. מכיוון שניתנו להכريع האם קידוד הוא חוקי (לכן זה ב- RE), מובע כי $L_{\bar{C}} \in RE$ אך יתקיים $L_{\bar{C}} \in RE$ אז $L_C \in RE$ בסתייה.

שיםו לב שלמעשה מעבר למשפט ריס הוכחנו:

- הרחבה 1: אם $\emptyset \subset C \subseteq RE \setminus \{\emptyset\}$
- הרחבה 2: אם $\emptyset \in C \subset RE$

דוגמה: נזכיר בשפה L_{REG} , ונשתמש בהרחבה 2.
נשים לב כי $L_{REG} = L_C$ עבור $\{L : L \text{ is regular}\} = C$. כל שפה רגולרית היא ב- RE , אך יש שפות ב- RE שאין רגולריות (ראינו דוגמה לשפות ב- R כאן). לכן $L_{REG} \notin RE$.

טכנית נוספת – ריצות חסומות:

השפה המלאה: יהי Σ א"ב. נגידו $\{\langle M \rangle : M \text{ is a TM and } L(M) = \Sigma^*\}$.

טענה: $L_{ALL} \notin RE$.

הוכחה: נשים לב כי עבור $\{\Sigma^* : \Sigma \in C = \{L : L \text{ is regular}\}$ מתקיים $\emptyset \notin C$ ו- $\Sigma \in C$ ולן לא ניתן להשתמש בהרחבות ריס שראינו. נראה $ALL \leq_m HALT$. כיוון שאנו יודעים כי $L_{HALT} \notin RE$ אז נקבל כי $L_{ALL} \notin RE$.

<p>בננה (w):</p> <ul style="list-style-type: none"> אם w אינו קידוד חוקי של מ"ט וקלט, נחזר $L_{ALL} \notin y$ (קוד שאינו מקבל הכלול). אם $\langle M, x \rangle = w$ קידוד חוקי, נחזר קידוד $\langle B_{M,x} \rangle$. <p>המכונה $B_{M,x}$ מוגדרת על קלט y:</p> <ul style="list-style-type: none"> מricane את (x) M במשר y צעדים. אם M לא עצרת – נקבל. אם M עצרת – נדחה. 	<p>פונקציה חשיבה</p>
<p>מתקיים כי f חסיבה, ומתקיים $:f(w) \in L_{ALL} \Leftrightarrow w \in \bar{L}_{HALT}$</p> <ul style="list-style-type: none"> אם w לא קידוד חוקי, נחזר $L_{ALL} \notin y$ אם $\langle M, x \rangle = w$ <ul style="list-style-type: none"> אם (x) M לא עצרת אז $L(B_{M,x}) = \Sigma^*$ כי נקבל הכלול, لكن $L(B_{M,x}) \in L_{ALL}$ אם (x) M עצרת לאחר בדיק k צעדים אז $L(B_{M,x}) = \{y \in \Sigma^* : y < k\}$ ולכן $L(B_{M,x}) \notin L_{ALL}$ 	<p>רצוקיות מיפוי</p>



תרגול 8 (משפט ריס ומוודה)

תרגיל 1 (משפט ריס):

נזכר במשפט: תהא C תת-קבוצה לא טריוויאלית של RE . תהא $\{C \in C \mid L(C) \in \text{L}(M)\}$. אז $L_C \notin R$. מהי C היא תכונה (או, תת-קבוצה) לא טריוויאלית?

1. אם קיימות $L_1, L_2 \in C$ כך ש- $L_1 \wedge L_2 \notin C$. יש שפה שנמצאת ב- C , ויש שפה שלא נמצאת ב- C .

2. אם $M_1, M_2 \in L_C = L(M_1) \cup L(M_2)$. התכונה צריכה להיות **תכונה של שפה בלבד (לא מדברים על במתו הצעדים לקבל מילה)**.

שפה	האם ניתן להשתמש במשפט ריס?
$A_{TM,\epsilon} = \{\langle M \rangle \mid \epsilon \in L(M)\}$	כן, התכונה היא $\{L \in \text{RE} \mid \epsilon \in L\} = C$. זה אומר $\epsilon \in L$ אם ורק אם $L \in \text{RE}$. וזה אינו תכונה טריוויאלית, שכן $\emptyset \notin C$. <u>שוד על השפה זו</u> .
$L_1 = \{\langle M \rangle \mid M \text{ accepts } 101\}$	כן, נעזר בכך שלגайд שמכונה מקבלת מילה, זה שקול לאמירה שהמילה נמצאת בשפה של המכונה . משפט ריס מדבר על תכונה של שפות.
$L_2 = \{\langle M \rangle \mid M \text{ halts on } 101\}$	<u>$L_1 \in \text{RE}$</u> : אין נדע שמכונה לעולם לא תקבל את 101 ? נגידו את $101 \in L$. <u>$L_2 \notin R$</u> : $L_2 \in \text{RE}$ מוכיחת ש- $L_1 \in \text{RE}$ שהוא תת-קבוצה של RE שהיא לא טריוויאלית שכן $\emptyset \notin C$. <u>אך $R \neq L_C$</u> : $\emptyset \in C \in \Sigma^*$.
$L_3 = \{\langle M \rangle \mid L(M) \in \text{RE}\}$	לא, כי זו אינה תכונה של שפות – מדובר רק על האם השפה עצרת או לא, לא האם היא עצרת ומתקבלת או עצרת ודוחה, אין דרישת השפה של המכונה .
$L_4 = \{\langle M \rangle \mid M \text{ always halts}\}$	<u>$L_3 \in \text{RE}$</u> : נגידו M על קלט $\langle M \rangle$: <ul style="list-style-type: none"> • נבדוק האם מדובר בקידוד חוקי. • אם M דוחה – נעצור (נקבל או לדחה). • אם M עצרת – נקבע, אחרת – נדחה. <u>לא</u> , כי זו אינה תכונה של שפות. למשל ישן שתי M_1, M_2 כך ש- $\emptyset = L(M_1) = L(M_2)$. M_1 תקינה שירדווחה, M_2 תקינה שיישר נכנסת לולאה אינסופית.

תרגיל 2: הוכחו כי $R \notin \{M \mid L(M) \in R\}$.

משמעות ריס, R משומש- R תכונה לא טריוויאלית:

$$\Sigma^* \in C \quad \bullet$$

$$L_{HALT} \in \text{RE} \setminus R \quad \text{כפי } L_{HALT} \notin C \quad \bullet$$

הוכחו בעת כי $\text{coRE} \notin R$. גנסה על המונחים, מתי נשתבע מה השפה של המכונה? זה מזכיר את L_∞ . בשום שלב לא נשתבע ש- C או שלא. בلومר בפועל $C \subseteq \text{coRE}$. נוכיח באמצעות דоказת $L_R \notin \text{coRE}$.

ניתן במכונה M_ϵ שעל קלט M מסמלצת את M על ϵ ועונה כמפורט. ידוע לנו כי $\epsilon \notin R$.

פונקציה חשיבה	בבנה (w): עבור $\langle x, M \rangle = w$ נחזיר קידוד $\langle B_{M,x} \rangle$. המכונה $B_{M,x}$ מוגדרת על קלט y :
דоказת מיפוי	<ul style="list-style-type: none"> • מರיצה את x במשר y צעדים. • אם M עצרת – נדחה. • אם M לא עצרת – נרץ את M_ϵ על y. נקבע $\Leftrightarrow M_\epsilon$ מקבלת.
$\text{התקיים כי } f \text{ חסיבה, ומתקיים } f(w) = \langle B_{M,x} \rangle \in L_R \Leftrightarrow w = \langle M, x \rangle \in L_{HALT}$	<ul style="list-style-type: none"> • עבור אם $\langle x, M \rangle = w$: • אם (x, M) לא עצרת אז $f(w) = L(B_{M,x}) = L(M_\epsilon)$, ולכן $f(w) \notin L_R$. • אם (x, M) עצרת לאחר בדיקת k צעדים אז $f(w) \in L_R$ ולכן $L(B_{M,x}) = \{y \in \Sigma^* : y < k\}$.



אופציה נוספת – **כדי להראות $L_R \in RE$ או $coRE$** נדרש לעשות רזוקציה מ- \overline{L}_{∞} .
נשים לב כי אנחנו רוצים להשתמש בשפה כלשהי שהיא ב- $R \setminus RE$ כמו $HALT$. קיימת M_H כך ש- $R \notin L(M_H) = HALT$.
בנוסף, אם עושים רזוקציה מ- \overline{L}_{∞} אנחנו מקבלים ההפך ממה שנצפה, ולכן **nezuka** מ- \overline{L}_{∞} שגם הוא לא ב-R.

פונקציה חשיבה	רזוקציה מיפוי
גנבה (w): $w \in \langle M \rangle = w \text{ נחזר קידוד } \langle M' \rangle$. המוכנה M' מוגדרת על קלט x :	מתקיים כי f חשיבה, ומתקיים $\overline{L}_{\infty} \in \langle M' \rangle \Leftrightarrow w \in \langle M \rangle \Leftrightarrow f(w) \in L_R$: <ul style="list-style-type: none"> אם $\overline{L}_{\infty} \notin w$ אז $w \in L_{\infty}$. לכן, $\infty = (M)$ בולם אינסופית אז שלב 1 בריצה יסתהים לכל x. ובשלב 2 מקבל $L(M') = L(M_H)$ ולכן $L_R \notin f(w)$. אם $\overline{L}_{\infty} \in w$ אז $w \in L(M)$. לכן, $k = (M)$ בולם סופית אז לכל x שקיימים $k > x$ שלב 1 לא יסתהים. אז את כל ה-x-ים האלה M' לא הולכת לקבל, ולכן $(M')_L$ סופית שכן $L_R \subseteq (M')_L$.

ריצה באופן מבוקר – טרייך שעשינו עם j , להריץ על אינסוף קלטים עד שימושו קורה, בלי להיתקע על קלט מסוים:

1. לכל $i = 1, 2, \dots$
- a. לכל $i, \dots, j = 1, 2, \dots$
- . הרץ את M על x_j (המילה ה- j בסדר הלקסיקוגרפי) למשך i צעדים.

מודא (Verifier): מודא לשפה L הוא מ"ט M שמקבל זוג קלטים (c, x) . נחשוב על c בתור רמז.

- אם $x \in c$ אז **קיים** c כך ש- M קיבל את x . בולם יש רמז שישבנו אותנו ש- x בשפה.
- אם $x \notin c$ מ"ט M תדחה את (c, x) . שום רמז שנתקבל לא ישבענו אותנו.

נכיח כי $c \in RE \Leftrightarrow \neg c \text{ יש מודא}$.

ל: תהא $RE \in L$. לכן קיימת מ"ט M_L המקבלת את L . בונה מ"ט דטרמיניסטי V כך שעלה קלט (c, x) ותסמלץ את M_L על x למשך $|c|$ צעדים, ותקבל אם "מ" M_L קיבלה. מתקיים:

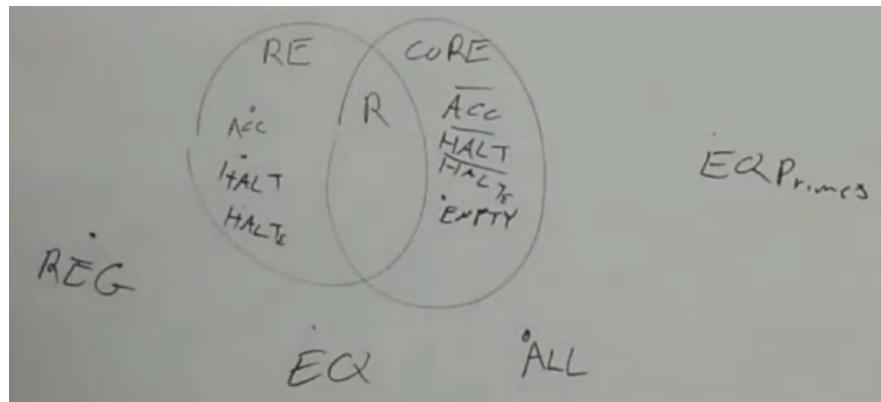
- אם $x \in c$ אז קיימ' c' כך ש- M_L מקבלת את x אחרי $|c'|$ צעדים ועוד 1.
- אם $x \notin c$ אז **לא** c' לא מקבל את x לכל מספר צעדים ועוד 0.

ר: יהא V מודא לשפה L מעיל. בונה מ"ט M_L המקבלת את L . על קלט x :

1. יהא c_1, c_2, \dots הסדר הלקסיקוגרפי של כל המילים ב- S^* – כל הרמזים האפשריים.
2. לכל $i = 1, \dots, |x|$:
 - a. לכל $i, \dots, j = 1, \dots, i$:
 - . סמלץ את (c_j, x) V למשך i צעדים.
 - ii. אם V מקבלת, M_L מקבל.

מתקיים:

- אם $x \in c$ אז מוכיחות המודא קיימ' c' כך ש- (c', x) V מקבלת ולבן גם M_L מקבל את x (כי קיימ' j כך ש- $c_j = c'$ ו- n כך ש- V עצרת על (x, c_j) לאחר i צעדים).
- אם $x \notin c$ מתקיים כי (c, x) V לא מקבלת (לא משנה לאחר כמה צעדים). לכן M_L לעולם לא מקבל.





3 – תורת הסיבוכיות

סיבוכיות זמן

מבוא

דוגמה: נסתכל על השפה שראינו בעבר $\{0^n1^n | n \geq 0\} = L$. ננסה להבין כמה זמן נדרש כדי להכريع את L עם מ"ט (חד-סרטית). כתוב אלגוריתם נאיבי M_1 :

1. נסרק את הסרט ונדחה אם מופיע 0 מימין ל-1.
2. כל עוד הסרט מכיל 0-1, נסרק באופן חוזר, ובכל סיריקה נמחק 0 ייחיד ו-1 יחיד.
3. אם לא נותרו 0 ו-1 נקבע, אחרת נדחה.

נחשב על **זמן הריצה במשаг של מספר הצעדים**, במה מערבי δ מבצעים כדי לפתור את הבעיה הזאת. זמן הריצה תלוי במובן באורך הקטלט. נחסום מלמעלה את זמן הריצה כפונקציה של אורכו הקטלט שיסמן $|x| = n$.

1. בדיקת השפויות של מעבר על כל הסרט פעם אחת (הולכים עד הסוף וחוזרים להתחלה) עולה $O(n)$.
2. מבצעים סיריקות הולכות וחזרות (לולאה). כל סיריקה עולה $O(n)$, ויש לנו לפחות n סיריקות.
3. הבדיקה האחורה גם היא בעלות $O(n)$.

החסם על מספר הצעדים הכלול הוא $(n^2)O$.

נסתכל בעת רך על מ"ט שתמיד עוצרות (אנחמו בתוך R).

ריצה בזמן T: תהי פונקציה $N \rightarrow N$: $T - M$ מ"ט. נאמר ש-**M ריצה בזמן (n)** אם לכל $N \in n$ ולכל קלט x באורך t , **M מבצעת לכל היוטר (n) T צעדים** לפני שהוא עצרת.

מחלקה DTime: בהינתן פונקציה $N \rightarrow N$, המחלקה DTime כוללת את כל השפטות שניתנו להכريع בזמן $(n)T$:

$$DTime(T(n)) = \{L(M) : M \text{ is a } \mathbf{single \, taped \, TM, \, running \, in } O(T(n))\}$$

- הדבר הזה מאפשר לנו להתעלם משאלות לא מהותיות בנוגע למודל ספירת הזמן (האם קרייה וכתייה במסגרת מעבר 6 צריכים להיספר בנפרד). لكن נגידיר את הזמן עד כדי קבוע - $O(n)$ ולא $T(n)$.
- ברור כי אם $t(n) \leq T(n)$ מתקיים $DTime(t(n)) \subseteq DTime(T(n))$.

היררכיית הזמן:

האם עוד זמן תמיד עוזר לאלגוריתמים? האם בהכרח באשר מגדים את זמן הריצה יש בעיות שנובל לפתור, כאשר לא יכוליםו לפתור אותן קודם (זמן ריצה קטן יותר)? משפט היררכיה הזמן אומר שיש דברים שאפשר לפתור ב- Δ^3 אבל לא ב- Δ^2 . באופן כללי יותר, בעיות שאפשר לפתור בזמן $(n)T$ אך לא בהרבה פחות. על מנת לנוכח את המשפט באופן פורמלי, נזדקק להגדירה של פונקציה חשיבה בזמן.

פונקציה חשיבה בזמן (time-constructible): פונקציה $N \rightarrow N$ היא חשיבה בזמן אם קיימת מ"ט שבහינת 1^n (או ביצוג אונארי) מחשבת את הקידוד הבינארי של $(n)T$ (coturbת ממש על הסרט) בזמן $(n)O$.

- פונקציה זו בהכרח מקיימת $(n)O = (n)T$.
- לדוגמה פונקציות "סבירות" כמו $\sqrt[n]{2}, \log^k n, n^k$. כל פונקציות הזמן שניתקל בהן בקורס הין חשיבות בזמן.

משפט היררכיה הזמן: תהי $(n)T$ פונקציה חשיבה בזמן ותהי $t(n)$ המקיימת $(n)O = o(T(n)) \cdot \log(t(n))$.

$$DTime(t(n)) \subsetneq DTime(T(n))$$

- בפרט, לכל $C \leq c \leq 1$ מתקיים $(n^c)O \subsetneq DTime(n^c)$.

רענון להוכחה: מ시מה שאפשר לבצע בזמן T אך לא ברור כיצד לבצע בזמן $T \ll t$, היא למשל לסמולץ תבנית שדורשת זמן של T צעדים קבועים האם היא מקבלת. איך נעשה את זה עם פחת צעדים? ההוכחה הפורמלית עובדת בעוררת לכשון. נראה בעת סקיצה של ההוכחה.



הוכחה: תהי (n) חישבה בזמן. נגידר מ"ט Flip שבහינתן קלט w (קידוד של מטריצה) באורך n :

1. מחשבת את $(n)T$: מספר הצעדים שאחננו מתעניינים בו.
2. אם $\langle M, 0^k \rangle = w$ עבור $M \in k$ (קידוד של אפסים בהמשך) ממשיכה, אחרת דוחה (קידוד לא חוקי).
3. מרים את מ"ט האוניברסאלית $\langle A, M \rangle U$ על המ"ט M שמקודדת ב- w עם w עצמו במשר $(n)T$ צעדים. אנחנו לא מסמלים T צעדים של M , אלא T צעדים של **חישוב במ"ט האוניברסאלית** (ואז יכול להיות שנctrורק יותר מ- T צעדים).
4. אם U עצרה וקיבלה – Flip דוחה. אחרת, Flip מקבלת.

טענה 1: $(\text{Flip}) \in \text{DTIME}(T(n))$.

1. $(n)T$ חישבה בזמן ולכן דורש $O(T(n))$.
2. עבר קידוד סביר הבדיקה עולה $O(n)$.
3. דורש $O(T(n))$ להרצת T צעדים במ"ט האוניברסאלית.
4. דורש $O(1)$.

טענה 2: $(L(\text{Flip})) \notin \text{DTIME}(t(n))$. תהי A אלגוריתם (מ"ט) שרצה בזמן $(n)t(n)$. נראה שعبור k מספיק גדול היא נבשלה על הקלט הבא: $\langle A, 0^k \rangle = w$ מתקיים $\langle A, 0^k \rangle \notin L(\text{Flip}) \Leftrightarrow w \in L(A)$ **ולומר הוא לא מביע את Flip** . רצחה להבטיח שהקלט יסימן את הריצה של A על אותו הקלט, כי אז הוא רואה מה A עונה ופשוט עונה הפוך. בולם $\langle A, 0^k \rangle \in L(A)$ **תעוצר** תוך T צעדים.

בעזר במשפט הבא: קיימת מ"ט אוניברסאלית U כך שלכל x, M אם $\langle x, M \rangle U$ עצרת תוך t צעדים, אז $\langle x, M \rangle$ עצרת תוך $t \log t$ צעדים.

מכיוון ש- $O(t(n) \log t(n)) = O(1)$ עבור k מספיק גדול, $|w| = n$ מספיק גדול כדי להבטיח שהוא $(n)T$ הוא מספיק כדי שהריצה המסומלצת של A תעוצר. במקרה זה $\text{Flip}(w)$ עונה הפוך M - $(w)A$.

תלות זמן הריצה במודל החישוב

דוגמה 2: נסתכל שוב על $\{0^n 1^n\} \subseteq L$. נבנה מ"ט משופרת M_2 .

שלב	סיבוכיות זמן
1. סריקת שפיות: נסורך את הסרט ונדחה אם מופיע 0 מימין ל-1.	בדיקת השפיות עולה $O(n)$.
2. כל עוד הסרט מכיל 0 ו-1: a. נסורך את הסרט ואם המספר הכלול בין 0 ו-1 איזוגי – נדחה. b. אחרת, נמחק כל 0 שכני וכל 1 שכני (אפקטיבית מוחקים חצי מהכמתות).	ב厮ירה עליה ($\log n$) סריקות.
3. אם לא נותרו 0 ו-1 נקבע, אחרת נדחה.	הבדיקה الأخيرة גם בעלות $O(n)$.

החסם על מספר הצעדים הכלול הוא $(\log n)$. אי אפשר להשיג זמן טוב יותר עם מ"ט חד-סרטי.

0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 #0 + #1 = 22

X 0 X 0 X 0 X 0 X X 1 X 1 X 1 X 1 X #0 + #1 = 10

X X X 0 X X X 0 X X X X X X 1 X X X 1 X X X #0 + #1 = 4

X X X X X X X 0 X X X X X X X X 1 X X X #0 + #1 = 2

X #0 + #1 = 0



משפט: אם $(o(n \log n)) \in L$ אז L רגולרית.

דוגמה 3: בניית מ"ט דו-סרטית M_3 . זמן הריצה כאן הוא (n) .

1. סריקת שפויות: נסורך את הסרטן ונמחה אם מופיע 0 מימין ל-1.
2. נעתיק את כל ה-0 מסרט הקלט לסרטן השני.
3. נעביר במקביל על ה-0 בסרטן הראשון וה-1 בסרטן השני, ועל כל 0 נמחק 1.
4. אם לא נותרו 0 ו-1 נקבל, אחרת נדחה.

אם רואים כאן הבדל אינגרנטי בין המודלים. במ"ט דו-סרטית (n) ובחד-סרטית $(n \log n)$. באופן כללי, בעוד שבחיבור של חישובות מודלים סבירים בהם דנו שקולים (יצרי-טיורינג), בהקשר של סיבוכיות זמן המודל משפייע. בעת הנסה להבין כמה משפייע בחירות המודל. נתחל במקרה של מ"ט רב-סרטית לעומת חד-סרטית.

טענה: כל מ"ט רב-סרטית בעלת זמן ריצה $a \geq T(n)$ ניתנת לסימולציה ע"י מ"ט חד-סרטית בעלת זמן ריצה $O(T^2(n))$.

הוכחה: יהיו $\lambda \in k$ ותהי M_k מ"ט k -סרטית. נזכר בסימולטור החד-סרטוי M_1 שהראינו בשיעורים קודמים. בהינתן קלט $x_1 \dots x_n = x$:

1. נרשום על הסרטן $\bar{x}_1 \# \bar{x}_2 \# \dots \# \bar{x}_n \# \bar{x}$.
2. נסורך את הסרטן על-מנת לקרוא את התווים מתחת הראש.
3. נסורך שוב על מנת לבתו חזרה תווים חדשים ולהזיז את הראשים (באופן וירטואלי ע"י תווים \bar{x}).
4. באשר הראש נע ימינה על אחד הסרטנים לתא ע' שאינו מאוחROL, נסיט את כל התווים שמיימי תא אחד ימינה.

זמן ריצה:

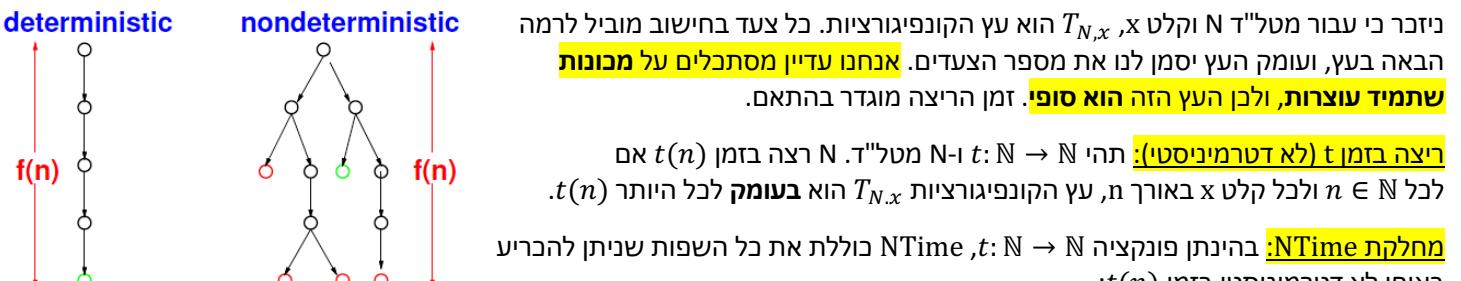
- לכל צעד של M_k , M_1 מבצעת לפחות בלב היותר $O(1)$ סריקות, $-O(1) = O(k)$ שיפוטים ימינה.

מכיוון ש- M_k מבצעת לפחות בלב היותר (n) T צעדים היא משתמשת בלב היותר (n) T תאים על כל סרטן (לא נובל להשתמש ביותר) נקבל כי:

- עלות בלב סריקה היא $O(T(n))$.
- עלות בלב שיפוט $O(T(n))$.
- סה"כ עלות כל צעד וירטואלי של M_k היא $O(T(n))$.
- סידור התחלתי של הסרטן הוא $O(n+k) = O(n+k)$.

סה"כ נקבל $(n) + T(n) \cdot O(T(n)) = O(T^2(n)) = O(T^2(n))$

זמן ריצה לא דטרמיניסטי:



$$NTime(t(n)) = \{L(N): N \text{ is a single taped non deterministic TM, running in } O(t(n))\}$$

טענה: כל מטל"ד בעלת זמן ריצה (n) ניתנת לסימולציה ע"י מ"ט (דטרמיניסטי) בעלת זמן ריצה $2^{O(t(n))}$. גשים לב כי בפרט זה מתקיים עבור סימולטור חד-סרטוי מהמשפט הקודם.

הוכחה: בהינתן מטל"ד N , בניית מ"ט M המסמלצת אותה: יהי b חסם על דרגת עץ הקונפיגורציות. M בהינתן קלט x :

1. לכל בתובת $\{a, b, \dots, 0\} \in a$ בסדר לקסיקוגרפי עולה (כל פעם יורדים רמה בעץ), נבדוק אם Ca (הkonfiguracija המתאימה) קיימת ומתקבלת. אם כן – נקבל.
2. באשר מגיעים לרמה האחורונה בעץ – דוחים.

זמן ריצה:

- הסריקה מגיעה לכל היותר לרמה $t(n)$.
- לכל כתובות $i^1, \dots, i^n \in \{0, \dots, b\}$, הזמן לבדוק האם Ca^i קיימת ומתקבלת היא $O(t(n))$. אם חוקרים ענף מסוים.
- מספר הכתובות הנבדקות הוא $O(b^n)$. בכל היותר-ב- b .

$$\text{סה"כ נקבל } O(t(n)) = 2^{O(\log(t(n)))} \cdot O(t(n)) = O(b^{t(n)}) \cdot O(t(n)).$$

סיכום: באופן כללי, עבור מודלים **דטרמיניסטיים** סבירים (מכונות טירוגר שב-סרטית, RAM, Python) **שקלים עד כדי תקורה פולינומית**. אם יש לנו ריצה בזמן $t(n)$ במודל אחד, תהיה לנו ריצה במודל אחר שהוא $Poly(t(n))$. בפרט:

- השאלה "האם בעיה פתירה בזמן לינארי" תלויות במודל.
- השאלה "האם בעיה פתירה בזמן פולינומי" לא תלויות במודל.

משמעותם שהפער האקספוננציאלי בין מודלים דטרמיניסטיים ללא-דטרמיניסטיים הוא **אינהרנטי**. מה שהופר את המודל ללא פיזיולוגי.

מחלקות P ו-NP

מחלקה P: מחלקת זו היא כל בעיות שניתן לפתור בזמן פולינומי עם מ"ט **דטרמיניסטיות**:

$$P = \bigcup_{c \in \mathbb{N}} DTime(n^c)$$

הערה לגבי **קיוד הקטלט**: בדר"כ אנחנו מניחים שלקלט יש מבנה כלשהו (גרפים, מטריצות, מספרים). סיבוכיות הזמן של אלגוריתמים מושפעת מואפן הקידוד ולעתים אף קובעת האם הבעיה ב-P או לא.

למשל, אלגוריתם למציאת גורם ראשוני בהינתן מספר n : עבור $\{n, \dots, 2\} \in i$ אם i מחלק את n נפלוט. זמן הריצה הוא $O(n)$. האם האלגוריתם לינארי באורך הקטלט? תלוי בקיוד: אם תקיוד באונארית 1^n אז כן. אם הוא מקיים בbijarity ע"י $log n$ ביטים אז האלגוריתם אקספוננציאלי.

בברית מחדל כאשר נדבר על בעיות עם מבנה, נניח כי **הקלט נתון בקיוד קצר**. למשל מספרים בקיוד בינארי/דצימלי, גרפים מקודדים במטריצה או רשימות שבנים. כל עוד אפשר לעבור בין קידודים בזמן פולינומי, לא משנה לנו באיזה קידוד משתמשים.

אם יש בעיות ב- $P \setminus R$? מבחון, למשל $P \notin DTime(n^{\log n})$. אלו הן בעיות שברירות לא בזמן פולינומי, לפי משפט ההיררכיה. נרצה למצוא בעיות אלגוריתמיות שאנו יכולים לתקן בהן, יותר טבעי, האם ה问题是 ב-P? פעמים רבות הנאיבי יהיה אקספוננציאלי.

דוגמאות לבעיות ב-P:

- האלגוריתם נאיבי יהיה לעבור על כל המסלולים, וזמן הריצה שלו יהיה אקספוננציאלי $N! = N^N$. אלגוריתם משופר יהיה DFS או BFS בזמן ריצה פולינומי.
- $\{p : p \text{ is prime}\}$. אלגוריתם נאיבי אקספוננציאלי. אלגוריתם פולינומי קיים אבל מאוד מורכב.

מחלקה NP: מחלקת זו היא כל בעיות שניתן לפתור בזמן פולינומי עם מ"ט **לא דטרמיניסטיות**:

$$NP = \bigcup_{c \in \mathbb{N}} NTime(n^c)$$

מסתבר שחלק גדול ביותר מהבעיות האלגוריתמיות שמעניינות אותנו הן ב-NP. בהמשך נראה שאלה שallow לא ניתן לפתור באופן יעיל, אבל ניתן להזות פתרון באופןיעיל. דוגמה: $\{G, s, t : G \text{ has a hamilton path from } s \text{ to } t\}$. HAMPATH =

מסלול המילוטוי בגרף מכון G הוא מסלול שעובר בכל צמתים הגרף, ועובר בכל צומת פעם אחת בלבד.

טענה: $HAMPATH \in NP$.

הוכחה: נרצה להראות שקיימת מטל"ד שմכברעה את הבעיה הזאת. אם נסתכל על עץ הקונפיגורציות, נמצא קונפיגורציה מקבלת אם יש מסלול, ואם אין מסלול בזה כל העלים לא יהיו מקבילים. מטל"ד עברו הבעיה:

- נבחר ("ננחש") באופן **לא דטרמיניסטי** סדרת צמתים v_n, \dots, v_1 .
- נבדוק באופן **דטרמיניסטי** האם מדובר במסלול המילוטוי.

זמן ריצה (העומק של העץ): פולינומי.

ויזוא פולינומי:

מודא פולינומי: תהי L מ"ט עם א"ב קלט Σ , ס ותהי $* \Sigma \subseteq L$. נאמר ש- L מודא פולינומי עבור L אם מתקיים:

- שלמות (completeness): לכל $L \in \Sigma$ קיים $x \in \Sigma$ כך ש- (x, L) מקבל.
- אנותות (soundness): לכל $L \notin \Sigma$ לא כל $x \in \Sigma$ תמיד (x, L) דוחה.
- יעילות: קיים פולינום $(n) \rightarrow k$ שכל $x \in \Sigma$ זמן הריצה של (x, L) הוא לכל היותר $(|x|)k$.

במילים אחרות, הוא מקבל את הבעה (הגראף), מועמד לפתרון (מסלול) ומודא אם אכן זה פתרון לבעה. מובטח שאם הבעה ניתנת לפתרון יש פתרון שיגרום לו לקבל, ואם לא, לא משנה איזה מועמד יהיה, הוא יוכל לדוחות. כל זה בזמן פולינומי.

הערות:

- האיבר w מכונה "עד"/"סרטיפיקט"/"הוכחה".
- המודא נדרש להיות פולינומי ב- $|x|$ (ולא $|w| + |x|$). בפרט לכל $L \in \Sigma$ מובטח שהקיים עד ש באורך לפחות $|x|p$.
- למשל ל-HAMPATH יש מודא פולינומי, כאשר עד הינו "מסלול המילוטני" מתאים.

טענה: $NP \in L \Leftrightarrow L$ יש מודא פולינומי.

הוכחה: הרעיון הוא שуд לשיעיות $L \in \Sigma$ הוא מעין בתובת לענף המקבל בעץ הקונפיגורציות.

\Leftarrow : נניח כי קיימת מטל"ד N עם זמן ריצה $(n) \rightarrow k$ המכירה את L . בונים מודא V עבור L .
 \Rightarrow יהי $(1)O = b$ חסם על דרגת עצי הקונפיגורציות של N . המודא (w, L) :

1. מודא כי w מתייחס לתובת בעץ הקונפיגורציות: בולם $\{0, \dots, b\}^k \rightarrow \{0, \dots, b\}^n$ הוא מודא L . אחרת – דוחה.
2. מחשב את הקונפיגורציה המתאימה C_w ומתקבל \leftrightarrow הקונפיגורציה המקורי.

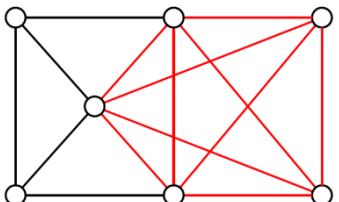
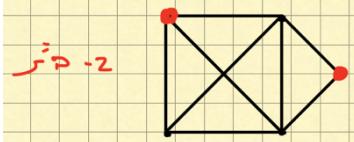
קיים w כך (w, L) מקבל \Leftrightarrow קיים ענף מקבל ב- $T_{N,x}$. זמן ריצה: $(|x|)dO$.

\Rightarrow : נניח כי קיים מודא (w, L) עם זמן ריצה $(|x|)d$. בונים מטל"ד N עם זמן ריצה פולינומי המכירה את L . המבונה N על קלט x :

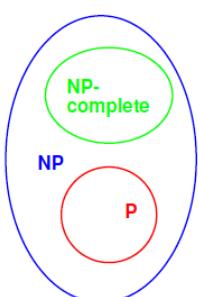
1. בוחרת באופן לא טרמיינטי w באורך $(n) \leq |w|$.
2. מקבל $\Leftrightarrow (w, L)$ מקבל.

N מקבל \Leftrightarrow קיים w באורך $(n) \leq |w|$ כך (w, L) מקבל.

דוגמאות:

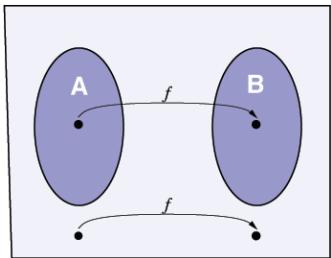
בעיה	הוכחה
	העד הינו תת-קבוצה S שבונה k -קליק. ניתן לבדוק שacky מהם מחוברים בקשר, נבדק שיש k קודקודים. קליק בגודל k מכונה k -קליק. $CLIQUE = \{(G, k) : G \text{ is a graph with a } k - \text{clique}\} \in NP$
	העד הוא הקבוצה הב"ת S . קבוצת צמתים בגרף לא מכון היא ב"ת אם אין קשת בין כל שניים. $IS = \{(G, k) : G \text{ is a graph with a } k - IS\} \in NP$
עד הוא המחלק n – אפשר להריץ אלגוריתם ולבדוק.	$FACTOR = \{(N, k) : \exists k \geq n > 1 \text{ which divides } N\} \in NP$

האם $NP = P$? בעיות אלגוריתמיות טיפוסיות ב- NP , ושאלת השאלה היא האם $NP = P$? כלומר, האם ניתן לפתור בעיות NP ביעילות. האמונה הרווחת היא כי $NP \neq P$ ואמונה זו מגובה בסוף רב. בפרט בעולם הкриптוגרפיה נשענים על ההנחה זו. כאמור, אנחנו לא יודעים להוכיח ש愧ה מושלמת ב- NP אינה ב- P . אם כן, באיזה מובן ניתן לטעון כי בעיות ב- NP הן "קשוחות"?



**בעיות NPC**

הבעיות השלומות (NP-Complete) הינן הבעיות "הקשות ביותר" ב-NP, במובן שאם נראה כי בעיה NP-שלמה ב-P, אזו כל בעיה ב-NP תהיה ב-P. זה יהיה הכל ההפוך שלנו לטעון כי בעיה היא "קשה".



חוקcitת מיפוי פולינומית: יהיו Σ_A, Σ_B , $\Sigma_A^* \subseteq \Sigma_B^*$. חוקcitת מיפוי פולינומית מ-A ל-B היא פונקציה $f: \Sigma_A^* \rightarrow \Sigma_B^*$: **חשיבות בזמן פולינומי**, כך שלכל $x \in \Sigma_A^*$ מתקיים $f(x) \in \Sigma_B^* \Leftrightarrow f(x) \in B$. כלומר $B \leq_P A$. ככלומר מתרגם ביעילות שאלת על שייכות ל-A לשאלה על שייכות ל-B. אם נפתרו את B ביעילות, אז יוכל לפתרו גם את A.

דוגמה: $CLIQUE \leq_P IS$

הוכחה: נגדיר את פונקציית המיפוי $\langle G, k \rangle = \langle \bar{G}, k \rangle$, כלומר נמפה לגרף המושלים (אותה קבוצת קודקודים, נסיר את הקשיותה הקיימות ונוסף את החסרות ביחס לגרף המקורי). קבוצת צמתים S ב-G היא קליק $\Leftrightarrow S$ היא ב"ת-ב- \bar{G} . הרדוקציה בבירור חשובה פולינומית.

טענה: אם $A \in P$ אז $B \in P$ -ו $A \leq_P B$.
הוכחה: תהי M_B המכנית את B. נבנה M_A המכנית את A. לכל קלט x:

1. M_A מחשבת את $f(x)$.
2. מרים את $(f(x))$ ומתקבלת/דוחה בהתאם.

אם M_B רצהה בזמן $P_B(n)$ ו- f -בזמן $P_f(n)$ אז M_A רצהה בזמן $O(P_f(P_b(n)))$.

שפה NP-שלמה: שפה L_0 היא NP-שלמה אם:

- **שייכות:** $L_0 \in NP$.
- **קיים:** לכל $NP \in L$ מתקיים $L \leq_P L_0$.

מחלקה השפות ה-NP שלמות מסומנת ב-NPC.
מסקנה מטענה קדמתה: אם קיימת NPC-ו $L_0 \in P$ אז $L_0 \in NPC$.

בעיית ACC_{NP}: נגדיר את הבעיה: $ACC_{NP} = \{\langle M, x, 1^t \rangle : M \text{ is a TM, and } \exists w. M(x, w) \text{ accepts in } t \text{ time}\}$

טענה: $ACC_{NP} \in NPC$.
הוכחה:

נראה $ACC_{NP} \in NP$: נגדיר את המודא הבא $\langle M, x, 1^t \rangle, w \rangle$ אשר מסמלץ t צעדים של w ב-M, ומתקבל $\Leftrightarrow M$ מקבלת. נראה $L \leq_P ACC_{NP}$ עבור $L \in NP$: יהי $V \in NP$ מודא עבור L עם זמן ריצה פולינומי $(n)^k$. נגדיר $\langle V, x, 1^{p(|x|)} \rangle$. נראה $f(x) = \langle V, x, 1^{p(|x|)} \rangle$. מתקיים $w \in L \Leftrightarrow \exists t. V(x, w) \text{ מקבל תור } (|x|)^p$ צעדים $\Leftrightarrow f(x) \in ACC_{NP}$.

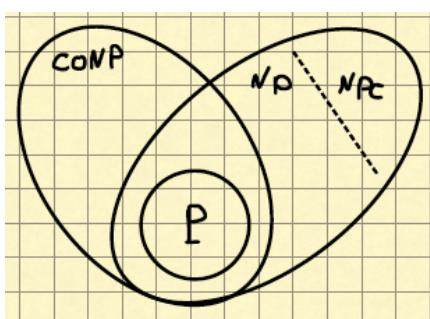
טענה: אם $B \in NPC$ ו- $A \in NPC$ -ו $A \leq_P B$, $B \in NP$.

בעיות שאין ב-NP:

נתבונן למשל על הבעיה **CLIQUE**, כאמור ב-G אין קליק בגודל k. לא ניתן לעבור על כל הקיליקים, יש $\binom{n}{k} \approx n^k$ באליה. להראות שמשהו לא קיימן זה נראה יותר קשה מהראות שמשהו קיימן. בעיות שימושיות בעיות ב-NP הן במחלקה coNP.

מחלקה coNP: $coNP = \{\bar{L} : L \in NP\}$.

יש לנו דוגמאות לבעיות שלא ידוע שהן ב-P אבל כן בחיתוך של $NP \cap coNP$. למשל, בעיית FACTOR: האם יש גורם קטן מ-k שמחילק מספר n.





תרגול 9 (P ו-NP)

המחלקות P ו-NP:

- המחלקה P:

$$P = \bigcup_{c \in \mathbb{N}} DTime(n^c)$$

o כל מה שאנו ידעים בזמן פולינומי להכريع – כן או לא.

- המחלקה NP:

$$NP = \bigcup_{c \in \mathbb{N}} NTime(n^c)$$

o $\in NP \Leftrightarrow$ יש מודא פולינומי: אם צריך להשתמש במשהו נתון ($L \in NP$), יותר נוח לעבוד עם זה.

o אולי כן אולץ לא הכל להכريع, אבל נוכל **להשתכנע מהר** – אם יש רמז שנקבל אותו (הرمز חייב להיות קצר, הריצה היא בזמן פולינומי ב- x אז אם c יותר מדי ארוך לא נצליח לקרוא אותו), דעת שהקלט הוא בשפה.

תרגיל 1: הוכחו כי $EXP \subseteq NP$, כאשר $.EXP = \bigcup_{c \in \mathbb{N}} DTime(2^{n^c})$

תהי $NP \in L$. לכן קיים ל- L מודא פולינומי V , כך של קלט x רץ לכל היותר ($|x|$) k צעדים. נרצה לבנות מ"ט דטרמיניסטיבית שתכיריע את L בזמן אקספוננציאלי.

M על קלט x :

1. לכל $\Sigma \in c$ כך ש- $(|x|)d \leq |c|$:

a. הרץ את $(c, V(x, c))$ למשך ($|x|)d$ צעדים.

b. אם V קיבל – נקבל.

2. נדחה.

זמן ריצה: עבור קלט x באורך t יש $O(2^{p(n)})$ עדמים אפשריים, ובכל ערך זמן הריצה הוא לכל היותר (n) p .
לכן סה"ב $O(2^{p(n)} \cdot n^p)$.

נכונות:

- אם $L \in x$ אז קיים "עד" $'c$ באורך לכל היותר ($|x|)d$ כך ש- $('c, x, V)$ רץ בכל היותר ($|x|)d$ צעדים ומתקבל. לכן נקבל.
 - אם $L \notin x$: לא קיים "עד", לכל c באורך לכל היותר ($|x|)d$ מתקיים כי $(c, V(x, c))$ לא מקבל תור ($|x|)d$ צעדים, לכן נדחה.
- לכן נסיק כי $L \in EXP$.

טיועני ריפוד:

משפט: אם $P = NP$ אז $EXP = NEXP$, כאשר $NEXP = \bigcup_{c \in \mathbb{N}} NTime(2^{n^c})$

הביוון $NEXP \subseteq EXP$ טריוויאלי. על מנת להוכיח את הביוון השני, תהי $NEXP \in L$ ונוכיח כי $L \in EXP$. ההוכחה תנבע משתי הטענות הבאות.

טענה 1: תהי $(L \in NP \text{ ונגדיר } L_{pad} = \{x01^{2^{|x|^c}} : x \in L\})$ הוכיחו $L_{pad} \in NTime(2^{n^c})$

לפי הגדרה, קיימת מטל"ד N שרצה בזמן $(n^p)2$ שמכריעת את L . בונה מטל"ד B המקבלת את L_{pad} בזמן פולינומי. השפה L_{pad} לוקחת את כל המילים בשפה, שמה חוץ "0" ומרפדת בהרבה "1", כמוות ה-1-ים היא $(|x|)2^p$.
 B על קלט y :

1. תבדוק ש- y מהצורה $01^{2^{|x|^c}}$, אם לא תדחה.
2. תרץ את N על x בלבד – ותעננה במזה.

זמן ריצה: פולינומי באורך הקלט y שהוא: $|x| + 2^{|x|^c}$.

נכונות: ברורה מהנכונות של N :

- אם $x \in L_{pad}$ אז $y \in L_{pad}$ ולבן ריצה מקבלת של N .
- אם $x \notin L_{pad}$ אז כל הריצות של N דוחות (מסתיימות ב- q_r).



טענה 2: אם $L \in EXP$ אז $L_{pad} \in P$.

לפי הגדירה, קיימת מ"ט דטרמיניסטיבית M שרצה בזמן (n) q ומכריעה את L_{pad} . נבנה מ"ט דטרמיניסטיבית A המכריעה את L בזמן אקספוננציאלי. A על קלט x :

1. הרץ את M על $2^{2^{|x|^c}} 01^c x$ ועננה כמוה.

זמן ריצה: פולינומי באורך הקלט החדש $2^{|x|^c} + |x|$. לכתוב את $1^{2^{p(|x|)}} 0 (2^{p(|x|)})$ לוקח $O(2^{p(|x|)})$ זמן ריצה $|x|$ צעדים וסה"כ נקבל זמן ריצה $(2^{p(|x|)})^2$.

נכונות: ברורה מהנכונות של M :

- אם $x \in L$ אז $2^{2^{|x|^c}} 01^c x \in L_{pad}$ שכן M מקבל, ו- A מקבל את x .
- אם $x \notin L$ אז $2^{2^{|x|^c}} 01^c x \notin L_{pad}$ שכן M תדחה, ו- A תדחה את x .

סיה"כ קיבלנו כי $L \in EXP$.

הוכחת המשפט:

תהי $L \in NEXP$. מהתאזרה, $L \in NTime(2^{n^c})$, כלומר L עברו c בלשחו. מטענה 1, $L_{pad} \in NP$, ולכן מהתאזרה ש- $P = NP$ מתקיים כי $L \in EXP$.

בעיות ספייקות

בעיות ספייקות

הינו רצים להשתמש ב- ACC_{NP} בתור "עוגן" בשביל להראות תוצאות NP-שלמות נוספת, ממש כמו שהשתמשנו ב- ACC בשביל להראות תוצאות אי-כrüוות. הבעיה היא ש- ACC_{NP} מדברת על חישוב כללי, ולא ברור איך לעשות דоказיות ממנה לביעות טיפסיות. למשל על גרפים או על מספרים. החלופה שנמצאה היא בעית הספייקות.

נוסחה בولיאנית בצורה CNF (conjunctive normal form) היא מכילה **ליטרלים** (משתנה או שלילו של משתנה), **ופסוקיות** שמחילות עליהם פעולות "וגם", "או". נוסחת kCNF היא נוסחה שבכל פסוקית יש בדיק k ליטרלים.

נוסחת 3CNF: נוסחה $\phi(x_1, \dots, x_n)$ הינה מהצורה $\bigwedge_{i=1}^k c_i$ כאשר כל פסוקית c_i היא מהצורה (z_{i1}, z_{i2}, z_{i3}) – שלושה ליטרלים בלשיהם: $\{x_n, \bar{x}_n, \dots, x_1, \bar{x}_1\} \subseteq \{x_1, \dots, x_n\}$. בין הפסוקיות יש "וגם".

הערות:

- בניסוח הכללי של הבעיה לא אסרנו חזרות של ליטרלים ומשתנים.
- זהה מקרה פרטיו של kCNF, שהוא מקרה פרטי של CNF, שהוא מקרה פרטי של נוסחה (معالג עם fanout שהוא 1).

The formula $\phi = (\bar{x}_1 \wedge x_2) \vee (x_1 \wedge \bar{x}_3)$ is satisfied for example by **נוסחה ספייקת:** נוסחה בוליאנית (x_1, \dots, x_n) ϕ ספייקת אם קיימת השמה בוליאנית למשתנים, כך שערך הנוסחה הוא 1.

$$\begin{array}{rcl} x_1 & = & 0 \\ x_2 & = & 1 \\ x_3 & = & 0 \end{array}$$

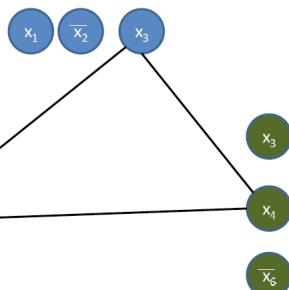
בעיית 3SAT: כל הנוסחאות מצורת 3CNF 3שהן ספייקות: $3SAT = \{\phi : \phi \text{ is satisfiable 3CNF formula}\}$

משפט קווק-לוי: $3SAT \in NPC$

טענה: $CLIQUE \in NPC$. בפרט $3SAT \leq_p CLIQUE$.

הוכחה:

<p>בבנה (ϕ): בהינתן נוסחת 3CNF נחדר קידוד $\langle G, k \rangle$ באשר:</p> <ul style="list-style-type: none"> • k הוא מספר הפסוקיות ב-ϕ. • לכל פסוקית של ϕ, נסיף שלישית צמתים בהתאם לגוף G, אחד לכל ליטרל. • זוג צמתים מחובר אמ: <ul style="list-style-type: none"> ◦ איןם שייכים לאותה שלישיה (פסוקיות). ◦ איןם שלולים זה את זה (בליטרלים). 	<p>פונקציה חסיבה</p>
<p>מתקיים כי $f(\phi) = \langle G, k \rangle \in CLIQUE \Leftrightarrow \phi \in 3SAT$:</p> <ul style="list-style-type: none"> • אם $\phi \in 3SAT$: נניח כי (x_1, \dots, x_n) מטופקת ע"י השמה $\{0,1\}^n$. אז לכל פסוקית ב-ϕ קיים ליטרל מסופק. נבחר אחד כזה בכל פסוקית, ונוסיף את הקודקוד המתאים ב-G לקבוצה S. בפרט $k = S$. כמו כן, לכל שני צמתים (שונים) $x, y \in S$ הם שייכים לשתי שלשות (פסוקיות) שונות. ◦ הליטרלים המתאימים לא שלולים זה את זה, שכן שניים מסופקים ע"י אותה השמה. ◦ לכן יש קשת (u, v) ב-G. <p>אם $\phi \notin 3SAT$: נוכיח באופן שקול:</p> <ul style="list-style-type: none"> • אם $\langle G, k \rangle \in CLIQUE$ אז $\phi \in 3SAT$. ◦ תהי S קליקה ב-G בגודל k. נשים לב כי כל שני צמתים ב-S בהכרח בשלישיות שונות, כי יש ביןיהם קשת. ◦ נובע כי S מכילה צומת אחד בדיק מכל שלישייה. לכל צומת v שזכה נתבון בליתרל המתאים $\{x_n, \bar{x}_n\}$ וניתן למשתנה המתאים x_n ערך 0 או 1 שיבתיich $= z_v$. ◦ בהשמה זו אין סתירות מכיוון שהליתרלים המתאים ל-S אינם שלילי זה של זה. קיבלנו השמה מספקת ל-ϕ (יתכן כי בהשמה הנ"ל גותרו משתנים חופשיים, ניתן להם השמה בלשיהם). 	<p>עובד ϕ:</p>



קוק-ליין

בעיתת ספיקות מוגלית: בהינתן קידוד של מעגל בוליאני C , וקלט חלקי x לבניות שלו נגדיר:
 $.CIRSAT = \{\langle C, x \rangle : C(xw) = 1\}$

כדי להוכיח את משפט קוק-ליין נוכיח:

$$\begin{aligned} 1. \quad & CIRSAT \in NPC \\ 2. \quad & CIRSAT \leq_P 3SAT \end{aligned}$$

כדי להוכיח את 1, נראה כי מ"ט הריצה בזמן (n) על קלטים בגודל t מוגדר בוגודל $\text{poly}(t)(n)$ המחשב את אותה פונקציה. תמיד נוכל לבתוב מעגל שמחשב אותו דבר עברו מ"ט נתונה.

лемה (הלב של קוק-ליין): $\text{תהי } t \leq n$ פונקציה חשיבה בזמן, ותהי M מ"ט הריצה בזמן (n) . t . קיימת פונקציה חשיבה בזמן (n) $\text{poly}(t)(n)$ שבהנתן n מחשבת (קידוד) מעגל $\{0, 1\}^n \rightarrow \{0, 1\}^n$ המקיים:

1. לכל קלט $z \in \{0, 1\}^n$: $C_{M,n}(z) = 1 \Leftrightarrow M(z) \text{ מקבלת } 1$.
2. חסם על גודל המעגל: $|C_{M,n}(n)| \leq O(t^2)$.

Proof sketch (similar to $\text{ACC}_{NP} \in \mathcal{NPC}$):

Let V be a polynomial-time verifier for L .

W.l.o.g: \exists polynomial $p(n)$, such that $V(x, w)$ accepts only if $|V(x, w)| = p(|x|)$.

The reduction maps $x \in \{0, 1\}^n \mapsto \langle C_{V,p(n)}, x \rangle$.

► $x \in L \iff \exists w : V(x, w) \text{ accepts} \iff \exists w : C_{V,p(n)}(x, w) = 1$

► Reduction is polynomial time

מסקנה 1: אם $\{0, 1\}^n \rightarrow \{0, 1\}^*$ אינה ניתנת לחישוב ע"י משפחת מעגלים בגודל (n) אז היא אינה ניתנת לחישוב על ידי מ"ט הריצה בזמן (n) , עברו $(n) \sqrt{s}$ חישיבה בזמן.

מסקנה 2: $CIRSAT \in NPC$.

הוכחת הלמה:

מעגל מעיל א"ב בילין: יהי Π א"ב, ו- $\Pi \rightarrow \Pi^k$ מוגדר באופן דומה למעגל בוליאני, אלא שהחוטים נשאים ערכיו Π (במקומות בויטים 0 או 1).

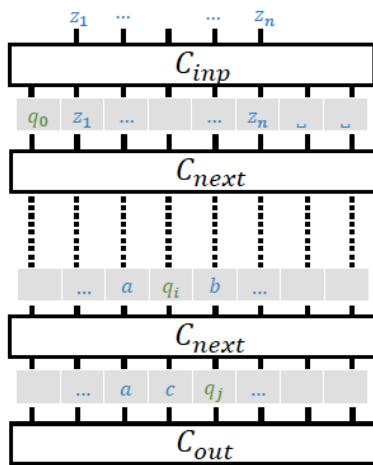
בנוסף, עברו מ"ט הריצה בזמן (n) וקלט z עבורו נגידיר את **מטריצת הקונפיגורציות** המכונה "טבלו" (tableau). נניח בה"כ כי פונקציית המעברים מוגדרת גם עברו מצב מקבל/דוחה ותמיד נשארת במצב המתאים.

טבלו: תהיו $(r_t) = \{0, 1\}, \Gamma, q_0, q_a, q_r = \{0, 1\}^n$ הריצה בזמן (n) . יהי $\Sigma = \{0, 1\}^n$ הטענו $\mathcal{T}_{M,z} = \Pi$ שבה לכל $i \in \{0, 1, \dots, t\}(n)$ השורה ה- i - במטריצה הינה הקונפיגורציה ה- i - בריצת (z) .

הרענון הוא לבנות מעגל Π שמחשב את הטבלו $\mathcal{T}_{M,z}$. בכל שבלה של חוטים במעגל תופיע קונפיגורציה, ואז הערכים שהחוטים הולכים לשאת הם הא"ב Π שמאפשר לתאר קונפיגורציות.

במעגל שנבנה כל תא $\{0, \dots, t^2\} \times \{j, i\}$ בטבלו יחוושב ע"י שער Π כלשהו $g_{i,j} \in B$ עברו בסיס מתאים B . מה נשאר לנו לעשות:

- נצלול לבב אחת מהשבותות, נגדיר את השערים $g_{i,j}$.
- בסופו של דבר נמיר את $C_{M,n}^\pi$ למעגל בוליאני $C_{M,n}$.



- C_{inp} given input z , outputs the initial configuration $q_0 z$.
- C_{next} given a configuration, outputs the next one (applied t times).
- C_{out} given a configuration, outputs 1 iff it is accepting.

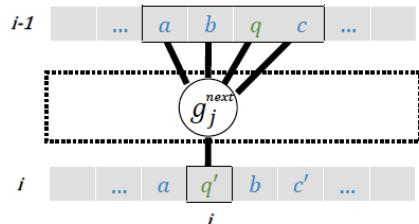


הمعالג: C_{inp}

3 סוגים של עירום : g_j^{inp}

- אם $0 = j$: הפונקציה הקבועה q_0 .
- עבור $n \leq j \leq 1$: פונקציית הזזה z_j (מעתיקת את הקלט).
- השאר: הפונקציה הקבועה \perp .

הمعالג: C_{next}



כל צעד חישוב הוא "локאלי", מעבר δ משנה רק את הקונפיגורציה באחור הראש: $\delta(q, c) = (q', c', L) \Leftrightarrow abqc \rightarrow aq'bc'$.

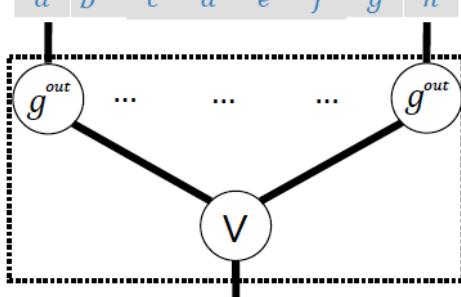
על כן, מרכיב משערם g_j^{next} אשר תלויים בכל היוטר 4 תאים בקונפיגורציה הקודמת $\{j-1, j, j+1, j+2\}$.

לכל $2 \leq j \leq t-1$ מתקיים:

$$g_j^{next}(s_1, s_2, s_3, s_4) = \begin{cases} \gamma & \text{if } s_1 \in Q \text{ and } \delta(s_1, s_2) = (q, \gamma, L) \\ q & \text{if } s_1 \in Q \text{ and } \delta(s_1, s_2) = (q, \gamma, R) \\ s_1 & \text{if } s_2 \in Q \text{ and } \delta(s_2, s_3) = (q, \gamma, L) \\ \gamma & \text{if } s_2 \in Q \text{ and } \delta(s_2, s_3) = (q, \gamma, R) \\ q & \text{if } s_3 \in Q \text{ and } \delta(s_3, s_4) = (q, \gamma, L) \\ s_2 & \text{otherwise} \end{cases}$$

עבור הקצאות $\in j \in \{0, t-1, t\}$ של הנ"ל.

הمعالג: C_{out}



מורכב משערם זהים g_j^{out} אשר בהינתן j מוחזירות 1 אם $s_j = q_a$ ו-0 אחרת. הפלט של המمعالג הוא OR על כל תוצאות השערם (ע"ז של שער בינהו V) – האם אחד מהמצבים הוא אכן מצב מקובל.

סה"כ קיבלנוمعالג:

$$C_{M,n}^\Pi = C_{out} \circ [C_{next} \circ \dots \circ C_{next}] \circ C_{inp}$$

הטענה הבאה נובעת ישירות מהבנייה: יהי $z \in \{0,1\}^n$ והקונפיגורציות בריצת $(z) M$. מתקיים:

- $C_{inp}(z) = \sigma_0$.
- לכל $1 \leq i \leq t$ מתקיים $\sigma_i = \sigma_{i-1}$.
- $C_{out}(\sigma_t) = 1 \Leftrightarrow \sigma_t$ קונפיגורציה מקבלת.

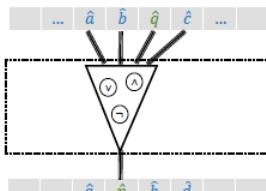
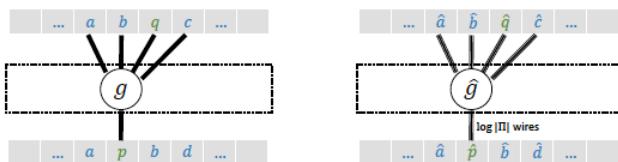
מסקנה: לכל $z \in \{0,1\}^n$ $M(z) = 1 \Leftrightarrow C_{M,n}^\Pi(z) = 1$

הערות:

- נשים לב כי כל אחד מהשערם g בהם השתמשנו היה פונקציה $\Pi \rightarrow \Pi^k$ עבור $4 \leq k$. בפרט המمعالג שבנו הוא מעלה הבסיס $\prod_{0 \leq k \leq 4} \Pi^k$.
- במונט ($t^2(n)$) $|C_{M,n}^\Pi| = O(t^2(n))$ והזמן הדרוש לחשב אותו הוא ($t(n)$) $poly(t(n))$.

על מנת להשלים את ההוכחה נשאר להמיר את המمعالג Π הזה למעגל בוליאני.

טענה: קיימת פונקציה חשיבה פולינומית שבහינתן מעגל Π מעלה B מחשבת מעגל בוליאני C מעלה בסיס דה-מורגן כך ש:



- לכל $z \in \{0,1\}^n$ מתקיים $C^\Pi(z) = 1 \Leftrightarrow C(z) = 1$.
- $|C| = \Theta(|C^\Pi|)$.



הוכחה: בהינתן מעגל C נמיר למעגל בוליани בשתי צעדים:

1. נחליף כל $\Pi \rightarrow \Pi^k$: $g: \{0,1\}^{\log(|\Pi|) \times k} \rightarrow \{0,1\}^{\log(|\Pi|)}$ בפונקציה בוליאנית (\hat{g}), שבහינתן קלט s_1, \dots, s_k מפרשת אותו בקדוד בינהרי של Π , מוחשבת (s_k, \dots, s_1 , $S = g(s_1, \dots, s_k)$ ומציאה את הקידוד שלו \hat{g} .
2. נממש את \hat{g} באמצעות מעגל בوليани מעלה דה-מורגן, נניח בגודל $O(2^{\log(|\Pi|) \times k}) = O(2^{\log(|\Pi|)})$.

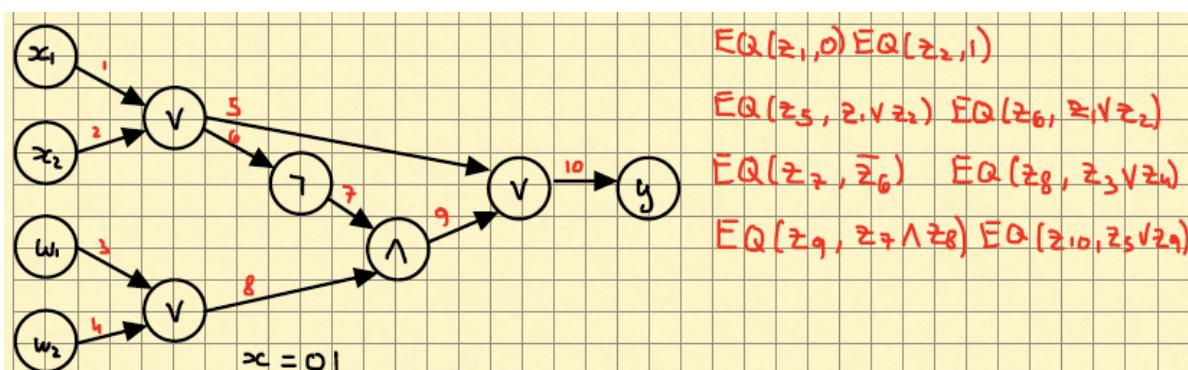
סימנו את הוכחת הלמה ממנה נובע כי $CIRSAT \in NPC$.

טענה: $CIRSAT \leq_P 3SAT$

הוכחה: בונה פונקציית מיפוי $(x, C) \mapsto f(x)$ שתחדיר נוסחת 3CNF ϕ שבה:

- לכל חוט α ב- C יהיה משתנה z_α ב- ϕ . נניח כי z_1, z_2, \dots, z_n החוטים המתאימים ל- α .
- נסיף פסוקיות המסתפקות אמ"מ:
 - לכל $1 \leq i \leq n$ מתקיים $z_i = x_i$.
 - משתנה חוט הפלט $1 = z_0$.
 - לכל חוט ℓ במעגל עם ערך (\neg, \wedge, \vee) $g(\alpha, \beta)$ עבור $\{g, \wedge, \vee\}$ וחותמים α, β מתקיים $EQ(y, z) = (y \vee z) = (\neg y \vee \neg z)$ שמוספקת אמ"מ $= y$. ב顺便 $EQ(a, b \wedge c), EQ(a, b \vee c), EQ(a, 1), EQ(a, 0)$.
- לבניית הפסוקיות הב"ל משתמש בנוסחה $(z \vee \neg z) = 1$ ב- ϕ . ב顺便 $EQ(a, b \wedge c), EQ(a, b \vee c), EQ(a, 1), EQ(a, 0)$.

המחשה: ניתן לעשות את זה כי האילוצים במעגל הנtentן הם לוקאליים ומעורבים בהם לכל היוטר 3 משתנים.



דоказיות נוספות

בעיית SS: מציאת תת-קובוצה שסכוםה t : $\{s_1, \dots, s_k, t \in \mathbb{N}: \exists I \subseteq [k]: \sum_{i \in I} s_i = t\}$

טענה: $SS \leq_P 3SAT$

הוכחה:

פונקציה	חשיבות																																																						
בונה (ϕ): בהינתן נוסחת 3CNF $\langle S, t \rangle$ כאשר נಡlik בית עבור כל ליטרל ופסוקית ב- ϕ . נניח כי יש לנו משתנים $\{x_1, \dots, x_\ell, \bar{x}_1, \dots, \bar{x}_\ell, \text{פסוקיות } \{c_1, \dots, c_k\}\}$.																																																							
$\phi = (x_1 \vee \bar{x}_2 \vee \dots) \wedge \dots \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \dots)$																																																							
קידוד משתנים: מושתנה i מקודד בשתי מספרים עם $k + \ell$ ספרות: X_i, \bar{X}_i .																																																							
מzd שמאלי – מדlik את הבית 1 עbor הספרה ה- i , השאר הם 0.																																																							
מzd ימינו, מדlik בית j אם הפסוקית c_j מכילה את המשתנה x_i .																																																							
קידוד פסוקיות: עbor פסוקית c_i בשני המספרים C_i, \bar{C}_i הספרה ה- $i + \ell$ תהיה 1, והשאר 0.																																																							
קידוד המטריה: קידוד t יעשה על ידי 1 ב- ℓ הספרות הראשונות, ו-3 ב- k הספרות האחרונות.																																																							
<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>1</td><td>...</td><td>ℓ</td><td>c_1</td><td>c_2</td><td>...</td><td>c_k</td></tr> <tr> <td>C_1</td><td>0</td><td>...</td><td>0</td><td>1</td><td>0</td><td>...</td><td>0</td></tr> <tr> <td>C'_1</td><td>0</td><td>...</td><td>0</td><td>1</td><td>0</td><td>...</td><td>0</td></tr> <tr> <td>C_2</td><td>0</td><td>...</td><td>0</td><td>0</td><td>1</td><td>...</td><td>0</td></tr> <tr> <td>C'_2</td><td>0</td><td>...</td><td>0</td><td>0</td><td>1</td><td>...</td><td>0</td></tr> </table> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>1</td><td>...</td><td>ℓ</td><td>c_1</td><td>c_2</td><td>...</td><td>c_k</td></tr> <tr> <td>t</td><td>1</td><td>...</td><td>1</td><td>3</td><td>3</td><td>...</td><td>3</td></tr> </table>	1	...	ℓ	c_1	c_2	...	c_k	C_1	0	...	0	1	0	...	0	C'_1	0	...	0	1	0	...	0	C_2	0	...	0	0	1	...	0	C'_2	0	...	0	0	1	...	0	1	...	ℓ	c_1	c_2	...	c_k	t	1	...	1	3	3	...	3	
1	...	ℓ	c_1	c_2	...	c_k																																																	
C_1	0	...	0	1	0	...	0																																																
C'_1	0	...	0	1	0	...	0																																																
C_2	0	...	0	0	1	...	0																																																
C'_2	0	...	0	0	1	...	0																																																
1	...	ℓ	c_1	c_2	...	c_k																																																	
t	1	...	1	3	3	...	3																																																



נכונות
(כיוון)
אחד

מתקיים כי f חסיבה, ומתקיים $f(\phi) = \langle S, t \rangle \in SS \Leftrightarrow \phi \in 3SAT$

עבור ϕ :

- אם $\phi \in 3SAT$: ניקח השמה מספקת ℓ - x_1, \dots, x_ℓ , ובניית תת-קובוצה שתתפרק את SS .

לכל $i \leq \ell$ אם $x_i = 1$ מוסף את X_i לקובוצה אחרת מוסף את \bar{X}_i .

ויהי t' סכום המספרים שלקחנו עד כה.

לכל $i < \ell$ אם $x'_i = 1$ מתקיים $t'_{\ell+i} \leq i \leq t'_{\ell+1} \leq 1$, מכיוון שבל פסוקית מכילה

ליטרל מסווק אחד לפחות ולכל היוטר שלושה. זה יהיה מספר המטריה שלנו.

בעת נשלים בעזרת המספרים $\{C_i\}$ בנדראש.

	C_1	C_2
x_1	1	2
\bar{x}_1	1	
x_2		1
\bar{x}_2	1	
x_3		1
\bar{x}_3	1	
x_4		1
\bar{x}_4	1	
C_1		
C_2		
C_3		
C_4		
t	1	1

	1	2	...	ℓ	c_1	c_2	...	c_k
x_1	1	0	...	0	1	0	...	0
\bar{x}_1	1	0	...	0	0	0	...	1
x_2	0	1	...	0	0	0	...	0
\bar{x}_2	0	1	...	0	1	0	...	1
\vdots								
C_1	0	0	...	0	1	0	...	0
C'_1	0	0	...	0	1	0	...	0
C_2	0	0	...	0	0	1	...	0
C'_2	0	0	...	0	0	1	...	0
\vdots								
t	1	1	...	1	3	3	...	3

$$\text{דוגמה: } \phi = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_2 \vee \bar{x}_3 \vee x_4)$$

בנייה (ϕ): בהינתן בסחתת 3CNF נחדר קידוד $\langle G, s, t \rangle$. נניח כי יש לנו משתנים $\{\bar{x}_\ell, x_1, \dots, x_\ell\}$, פסוקיות $\{c_1, \dots, c_k\}$.

לכל פסוקית c_i מוסף צומת.

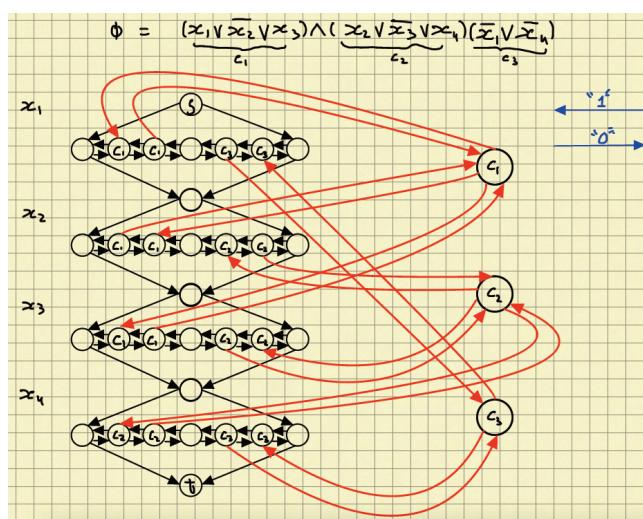
לכל משתנה x_i מוסף יהלום (כמו בדוגמה).

בשבבה האמצעית של היהלום, זוג צמתים לכל פסוקית בה המשתנה נוכח ו"צומת מפheid" בין כל זוג.

אם $c_j \in x_i$ מוסף מעקב בכוון שמאלת דרכ c_j .

אם $c_j \in \bar{x}_i$ מוסף מעקב בכוון ימינה דרכ c_j .

חסיבה
פונקציה



מתקיים כי f חסיבה, ומתקיים $f(\phi) = \langle G, s, t \rangle \in 3SAT \Leftrightarrow \phi \in 3SAT$

אם $\phi \in 3SAT$: ניקח השמה מספקת x_1, \dots, x_ℓ .

נתחיל מסלול שמכסה את היהלומים: אם $x_i = 1$ נלק שמאלה בשכבת האמצעית, אחרת ימינה.

לכל פסוקית, נבחר משתנה שמספק אותה x_j ונרחיב את המסלול בהילום כך שייעבור דרך c_j , ביוון המעקב נבחר באופן קונסיסטנטי עם סיפוק c_j (בכוון שמאלת או ימינה).

נכונות
(כיוון)
אחד



תרגול 10 (רדווקציות)

תרגיל 1: בבעית הסוכן הנוסף, נתונה לנו מפה M: רשיימה של ערים, זמן נסעה בעיר לעיר. נתון מספר K. האם ניתן לעבור בכל עיר פעם אחת (אם קיים מעגל המילוטני) ושרמהק הנסעה הוא לפחות K. הוכיחו שהבעיה ב-NPC.

נגידור $\{ \langle M, K \rangle : \text{there is a ham path in } M \text{ in length } \leq K \}$

נראה כי $NP \in TS$: או להראות מודוא עברו השפה, או מטלי"ד פולינומית. באן נגידור מודוא V_{TS} על קלט $(\langle M, K \rangle, c)$:

1. נפרש את הרمز $c_r, \dots, c_1 = c$ בטור מסלול של ערים.
2. נבדוק ש- c היא כמות הערים במפה M, וגם בולן שנות זו מדו.
3. נסכום את המרחקים (c_i, c_{i+1}) ב-M ונקבל \Leftrightarrow הסכום קטן שווה מ-K.

המודוא פולינומי – כל הפעולות פשוטות (פרסור הקלט וסיריקה שלו). נראה כי V_{TS} הוא מודוא עברו TS:

- אם $\langle M, K \rangle \in TS$ אז קיים מסלול c שעובר בכל הערים במרחב שקטן שווה מ-K, ולכן $\langle M, K \rangle, c \in V_{TS}$ מתקבל.
- אם $\langle M, K \rangle \notin TS$ לא קיים כזה מסלול, ולכן $\langle M, K \rangle, c \notin V_{TS}$.

נראה כי $TS \in NPC$: נראה דיווקה משפה שאנו יודעים שהיא ב-NPC. נראה $TS \leq_p HAM$:

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">בבנה (f): בהינתן נוסחת גרפ G נחזר קידוד $\langle M_G, k \rangle$.</td><td style="padding: 5px; vertical-align: top;"> פונקציה חשיבה </td></tr> </table>	בבנה (f): בהינתן נוסחת גרפ G נחזר קידוד $\langle M_G, k \rangle$.	פונקציה חשיבה	
בבנה (f): בהינתן נוסחת גרפ G נחזר קידוד $\langle M_G, k \rangle$.	פונקציה חשיבה		
<ul style="list-style-type: none"> • אם מה שקיבלוינו אינם גרפ, נחזר קידוד שאינו חוקי. • נשייך עיר לכל צומת בגרף G. • אם קשת $E \in \langle n, u \rangle$ אז $M_G(u, v) = 1$. • אחרת עבור $E \notin \langle n, u \rangle$ אז $M_G(u, v) = 2$. • $k = n - 1$. 			
<p>מתקיים כי f חישבה בזמן פולינומי, ומתקיים $M \in HAM \Leftrightarrow G \in TS \Leftrightarrow f(G) = \langle M_G, k \rangle$:</p> <ul style="list-style-type: none"> • אם $G \in HAM$: קיים מסלול המילוטני ב-G, אותו המסלול הוא מסלול באורך בדיק 1 – בין הערים במפה. • אם $G \notin HAM$: נראה באופן שקול, כי אם $S \in TS$ אז $\langle M_S, n - 1 \rangle \in HAM$. קיים מסלול באורך בדיק 1 – n ובנוביה קיים מסלול ב-G-באורך בדיק 1 – n (כי המרחקים הם בגודל 1 כל אחד). לא ניתן צעד במשקל 5 ולכן אותו המסלול הוא המילוטני ב-G. 	נוכחות (כיוון אחד)		

תרגיל 2: עברו נוסחת AND של פסוקיות OR והשמה v נסמן את $\langle \varphi, v \rangle$ את מספר הפסוקיות ב- φ ש- v מספקת. נגידור:

$$CCNF = \{ \langle \varphi, k \rangle : \varphi \text{ is a CNF and } \exists v. N(\varphi, v) = k \}$$

כלומר קיימת השמה שמספקת בדיק k פסוקיות.

1. **נוכיח $CCNF \leq_p SAT$:** אנחנו יודעים כי $SAT \in NPC$, לכן מההגדרה לכל $NP \in NP$ מתקיים $L \in L$ מתקיים $SAT \leq_p L$. מספיק להראות רק כי $CCNF \in NP$. נגידור מטלי"ד B על קלט $\langle \varphi, v \rangle$:
 - a. נחש השמה a (לכמota המשתנים של φ).
 - b. חשב את $N(\varphi, a) = k \Leftrightarrow N(\varphi, a) = k$.
2. **נוכיח $SAT \leq_p CCNF$:** בבנה (f): בהינתן נוסחה נחזר קידוד $\langle \varphi, m(\varphi) \rangle$. כאשר $m(\varphi) = m$ הוא מספר הפסוקיות ב- φ .
 - a. אם $\varphi \in SAT$ אז קיימת השמה שמספקת את כל הפסוקיות, $\exists v. N(\varphi, v) = m(\varphi) = m$.
 - b. אם $\varphi, m(\varphi) \in CCNF$: אנחנו הגדרנו את התמונה, ובהכרח קיימת השמה מספקת.
3. **נוכיח $CCNF \leq_p CDNF$:** בבנה (f): בהינתן נוסחה ומספר הפסוקיות המסופקות שבה a , נחזר קידוד $\langle \varphi | \bar{\varphi} \rangle$ מהשהסתפק ב- $CDNF$ לא יסתפק ב- DNF ולהיפך, כי אנחנו משתמשים בדה-מורגן והכל מתהפן. פולינומית: כי ניתן לחשב את $\bar{\varphi}$ בתורו DNF בזמןיעיל עם דה-מורגן.

$$\langle \varphi, k \rangle \in CCNF \Leftrightarrow \exists v. N(\varphi, v) = k \Leftrightarrow \exists v. N(\bar{\varphi}, v) = k \Leftrightarrow \langle \varphi | \bar{\varphi} \rangle \in CDNF$$



תרגיל 3: בבנייה הרדוקטיה אנחנו דואגים שתמיד יש IS בגודל k (התנאי השני), ואז נראה קל לנתח מה קורה לגבי התנאי הראשון. כפונקציה של האם בקלט כבר היה CLIQUE בגודל k או לא.

הוכחו כי:

$$\text{IS} \wedge \text{CLIQUE} = \{\langle G, k \rangle \mid G \text{ has an IS and a clique of size } k\} \in \mathcal{NPC}$$

פתרון

נראה כי $\text{IS} \wedge \text{CLIQUE} \in \mathcal{NP}$ ושהיא \mathcal{NP} -קשה.

- קל לראות כי $\text{IS} \wedge \text{CLIQUE} \in \mathcal{NP}$ ע"י מודא פולינומי על קלט $\langle G, k \rangle$, הוא יודא כי A מקודד שתי קבוצות בגודל k של הצמתים של G , שהראשונה מהויה קבוצה ב"ת (כלומר, אין קשר בין אף זוג צמתים) והשנייה מהויה קליק (כלומר, בין כל זוג צמתים יש קשר). ברור כי ניתן לעשות זאת בזמן פולינומי, שהקלט בשפה אם"ס קיים עד כזה.
- כעת, כדי להראות שהיא קשה, נראה $\text{CLIQUE} \leq_p \text{IS} \wedge \text{CLIQUE}$ (ואז, לפי המשפט הקודם, זה מספיק). הרדוקטיה תהיה $f(\langle G', k \rangle) = \langle G, k \rangle$ (בתוספת k צמתים מבודדים. ואז:
 - ברור כי f פולינומית (צריך בסה"כ להוסיף עוד k צמתים לגרף הנתון).
 - אם ל- G יש קליק בגודל k אז ל- G' יש קליק בגודל k (לא "הרסנו" אותו) וגם קבוצה ב"ת בגודל k ולכן $\langle G', k \rangle \in \text{IS} \wedge \text{CLIQUE}$
 - אם ל- G אין קליק בגודל k אז גם ל- G' אין (הצמתים המבודדים לא יכולים להשתחף בקליק) ולכן $\langle G', k \rangle \notin \text{IS} \wedge \text{CLIQUE}$.