חישוביות וסיבוכיות

תוכן העניינים

שיעור 1
מכונות טיורינג

1.1 הגדרה של מכונת טיורינג

הגדרה 1.1 מכונת טיורינג (הגדרה היוריסטית)

הקלט והסרט

- מכונת טיורינג (מ"ט) קורא קלט.
- הקלט עצמו נמצא על סרט אינסופי מחולק למשבצות.
- כל תו של הקלט כתוב במשבצת אחת של הסרט.
- במכונת טיורינג אנחנו מניחים שהסרט אינסופי לשני הכיוונים.
  - משמאל לתחילת הקלט יש רצף אינסופי של תווי רווח "␣".
  - מימין לסוף הקלט יש רצף אינסופי של תווי רווח "␣".

$\cdots$	␣	␣	a	b	b	b	a	a	␣	␣	␣	$\cdots$

הראש

- במצב ההתחלתי הראש בקצה השמאלי של הקלט.

$\cdots$	␣	␣	a	b	b	b	a	a	␣	␣	␣	$\cdots$
			↑									

- הראש יכול לזוז ימינה על הסרט וגם שמאלה על הסרט.
- הראש קורא את התוכן של המשבצת שבה הוא נמצא.
- הראש יכול לכתוב על משבצת, אבל רק על המשבצת שבה הראש נמצא.

תאור העבודה של המכונה

- בתחילת הריצה, הקלט כתוב התחילת הסרט כאשר מימינו נמצא רצף אינסופי של תווי ␣ -ים.
- הראש מצביע על התא הראשון בסרט והמכונה נמצאת במצב התחלתי q_0.

q_0	$\cdots$	␣	␣	a	b	b	b	a	a	␣	␣	␣	$\cdots$
				↑									

- בכל צעד חישוב, בהתאם למצב הנוכחי ולאות שמתחת לראש (התו הנקרא), המכונה מחליטה:
  - לאיזה מצב לעבור
  - מה לכתוב מתחת לראש (התו הנכתב)
  - לאן להזיז את הראש (תא אחד ימינה, או תא אחד שמאלה, או להישאר במקום).
- למכונה ישנם שני מצבים מיוחדים:
  - q_{acc}: אם במשך הריצה המכונה מגיעה ל- q_{acc} היא עוברת ומקבלת.
  - q_{rej}: אם במשך הריצה המכונה מגיעה ל- q_{rej} היא עוברת ודוחה.
- אם המכונה לא מגיעה ל- q_{acc} או q_{rej} היא תמשיך לרוץ לנצח.

הגדרה 1.2 מכונת טיורינג

מכונת טיורינג (מ"ט) היא שביעיה

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}})$$

כאשר:

Q	קבוצת מצבים סופית ולא ריקה	
Σ	אלפבית הקלט	$\_ \notin \Sigma$
Γ	אלפבית הסרט	$\Sigma \subseteq \Gamma\,,\ \_ \in \Gamma$
δ	פונקצית המעברים	$\delta : (Q \backslash \{q_{\text{rej}}, q_{\text{acc}}\} \times \Gamma \to Q \times \Gamma \times \{L, R\}$
q_0	מצב התחלתי	
q_{acc}	מצב מקבל יחיד	
q_{rej}	מצב דוחה יחיד	

דוגמה 1.1

נבנה מכונת טיורינג אשר מקבלת מילה אם היא בשפה

$$L = \left\{ w \in \{a, b\}^* \middle| \#a_w = \#b_w \right\} \ .$$

ז"א השפת כל המילים עם מספר שווה אותיות a ו b.

הרעיון של האלגוריתם של המכונה היא כדלקמן:

- נסרוק את הקלט משמאל לימין, נחפש את האות a הראשונה, נסמן אותה איכשהו כ"נקראת".
- אחר כך נחפש b תואם.
  - אם מצאנו b תואם נסמן אותו כ"נקרא", נחזור לתחילת הקלט ונתחיל סיבוב חדש.
  - אם לא מצאנו b תואם אז המכונה תדחה.
- אם נגיע לסיבוב שבו אינן נשארות אף אותיות לא a:
  - אם יש b לא מסומן אז המכונה תדחה.
  - אחרת אם לא נשאר אף b לא מסומן אז המכונה תקבל.

כעת נתאר את הפאודו-קוד של המכונה, כדלקמן.

פסאודו-קוד

1) סורקים את הקלט משמאול לימין.
- אם לא מצאנו a וגם לא מצאנו b $\Leftarrow$ מקבלת.
- אם האות הראשונה שהראש מצא היא a, כותבים עליו $\checkmark$, חוזרים לתחילת הקלט ועוברים לשלב 2).
- אם האות הראשונה שהראש מצא היא b, כותבים עליו $\checkmark$, חוזרים לתחילת הקלט ועוברים לשלב 3).

2) סורקים את הקלט משמאול לימין.
- אם לא מצאנו b $\Leftarrow$ דוחה.
- אם מצאנו b כותבים עליו $\checkmark$, חוזרים לתחילת הקלט וחוזרים לשלב 1).

3) סורקים את הקלט משמאול לימין.
- אם לא מצאנו a $\Leftarrow$ דוחה.
- אם מצאנו a כותבים עליו $\checkmark$, חוזרים לתחילת הקלט וחוזרים לשלב 1).

כעת נתן הגדרה פורמלית של המכונת טיורינג שמבצעת את האלגוריתם הזה.

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}})$$

כאשר Q הקבוצת המצבנים הבאה:

$$Q = \{q_0, q_a, q_b, q_{\text{back}}, q_{\text{rej}}, q_{\text{acc}}\} \ .$$

המשמעותם של כל המצבים נרשמים בטבלה למטה:

q_0	המצב ההתחלתי. אליו נחזור אחרי כל סבב התאמה של זוג אותיות.
q_a	מצב שבו ראינו a ומחפשים b תואם.
q_b	מצב שבו ראינו b ומחפשים a תואם.
q_{back}	מצב שנשתמש בו כדי לחזור לקצה השמאלי של הקלט ולהתחיל את הסריקה הבאה (סבב ההתאמה הבא).
q_{acc}	מצב מקבל.
q_{rej}	מצב דוחה.

האלפבית של הקלט, Σ, והלפבית של הסרט, Γ, הינן:

$$\Sigma = \{\mathtt{a},\mathtt{b}\} \ , \qquad \Gamma = \{\mathtt{a},\mathtt{b},\sqcup,\checkmark\} \ .$$

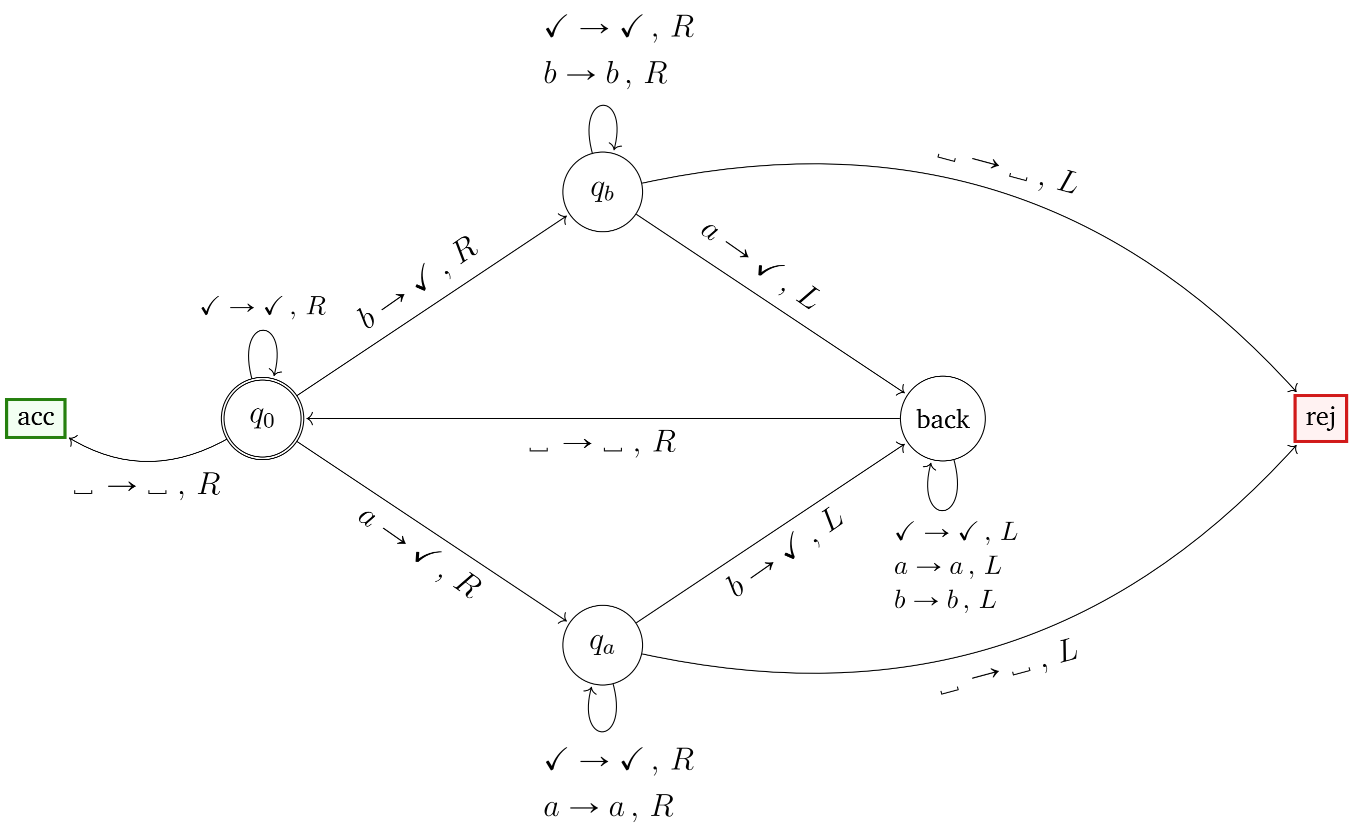
הפונקצית המעברים $\delta : Q \times \Sigma \to Q \times \Gamma \times \{L, R\}$ היא מוגדרת כדלקמן.

$$\begin{aligned}
\delta(q_0, \mathtt{a}) &= (q_a, \checkmark, R) \ , \\
\delta(q_0, \mathtt{b}) &= (q_b, \checkmark, R) \ , \\
\delta(q_0, \sqcup) &= (q_{\text{acc}}, \sqcup, R) \ , \\
\delta(q_a, \checkmark) &= (q_a, \checkmark, R) \ , \\
\delta(q_a, \mathtt{a}) &= (q_a, \mathtt{a}, R) \ , \\
\delta(q_a, \mathtt{b}) &= (\text{back}, \checkmark, L) \ , \\
\delta(q_b, \checkmark) &= (q_b, \checkmark, R) \ , \\
\delta(q_b, \mathtt{b}) &= (q_a, \mathtt{b}, R) \ , \\
\delta(q_b, \mathtt{a}) &= (\text{back}, \checkmark, L) \ .
\end{aligned}$$

לעתים קל יותר לרשום את פונקציית המעבירים δ כטבלה:

$Q \backslash \Gamma$	a	b	$\sqcup$	$\checkmark$
q_0	$(q_a, \checkmark, R)$	$(q_b, \checkmark, R)$	$(q_{\text{acc}}, \sqcup, R)$	$(q_0, \checkmark, R)$
q_a	$(q_a, \mathtt{a}, R)$	$(\text{back}, \checkmark, L)$	$(q_{\text{rej}}, \sqcup, L)$	$(q_a, \checkmark, R)$
q_b	$(\text{back}, \checkmark, L)$	$(q_b, \mathtt{b}, R)$	$(q_{\text{rej}}, \sqcup, L)$	$(q_b, \checkmark, R)$
back	$(\text{back}, \mathtt{a}, L)$	$(\text{back}, \mathtt{b}, L)$	$(q_0, \sqcup, R)$	$(\text{back}, \checkmark, L)$

תרשים מצבים



- בכל צעד המכונה מבצעת שתי פעולות:
  1. כותבת אות במיקום הראש
  2. זזה צעד אחד שמאלה או צעד אחד ימינה.
- בכל צעד המכונה יכולה לעבור למצב אחר או להישאר באותו מצב.

דוגמה 1.2

בדקו אם המכונת טיורינג של הדוגמה 1.1 מקבלת את המילה abbbaa.

פתרון:

$\sqcup$	q_0	a	b	b	b	a	a	$\sqcup$
$\sqcup$	$\checkmark$	q_a	b	b	b	a	a	$\sqcup$
$\sqcup$	back	$\checkmark$	$\checkmark$	b	b	a	a	$\sqcup$
back	$\sqcup$	$\checkmark$	$\checkmark$	b	b	a	a	$\sqcup$
$\sqcup$	q_0	$\checkmark$	$\checkmark$	b	b	a	a	$\sqcup$
$\sqcup$	$\checkmark$	q_0	$\checkmark$	b	b	a	a	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	q_0	b	b	a	a	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	q_b	b	a	a	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	b	q_b	a	a	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	back	b	$\checkmark$	a	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	back	$\checkmark$	b	$\checkmark$	a	$\sqcup$
$\sqcup$	$\checkmark$	back	$\checkmark$	$\checkmark$	b	$\checkmark$	a	$\sqcup$

$\sqcup$	back	$\checkmark$	$\checkmark$	$\checkmark$	b	$\checkmark$	a	$\sqcup$
back	$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	b	$\checkmark$	a	$\sqcup$
$\sqcup$	q_0	$\checkmark$	$\checkmark$	$\checkmark$	b	$\checkmark$	a	$\sqcup$
$\sqcup$	$\checkmark$	q_0	$\checkmark$	$\checkmark$	b	$\checkmark$	a	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	q_0	$\checkmark$	b	$\checkmark$	a	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	q_0	b	$\checkmark$	a	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	q_b	$\checkmark$	a	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	q_b	a	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	back	$\checkmark$	$\checkmark$	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	back	$\checkmark$	$\checkmark$	$\checkmark$	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	back	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\sqcup$
$\sqcup$	$\checkmark$	back	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\sqcup$
$\sqcup$	back	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\sqcup$
back	$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\sqcup$
$\sqcup$	q_0	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\sqcup$
$\sqcup$	$\checkmark$	q_0	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	q_0	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	q_0	$\checkmark$	$\checkmark$	$\checkmark$	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	q_0	$\checkmark$	$\checkmark$	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	q_0	$\checkmark$	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	q_0	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\sqcup$	acc

דוגמה 1.3

בדקו אם המכונת טיורינג של הדוגמה 1.1 מקבלת את המילה aab.

פתרון:

$\sqcup$	q_0	a	a	b	$\sqcup$
$\sqcup$	$\checkmark$	q_a	a	b	$\sqcup$
$\sqcup$	$\checkmark$	a	q_a	b	$\sqcup$
$\sqcup$	$\checkmark$	back	a	$\checkmark$	$\sqcup$
$\sqcup$	back	$\checkmark$	a	$\checkmark$	$\sqcup$
back	$\sqcup$	$\checkmark$	a	$\checkmark$	$\sqcup$
$\sqcup$	q_0	$\checkmark$	a	$\checkmark$	$\sqcup$
$\sqcup$	$\checkmark$	q_0	a	$\checkmark$	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	q_a	$\checkmark$	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	q_a	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	rej	$\checkmark$	$\sqcup$

הגדרה 1.3 קונפיגורציה

תהי $M = (Q, q_0, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}})$ מכונת טיורינג. **קונפיגורציה** של M הינה מחרוזת

$$u q \sigma \mathrm{v}$$

כאשר משמעות:

$$u,\mathrm{v}\in\Gamma^*\ ,\qquad \sigma\in\Gamma\ ,\qquad q\in Q\ .$$

q מצב המכונה,
σ הסימון במיקום הראש
u תוכן הסרט משמאל לראש,
v תוכן הסרט מימין לראש.

דוגמה 1.4 (המשך של דוגמה 1.3)

u	q	σ	v
$\sqcup$	q_0	a	a b $\sqcup$
$\sqcup\checkmark$	q_a	a	b $\sqcup$
$\sqcup$ $\checkmark$ a	q_a	b	$\sqcup$
$\sqcup$ $\checkmark$	back	a	$\checkmark$ $\sqcup$
$\sqcup$	back	$\checkmark$	a $\checkmark$ $\sqcup$
$\sqcup$	back	$\sqcup$	$\checkmark$ a $\checkmark$ $\sqcup$
$\sqcup$	q_0	$\checkmark$	a $\checkmark$ $\sqcup$
$\sqcup$ $\checkmark$	q_0	a	$\checkmark$ $\sqcup$
$\sqcup$ $\checkmark$ $\checkmark$	q_a	$\checkmark$	$\sqcup$
$\sqcup$ $\checkmark$ $\checkmark$ $\checkmark$	q_a	$\sqcup$	$\sqcup$
$\sqcup$ $\checkmark$ $\checkmark$	rej	$\checkmark$	$\sqcup$

דוגמה 1.5

בנו מכונת טיורינג אשר מקבלת כל מילה בשפה

$$L = \{a^n \,|\, n = 2^k \;,\; k \in \mathbb{N}\}$$

ז"א מילים בעלי מספר אותיות a אשר חזקה של 2.

פתרון:

ראשית נשים לב למשפט הבא:

משפט 1.1

מספר שלם n שווה לחזקה אי-שלילית של 2, כלומר $(k \geqslant 0)$ $n = 2^k$ אם ורק אם קיים שלם m עבורו חילוק של n ב- 2 בדיוק m פעמים נותן 1.

הוכחה:

<u>כיוון $\Leftarrow$</u>

אם $n = 2^k$ ו- $k \geqslant 0$ אז $\frac{n}{2^k} = 1$.

<u>כיוון $\Rightarrow$</u>

אם קיים $m \geqslant 0$ עבורו $\frac{n}{2^m} = 1$ אז $n = 2^m$ ולכן n שווה לחזקה אי-שלילית של 2. ■

לאור המשפט הזה נבנה אלגוריתם אשר מחלק את מספר האותיות במילה ב- 2 בצורה איטרטיבית. אם אחרי סבב מסויים נקבל מספר אי-זוגי גדול מ- 1 אז מספר האותיות a במילה לא יכול להיות חזקה של 2. אם אחרי כל הסבבים לא קיבלנו מספר אי-זוגי גדול מ-1 אז מובטח לנו שיש מספר אותיות a אשר חזקה של 2.

- נתון הקלט

$\sqcup$	$\sqcup$	a	a	a	a	a	a	a	a	$\sqcup$	$\sqcup$
		$\uparrow$									

נעבור על סרט הקלט. משמאל לימין.

- מבצעים מחקיה לסירוגין של האות a כלומר אות אחת נמחק ואות אחת נשאיר וכן הלאה.

$\sqcup$	$\sqcup$	✓	a	✓	a	✓	a	✓	a	$\sqcup$	$\sqcup$
									↑		

אם אחרי סבב הראשון

* יש ✓ בתו האחרון $\Leftarrow$ קיבלנו מספר אי-זוגי של אותיות a אחרי חילוק ב- 2 $\Leftarrow$ אין חזקה של 2 אותיות a במילה.
* יש a בתו האחרון $\Leftarrow$ קיבלנו מספר זוגי של אותיות a אחרי חילוק ב- 2 ונמשיך לסבב הבא.

- הראש חוזר לתו הראשון של הקלט

$\sqcup$	$\sqcup$	✓	a	✓	a	✓	a	✓	a	$\sqcup$	$\sqcup$
		↑									

- בסבב הבא חוזרים על התהליך של מחיקה לסירוגין של האות a (אות אחת נמחק ואות אחת נשאיר)

$\sqcup$	$\sqcup$	✓	✓	✓	a	✓	✓	✓	a	$\sqcup$	$\sqcup$
									↑		

אם אחרי סבב השני

* יש ✓ בתו האחרון $\Leftarrow$ קיבלנו מספר אי-זוגי של אותיות a אחרי חילוק ב- 2 $\Leftarrow$ אין חזקה של 2 אותיות a במילה.
* יש a בתו האחרון $\Leftarrow$ קיבלנו מספר זוגי של אותיות a אחרי חילוק ב- 2 ונמשיך לסבב הבא.

- הראש חוזר לתו הראשון של הקלט

$\sqcup$	$\sqcup$	✓	✓	✓	a	✓	✓	✓	a	$\sqcup$	$\sqcup$
		↑									

- בסבב הבא חוזרים על התהליך של מחיקה לסירוגין של האות a (אות אחת נמחק ואות אחת נשאיר)

$\sqcup$	$\sqcup$	✓	✓	✓	✓	✓	✓	✓	a	$\sqcup$	$\sqcup$
									↑		

אם אחרי סבב השלישי

* יש ✓ בתו האחרון $\Leftarrow$ קיבלנו מספר אי-זוגי של אותיות a אחרי חילוק ב- 2 $\Leftarrow$ אין חזקה של 2 אותיות a במילה.
* יש a בתו האחרון $\Leftarrow$ קיבלנו מספר זוגי של אותיות a אחרי חילוק ב- 2 ונמשיך לסבב הבא.

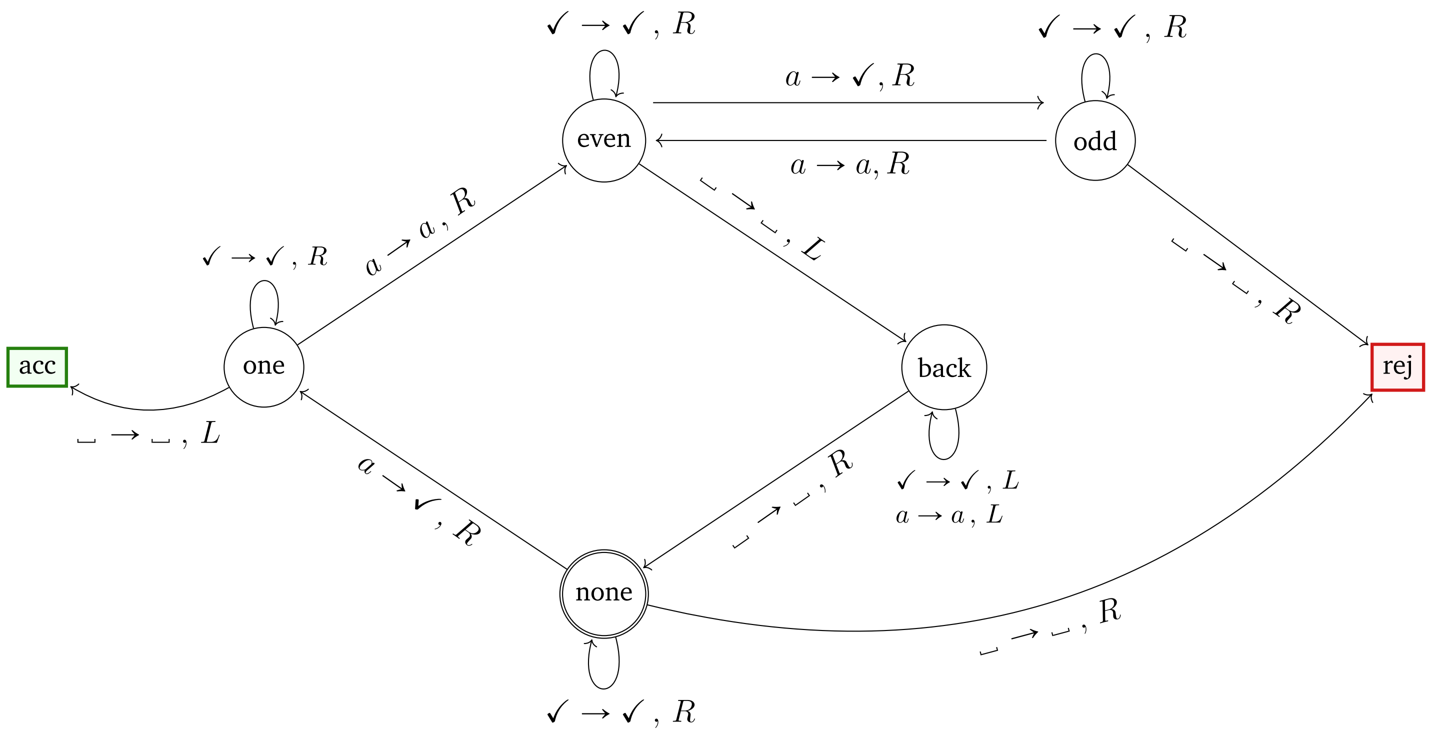
- הראש חוזר לתו הראשון של הקלט.

בסבב האחרון נשאר רק אות a אחת.

לכן לפי המשפט למעלה מובטח לנו כי המילה מורכבת ממספר אותיות a אשר חזקה של 2.

$\sqcup$	$\sqcup$	✓	✓	✓	✓	✓	✓	✓	a	$\sqcup$	$\sqcup$
		↑									

המכונת טיורינג אשר מקבלת מילים בשפה שעובדת לפי האלגוריתם המתואר למעלה מתואר בתרשים למטה.



המצבים:

מצב none: מצב התחלתי. עדיין לא קראנו a בסבב סריקה זה.

מצב one: קראנו a בודד.

מצב even: קראנו מספר זוגי של a .

מצב odd: קראנו מספר אי-זוגי של a .

מצב back: חזרה שלמאלה.

דוגמה 1.6

בדקו אם המילה

aaaa

מתקבלת על ידי המכונת טיורינג בדוגמה 1.5.

פתרון:

␣	none	a	a	a	a	␣
␣	✓	one	a	a	a	␣
␣	✓	a	even	a	a	␣
␣	✓	a	✓	odd	a	␣
␣	✓	a	✓	a	even	␣
␣	✓	a	✓	back	a	␣
␣	✓	a	back	✓	a	␣
␣	✓	back	a	✓	a	␣
␣	back	✓	a	✓	a	␣
back	␣	✓	a	✓	a	␣

␣	none	✓	a	✓	a	␣
␣	✓	none	a	✓	a	␣
␣	✓	✓	one	✓	a	␣
␣	✓	✓	✓	one	a	␣
␣	✓	✓	✓	a	even	␣
␣	✓	✓	✓	back	a	␣
␣	✓	✓	back	✓	a	␣
␣	✓	back	✓	✓	a	␣
␣	back	✓	✓	✓	a	␣
back	␣	✓	✓	✓	a	␣
␣	none	✓	✓	✓	a	␣
␣	✓	none	✓	✓	a	␣
␣	✓	✓	none	✓	a	␣
␣	✓	✓	✓	none	a	␣
␣	✓	✓	✓	✓	one	␣
␣	✓	✓	✓	acc	✓	␣

u	q	σ	v
␣	none	a	a a a ␣
␣ ✓	one	a	a a ␣
␣ ✓ a	even	a	a ␣
␣ ✓ a ✓	odd	a	␣
␣ ✓ a ✓ a	even	␣	␣
␣ ✓ a ✓	back	a	␣
␣ ✓ a	back	✓	a ␣
␣ ✓	back	a	✓ a ␣
␣	back	✓	a ✓ a ␣
␣	back	␣	✓ a ✓ a ␣
␣	none	✓	a ✓ a ␣
␣✓	none	a	✓ a ␣
␣✓✓	one	✓	a ␣
␣✓✓✓	one	a	␣
␣✓✓✓ a	even	␣	␣
␣✓✓✓	back	a	␣
␣✓✓	back	✓ a	␣
␣✓	back	✓	✓ a ␣
␣	back	✓	✓✓ a ␣
␣	back	␣	✓✓✓ a ␣
␣	none	✓	✓✓ a ␣
␣✓	none	✓	✓ a ␣
␣✓✓	none	✓	a ␣
␣✓✓✓	none	a	␣
␣✓✓✓✓	one	␣	␣
␣✓✓✓	acc	✓	␣

דוגמה 1.7

בדקו אם המילה

aaa

מתקבלת על ידי המכונת טיורינג בדוגמה 1.5.

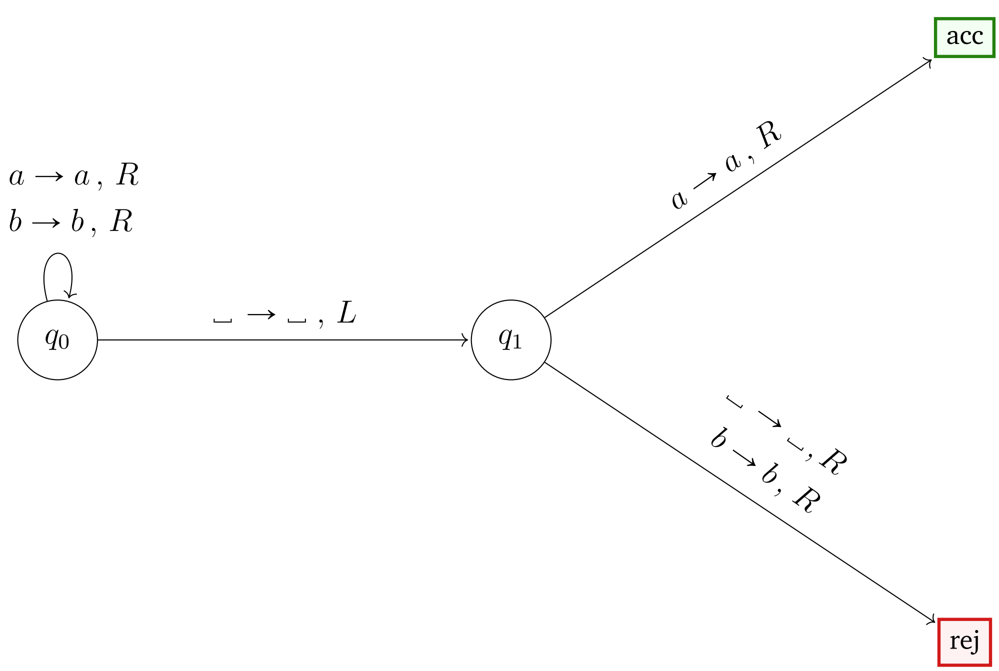
פתרון:

␣	none	a	a	a	␣
␣	✓	one	a	a	␣
␣	✓	a	even	a	␣
␣	✓	a	✓	odd	␣
␣	✓	a	✓	␣	rej

u	q	σ	v
␣	none	a	aa ␣
␣ ✓	one	a	a ␣
␣ ✓ a	even	a	␣
␣ ✓ a ✓	odd	␣	␣
␣ ✓ a ✓ ␣	rej	␣	␣

דוגמה 1.8

מהי שפת המכונה:



פתרון:

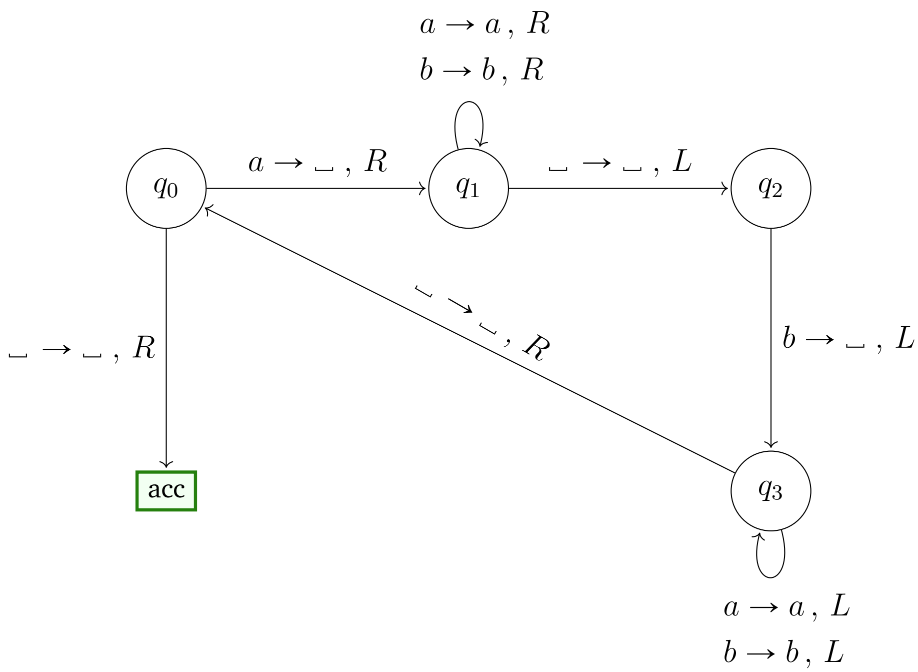
תיאור מילולי:

- במצב התחלתי q_0:
  - אם אנחנו רואים a, עוברים למשבצת הבאה לימין הראש.
  - אם אנחנו רואים b, עוברים למשבצת הבהאה לשמאל הראש.
- ממשיכים כך עד שנגיע לתו רווח, כלומר לסוף המילה, ואז עוברים למשבצת לשמאל הראש, כלומר לתו האחרון של המילה.
  - אם אנחנו רואים a, המילה מתקבלת. (ז"א התו האחרון הינו a.)
  - אם אנחנו רואים b, המילה נדחית. (ז"א התו האחרון הינו b.)
  - אם אנחנו רואים תו-רווח המילה נדחית. (ז"א המילה הינה ריקה.)

תשובה סופית: המכונה מקבלת שפת המילים המסתיימות באות a.

דוגמה 1.9

מהי שפת המכונה:



פתרון:

תיאור מילולי:

- במצב התחלתי q_0:
  - * אם אנחנו רואים b, המילה נדחית.
  - * אם אנחנו רואים $\sqcup$, המילה מתקבלת.
  - * אם אנחנו רואים a, כותבים עליה $\sqcup$ ועוברים למשבצת הבאה לימין הראש, והמ"ט עוברת למצס q_1.
- במצב q_1 אנחנו ראינו a וכתבנו עליה $\sqcup$.
  - * אם אנחנו רואים במשבצת הבאה a או b, ממשיכים למשבצת הבאה לימין והמ"ט נשארת במצב q_1.
  - * אם אנחנו רואים תו רווח (כלומר הגענו לסוף המילה) הראש זז למשבצת השמאלי, כלומר לאות האחרונה של המילה והמ"ט עוברת למצב q_2.
- במצב q_2 ראינו a בתו הראשון, כתבנו עליה $\sqcup$ והראש קורא התו האחרון.
  - * אם אנחנו רואים a המילה נדחית.
  - * אם אנחנו רואים $\sqcup$, המילה נדחית.
  - * אם אנחנו רואים b כותבים עליה $\sqcup$ והמ"ט עוברת למצב q_3.
- במצב q_3 קראנו a בתו הראשון ומחקנו אותה, קראנו b ומחקנו אותה.
- הראש זז משבצת אחת שמאלה עד שיגיע לתו הרשאון ומ"ט חוזרת למצב התחלת q_0.

- המ"ט באופן איטרטיבי, עוברת על הקלט ובכל מעבר:
  - אם יש a בתחילת המילה מורידה אותה ומחליפה אותה עם ␣, אחרת המילה נדחית,
  - אם יש b בסופה של המילה מורידה אותה ומחליפה אותה עם ␣, אחרת המילה נדחית.
- אם לאחר מספר מעברים כאלו הסרט ריק, המ"ט מקבלת, וזה יתקיים לכל מילה ורק למילים בשפה

$$\left\{a^n b^n \middle| n \geqslant 0\right\} \ .$$

תשובה סופית: המכונה מקבלת שפת המילים

$$\left\{a^n b^n \middle| n \geqslant 0\right\} \ .$$

דוגמה 1.10

פתרון:

μ	q	σ	ν
␣ ␣	q_0	a	aaabbbb␣ ␣
␣ ␣ ␣	q_1	a	aabbbb␣ ␣
␣ ␣ ␣ a	q_1	a	abbbb␣ ␣
␣ ␣ ␣ aa	q_1	a	bbbb␣ ␣
␣ ␣ ␣ aaa	q_1	b	bbb␣ ␣
␣ ␣ ␣ aaab	q_1	b	bb␣ ␣
␣ ␣ ␣ aaabb	q_1	b	b␣ ␣
␣ ␣ ␣ aaabbb	q_1	b	␣ ␣
␣ ␣ ␣ aaabbbb	q_1	␣	␣
␣ ␣ ␣ aaabbb	q_2	b	␣ ␣
␣ ␣ ␣ aaabb	q_3	b	␣ ␣ ␣
␣ ␣ ␣ aaab	q_3	b	b␣ ␣ ␣
␣ ␣ ␣ aaa	q_3	b	bb␣ ␣ ␣
␣ ␣ ␣ aa	q_3	a	bbb␣ ␣ ␣
␣ ␣ ␣ a	q_3	a	abbb␣␣␣
␣␣␣	q_3	a	aabbb␣␣␣
␣␣	q_3	␣	aaabbb␣␣␣
␣ ␣ ␣	q_0	a	aabbb␣ ␣ ␣
␣ ␣ ␣ ␣	q_1	a	abbb␣ ␣ ␣
␣ ␣ ␣ ␣ a	q_1	a	bbb␣ ␣ ␣
␣ ␣ ␣ ␣ aa	q_1	b	bb␣ ␣ ␣
␣ ␣ ␣ ␣ aab	q_1	b	b␣ ␣ ␣
␣ ␣ ␣ ␣ aabb	q_1	b	␣ ␣ ␣
␣ ␣ ␣ ␣ aabbb	q_1	␣	␣ ␣
␣ ␣ ␣ ␣ aabb	q_2	b	␣ ␣ ␣
␣ ␣ ␣ ␣ aab	q_3	b	␣ ␣ ␣ ␣
␣ ␣ ␣ ␣ aa	q_3	b	b␣ ␣ ␣ ␣
␣ ␣ ␣ ␣ a	q_3	a	bb␣ ␣ ␣ ␣
␣ ␣ ␣ ␣	q_3	a	abb␣ ␣ ␣ ␣

␣ ␣ ␣	q_3	␣	aabb␣ ␣ ␣ ␣
␣ ␣ ␣ ␣	q_0	a	abb␣ ␣ ␣ ␣
␣ ␣ ␣ ␣ ␣	q_1	a	bb␣ ␣ ␣ ␣
␣ ␣ ␣ ␣ ␣ a	q_1	b	b␣ ␣ ␣ ␣
␣ ␣ ␣ ␣ ␣ ab	q_1	b	␣ ␣ ␣ ␣
␣ ␣ ␣ ␣ ␣ abb	q_1	␣	␣ ␣ ␣
␣ ␣ ␣ ␣ ␣ ab	q_2	b	␣ ␣ ␣ ␣
␣ ␣ ␣ ␣ ␣ a	q_3	b	␣ ␣ ␣ ␣ ␣
␣ ␣ ␣ ␣ ␣	q_3	a	b␣ ␣ ␣ ␣ ␣
␣ ␣ ␣ ␣	q_3	␣	ab␣ ␣ ␣ ␣ ␣
␣ ␣ ␣ ␣ ␣	q_0	a	b␣ ␣ ␣ ␣ ␣
␣ ␣ ␣ ␣ ␣ ␣	q_1	b	␣ ␣ ␣ ␣ ␣
␣ ␣ ␣ ␣ ␣ ␣ b	q_1	␣	␣ ␣ ␣ ␣
␣ ␣ ␣ ␣ ␣ ␣	q_2	b	␣ ␣ ␣ ␣ ␣
␣ ␣ ␣ ␣ ␣	q_3	␣	␣ ␣ ␣ ␣ ␣ ␣
␣ ␣ ␣ ␣ ␣ ␣	q_0	␣	␣ ␣ ␣ ␣ ␣

הגדרה 1.4 גרירה בצעד אחד

תהי $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}})$ מכונת טיורינג, ותהיינה c_1 ו- c_2 קונפיגורציות של M. נסמן

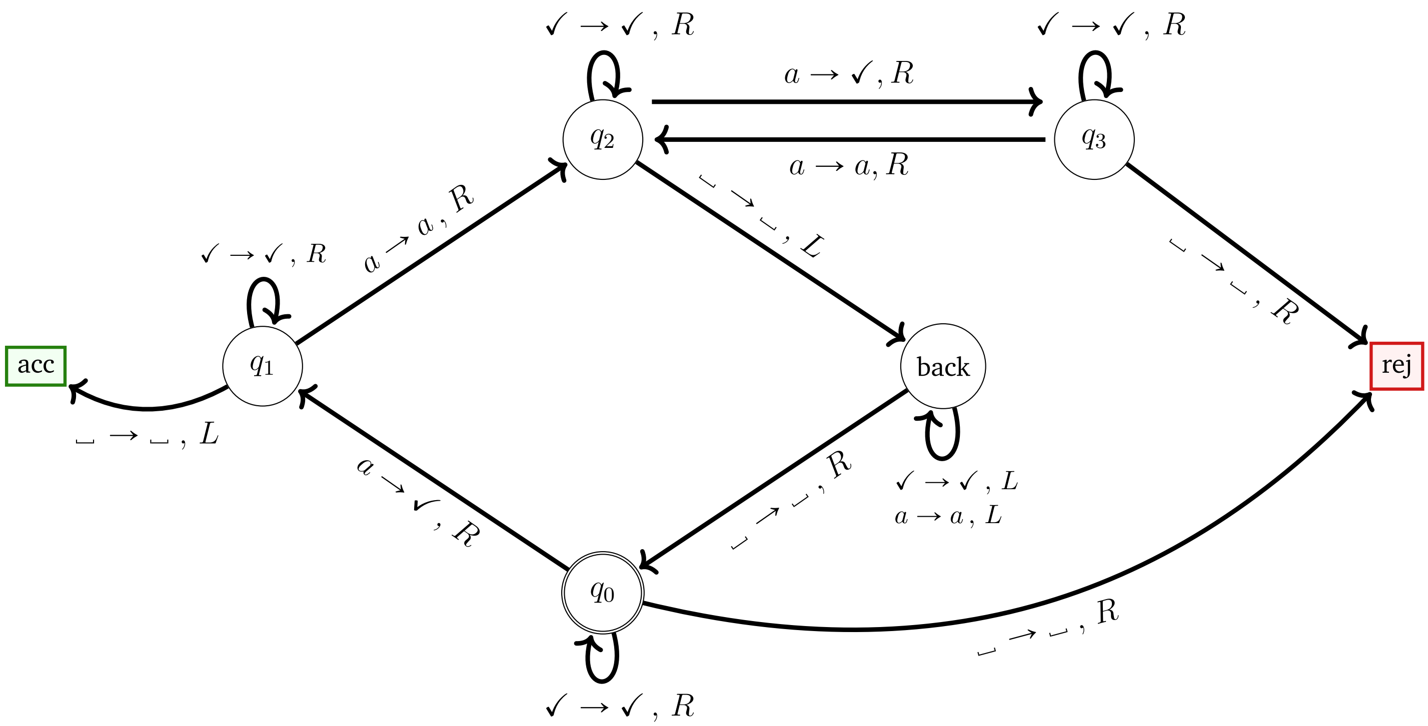
$$c_1 \vdash_M c_2$$

(במילים, c_1 גורר את c_2) אם כשנמצאים ב- c_1 עוברים ל- c_2 בצעד בודד.

דוגמה 1.11 (המשך של דוגמה 1.5)

במכונת טיורינג שמתואר בתרשים דמטה (אשר שווה למ"ט בדוגמה 1.5 רק עם סימנים שונים למצבים) מתקיים

$$\checkmark q_0 a \checkmark a \ \vdash_M \ \checkmark\checkmark q_1 \checkmark a$$



הגדרה 1.5 גרירה בכללי

תהי $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}})$ מכונת טיורינג, ותהיינה c_1 ו- c_2 קונפיגורציות של M. נסמן

$$c_1 \vdash_M^* c_2$$

אם ניתן לעבור מ- c_1 ל- c_2 ב- 0 או יותר צעדים.

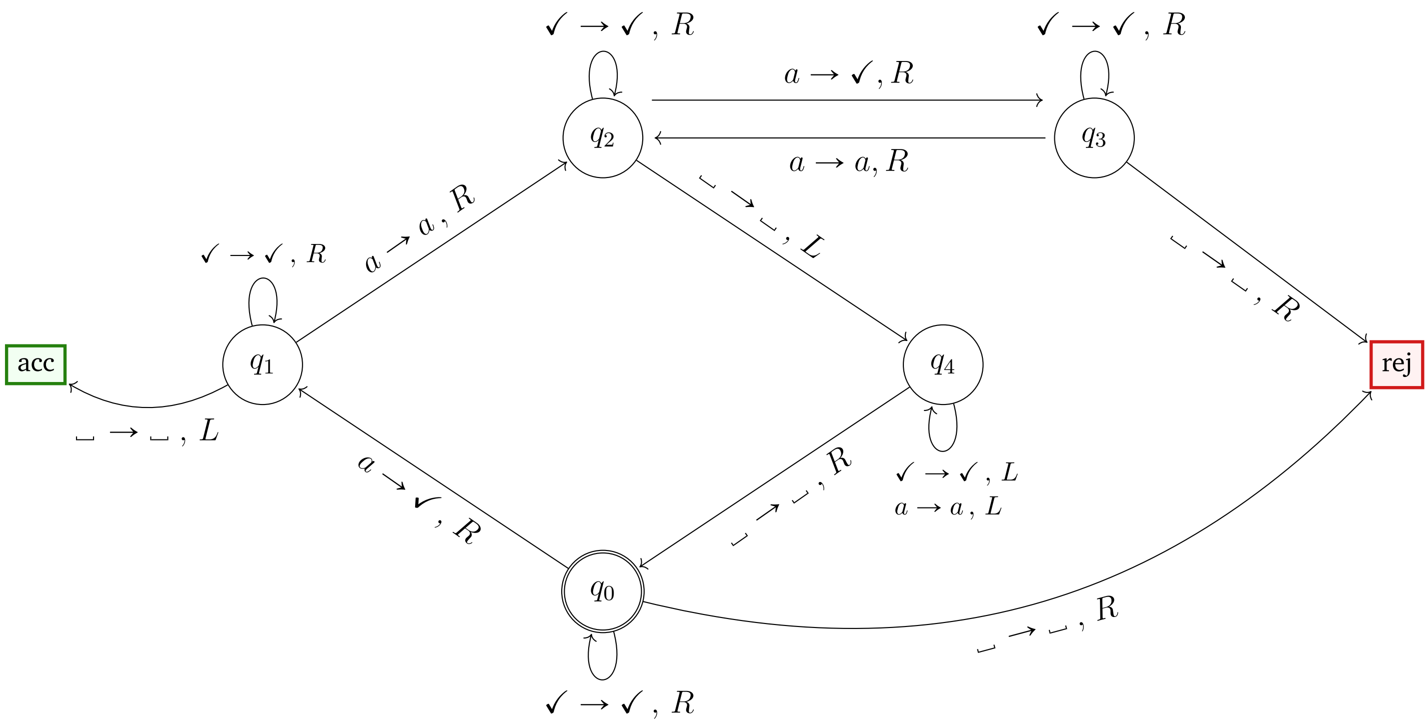
דוגמה 1.12 (המשך של דוגמה 1.5)

במכונת טיורינג שמתואר בתרשים דמטה (אשר שווה למ"ט בדוגמה 1.5 רק עם סימנוים שונים למצבים) מתקיים

$$\checkmark q_0 a \checkmark a \ \vdash_M^* \ \checkmark\checkmark\checkmark q_4 a$$

כי

$$\begin{aligned} \checkmark q_0 a \checkmark a \ &\vdash_M \checkmark\checkmark q_1 \checkmark a \\ &\vdash_M \checkmark\checkmark\checkmark q_1 a \\ &\vdash_M \checkmark\checkmark\checkmark a q_2 \sqcup \\ &\vdash_M \checkmark\checkmark\checkmark q_4 a \ . \end{aligned}$$



הגדרה 1.6 קבלה ודחייה של מחרוזת

תהי $M=(Q,\Sigma,\Gamma,\delta,q_0,q_{\text{acc}},q_{\text{rej}})$ מכונת טיורינג, ו- $w\in\Sigma^*$ מחרוזת. אומרים כי:

- M **מקבלת** את w אם
$$q_0 w \quad \vdash^*_M \quad u\ q_{\text{acc}}\,\sigma\,\text{v}$$
כאשר $\sigma\in\Gamma$, $u,\text{v}\in\Gamma^*$ כלשהם.
- M **דוחה** את w אם
$$q_0 w \quad \vdash^*_M \quad u\ q_{\text{rej}}\ \sigma\ \text{v}$$
כאשר $\sigma\in\Gamma$, $u,\text{v}\in\Gamma^*$ כלשהם.

הגדרה 1.7 הכרעה של שפה

תהי $M=(Q,\Sigma,\Gamma,\delta,q_0,\text{acc},q_{\text{rej}})$ מכונת טיורינג, ו- $L\subseteq\Sigma^*$ שפה. אומרים כי M **מכריעה** את L אם לכל $w\in\Sigma^*$ מתקיים:

- $w\in L \Leftarrow M$ מקבלת את w.
- $w\notin L \Leftarrow M$ דוחה את w.

הגדרה 1.8 קבלה של שפה

תהי $M=(Q,\Sigma,\Gamma,\delta,q_0,q_{\text{acc}},q_{\text{rej}})$ מכונת טיורינג, ו- $L\subseteq\Sigma^*$ שפה. אומרים כי M **מקבלת** את L אם לכל $w\in\Sigma^*$ מתקיים:

- אם $w\in L$ אז M מקבלת את w.
- אם $w\notin L$ אז M לא מקבלת את w.

במקרה כזה כאשר M מקבלת את השפה L, נכתוב ש-

$$L(M) = L \ .$$

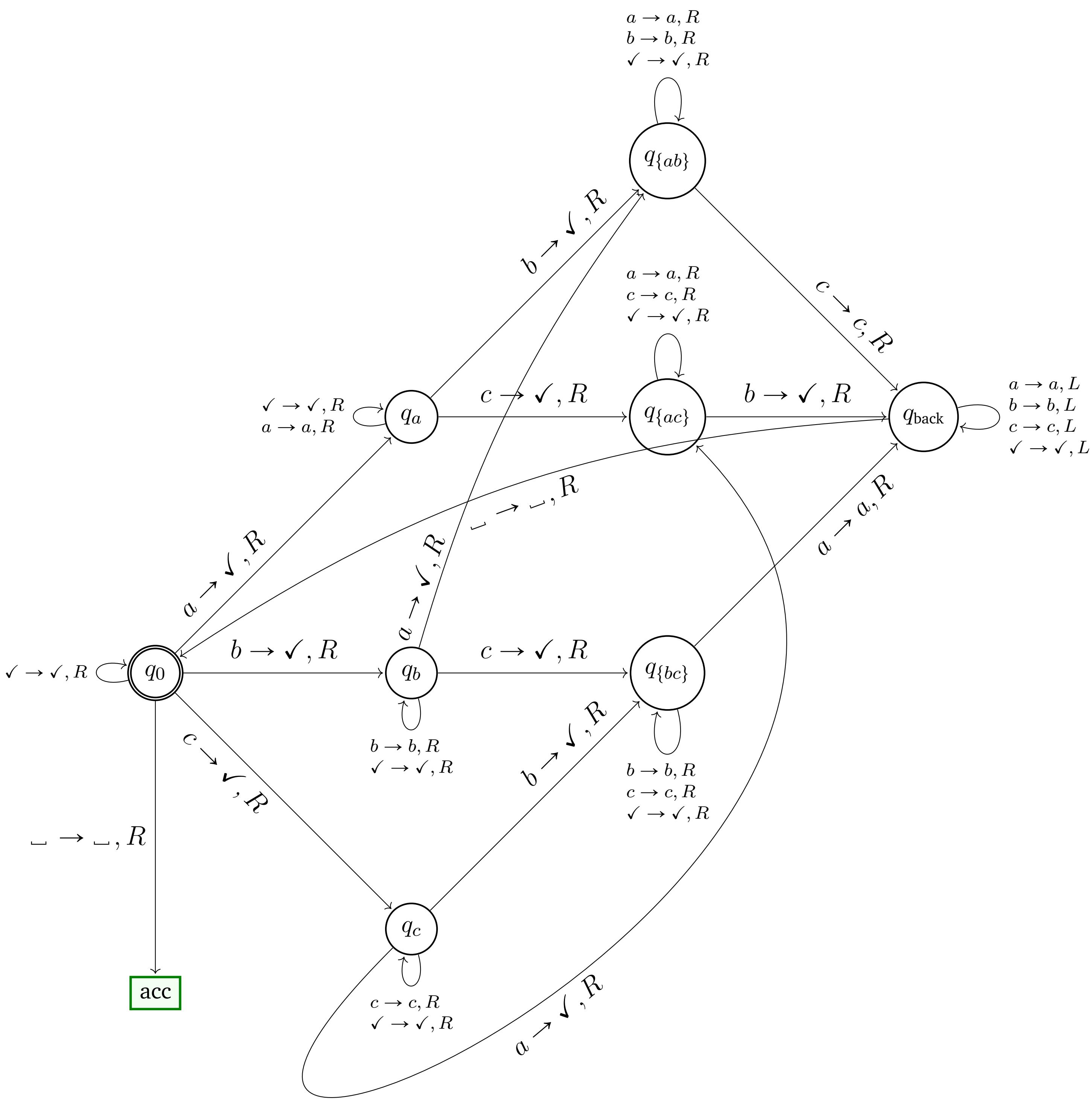
1.2 טבלת המעברים

דוגמה 1.13

בנו מכונת טיורינג שמכריעה את השפה

$$L = \left\{ w = \{a,b,c\}^* \middle| \#a_w = \#b_w = \#c_w \right\}$$

פתרון:


$a \to a, R$
$b \to b, R$
$\checkmark \to \checkmark, R$
$q_{\{ab\}}$
$b \to \checkmark, R$
$c \to c, R$
$a \to a, R$
$c \to c, R$
$\checkmark \to \checkmark, R$
$\checkmark \to \checkmark, R$
$a \to a, R$
q_a
$c \to \checkmark, R$
$q_{\{ac\}}$
$b \to \checkmark, R$
q_{back}
$a \to a, L$
$b \to b, L$
$c \to c, L$
$\checkmark \to \checkmark, L$
$\sqcup \to \sqcup, R$
$a \to \checkmark, R$
$a \to \checkmark, R$
$a \to a, R$
$\checkmark \to \checkmark, R$
q_0
$b \to \checkmark, R$
q_b
$c \to \checkmark, R$
$q_{\{bc\}}$
$b \to b, R$
$\checkmark \to \checkmark, R$
$c \to \checkmark, R$
$b \to \checkmark, R$
$b \to b, R$
$c \to c, R$
$\checkmark \to \checkmark, R$
$\sqcup \to \sqcup, R$
q_c
acc
$c \to c, R$
$\checkmark \to \checkmark, R$
$a \to \checkmark, R$

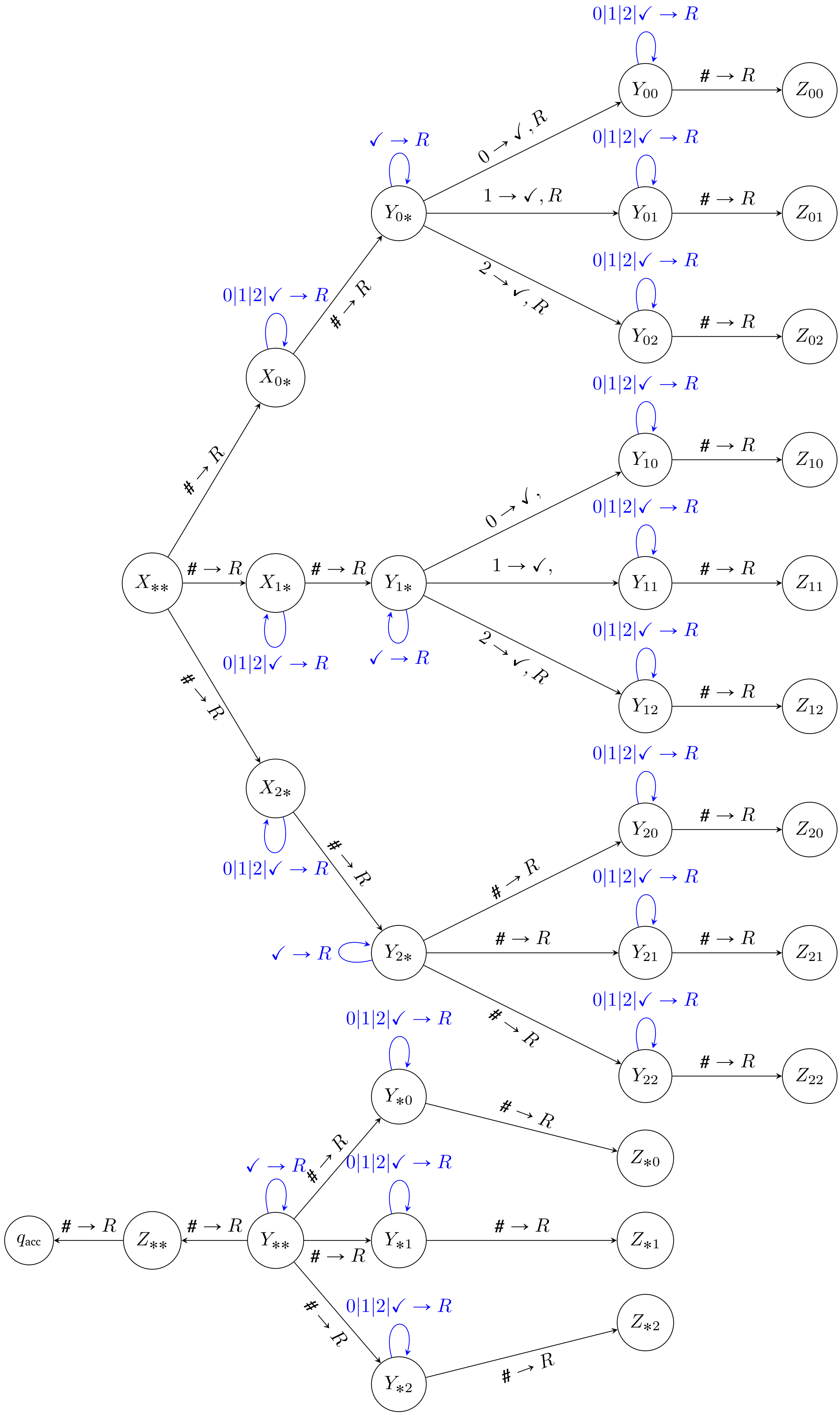
תנאי	תזוזה	כתיבה	מצב חדש	סימון בסרט	מצב
$\sigma \notin S$	R	✓	$q.\left(S \cup \{\sigma\}\right)$	σ	$q.S$
$\sigma \in S$	R		$q.S$	σ	$q.S$
	L		back	$a, b, c, \checkmark$	$q/\{a, b, c\}$
	R		acc	␣	$q.\varnothing$
	L		back	$a, b, c, \checkmark$	back
	R		$q.\varnothing$	␣	back

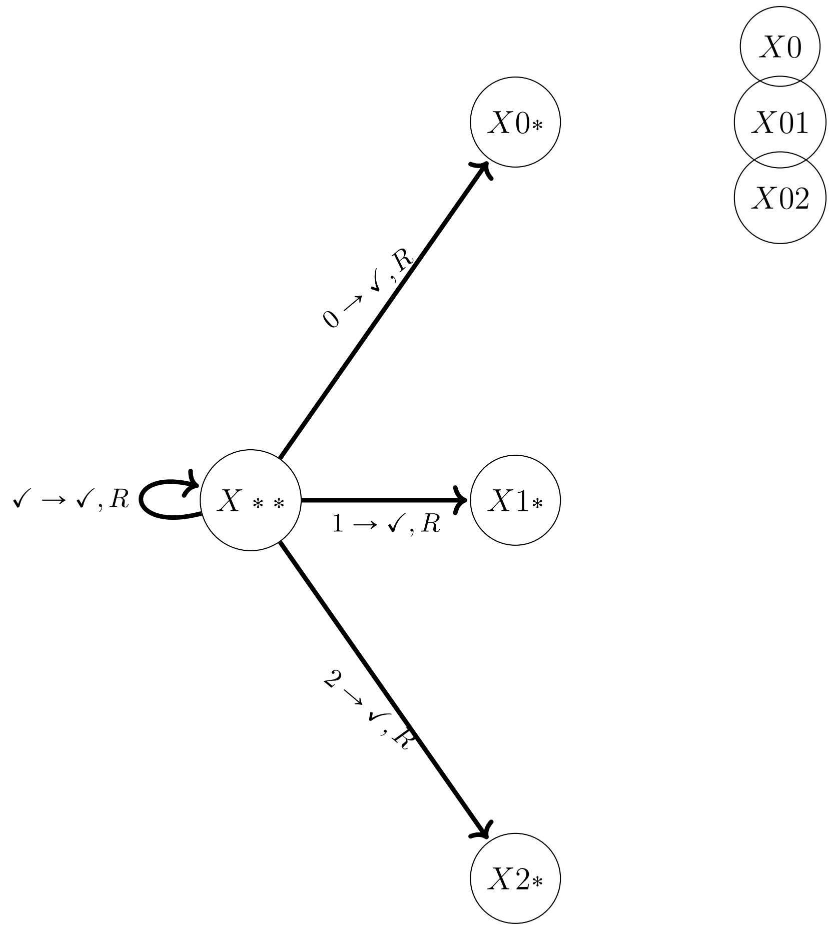
דוגמה 1.14

בנו מכונת טיורינג שמכריעה את השפה

$$\left\{x_1 \ldots x_k \# y_1 \ldots y_k \# z_1 \ldots z_k \;\middle|\; x_i, y_i, z_i \in \{0, \ldots, 3\}\ ,\ \forall i, x_i \geqslant z_i \geqslant y_i\right\}$$

פתרון:


X_{**}
X_{0*}
X_{1*}
X_{2*}
Y_{0*}
Y_{1*}
Y_{2*}
Y_{00}
Y_{01}
Y_{02}
Y_{10}
Y_{11}
Y_{12}
Y_{20}
Y_{21}
Y_{22}
Z_{00}
Z_{01}
Z_{02}
Z_{10}
Z_{11}
Z_{12}
Z_{20}
Z_{21}
Z_{22}
Y_{*0}
Y_{*1}
Y_{*2}
Z_{*0}
Z_{*1}
Z_{*2}
Y_{**}
Z_{**}
q_{acc}
$\# \to R$
$0|1|2|\checkmark \to R$
$\checkmark \to R$
$0 \to \checkmark, R$
$1 \to \checkmark, R$
$2 \to \checkmark, R$
$0 \to \checkmark,$
$1 \to \checkmark,$



מצב	סימון בסרט	מצב חדש	כתיבה	תזוזה	תנאי
$X**$	σ	$X\sigma*$	$\checkmark$	R	
$X**$	$\checkmark$	$X**$	$\checkmark$	R	
$X\sigma*$	$0,1,\ldots,9,\checkmark$	$X\sigma*$		R	
$X\tau*$	#	$Y\tau*$		R	
$Y\tau*$	σ	$Y\tau\sigma$		R	
$Y\tau*$	$\checkmark$	$Y\tau*$		R	
$Y\tau\sigma$	$0,1,\ldots,9,\checkmark$	$Y\tau\sigma$		R	
$Y\tau_1\tau_2$	#	$Z\tau_1\tau_2$		R	
$Z\tau_1\tau_2$	$\checkmark$	$Z\tau_1\tau_2$		R	
$Z\tau_1\tau_2$	σ	back	$\checkmark$	L	
$Z**$	␣	acc		R	
back	$0,1,\ldots,9,\checkmark$	back		L	
back	␣	$X**$		R	

1.3 חישוב פונקציות

הגדרה 1.9 מכונת טיורינג שמחשבת פונקציה f

תהי $f:\Sigma_1^* \to \Sigma_2^*$ ותהי $M=(Q,\Sigma,\Gamma,\delta,q_0,q_{\text{acc}},q_{\text{rej}})$ מכונת טיורינג.
אומרים כי M מחשבת את f אם:

- $\Sigma=\Sigma_1$ ו- $\Sigma_2 \subset \Gamma$.

- לכל $w \in \Sigma_1^*$ מתקיים $q_0 w \vdash q_{\text{acc}} f(w)$.

דוגמה 1.15 חיבור אונרי

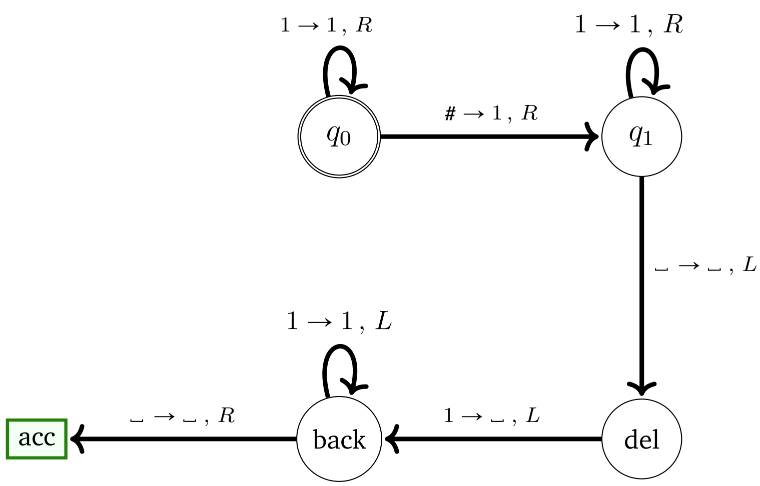
בנו מכונת טיורינג אשר מקבלת את הקלט

$$1^i\#1^j$$

ומחזירה את פלט

$$1^{i+j} \ .$$

פתרון:



דוגמה 1.16 כפל אונרי

בנו מכונת טיורינג אשר מקבלת את הקלט

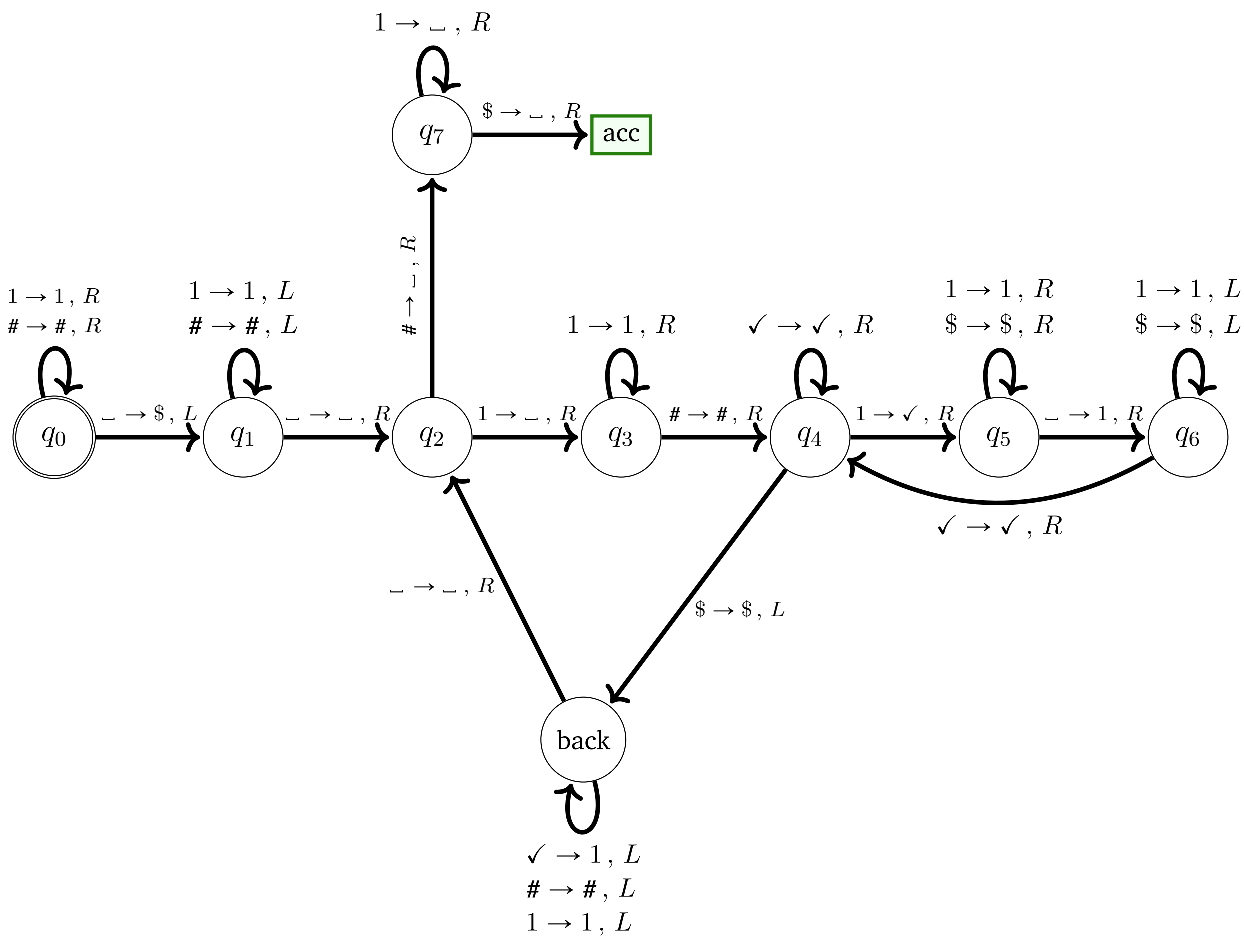
$$1^i\#1^j$$

ומחזירה את פלט

$$1^{i\cdot j} \ .$$

פתרון:

- לדוגמה, נניח שהקלט הוא 2 כפול 2.
  הקלט הוא 11#11.
- נרצה להבדיל בין הקלט לבין הפלט.
  לכן בתחילת הריצה, נתקדם ימינה עד סוף הקלט ונוסיף שם את התו \$.
  לאחר מכן נחזור לתחילת הקלט.
- על כל אות 1 במילה השמאלית נעתיק את המילה הימינית לאחר סימן ה- \$.
- לאחר מכן נשאיר רק את התווים שלאחר סימן ה \$. כלומר, נמחק את כל מה שאינו פלט.



μ	q	σ	ν
␣	q_0	1	1#11␣
␣11#11	q_1	␣	␣
␣11#11	q_1	\$	␣
␣	q_1	␣	11#11\$
␣	q_2	1	1#11\$
␣ ␣	q_3	1	#11\$
␣ ␣1#	q_4	1	1\$
␣ ␣1#✓	q_5	1	\$
␣ ␣1#✓1\$	q_5	␣	␣
␣ ␣1#✓1\$1	q_6	␣	␣
␣ ␣1#	q_6	✓	1\$1 ␣
␣ ␣1#✓	q_4	1	\$1 ␣
␣ ␣1#✓✓	q_5	\$	1 ␣
␣ ␣1#✓✓\$1	q_5	␣	␣
␣ ␣1#✓✓\$11	q_6	␣	␣
␣ ␣1#✓	q_6	✓	\$11␣
␣ ␣1#✓✓	q_4	\$	11␣
␣ ␣1#✓	back	✓	\$11␣
␣	back	␣	1#11\$11␣
␣ ␣	q_2	1	#11\$11␣
␣ ␣ ␣	q_3	#	11\$11␣
␣ ␣ ␣#	q_4	1	1\$11␣

$\sqcup\sqcup\sqcup\#\checkmark$	q_5	1	$\$11\sqcup$
$\sqcup\sqcup\sqcup\#\checkmark 1\$11$	q_5	$\sqcup$	$\sqcup$
$\sqcup\sqcup\sqcup\#\checkmark 1\$111$	q_6	$\sqcup$	$\sqcup$
$\sqcup\sqcup\sqcup\#$	q_6	$\checkmark$	$1\$111\sqcup$
$\sqcup\sqcup\sqcup\#\checkmark$	q_4	1	$\$111\sqcup$
$\sqcup\sqcup\sqcup\#\checkmark\checkmark$	q_5	$\$$	$111\sqcup$
$\sqcup\sqcup\sqcup\#\checkmark\checkmark\$111$	q_5	$\sqcup$	$\sqcup$
$\sqcup\sqcup\sqcup\#\checkmark\checkmark\$1111$	q_6	$\sqcup$	$\sqcup$
$\sqcup\sqcup\sqcup\#\checkmark$	q_4	$\checkmark$	$\$1111$
$\sqcup\sqcup\sqcup\#\checkmark\checkmark$	q_4	$\$$	1111
$\sqcup\sqcup\sqcup\#\checkmark$	back	$\checkmark\$$	1111
$\sqcup\sqcup$	back	$\sqcup$	$\#11\$1111$
$\sqcup\sqcup\sqcup$	q_2	$\#$	$11\$1111$
$\sqcup\sqcup\sqcup\sqcup$	q_7	1	$1\$1111$
$\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup$	q_7	$\$$	1111
$\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup$	acc	1	111

שיעור 2
מודלים חישובים שקולית

הגדרה 2.1 מודל חישובי

מודל חישובי הוא אוסף של מכונות טיורינג שעבורן מוגדרים המושגים של הכרעה וקבלה של שפות.

הגדרה 2.2 מודלים שקולים חישובית

יהיו A ו- B מודלים חישוביים. אומרים כי A ו- B שקולים אם לכל שפה L התנאים הבאים מתקיימים:

(1 קיימת מ"ט במודל A שמכריעה את L אם"ם קיימת מ"ט במודל B שמכריעה את L.

(2 קיימת מ"ט במודל A שמקבלת את L אם"ם קיימת מ"ט במודל B שמקבלת את L.

דוגמה 2.1

נסמן ב-T את מודל מכונת הטיורינג הבסיסי.
במודל זה בכל צעד ניתן לזוז ימינה או שמאלה. אך לא ניתן להישאר במקום, באותה המשבצת בסרט.
במודל זה, הסרט הוא אינסופי לשני הכיוונים. ב תחילת החישוב הראש נמצא בתחילת הקלט.

נסמן ב-O את מודל מכונת הטיורינג עם סרט ימינה בלבד.
במודל זה בכל צעד ניתן לזוז ימינה או שמאלה. אך לא ניתן לה ישאר במקום, באותה המשבצת בסרט.
במודל זה, הסרט הוא אינסופי לכיוון אחד בלבד - ימינה. בתחילת החישוב, הקלט ממוקם בקצה השמאלי של הסרט והראש נמצאת בתחילת הקלט. החישוב מתנהל כמו במכונה במודל, T למעט כאשר הראש נמצא במשבצת השמאלית ביותר בסרט וצריך לזוז ש מאלה– במקרה כזה הראש נשאר במקום ולא זז.

הוכיחו כי המודל T והמודל O שקולים חישובית.

פתרון:
יש להוכיח ש:

- לכל מ"ט במודל O קיימת מ"ט שקולה במודל T.
- לכל מ"ט במודל T קיימת מ"ט שקולה במודל O.

<u>כיוון ראשון</u>

נוכיח כי לכל מ"ט במודל O קיימת מ"ט שקולה במודל T. כלומר:

נתונה $M^O = \left(Q^O, \Sigma^O, \Gamma^O, \delta^O, q_0^O, \mathrm{acc}^O, \mathrm{rej}^O\right)$ במודל O.

נבנה , $M^T = \left(Q^T, \Sigma^T, \Gamma^T, \delta^T, q_0^T, \mathrm{acc}^T, \mathrm{rej}^T\right)$ שקולה במודל T.

נעבוד רק עם צד ימין של הסרט האינסופי של M^T ואז M^T תהיה שקולה ל- M^O.

רכיבי המ"ט M^T זהים לאלו של המ"ט M^O, מלבד מהתכונה שהראש של M^O לא זז מעבר לקצה השמאולי של הקלט.

לכן כדי ש- M^T תהיה שקולה ל- M^O נוסיף מעברים לפונקצית המעברים של M^T כדי שהראש של M^T לא זז מעבר לקצה השמאולי של הקלט.

הרעיון הוא לסמן את המשבצת שמשמאול לתחילת הקלט עם סימן מיוחד \$, ואז להוסיף מעברים לפונקצית המעברים של M^T שמבטיחים שאם הראש נמצא למשבצת שמסומנת \$ אז הוא מיד חוזר ו- M^T חוזרת למצב ההתחתי של המ"ט M^O. זה מתבצע על ידי הוספת השורות הבאות לטבלת המעברים של M^T:

מצב	סימון	מצב חדש	כתיבה	תזוזה	תנאי
q_0^T	σ	$q_\$$	$\circlearrowleft$	L	
$q_\$$	$\sqcup$	q_0^O	$\$$	R	
q	$\$$	q	$\$$	R	$\forall q \in Q^O$

$$Q^T = Q^O \cup \{q_0^T, q_\$\} , \qquad \Sigma^T = \Sigma^O , \qquad \Gamma^T = \Gamma^O \cup \{\$\} , \qquad \text{acc}^T = \text{acc}^O , \qquad \text{rej}^T = \text{rej}^O .$$

<u>כיוון שני</u>

נוכיח כי לכל מ"ט במודל T קיימת מ"ט שקולה במודל O. כלומר:

נתונה $M^T = \left(Q^T, \Sigma^T, \Gamma^T, \delta^T, q_0^T, \text{acc}^T, \text{rej}^T\right)$ במודל T.

נבנה $M^O = \left(Q^O, \Sigma^O, \Gamma^O, \delta^O, q_0^O, \text{acc}^O, \text{rej}^O\right)$ שקולה במודל O.

הרעיון הוא לסמן קו על הסרט במשבת באמצע הקלט, ואז לקפל את הסרט בקו הזה. באופן הזה נקבל סרט עם קצה שמאולי ואינסופי ימינה. במשבצות של הסרט החדש המקופל יש שני תווים, אחד למעלה (U) ואחד למטה (D), מלבד מנקודת הקיפול שבו יש משבצת אחת שמסומנת \$.

באופן הזה אפשר לסמלץ את המכונה M^T במכונה M^O על ידי הוספת המעברים הבאים לטבלת המעברים של M^T. לכל $\tau, \sigma, \pi \in \Gamma^T$:

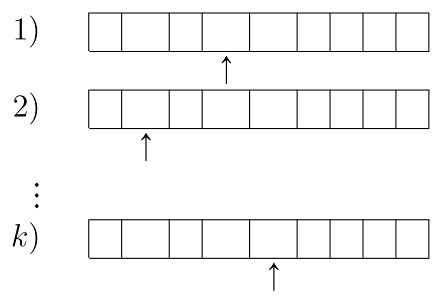
תנאי	תזוזה	כתיבה	מצב חדש	סימון	מצב
תזוזה שמאלה: $(q,\sigma) \xrightarrow{M^T} (p,\tau,L)$	L	π τ	$p.D$	π σ	$q.D$
	R	τ π	$p.U$	σ π	$q.U$
תזוזה שמאלה: $(q,\sqcup) \xrightarrow{M^T} (p,\tau,L)$	L	$\sqcup$ τ	$p.D$	$\sqcup$	$q.D$
	R	τ $\sqcup$	$p.U$	$\sqcup$	$q.U$
תזוזה ימינה: $(q,\sigma) \xrightarrow{M^T} (p,\tau,R)$	R	π τ	$p.D$	π σ	$q.D$
	L	τ π	$p.U$	σ π	$q.U$
תזוזה ימינה: $(q,\sqcup) \xrightarrow{M^T} (p,\tau,R)$	R	$\sqcup$ τ	$p.D$	$\sqcup$	$q.D$
	L	τ $\sqcup$	$p.U$	$\sqcup$	$q.U$
	R	$\circlearrowleft$	$q.U$	$\$$	$q.D$
	R	$\circlearrowleft$	$q.D$	$\$$	$q.U$
אתחול					
$\tau \in \Sigma \cup \{\sqcup\}$ $\sigma \in \Sigma$	R	$\$$	$q.\tau$	τ	q_0^O
	R	$\sqcup$ σ	$q.\tau$	τ	$q.\sigma$
	L	$\sqcup$ $\sqcup$	back	$\sqcup$	$q.\sqcup$
	L	$\circlearrowleft$	back	$\sqcup$ τ	back
	R	$\circlearrowleft$	$q_0^T.D$	$\$$	back
סיום					
			acc^O	הכל	acc$^T.D$
			acc^O	הכל	acc$^T.U$
			rej^O	הכל	rej$^T.D$
			rej^O	הכל	rej$^T.U$
כל השאר עוברים ל-rej					

$$\Gamma^O \supseteq \left(\Gamma^T \times \Gamma^T\right) \cup \{\$\} \ .$$

שיעור 3
מכונות טיורינג מרובת סרטים

3.1 מכונת טיורינג מרובת סרטים: הגדרה היוריסטית

מכונת טיורינג מרובת סרטים (מטמ"ס) היא הכללה של מ"ט עם סרט יחיד. ההבדל הוא שלמטמ"ס ישנו מספר סופי של סרטים, נניח $k>1$ סרטים.



- לכל סרט יש ראש שלו.
- בתחילת העבודה הקלט w כתוב בתחילת הסרט הראשון וכל שאר הסרטים ריקים. הראשים בכל סרט מצביעים על התא הראשון בסרט, והמכונה נמצאת במצב התחלתי q_0.
- בכל צעד חישוב, לפי המצב הנוכחי ול- k התווים שמתחת ל- k הראשים, המכונה מחליטה לאיזה מצב לעבור, מה לכתוב מתחת לכל אחד מ-k הראשים ולאן להזיז את הראש בכל אחד מ-k סרטים.
- הראשים של הסרטים יכולים לזוז באופן בלתי-תלוי בהתאם לפונקצית המעברים של המטמ"ס.

3.2 מכונת טיורינג מרובת סרטים: הגדרה פורמלית

הגדרה 3.1 מכונט טיורינג מרובת סרטים

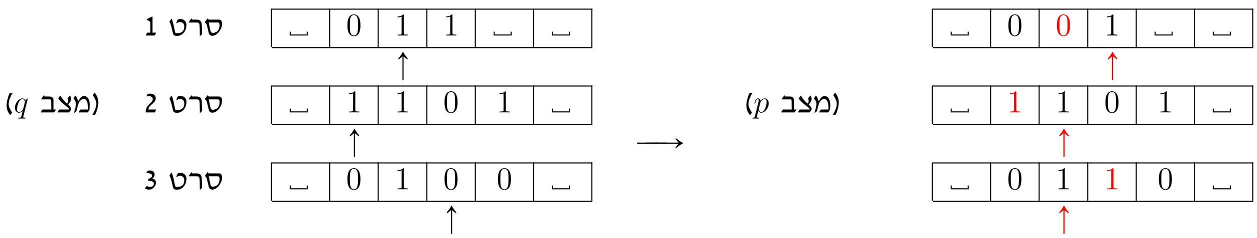
מכונת טיורינג מרובת סרטים היא שביעייה:

$$M=(Q,\Sigma,\Gamma,\delta_k,q_0,q_{\text{acc}},q_{\text{rej}})$$

כאשר Q, Σ, Γ, q_0, q_{acc},q_{rej} מוגדרים כמו מ"ט עם סרט יחיד (ראו הגדרה 1.2).
ההבדל היחיד בין מ"ט עם סרט יחיד לבין מטב"ס הוא הפונקצית המעברים. עבור מטמ"ס הפונקצית המעברים היא מצורה הבאה:

$$\delta_k:(Q\backslash\{q_{\text{acc}},q_{\text{rej}}\})\times\Gamma^k\to Q\times\Gamma^k\times\{L,R,S\}^k$$

דוגמה 3.1



$$\delta_k\left(q,\begin{pmatrix}1\\1\\0\end{pmatrix}\right)=\left(p,\begin{pmatrix}0\\1\\1\end{pmatrix},\begin{pmatrix}R\\R\\L\end{pmatrix}\right)\ .$$

3.3 קונפיגורציה של מטמ"ס

הכללה של קונפיגורציה של מ"ט עם סרט יחיד:

$$\begin{pmatrix}u_1q\ \mathrm{v}_1\\u_2q\ \mathrm{v}_2\\\vdots\\u_kq\ \mathrm{v}_k\end{pmatrix}$$

דוגמה 3.2

בנו מטמ"ס שמכריעה את השפה:

$$L_{w^R}=\left\{w=\{a,b\}^*\ \middle|\ w=w^R\ .\right\}$$

כלומר שפת הפלינדרומים.

פתרון:

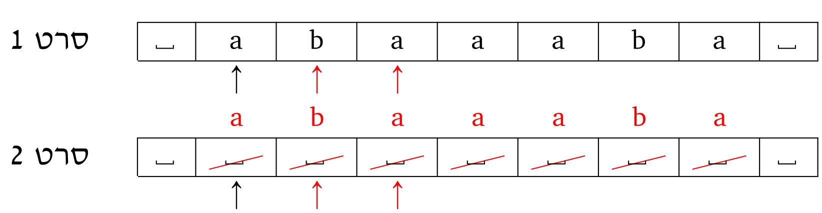
נבנה מ"ט עם שני סרטים:

<u>תאור המכונה:</u>

נסמן M_2 המ"ט עם 2 סרטים שמכריעה את השפה L_{w^R}.

$M_2 =$ על הקלט w:

(1) מעתיקה את w לסרט 2.



(2) מזיזה את הראש בסרט 1 לתו הראשון ב- w ואת הראש בסרט 2 לתו האחרון ב- w.

(3) משווה בין התווים שמתחת לראשים:

- אם התו שמתחת לראש בסרט 1 הוא $\sqcup$ $\Leftarrow$ acc.
- אם התווים שמתחת לראשים שונים $\Leftarrow$ rej.
- אחרת מזיזה את הראש בסרט 1 ימינה ואת הראש בסרט 2 שמאלה, וחוזרת לשלב (3).

הפונקצית המעברים של M_2 היא:

$$\delta\left(q_0, \begin{pmatrix} a \\ \sqcup \end{pmatrix}\right) = \left(q_0, \begin{pmatrix} a \\ a \end{pmatrix}, \begin{pmatrix} R \\ R \end{pmatrix}\right) ,$$

$$\delta\left(q_0, \begin{pmatrix} b \\ \sqcup \end{pmatrix}\right) = \left(q_0, \begin{pmatrix} b \\ b \end{pmatrix}, \begin{pmatrix} R \\ R \end{pmatrix}\right) ,$$

$$\delta\left(q_0, \begin{pmatrix} \sqcup \\ \sqcup \end{pmatrix}\right) = \left(q_{\text{back}}, \begin{pmatrix} \sqcup \\ \sqcup \end{pmatrix}, \begin{pmatrix} L \\ L \end{pmatrix}\right) .$$

נשים לב כי הסיבוכיות זמן של המכונה עם שני סרטים, M_2 היא $O(|w|)$, כאשר w האורך של המילה.

כעת נבנה מ"ט עם סרט יחיד שמכריעה את השפה L_{W^R}.

<u>תאור המכונה:</u>

נסמן M_1 המכונה עם סרט יחיד שמכריעה את השפה L_{w^R}.

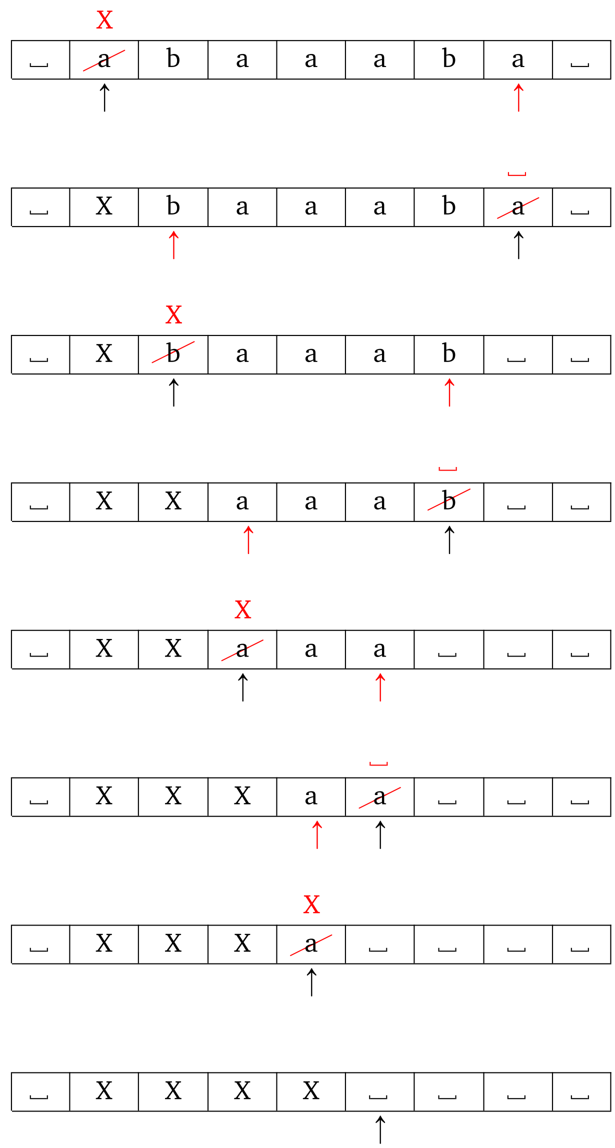
$M_1 =$ על הקלט w:

(1) אם התו שמתחת לראש הוא $\sqcup$ אז $M_1 \leftarrow$ acc.

(2) זוכרת את התו שמתחת לראש ומוחקת אותו ע"י X.

(3) מזיזה את הראש ימינה עד התו הראשון משמאול ל- $\sqcup$.

- אם התו שמתחת לראש הוא X $\Leftarrow$ acc.
- אם התו שונה מהתו שזכרנו $\Leftarrow$ rej.
- מוחקת את התו שמתחת לראש ע"י $\sqcup$, מזיזה את הראש שמאולה עד התו הראשון מימין ל- X וחוזרת לשלב (1).



3.4 שקילות בין מטמ"ס למ"ט עם סרט יחיד

מ"ט עם סרט יחיד היא מקרה פרטי של מטמ"ס.

משפט 3.1 שקילות בין מטמ"ס למ"ט עם סרט יחיד

לכל מטמ"ס M קיימת מ"ט עם סרט יחיד M' השקולה ל- M.

כלומר, לכל קלט $w \in \Sigma^*$:

- אם M מקבלת את w $\Leftarrow$ M' מקבלת את w.
- אם M דוחה את w $\Leftarrow$ M' דוחה את w.
- אם M לא עוצרת על w $\Leftarrow$ M' לא עוצרת על w.

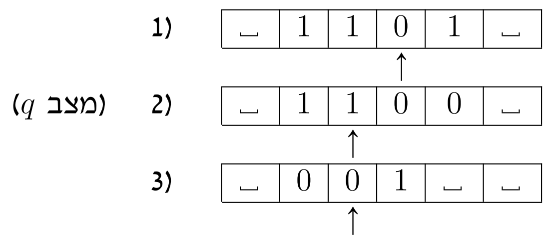
הוכחה:

בהינתן מטמ"ס $M = (Q, \Sigma, \Gamma, \delta_k, q_0, q_{\text{acc}}, q_{\text{rej}})$ עם k סרטים, נבנה מ"ט עם סרט יחיד $M' = (Q', \Sigma, \Gamma', \delta', q'_0, q'_{\text{acc}}, q'_{\text{rej}})$ השקולה ל- M באופן הבא:

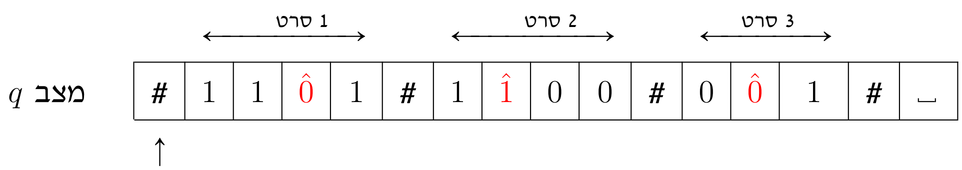
<u>רעיון הבנייה:</u>

בהינתן קלט $w \in \Sigma^*$, M' תבצע "סימולציה" של ריצה M על w.

<u>ב- M</u>



<u>ב- M'</u>



- M' תשמור את התוכן של k הסרטים של M על הסרט, רק שהתוכן של סרט i יופיע בין $\#_i$ ל- $\#_{i+1}$.
- M' תשמור את המיקום של הראשים של M ע"י הכפלת הא"ב Γ.

  כלומר, לכל אות $\alpha \in \Gamma$, M' תשמור שתי אותיות α ו- $\hat{\alpha}$ ב- Γ', כך ש- $\hat{\alpha}$ תסמן את התו שמתחת לראש בכל סרט.
- בכל צעד חישוב, M' סורקת את הסרט שלה משמאל לימין כדי ללמוד מהם התווים שמתחת לראשים (התווים שמסומנים ב- $\hat{\alpha}$).
- M' משתמשת בפונקצית המעברים δ_k של M כדי לחשב את המעבר הבא.
- M' סורקת את הסרט שלה משמאל לימין כדי לעדכן את הסרטים ואת המיקום הראשים בהם.

<u>תאור הבנייה של M':</u>

1) <u>שלב האיתחול</u>

בהינתן קלט $w = \sigma_1\sigma_2\cdots\sigma_n$, M' מאתחלת את הקונפיגורציה ההתחלתית של M על הסרט שלה.

<u>ב- M</u>

1)	$\sqcup$	σ_1	σ_2	$\cdots$	σ_n	$\sqcup$
		↑				
(מצב q_0) 2)	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$	$\sqcup$	$\sqcup$
		↑				
3)	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$	$\sqcup$	$\sqcup$
		↑				

<u>ב- M'</u>

מצב q_0'	σ_1	σ_2	$\cdots$	σ_n	$\sqcup$	$\cdots$

$$\Downarrow$$

מצב q_0	#	$\hat{\sigma}_1$	σ_2	$\cdots$	σ_n	#	$\hat{\sqcup}$	#	$\hat{\sqcup}$	#	$\cdots$	#	$\sqcup$	$\sqcup$
	↑													

2) תאור צעד חישוב של M

<u>ב- M</u>

1)	$\sqcup$	1	1	0	1	$\sqcup$
				↑		
(מצב q) 2)	$\sqcup$	1	1	0	0	$\sqcup$
			↑			
3)	$\sqcup$	0	0	1	$\sqcup$	$\sqcup$
			↑			

$$\delta_k\left(q,\begin{pmatrix}0\\1\\0\end{pmatrix}\right)=\left(p,\begin{pmatrix}1\\0\\0\end{pmatrix},\begin{pmatrix}L\\R\\L\end{pmatrix}\right)$$

1)	$\sqcup$	1	1	1	1	$\sqcup$
			↑			
(מצב p) 2)	$\sqcup$	1	0	0	0	$\sqcup$
				↑		
3)	$\sqcup$	0	0	1	$\sqcup$	$\sqcup$
		↑				

ב- M'

מצב q

#	1	1	$\hat{0}$	1	#	1	$\hat{1}$	0	0	#	0	$\hat{0}$	1	#	␣
$\uparrow$															

$$\Downarrow$$

מצב p

#	1	$\hat{1}$	1	1	#	1	0	$\hat{0}$	0	#	$\hat{0}$	0	1	#	␣
$\uparrow$															

- איסוף מידע
- M' סורקת את הסרט שלה משמאל לימין ומזהה את התווים שמסומנים ב- $\hat{\alpha}$.
  מידע זה ניתן לשמור במצבים. לדוגמה:

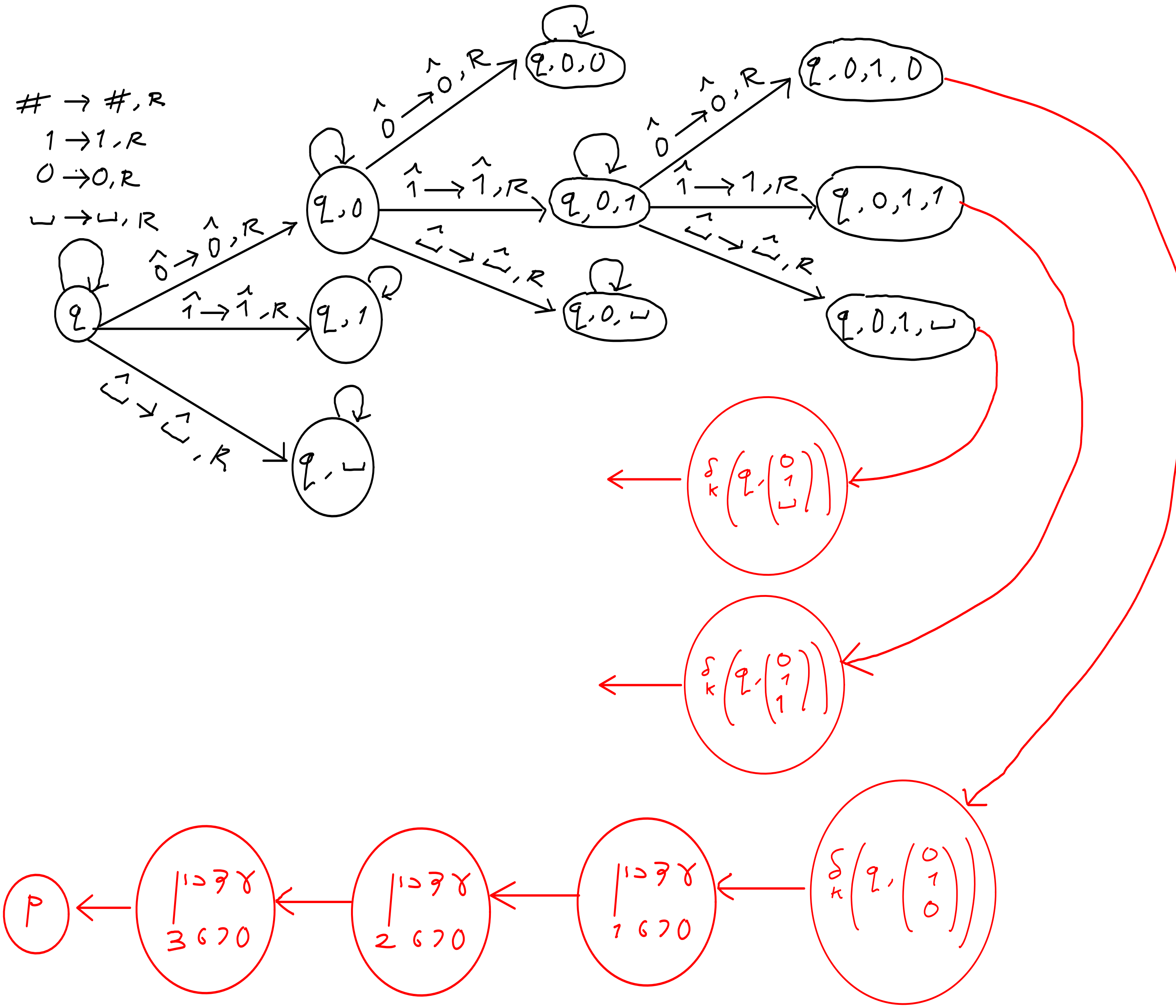
$$q\ ,\ \begin{pmatrix}0\\1\\0\end{pmatrix}\ .$$

זה אפשרי מכיוון שמספר המצבים הנדרש הוא סופי:

$$|Q|\times|\Gamma|^k\ .$$

מצב q

#	1	1	$\hat{0}$	1	#	1	$\hat{1}$	0	0	#	0	$\hat{0}$	1	#	␣
$\uparrow$															



מצב p

#	1	$\hat{1}$	1	1	#	1	0	$\hat{0}$	0	#	$\hat{0}$	0	1	#	␣

$\uparrow$

- <u>עדכון הסרטים</u>

M' סורקת את הסרט שלה פעם נוספת כדי לפעול על פי פונקצית המעברים, כלומר, לעדכן את התאים שמתחת לראשים ולעדכן את מיקום הראשים.

■

שיעור 4
מכונת טיורינג אי דטרמיניסטיתם

4.1 הגדרה של מכונת טיורינג אי-דטרמיניסטית

הגדרה 4.1 מכונת טיורינג אי-דטרמיניסטית

מכונת טיורינג אי-דטרמיניסטית (מ"ט א"ד) היא שביעייה

$$M = (Q, \Sigma, \Gamma, \Delta, q_0, q_{\text{acc}}, q_{\text{rej}})$$

כאשר $Q, \Sigma, \Gamma, q_0, q_{\text{acc}}, q_{\text{rej}}$ מוגדרים כמו במ"ט דטרמיניסטי (ראו הגדרה 1.2).

Δ היא פונקצית המעברים

$$\Delta : (Q \backslash \{q_{\text{acc}}, q_{\text{rej}}\}) \times \Gamma \to P\left(Q \times \Gamma \times \{L, R, S\}\right) \ .$$

$$\Delta(q, a) = \{(q_1, a, S) \ \ , \ \ (q_2, b, L) \ \ , \ \ \ldots\} \ .$$

כלומר, לכל זוג $q \in Q, \alpha \in \Gamma$ ייתכן מספר מעברים אפשריים, 0, 1 או יותר.

- קונפיגורציה של מ"ט א"ד זהה לקונפיגורציה של מ"ט דטרמיניסטית.
- לכל קונפיגורציה ייתכן מספר קונפיגורציות עוקבות.
- לכל מילה $w \in \Sigma^*$ ייתכן מספר ריצות שונות:
  * ריצות שמגיעות ל- q_{acc}.
  * ריצות שמגיעות ל- q_{rej}.
  * ריצות שלא עוצרות.
  * ריצות שנתקעות.

הגדרה 4.2

מילה $w \in \Sigma^*$ מתקבלת במ"ט א"ד M אם קיימת לפחות ריצה אחת שמגיעה ל- q_{acc}.

השפה של מ"ט א"ד M היא

$$L(M) = \left\{ w \in \Sigma^* \ \middle| \ \exists u, \mathrm{v} \in \Gamma^* \ \ : \ \ q_0 w \vdash_* \ u \ q_{\text{acc}} \ \mathrm{v} \right\}$$

כלומר,

$w \in L(M)$ אם קיימת ריצה אחת שבה M מקבלת את w.

$w \notin L(M)$ אם בכל ריצה של M על w, M דוחה או לא עוצרת, או נתקעת.

הגדרה 4.3 מ"ט א"ד המכריעה שפה L

תהי M מ"ט א"ד.
אומרים כי מ"ט א"ד M מכריעה שפה L אם לכל $w \in \Sigma^*$:

- אם $w \in L \Leftarrow M$ מקבלת את w.
- אם $w \notin L \Leftarrow M$ דוחה את w.

הגדרה 4.4 מ"ט א"ד המקבלת שפה L

תהי M מ"ט א"ד.
אומרים כי מ"ט א"ד M מקבלת שפה L אם לכל $w \in \Sigma^*$:

- אם $w \in L \Leftarrow M$ מקבלת את w.
- אם $w \notin L \Leftarrow M$ דוחה את w או M לא עוצרת על w.

דוגמה 4.1

נתונה השפה

$$L = \left\{1^n \;\middle|\; \text{אינו ראשוני } n\right\} \ , \qquad \Sigma = \{1\} \ .$$

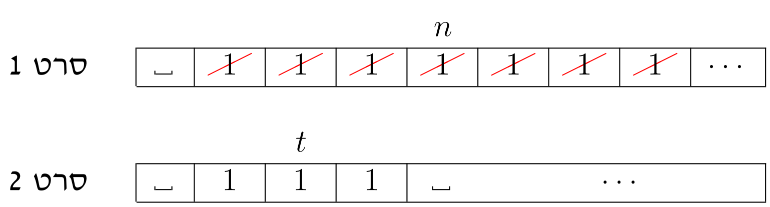
בנו מ"ט המכריעה את השפה L.

פתרון:

<u>הרעיון</u>

נבנה מ"ט א"ד N המכריעה את L.

N תבחר באופן א"ד מספר $1 < t < n$ ותבדוק האם t מחלק את n.



<u>תאור הבניה</u>

$N =$ על קלט $w = 1^n$:

שלב 1)

- N בוחרת באופן א"ד מספר $1 < t < n$.
- מעתיקה את w לסרט 2.
- עוברת על העותק משמאל לימין, ובכל תא מחליטה באופן א"ד האם להשאיר את ה- 1 או למחוק אותו ע"י X (לדאוג שהמספר שנבחר הוא לא 1 ולא n).

- בסוף המעבר המספר t שנבחר הוא כמות ה- 1 -ים שלא נמחקו.

$\sqcup$	1	1	1	1	$\cdots$	1	1	$\sqcup$	$\cdots$	סרט 1

		X	X			X				
$\sqcup$	1	$\cancel{1}$	$\cancel{1}$	1	$\cdots$	$\cancel{1}$	1	$\sqcup$	$\cdots$	סרט 2

שלב 2) N בודקת האם t שנבחר מחלק את n.

- אם כן $\Leftarrow$ N מקבלת.
- אם לא $\Leftarrow$ N דוחה.

4.2 עץ החישוב של מ"ט א"ד

הגדרה 4.5 עץ החישוב של מ""ט א"ד

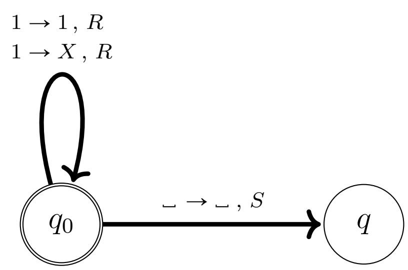
בהינתן מ"ט א"ד M ומילה $w \in \Sigma^*$, עץ החישוב של M ו- w הוא עץ מושרש שבו:

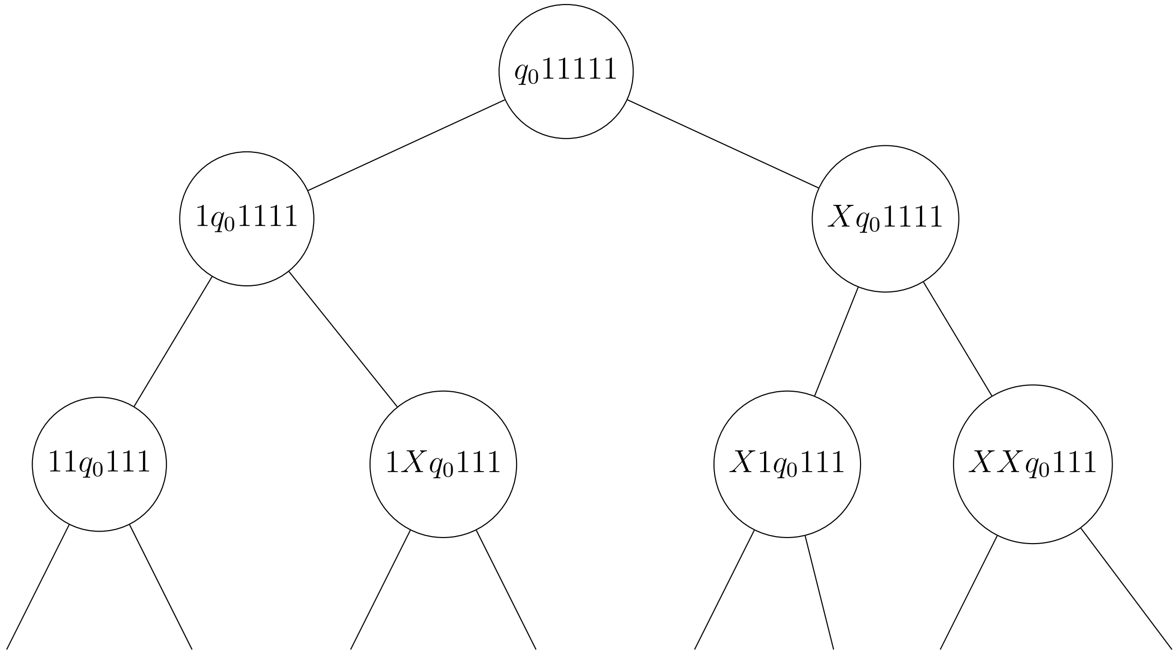
1) כל קדקוד בעץ מתאר קונפיגורציה בחישוב של M על w.

2) שורש העץ מתאר את הקונפיגורציה ההתחלתית q_0w.

3) לכל קדקוד v בעץ הבנים של v הם כל הקדקודים הנובעים מהקונפיגורציה המתוארת ע"י v.

דוגמה 4.2





4.3 שקילות בין מ"ט א"ד למ"ט דטרמיניסטית

משפט 4.1 שקילות בין מ"ט א"ד למ"ט דטרמיניסטית ב- RE

לכל מ"ט א"ד N קיימת מ"ט דטרמיניסטית D כך ש-

$$L(N) = L(D) \ .$$

כלומר לכל $w \in \Sigma^*$:

- אם N מקבלת את $w \Leftarrow$ D תקבל את w.
- אם N לא מקבלת את $w \Leftarrow$ D לא תקבל את w.

הוכחה: בהינתן מ"ט א"ד N נבנה מ"ט דטרמיניסטית D ונוכיח כי

$$L(N) = L(D) \ .$$

רעיון ההוכחה

בהינתן קלט $w \in \Sigma^*$, D תבצע ריצה של כל החישובים האפשריים של N על w, ואם אחד החישובים מסתיים ב- q_{acc} אז D תעצור ותקבל.

מכיוון שייתכנו חישובים אינסופיים, לא נוכל לסרוק את עץ החישוב לעומק. במקום זה נסרוק את העץ לרוחב. כלומר, נבדוק את כל החישובים באורך 1, ואחרי זה נבדוק את כל החישובים באורך 2, וכן הלאה. אם אחד החישובים הסתיים ב- q_{acc}, אזי D תעצור ותקבל.

תאור הבניה

מכיוון שלכל $q \in Q$ ולכל $\alpha \in \Gamma$:

$$\Delta(q,\alpha) \subseteq Q \times \Gamma \times \{L,R,S\} \ .$$

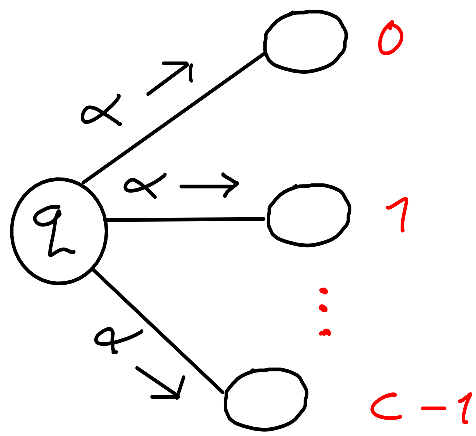
אזי

$$|\Delta(q,\alpha)| \leqslant |Q| \cdot |\Gamma| \cdot |\{L,R,S\}| = 3|Q| \cdot |\Gamma| \ .$$

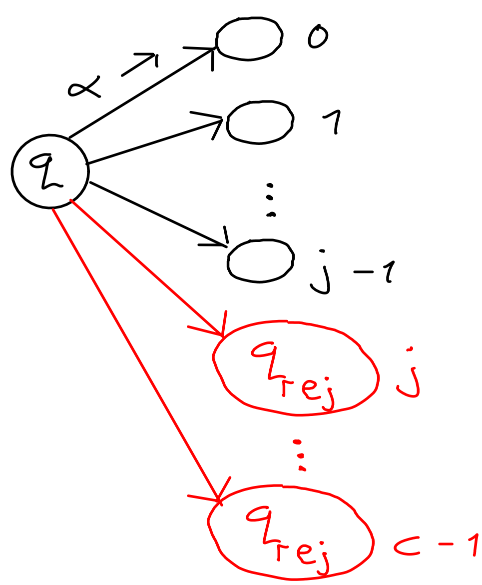
נסמן:

$$C = 3|Q| \cdot |\Gamma| \ .$$

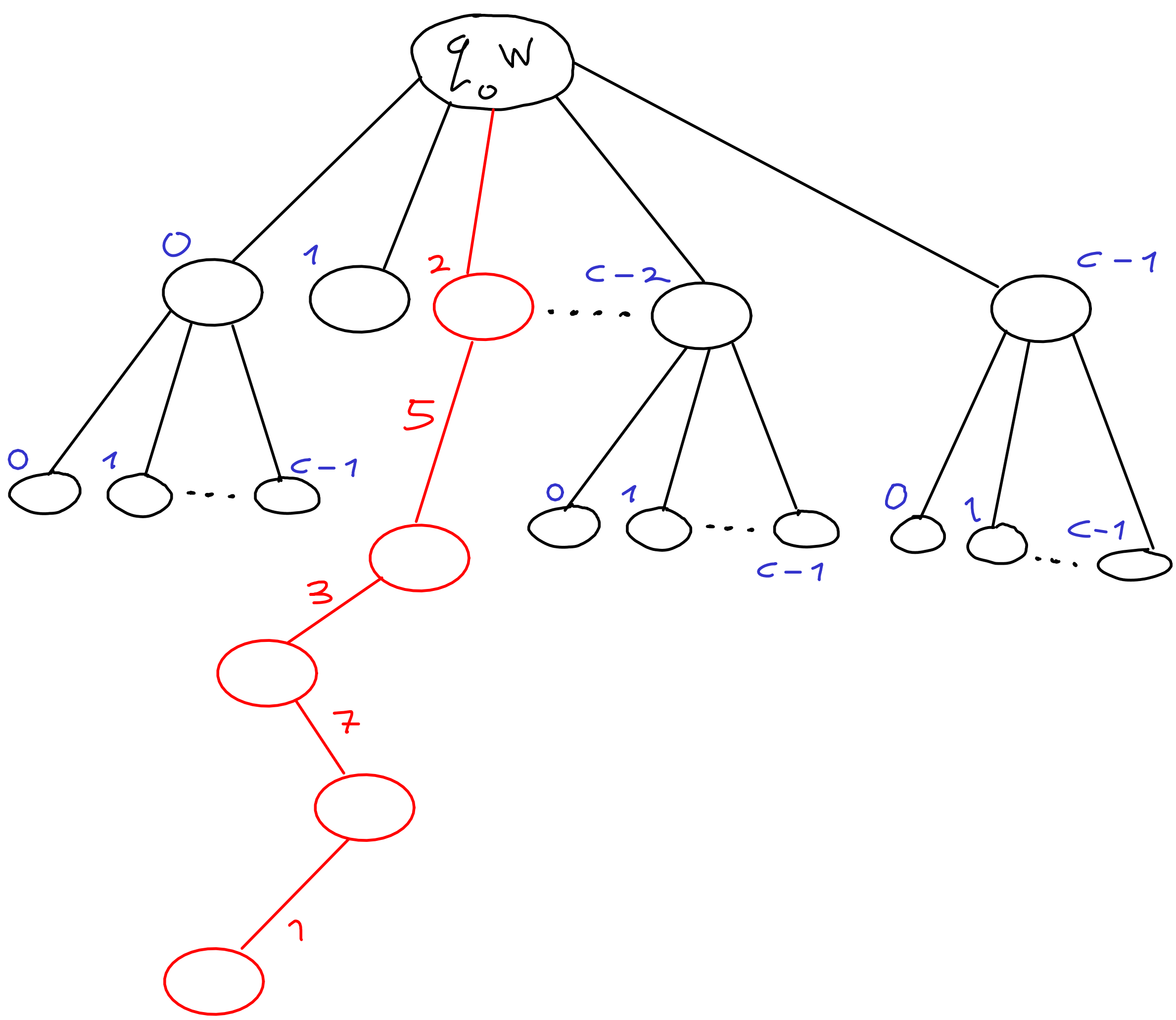
- לכל מצב $q \in Q$ ולכל אות $\alpha \in \Gamma$ נמספר את המעברים ב- $\Delta(q,\alpha)$ שרירותית
$$\{0,1,2,\cdots,C-1\} \ .$$



- אם $|\Delta(q,\alpha) = j < C$,
אזי לכל $j \leqslant k \leqslant C-1$
נקבע $k = (q_{\text{rej}}, \alpha, S)$.



- נשים לב כי שינוי זה לא משנה את השפה של N.



קידום לקסיקוגרפי:

0	00	10	$\cdots$	$(C-1)0$	000	
1	01	11	$\cdots$	$(C-1)1$	001	
2	02	12	$\cdots$	$(C-1)2$	002	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	
$C-1$	$0(C-1)$	$1(C-1)$	$\cdots$	$(C-1)(C-1)$	$00(C-1)$	

הבניה של D

D מכילה 3 סרטים:

					n				
סרט 1	␣	$\cancel{1}$	$\cancel{1}$	$\cancel{1}$	$\cancel{1}$	$\cancel{1}$	$\cancel{1}$	$\cancel{1}$	$\cdots$
			t						
סרט 2	␣	1	1	1	␣	$\cdots$			
סרט 3	␣	1	1	1	␣	$\cdots$			

$D =$ " על קלט w:

1) מאתחלת את המחרוזת בסרט 3 ל- 0.

2) מעתיקה את w לסרט 2.

3) מריצה את N על w לפי המחרוזת בסרט 3.

- אם N קיבלה את $w \Leftarrow D$ עוצרת ומקבלת.
- אחרת, D מוחקת את סרט 2, מקדמת את המחרוזת בסרט 3 לקסיקוגרפית וחוזרת לשלב 2).

"

■

שיעור 5
התזה של צרץ טיורינג ודקדוקים כלליים

5.1 היחס בין הכרעה וקבלה

משפט 5.1 כל שפה כריעה היא גם קבילה

כל שפה כריעה היא גם קבילה.

הוכחה: המכונה טיורינג שמכריעה את L גם מקבלת אותה. ■

נשאל שאלה. האם כל שפה קבילה היא גם כריעה? זאת שאלה שכרגע אין לנו מספיק כלים לענות עליה. נחזור לשאלה הזו בפרק הבא. לבינתיים נוכיח טענה חלשה יותר.

משפט 5.2

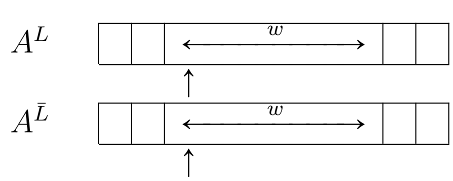
תהי L שפה.

אם גם L וגם $\bar{L}$ קבילות אזי L כריעה.

הוכחה: תהי A^L מ"ט שמקבלת את L, ותהי $A^{\bar{L}}$ מ"ט שמקבלת את $\bar{L}$.
נבנה מ"ט D^L שמכריעה את L.

כיצד תעבוד המ"ט D^L המכריעה?

- נריץ במקביל את A^L ואת $A^{\bar{L}}$.
- אם A^L מקבלת את המילה אז נעבור ל- acc.
- אם $A^{\bar{L}}$ מקבלת את המילה אז נעבור ל- rej.



■

- הסימולציה מתבצעת ע"י סימלוץ צעד צעד.
  * צעד במכונה A^L.
  * צעד במכונה $A^{\bar{L}}$.
- נמשיך בסימולציה המקבילית עד שאחת המכונות מגיעה למצב acc.

* אם A^L מקבלת ← acc.
* אם $A^{\bar{L}}$ מקבלת ← rej.

- לא יכול להיות מצב כי אף אחת מהמכונות לא מגיעה למצב acc כי כל מחרוזת $w \in L$ או $w \in \bar{L}$.

5.2 שקילות של מכונת טיורינג ותוכנית מחשב

- מכונת טיורינג היא מודל חישובי למחשב.
- מחשב = תוכנית מחשב.
- תוכנית משחב כתובה בשפת תכנות, למשל
  * ג'אווה
  * פייתון
  * C
  * SIMPLE
- המרכיבים של שפת תכנות הם
  * משתנים
  * פעולות
  * תנאים
  * זרימה

נוכיח כי מכונט טיורינג ותוכנית משחב שקולים חישובי.

5.3 SIMPLE

<u>משתנים</u>

- טבעיים:

```
i , j , k , . . . .
```

  מקבלים כערך מספר טבעי.
- אתחול: הקלט נמצא בתאים הראשונים של `A[]`
  כל שאר המשתנים מאותחלים ל- 0.

<u>פעולות</u>

- השמה בקבוע:

```
i =3,  B[ i ]="#"
```

- השמה בין משתנים:

```
i=k,    A[k]=B[ i ]
```

- פעולות חשבון:

```
x = y + z , x = y - z , x = y.z
```

תנאים

- `B[i]==A[j]` (מערכים).
- `x >= y` (משתנים טבעיים).

זרימה

- סדרה פקודות ממוספרות.
- goto : מותנה ולא מותנה.
- stop : עצירה עם ערך חזרה.

```
one = 1
zero = 0
B[zero] = "0"
i=0
j=i
if A[i] == B[zero] goto 9
i=j + one
goto 3
C[one] = A[j]
if C[one] == A[zero] goto 12
stop(0)
stop(1)
```

כעת נגדיר את מושגי הקבלה והדחייה של מילים בשפה SIMPLE,
ונגדיר את מושגי הכרעה והקבלה של שפות בשפה SIMPLE.

הגדרה 5.1 קבלה ודחייה של מחרוזת בשפה SIMPLE

עבור קלט w ותוכנית P בשפת SIMPLE. אומרים כי

- P **מקבלת** את w אם הריצה של P על w עוצרת עם ערך חזרה 1.
- P **דוחה** את w אם הריצה של P על w עוצרת עם ערך חזרה 0.

הגדרה 5.2 הכרעה וקבלה של מחרוזת בשפה SIMPLE

עבור שפה L ותוכנית P בשפת SIMPLE. אומרים כי

- P **מכריעה** את L אם היא מקבלת את המילים שב- L ודוחה את אלה שלא ב- L.
- P **מקבלת** את L אם היא מקבלת את כל ורק המילים ב- L.

משפט 5.3

המודלים של מכונת טיורינג ותוכניות SIMPLE שקולים.

הוכחה:

כיוון ראשון:

$\forall\ M\ \exists\ P$ שקולה .

במילים, לכל מ"ט M קיימת תוכנית P שקולה.
נבצע סימולציה של מ"ט M במשחב P .

בלי להכנס ,לפרטים די ברור שבשפה ,עילית כגון ג'אווה ,ניתן להגדיר מבני נתוניםעבור כל מרכיבי מכונת טיורינג:

- הסרט.
- המצבים.
- מיקום הראש.
- טבלת המעברים.

וברור שניתן לבצע סימולציה של פעילות המכונה.
ואם ניתן לעשות זאת בשפה עילית, ניתן לעשות זאת גם בשפת SIMPLE.

כיוון שני:

$\forall\ P\ \exists\ M$ שקולה .

במילים, לכל תוכנית P בשפה SIMPLE קיימת מ"ט שקולה.

אנחנו צריכים להראות כיצד ניתן לממש את הרכיבים השונים של תוכניות SIMPLE במ"ט.

הרכיבים הם:

- משתנים.
- פעולות.
- תנאים.
- זרימה.

משתנים

לכל משתנה יהיה סרט משלו.
המספר שהמשתנה יחזיק ייוצג בבסיס אונרי.
בהתחלה הסרט יהיה רק עם רווחים ,זה מייצג את המספר אפס בבסיס אונרי.

לכל מערך יהיה סרט משלו.
בכל תא בסרט המערך תהיה אות.
בהתחלה כל המערכים יהיו מאופסים למעט הסרט הראשון ,שיחזיק את הקלט.

למשל ההשמה הבאה של משתים בשפה SIMPLE:

```
A[1] = a, A[2] = b, A[3] = b ,  A[4] = a
B[1] = b, B[2] = a
i = 3
j = 1
k = 2
```

ניתן לממש במ"ט על ידי לכתוב על סרטים, שרט אחד לכל משתנה:

A[]	␣	a	b	b	a	␣
		↑				
B[]	␣	b	a	␣	␣	␣
		↑				
i	␣	1	1	1	␣	␣
		↑				
j	␣	1	␣	␣	␣	␣
		↑				
k	␣	1	1	␣	␣	␣
		↑				

<u>פעולות</u>

כעת נניח שנשים

j = k

אפשר לממש את ההשמה הזאת על ידי להעתיק את תוכן הסרט של המשתנה k לסרט של המשתנה j.

j	␣	1	␣	␣	␣	␣
		↑				
k	␣	1	1	␣	␣	␣
		↑				

j	␣	1	␣	␣	␣	␣
			↑			
k	␣	1	1	␣	␣	␣
			↑			

j	␣	1	1	␣	␣	␣
			↑			
k	␣	1	1	␣	␣	␣
			↑			

כעת נניח שנשים B[i]=A[j], ז"א B[3]=A[2].
נממש זה במ"ט ע"י להעתיק את תוכן משבצת 2 בסרט של A[] למשבצת 3 בסרט של B[].

שלב 1)

A[]	␣	a	b	b	a	␣
		↑				
B[]	␣	b	a	␣	␣	␣
		↑				
i	␣	1	1	1	␣	␣
		↑				
j	␣	1	1	␣	␣	␣
		↑				

שלב 2)

A[]	␣	a	b	b	a	␣
				↑		
B[]	␣	b	a	␣	␣	␣
		↑				
i	␣	1	1	1	␣	␣
		↑				
j	␣	1	1	␣	␣	␣
				↑		

שלב 3)

A[]	␣	a	b	b	a	␣
				↑		
B[]	␣	b	a	␣	␣	␣
					↑	
i	␣	1	1	1	␣	␣
					↑	
j	␣	1	1	␣	␣	␣
				↑		

שלב 4)

A[]	␣	a	b	b	a	␣
				↑		
B[]	␣	b	a	␣	b	␣
					↑	
i	␣	1	1	1	␣	␣
					↑	
j	␣	1	1	␣	␣	␣
				↑		

נניח עכשיו שאנחנו רוצים לשים

```
B[k] <- "a"
```

ז"א

```
B[3] <- "a"
```

נממש זה במ"ט ע"י על ידי הפעולות הבאות עם הסרט של B[] והסרט של k.

שלב 1)

B[]	␣	b	a	␣	b	␣
		↑				
k	␣	1	1	␣	␣	␣
		↑				

שלב 2)

B[]	␣	b	a	␣	b	␣
			↑			
k	␣	1	1	␣	␣	␣
			↑			

שלב 3)

B[]	␣	b	a	␣	b	␣
				↑		
k	␣	1	1	␣	␣	␣
				↑		

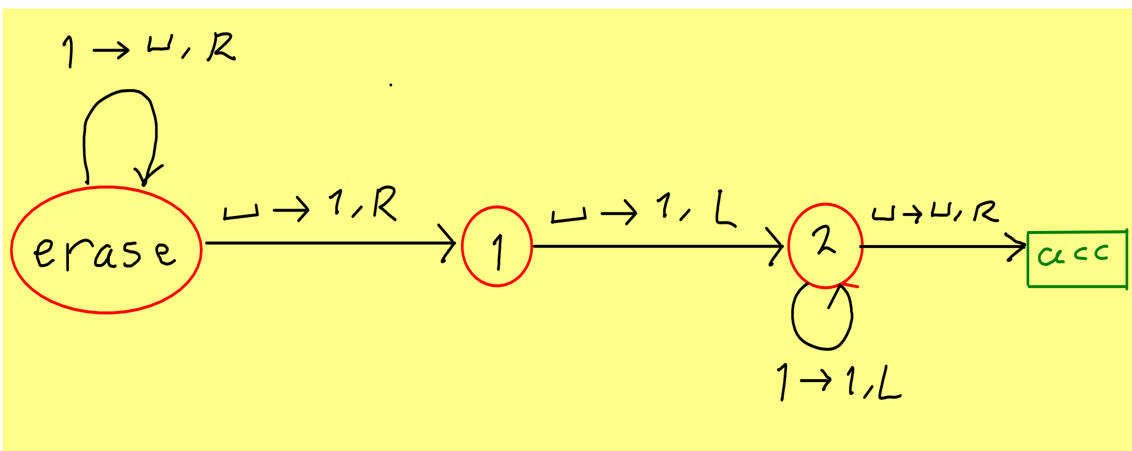
שלב 4)

B[]	␣	b	a	a	b	␣
				↑		
k	␣	1	1	␣	␣	␣
				↑		

כעת נניח שאנחנו רוצים לשים

j = 2

אז נממש זה במ"ט עם הפעולות הבאות:



שלב 1)

j	␣	1	␣	␣	␣	␣
		↑				

שלב 2)

j	␣	␣	␣	␣	␣	␣
			↑			

שלב 3)

j	␣	␣	1	␣	␣	␣
			↑			

שלב 4)

j	␣	␣	1	1	␣	␣
				↑		

שלב 5)

j	␣	␣	1	1	␣	␣
			↑			

שלב 6)

j	␣	␣	1	1	␣	␣
		↑				

שלב 7)

j	␣	␣	1	1	␣	␣
			↑			

<u>תנאים</u>

נניח שאנחנו רוצים לממש את התנאי

i >= k

ניתן לבדוק את התנאי במ"ט על ידי הפעולות הבאות:

שלב 1)

i	␣	1	1	1	␣	␣
		↑				
k	␣	1	1	␣	␣	␣
		↑				

שלב 2)

i	␣	1	1	1	␣	␣
			↑			
k	␣	1	1	␣	␣	␣
			↑			

שלב 3)

i	␣	1	1	1	␣	␣
				↑		
k	␣	1	1	␣	␣	␣
				↑		

■

5.4 דקדוקים כלליים

הגדרה 5.3 דקדוקים חסרי קשר

דקדוק חסר קשר הוא קבוצה
$$(V, \Sigma, R, S)$$
כאשר

- V קבוצה סופית של **משתנים** שמורכב מאותיות גדולות של אלפיבית.
- Σ קבוצה סופית של **טרמינלים** שמורכב מאותיות קטנות וסימנים של אלפיבית.
- R קבוצה של כללים. כל כלל הוא מצורה
$$\gamma \to u$$
כאשר $\gamma \in V$ משתנה בודד בצד שמאל ו- $u \in (V \cup \Sigma)^*$ מחרוזת של משתנים וטרמינילים בצד ימין
- $S \in V$ המשתנה ההתחלתי.

דוגמה 5.1

נתון הדקדוק חסר קשר:
$$G_1 = (\{A, B\}, \{0, 1, \#\}, R, A)$$
הקבוצת משתנים היא $V = \{A, B\}$, הקבוצת טרמינלים היא $V = \{0, 1, \#\}$, המשתנה ההתחלתי הוא $S = A$ והכללים של הדקדוק הם
$$R = \begin{cases} A \to & 0A1 \\ A \to & B \\ B \to & \# \ . \end{cases}$$

הגדרה 5.4 יצירה של מילה על ידי דקדוק חסר קשר

1) כתבו את המשתנה ההתחלתי S.

2) מצאו משתנה וכלל אשר מתחיל אם משתנה זה, והחליפו אותו עם המחרוזת בצד ימין של הכלל.

3) חזרו על שלבים 1 ו- 2 עד שלא נשאר אף משתנים של V.

דוגמה 5.2

הדקדוק G_1 יוצר את המחרוזת 000#111:
$$A \xrightarrow{A \to 0A1} 0A1 \xrightarrow{A \to 0A1} 00A11 \xrightarrow{A \to 0A1} 000A111 \xrightarrow{A \to B} 00B11 \xrightarrow{B \to \#} 000\#111$$

דוגמה 5.3

נתון את הדקדוק
$$G_2 = (\{S, T, F\}, \{(,), +, \times, a\}, R, S)$$
כאשר הכללים הם

$$R = \begin{cases} S \to & S+T \\ S \to & T \\ T \to & T\times F \\ T \to & F \\ F \to & (S) \\ F \to & a\ . \end{cases}$$

G_2 יוצר את המילה: $a+a$:

$$S \xrightarrow{S\to S+T} S+T \xrightarrow{SA\to T} T+T \xrightarrow{T\to F} F+F \xrightarrow{F\to a} a+a$$

בדקדוק כללי, גם בצד ימין וגם בצד שמאל יכולה להופיעמחרוזת של משתנים וטרמינליים. פורמלי:

הגדרה 5.5 דקדוקים כלליים

דקדוק חסר קשר הוא קבוצה

$$(V,\Sigma,R,S)$$

כאשר

- V קבוצה סופית של **משתנים** שמורכב מאותיות גדולות של אלפיבית.
- Σ קבוצה סופית של **טרמינלים** שמורכב מאותיות קטנות וסימנים של אלפיבית.
- R קבוצה של כללים. כל כלל הוא מצורה

$$\gamma \to u$$

כאשר $\gamma\in(V\cup\Sigma)^+$, $u\in(V\cup\Sigma)^*$. מחרוזת של משתנים וטרמינלים בצד ימין

- $S\in V$ המשתנה ההתחלתי.

דוגמה 5.4

נתון את הדקדוק

$$G=(\{\mathtt{S},\mathtt{[},\mathtt{]}\},\{\mathtt{a}\},R,\mathtt{S})$$

שבו הקבוצת משתנים היא $V=\{\mathtt{S},\mathtt{[},\mathtt{]}\}$, הקבוצת טרמינליים היא $\Sigma=\{\mathtt{a}\}$ והכללים הם

$$R = \begin{cases} \mathtt{S} \to & \mathtt{[S]} \\ \mathtt{S} \to & \mathtt{a} \\ \mathtt{[a} \to & \mathtt{aa[} \\ \mathtt{[]} \to & \varepsilon\ . \end{cases}$$

G יוצר את המילה: $\mathtt{aaaa}$:

$$\mathtt{S} \xrightarrow{\mathtt{S}\to\mathtt{[S]}} \mathtt{[S]} \xrightarrow{\mathtt{S}\to\mathtt{[S]}} \mathtt{[[S]]} \xrightarrow{\mathtt{S}\to\mathtt{a}} \mathtt{[[a]]} \xrightarrow{\mathtt{[a}\to\ \mathtt{aa[}} \mathtt{[aa[]]}$$

$$\xrightarrow{\mathtt{[]}\ \to\varepsilon} \mathtt{[aa]} \xrightarrow{\mathtt{[a}\to\ \mathtt{aa[}} \mathtt{aa[a]} \xrightarrow{\mathtt{[a}\to\ \mathtt{aa[}} \mathtt{aa\ aa[]} \xrightarrow{\mathtt{[]}\ \to\varepsilon} \mathtt{aaaa}$$

דוגמה 5.5

נתון את הדקדוק

$$G=(\{\mathtt{S},\mathtt{[},\mathtt{]}\},\{\mathtt{a}\},R,\mathtt{S})$$

שבו הקבוצת משתנים היא $V=\{\mathtt{S},\mathtt{[},\mathtt{]}\}$, הקבוצת טרמינליים היא $\Sigma=\{\mathtt{a}\}$ והכללים הם

$$
R = \begin{cases}
\mathtt{S} \to & [\mathtt{S}] \\
\mathtt{S} \to & \mathtt{a} \\
[\mathtt{a} \to & \mathtt{aa}[\\
[\,] \to & \varepsilon \ .
\end{cases}
$$

מהן המילים שניתן לצור בעזרת הדקדוק הזה,
או במילים אחרות :מהי השפה של הדקדוק?

פתרון:

תשובה:

$$L\left(G\right)=\left\{\mathtt{a}^n \;\middle|\; n=2^k\ ,\ k\geqslant 1\right\}\ .$$

הסבר:

$$
\begin{array}{lllllll}
\mathtt{S} & \xrightarrow{\mathtt{S}\to[\mathtt{S}]} & [\mathtt{S}] & \xrightarrow{\mathtt{S}\to[\mathtt{S}]} & \cdots & \xrightarrow{\mathtt{S}\to[\mathtt{S}]} & [^k\mathtt{S}]^k \\
 & \xrightarrow{\mathtt{S}\to\mathtt{a}} & [^k\mathtt{a}]^k & & & & \\
 & \xrightarrow{[\mathtt{a}\to\ \mathtt{aa}[} & [^{k-1}\mathtt{aa}[\,]^k & \xrightarrow{[\,]\to\varepsilon} & [^{k-1}\mathtt{aa}]^{k-1} & & \\
 & \xrightarrow{[\mathtt{a}\to\ \mathtt{aa}[} & [^{k-2}\mathtt{aa}[\mathtt{a}]^{k-1} & \xrightarrow{[\mathtt{a}\to\ \mathtt{aa}[} & [^{k-2}\mathtt{aaaa}[\,]^{k-1} & \xrightarrow{[\,]\to\varepsilon} & [^{k-2}\mathtt{aaaa}]^{k-2} \\
 & & & & \cdots & \to & \mathtt{a}^k
\end{array}
$$

דוגמה 5.6

בנו דקדוק כללי אשר יוצר את הפשה

$$L=\left\{w\in\{a,b\}^* \;\middle|\; \#a_w=\#b_w\right\}\ .$$

פתרון:

$$G=(\{\mathtt{S}\},\{\mathtt{a},\mathtt{b}\},R,\{\mathtt{S}\})$$

$$S \to \mathtt{abS}\ , \qquad (1)$$
$$\mathtt{ab} \to \mathtt{ba}\ , \qquad (2)$$
$$\mathtt{ba} \to \mathtt{ab}\ , \qquad (3)$$
$$\mathtt{S} \to \varepsilon\ . \qquad (4)$$

$$\mathtt{S} \xrightarrow{1} \mathtt{abS} \xrightarrow{1} \mathtt{ababS} \xrightarrow{2} \mathtt{baabS} \xrightarrow{4} \mathtt{baab}$$

שימו לב: בדקדוק כללי אנו מאפשרים גם כללייצירה בהם בצד שמאל יש רק טרמינלים.
לכן ,יתכן גם שנמשיך ונפתח מחרוזתשכולה טרמינליים. למשל

$$\mathtt{S} \xrightarrow{1} \mathtt{abS} \xrightarrow{1} \mathtt{ababS} \xrightarrow{4} \mathtt{abab} \xrightarrow{2} \mathtt{baab}$$

נשאל שאלה כללית:

- אלו שפות ניתן לצור בעזרת דקדוק כללי?
- האם יש שפות שלא ניתן לצור בעזרת דקדוק כללי?
- האם יש מודל חישובי שמקבל שפות שנוצרות ע"י דקדוקים כלליים?

דוגמה 5.7

בנו דקדוק כללי שיוצר את השפה

$$w = \{w \in \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}^* \mid w = \mathrm{a}^n \mathrm{b}^n \mathrm{c}^n\}$$

פתרון:

נראה דקדוק כללי עבור שפה זו.
שפה זו אינה חסרת הקשר.
לכן ,לא ניתן לבנות עבורה דקדוק חסר הקשר.
אנו נבנה לה דקדוק כלל.
נגזור את האותיות a,b,c יחד.
נעשה זאת בצורה כזו שכדי לסיים את תהליך הגזירה יש לסדר את האותיות בסדר הרצוי:
תחילה a,
אחר כך b,
ובסוף c.

$$\texttt{S} \to \texttt{S']} \qquad (1)$$

$$\texttt{S'} \to \texttt{aS'bC} \mid \varepsilon \qquad (2)$$

$$\texttt{Cb} \to \texttt{bC} \qquad (3)$$

$$\texttt{C]} \to \texttt{]c} \qquad (4)$$

$$\texttt{]} \to \varepsilon \qquad (5)$$

$$\begin{aligned}
\texttt{S} &\xrightarrow{1} \texttt{S']} \xrightarrow{2} \texttt{aS'bC]} \xrightarrow{2} \texttt{aaS'bCbC]} \xrightarrow{2} \texttt{aaaS'bCbCbC]} \\
&\xrightarrow{3} \texttt{aaaS'bbCCbC]} \xrightarrow{3} \texttt{aaaS'bbCbCC]} \xrightarrow{3} \texttt{aaaS'bbbCCC]} \\
&\xrightarrow{4} \texttt{aaaS'bbbCC]c} \xrightarrow{4} \texttt{aaaS'bbbC]cc} \xrightarrow{4} \texttt{aaaS'bbb]ccc} \\
&\xrightarrow{5} \texttt{aaaS'bbbccc} \xrightarrow{1} \texttt{aaabbbccc}
\end{aligned}$$

דוגמה 5.8

בנו דקדוק כללי אשר יוצר את שפת המילים

$$L = \{\ \mathrm{uu} \mid \mathrm{u} \in \{\mathrm{a}, \mathrm{b}\}^*\}$$

פתרון:

דוגמא זאת תמחיש ביצד דקדוק כללי יכול "לפעול בדומה" למכונת טיורינג.
בדקדוק נשתמש במשתנים וכללי גזירה שיאפשרו מעין תנועה על גבי המחרוזת הנגזרת, בדומה לתנועת הראש של מכונת טיורינג על גבי הסרט.

1	כלל גזירה יחיד מהמשתנה ההתחלתי. המשתנה H ידמה את הראש של המ"ט ש"יזוז" מצד לצד על המחרוזות הנגזרות. הסוגר המרובע] מסמן את הקצה השמאלי של המילה השמאלית. הסוגר המסולסל } מסמן את הקצה השמאלי של המילה הימנית.	$S \to [H\{$
2	כלל זה מאפשר הוספת אות a לקצה השמאלי של המילה השמאלית. המשתנה H מוחלף במשתנה H_a כדי "לזכור" שיש עכשיו להוסיף a גם במחרוזת הימנית. (בדומה לזיכרונות של מ"ט).	$[H \to [aH_a$
3	כלל זה מאפשר לראש "לזוז" ימינה.	$H_a a \to aH_a$
4	כאשר המשתנה H_a "יגיע" לסוגר המסולסל, הוא יגזור אות a נוספת מימין לסוגר, שהוא הקצה השמאלי של המחרוזת הימנית. כך אפשרנו להוסיף שתי אותיות a: אחת מימין לסוגר] ואחת תואם ימין לסוגר }. כלומר אות a בקצה השמאלי של כל אחת המחרוזות.	$H_a\{ \to H\{a$
5	כעת צריך לאפשר למשתנה H לחזור שמאלה על גבי האותיות שבין הסוגרים, עד הסוגר].	$aH \to Ha$

ברגע "שהראש" H חזר לתחילת המחרוזת ועומד ליד הסוגר] עוברים על השלבים 2-5 שוב. בסבב הבא נחק בחשבון גם יצירה של שתי אותיות b.

2'	כלל זה מאפשר הוספת אות b לקצה השמאלי של המילה השמאלית. המשתנה H מוחלף במשתנה H_b כדי "לזכור" שיש עכשיו להוסיף b גם במחרוזת הימנית.	$[H \to [bH_b$
3'	כללים האלה מאפשרים לראש "לזוז" ימינה.	$H_a a \to aH_a$ $H_a b \to bH_a$ $H_b a \to aH_b$ $H_b b \to bH_b$
4'	כאשר המשתנה H_b "יגיע" לסוגר המסולסל, הוא יגזור אות b נוספת מימין לסוגר, שהוא הקצה השמאלי של המחרוזת הימנית. כך אפשרנו להוסיף שתי אותיות b: אחת מימין לסוגר] ואחת תואם ימין לסוגר }. כלומר אות b בקצה השמאלי של כל אחת המחרוזות.	$H_b\{ \to H\{b$
5'	כעת צריך לאפשר למשתנה H לחזור שמאלה על גבי האותיות שבין הסוגרים, עד הסוגר].	$bH \to Hb$

בכדי לסיים את הגזירה יש להפטר ממשתני העזר על ידי הכללים הבאים:

6	הכללים האלה מאפשרים להעלים את המשתנים { ,] ,H	$H \to \varepsilon$ $[\to \varepsilon$ $\{ \to \varepsilon$

למשל:

$$S \xrightarrow{1} [H\{ \xrightarrow{2} [aH_a\{ \xrightarrow{4} [\ aH\{a \xrightarrow{5} [\ Ha\{a$$

$$\xrightarrow{2} [aH_a a\{a \xrightarrow{3} [aaH_a\{a \xrightarrow{4} [aa\ H\{aa \xrightarrow{5} [Haa\ \{aa$$

$$\xrightarrow{2} [bH_b aa\{aa \xrightarrow{3} [baaH_b\{aa \xrightarrow{4} [baa\ H\{baa \xrightarrow{5} [Hbaa\ \{baa$$

$$\xrightarrow{6} baabaa$$

5.5 דקדוקים כלליים ומכונת טיורינג

משפט 5.4 קדוקים כלליים ומכונת טיורינג

תהי L שפה. L קבילה אם ורק אם קיים דקדוק כללי G כך ש- $L(G)=L$.

הוכחה: **כיוון ראשון.**

נוכיח שאם קיים דקדוק כללי G אז $L(G)$ קבילה.

נניח שקיים דקדוק כללי G. נוכיח כי $L(G)$ קבילה על ידי להוכיח שקיימת תוכנית מחשב P שמקבלת $L(G)$.

נתון דקדוק כללי G. נבנה תוכנית מחשב שמקבלת את $L(G)$.
יהי הקלט $w\in L(G)$, כלומר w מילה בשפה $L(G)$.

(1 `u=S`.

(2 `repeat`:

- פצל באופן לא דטרמיניסטי את `u` ל- `xyz`.
- בחר באופן לא דטרמיניסטי גזירה `v` $\to$ `t` של G.
- אם `t` $\neq$ `y` דחה.
- `u=xvz`
- אם `w==u` קבל.

כיוון שני.

נוכיח שאם $L(G)$ קבילה אז קיים דקדוק כללי G.

ז"א, נניח שקיימת מ"ט M שמקבלת את השפה L. נוכיח שקיים דקדוק כללי G כך ש- $L(G)=L(M)$, כלומר השפה המתקבלת על ידי M היא השפה של דקדוק כללי G.

נתונה מ"ט M בעלת הטבלת המעברים להלן. נבנה דקדוק כללי G שמממש אותם צעדים.

מצב	סימן	מצב חדש	כתיבה	תזוזה
q_0	a	q_0	a	R
q_0	b	q_1	a	R
q_0	␣	acc	␣	L
q_1	a	q_0	b	L
q_1	b,␣	q_1	b	L

לפי הטבלת המעברים קיים הצעד

$$\mathtt{q}\ q_0\ \mathtt{b}\ \ \mathtt{a}\ \ \mathtt{b} \vdash_M \ \ \mathtt{aa}q_1\ \mathtt{ab}$$

נניח שבדקדוק כללי G קיים אותו הצעד

$$\mathtt{q}\ q_0\ \mathtt{bab} \xrightarrow{G} \ \ \mathtt{aa}q_1\ \mathtt{ab}$$

ניתן לממש צעד זה על ידי הכלל

$$q_0 \ \mathrm{b} \to \mathrm{a} \ q_1$$

באופן כללי,

- עבור כל פונקצית המעברים של M שגוררת תזוזה ימינה מצורה

  $$\delta(q,\sigma) = (p,\pi,R)$$

  נממש מעבר זה על ידי כלל של הדקדוק G מצורה

  $$q\sigma \to \pi p \ .$$

- עבור כל פונקצית המעברים של M שגוררת תזוזה שמאלה מצורה

  $$\delta(q,\sigma) = (p,\pi,L)$$

  אז לכל $\tau \in \Gamma$ ב- G נממש מעבר זה על ידי הכלל

  $$\tau q\sigma \to p\tau\pi \ .$$

■

5.6 ההיררכיה של חומסקי

משפחת שפות	דקדוק	מודל חישובי
קבילות	כללי	מכונת טיורינג
חסרות הקשר	חסר הקשר	אוטומט מחסנית
רגולריות	רגולרי	אוטומט סופי

- היררכיה של חומסקי קושרת לנו בין משפחותשל ,שפות דקדוקים ומודלים חישוביים.
- בתחתית ההיררכיה נמצאות השפות הרגולריות שנוצרות על ידי דקדוקים רולריים ומתקבלות על ידי אוטומטים סופיים.
- מעליהן נמצאות השפות חסרות ההקשר שנוצרות על ידי דקדוקים חסרי הקשר ומתקבלות על ידי אוטומטי מחסנית.
- ומעליהן נמצאות השפות הקבילות שנוצרותעל ידי דקדוקים כלליים ומתקבלותעל ידי מכונות טיורינג.
- כל רמה בהיררכיה מכילה ממש את הרמה שמתחתיה.
  * כל שפה רגולרית היא גם חסרתהקשר ,אבל יש שפות חסרות הקשרשאינן רגולריות.
  * כל שפה חסרת הקשר היא קבילה, אבל יש שפות קבילות שאינן חסרות הקשר.

5.7 כל שפה חסרת הקשר הינה כריעה

לפי ההיררכיה של חומסקי אנחנו יודעים לקבוע שכל שפה חסרת הקשר היא קבילה.

האם כל שפה חסרת הקשר הינה כריעה?

משפט 5.5

יהי $G=(V,\Sigma,S,R)$ דקדוק חסר הקשר ו- $w\in L(G)$. אזי קיים עץ גזירה של w שעומקו לכל היותר $(|V|+1)(|w|+1)$.

הוכחה: יהי T עץ הגזירה הקטן ביותר (מבחינת מספר קדקודים) של w.

בשלילה נניח שב- T יש מסלול מהשורש לעלה שמכיל לפחות $(|V|+1)(|w|+1)$ קודקודים פנימיים.

נסמן מסלול זה ב-

$$p=(u_1,u_2,\ldots,u_m)\ .$$

עבור קודקוד u_i במסלול נסמן ב- $s(u_i)$ את תת-המחרוזת של w שנוצרת מ- u_i.

מתקיים ש- $s(u_{i+1})$ היא תת-מחרוזת של $s(u_i)$. אומרים שקדקוד u_i הוא משמעותי אם $s(u_i)$ מכיל ממש את $s(u_{i+1})$.

כל קודקוד משמעותי מוסיף לפחות אות אחת ל- w.
לכן, ישנם לכל היותר $|w|$ קודקודים משמעותיים.
לכן, ברצף הקודקודים הפנימיים $(u_1,u_2,\ldots,u_m)$ שאורכן לפחות $(|V|+1)(|w|+1)$, בהכרח ישנו תת רצף $(u_i,u_{i+1},\ldots,u_{i+|V|})$ באורך $|V|+1$, שבו כל הקודדוקים לא משמעותיים.
ברצף זה בהכרח ישנם שני קודקודים, נאמר u_j,u_k, $j<k$ שמסומנים עם אותו משתנה.
לכן בעץ הגזירה, ניתן להחליף את הקודקוד u_j יחד עם כל תת העץ שמתחתיו- בקודקוד u_k, יחד עם כל תת העץ שמתחתיו.
כיוון שכל הקודקודים שבין u_j ל- u_k (כולל) הם לא משמעותיים, **החלפה זו לא משנה את המחרוזת הנוצרת.**

כלומר, העץ החדש גם הוא עץ הגזירה עבור w.
בסתירה להנחת המינימליות של העץ.

■

משפט 5.6

כל שפה חסרת הקשר היא כריעה.

הוכחה: בהינתן דקדוק חסר הקשר $G=(V,\Sigma,S,R)$, התוכנית הלא דטרמיניס הבאה מכריעה את $L(G)$.

קלט: מחרוזת w.
פלט: כן או לא.

1) נחש עץ גזירה של הדקדוק G בעומק לכל היותר $(|V|+1)(|w|+1)$.

2) בדוק האם העץ יוצר את המחרוזת w. אם כן, החזר "כן" איתר החזר "לא".

שני שלבי התוכנית בהכרח מסתיימים. לכן, זו תוכנית להכרעה. ישנו חישוב שמחזיר "כן" אם ורק אם $w\in L(G)$ לכן זו תוכנית שמכריעה את $L(G)$.

■

שיעור 6
תכונות סגירות של R ו- RE

6.1 הגדרה של השפות R ו- RE

הגדרה 6.1 R

אוסף השפות הכריעות מסומן R ומוגדר

$$R = \{L \subseteq \Sigma^* \ : \ \text{קיימת מ"ט המכריעה את } L\} \ .$$

הגדרה 6.2 RE

אוסף השפות הקבילות מסומן RE ומוגדר

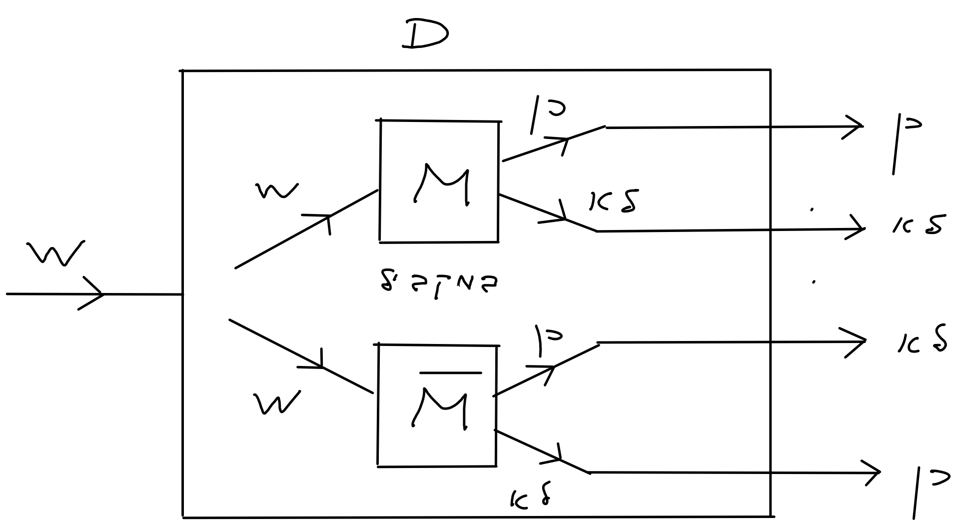
$$RE = \{L \subseteq \Sigma^* \ : \ \text{קיימת מ"ט המקבלת את } L\} \ .$$

למה 6.1

אם $L \in RE$ וגם $\bar{L} \in RE$ אזי $L \in R$.

הוכחה: תהי M מ"ט המקבלת את L ותהי $\bar{M}$ מ"ט המקבלת את $\bar{L}$.

נבנה מ"ט D המכריעה את L.



$D =$ על קלט w:

1) D מעתיקה את w לסרט נוסף.

2) מריצה במקביל את M על w ואת $\bar{M}$ על העותק של w.

- אם M מקבלת $\Leftarrow$ D מקבלת.
- אם $\bar{M}$ מקבלת $\Leftarrow$ D דוחה.
- אם M דוחה $\Leftarrow$ D דוחה.
- אם $\bar{M}$ דוחה $\Leftarrow$ D מקבלת.

נוכיח כי D מכריעה את L.

אם $w \in L$

$\Leftarrow$ $w \in L(M)$

$\Leftarrow$ (M מקבלת את w) או ($\bar{M}$ דוחה את w)

$\Leftarrow$ D עוצרת ומקבלת את w.

אם $w \notin L$

$\Leftarrow$ $w \in \bar{L}$

$\Leftarrow$ $w \in L\left(\bar{M}\right)$

$\Leftarrow$ ($\bar{M}$ מקבלת את w) או (M דוחה את w)

$\Leftarrow$ D עוצרת ודוחה את w.

■

משפט 6.1 סגירות של השפה הכריעות

R סגורה תחת:

1) איחוד

2) חיתוך

3) משלים

4) שרשור

5) סגור קלין

משפט 6.2 סגירות של השפה הקבילות

RE סגורה תחת:

1) איחוד

2) חיתוך

3) שרשור

4) סגור קלין

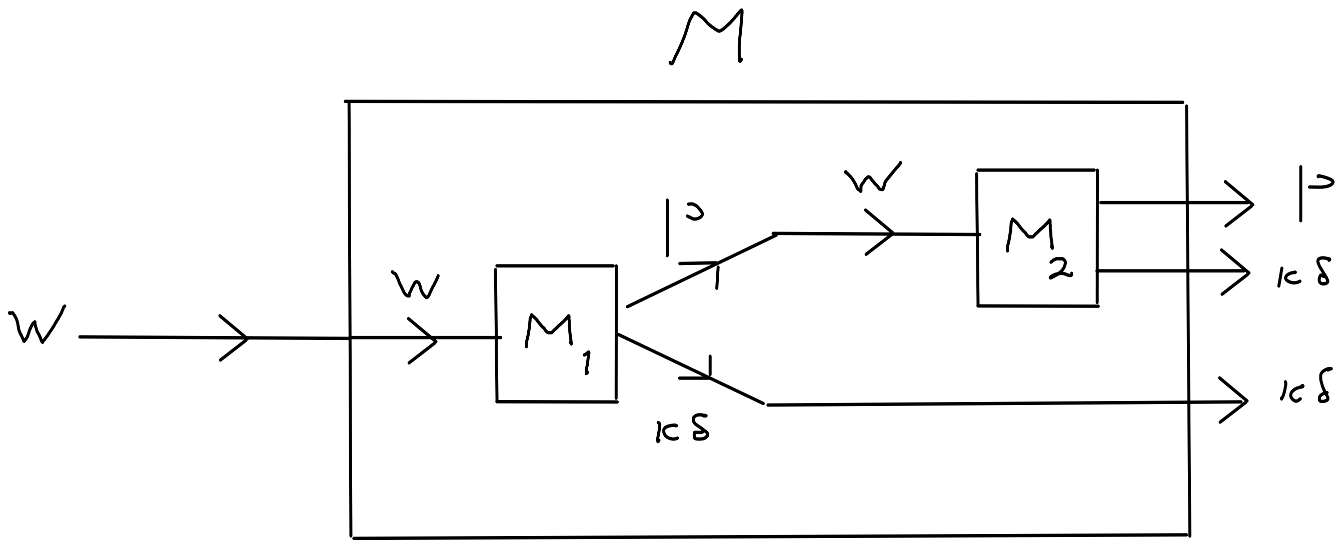
הוכחה:

1) חיתוך:

(א) R סגורה תחת חיתוך

נוכיח כי לכל שתי שפות $L_1, L_2 \in R$ מתקיים $L_1 \cap L_2 \in R$.

תהי M_1 ו- M_2 מ"ט המכריעות את L_1 ו- L_2 בהתאמה. נבנה מ"ט M המכריעה את $L_1 \cap L_2$.



תאור הבנייה

$M =$ על קלט w:

1) M מעתיקה את w לסרט נוסף.
2) M מריצה את M_1 על w.
- אם M_1 דוחה $\Leftarrow$ M דוחה.
- אחרת M מריצה את M_2 על העותק של w ועונה כמוה.

נכונות:

נוכיח כי M מכריעה את $L_1 \cap L_2$.

אם $w \in L_1 \cap L_2$

$\Leftarrow$ $w \in L_1$ וגם $w \in L_2$

$\Leftarrow$ M_1 מקבלת את w וגם M_2 מקבלת את w

$\Leftarrow$ M מקבלת את w.

אם $w \notin L_1 \cap L_2$

$\Leftarrow$ $w \notin L_1$ או $w \notin L_2$

$\Leftarrow$ M_1 דוחה את w או M_2 דוחה את w

$\Leftarrow$ M דוחה את w.

(ב) <u>RE **סגורה תחת חיתוך**</u>

נוכיח כי לכל שתי שפות $L_1, L_2 \in RE$ מתקיים $L_1 \cap L_2 \in RE$.

תהיינה M_1 ו- M_2 שתי מכונות טיורינג המקבלות את L_1 ו- L_2 בהתאמה.
נבנה מ"ט M המקבלת את $L_1 \cap L_2$ באותו אופן כמו **(א)**.

2) איחוד:

(א) <u>R **סגורה תחת איחוד**</u>

נוכיח כי לדל שתי שפות $L_1, L_2 \in R$ מתקיים $L_1 \cup L_2 \in R$.

תהיינה M_1 מ"ט המכריעה את L_1 ו- M_2 מ"ט המכריעה את L_2.
נבנה מ"ט M המכריעה את $L_1 \cup L_2$.

<u>תאור הבנייה</u>

$M =$ על קלט w:

1) M מעתיקה את w לסרט נוסף.
2) M מריצה את M_1 על w.
- אם M_1 מקבלת $\Leftarrow$ M מקבלת.
- אחרת, M מריצה את M_2 על העותק של w ועונה כמוה.

(ב) <u>RE **סגורה תחת איחוד**</u>

נוכיח כי לכל שתי שפות $L_1, L_2 \in RE$ מתקיים $L_1 \cup L_2 \in RE$.
תהיינה M_1 מ"ט המקבלת את L_1 ו- M_2 מ"ט המקבלת את L_2.
נבנה מ"ט א"ד M המקבלת את $L_1 \cup L_2$.

<u>תאור הבנייה</u>

$M =$ על קלט w:

1) M בוחרת באופן א"ד $i \in \{1, 2\}$.
2) M מריצה את M_i על w ועונה כמוה.

3) שרשור:

(א) <u>R **סגורה תחת שרשור**</u>

נוכיח כי לכל שתי שפות $L_1, L_2 \in R$ מתקיים $L_1 \cdot L_2 \in R$ כאשר

$$L_1 \cdot L_2 = \left\{w = w_1 w_2 \;\middle|\; w_1 \in L_1 \wedge w_2 \in L_2\right\} .$$

תהיינה M_1 מ"ט המכריעה את L_1 ו- M_2 מ"ט המכריעה את L_2.
נבנה מ"ט א"ד M המכריעה את $L_1 \cdot L_2$.

<u>תאור הבנייה</u>

$M =$ על קלט w:

1) M בוחרת באופן א"ד חלוקה של w ל- $w = w_1 w_2$.
2) M מריצה את M_1 על w_1.
- אם M_1 דוחה $\Leftarrow M$ דוחה.
- אחרת, M מריצה את M_2 על w_2 ועונה כמוה.

(ב) RE **סגורה תחת שרשור**

RE סגורה תחת שרשור באותו אופן כמו ב- **(א)**

4) $*$ קליני

א) R **סגורה תחת $*$ קליני**

נוכיח כי לכל שפה L:

$$L \in R \quad \Rightarrow \quad L^* \in R$$

כאשר

$$L^* = \left\{ w = w_1 w_2 \cdots w_k \;\middle|\; \forall 1 \leqslant i \leqslant k \;,\; w_i \in L \right\} \;.$$

תהי M מ"ט המכריעה את L.
נבנה מ"ט M^* א"ד המכריעה את L^*.

תאור הבנייה

$M^* =$ על קלט w:

1) אם $w = \varepsilon$ אז M^* מקבלת.
2) אחרת M^* בוחרת באופן א"ד חלוקה של w ל- $w = w_1 \cdots w_k$.
3) לכל $1 \leqslant i \leqslant k$:

M^* מריצה את M על w_i.

- אם M דוחה את $w_i \Leftarrow M^*$ דוחה.
- אחרת חוזרים לשלב 3).

4) אם M קיבלה את כל המחרוזות $\{w_i\}$ אזי M^* מקבלת.

ב) RE **סגורה תחת $*$ קליני**

5) משלים

א) R **סגורה תחת המשלים**

נוכיח כי

$$L \in R \quad \Rightarrow \quad \bar{L} \in R \;,$$

כאשר

$$\bar{L} = \left\{ w \in \Sigma^* \;\middle|\; w \notin L \right\} \;.$$

תהי M מ"ט המכריעה את L.
נבנה מ"ט $\bar{M}$ המכריעה את $\bar{L}$.

$\bar{M} =$ על קלט w:

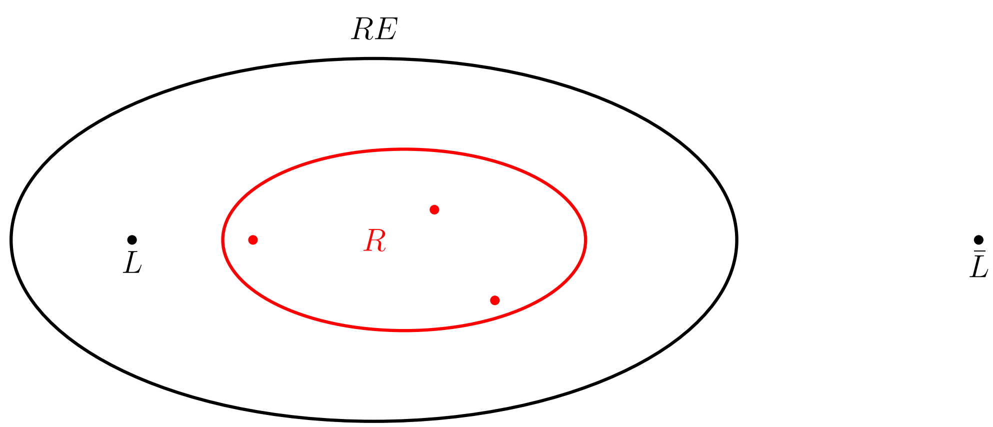
(1) $\bar{M}$ מריצה את M על w.
- אם M מקבלת $\Leftarrow$ $\bar{M}$ דוחה.
- אם M דוחה $\Leftarrow$ $\bar{M}$ מקבלת.

ב) <u>**RE אינה סגורה תחת המשלים**</u>

■

משפט 6.3 RE אינה סגורה תחת המשלים

$$L \in RE \backslash R \quad \Rightarrow \quad \bar{L} \notin RE \ .$$



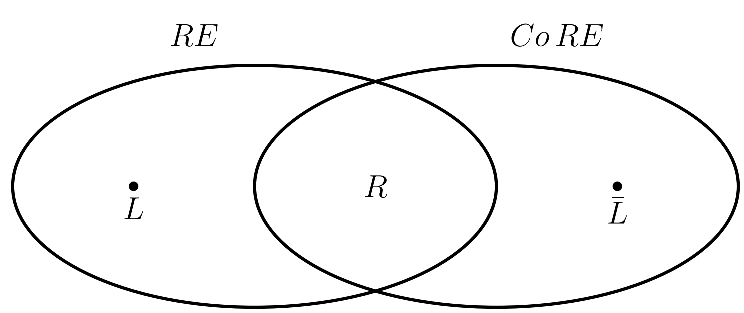
הוכחה:
נניח כי $L \in RE \backslash R$ ונניח בשלילה כי $\bar{L} \in RE$.

אזי לפי טענת עזר (למה 6.1), $L \in R$ וזו סתירה.

■

הגדרה 6.3 $Co\,RE$

$$Co\,RE = \left\{ L \subseteq \Sigma^* \mid \bar{L} \in RE \right\} \ .$$



<u>אבחנה</u>

לפי למה 6.1:

$$RE \cap Co\,RE = R \ .$$

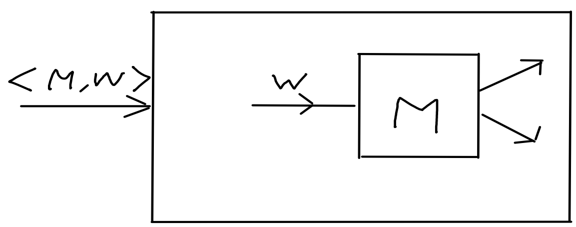
6.2 קידוד של מ"ט דטרמיניסטית

הגדרה 6.4 קידוד של מ"ט

בהינתן קבוצה O של עצמים מופשטים (למשל מכונת טיורינג, תוכנית מחשב, גרף).
הקידוד של O, מסומן $\langle O\rangle$, הוא מיפוי של O אל מחרוזת מעל אלפבית סופי שיש בו לפחות שני סימנים.

במידה ויש רב עצמים $O_1,\ldots,O_k$ נסמן את הקידוד שלהם $\langle O_1,O_2,\ldots,O_k\rangle$.

6.3 מ"ט אוניברסלית U



מ"ט אוניברסלית U מקבלת כקלט זוג, קידוד של מ"ט $\langle M\rangle$ וקידוד של מילה $\langle w\rangle$, ומבצעת סימוציה של ריצה של M על w ועונה בהתאם.

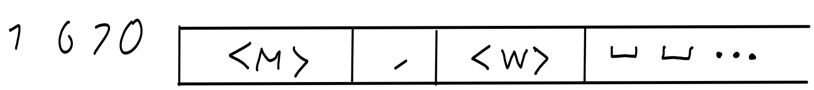
<u>תאור הפעולה של U</u>

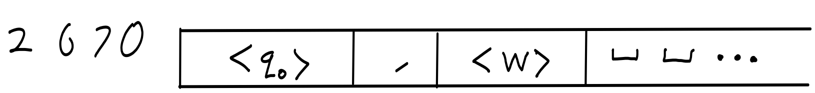
$U=$ על קלט x:

(1) בודקת האם x הוא קידוד של מ"ט $\langle M\rangle$ וקידוד של מילה $\langle w\rangle$.

- אם לא $\Leftarrow$ דוחה.

(2) מבצעת סימולציה של M על w:





- רושמת את הקונפיגורציה ההתחלתית q_0w על סרט 2.
- מחשבת את הקונפיגורציה הבאה בעזרת טבלת המעברים.
- בסוף כל מעבר בין שתי קונפיגורציות, U בודקת האם המצב הנוכחי הוא q_{acc}.
  * אם כן U עוצרת ומקבלת.

* אחרת U בודקת האם המצב הוא q_{rej}.
* אם כן U עוצרת ודוחה.
* אחרת U ממשיכה לקונפיגורציה הבאה.

<u>מהי השפה של U?</u>

לכל x:

(1) אם $x \neq \langle M,w\rangle \Leftarrow U$ דוחה את x.

(2) אם $x = \langle M,w\rangle$:

- אם M מקבלת $w \Leftarrow U$ מקבלת את x.
- אם M דוחה את $w \Leftarrow U$ דוחה את x.
- אם M לא עוצרת על $w \Leftarrow U$ לא עוצרת על x.

$$L(U) = \{\langle M,w\rangle \mid w \in L(M)\} \ .$$

הגדרה 6.5 L_{acc}

$$L_{\text{acc}} = \{\langle M,w\rangle \mid w \in L(M)\} \in RE \backslash R$$

הגדרה 6.6 L_{halt}

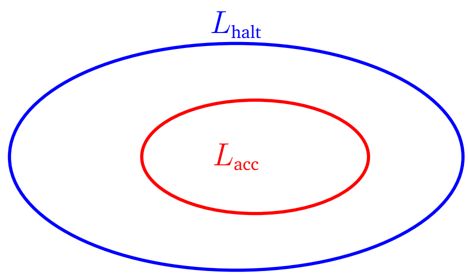
$$L_{\text{halt}} = \{\langle M,w\rangle \mid \text{עוצרת על } w \ M\} \in RE \backslash R$$

הגדרה 6.7 L_{d}

$$L_{\text{d}} = \{\langle M\rangle \mid \langle M\rangle \notin L(M)\} \notin RE$$

<u>אבחנה:</u>

$$L_{\text{acc}} \subseteq L_{\text{halt}} \ .$$



משפט 6.4

$$L_{\text{acc}} \in RE\ .$$

הוכחה: מכיוון ש- $L(U) = L_{\text{acc}}$, U מקבלת את L_{acc} ולכן $L_{\text{acc}} \in RE$. ■

משפט 6.5

$$L_{\text{halt}} \in RE\ .$$

הוכחה: נבנה מ"ט U' שהיא למעשה U פרט למקום שבו U עצרה ודחתה, U' תעצור ותקבל.

נוכיח כי U' מקבלת את L_{halt}:

אם $x \in L_{\text{halt}}$

$\Leftarrow$ $x = \langle M, w\rangle$ ו- M עוצרת על w

$\Leftarrow$ U' עוצרת ומקבלת את x.

אם $x \notin L_{\text{halt}}$ $\Leftarrow$ שני מקרים:

- $x \neq \langle M, w\rangle$ $\Leftarrow$ U' דוחה את x.
- $x = \langle M, w\rangle$ ו- M לא עוצרת על w $\Leftarrow$ U' לא עוצרת על x.

■

שיעור 7
אי-כריעות

7.1 השפות L_{acc}, L_{halt}, L_{d} לא כריעות

הגדרה 7.1 L_{acc}

$$L_{\text{acc}} = \left\{\langle M, w\rangle \;\middle|\; w \in L(M)\right\} \in RE \backslash R$$

הגדרה 7.2 L_{halt}

$$L_{\text{halt}} = \left\{\langle M, w\rangle \;\middle|\; \text{עוצרת על } w \; M\right\} \in RE \backslash R$$

הגדרה 7.3 L_{d}

$$L_{\text{d}} = \left\{\langle M\rangle \;\middle|\; \langle M\rangle \notin L(M)\right\} \notin RE$$

משפט 7.1 $L_{\text{acc}} \in RE$

$$L_{\text{acc}} \in RE\ .$$

הוכחה: מכיוון ש- $L(U) = L_{\text{acc}}$, כאשר U המכונת טיורינג האוניברסלית אשר מקבלת את L_{acc}, לכן $L_{\text{acc}} \in RE$. ■

משפט 7.2 $L_{\text{halt}} \in RE$

$$L_{\text{halt}} \in RE\ .$$

הוכחה: נבנה מ"ט U' שהיא למעשה U פרט למקום שבו U עצרה ודחתה, U' תעצור ותקבל.

נוכיח כי U' מקבלת את L_{halt}:

אם $x \in L_{\text{halt}}$

$\Leftarrow$ $x = \langle M, w\rangle$ ו- M עוצרת על w

$\Leftarrow$ U' עוצרת ומקבלת את x.

אם $x \notin L_{\text{halt}} \Leftarrow$ שני מקרים:

- $x \neq \langle M, w\rangle \Leftarrow U'$ דוחה את x.
- $x = \langle M, w\rangle$ ו- M לא עוצרת על $w \Leftarrow U'$ לא עוצרת על x.

■

משפט 7.3 $L_{\text{d}} \notin RE$

$$L_{\text{d}} \notin RE \ .$$

הוכחה:
נניח בשלילה כי $L_{\text{d}} \in RE$.

$\Leftarrow$ ∃ מ"ט M_{d} המקבלת את L_{d}.

$\Leftarrow$ $L(M_{\text{d}}) = L_{\text{d}}$.

נבדוק ריצה של M_d על $\langle M_d\rangle$:

- אם $\langle M_{\text{d}}\rangle \in L(M_{\text{d}}) \quad \Leftarrow \quad \langle M_{\text{d}}\rangle \notin L_{\text{d}} \quad \Leftarrow \quad L(M_{\text{d}}) \neq L_{\text{d}}$.
- אם $\langle M_{\text{d}}\rangle \notin L(M_{\text{d}}) \quad \Leftarrow \quad \langle M_{\text{d}}\rangle \in L_{\text{d}} \quad \Leftarrow \quad L(M_{\text{d}}) \neq L_{\text{d}}$.

בשני המקרים קיבלנו סתירה לכך ש- $L(M_{\text{d}}) = L_{\text{d}}$ ולכן $L_{\text{d}} \notin RE$. ■

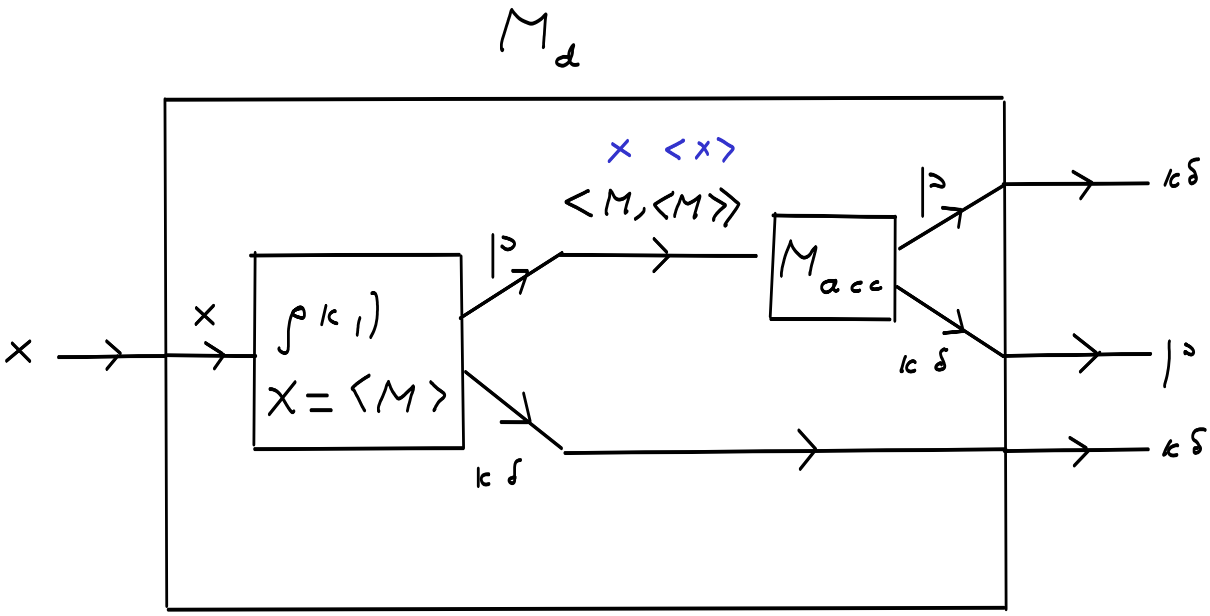
משפט 7.4 L_{acc} לא כריעה

$$L_{\text{acc}} = \left\{\langle M, w\rangle \ \middle|\ w \in L(M)\right\} \notin R \ .$$

הוכחה:
נניח בשלילה כי $L_{\text{acc}} \in R$ ותהי M_{acc} המ"ט המכריעה את L_{acc}.

נשתמש ב- M_{acc} כדי לבנות מ"ט M_{d} המכריעה את L_{d} (בסתירה לכך ש- $L_{\text{d}} \notin RE$ כפי שהוכחנו במשפט 7.3).

$$L_{\text{d}} = \left\{\langle M, w\rangle \ \middle|\ \langle M\rangle \notin L(M)\right\} \ .$$



<u>התאור של M_{d}</u>

$M_{\mathrm{d}} =$ על קלט x:

(1 בודקת האם $\langle M \rangle = x$. אם לא $\Leftarrow$ דוחה.

(2 מחשבת את $\langle\langle M \rangle\rangle = \langle x \rangle$.

(3 מריצה את M_{acc} על הזוג $\langle M, \langle M \rangle \rangle$:

- אם M_{acc} מקבלת $\Leftarrow$ M_{d} דוחה.
- אם M_{acc} דוחה $\Leftarrow$ M_{d} מקבלת.

כעת נוכיח כי M_{d} מכריעה את L_{d}:

אם $x \in L_{\mathrm{d}}$

$\Leftarrow$ $x = \langle M \rangle$ ו- $\langle M \rangle \notin L(M)$

$\Leftarrow$ M_{acc} דוחה את הזוג $\langle M, \langle M \rangle \rangle$

$\Leftarrow$ M_{d} מקבלת את x.

אם $x \notin L_{\mathrm{d}}$ שני מקרים:

מקרה (1): $x \neq \langle M \rangle \quad \Leftarrow \quad M_{\mathrm{d}}$ דוחה את x.

מקרה (2): $x = \langle M \rangle$ ו- $\langle M \rangle \in L(M)$

$\Leftarrow$ M_{acc} מקבלת את זוג $\langle M, \langle M \rangle \rangle$

$\Leftarrow$ M_{d} דוחה את x.

■

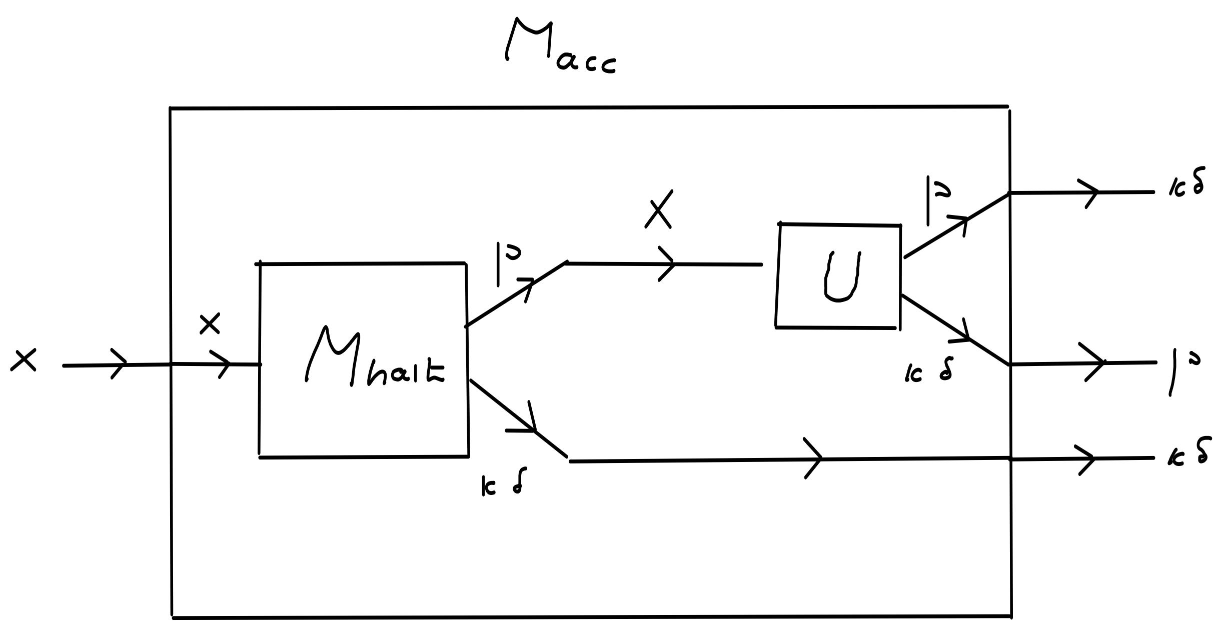
משפט 7.5 L_{halt} לא כריעה

$$L_{\text{halt}} = \left\{\langle M, w\rangle \;\middle|\; M \text{ עוצרת על } w\right\} \notin R\ .$$

הוכחה:

נניח בשלילה כי $L_{\text{halt}} \in R$ ותהי M_{halt} מ"ט המכריעה את L_{halt}.

נשתמש ב- M_{halt} כדי לבנות מ"ט M_{acc} המכריעה את L_{acc} (בסתירה לכך ש- $L_{\text{acc}} \notin R$ כפי שהוכחנו במשפט 7.4).



<u>התאור של M_{acc}</u>

$M_{\text{acc}} =$ על קלט x:

(1 מריצה את M_{acc} על x.

- אם M_{halt} דוחה $\Leftarrow$ M_{acc} דוחה.
- אם M_{halt} מקבלת $\Leftarrow$ M_{acc} מריצה את U על x ועונה כמוה.

<u>אבחנה</u>

נוכיח כי M_{acc} מכריעה את L_{acc}:

אם $x \in L_{\text{acc}}$

$\Leftarrow$ $x = \langle M, w\rangle$ ו- $\langle w\rangle \in L(M)$

$\Leftarrow$ M_{halt} מקבלת את x וגם U מקבלת את x

$\Leftarrow$ M_{acc} מקבלת את x.

אם $x \notin L_{\text{acc}}$ $\Leftarrow$ שני מקרים:

מקרה (1): $x \neq \langle M, w\rangle$

$\Leftarrow$ M_{halt} דוחה את x

$\Leftarrow$ M_{acc} דוחה את x.

מקרה (2): $x = \langle M, w\rangle$ ו- $\langle w\rangle \notin L(M)$ $\Leftarrow$ שני מקרים:

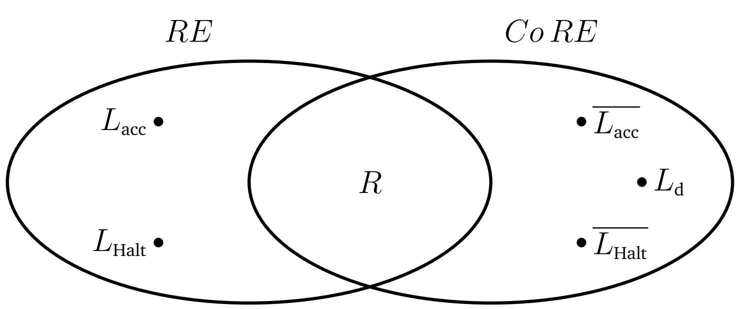
מקרה (א): M לא עוצרת על w $\Leftarrow$ M_{halt} דוחה את x $\Leftarrow$ M_{acc} דוחה את x.

מקרה (ב): M דוחה את w $\Leftarrow$ M_{halt} מקבלת את x אבל U דוחה את x $\Leftarrow$ M_{acc} דוחה את x.

הראנו כי M_{acc} מכריעה את L_{acc} בסתירה לכך ש- $L_{\text{acc}} \notin R$.
לכן $L_{\text{halt}} \notin R$. ■

משפט 7.6

$$\begin{aligned} &L_{\text{acc}} \in RE\backslash R \quad \Rightarrow \quad \bar{L}_{\text{acc}} \notin RE\ ,\\ &L_{\text{halt}} \in RE\backslash R \quad \Rightarrow \quad \bar{L}_{\text{halt}} \notin RE\ ,\\ &L_{\text{d}} \notin RE\backslash R\ . \end{aligned}$$



7.2 השפה L_E לא כריעה

הגדרה 7.4 השפה L_E

$$L_E = \{\langle M\rangle \mid L(M) = \varnothing\}\ .$$

משפט 7.7 $L_E \notin R$

$$L_E \notin R\ .$$

כלומר L_E לא כריעה.

הוכחה:
נניח בשלילה כי L_E כריעה. אז נבנה מ"ט M_{acc} המכריעה את L_{acc} באופן הבא.

<u>בנייה של M_w</u>

ראשית נגדיר את המ"ט M_w:

$M_w =$ על כל קלט x:

1) אם $x \neq w$ $\Leftarrow$ דוחה.

2) אם $x = w$ אז מריצה M על w ועונה כמוה.

<u>אבחנה</u>

אם $x = w$ ו- M מקבלת את w אז $L(M_w) = \Sigma^*$.

אם $x \neq w$ או אם $x = w$ ו- M דוחה את w אז $L(M_w) = \varnothing$.

<u>בנייה של M_{acc}</u>

נניח כי קיימת מ"ט M_E המכריעה את L_E. אז נבנה מ"ט M_{acc} המכריעה את L_{acc}:

$M_{\text{acc}} =$ על כל קלט x:

1) אם $x \neq \langle M, w \rangle$ $\Leftarrow$ דוחה.

2) אם $x = \langle M, w \rangle$, בעזרת התאור $\langle M, w \rangle$, בונה מ"ט M_w.

3) מריצה M_E על $\langle M_w \rangle$:

4)
- אם M_E מקבלת $\Leftarrow$ דוחה.
- אם M_E דוחה $\Leftarrow$ מקבלת.

<u>נכונות</u>

אם $x \in L_{\text{acc}}$ $\Leftarrow$ $x = \langle M, w \rangle$ ו- $w \in L(M)$ $\Leftarrow$ $L(M_w) = \Sigma^* \neq \varnothing$ $\Leftarrow$ M_E דוחה $\langle M_w \rangle$ $\Leftarrow$ M_{acc} מקבלת.

אם $x \notin L_{\text{acc}}$ $\Leftarrow$ שני מקרים:

<u>מקרה 1:</u> $x \neq \langle M, w \rangle$ $\Leftarrow$ $L(M_w) = \varnothing$ $\Leftarrow$ M_E מקבלת $\langle M_w \rangle$ $\Leftarrow$ M_{acc} דוחה.

<u>מקרה 2:</u> $x = \langle M, w \rangle$ ו- $w \notin L(M)$ $\Leftarrow$ $L(M_w) = \varnothing$ $\Leftarrow$ M_E מקבלת $\langle M_w \rangle$ $\Leftarrow$ M_{acc} דוחה.

לסיכום:
אם L_E כריעה אז אפשר לבנות מ"ט M_{acc} המכריעה את L_{acc} בסתירה לכך ש- $L_{\text{acc}} \notin R$.
לכן $L_E \notin R$. ■

משפט 7.8 $L_E \notin RE$

$$L_E \notin RE$$

הוכחה:
<u>הרעיון</u>

נבנה מ"ט א"ד N המקבלת את

$$\bar{L}_E = \{\langle M\rangle \mid L(M) \neq \varnothing\}$$

□ $N =$ על קלט x:

1) אם $\langle M\rangle \neq x$ $\Leftarrow$ דוחה.

2) אם $\langle M\rangle = x$ אז N בוחרת מילה $w \in \Sigma^*$ באופן א"ד.

3) מריצה M על w.

- אם M מקבלת $\Leftarrow$ N מקבלת.
- אם M דוחה $\Leftarrow$ N דוחה.

<u>הוכחת הנכונות</u>

אם $x \in \bar{L}_E$

$\Leftarrow$ $x = \langle M\rangle$ ו- $L(M) \neq \varnothing$

$\Leftarrow$ קיימת מילה $w \in \Sigma^*$ כך ש- $w \in L(M)$

$\Leftarrow$ $\exists$ ניחוש $w \in \Sigma^*$ כך ש M מקבלת את w

$\Leftarrow$ קיים חישוב של N המקבל את $x = \langle M\rangle$

$\Leftarrow$ $x \in L(N)$.

לכן קיימת מ"ט א"ד N המקבלת את השפה $\bar{L}_E$ לכן $\bar{L}_E \in RE$.

כעת נוכיח כי $L_E \notin RE$.

נניח בשלילה כי $L_E \in RE$. הוכחנו למעלה ש- $\bar{L}_E \in RE$. לכם לפי משפט 6.1, $L_E \in R$.
זו בסתירה לכך ש- $L_E \notin R$.
לכן $L_E \notin RE$.

■

7.3 השפה L_{EQ} לא כריעה

הגדרה 7.5 L_{EQ}

$$L_{EQ} = \{\langle M_1, M_2\rangle \mid L(M_1) = L(M_2)\}$$

משפט 7.9 $L_{EQ} \notin R$

$$L_{EQ} \notin R$$

השפה L_{EQ} לא כריעה.

הוכחה:

נניח בשלילה כי L_{EQ} כריעה. תהי M_{EQ} מ"ט המכריעה את L_{EQ}. אז נבנה מ"ט M_E המכריעה את L_E באופן הבא.

<u>בנייה של M_E</u>

$M_E =$ על כל קלט x:

1) אם $x \neq \langle M \rangle \Leftarrow$ דוחה.

2) אם $x = \langle M \rangle$, מריצה M_{EQ} על $\langle M, M_\varnothing \rangle$ כאשר $M_\varnothing$ המ"ט שדוחה כל קלט.

3)
- אם M_{EQ} מקבלת $\Leftarrow$ מקבלת.
- אם M_{EQ} דוחה $\Leftarrow$ דוחה.

<u>נכונות</u>

אם $x \in L_E$

$\Leftarrow$ $x = \langle M \rangle$ ו- $L(M) = \varnothing$

$\Leftarrow$ $L(M) = L(M_\varnothing)$

$\Leftarrow$ $\langle M, M_\varnothing \rangle \in L_{EQ}$

$\Leftarrow$ M_{EQ} מקבלת $\langle M, M_\varnothing \rangle$

$\Leftarrow$ M_E מקבל.

אם $x \notin L_E$ $\Leftarrow$ שני מקרים:

<u>מקרה 1:</u> $x \neq \langle M \rangle$ $\Leftarrow$ M_E דוחה.

<u>מקרה 2:</u> $\Leftarrow$ $x = \langle M \rangle$ ו- $L(M) \neq \varnothing$

$\Leftarrow$ $L(M) \neq L(M_\varnothing)$

$\Leftarrow$ $\langle M, M_\varnothing \rangle \notin L_{EQ}$

$\Leftarrow$ M_{EQ} דוחה $\langle M, M_\varnothing \rangle$

$\Leftarrow$ M_E דוחה.

לסיכום:
אם L_{EQ} כריעה אז אפשר לבנות מ"ט M_E המכריעה את L_E בסתירה למשפט 7.7 האומר ש- $L_E \notin R$.
לכן $L_{EQ} \notin R$. ■

משפט 7.10 $L_{EQ} \notin RE$

$$L_{EQ} \notin RE$$

L_{EQ} לא קבילה.

הוכחה:
נניח בשלילה כי L_{EQ} קבילה. תהי M_{EQ} מ"ט המקבלת את L_{EQ}. אז נבנה מ"ט M_E המקבלת את L_E באופן הבא.

<u>בנייה של M_E</u>

$M_E =$ על כל קלט x:

1) אם $x \neq \langle M \rangle$ $\Leftarrow$ דוחה.

2) אם $x = \langle M \rangle$, מריצה M_{EQ} על $\langle M, M_{\varnothing} \rangle$ כאשר $M_{\varnothing}$ המ"ט שדוחה כל קלט.

3) • אם M_{EQ} מקבלת $\Leftarrow$ מקבלת.

<u>נכונות</u>

אם $x \in L_E$

$\Leftarrow$ $x = \langle M \rangle$ ו- $L(M) = \varnothing$

$\Leftarrow$ $L(M) = L(M_{\varnothing})$

$\Leftarrow$ $\langle M, M_{\varnothing} \rangle \in L_{EQ}$

$\Leftarrow$ M_{EQ} מקבלת $\langle M, M_{\varnothing} \rangle$

$\Leftarrow$ M_E מקבל.

לסיכום:
אם L_{EQ} קבילה אז אפשר לבנות מ"ט M_E המקבלת את L_E בסתירה למשפט 7.8 האומר ש- $L_E \notin RE$.
לכן $L_{EQ} \notin RE$. ■

משפט 7.11 $\bar{L}_{EQ} \notin RE$

$$\bar{L}_{EQ} \notin RE \ .$$

הוכחה:

נניח בשלילה כי $\bar{L}_{EQ}$ קבילה. תהי $M_{\bar{E}Q}$ מ"ט המקבלת את $\bar{L}_{EQ}$. אז נבנה מ"ט $M_{\overline{\text{acc}}}$ המקבלת את $\bar{L}_{\text{acc}}$ באופן הבא.

<u>בנייה של M_1</u>

ראשית נגדיר מ"ט M_1 באופן הבא:

$M_1 =$ על קלט x:

1) מריצה M על w ועונה כמוה.

<u>בנייה של $M_{\overline{\text{acc}}}$</u>

$M_{\overline{\text{acc}}} =$ על כל קלט x:

1) אם $x \neq \langle M, w \rangle$ $\Leftarrow$ מקבלת.

2) אם $x = \langle M, w \rangle$ אז בונה M_1.

3) מריצה $M_{\overline{EQ}}$ על $\langle M_1, M^* \rangle$ כאשר M^* המ"ט שמקבלת כל קלט.

4) • אם $M_{\overline{EQ}}$ מקבלת $\Leftarrow$ מקבלת.

<u>נכונות</u>

אם $x \in L_{\overline{\text{acc}}}$

$\Leftarrow$ M לא מקבלת w

$\Leftarrow$ $L(M_1) = \varnothing$

$\Leftarrow$ $\langle M_1, M^* \rangle \in L_{\overline{EQ}}$

$\Leftarrow$ $M_{\overline{EQ}}$ מקבלת $\langle M_1, M^* \rangle$

$\Leftarrow$ $M_{\overline{\text{acc}}}$ מקבל.

לסיכום:

אם $L_{\overline{EQ}}$ קבילה אז אפשר לבנות מ"ט $M_{\overline{\text{acc}}}$ המקבלת את $L_{\overline{\text{acc}}}$ בסתירה למשפט 7.6 האומר ש- $L_{\overline{\text{acc}}} \notin RE$.
לכן $L_{\overline{EQ}} \notin RE$.

■

7.4 סיכום: כריעות וקבילות של שפות

	כריעה	קבילה
L_{acc}	$\times$	✓
$\overline{L_{\text{acc}}}$	$\times$	$\times$
L_{d}	$\times$	$\times$
L_{Halt}	$\times$	✓
$\overline{L_{\text{Halt}}}$	$\times$	$\times$
L_{E}	$\times$	$\times$
$\overline{L_{\text{E}}}$	$\times$	✓
L_{EQ}	$\times$	$\times$
$\overline{L_{\text{EQ}}}$	$\times$	$\times$
L_{REG}	$\times$	$\times$
L_{NOTREG}	$\times$	$\times$

שיעור 8
רדוקציה

8.1 טבלה של רדוקציות

טבלה של רדוקציות

רדוקציה	עמוד
$L_{\text{HALT}} \leqslant L_{\text{acc}}$	דוגמה 8.6 עמוד 86
$\bar{L}_{\text{acc}} \leqslant L_{\text{NOTREG}}$	דוגמה 8.11 עמוד 90
$L_{\text{acc}} \leqslant L_{\text{NOTREG}}$	דוגמה 8.12 עמוד 91
$L_{\text{HALT}} \leqslant L_{\text{NOTREG}}$	דוגמה 8.13 עמוד 92
$L_{\text{acc}} \leqslant L_{\text{REG}}$	דוגמה 8.15 עמוד 94
$\bar{L}_{\text{acc}} \leqslant L_{\text{REG}}$	דוגמה 8.14 עמוד 93
$\bar{L}_{\text{acc}} \leqslant L_{M_1 \neg M_2}$ כאשר $L_{M_1 \neg M_2} = \{\langle M_1, M_2, w\rangle \mid w \in L(M_1) \wedge w \notin L(M_2)\}$.	דוגמה 8.16 עמוד 95
$\bar{L}_{\text{acc}} \leqslant L_{M_1 \subset M_2}$ כאשר $L_{M_1 \subset M_2} = \{\langle M_1, M_2\rangle \mid L(M_1) \subset L(M_2)\}$.	דוגמה 8.17 עמוד 95

8.2 מ"ט המחשבת את פונקציה

הגדרה 8.1 מ"ט המחשבת פונקציה

בהינתן פונקציה $f : \Sigma^* \to \Sigma^*$ אומרים כי מ"ט M מחשבת את f אם לכל $x \in \Sigma^*$:

- M מגיעה ל- q_{acc} בסוף החישוב של $f(x)$ וגם
- על סרט הפלט של M רשום $f(x)$.

הערה 8.1

מ"ט שמחשבת פונקציה עוצרת תמיד.

הגדרה 8.2 מ"ט המחשבת פונקציה

בהינתן פונקציה $f : \Sigma^* \to \Sigma^*$ אומרים כי f חשיבה אם קיימת מ"ט המחשבת את f.

דוגמה 8.1

$$f_1(x) = xx \ . \tag{8.1}$$

$f_1(x)$ חשיבה.

דוגמה 8.2

$$f_2(x) = \begin{cases} x & \text{זוגי } |x| \\ xx & \text{אי-זוגי } |x| \end{cases} \ . \tag{8.2}$$

$f_2(x)$ חשיבה.

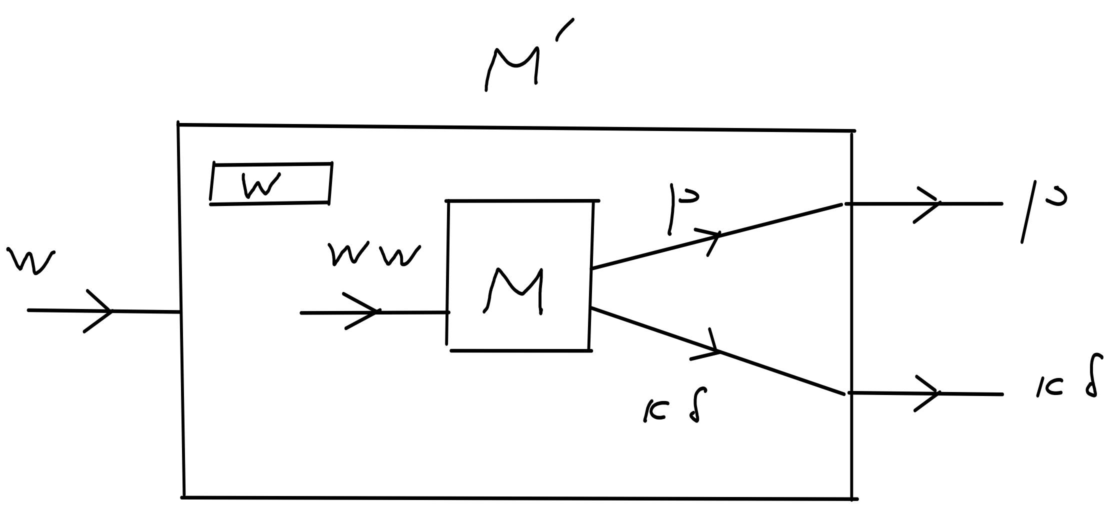
דוגמה 8.3

$$f_3(x) = \begin{cases} \langle M' \rangle & x = \langle M \rangle \\ \langle M^* \rangle & x \neq \langle M \rangle \end{cases} \ . \tag{8.3}$$

כאשר

- M^* מ"ט שמקבלת כל קלט.
- M' מ"ט המקבלת את השפה

$$L(M') = \left\{ w \in \Sigma^* \;\middle|\; ww \in L(M) \ . \right\}$$



$f_3(x)$ חשיבה כי ניתן לבנות מ"ט שבודקת האם $x = \langle M \rangle$. אם לא, מחזירה קידוד קבוע $\langle M^* \rangle$. ואם כן, מחזירה קידוד $\langle M' \rangle$ ע"י הוספת מעברים המשכפלים את הקלט בתחילת הקידוד $\langle M \rangle$.

דוגמה 8.4

$$f_4(x) = \begin{cases} 1 & x = \langle M\rangle \wedge \langle M\rangle \in L(M) \\ 0 & \text{אחרת} \end{cases} \tag{8.4}$$

$f_4(x)$ לא חשיבה כי ייתכנו קלטים $x = \langle M\rangle$ ו- M לא עוצרת על $\langle M\rangle$.

8.3 רדוקציות

הגדרה 8.3 רדוקציות

בהינתן שתי שפות $L_1, L_2 \subseteq \Sigma^*$ אומרים כי L_1 ניתנת לרדוקציה ל- L_2, ומסמנים

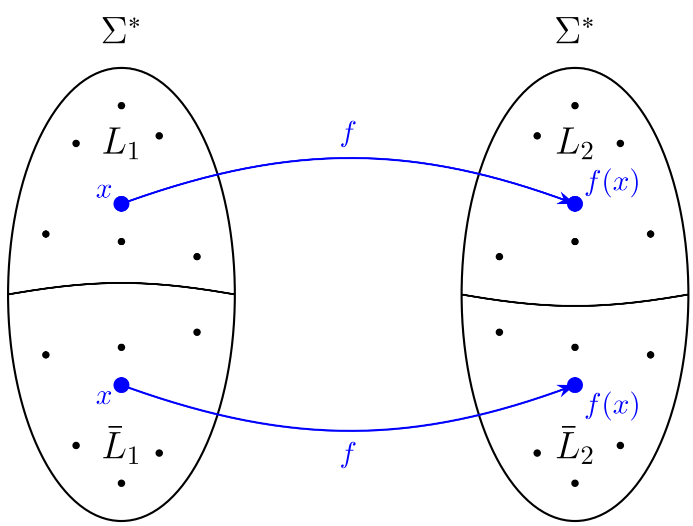
$$L_1 \leqslant L_2 \ ,$$

אם $\exists$ פונקציה $f : \Sigma^* \to \Sigma^*$ המקיימת:

(1 f חשיבה

(2 לכל $x \in \Sigma^*$:

$$x \in L_1 \quad \Leftrightarrow \quad f(x) \in L_2 \ .$$



דוגמה 8.5

נתונות השפות

$$L_1 = \left\{x \in \{0,1\}^* \;\middle|\; \text{זוגי } |x|\right\} \ ,$$
$$L_2 = \left\{x \in \{0,1\}^* \;\middle|\; \text{אי-זוגי } |x|\right\} \ .$$

הוכיחו כי

$$L_1 \leqslant L_2 \ .$$

פתרון:

נגדיר את הפונקציה

$$f(x) = \begin{cases} 1 & \text{זוגי } |x|, \\ 10 & \text{אי-זוגי } |x| \end{cases}$$

<u>הוכחת הנכונות:</u>

$x \in L_1 \Leftarrow$ $|x|$ זוגי $\Leftarrow f(x) = 1 \Leftarrow$ $|f(x)|$ אי-זוגי $f(x) \in L_2$.

$x \notin L_1 \Leftarrow$ $|x|$ אי-זוגי $\Leftarrow f(x) = 10 \Leftarrow$ $|f(x)|$ זוגי $f(x) \notin L_2$.

משפט 8.1 משפט הרדוקציה

לכל שתי שפות $L_1, L_2 \subseteq \Sigma^*$, אם קיימת רדוקציה

$$L_1 \leqslant L_2$$

אזי התנאים הבאים מתקיימים:

$$L_1 \in R \Leftarrow L_2 \in R \quad (1)$$

$$L_1 \in RE \Leftarrow L_2 \in RE \quad (2)$$

$$L_1 \notin R \Rightarrow L_2 \notin R \quad (3)$$

$$L_1 \notin RE \Rightarrow L_2 \notin RE \quad (4)$$

הוכחה: מכיוון ש-

$$L_1 \leqslant L_2$$

קיימת פונקציה f חשיבה המקיימת:

$$x \in L_1 \quad \Leftrightarrow \quad f(x) \in L_2$$

לכל $x \in \Sigma^*$.

תהי M_f מ"ט המחשבת את f.

(1) <u>נוכיח $L_1 \in R \Leftarrow L_2 \in R$</u>

תהי M_2 מ"ט המכריעה את L_2.

נבנה מ"ט M_1 המכריעה את L_1.

<u>התאור של M_1</u>

$M_1 =$ על קלט x:

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

נוכיח כי M_1 מכריעה את L_1.

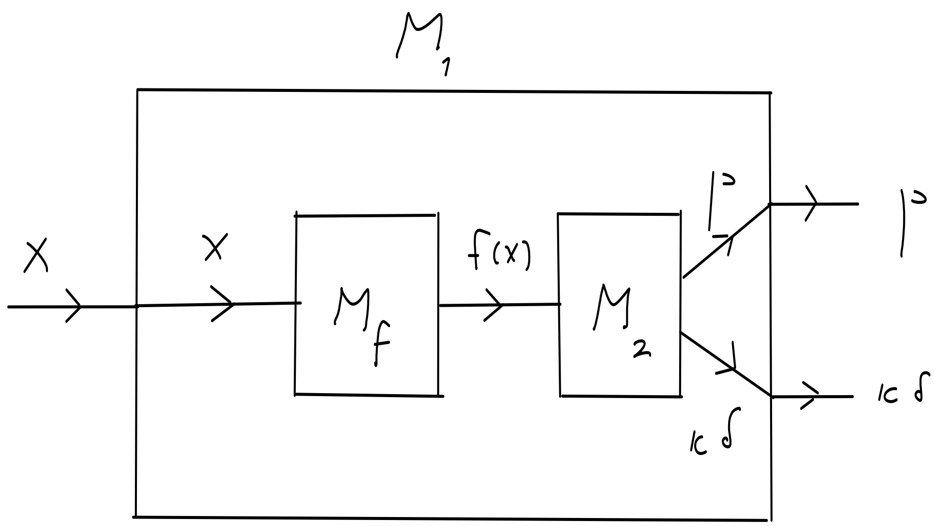
- אם $x \in L_1 \Leftarrow f(x) \in L_2 \Leftarrow$ M_2 מקבלת את $f(x) \Leftarrow$ M_1 מקבלת את x.

- אם $x \notin L_1$ $\Leftarrow$ $f(x) \notin L_2$ $\Leftarrow$ M_2 דוחה את $f(x)$ $\Leftarrow$ M_1 דוחה את x.

(2) נוכיח $L_2 \in RE \Leftarrow L_1 \in RE$

תהי M_2 מ"ט המקבלת את L_2.

נבנה מ"ט M_1 המקבלת את L_1.



התאור של M_1

$M_1 =$ על קלט x:

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

נוכיח כי M_1 מקבלת את L_1:

- אם $x \in L_1$ $\Leftarrow$ $f(x) \in L_2$ $\Leftarrow$ M_2 מקבלת את $f(x)$ $\Leftarrow$ M_1 מקבלת את x.
- אם $x \notin L_1$ $\Leftarrow$ $f(x) \notin L_2$ $\Leftarrow$ M_2 לא מקבלת את $f(x)$ $\Leftarrow$ M_1 לא מקבלת את x.

(3)

(4)

■

כלל 8.1

- אם רוצים להוכיח כי שפה כלשהי $L \in RE$, בוחרים שפה אחרת $L' \in RE$ ומראים שקיימת רדוקציה $L \leqslant L'$.

לדוגמה:

$$L \leqslant L_{\text{acc}}$$

(כנ"ל לגבי R)

- אם רוצים להוכיח כי שפה כלשהי $L \notin RE$ בוחרים שפה אחרת $L' \notin RE$ ומראים שקיימת רדוקציה

$$L' \leqslant L \ .$$

לדוגמה

$$L_{\text{d}} \leqslant L$$

(כנ"ל לגבי R).

דוגמה 8.6

נתונות השפות $L_{\text{acc}} = \{\langle M, w\rangle \mid w \in L(M)\}$ ו- $L_{\text{halt}} = \{\langle M, w\rangle \mid M \text{ עוצרת על } w\}$.
הוכיחו כי $L_{\text{acc}} \notin R$ ע"י רדוקציה $L_{\text{acc}} \leqslant L_{\text{halt}}$.

פתרון:
נבנה פונקציה f חשיבה ומקיימת

$$x \in L_{\text{acc}} \quad \Leftrightarrow \quad f(x) \in L_{\text{halt}} \ .$$

M מקבלת את w $\Leftarrow$ M' תעצור על w'.

M דוחה את w $\Leftarrow$ M' לא תעצור על w'.

M לא עוצרת את w $\Leftarrow$ M' לא תעצור על w'.

$$f(x) = \begin{cases} \langle M', w\rangle & : x = \langle M, w\rangle \\ \langle M_{\text{loop}}, \varepsilon\rangle & : x \neq \langle M, w\rangle \end{cases}$$

כאשר

- M_{loop} מ"ט שלא עוצרת על אף קלט.
- M' מ"ט המתנהגת כמו M פרט למקומות בהם M עצרה ודחתה, M' תיכנס ללולאה אינסופית.

<u>נכונות הרדוקציה</u>

f חשיבה כי ניתן לבנות מ"ט שתבדוק האם $x = \langle M, w\rangle$.

אם לא, תחזיר קידוד קבוע $\langle M_{\text{loop}}, w\rangle$

ואם כן, תחזיר קידוד $\langle M', w\rangle$ ע:י ביצוע שינויים לוקלים בקידוד של M.

נוכיח כי $f(x) \in L_{\text{halt}} \quad \Leftrightarrow \quad x \in L_{\text{acc}}$.

אם $x \in L_{\text{acc}}$:

$\Leftarrow$ $x=\langle M,w\rangle$ ו- $w\in L(M)$

$\Leftarrow$ $f(x)=\langle M',w\rangle$ ו- M' עוצרת ומקבלת את w

$\Leftarrow$ $f(x)\in L_{\text{halt}}$.

אם $x\notin L_{\text{acc}}$ אז שני מקרים:

מקרה 1:

$x\neq\langle M,w\rangle$ $\Leftarrow$ $f(x)=\langle M_{\text{loop}},\varepsilon\rangle$ ו- M_{loop} לא עוצרת על ε $\Leftarrow$ $f(x)\notin L_{\text{halt}}$.

מקרה 2:

$x=\langle M,w\rangle$ ו- $w\notin L(M)$ $\Leftarrow$ $f(x)=\langle M',w\rangle$ $\Leftarrow$ שני מקרים:

מקרה א: M לא עוצרת על w $\Leftarrow$ M' לא עוצרת על w $\Leftarrow$ $f(x)\notin L_{\text{halt}}$.

מקרה ב: M דוחה את w $\Leftarrow$ M' לא עוצרת על w $\Leftarrow$ $f(x)\notin L_{\text{halt}}$.

לסיכום, הוכחנו רדוקציה $L_{\text{halt}}\leqslant L_{\text{acc}}$. ומכיוון ש- $L_{\text{acc}}\notin R$ (משפט 7.4) אז ממשט הרדוקציה 8.1, מתקיים $L_{\text{halt}}\notin R$.

דוגמה 8.7

נתונה השפה

$$L_{\Sigma^*}=\{\langle M\rangle \mid L(M)=\Sigma^*\}$$

ונתונה השפה

$$\bar{L}_{\Sigma^*}=\{\langle M\rangle \mid L(M)\neq\Sigma^*\}\cup\{x\neq\langle M\rangle\}\ .$$

הוכיחו כי:

א) $L_{\Sigma^*}\notin RE$

ב) $L_{\Sigma^*}\notin R$

ג) $\bar{L}_{\Sigma^*}\notin RE$

פתרון:

נוכיח כי $L_{\Sigma^*}\notin R$ ע"י רדוקציה

$$L_{\text{acc}}\leqslant L_{\Sigma^*}\ .$$

נבנה פונקציה חשיבה f המקיימת

$$x\in L_{\text{acc}}\quad\Leftrightarrow\quad f(x)\in L_{\Sigma^*}\ .$$

$$w\in L(M)\quad\Rightarrow\quad L(M')=\Sigma^*$$

$$w\notin L(M)\quad\Rightarrow\quad L(M')\neq\Sigma^*$$

$$f(x)=\begin{cases}\langle M'\rangle & : x=\langle M,w\rangle\\ \langle M_{\varnothing}\rangle & : x\neq\langle M,w\rangle\end{cases}$$

כאשר

- $M_{\varnothing}$ מ"ט שדוחה כל קלט.
- M' היא מ"ט שעל כל קלט x, מתעלמת מ- x ומריצה את M על w ועונה כמוה.

אבחנה:

$$L(M') = \begin{cases} \Sigma^* & : w \in L(M) \\ \varnothing & : w \notin L(M) \end{cases}$$

נכונות הרדוקציה:

f חשיבה כי ניתן לבנות מ"ט שתבדוק האם $x = \langle M, w \rangle$.

אם לא תחזיר קידוד קבוע $\langle M_{\varnothing} \rangle$.

אם כן, תחזיר קידוד $\langle M' \rangle$ ע"י הוספת קוג ל- M שמוחק את הקלט מהסרט וכותב w במקומו.

נוכיח כי

$$x \in L_{\text{acc}} \quad \Leftrightarrow \quad f(x) \in L_{\Sigma *}$$

אם $x \in L_{\text{acc}}$ $\Leftarrow$ $x = \langle M, w \rangle$ ו- $w \in L(M)$ $\Leftarrow$ $f(x) = \langle M' \rangle$ ולפי האבחנה $L(M') = \Sigma^*$ $\Leftarrow$ $f(x) \in L_{\Sigma *}$.

אם $x \in L_{\text{acc}}$ $\Leftarrow$ שני מקרים:

מקרה 1): $x \neq \langle M, w \rangle$ $\Leftarrow$ $f(x) = \langle M_{\varnothing} \rangle$ ו- $L\left(M_{\varnothing}\right) = \varnothing$ $\Leftarrow$ $f(x) \notin L_{\Sigma *}$.

מקרה 2): $x = \langle M, w \rangle$ ו- $w \notin L(M)$ $\Leftarrow$ $f(x) = \langle M' \rangle$ ולפי האבחנה $L\left(M'\right) = \varnothing$ $\Leftarrow$ $f(x) \notin L_{\Sigma *}$.

לסיכום, הוכחנו רדוקציה $L_{\Sigma *} \leqslant L_{\text{acc}}$. ומכיוון ש- $L_{\text{acc}} \notin R$ (משפט 7.4) אז ממשפט הרדוקציה 8.1, מתקיים $L_{\Sigma *} \notin R$.

דוגמה 8.8

נתונה השפה

$$L_{\text{d}} = \left\{ \langle M \rangle \mid \langle M \rangle \notin L(M) \right\}$$

ונתונה השפה

$$\bar{L}_{\text{acc}} = \left\{ \langle M, w \rangle \; | w \notin L(M) \right\} \cup \left\{ x \neq \langle M, w \rangle \right\} \; .$$

הוכיחו כי
$\bar{L}_{\text{acc}} \notin RE$ ע"י רדוקציה

$$L_{\text{d}} \leqslant \bar{L}_{\text{acc}} \; .$$

פתרון:
נבנה פונקציה חשיבה f המקיימת

$$x \in L_{\text{d}} \quad \Leftrightarrow \quad f(x) \in L_{\text{acc}} \; .$$

$$\langle M \rangle \notin L(M) \quad \Leftarrow \quad w' \notin L(M')$$

$$\langle M \rangle \in L(M) \quad \Leftarrow \quad w' \in L(M')$$

$$f(x)=\begin{cases}\langle M,\langle M\rangle\rangle & : x=\langle M\rangle\\ \langle M^*,\varepsilon\rangle & : x\neq\langle M\rangle\end{cases}$$

כאשר M^* המ"ט שמקבלת כל קלט.

נכונות הרדוקציה:

f חשיבה כי ניתן לבנות מ"ט שתבדוק האם $x=\langle M,w\rangle$.

אם לא תחזיר קידוד קבוע $\langle M^*,\varepsilon\rangle$.

אם כן, תחשב $\langle M,\langle M\rangle\rangle$.

נוכיח כי

$$x\in L_{\mathrm{d}} \quad\Leftrightarrow\quad f(x)\in \bar{L}_{\mathrm{acc}}$$

אם $x\in L_{\mathrm{d}}$ $\Leftarrow$ $x=\langle M\rangle$ ו- $\langle M\rangle\notin L(M)$ $\Leftarrow$ $f(x)=\langle M,\langle M\rangle\rangle$ ו- $\langle M\rangle\notin L(M)$ $\Leftarrow$ $f(x)\in\bar{L}_{\mathrm{acc}}$.

אם $x\notin L_{\mathrm{d}}$ $\Leftarrow$ שני מקרים:

מקרה 1): $x\neq\langle M\rangle$ $\Leftarrow$ $f(x)=\langle M^*,\varepsilon\rangle$ ו- $\varepsilon\in L(M^*)$ $\Leftarrow$ $f(x)\notin\bar{L}_{\mathrm{acc}}$.

מקרה 2): $x=\langle M\rangle$ ו- $\langle M\rangle\in L(M)$ $\Leftarrow$ $f(x)=\langle M,\langle M\rangle\rangle$ ו- $\langle M\rangle\in L(M)$ $\Leftarrow$ $f(x)\notin\bar{L}_{\mathrm{acc}}$.

לסיכום, הוכחנו רדוקציה $L_{\mathrm{d}}\leqslant\bar{L}_{\mathrm{acc}}$, ומכיוון ש- $L_{\mathrm{d}}\notin RE$ (משפט 7.3) אז ממשט הרדוקציה 8.1, מתקיים $\bar{L}_{\mathrm{acc}}\notin RE$.

משפט 8.2 משפט הרדוקציה בין שפות משלימות

אם קיימת רדוקציה $L_1\leqslant L_2$, אזי קיימת רדוקציה $\bar{L}_1\leqslant\bar{L}_2$.

הוכחה:
אם $\exists$ רדוקציה

$$L_1\leqslant L_2$$

אזי $\exists$ פונקציה חשיבה f המקיימת

$$x\in L_1\quad\Leftrightarrow\quad f(x)\in L_2$$

ולכן עבור אותה פונקציה f היא גם חשיבה וגם מקיימת

$$x\in\bar{L}_1\quad\Leftrightarrow\quad f(x)\in\bar{L}_2$$

ולכן

$$\bar{L}_1\leqslant\bar{L}_2\ .$$

■

8.4 דוגמאות בשימוש של משפט הרדוקציה בין שפות משלימות (משפט 8.2)

דוגמה 8.9

הוכחנו בדוגמה 8.7 רדוקציה
$$L_{\text{acc}} \leqslant L_{\Sigma*} \ .$$
לכן לפי משפט 8.2 קיימת רדוקציה
$$\bar{L}_{\text{acc}} \leqslant \bar{L}_{\Sigma*} \ .$$
מכיוון ש- $\bar{L}_{\text{acc}} \notin RE$, אזי ממשפט הרדוקציה 8.1 מתקיים $\bar{L}_{\Sigma*} \notin RE$.

דוגמה 8.10

הוכחנו בדוגמה 8.8 רדוקציה
$$L_{\text{d}} \leqslant \bar{L}_{\text{acc}} \ .$$
לכן לפי משפט 8.2 קיימת רדוקציה
$$\bar{L}_{\text{d}} \leqslant L_{\text{acc}} \ .$$
מכיוון ש- $L_{\text{acc}} \in RE$, אזי ממשט הרדוקציה 8.1 מתקיים $\bar{L}_{\text{d}} \in RE$.

8.5 דוגמאות בשימוש של משפט הרדוקציה (משפט 8.1)

דוגמה 8.11 $\bar{L}_{\text{acc}} \leqslant L_{\text{NOTREG}}$

תהי L_{NOTREG} השפה
$$L_{\text{NOTREG}} = \big\{\langle M\rangle \ \big|\ \text{לא רגולרית } L(M)\big\} \ .$$
הוכיחו כי השפה L_{NOTREG} לא כריעה על ידי רדוקציה מ- $\bar{L}_{\text{acc}}$.

פתרון:

השפה $\bar{L}_{\text{acc}}$ מוגדרת
$$\bar{L}_{\text{acc}} = \big\{\langle M,w\rangle \ \big|\ w \text{ לא מקבלת } M\big\} \cup \{x \neq \langle M,w\rangle\} \ .$$
והשפה L_{NOTREG} מוגדרת
$$L_{\text{NOTREG}} = \big\{\langle M\rangle \ \big|\ \text{לא רגולרית } L(M)\big\} \ .$$
נגדיר הפונקציה הבאה:
$$f(x) = \begin{cases} \langle M'\rangle & x = \langle M,w\rangle \\ w' \in PAL & x \neq \langle M,w\rangle \end{cases}$$
כאשר M' מ"ט הבאה:

$M' =$ על כל קלט y:

1) אם $y \in PAL$ $\Leftarrow$ מקבלת.

2) אחרת מריצה M על w ועונה כמוה.

<u>הוכחת נכונות הרדוקציה</u>

אם $x \in \bar{L}_{\text{acc}}$ $\Leftarrow$ שני מקרים:

<u>מקרה 1:</u> $x = \langle M, w\rangle$

$\Leftarrow$ M לא מקבלת w

$\Leftarrow$ $L(M') \in PAL$

$\Leftarrow$ $\langle M'\rangle \in PAL$

$\Leftarrow$ $f(x) \in PAL$

$\Leftarrow$ $f(x) \in L_{\text{NOTREG}}$.

<u>מקרה 2:</u> $x \neq \langle M, w\rangle \Leftarrow f(x) \in PAL \Leftarrow f(x) \in L_{\text{NOTREG}}$.

אם $x \notin \bar{L}_{\text{acc}}$ $\Leftarrow$ M מקבלת w $\Leftarrow$ $L(M') = \Sigma^*$ $\Leftarrow$ $f(x) \in \Sigma^*$ $\Leftarrow$ $f(x) \in L_{\text{NOTREG}}$.

לכן הוכחנו כי $f(x) \in NOTERG \Leftrightarrow x \in \bar{L}_{\text{acc}}$, ז"א $f(x)$ היא רדוקציה מ- L_{acc} ל- L_{NOTREG}.

השפה $\bar{L}_{\text{acc}}$ לא כריעה. לפיכך, לפי משפט הרדוקציה גם L_{NOTREG} לא כריעה.

דוגמה 8.12 $L_{\text{acc}} \leqslant L_{\text{NOTREG}}$

תהי L_{NOTREG} השפה

$$L_{\text{NOTREG}} = \left\{\langle M\rangle \;\middle|\; \text{לא רגולרית } L(M)\right\} .$$

הוכיחו כי השפה L_{NOTREG} לא כריעה על ידי רדוקציה מ- L_{acc}.

פתרון:

השפה L_{acc} מוגדרת

$$L_{\text{acc}} = \left\{\langle M, w\rangle \;\middle|\; w \text{ מקבלת } M\right\} .$$

והשפה L_{NOTREG} מוגדרת

$$L_{\text{NOTREG}} = \left\{\langle M\rangle \;\middle|\; \text{לא רגולרית } L(M)\right\} .$$

נגדיר הפונקציה הבאה:

$$f(x) = \begin{cases} \langle M'\rangle & x = \langle M, w\rangle \\ \langle M_{\varnothing}\rangle & x \neq \langle M, w\rangle \end{cases}$$

כאשר M' מ"ט הבאה:

$M' =$ על כל קלט y:

1) M' מריצה M על w.

2) • אם M דוחה $\Leftarrow$ דוחה.

- אם M מקבלת $\Leftarrow M'$ בודקת אם y פלינדרום.
  - אם כן $\Leftarrow$ מקבלת.
  - אם לא $\Leftarrow$ דוחה.

<u>הוכחת נכונות הרדוקציה</u>

אם $x \in L_{\text{acc}}$ $\Leftarrow$ M מקבלת w $\Leftarrow$ $L(M') = PAL$ $\Leftarrow$ $f(x) \in PAL$ $\Leftarrow$ $f(x) \in L_{\text{NOTREG}}$.

$x \notin L_{\text{acc}}$ $\Leftarrow$ שני מקרים.

<u>מקרה 1:</u> $x \neq \langle M, w\rangle$ $\Leftarrow$ $f(x) = \langle M_{\varnothing}\rangle$ ו- $L(M_{\varnothing}) = \varnothing$ $\Leftarrow$ $\langle M_{\varnothing}\rangle \notin L_{\text{NOTREG}}$
$\Leftarrow$ $f(x) \notin L_{\text{NOTREG}}$.

<u>מקרה 2:</u> $x = \langle M, w\rangle$ ו- M לא מקבלת w $\Leftarrow$ $L(M') = \varnothing$ $\Leftarrow$ $\langle M'\rangle \notin L_{\text{NOTREG}}$
$\Leftarrow$ $f(x) \notin L_{\text{NOTREG}}$.

דוגמה 8.13 $L_{\text{HALT}} \leqslant L_{\text{NOTREG}}$

תהי השפה L_{NOTREG}

$$L_{\text{NOTREG}} = \left\{\langle M\rangle \;\middle|\; L(M) \text{ לא רגולרית} \right\} \ .$$

הוכיחו כי השפה L_{NOTREG} לא כריעה על ידי רדוקציה מ- L_{HALT}.

פתרון:

השפה L_{HALT} מוגדרת

$$L_{\text{HALT}} = \left\{\langle M, w\rangle \;\middle|\; M \text{ עוצרת על } w \right\} \ .$$

והשפה L_{NOTREG} מוגדרת

$$L_{\text{NOTREG}} = \left\{\langle M\rangle \;\middle|\; L(M) \text{ לא רגולרית} \right\} \ .$$

נגדיר הפונקציה הבאה:

$$f(x) = \begin{cases} \langle M'\rangle & x = \langle M, w\rangle \\ \langle M_{\varnothing}\rangle & x \neq \langle M, w\rangle \end{cases} \ .$$

כאשר M' מ"ט הבאה:

$M' =$ על כל קלט y:

1) M' מריצה M על w.
2)
   - אם M דוחה $\Leftarrow$ דוחה.
   - אם M מקבלת $\Leftarrow$ ממשיכה לשלב 3).
3) M' בודקת אם $y \in PAL$.
   - אם כן $\Leftarrow$ מקבלת.
   - אם לא $\Leftarrow$ דוחה.

הוכחת הנכונות

$x \in L_{\text{HALT}} \quad \Rightarrow \quad L(M') \in PAL \quad \Rightarrow \quad L(M') \in L_{\text{NOTREG}}$.

$x \notin L_{\text{HALT}} \quad \Rightarrow$ שני מקרים:

מקרה 1: $x \neq \langle M, w\rangle \quad \Rightarrow \quad f(x) = \langle M_\varnothing\rangle$ ו- $L(M_\varnothing) = \varnothing \quad \Rightarrow \quad \langle M_\varnothing\rangle \notin L_{\text{NOTREG}}$
$\Rightarrow \quad f(x) \notin L_{\text{NOTREG}}$.

מקרה 2: $x = \langle M, w\rangle$ ו- M לא עוצרת על $w \quad \Rightarrow \quad L(M_\varnothing) = \varnothing \quad \Rightarrow \quad \langle M_\varnothing\rangle \notin L_{\text{NOTREG}}$
$\Rightarrow \quad f(x) \notin L_{\text{NOTREG}}$.

דוגמה 8.14 $\bar{L}_{\text{acc}} \leqslant L_{\text{REG}}$

תהי L_{REG} השפה

$$L_{\text{REG}} = \left\{\langle M\rangle \mid \text{רגולרית } L(M)\right\} .$$

הוכיחו כי השפה L_{REG} לא כריעה על ידי רדוקציה מ- $\bar{L}_{\text{acc}}$.

פתרון:

השפה $\bar{L}_{\text{acc}}$ מוגדרת

$$\bar{L}_{\text{acc}} = \left\{\langle M, w\rangle \mid \text{לא מקבלת } w \text{ } M\right\} \cup \{x \mid x \neq \langle M, w\rangle\} .$$

והשפה L_{REG} מוגדרת

$$L_{\text{REG}} = \left\{\langle M\rangle \mid \text{רגולרית } L(M)\right\} .$$

נגדיר הפונקציה הבאה:

$$f(x) = \begin{cases} \langle M'\rangle & x = \langle M, w\rangle \\ \langle M_\varnothing\rangle & x \neq \langle M, w\rangle \end{cases}$$

כאשר $M_\varnothing$ המ"ט שדוחה כל קלט ו- M' מ"ט הבאה:

$M' =$ על כל קלט y:

1) מריצה M על w.

2)
- אם M דוחה $\Leftarrow$ דוחה.
- אם M מקבלת $\Leftarrow$ בודקת אם y פלינדרום:
  - אם כן $\Leftarrow$ מקבלת.
  - אם לא $\Leftarrow$ דוחה.

אבחנה

$$L(M') = \begin{cases} PAL & w \in L(M) \\ \varnothing & w \notin L(M) \end{cases}$$

הוכחת נכונות הרדוקציה

אם $x \in \bar{L}_{\text{acc}}$ $\Leftarrow$ שני מקרים:

<u>מקרה 1:</u> $x \neq \langle M, w \rangle$ $\Leftarrow$ $f(x) = \langle M_{\varnothing} \rangle$ ו- $L(M_{\varnothing}) = \varnothing$ $\Leftarrow$ $\langle M_{\varnothing} \rangle \in L_{\text{REG}}$ $\Leftarrow$ $f(x) \in L_{\text{REG}}$.

<u>מקרה 2:</u> $x = \langle M, w \rangle$ ו- $x \notin L(M)$ $\Leftarrow$ $f(x) = \langle M' \rangle$ ולפי האבחנה $L(M') = \varnothing$ $\Leftarrow$ $\langle M_{\varnothing} \rangle \in L_{\text{REG}}$
$\Leftarrow$ $f(x) \in L_{\text{REG}}$.

אם $x \notin \bar{L}_{\text{acc}}$ $\Leftarrow$ $w \in L(M)$ $\Leftarrow$ $f(x) = \langle M' \rangle$ ולפי האבחנה $L(M') \in PAL$ $\Leftarrow$ $f(x) \in PAL$
$\Leftarrow$ $f(x) \notin L_{\text{REG}}$.

דוגמה 8.15 $L_{\text{acc}} \leqslant L_{\text{REG}}$

תהי L_{REG} השפה

$$L_{\text{REG}} = \left\{ \langle M \rangle \;\middle|\; \text{רגולרית } L(M) \right\} .$$

הוכיחו כי השפה L_{REG} לא כריעה על ידי רדוקציה מ- L_{acc}.

פתרון:

השפה L_{acc} מוגדרת

$$L_{\text{acc}} = \left\{ \langle M, w \rangle \;\middle|\; w \text{ מקבלת } M \right\} .$$

והשפה L_{REG} מוגדרת

$$L_{\text{REG}} = \left\{ \langle M \rangle \;\middle|\; \text{רגולרית } L(M) \right\} .$$

נגדיר הפונקציה הבאה:

$$f(x) = \begin{cases} \langle M' \rangle & x = \langle M, w \rangle \\ \langle M_{PAL} \rangle & x \neq \langle M, w \rangle \end{cases}$$

כאשר M_{PAL} המ"ט שמכריעה את השפה של פלינדרומים, ו- M' מ"ט הבאה:

$M' =$ על כל קלט y:

1) M' בודקת אם y פלינדרום:

- אם כן $\Leftarrow$ מקבלת.
- אם לא מריצה M על w ועונה כמוה.

<u>הוכחת נכונות הרדוקציה</u>

אם $x \in L_{\text{acc}}$ $\Leftarrow$ M מקבלת w $\Leftarrow$ $L(M') = \Sigma^*$ $\Leftarrow$ $f(x) \in REG$.

$x \notin L_{\text{acc}}$ $\Leftarrow$ שני מקרים:

<u>מקרה 1:</u> $x \neq \langle M, w \rangle$ $\Leftarrow$ $f(x) = \langle M_{PAL} \rangle$ ו- $L(M_{PAL}) = PAL$ $\Leftarrow$ $\langle M_{PAL} \rangle \notin L_{\text{REG}}$
$\Leftarrow$ $f(x) \notin L_{\text{REG}}$.

<u>מקרה 2:</u> $x = \langle M, w \rangle$ ו- M לא מקבלת w $\Leftarrow$ $L(M') = PAL$ $\Leftarrow$ $\langle M' \rangle \notin L_{\text{REG}}$
$\Leftarrow$ $f(x) \notin L_{\text{REG}}$.

דוגמה 8.16 $\bar{L}_{\text{acc}} \leqslant L_{M_1 \neg M_2}$

נתונה השפה הבאה:

$$L = \left\{ \langle M_1, M_2, w\rangle \;\middle|\; w \in L(M_1) \wedge w \notin L(M_2) \right\} \ .$$

הוכיחו כי $L \notin RE$ ע"י רדוקציה מ- $\bar{L}_{\text{acc}}$.

פתרון:

<u>פונקצית הרדוקציה:</u>

$$f(x) = \begin{cases} \langle M^*, M, w\rangle & x = \langle M, w\rangle \\ \langle M^*, M_\varnothing, \varepsilon\rangle & x \neq \langle M, w\rangle \end{cases}$$

כאשר

- M^* היא מ"ט שמקבלת כל קלט
- $M_\varnothing$ היא מ"ט שדוחה כל קלט.

<u>נכונת הרדוקציה:</u>

ראשית, f חשיבה כי ניתן לבנות מ"ט שתבדוק האם $x = \langle M, w\rangle$. אם לא, תחזיר קידוד קבוע $\langle M^*, M_\varnothing, \varepsilon\rangle$ ואם כן, תחזיר קידוד $\langle M^*, M, w\rangle$.

נוכיח כי

$$x \in \bar{L}_{\text{acc}} \quad \Leftrightarrow \quad f(x) \in L_{M_1 \neg M_2} \ .$$

אם $x \in \bar{L}_{\text{acc}}$ $\Leftarrow$ שני מקרים:

<u>מקרה 1:</u> $x \neq \langle M, w\rangle$ $\Leftarrow$ $f(x) = \langle M^*, M_\varnothing, \varepsilon\rangle$ ו- $\varepsilon \in L(M^*)$ ו- $\varepsilon \notin L(M_\varnothing)$ $\Leftarrow$ $f(x) \in \bar{L}_{M_1 \neg M_2}$.

<u>מקרה 2:</u> $x = \langle M, w\rangle$ ו- $w \notin L(M)$ $\Leftarrow$ $f(x) = \langle M^*, M, w\rangle$ ו- $w \in L(M^*)$ ו- $w \notin L(M)$ $\Leftarrow$ $f(x) \in L_{M_1 \neg M_2}$.

אם $x \notin \bar{L}_{\text{acc}}$ $\Leftarrow$ $x = \langle M, w\rangle$ ו- $w \in L(M)$ $\Leftarrow$ $f(x) = \langle M^*, M, w\rangle$ ו- $w \in L(M^*)$ ו- $w \notin L(M)$ $\Leftarrow$ $f(x) \notin L_{M_1 \neg M_2}$.

לסיכום, הוכחנו רדוקציה $\bar{L}_{\text{acc}} \leqslant L_{M_1 \neg M_2}$, ומכיוון ש- $\bar{L}_{\text{acc}} \notin RE$ ממשפט הרדוקציה מתקיים $L_{M_1 \neg M_2} \notin RE$.

דוגמה 8.17 $L_{\text{acc}} \leqslant L_{M_1 \subset M_2}$

נתונה השפה הבאה:

$$L = \left\{ \langle M_1, M_2, w\rangle \;\middle|\; w \in L(M_1) \wedge w \notin L(M_2) \right\} \ .$$

הוכיחו כי $L \notin RE$ ע"י רדוקציה מ- L_{acc}.

פתרון:

<u>פונקצית הרדוקציה:</u>

$$f(x) = \begin{cases} \langle M_\varnothing, M'\rangle & x = \langle M, w\rangle \\ \langle M_\varnothing, M_\varnothing\rangle & x \neq \langle M, w\rangle \end{cases}$$

כאשר

- $M_\varnothing$ היא מ"ט שדוחה כל קלט.
- M' היא מ"ט שעל קלט y מתעלמת מ- y ומריצה M על w ועונה כמוה.

<u>אבחנה:</u>

$$L\left(M'\right)=\begin{cases}\Sigma^* & w\in L(M)\\ \varnothing & w\notin L(M)\end{cases}\ .$$

<u>נכונת הרדוקציה:</u>

ראשית, f חשיבה כי ניתן לבנות מ"ט שתבדוק האם $x=\langle M,w\rangle$. אם לא, תחזיר קידוד קבוע $\langle M_\varnothing,M_\varnothing\rangle$ ואם כן, תחזיר קידוד $\langle M^*,M'\rangle$, כאשר $M_\varnothing$ היא קידוד קבוע ו- M' נוצר ע"י הוספת קוד ל- $\langle M\rangle$ המוחק את הקלט y ורושם w במקומו.

נוכיח כי

$$x\in L_{\text{acc}}\quad\Leftrightarrow\quad f(x)\in L_{M_1\subset M_2}\ .$$

אם $x\in L_{\text{acc}}$ $\Leftarrow$ $x=\langle M,w\rangle$ ו- $w\in L(M)$ $\Leftarrow$ $f(x)=\langle M_\varnothing,M'\rangle$ ולפי האבחנה $L\left(M'\right)=\Sigma^*$ $\Leftarrow$ $L\left(M_\varnothing\right)\subset L\left(M'\right)$ $\Leftarrow$ $f(x)\in L_{M_1\subset M_2}$.

$x\notin L_{\text{acc}}$ $\Leftarrow$ שני מקרים:

<u>מקרה 1:</u> $x\neq\langle M,w\rangle$ $\Leftarrow$ $f(x)=\langle M_\varnothing,M_\varnothing\rangle$ ו- $L\left(M_\varnothing\right)=L\left(M_\varnothing\right)$ $\Leftarrow$ $f(x)\notin L_{M_1\subset M_2}$.

<u>מקרה 2:</u> $x=\langle M,w\rangle$ ו- $w\notin L(M)$ $\Leftarrow$ $f(x)=\langle M_\varnothing,M'\rangle$ ולפי האבחנה $L\left(M'\right)=\varnothing$ $\Leftarrow$ $L\left(M'\right)=L\left(M_\varnothing\right)$ $f(x)\notin L_{M_1\subset M_2}$.

לסיכום, הוכחנו רדוקציה $L_{\text{acc}}\leqslant L_{M_1\subset M_2}$, ומכיוון ש- $L_{\text{acc}}\notin R$ ממשפט הרדוקציה מתקיים $L_{M_1\subset M_2}\notin R$.

שיעור 9
מבוא לסיבוכיות

9.1 הגדרה של סיבוכיות

הערה 9.1

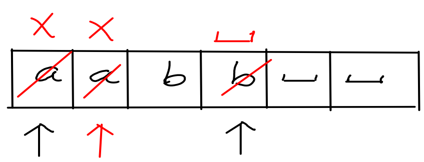
זמן ריצה של מ"ט M על קלט w, נמדד ביחס לגודל הקלט w, כלומר $f\left(|w|\right)$.

הגדרה 9.1

נאמר כי ניתן להכריע שפה L בזמן $f(n)$, אם קיימת מ"ט M המכריעה את L ולכן קלט $w \in \Sigma^*$, זמן הריצה של M על w חסום ע"י $f\left(|w|\right)$.

דוגמה 9.1

נבנה מ"ט M המכריעה השפה $L = \left\{a^n b^n \mid n \geqslant 0\right\}$.



<u>התאור של M:</u>

על קלט w:

(1) אם התו שמתחת לראש הוא $\sqcup$ $\Leftarrow$ מקבלת.

(2) אם התו שמתחת לראש הוא b $\Leftarrow$ דוחה.

(3) מוחקת את התו שמתחת לראש ע"י X.

(4) מזיזה את הראש ימינה עד התו הראשון משמאל ל- $\sqcup$.

- אם התו הוא a או X $\Leftarrow$ דוחה.
- מוחקת את התו שמתחת לראש ע"י $\sqcup$, מזיזה את הראש שמאלה עד התו הראשון מימין ל- X וחוזרת ל- (1).

<u>זמן הריצה</u>

- $\dfrac{|w|}{2}$ איטרציות.
- בכל איטרציה מבצעים $O\left(|w|\right)$ צעדים.

$$\frac{|w|}{2} \cdot O\left(|w|\right) = O\left(|w|^2\right) \ .$$

הגדרה 9.2 זמן הריצה

זמן הריצה של מ"ט M על קלט w היא פונקציה $f(|w|)$ השווה למספר הצעדים הנדרש בחישוב של M על w.

הערה 9.2

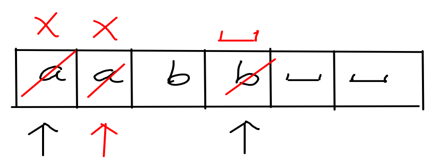
זמן הריצה של מ"ט נמדד ביחס לגודל הקלט $|w|$.

הגדרה 9.3

אומרים כי ניתן להכריעה שפה L בזמן $f(n)$ אם קיימת מ"ט M המכריעה את L כך שלכל $w \in \Sigma^*$, זמן הריצה של M על w חסום ע"י $f(|w|)$.

דוגמה 9.2

נבנה מ"ט M עם סרט יחיד שמכריעה את השפה $L = \left\{a^n b^n \mid n \geqslant 0\right\}$.



<u>התאור של M:</u>

על קלט w:

(1) אם התו שמתחת לראש הוא $\sqcup$ $\Leftarrow$ מקבלת.

(2) אם התו שמתחת לראש הוא b $\Leftarrow$ דוחה.

(3) מוחקת את התו שמתחת לראש ע"י X.

(4) מזיזה את הראש ימינה עד התו הראשון משמאל ל- $\sqcup$.

- אם התו הוא a או X $\Leftarrow$ דוחה.
- מוחקת את התו שמתחת לראש ע"י $\sqcup$, מזיזה את הראש שמאלה עד התו הראשון מימין ל- X וחוזרת ל- (1).

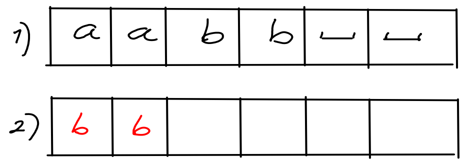
<u>זמן הריצה</u>

- M מבצעת $\frac{|w|}{2}$ איטרציות.
- בכל איטרציה M סורקת את הסרט פעמיים וזה עולה $O(|w|)$.
- לכן סה"כ זמן הריצה של M חסום ע"י

$$\frac{|w|}{2} \cdot O(|w|) = O\left(|w|^2\right) .$$

דוגמה 9.3

נבנה מ"ט מרובת סרטים M' שמכריעה את השפה $L=\{a^nb^n \mid n\geqslant 0\}$.



<u>התאור של M':</u>

על קלט w:

(1) מעתיקה את ה- b-ים לסרט 2 (ותוך כדי בודקת האם w מהצורה a^*b^*). $O(|w|)$.

(2) מזיזה את הראשים לתחילת הסרטים. $O(|w|)$.

(3) אם שני הראשען מצביעים על $\sqcup$ $\Leftarrow$ מקבלת.

(4) אם אחד הראשים מצביע על $\sqcup$ והשני לא $\Leftarrow$ לא.

(5) מזיזה את שהע הראשים ימינה וחוזרת לשלב (3). שלבים (3-5): $O(|w|)$.

<u>זמן הריצה</u>

זמן הריצה של M' הוא $O(|w|)$.

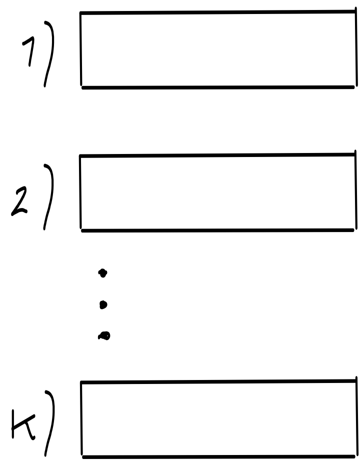
9.2 יחס בין הסיבוכיות של מ"ט סרט יחיד ומטמ"ס

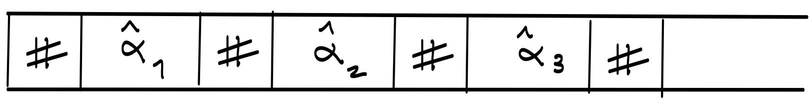
משפט 9.1

לכל מ"ט מרובת סרטים M הרצה בזמן $f(n)$ קיימת מ"ט סרט יחיד M' השקולה ל- M ורצה בזמן $O(f^2(n))$.

הוכחה:
בהינתן מ"ט מרובת סרטים M, הרצה בזמן $f(n)$, נבנה מ"ט עם סרט יחיד M' באותו אופן כמו בהוכחת השקילות במשפט 3.1.

כלומר, M' שומרת את התוכן של k סרטים של M על הסרט היחיד שלה (עם הפרדה ע"י #), ובכל צעד חישוב, M' סורקת את הסרט שלה כדי לזהות שת האותיות שמתחת לראשים (שמסומנות ב- $\hat{\alpha}$) ואחרי זה, משתמשת בפונקצית המעברים של M, וסורקת את הסרט פעם נוספת כדי לעדכן את התוכן בכל אחד מהסרטים ואת מיקום הראש בכל אחד מהסרטים.





כמה לוקח ל- M' לסרוק את הסרט שלה? מכיוון שהסרט של M' מכיל את התוכן של k הסרטים של M, והגודל של כל אחד מהסרטים של M חסום ע"י $f(n)$, גודל הסרט של M' חסום ע"י

$$k \cdot f(n) = O\left(f(n)\right) \ .$$

העלות של הסריקה של M' לסרט שלה היא $O\left(f(n)\right)$ וזה עלות של צעד חישוב בריצה של M' על הקלט.

מכיוון ש- M רצה בזמן $f(n)$, זמן היצרה של M' חסום ע"י

$$f(n) \cdot O\left(f(n)\right) = O\left(f^2(n)\right) \ .$$

■

9.3 יחס בין הסיבוכיות של מ"ט דטרמיניסטית ומ"ט א"ד

הגדרה 9.4

בהינתן מ"ט א"ד M, זמן הריצה של M על קלט w, היא פונקציה $f\left(|w|\right)$ השווה למספר הצעדים בחישוב המקסימלי של M על w.

משפט 9.2

לכל מ"ט א"ד N הרצה בזמן $f(n)$, קיימת מ"ט דטרמיניסטית D השקולה ל- N ורצה בזמן $2^{)(f(n))}$.

הוכחה:
בהינתן מ"ט א"ד N הרצה בזמן $f(n)$ מ"ט דטרמיניסטית D באותו אופן כמו בהוכחת השקילות במשפט 4.1.

כלומר, בהינתן קלט w, D תסרו' את עץ החישוב של N ו- w לרוחב ותקבל כל אחד החישובים של N המסתיים ב- q_{acc}.

בהינתן קלט w באורך n:

- כל מסלול בעץ החישוב של N על w חסום ע"י $f(n)$.
- מספר החישובים ש- D מבצעת חסום ע"י מספר הקודקודים בעץ החישוב של N ו- w.
- מכיוון שמספר הבנים של כל קודקוד בעץ החישוב חסום ע"י

$$C = 3|Q| \cdot |\Gamma|$$

מספר הקודקודים בעץ החישוב חסום ע"י

$$C^0 + C^2 + \ldots C^{f(n)} \leqslant C^{f(n)+1} = C \cdot C^{f(n)} \ .$$

ולכן זמן הריצה של D חסום ע"י

$$f(n) \cdot C \cdot C^{f(n)} \leqslant C^{f(n)} \cdot C^{f(n)} = C^{2f(n)} = \left(C^2\right)^{f(n)} = 2^{C' \cdot f(n)} = 2^{O(f(n))} \ .$$

■

נתייחס כאן לשני החסמים הבאים:

1) חסם פולינומיאלי הוא חסם מהצורה n^c עבטר $c > 0$ כלשהו.

2) חסם אקספוננציאלי הוא חסם מהצורה 2^{n^c} עבור $c > 0$ כלשהו.

הגדרה 9.5 בעיית הכרעה

בעיית הכרעה מוגדרת באופן הבא:

"בהינתן קלט כלשהו, האם הקלט מקיים תנאי מסוים "

דוגמה 9.4

בהינתן מספר n, האם n ראשוני?

כל בעיית הכרעה ניתן לתאר כפשה שקולה:

$$L_{\text{prime}} = \left\{ \langle n \rangle \ \middle| \ \text{ראשוני } n \right\} \ .$$

משפט 9.3

שפה $\equiv$ בעיית הכרעה .

הגדרה 9.6 אלגוריתם זמן פולינומיאלי

אומרים כי אלגוריתם A מכריעה בעייה בזמן פולינומיאלי אם קיים קבוע $c > 0$ כך שזמן הריצה של A על כל קלט w חסום ע"י $O\left(|w|^c\right)$.

משפט 9.4 התזה של צירץ' (Church Thesis)

אם קיים אלגוריתם המכריע בעייה בזמן פולינומיאלי, אז קיימת מ"ט דטרמיניסטית המכריעה את השפה השקולה לבעייה זו בזמן פולינומיאלי.

$$\text{מכונת טיורינג} \equiv \text{אלגוריתם מכריעה} \ .$$

9.4 המחלקה P

הגדרה 9.7 המחלקה P

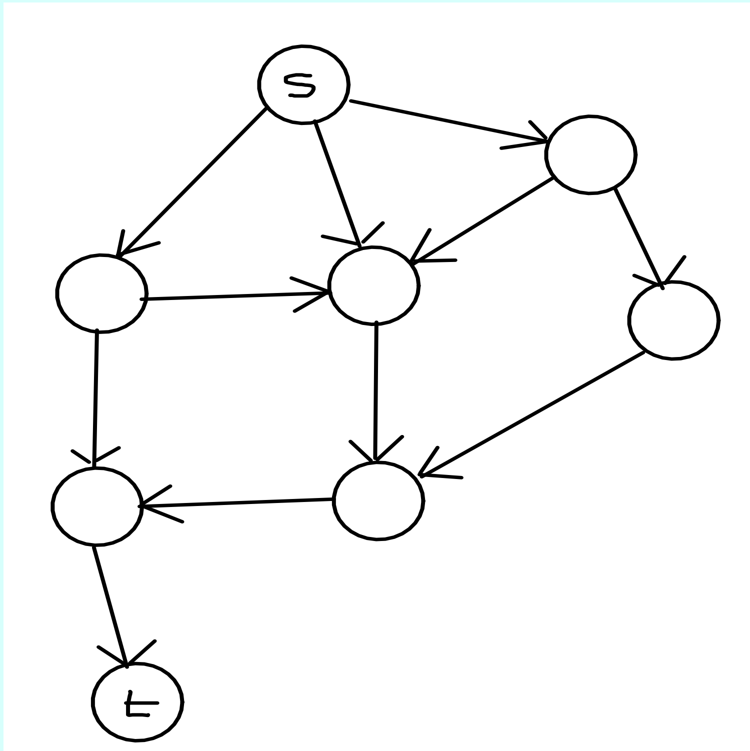
המחלקה P היא אוסף כל הבעיות (השפות) שקיים עבורן אלגוריתם (מכונת טיורינג דטרמיניסטית) המכריע אותן בזמן פולינומיאלי.

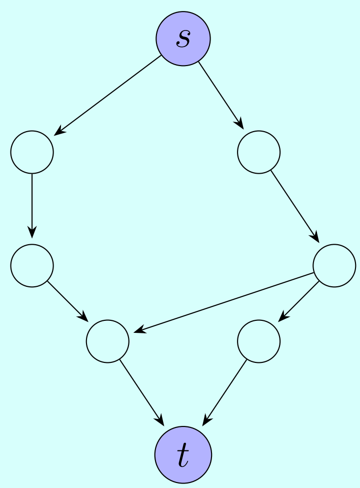
דוגמה 9.5

$$L = \left\{a^n b^n \;\middle|\; n \geqslant 0\right\} \in P \ .$$

9.5 בעיית PATH

הגדרה 9.8 בעיית המסלול בגרף מכוון





קלט: גרף מכוון $G=(V,E)$ ושני קודקודים $s,t\in V$.

פלט: האם קיים מסלול ב- G מ- s ל- t?

$$PATH = \left\{ \langle G,s,t\rangle \;\middle|\; \text{קיים מסלול ב- } G \text{ מ- } s \text{ ל- } t \right\}$$

משפט 9.5

$$PATH \in P \ .$$

הוכחה: נבנה אלגוריתם A עבור הבעייה $PATH$.

$A=$ על קלט $\langle G,s,t\rangle$:

1) צובע את s.

2) מבצע $|V|-1$ פעמים:

- לכל צלע $(u,\mathrm{v})\in E$:
  - אם u צבוע $\Leftarrow$ צבע את v.

3)
- אם t צבוע $\Leftarrow$ החזיר "כן".
- אחרת $\Leftarrow$ החזיר "לא".

זמן ריצה של האלגוריתם הוא $O\left(|V|\cdot|E|\right)$ פולינומיאלי במספר הקודקודים $|V|$.

האם זה פולינומיאלי בגודל הקלט $|\langle G\rangle|$?

איך נקודד את G?

- נניח כי $|V|=n$ ו- $V=\{1,2,3,\quad\cdots\quad,n\}$.
- נניח כי הצלעות נתונות ע"י מטריצה M בגודל $n\times n$ כך ש-

$$M_{ij}=\begin{cases}1 & (i,j)\in E\\ 0 & (i,j)\notin E\end{cases}\ .$$

- נניך כי מספרים מקודדים בבסיס ביניארי.
- אזי גודל הקידוד של G שווה $n^2+n\log_2 n$, כלומר

$$|\langle G\rangle|=\Omega\left(|V|^2\right)\qquad\Rightarrow\qquad |V|=O\left(|\langle G\rangle|\right)\ .$$

ולכן כל אלגוריתם הרץ בזמן פולינומיאלי במספר הקודקודים $|V|$ ירוץ בזמן פולינומיאלי בגודל הקידוד $|\langle G\rangle|$.

ולכן A רץ בזמן פולינומיאלי בגודל הקלט. ■

9.6 בעיית $RELPRIME$

הגדרה 9.9 מספרים זרים (Relatively prime)

שני מספרים x,y זרים אם המחלק המשותף הגדול ביותר, מסומן $\gcd(x,y)$, שווה 1.

הגדרה 9.10 בעיית $RELPRIME$

קלט: שני מספרים x ו- y.

פלט: האם x ו- y זרים?

$$RELPRIME=\left\{\langle x,y\rangle\ \middle|\ \gcd(x,y)=1\right\}\ .$$

משפט 9.6

$$RELPRIME \in P \ .$$

הוכחה: נבנה אלגוריתם A המכריע את $RELPRIME$ בזמן פולינומיאלי.

האלגוריתם מבוסס על העובדה ש-

$$\gcd(x,y)=1 \qquad \Leftrightarrow \qquad \langle x,y\rangle \in RELPRIME \ .$$

ולכן נשתמש באלגוריתם האוקלידי לחישוב gcd:

$$\gcd(x,y)=\begin{cases} x & y=0\\ \gcd(y,x \mod y) & y\neq 0\end{cases} \ .$$

הוכחה: נסמן $d=\gcd(x,y)$. אזי קיימים שלמים s,t כך ש- $sx+ty=d$. נסמן $r=x \mod y$. ז"א $x=qy+r$. לכן

$$s(qy+r)+ty=d \quad\Rightarrow\quad sr+(t+sq)y=d \quad\Rightarrow\quad \gcd(x,y)=d=\gcd(y,r) \ .$$

לדוגמה:

$$\gcd(18,32)=\gcd(32,18)=\gcd(18,14)=\gcd(14,4)=\gcd(4,2)=\gcd(2,0)=2 \ .$$

האלגוריתם האוקלידי:

על קלט x ו- y:

(1) כל עוד $y\neq 0$:
- $x \mod y \to x$
- $\text{swap}(x,y)$

(כלומר מחליפים בין x ו- y).

(2) מחזירים את x.

האלגוריתם A המכריע $RELPRIME$:

$A=$ על קלט $\langle x,y\rangle$:

(1) מריץ את האלגוריתם האוקלידי על x ו- y.
- אם האלגורים האוקלידי החזיר $1 \Leftarrow$ מקבל.
- אחרת $\Leftarrow$ דוחה.

נכונות האלגוריתם נובעת מנכונות האלגוריתם האוקלדי.

נוכיח כי A רץ בזמן פולינומיאלי בגודל הקלט.

טענת עזר:

אם $y<x$ אזי $x \mod y<\frac{x}{2}$.

<u>הוכחה:</u>

יש שתי אפשרויות:

- אם $y \leqslant \dfrac{x}{2}$ אזי
$$x \mod y < y \leqslant \frac{x}{2} \ .$$
- נניח ש- $\dfrac{x}{2} < y < x$.
מכיוון ש- $x = qy + (x \mod y)$, וגם $x < 2y$ אז בהכרח $q < 2$ ולכן $x = y + (x \mod y)$ ולכן $x - y = x \mod y$.

לפיכך
$$x \mod y = x - y < \frac{x}{2} \ .$$

■

לפי טענת העזר, אחרי כל איטרציה x קטן בלפחות חצי.

מכיוון שבכל איטרציה מחליפים בין x ו- y, אחרי כל שתי איטרציות גם x וגם y קטנים בלפחות חצי.

ולכן לאחר $\log_2 x + \log_2 y$ איטרציות לפחות x או y שווים ל- 0.

ולכן מספר האיטרציות באלגוריתם האוקלידי חסום ע"י $\log_2 x + \log_2 y$, וזה בדיוק זמן הריצה של האלגוריתם A.

ולכן A רץ בזמן פולינומיאלי בגודל הקלט.

ולכן
$$RELPRIME \in P \ .$$

■

שיעור 10
המחלקה P והמחלקה NP

10.1 המחלקה P

- המחלקה P היא אוסף כל הבעיות שקיים עבורן אלגוריתם המכריע אותן בזמן פולינומיאלי.

אלגוריתם מכריע $\equiv$ מ"ט דטרמיניסטית ,

בעיית הכרעה $\equiv$ שפה ,

- אלגוריתם A מכריע בעייה בזמן פולינומיאלי אם קיים קבוע $c>0$ כך שזמן הריצה של A על כל קלט w חסום ע"י $O\left(|w|^c\right)$.

10.2 דוגמאות לבעיות ב- P

(1

$$PATH=\left\{\langle G,s,t\rangle \ \middle|\ \text{גרף מכוון המכיל מסלול מ- } s \text{ ל- } t \ G\right\}\in P$$

(2

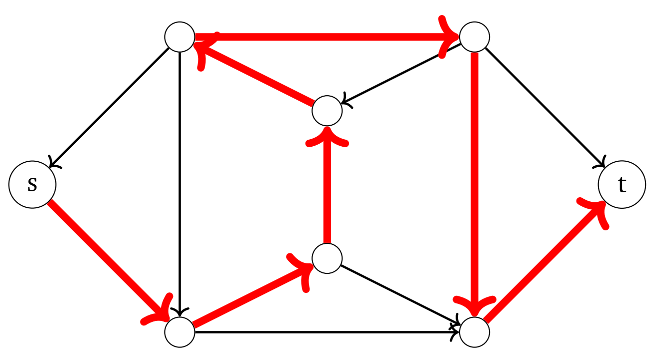
$$RELPRIME=\left\{\langle x,y\rangle \ \middle|\ x \text{ ו- } y \text{ זרים}\right\}\in P$$

10.3 בעיית המסלול ההמילטוני HAMPATH

הגדרה 10.1 HAMPATH

בהינתן גרף מכוון $G=(V,E)$ ושני קודקודים $s,t\in V$. מסלול המילטוני מ- s ל- t ב- G הוא מסלול מ- s ל- t שעובר דרך כל קודקוד בגרף בדיוק פעם אחת.

לדוגמה:



הגדרה 10.2 בעיית HAMPATH

קלט: גרף מכוון $G=(V,E)$ ושני קודקודים $s,t\in V$.

<u>פלט:</u> האם G מכיל מסלול המילטוני מ- s ל- t?

$$HAMPATH = \{\langle G,s,t\rangle \mid \text{גרף מכוון המכיל מסלול המילטוני מ- } s \text{ ל- } t\text{?} \ G\}$$

נשאל שאלה: האם $HAMPATH \in P$?

לא ידוע האם קיים אלגוריתם המכריע את $HAMPATH$ בזמן פולינומיאלי (שאלה פתוחה).

- בהינתן קלט $\langle G,s,t\rangle$, האם $\langle G,s,t\rangle \in HAMPATH$?
- נענה על שאלה אחרת:

  בהינתן קלט $\langle G,s,t\rangle$, ומחרוזת y, האם $\langle G,s,t\rangle \in HAMPATH$?
- יתן לבדוק האם y היא מסלול המילטוני מ- s ל- t ב- G בזמן פולינומיאלי ולענות בהתאם.
- במקרה זה, אומרים כי $HAMPATH$ ניתנת לאימות בזמן פולינומיאלי.

10.4 אלגוריתם אימות

הגדרה 10.3 אלגוריתם אימות

אלגוריתם אימות עבור בעייה A הוא אלגוריתם V כך שלכל קלט $w \in \Sigma^*$ מתקיים:

$w \in A$ אם ורק אם קיימת מחרוזת (עדות) y באורך פולינומיאלי ב- $|w|$ כך ש- V מקבל את הזוג (w,y) כלומר:

- אם $w \in A \quad \Leftarrow \quad \exists y \ : \ V(w,y) = T$.
- אם $w \notin A \quad \Leftarrow \quad \forall y \ : \ V(w,y) = F$.

הערה 10.1

- זמן ריצה של אלגוריתום אימות נמדד ביחס לגודל הקלט $|w|$.
- אלגוריתם אימות פולינומיאלי אם הוא רץ בזמן פולינומיאלי.

10.5 המחלקה NP

הגדרה 10.4 המחלקה NP

המחלקה NP היא אוסף כל הבעיות שקיים עבורן אלגוריתם אימות פולינומיאלי.

משפט 10.1 $HAMPATH \in NP$

בעיית המסלול ההמילטוני $HAMPATH$:

<u>קלט:</u> גרף מכוון $G=(V,E)$ ושני קודקודים $s,t \in V$.

<u>פלט:</u> האם G מכיל מסלול המילטוני מ- s ל- t?

$$HAMPATH = \left\{\langle G,s,t\rangle \;\middle|\; \text{גרף מכוון המכיל מסלול המילטוני מ- } s \text{ ל- } t \text{? } G \right\}$$

הוכיחו כי $HAMPATH \in NP$.

הוכחה: נבנה אלגוריתם אימות V עבור $HAMPATH$.

$V =$ על קלט $(\langle G,s,t\rangle, y)$:

1) בודק האם y היא סדרה של

$$u_1, u_2, \quad \ldots \quad u_n$$

השונים זה מזה.

- אם לא $\Leftarrow$ דוחה.

2) בודק האם $u_1 = s$ ו- $u_n = t$.

- אם לא $\Leftarrow$ דוחה.

3) בודק שכל הצלעות (u_i, u_{i+1}) (לכל $1 \leqslant i \leqslant n$) קיימות ב- G.

- אם כן $\Leftarrow$ מקבל.
- אם לא $\Leftarrow$ דוחה.

<u>נכונות</u>

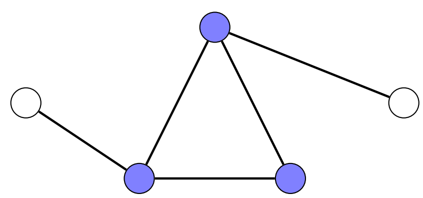
- זמן הריצה של האלגוריתם הוא פולינומיאלי בגודל הקלט.
- אם $\langle G,s,t\rangle \in HAMPATH$ $\Leftarrow$ G מכיל מסלול המילטוני מ- s ל- t $\Leftarrow$ עבור y שהוא קידוד של מסלול זה, V יקבל את הזוג $(\langle G,s,t\rangle, y)$.
- אם $\langle G,s,t\rangle \notin HAMPATH$ $\Leftarrow$ G לא מכיל מסלול המילטוני מ- s ל- t $\Leftarrow$ לכל y, האלגוריתם ידחה את הזוג $(\langle G,s,t\rangle, y)$.

■

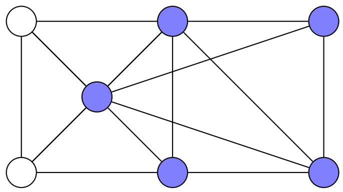
הגדרה 10.5 קליקה

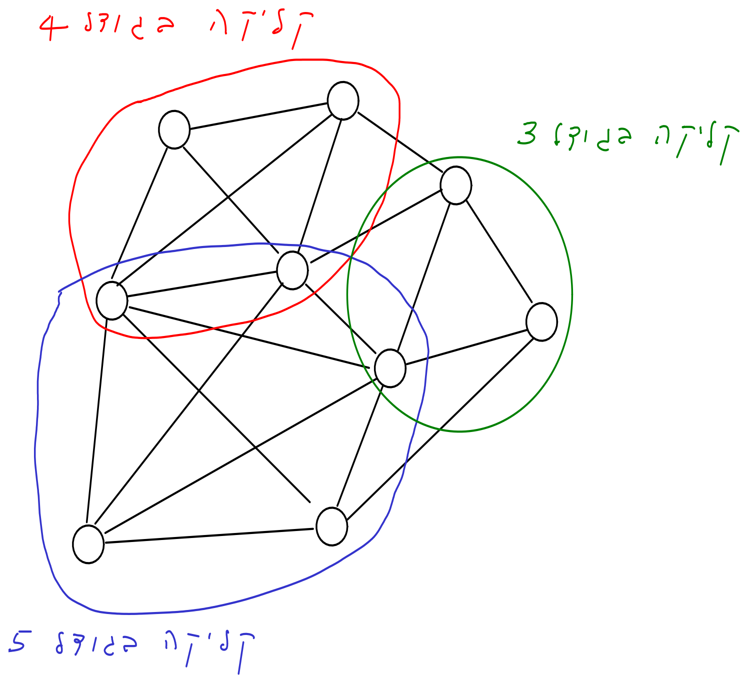
בהינתן גרף לא מכוון $G = (V,E)$, קליקה ב- G היא תת-קבוצה של קודקודים $C \subseteq V$ כך שלכל שני קודקודים $u, \mathrm{v} \in C$ מתקיים $(u, \mathrm{v}) \in E$.

קליקה בגודל $k = 3$:



קליקה בגודל $k=5$:





הגדרה 10.6 בעיית הקליקה

<u>קלט:</u> גרף לא מכוון $G=(V,E)$ ומספר k.

<u>פלט:</u> האם G קליקה בגודל k?

$$CLIQUE = \left\{ \langle G,k\rangle \;\middle|\; G \text{ גרף לא מכוון המכיל קליקה בגודל } k \right\}$$

משפט 10.2 $CLIQUE\in NP$

$$CLIQUE\in NP\ .$$

הוכחה: נבנה אלגוריתם אימות V עבור $CLIQUE$.

$V=$ על קלט $(\langle G,k\rangle ,y)$:

1) בודק האם y היא קבוצה של k קודקודים שונים מ- G.
   - אם לא $\Leftarrow$ דוחה.
2) בודק האם כל שני קודקודים מ- y מחוברים בצלע ב- G.
   - אם כן $\Leftarrow$ מקבל.

- אם לא $\Leftarrow$ דוחה.

■

הגדרה 10.7 בעיית $SubSetSum$

קלט: קבוצת מספרים $S=\{x_1,x_2, \quad \ldots \quad x_n\}$ ומספר t.

פלט: האם קיימת תת-קבוצה של S שסכום איבריה שווה t?

$$SubSetSum = \left\{ \langle S,t\rangle \mid \text{קיימת } S \subseteq Y \text{ כך ש- } \sum_{x\in Y} x = t \right\}$$

משפט 10.3 $SubSetSum \in NP$

$$SubSetSum \in NP \ .$$

הוכחה: נבנה אלגוריתם אימות V עבור $SubSetSum$.

$V=$ על קלט $(\langle S,t\rangle, y)$:

1) בודק האם y היא תת-קבוצה של S.
   - אם לא $\Leftarrow$ דוחה.

2) בודק האם סכום המספרים ב- y שווה t.
   - אם לא $\Leftarrow$ דוחה.
   - אחרת $\Leftarrow$ מקבל.

■

10.6 הקשר בין NP למ"ט א"ד

NP=Non-deterministic polynomial-time.

משפט 10.4

לכל בעייה A:

$A \in NP$ אם ורק אם קיימת מ"ט א"ד המכריעה את A בזמן פולינומיאלי.

דוגמה 10.1

נבנה מ"ט א"ד M המכריעה את $CLIQUE$ בזמן פולינומיאלי.

$M=$ על קלט $\langle G,k\rangle$:

- בוחרת באופן א"ד קבוצה y של k קודקודים מ- G.
- בודקת האם כל שני קודקודים מ- y מחוברים בצלע ב- G.

* אם כן $\Leftarrow$ מקבלת.
* אחרת $\Leftarrow$ דוחה.

אלגוריתם אימות $\equiv$ מ"ט א"ד.

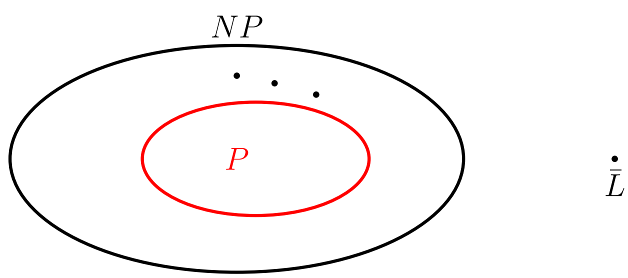
10.7 הקשר בין המחלקה P ו- NP

$P =$ כל הבעיות שניתן להכריע בזמן פולינומיאלי.

$NP =$ כל הבעיות שניתן לאמת בזמן פולינומיאלי.

משפט 10.5

$$P \subseteq NP\ .$$



שאלה פתוחה: האם $P = NP$?

משפט 10.6

P סגורה תחת משלים.

הוכחה: אם $A \in P$ אזי גם $\bar{A} \in P$. ■

הגדרה 10.8 $CoNP$

$$CoNP = \{A \mid \bar{A} \in NP\ .\}$$

לדוגמה:

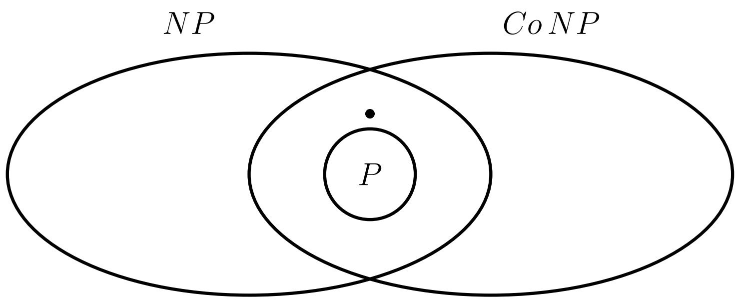
$$\overline{HAMPATH} \in CoNP\ .$$

$$\overline{CLIQUE} \in CoNP\ .$$

שאלה פתוחה: האם $NP = CoNP$?

משפט 10.7

$$P \subseteq NP \cap CoNP\ .$$



שאלה פתוחה: האם $P = NP \cap CoNP$?

נדון בשאלה המרכזית: האם $P = NP$?

הגדרה 10.9 פונקציה פולינומיאלית

בהינתן פונקציה $f : \Sigma^* \to \Sigma^*$, אומרים כי f חשיבה בזמן פולינומיאלי אם קיים אלגוריתם (מ"ט דטרמיניסטית) המחשב את f בזמן פולינומיאלי.

הגדרה 10.10 רדוקציה פולינומיאלית

בהינתן שתי הבעיות A ו- B. אומרים כי A ניתנת לרדוקציה פולינומיאלית ל- B, ומסמנים $A \leqslant_P B$, אם קיימת פונקציה $f : \Sigma^* \to \Sigma^*$ המקיימת:

(1 f חשיבה בזמן פולינומיאלי

(2 לכל $w \in \Sigma^*$:

$$w \in A \quad \Leftrightarrow \quad f(w) \in B \ .$$

משפט 10.8 משפט הרדוקציה

לכל שתי בעיות A ו- B, אם $A \leqslant_P B$ אזי

(1 אם $B \in P$ אזי $A \in P$.

(2 אם $B \in NP$ אזי $A \in NP$.

מסקנה מ- (1) ו- (2):

(3 אם $A \notin P$ אזי $B \notin P$.

(4 אם $A \notin NP$ אזי $B \notin NP$.

הוכחה: מכיוון שקיימת רדוקציה $A \leqslant_P B$, קיימת פנקציה f חשיבה בזמן פולנומיאלי המקיימת, לכל $w \in \Sigma^*$,

$$w \in A \quad \Leftrightarrow \quad f(w) \in B \ .$$

יהי M_f האלגוריתם שמחשבת את f בזמן פולינומיאלי.

(1) נוכיח כי אם $B \in P$ אזי $A \in P$.

יהי M_B האלגוריתם שמכריע עת B בזמן פולינומיאלי. נבנה אלגוריתם M_A המכריע את A בזמן פולינומיאי.

<u>התאור של M_A</u>

$M_A =$ על כל קלט w:

1. מחשב את $f(w)$ ע"י M_f.
2. מריץ את M_B על $f(w)$ ועונה כמוה.

נוכיח כי M_A מכריע את A בזמן פולינומיאלי:

- אם $A \Leftarrow w \in A$ ... $w \in A \Rightarrow f(w) \in B \Rightarrow M_B$ מקבל את $f(w) \Rightarrow M_A$ מקבל את w.
- אם $w \notin A \Rightarrow f(w) \notin B \Rightarrow M_B$ דוחה את $f(w) \Rightarrow M_A$ דוחה את w.

נוכיח כי זמן הריצה של M_A הוא פולינומיאי בגודל הקלט $|w|$ בזמן פולינומיאלי:

- נסמן ב- P_f את הפולינום של M_f.
- נסמן ב- P_B את הפולינום של M_B.

זמן הריצה של M_A על קלט w שווה

$$P_f\left(|w|\right) + P_B\left(|f(w)|\right)$$

מכיוו ש- $|f(w)| \leqslant P_f\left(|w|\right)$, זמו הריצה של M_A על w חסום ע"י

$$P_f\left(|w|\right) + P_B\left(P_f\left(|w|\right)\right) = P_f\left(|w|\right) + \left(P_B \circ P_f\right)\left(|w|\right)$$

כאשר $P_B \circ P_f$ מסמן את ההרכבה של שני פולינומים. לכן M_A רץ בזמן פולינומיאלי בגודל $|w|$.

■

שיעור 11
NP שלמות

11.1 המחלקות NPC ו- NPH

הגדרה 11.1 NP-hard

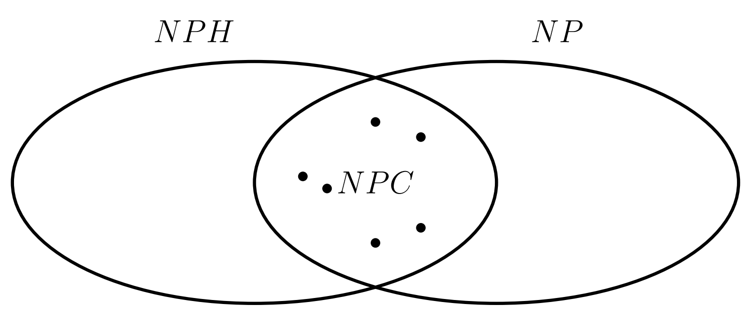
בעייה B נקראת NP קשה אם לכל בעייה $A \in NP$ קיימת רדוקציה $A \leqslant_P B$.

הגדרה 11.2 NP-complete

בעייה B נקראת NP שלמה אם

1) $B \in NP$

2) לכל בעייה $A \in NP$ קיימת רדוקציה $A \leqslant_p B$.



משפט 11.1

אם B בעייה NP שלמה וגם $B \in P$ אזי $P = NP$.

הוכחה:

- הוכחנו כבר ש- $P \subseteq NP$.
- נוכיח כי $NP \subseteq P$.

לכל בעייה $A \subseteq NP$ קיימת רדוקציה $A \leqslant_P B$ ומכיוון ש- $B \in P$, ממשפט הרדוקציה מתקיים $A \in P$.

■

מסקנה 11.1

אם $A \leqslant_P B$ אזי $\bar{A} \leqslant_P \bar{B}$.

משפט 11.2

אם $A \leqslant_p B$ וגם $B \leqslant_p C$ אזי $A \leqslant_p C$.

הוכחה:

■

משפט 11.3

תהי B בעייה NP- שלמה. אזי לכל בעייה $C \in NP$, אם $C \leqslant_p B$ אזי גם C היא NP שלמה.

הוכחה: מכיוון ש- B היא NP-שלמה, לכל בעייה $A \in NP$ קיימת רדוקציה $A \leqslant_p B$. מכיוון ש- $C \leqslant_p B$ מהטרנזטיביות מתקיים $A \leqslant_p C$ לכל בעייה $A \in NP$.
ולכן C היא NP- שלמה.

■

11.2 בעיית הספיקות

הגדרה 11.3

נוסחת CNF ϕ היא נוסחה בוליאנית מעל n משתנים $x_1, x_2, \ldots, x_n$ המכילה m פסוקיות $C_1, C_2, \ldots, C_m$, כאשר כל פסוקית מכילה אוסף של ליטרלים $(x_i \backslash \bar{x}_i)$ המחוברים ע"י OR $(\vee)$ בוליאני והפסוקיות מחוברים ע"י AND $(\wedge)$ בוליאני.

לדוגמה

$$\phi = \left(\overset{C_1}{x_1 \vee \bar{x}_2 \vee x_4 \vee \bar{x}_7}\right) \wedge \left(\overset{C_2}{x_3 \vee x_5 \vee \bar{x}_8}\right) \wedge \quad \cdots$$

הגדרה 11.4 נוסחת CNF ספיקה

נוסחת CNF ϕ היא ספיקה אם קיימת השמה למשתנים $x_1, x_2, \ldots, x_n$ ע"י $T \backslash F$ כך ש- ϕ מקבלת ערך T, כלומר בכל פסוקית ישנו לפחות ליטרל אחד שקיבל ערך T.

11.3 בעיית SAT

הגדרה 11.5 בעיית SAT

קלט: נוסחת CNF, ϕ.
<u>פלט</u>: האם ϕ ספיקה?

$$SAT = \{\langle\phi\rangle \mid \phi \text{ נוסחת } CNF \text{ ספיקה}\}$$

משפט 11.4 $SAT \in NP$

$$SAT \in NP \ .$$

הוכחה: נבנה אלגוריתם אימות V עבור SAT.

$V =$ על קלט $(\langle\phi\rangle, y)$:

1) בודק האם y היא השמה למשתנים $x_1, x_2, \ldots, x_n$.

- אם לא $\Leftarrow 3CNF$ דוחה.

2) בודק האם השמה זו מספקת את ϕ.

- אם כן $\Leftarrow$ מקבל.
- אם לא $\Leftarrow$ דוחה.

■

11.4 משפט קוק לוין

משפט 11.5 (1973) משפט קוק לוין

הבעיית SAT היא NP - שלמה.

רעיון ההוכחה:
לכל $A \in NP$, $SAT \leqslant_p A$.
לכל $w \in \Sigma^*$:

$$w \in A \quad \Leftrightarrow \quad f(w) \in SAT \ ,$$

כאן $f(w) = \langle \phi_w \rangle$.

מסקנה 11.2

$$P = NP \quad \Leftrightarrow \quad SAT \in P \ .$$

11.5 גרסאות של $kSAT$

ישנן לכל היותר k ליטרלים בכל פסוקית:

- $1SAT \in P$.

$$\phi = x_1 \wedge \bar{x}_2 \wedge x_3 \wedge \cdots$$

- $2SAT \in P$.

$$\phi = (x_1 \vee \bar{x}_2) \wedge (x_2 \vee x_4) \wedge \cdots$$

- $3SAT$ היא NP - שלמה.

11.6 בעיית $3SAT$

הגדרה 11.6 בעיית $3SAT$

<u>קלט:</u> נוסחת $3CNF$, ϕ.
<u>פלט:</u> האם ϕ ספיקה?

$$3SAT = \{\langle \phi \rangle \mid \text{ספיקה } 3CNF \text{ נוסחת } \phi\}$$

משפט 11.6 $3SAT$ היא NP שלמה.

$3SAT$ היא NP שלמה.

הוכחה:
יש לקיים את השני תנאים הבאים:

1) $3SAT\in NP$.

ניתן לבנות אלגוריתם אימות עבור $3SAT\in NP$ דומה לאלגוריתם האימות עבור SAT שבנינו בהוכחה של המשפט קוק-לוין 11.5 למעלה.

2) $3SAT$ היא NP קשה ע"י רדוקציה

$$SAT\leqslant_p 3SAT\ .$$

ואז בגלל ש- SAT היא NP שלמה (לפי משפט קוק-לוין 11.5) ומכיוון ש- $3SAT\in NP$ אז לפי משפט האסימפטוטית 11.2 גם $3SAT$ היא NP- שלמה.

קיום פונקצית הרדוקציה $SAT\leqslant_p 3SAT$

כעת נבנה את פונקציה הרדוקציה מ- SAT ל- $3SAT$.

ראשית נציין כי כל נוסחה בוליאנית ϕ ניתנת לרשום בצורה CNF בזמן פולינומיאלי.

בהינתן נוסחת CNF, ϕ (הקלט של SAT) נבנה בזמן פולינומיאלי נוסחת $3CNF$ ϕ' (הקלט של $3SAT$) ואז נוכיח שמתקיים

$$\langle\phi'\rangle\in 3SAT\quad\Leftrightarrow\quad\langle\phi\rangle\in SAT\ .$$

לכל פסוקית C ב- ϕ המכילה יותר מ- 3 ליטרלים, ניצור אוסף C' ב- ϕ' של פסוקיות כך שכל פסוקית ב- C' תכיל 3 ליטרלים. למשל בהינתן הפסוקית C הבאה של ϕ:

$$C=x_1\vee x_2\vee x_3\vee\bar{x}_4\vee\bar{x}_5$$

ניצור את הפסוקית C' הבאה ב- ϕ':

$$C'=(x_1\vee x_2\vee y_1)\wedge(\bar{y}_1\vee x_3\vee y_2)\wedge(\bar{y}_2\vee\bar{x}_4\vee\bar{x}_5)\ .$$

באופן כללי, לכל פסוקית $C=a_1\vee a_2\vee\ldots\vee a_k$ המכיל $k>3$ ליטרלים, ניצור אוסף C' של פסוקיות שבו כל פסוקית מכילה 3 ליטרלים, ע"י הוספת $k-3$ משתנים $y_1,y_2,\ldots,y_{k-3}$:

$$C'=(a_1\vee a_2\vee y_1)\wedge(\bar{y}_1\vee a_3\vee y_2)\wedge\ldots\wedge(\bar{y}_{i-2}\vee a_i\vee y_{i-1})\wedge\ldots\wedge(\bar{y}_{k-3}\vee a_{k-1}\vee a_k)\ .$$

בפרט, עבור כל פסוקית $C=(a_1\vee a_2\vee\ldots\vee a_k)$ נניח $a_i=1$ הוא הליטרל הראשון ששווה ל- 1. אז

- נשים $y_j=1$ לכל $1\leqslant j\leqslant i-2$,
- ונשים $y_j=0$ לכל $i-1\leqslant j\leqslant k-3$.

סיימנו להגדיר את הפונקציה הרדוקציה.
כעת נוכיח כי הפונקציה הזאת מקיימת את התנאי ההכרחי

$$\langle\phi'\rangle\in 3SAT\quad\Leftrightarrow\quad\langle\phi\rangle\in SAT\ .$$

כיוון $\Leftarrow$:

נניח כי $\langle\phi\rangle\in SAT$ ותהי X השמה המספקת את ϕ.
נוכיח שקיימת השמה X' מתאימה המספקת את ϕ'.

- בכל פסוקית C של ϕ, עבור הליטרלים $a_1, a_2, \ldots, a_k$ ניתן אותם ערכים כמו ב- X.
- מכיוון ש- X מספקת את ϕ, בכל פסוקית $C = (a_1 \vee a_2 \vee \ldots \vee a_k)$ יש לפחות ליטרל אחד שקיבל ערך 1. נניח $a_i = 1$. אז על פי ההגדרה של פונקצית הרודקציה:
  - נשים $y_j = 1$ לכל $1 \leqslant j \leqslant i-2$,
  - ונשים $y_j = 0$ לכל $i-1 \leqslant j \leqslant k-3$.

באופן הזה אנחנו ניצור אוסף C' של פבוקיות עם המבנה הבא:

$$\left(a_1 \vee a_2 \vee \overset{1}{y_1}\right) \wedge \left(\overset{0}{\bar{y}_1} \vee a_2 \vee \overset{1}{y_2}\right) \wedge \ldots \wedge \left(\overset{0}{\bar{y}_{i-3}} \vee a_{i-1} \vee \overset{1}{y_{i-2}}\right) \wedge \left(\overset{0}{\bar{y}_{i-2}} \vee \overset{1}{a_i} \vee \overset{0}{y_{i-1}}\right) \wedge \left(\overset{1}{\bar{y}_{i-1}} \vee a_{i+1} \vee \overset{0}{y_i}\right)$$
$$\wedge \ldots\ldots \wedge \left(\overset{1}{\bar{y}_{k-3}} \vee a_{k-1} \vee a_k\right)$$

ולכן השמה זו מספקת את C' ולכן $\langle \phi' \rangle \in 3SAT$.

<u>כיוון $\Rightarrow$:</u>

נניח כי $\langle \phi' \rangle \in 3SAT$ ותהי X' השמה המספקת את ϕ'.
נוכיח שקיימת השמה X המספקת את ϕ.

נסתכל על פסוקית $C = (a_1 \vee a_2 \vee \ldots \vee a_k)$.
נניח בשלילה שלא קיימת השמה X המספקת את C. אז בהכרח

$$a_1 = a_2 = \ldots = a_k = 0$$

לפי זה, באוסף פסוקיות C' שנקבל על פי ההגדרה של פונקצית הרדוקציה, $y_j = 1$ לכל $1 \leqslant j \leqslant k-3$.
כלומר מתקיים $y_1 = y_2 = \ldots = y_{k-3} = 0$. לכן

$$C' = \left(\overset{0}{a_1} \vee \overset{0}{a_2} \vee \overset{1}{y_1}\right) \wedge \left(\overset{0}{\bar{y}_1} \vee \overset{0}{a_3} \vee \overset{1}{y_2}\right) \wedge \ldots \wedge \left(\overset{0}{\bar{y}_{i-2}} \vee \overset{0}{a_i} \vee \overset{1}{y_{i-1}}\right) \wedge \ldots \wedge \left(\overset{0}{\bar{y}_{k-3}} \vee \overset{0}{a_{k-1}} \vee \overset{0}{a_k}\right)$$

הפסוקית האחרונה $\left(\overset{0}{\bar{y}_{k-3}} \vee \overset{0}{a_{k-1}} \vee \overset{0}{a_k}\right)$ אינה מסופקת.
לכן C' אינה מסופקת, בסתירה לכך ש- X' מספקת את C'.

ולכן $\langle \phi \rangle \in SAT$.

הוכחנו שקיימת הרדוקציה $SAT \leqslant 3SAT$.

כעת נוכיח כי הרדוקציה הזו היא זמן פולינומיאלית.

<u>סיבוכיות</u>

החישוב של הפונקציה מתבצע בזמן פולינומיאלי. ספציפי, אם האורך של הנוסחה ϕ הוא $n = |\phi|$ אז הרודקציה היא $O(n)$.

■

11.7 הוכחת משפט קוק לוין*

משפט 11.7 משפט קוק לוין

הבעיית SAT היא NP - שלמה.

הוכחה:
חשיפה מלאה: ההוכחה הבאה מתבססת על ההוכחה שנתונה בהספר של Sipser.

על פי הגדרה 11.2 יש להוכיח ששני התנאים הבאים מתקיימים:

תנאי 1: $SAT \in NP$.

תנאי 2: $SAT \leqslant_p A$ לכל $A \in NP$.

ראשית נוכיח כי $SAT \in NP$:

כדי להוכיח כי SAT שייכת ל- NP, נוכיח כי אישור המורכב מהשמה מספקת עבור נוסחת קלט ϕ ניתן לאימות בזמן פולינומיאלי.
נניח כי $n = |\phi|$. כלומר ב- ϕ מופיעים n ליטרלים.
ז"א השמה כלשהי דורשת n משתני בוליאניים לכל היותר.

- אלגוריתם האימות מחליף כל משתנה בנוסחה בערך המתאים לו על פי ההשמה.
  השלב הזה הוא $O(n)$.
- אחר כך האלגוריתם מחשב את ערכו של הביטוי:
  * נניח כי הנוסחה ϕ מכילה k דורות של סוגריים בתוך סוגריים.
  * החישוב מתחיל עם החישובים של הביטויים בתוך הסוגריים הכי בפנים.
  * יש n סוגריים הכי-בפנים לכל היותר, וכל אחד של הסוגריים האלה מכיל n ליטרלים לכל היותר.
    לכן החישוב הזה הוא $O(n^2)$.
  * יש k דורות של סוגריים לכן החישוב כולו הוא $O(kn^2)$
- בסה"כ הסיבוכיות זמן הריצה היא

$$O(n) + O\left(kn^2\right) = O\left(n^2\right)$$

  לפיכך אישור של השמה כלשהי מתבצע בזמן פולינומיאלי.

- אם ערכו של הביטוי הוא 1 הנוסחה ספיקה.

הוכחנו כי $SAT \in NP$. עכשיו נוכיח כי $A \leqslant_p SAT$.

תהי N מ"ט אי-דטרמיניסטית זמן-פולינומיאלית שמכריעה שפה A כלשהי בזמן $O\left(n^k\right)$ עבור k טבעי.
התרשים למטה מראה טבלה של קונפיגורציות של N.
ברשימה הבאה רשומות ההגדרות של הטבלה:

- כל שורה מראה את תוכן הסרט בשלב מסוים של מסלול אחד של N.
- בשורה הראשונה יש את הקונפיגורציה ההתחלתית.
- אנחנו מניחים כי האורך של המילה, כלומר אורך הקלט הוא n.
  הסימנים $w_1, \ldots, w_n$ מסמנים את התווים של הקלט.

- בתא הראשון בכל שורה יש #, ואחר כך רשומה הקונפיגורציה של N.
  בסוף הקונפיגורציה בכל שורה יש #.
  אחרי ה- # בקצה הימין של המילה, בכל תא יש תו רווח עד הסוף של השורה.
  התווי רווח לפני המשבצת הראשונה של הקונפיגורציה לא מופיעים בטבלה.
- האורך של כל שורה הוא בדיוק n^k תאים.
- בטבלה יש בדיוק n^k שורות לסיבה הבאה:
  - המכונת טיורינג מבצעת n^k צעדים לכל היותר.
  - בכל צעד המ"ט עוברת לקונפיגורציה חדשה.
  - בכל שורה רשומה קונפיגורציה.
  - בסה"כ יש n^k שורות עבור ה- n^k קונפיגוריות שונות האפשריות.

#	q_0	w_1	w_2	$\dots$	w_n	$\sqcup$	$\dots$	$\sqcup$	#
#	q_0								#
#	q_0								#
#									#

אנחנו אומרים כי טבלה שלהי היא **טבלה המקבלת** אם באחת השורות יש קונפיגורציה אשר N מקבלת אותה.

בעזרת הטבלה נתאר את הרדוקציה זמן-פולינומיאלית f משפה A כלשהי ל- SAT.

הפונקצית הרדוקציה
f מקבלת קלט w ומחזירה נוסחה $\phi = f(w)$, אשר לפי ההגדרה של פונקצית הרדוקציה, עומדת בתנאי הבא:

$$w \in A \qquad \Leftrightarrow \qquad f(w) \in SAT \ .$$

יהיו Q קבוצת המצבים ו- Γ האלפיבית של הסרט של N. נגדיר

$$C = Q \cup \Gamma \cup \{\#\} \ .$$

נסמן ב- s איבר כלשהו של C.
עבור כל תא ה- (i,j) של הטבלת הקונפיגורציות נגדיר משתנה בוליאני $x_{i,j,s}$ לכל $1 \leqslant i,j \leqslant n^k$.
המשתנה x_{ijs} מוגדר על פי התנאי

$$x_{ijs} = 1$$

אם בתא ה- ij של הטבלה יש $s \in C$. למשל, אם בתא ה- $(2,5)$ של הטבלה מופיע התו a אז

$$x_{2,5,\text{a}} = 1$$

בעוד

$$x_{2,5,\mathrm{b}} = 0 \ .$$

במובן הזה, התכנים של כל התאים של הטבלה מסומנים על ידי המשתנים של ϕ.

עכשיו נבנה נוסחה ϕ על סמך התנאי שהשמה מספקת של ϕ תהיה מתאימה לטבלה המקבלת של N.
נגדיר

$$\phi = \phi_{\text{cell}} \wedge \phi_{\text{start}} \wedge \phi_{\text{move}} \wedge \phi_{\text{acc}} \ . \tag{11.1}$$

אנחנו נסביר את כל הנוסחאות ϕ_{cell}, ϕ_{start}, ϕ_{move} ו- ϕ_{acc} אחד אחד למטה.

- הנוסחה ϕ_{cell}

כפי שמצויין לעיל, אם המשתנה $x_{i,j,s}$ "דולק", כלומר אם $x_{i,j,s}=1$, זאת אומרת שיש סימן s בתא ה- ij של הטבלה. אנחנו רוצים להבטיח שהשמה כלשהי בנוסחה אשר מתאימה לקונפיגורציה של הטבלה, מדליקה בדיוק משתנה אחד לכל תא של הטבלה. למטרה זו נגדיר ϕ_{cell} כך:

$$\phi_{\text{cell}} = \bigwedge_{1\leqslant i,j\leqslant n^k} \left[\left(\bigvee_{s\in C} x_{i,j,s} \right) \wedge \left(\bigwedge_{\substack{s,t\in C\\ s\neq t}} \left(\overline{x}_{i,j,s} \vee \overline{x}_{i,j,t} \right) \right) \right] \tag{11.2}$$

* האיבר הראשון בסוגריים מרובעים, $\bigvee\limits_{s\in C} x_{i,j,s}$ מבטיח שלכל תא של הטבלה, לפחות משתנה אחד דולק.
* האיבר השני $\bigwedge\limits_{\substack{s,t\in C\\ s\neq t}} \left(\overline{x}_{i,j,s} \vee \overline{x}_{i,j,t} \right)$ מבטיח שעבור כל תא של הטבלה, משתנה אחד לכל היותר דולק.

לפיכך כל השמה מספקת עומדת בתנאי שיהיה בדיוק סימן אחד, s, בכל תא של הטבלה.

- הנוסחה ϕ_{start}

נוסחה ϕ_{start} מבטיחה ששורה הראשונה של הטבלה היא הקונפיגורציה ההתחלתית של N על הקלט w:

$$\begin{aligned}\phi_{\text{start}} =& x_{1,1,\#} \wedge x_{1,2,q_0} \wedge x_{1,3,w_1} \wedge x_{1,4,w_2} \\ & \wedge \ldots \wedge \\ & x_{1,n+2,w_n} \wedge x_{1,n+3,\sqcup} \wedge \ldots \wedge x_{1,n^k-1,\sqcup} \wedge x_{1,n^k,\#} \end{aligned} \tag{11.3}$$

- הנוסחה ϕ_{acc}

הנוסחה ϕ_{acc} מבטיחה שקיימת טבלה קונפיגורציה אשר המ"ט N מקבלת אותה.

בפרט ϕ_{acc} מבטיחה שהסימן q_{acc} מופיע בתא אחד של הטבלה דרך התנאי שלפחות אחד המשתנים $x_{i,j,q_{\text{acc}}}$ דולק:

$$\phi_{\text{acc}} = \bigvee_{1\leqslant i,j\leqslant n^k} x_{i,j,q_{\text{acc}}} \tag{11.4}$$

• <u>הנוסחה ϕ_{move}</u>

הנוסחה ϕ_{move} מבטיחה שכל שורה של הטבלה היא "שורה חוקית".

כלומר בכל שורה, הקונפיגורציה היא כך שאפשר להגיע אליה על ידי תזוזה חוקית של N מהקונפיגורציה הקודמת שמופיעה בשורה אחת למעלה.

תזוזה חוקית בין כל שתי קונפיגורציות נקבעת על ידי הפונקצית המעברים של המ"ט N.

בשפה פורמלית, אם c_i הקונפיגורציה של שורה i, ו- c_{i+1} הקונפיגורציה של השורה $i+1$ אחת למטה, אז ϕ_{move} מבטיחה כי לכל $1 \leqslant i \leqslant n^k - 1$ מתקיים

$$c_i \vdash_N c_{i+1} \ .$$

במונחי הטבלה, אפשר להגדיר תזוזה חוקית בין כל שתי שורות על ידי תת-טבלה מסדר 2×3 שמכילה 3 תאים מתאימים של שתי שורות שכנות.

מכאן ואילך אנחנו נקרא לתת-טבלה כזאת "חלון".

למטה יש דוגמאות של חלונות חוקיים:

a	a	q_1
a	a	b

a	q_1	b
a	a	q_2

a	q_1	b
q_2	a	c

b	b	b
c	b	b

a	b	a
a	b	q_2

#	b	a
#	b	a

החלונות האלה למטה הם דוגמאות לחלונות לא חוקיים:

b	q_1	b
q_2	b	q_2

a	q_1	b
q_1	a	a

a	b	a
a	a	a

הנוסחה ϕ_{move} קובעת שכל חלון של הטבלה חוקי. בפרט, כל חלון מכיל 6 תאים. לכן ϕ_{move} קובעת שהתכנים של ה-6 תאים של כל חלון מהווה חלון חוקי. ז"א

$$\phi_{\text{move}} = \bigwedge_{\substack{1\leqslant i\leqslant n^k \\ 1\leqslant j\leqslant n^k}} \left(\text{חלון ה- } i,j \text{ חוקי}\right) \tag{11.5}$$

אנחנו מציבים בטקסט " חלון ה- i,j חוקי " את הנוסחה הבאה, כאשר $a_1,\ldots,a_6$ מסמנים את התכנים של ה-6 תאים של כל חלון:

$$\bigvee_{\substack{\{a_1,a_2,a_3,a_4,a_5,a_6\} \\ \text{חלון חוקי}}} \left(x_{i,j-1,a_1} \wedge x_{i,j,a_2} \wedge x_{i,j+1,a_3} \wedge x_{i+1,j-1,a_4} \wedge x_{i+1,j,a_5} \wedge x_{i+1,j+1,a_6}\right) \tag{11.6}$$

עד כה הוכחנו שקיים רדוקציה מכל שפה $A \in NP$ ל- SAT.
כעת נוכיח כי הרדוקציה זו חישובית בזמן-פולינומיאלי.

הטבלה של N היא מסדר $n^k \times n^k$ ולכן היא מכילה n^{2k} תאים.

נחשב את הסיבוכיות של כל הנוסחאות ϕ_{cell}, ϕ_{start}, ϕ_{acc}, ϕ_{move}.

- <u>הנוסחה ϕ_{cell}</u>

הנוסחה (11.2) של ϕ_{cell} מכילה n^{2k} נוסחאות עם 3 ליטרלים. לכן

$$\phi_{\text{cell}} = O\left(n^{2k}\right) \ .$$

- <u>הנוסחה ϕ_{start}</u>

הנוסחה (11.3) של ϕ_{start} מכילה בדיוק n^k ליטרלים. לכן

$$\phi_{\text{start}} = O\left(n^{k}\right) \ .$$

- <u>הנוסחה ϕ_{acc}</u>

הנוסחה (11.4) של ϕ_{acc} מכילה בדיוק n^k ליטרלים. לכן

$$\phi_{\text{acc}} = O\left(n^{k}\right) \ .$$

- <u>הנוסחה ϕ_{move}</u>

הנוסחה (11.6,11.5) של ϕ_{move} מכילה n^{2k} נוסחאות עם 6 ליטרלים. לכן

$$\phi_{\text{move}} = O\left(n^{2k}\right) \ .$$

לכן בסה"כ

$$\phi = O\left(n^{2k}\right) + O\left(n^{k}\right) + O\left(n^{k}\right) + O\left(n^{2k}\right) = O\left(n^{2k}\right) \ .$$

לפיכך קיימת רדוקציה חישובית בזמן פולינומיאלי מכל שפה $A \in NP$ ל- SAT.

■

שיעור 12
רדוקציות פולינומיאליות

12.1 $CLIQUE$ היא NP- שלמה

משפט 12.1 $CLIQUE \in NPC$

הבעיית $CLIQUE$ היא (ראו הגדרה 10.5):

$$CLIQUE = \{\langle G,k\rangle \mid G \text{ מכיל קליקה בגודל } k\} \ .$$

$CLIQUE$ היא NP- שלמה

הוכחה:

1) הוכחנו כי $CLIQUE \in NP$ במשפט 10.2.

2) נוכיח כי $CLIQUE$ היא NP קשה ע"י רדוקציה $3SAT \leqslant_P CLIQUE$.

<u>פונקצית הרדוקציה</u>

בהינתן נוסחת $3CNF$ ϕ מעל n משתנים $x_1,x_2,\ldots,x_n$ המכיל m פסוקיות, ניצור זוג $\langle G,k\rangle$ ונוכיח כי

$$\langle\phi\rangle \in 3SAT \quad \Leftrightarrow \quad \langle G,k\rangle \in CLIQUE \ .$$

נבנה את הגרף G באופן הבא:

<u>הקדקודים של G:</u>

לכל פסוקית C_i ב- ϕ המכילה 3 ליטרלים ניצור שלשה t_i המכילה 3 קודקודים המתאימים ללחטרלים של C_i:

$$x_1 \vee \bar{x}_3 \vee x_4 \quad \longrightarrow \quad x_1 \qquad \bar{x}_3 \qquad x_4$$

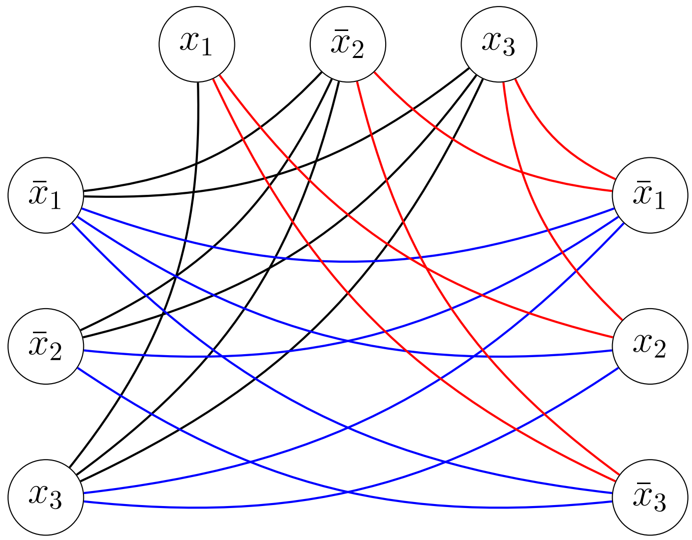
<u>הצלעות של G:</u>

נבחר בין כל שני קודקודים פרט לזוגות הבאים:

- זוג קודקודים המתאימים למשתנה ומשלים שלו.
- זוג קודקודים שנמצאים באותה שלושה.

לדוגמה:

$$\phi = \underset{C_1}{\left(\overset{T}{x_1} \vee \overset{T}{\bar{x}_2} \vee x_3\right)} \wedge \underset{C_2}{\left(\bar{x}_1 \vee \overset{T}{\bar{x}_2} \vee x_3\right)} \wedge \underset{C_3}{\left(\bar{x}_1 \vee x_2 \vee \overset{T}{\bar{x}_3}\right)}$$



נקבע $k = m$.

נכונות הרדוקציה

1) ניתן לבנות את G בזמן פולינומיאלי בגודל ϕ.

2) נוכיח כי

$$\langle \phi \rangle \in 3SAT \quad \Leftrightarrow \quad \langle G, k \rangle \in CLIQUE\ .$$

כיוון $\Leftarrow$

- נניח כי ϕ ספיקה ונסתכל על השמה המספקת את ϕ.
- בכל פסוקית C_i ב- ϕ יש לפחות ליטרל אחד שקיבל ערך T.
- נבחר מכל שלשה t_i בקודקוד אחד המתאים לליטרל שקיבל ערך T ב- C_i ונוסיף אותו לקליקה.
- מספר הקודקודים שבחרנו שווה k וכל שניים מהם מחוברים בצלע כי לא בחרנו שני קודקודים מאותה שלשה ולא בחרנו שני קודקודים המתאימים למשתנה ומשלים שלו.
- ולכן G מכיל קליקה בגודל k.

כיוון $\Rightarrow$

- נניח כי G מכיל קליקה בגודל k ונסתכל על קליקה כזו.
- לפי הבנייה הקליקה מכילה בדיוק קודקוד אחד מכל שלשה t_i. ניתן השמה למשתנים של ϕ כך שהליטרל שמתאים לקודקוד שנמצא בקליקה יקבל ערך T.
- השמה זו אפשרית מכיוון שהקליקה לא מכילה שני קודקודים המתאימים למשתנה ומשלים שלו.

- בנוסף השמ זו מספקת את ϕ מכיוון שהקליקה מכילה קודקוד מכל שלשה t_i ולכן הליטרל המתאים לקודקוד בשלשה t_i קיבל ערך T ולכן הוא מספק את הפסוקית C_i.
- לכן ϕ ספיקה.

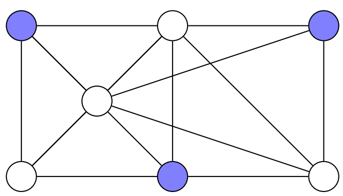
■

12.2 בעיית הקבוצה הבלתי תלויה

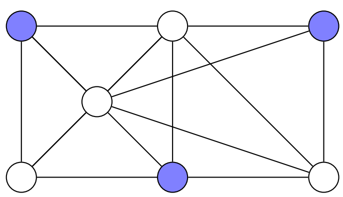
הגדרה 12.1 קבוצה בלתי תלויה

בהינתן גרף לא מכוון $G=(V,E)$, קבוצה בלתי תלויה ב- G היא תת-קבוצה של קודקודים $S\subseteq V$ כך שלכל שני קודקודים $u,\mathrm{v}\in S$ מתקיים $(u,\mathrm{v})\notin E$.

קבוצה בלתי תלוייה בגודל $k=3$:



קבוצה בלתי תלוייה בגודל $k=3$:



הגדרה 12.2 בעיית IS

קלט: גרף לא מכוון $G=(V,E)$ ומספר k.
פלט: האם קיימת קבוצה בלתי תלויה ב- G בגודל k?

$$IS=\left\{\langle G,k\rangle \;\middle|\; \text{גרף לא מכוון המכיל קבוצה בלתי תלויה בגודל } k \text{ } G\right\}$$

משפט 12.2 $IS\in NPC$

הבעייה IS היא NP - שלמה.

הוכחה:

(1) נוכיח כי $IS\in NP$

נבנה אלגוריתם אימות V עבור IS.
$V=$ על קלט $(\langle G,k\rangle,y)$:

- בודק האם y היא קבוצה של k קודקודים מ- G השונים זה מזה.
  - אם לא $\Rightarrow$ דוחה.
- בודק האם כל שני קודקודים מ- y לא מחוברים בצלע ב- G.
  - אם כן $\Rightarrow$ מקבל.

- אם לא $\Rightarrow$ דוחה.

(2) נוכיח כי $CLIQUE \leqslant_P IS$

פונקצית הרדוקציה:

בהינתן זוג $\langle G,k\rangle$ הקלט של $CLIQUE$, ניצור זוג $\langle G',k'\rangle$, הקלט של IS, ונוכיח כי:

$$\langle G,k\rangle \in CLIQUE \quad \Leftrightarrow \quad \langle G',k'\rangle \in IS \ .$$

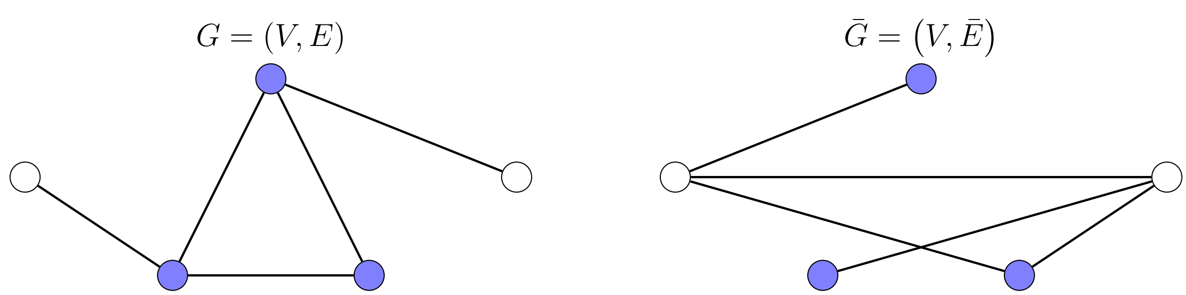
הפונקציית הרדוקציה מוגדרת כך שהתנאים הבאים מתקיימים:

1) נניח שהגרף הוא $G=(V,E)$.
אז הגרף G' הוא הגרף המשלים של $G=(V,E)$.
ז"א $G'=\bar{G}=\left(V,\bar{E}\right)$ כאשר

$$\bar{E}=\left\{(u_1,u_2)\;\middle|\;(u_1,u_2)\notin E\right\} \ .$$

2) $k'=k$.

לדוגמה, בהינתן הגרף $G=(V,E)$ שמכיל קליקה בגודל $k=3$, הפונקציית הרדוקציה R מחזירה את הגרף $\bar{G}=(V,\bar{E})$ ואת המספר $k'=k=3$, כמתואר בתרשים למטה:



נכונות הרדוקציה

1) ניתן לבנות G' בזמן פולינומיאלי בגודל G.

2) נוכיח כי . $\langle G,k\rangle \in CLIQUE \quad \Leftrightarrow \quad \langle G',k'\rangle \in IS$

כיוון $\Leftarrow$

בהינתן גרף $G=(V,E)$ ושלם k.
נניח כי $\langle G,k\rangle \in CLIQUE$.

$\Leftarrow$ G מכיל קליקה S בגודל k.

$\Leftarrow$ אם $u_1,u_2\in S$ (אם u_1,u_2 שני קודקודים בקליקה S) אזי $(u_1,u_2)\in E$.
כלומר, כל שני קדקודים ב- S מחוברים בצלע של G.

$\Leftarrow$ אם $u_1,u_2\in S$ אזי $(u_1,u_2)\notin\bar{E}$.
כלומר, כל שני קדקודים ב- S לא מחוברים בצלע של המשלים של הגרף $\bar{G}$, דהיינו G'.

$\Leftarrow$ אותה הקבוצה S היא קבוצה בלתי תלוייה ב- G' בגודל $k' = k$.

$\Leftarrow$ $G' = \bar{G}(V, \bar{E})$ מכיל קבוצה בלתי תלויה בגודל k'.

$\Leftarrow$ $\langle G', k \rangle \in IS$.

<u>כיוון $\Rightarrow$</u>

בהינתן גרף G' ושלם k'.

נניח כי $\langle G', k' \rangle \in IS$.

$\Leftarrow$ G' מכיל קבוצה בלתי תלוייה S בגודל k'.

$\Leftarrow$ אם $u_1, u_2 \in S$ אזי $(u_1, u_2) \notin \bar{E}$.

כלומר, כל שני קדקודים ב- S לא מחוברים בצלע של G'.

$\Leftarrow$ אם $u_1, u_2 \in S$ אזי $(u_1, u_2) \in E$.

כלומר, כל שני קדקודים ב- S מחוברים בצלע של $G(V, E)$.

$\Leftarrow$ אותה הקבוצה S היא קליקה ב- G בגודל $k = k'$.

$\Leftarrow$ G מכיל קליקה בגודל k.

$\Leftarrow$ $\langle G, k \rangle \in CLIQUE$.

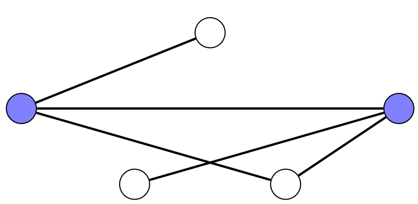
■

12.3 בעיית הכיסוי בקודקודים

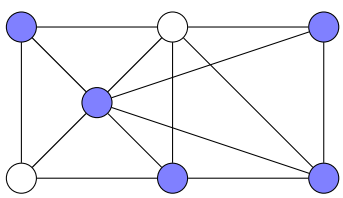
הגדרה 12.3 כיסוי בקודקודים

בהינתן גרף לא מכוון $G = (V, E)$, כיסוי בקודקודים ב- G הוא תת-קבוצה של קודקודים $C \subseteq V$ כך שלכל צלע $u, v \in S$ מתקיים $u \in C$ או $v \in C$.

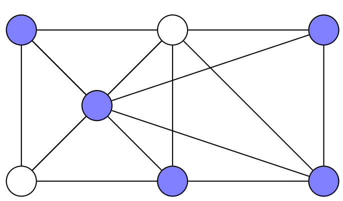
כיסוי בקודקודים בגודל $k = 2$:



כיסוי בקודקודים בגודל $k = 5$:



כיסוי בקודקודים בגודל $k = 5$:



12.4 הבעייה VC

הגדרה 12.4 בעיית VC

קלט: גרף לא מכוון $G=(V,E)$ ומספר k.
פלט: האם קיים כיסוי בקודקודים ב- G בגודל k?

$$VC = \left\{\langle G,k\rangle \;\middle|\; \text{גרף לא מכוון המכיל כיסוי בקודקודים בגודל } k \;\; G \right\}$$

משפט 12.3 $VC \in NPC$

הבעייה VC היא NP - שלמה.

הוכחה:
נוכיח כי $VC \in NP$

נבנה אלגוריתם אימות V עבור VC.
$V=$ על קלט $(\langle G,k\rangle, y)$:

- בודק האם כל צלע ב- G מכילה לפחות קצה אחד ב- y.
  - אם כן $\Rightarrow$ מקבל.
  - אם לא $\Rightarrow$ דוחה.

נוכיח כי VC היא NP- קשה ע"י רדוקציה $IS \leqslant_P VC$

פונקצית הרדוקציה:

בהינתן זוג $\langle G,k\rangle$ הקלט של IS, ניצור זוג $\langle G',k'\rangle$ הקלט של VC ונוכיח כי

$$\langle G,k\rangle \in IS \quad \Leftrightarrow \quad \langle G',k'\rangle \in VC \ .$$

הפונקציית הרדוקציה מוגדרת כך שהתנאים הבאים מתקיימים:

1) נניח שהגרף הוא $G=(V,E)$.
אז הגרף G' הוא אותו גרף $G=(V,E)$.

2) $k'=|V|-k$.

נכונות הרדוקציה

1) ניתן לבנות G' בזמן פולינומיאלי בגודל G.

2) נוכיח כי $\langle G',k'\rangle \in VC \quad \Leftrightarrow \quad \langle G,k\rangle \in IS$.

כיוון $\Leftarrow$

בהינתן גרף $G=(V,E)$ ושלם k.
נניח כי $\langle G,k\rangle \in IS$.

$\Leftarrow$ G מכיל קבוצה בלתי תלוייה S בגודל k.

$\Leftarrow$ אם $u_1 \in S$ וגם $u_2 \in S$ אז $(u_1,u_2) \notin E$.
כלומר, כל שני קדקודים ב- S לא מחוברים בצלע ב- G.

$\Leftarrow$ השלילה הלוגית של הגרירה הזאת היא:
אם $(u_1,u_2) \in E$ אז $u_1 \notin S$ או $u_2 \notin S$.

$\Leftarrow$ אם $(u_1,u_2) \in E$ אז $u_1 \in V \backslash S$ או $u_2 \in V \backslash S$.

$\Leftarrow$ $V \backslash S$ היא כיסוי קדקודים ב- G בגודל $k' = |V| - k$.

$\Leftarrow$ הגרף $G' = G$ מכיל כיסוי קדקודים בגודל k'.

$\Leftarrow$ $\langle G', k' \rangle \in VC$.

<u>כיוון $\Rightarrow$</u>

בהינתן גרף G' ושלם k'.
נניח כי $\langle G', k' \rangle \in VC$.

$\Leftarrow$ G' מכיל כיסוי בקדקודים C בגודל k'.

$\Leftarrow$ אם $(u_1,u_2) \in E$ אז $u_1 \in C$ או $u_2 \in C$.

$\Leftarrow$ השלילה הלוגית של הגרירה הזאת היא:
אם $u_1 \notin C$ וגם $u_2 \notin C$ אז $(u_1,u_2) \notin E$.

$\Leftarrow$ אם $u_1 \in V \backslash C$ וגם $u_2 \in V \backslash C$ אז $(u_1,u_2) \notin E$.

$\Leftarrow$ כל שני קדקודים ב- $V \backslash C$ לא מחוברים בצלע ב- G'.

$\Leftarrow$ $V \backslash C$ החא קבוצת בלתי תלוייה ב- G' בגודל $k = |V| - k'$.

$\Leftarrow$ הגרף $G = G'$ מכיל קבוצה בלתי תלוייה בגודל k.

■

12.5 $PARTITION$

הגדרה 12.5 בעיית $PARTITION$

<u>קלט:</u> קבוצת מספרים שלמים $S = \{x_1, x_2, \ldots, x_n\}$.
<u>פלט:</u> האם קיימת תת-קבוצה $Y \subseteq S$ כך ש- $\sum\limits_{y \in S \backslash Y} y = \sum\limits_{y \in Y} y$?

$$PARTITION = \left\{ S \;\middle|\; \sum_{y \in Y} y = \sum_{y \in S \backslash Y} y \text{ כך ש- } Y \subseteq S \text{ קבוצת שלמים, וקיימת תת-קבוצה } S \right\}$$

12.6 רדוקציות פולינומיאליות

משפט 12.4 רדוקציות פולינומיאליות

$$\begin{aligned} SAT &\leqslant_P 3SAT \\ 3SAT &\leqslant_P CLIQUE \\ CLIQUE &\leqslant_P IS \\ IS &\leqslant_P VC \\ SubSetSum &\leqslant_P PARTITION \\ HAMPATH &\leqslant_P HAMCYCLE \end{aligned}$$

12.7 שפות NP שלמות

משפט 12.5 שפות NP- שלמות

SAT	NP- שלמה.	(משפט קוק לוין)
$3SAT$	NP- שלמה.	
$HAMPATH$	NP- שלמה.	
$CLIQUE$	NP- שלמה.	
IS	NP- שלמה.	
VC	NP- שלמה.	

שיעור 13
סיבוכיות מקום ושלמות ב PSPACE

13.1 משפט סביץ'
13.2 המחלקה PSPACE
13.3 שלמות ב- PSPACE
13.4 המחלקה L
13.5 המחלקה NL
13.6 שלמות ב- NL
13.7 שיוויון NL ו- coNL