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 "␣".

⋯	␣	␣	a	b	b	b	a	a	␣	␣	␣	⋯

הראש

- במצב ההתחלתי הראש בקצה השמאלי של הקלט.

⋯	␣	␣	a	b	b	b	a	a	␣	␣	␣	⋯
			↑									

- 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	⋯	␣	␣	a	b	b	b	a	a	␣	␣	␣	⋯
				↑									

- 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. 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 ✓, ועוברים לשלב 2).
- אם האות הראשונה שהראש מצא היא b, כותבים עליו ✓, ועוברים לשלב 3).

2) ממשיכים לזוז ימינה עד שנמצא b תואם.
- אם לא מצאנו b $\Leftarrow$ דוחה.
- אם מצאנו b כותבים עליו ✓, חוזרים לתחילת הקלט וחוזרים לשלב 1).

3) ממשיכים לזוז ימינה עד שנמצא a תואם.
- אם לא מצאנו a $\Leftarrow$ דוחה.
- אם מצאנו a כותבים עליו ✓, 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:

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

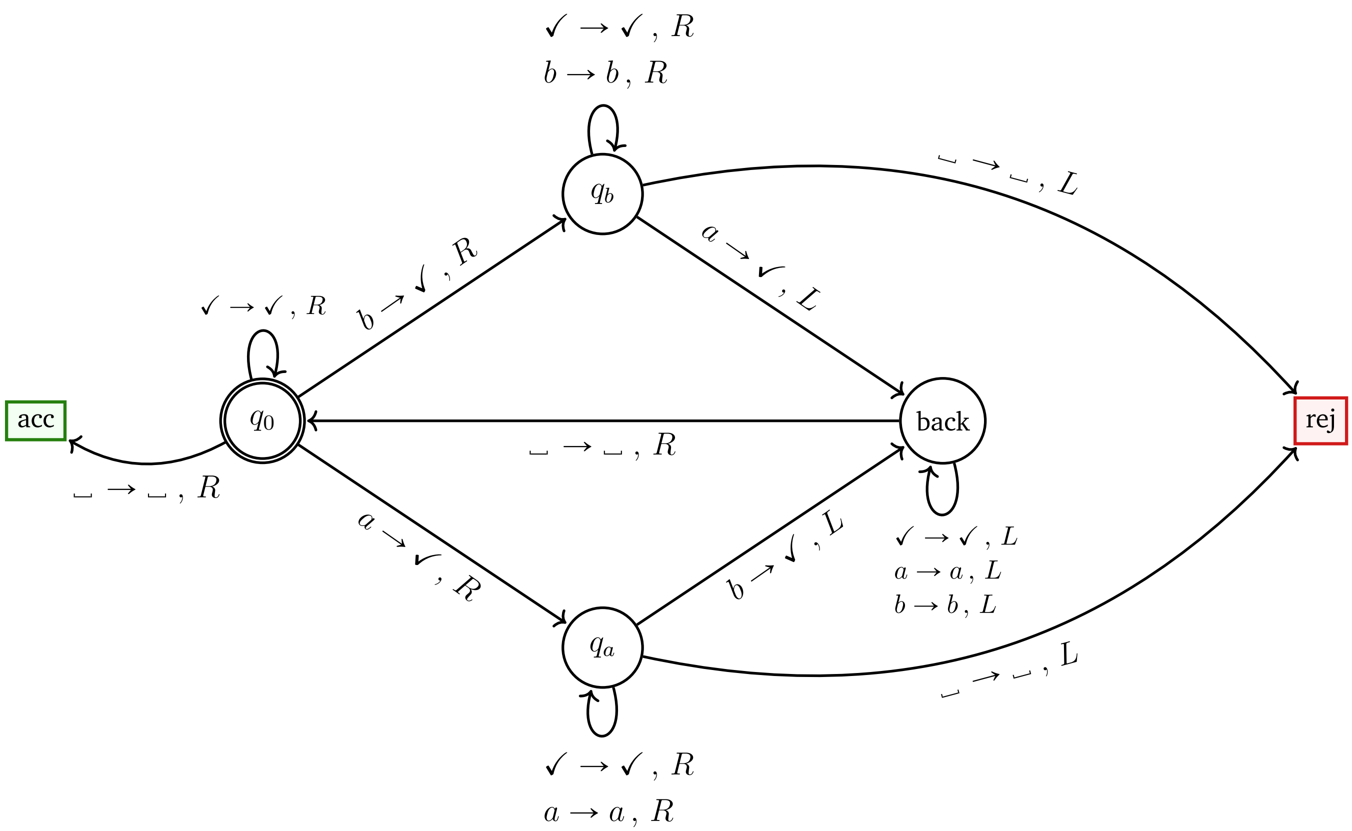
הפונקצית המעברים $\delta : Q \times \Sigma \to Q \times \Gamma \times \{L,R\}$ היא מוגדרת כדלקמן.

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

לעתים קל יותר לרשום את פונקציית המעבירים δ כטבלה:

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

<u>תרשים מצבים</u>



דוגמה 1.2

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	$\sqcup$
$\sqcup$	$\checkmark$	q_{back}	a	$\checkmark$	$\sqcup$
$\sqcup$	q_{back}	$\checkmark$	a	$\checkmark$	$\sqcup$
q_{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	$\checkmark$	$\sqcup$

דוגמה 1.3

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	$\sqcup$
$\sqcup$	q_{back}	$\checkmark$	$\checkmark$	b	b	a	a	$\sqcup$
q_{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	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	q_{back}	b	$\checkmark$	a	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	q_{back}	$\checkmark$	b	$\checkmark$	a	$\sqcup$
$\sqcup$	$\checkmark$	q_{back}	$\checkmark$	$\checkmark$	b	$\checkmark$	a	$\sqcup$
$\sqcup$	q_{back}	$\checkmark$	$\checkmark$	$\checkmark$	b	$\checkmark$	a	$\sqcup$
q_{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	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	q_{back}	$\checkmark$	$\checkmark$	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	q_{back}	$\checkmark$	$\checkmark$	$\checkmark$	$\sqcup$

$\sqcup$	$\checkmark$	$\checkmark$	q_{back}	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\sqcup$
$\sqcup$	$\checkmark$	q_{back}	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\sqcup$
$\sqcup$	q_{back}	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\sqcup$
q_{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	$\sqcup$
$\sqcup$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\sqcup$	q_{acc}

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 1.2)

u	q	σ	v
$\sqcup$	q_0	a	a b $\sqcup$
$\sqcup\checkmark$	q_a	a	b $\sqcup$
$\sqcup$ $\checkmark$ a	q_a	b	$\sqcup$
$\sqcup$ $\checkmark$	q_{back}	a	$\checkmark$ $\sqcup$
$\sqcup$	q_{back}	$\checkmark$	a $\checkmark$ $\sqcup$
$\sqcup$	q_{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	$\sqcup$	$\sqcup$
$\sqcup$ $\checkmark$ $\checkmark$	q_{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 $n = 2^k \quad (k \geqslant 0)$ אם ורק אם קיים שלם m עבורו חילוק של n ב- 2 בדיוק m פעמים נותן 1.

הוכחה:
<u>כיוון $\Leftarrow$</u>

אם $n = 2^k \ (k \geqslant 0)$ 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 שוב ושוב בצורה איטרטיבית.

- אם אחרי סיבוב מסויים נקבל מספר אי-זוגי שונה מ- 1, אז אין מצב שמספר האותיות a הוא חזקה של 2.
- בצד שני אם אחרי סיבוב כלשהו נקבל בדיוק a אחת הנשארת, ז"א אחרי מספר מסוים של חילוקים של המספר אותיות a קיבלנו 1, אזי מובטח לנו שהמספר של אותיות a הוא שווה לחזקה של 2.

כעת נסביר כיצד המכונה טיורינג מבצעת את השיטה הזאת בפועל כדלקמן.

1) במצב ההתחלתי יש מחרוזת של רצף אותיות a כתובה על הסרט והראש נמצא מתחת האות הראשונה.

⊔	⊔	a	a	a	a	a	a	a	a	⊔	⊔
		↑									

2) עוברים על הקלט משמאל לימין ומבצעים מחיקה לסירוגין של האות a. כלומר, אות אחת נמחק ואות אחת נשאיר וכן הלאה, עד שמגיעים לקצה הימין של המילה.

⊔	⊔	✓	a	✓	a	✓	a	✓	a	⊔	⊔
									↑		

3) אחרי שהראש הגיע לסוף המילה:

- אם מצאנו אות a אחת בדיוק $\Leftarrow$ המכונה תקבל.
- אם כתוב ✓ בתו האחרון $\Leftarrow$ המכונה תדחה.
- אחרת, אם כתוב a בתו האחרון הראש חוזר לתחילת המחרוזת וחוזרים לשלב 2).

כדוגמה של מילה המתקבלת על ידי האלגוריתם, למטה רשומות האיטרציות של האלגוריתם הזה על המילה $w = aaaaaaaa$ (8 אותיות a). במצב ההתחלתי הסרט נראה כדלקמן.

⊔	⊔	a	a	a	a	a	a	a	a	⊔	⊔
		↑									

איטרציה 1) לבסוף האיטרציה $i = 1$ הסרט נראה כך:

⊔	⊔	✓	a	✓	a	✓	a	✓	a	⊔	⊔
									↑		

התו האחרון a אז ממשיכים לאיטרציה הבאה.

איטרציה 2 בסוף האיטרציה $i=2$ הסרט נראה כך:

␣	␣	✓	✓	✓	a	✓	✓	✓	a	␣	␣
									↑		

התו הראשון הוא a אז ממשיכים לאיטרציה הבאה.

איטרציה 3 לאחר האיטרציה $i=3$ הסרט נראה כך:

␣	␣	✓	✓	✓	✓	✓	✓	✓	a	␣	␣
									↑		

התו האחרון הוא a אז ממשיכים לאיטרציה הבאה.

איטרציה 4 באיטרציה $i=4$ יש אות a אחת בדיוק אז המכונה מקבלת.

כדוגמה של מילה הלא המתקבלת על ידי האלגוריתם, למטה רשומות האיטרציות של האלגוריתם הזה על המילה $w=aaaaaa$ (6 אותיות a).
במצב ההתחלתי הסרט נראה כדלקמן.

␣	␣	a	a	a	a	a	a	␣	␣
		↑							

איטרציה 1 לבסוף האיטרציה $i=1$ הסרט נראה כך:

␣	␣	✓	a	✓	a	✓	a	␣	␣
							↑		

התו האחרון a אז ממשיכים לאיטרציה הבאה.

איטרציה 2 בסוף האיטרציה $i=2$ הסרט נראה כך:

␣	␣	✓	✓	✓	a	✓	✓	␣	␣
							↑		

התו הראשון הוא $\checkmark$ אז דוחה.

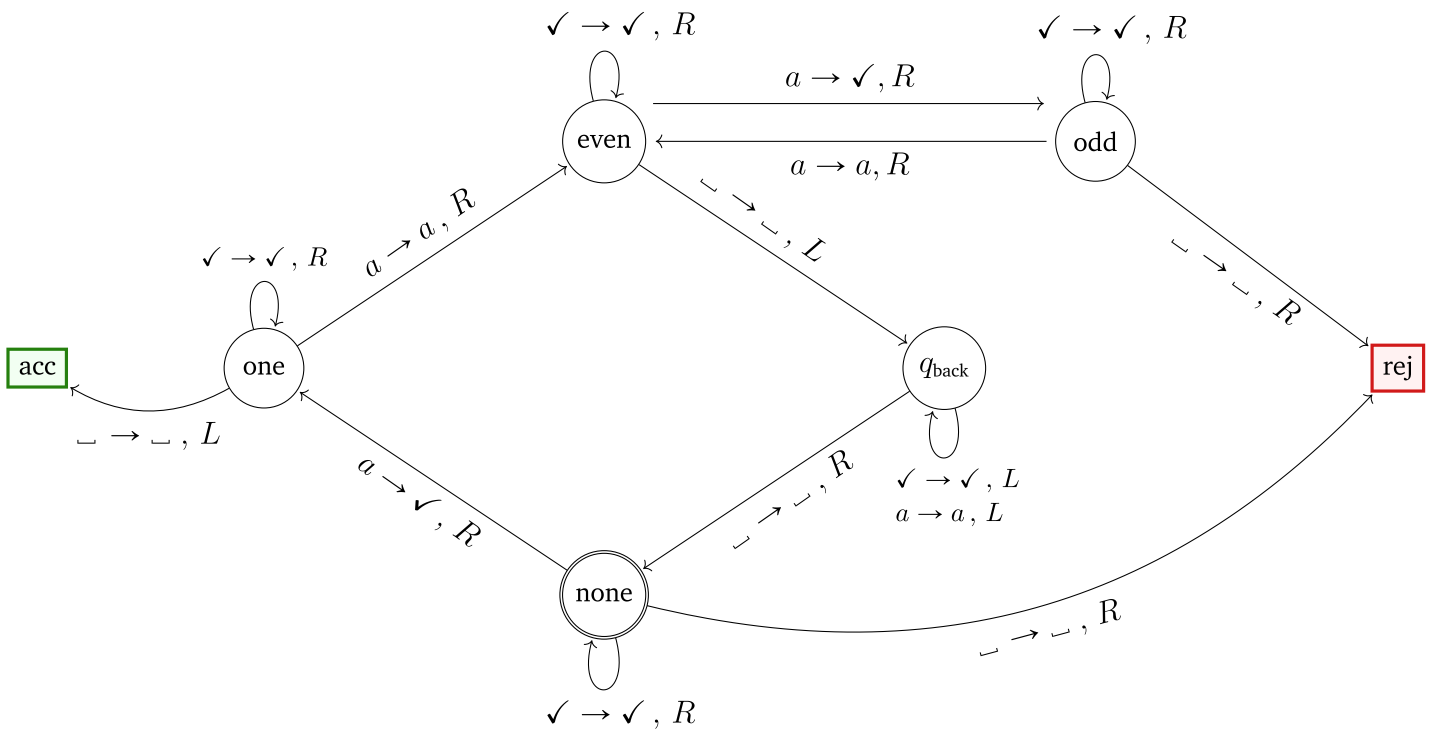
כעת נתן הגדרה פורמלית של המכונת טיורנג שמקבלת השפה הזאת:

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

כאשר $\Sigma=\{a\}$, $\Gamma=\{a,\sqcup,\checkmark\}$, והקבוצת המצבים היא $Q=\{q_0,\text{one},\text{even},\text{odd},q_{\text{acc}},q_{\text{rej}}\}$ כאשר המשמעותם הם מפורטים למטה:

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 q_{back}: חזרה שלמאלה.

הפונקצית המעברים מתוארת על ידי התרשים מצבים למטה.



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	aaa ␣
␣ ✓	one	a	aa ␣
␣ ✓ 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:

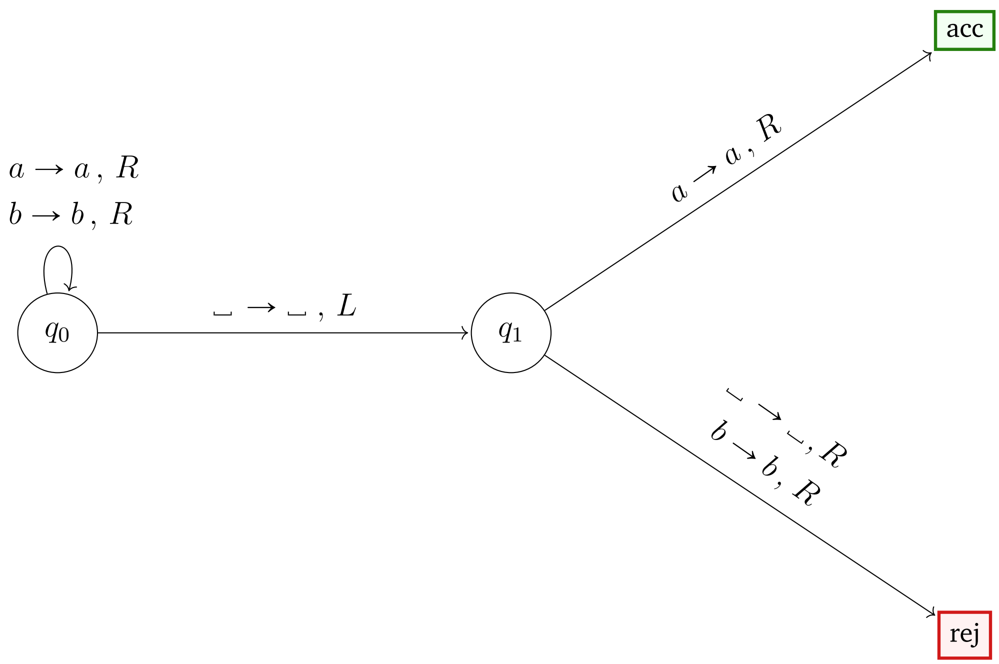
␣	none	a	a	a	␣
␣	✓	one	a	a	␣
␣	✓	a	even	a	␣
␣	✓	a	✓	odd	␣
␣	✓	a	✓	␣	rej

u	q	σ	v
␣	none	a	aa ␣

$\sqcup$ ✓	one	a	a $\sqcup$
$\sqcup$ ✓ a	even	a	$\sqcup$
$\sqcup$ ✓ a ✓	odd	$\sqcup$	$\sqcup$
$\sqcup$ ✓ a ✓ $\sqcup$	rej	$\sqcup$	$\sqcup$

דוגמה 1.8

מהי השפה של המכונה למטה:



פתרון:

1) סורקים את הקלט משמאל לימין.

- אם התו הנקרא a או b עוברים לתו ימינה הבא וחוזרים לשלב 1).
- אם התו הנקרא $\sqcup$ אז הגענו לסוף הקלט, ועוברים לשלב 2).

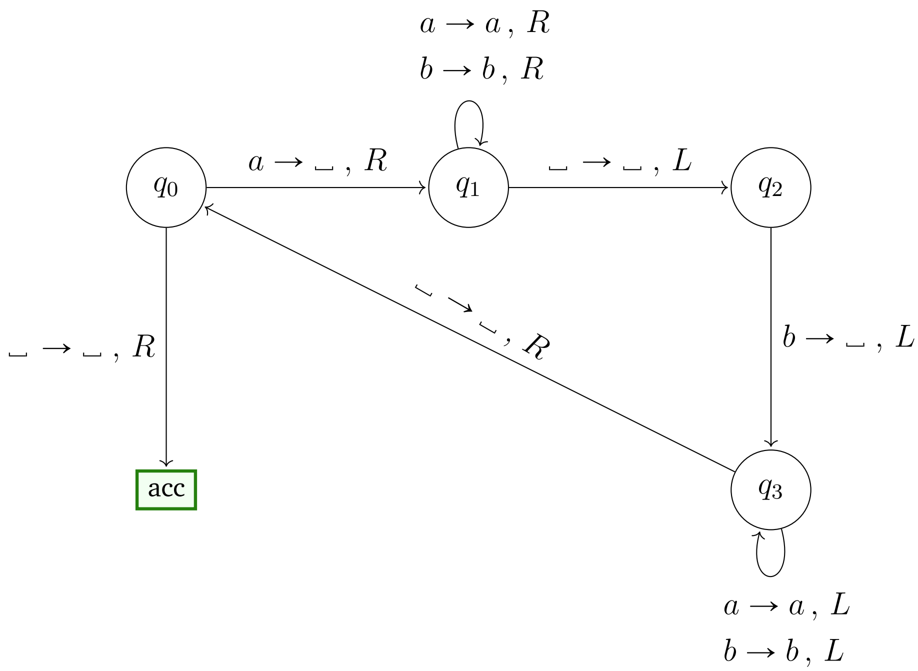
2) עוברים שמאלה לתו הארון של המילה.

- אם התו הנקרא $a \Leftarrow$ מקבל.
- אחרת דוחה.

לכן המכונה מקבלת שפת המילים המסתיימות באות a.

דוגמה 1.9

מהי השפה של המכונה למטה:



פתרון:

1) במצב ההתחלתי:

- אם התו הנקרא ␣ $\Leftarrow$ מקבל.
- אם התו הנקרא a מורידים אותו על ידי ␣ ועוברים לשלב 2).
- אחרת $\Leftarrow$ דוחה.

2) עוברים ימינה עד שמגיעים לסוף המילה.

- אם התו האחרון הוא b, מורידים אותו על ידי ␣, חוזרים לתחילת המילה וחוזרים לשלב 1).
- אחרת דוחה.

בכל איטרציה המכונה מורידה תו a בתחילת המילה וחוזרת ומורידה תו b תואם בסוף המילה. בכל איטרציה אם המכונה לא מוצאת b תואם בסוף המילה היא דוחה המילה. אחרת אם המכונה לא דחתה המילה וכל האותיות נמחקות אז המילה מתקבלת. לכן המכונה מקבלת שפת המילים

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

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

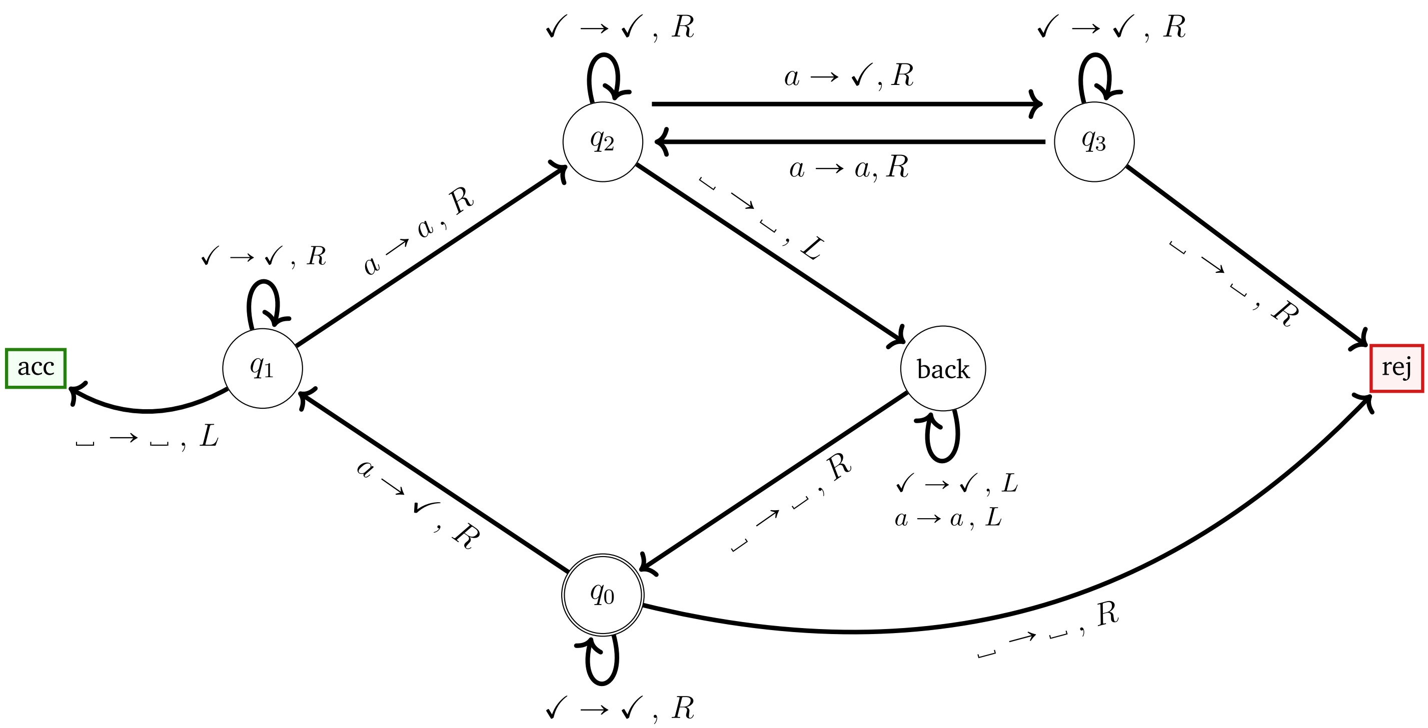
$$c_1 \vdash_M c_2$$

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

דוגמה 1.10 (המשך של דוגמה 1.5)

עבור המכונת טיורינג שמתוארת בתרשים למטה 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.

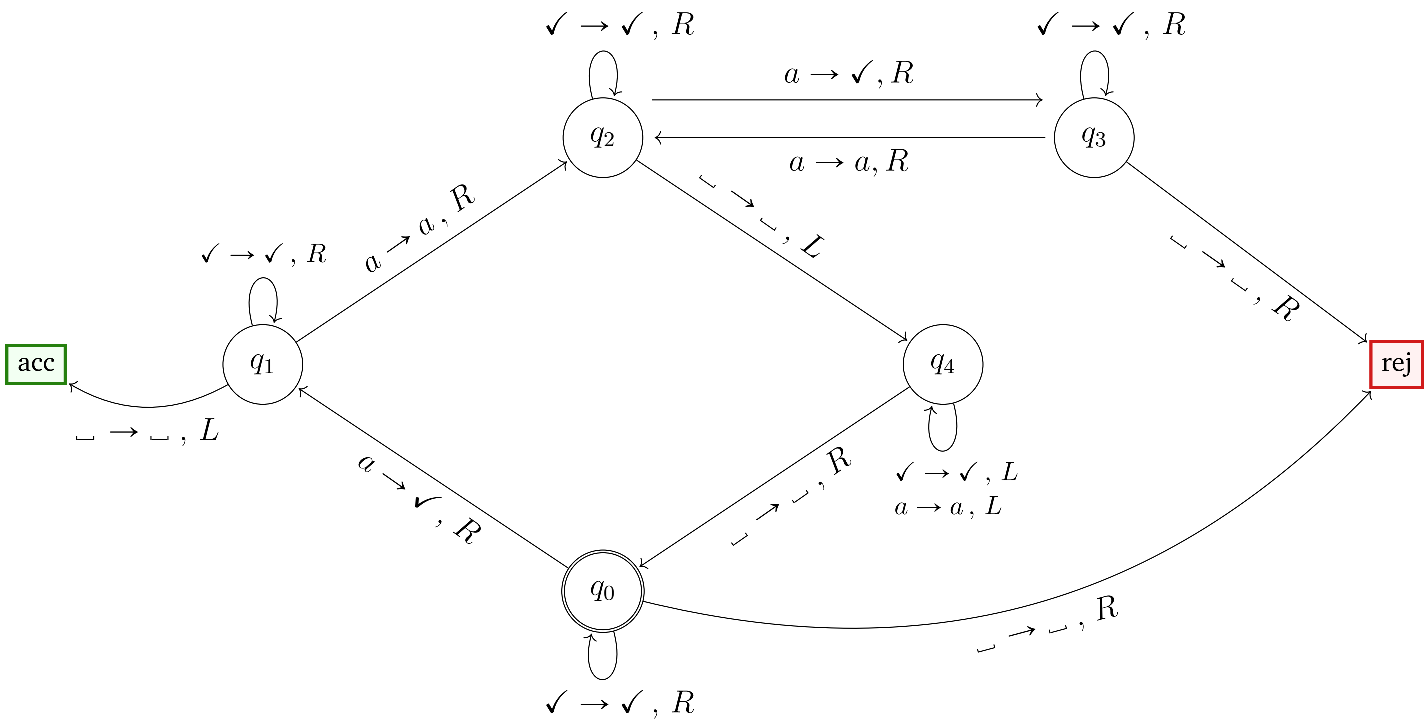
דוגמה 1.11 (המשך של דוגמה 1.5)

עבור המכונת טיורינג שמתוארת בתרשים למטה מתקיים

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

בגלל ש:

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



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
$$q_0 w \quad \vdash_M^* \quad u\ q_{\text{acc}}\, \sigma\, \mathrm{v}$$
כאשר $\sigma \in \Gamma$, $u, \mathrm{v} \in \Gamma^*$ כלשהם.

- M **דוחה** את w אם
$$q_0 w \quad \vdash_M^* \quad u\ q_{\text{rej}}\ \sigma\ \mathrm{v}$$
כאשר $\sigma \in \Gamma$, $u, \mathrm{v} \in \Gamma^*$ 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 1.12

בנו מכונת טיורינג שמכריעה את השפה

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

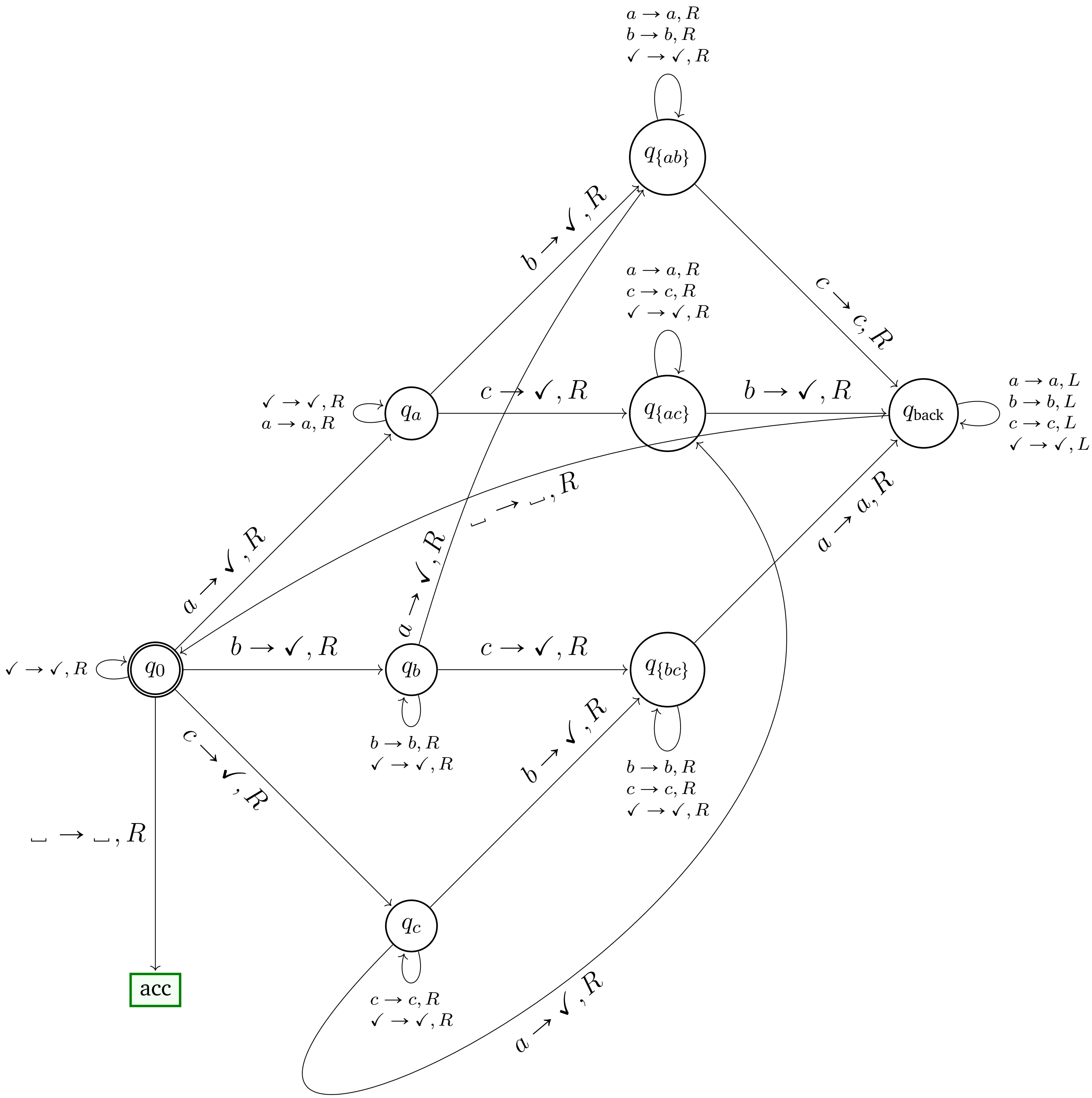
פתרון:

נתאר את המכונה על ידי הטבלת המעברים של המכונה. הסימן S מסמן כל זוג אותיות שונות מהקבוצה $\{a,b,c\}$ ללא חשיבות לסדר. כלומר:

$$S = \{a,b\} \ , \quad S = \{b,c\} \ , \quad S = \{a,c\} \ .$$

מצב	סימון בסרט	מצב חדש	כתיבה	תזוזה	תנאי
q_0	σ	$q.\sigma$	$\checkmark$	R	$\sigma \in \{a,b,c\}$
$q.\sigma$	σ	$q.\sigma$	$\circlearrowleft$	R	$\sigma \in \{a,b,c\}$
$q.\sigma$	τ	$q.\{\sigma\tau\}$	$\checkmark$	R	$\sigma,\tau \in \{a,b,c\} \ \wedge \ \sigma \neq \tau$
$q.S$	σ	$q.S$	σ	R	$\sigma \in S$
qS	σ	q_{back}	$\checkmark$	L	$\sigma \notin S$
q_{back}	$a,b,c,\checkmark$	q_{back}	$\circlearrowleft$	L	
q_0	$\sqcup$	q_{acc}	$\circlearrowleft$	R	
q_{back}	$a,b,c,\checkmark$	q_{back}	$\circlearrowleft$	L	
q_{back}	$\sqcup$	q_0	$\circlearrowleft$	R	

כעת נתאר את המכונה על ידי תרשים המצבים של המכונה:


$a \to a, R$
$b \to b, R$
$\checkmark \to \checkmark, R$
$q_{\{ab\}}$
$b \to \checkmark, R$
$c \to c, R$
$a \to a, R$
$c \to c, R$
$\checkmark \to \checkmark, R$
$\checkmark \to \checkmark, R$
$a \to a, R$
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
$c \to c, R$
$\checkmark \to \checkmark, R$
$a \to \checkmark, R$

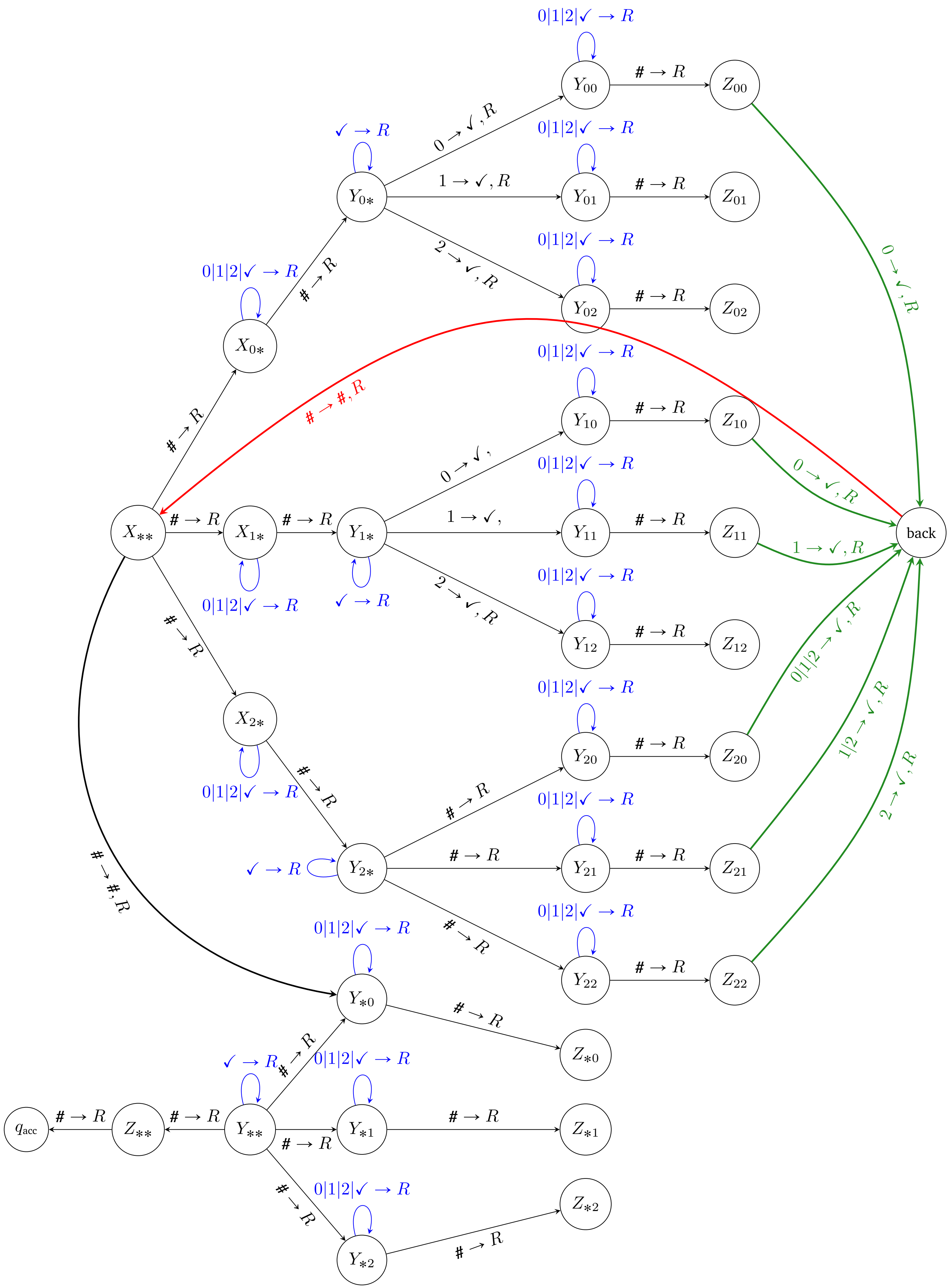
דוגמה 1.13

בנו מכונת טיורינג שמכריעה את השפה

$$\left\{x_1 \dots x_k \# y_1 \dots y_k \# z_1 \dots z_k \;\middle|\; x_i, y_i, z_i \in \{0,1,2\}\ ,\ \forall i, x_i \geqslant z_i \geqslant y_i\right\}$$

פתרון:

מצב	סימון בסרט	מצב חדש	כתיבה	תזוזה	תנאי
$X**$	σ	$X\sigma*$	$\checkmark$	R	
$X**$	$\checkmark$	$X**$	$\checkmark$	R	
$X\sigma*$	$0,1,2,\checkmark$	$X\sigma*$	$\circlearrowleft$	R	
$X\tau*$	#	$Y\tau*$	$\circlearrowleft$	R	
$Y\tau*$	σ	$Y\tau\sigma$	$\circlearrowleft$	R	
$Y\tau*$	$\checkmark$	$Y\tau*$	$\circlearrowleft$	R	
$Y\tau\sigma$	$0,1,2,\checkmark$	$Y\tau\sigma$	$\circlearrowleft$	R	
$Y\tau_1\tau_2$	#	$Z\tau_1\tau_2$	$\circlearrowleft$	R	
$Z\tau_1\tau_2$	$\checkmark$	$Z\tau_1\tau_2$	$\circlearrowleft$	R	
$Z\tau_1\tau_2$	σ	q_{back}	$\checkmark$	L	
$Z**$	$\sqcup$	q_{acc}	$\circlearrowleft$	R	$\tau_1 \geqslant \sigma \geqslant \tau_2$
q_{back}	$0,1,2,\checkmark$	q_{back}	$\circlearrowleft$	L	
q_{back}	$\sqcup$	$X**$	$\circlearrowleft$	R	


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

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:

- $\Sigma_1=\Sigma$ ו- $\Sigma_2\subset\Gamma$.
- לכל $w\in\Sigma_1^*$ מתקיים $q_0w\vdash q_{\text{acc}}f(w)$.

דוגמה 1.14 חיבור אונרי

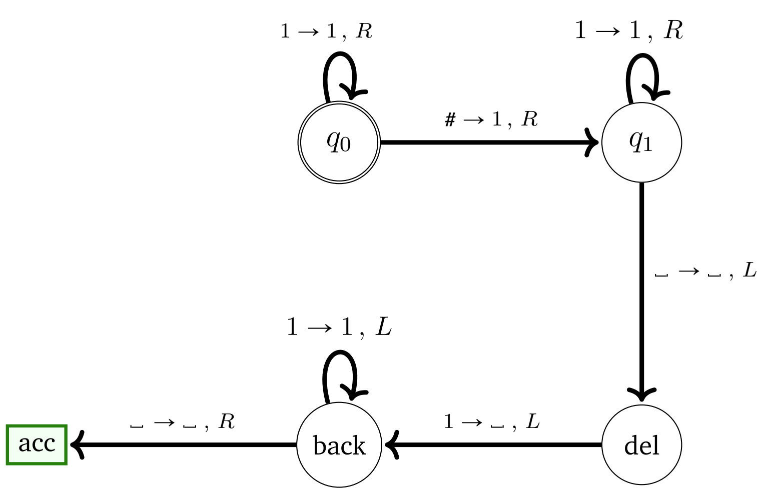
בנו מכונת טיורינג אשר מקבלת את הקלט

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

ומחזירה את פלט

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

פתרון:



דוגמה 1.15 כפל אונרי

בנו מכונת טיורינג אשר מקבלת את הקלט

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

ומחזירה את פלט

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

פתרון:

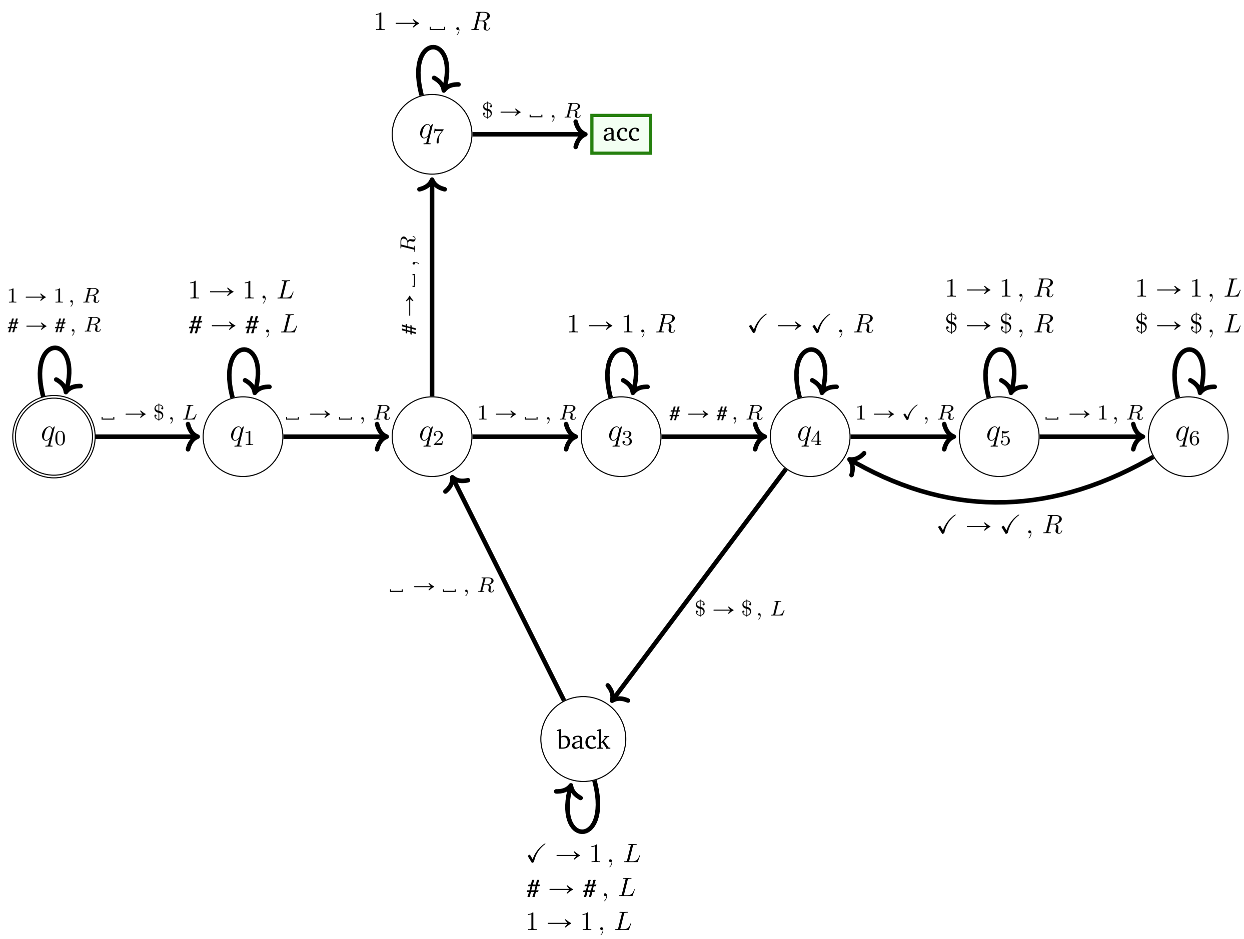
- לדוגמה, נניח שהקלט הוא 2 כפול 2.

  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	␣	␣

$\sqcup\sqcup 1\#\checkmark$	q_6	$\checkmark$	$\$11\sqcup$
$\sqcup\sqcup 1\#\checkmark\checkmark$	q_4	$\$$	$11\sqcup$
$\sqcup\sqcup 1\#\checkmark$	back	$\checkmark$	$\$11\sqcup$
$\sqcup$	back	$\sqcup$	$1\#11\$11\sqcup$
$\sqcup\sqcup$	q_2	1	$\#11\$11\sqcup$
$\sqcup\sqcup\sqcup$	q_3	$\#$	$11\$11\sqcup$
$\sqcup\sqcup\sqcup\#$	q_4	1	$1\$11\sqcup$
$\sqcup\sqcup\sqcup\#\checkmark$	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. בתחילת 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 שמאלה– 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 $M^O = \left(Q^O, \Sigma^O, \Gamma^O, \delta^O, q_0^O, \text{acc}^O, \text{rej}^O\right)$ במודל O.

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

- רכיבי המ"ט M^T זהים לאלו של המ"ט M^O, מלבד מהתכונה שהראש של M^O לעולם לא זז מעבר לקצה השמאל של הקלט.
- נעבוד רק עם צד ימין של הסרט האינסופי של M^T ואז M^T תהיה שקולה ל- M^O.

- כדי לדאוג שהראש של המכונה הדו-כיוונית M^T לא זז מעבר לקצה השמאל של הקלט, נוסיף מצבים חדשים וגם מעברים חדשים לפונקצית המעברים של M^T, שמבטיחים שהראש של M^T לא זז מעבר לקצה השמול של הקלט, באופן הבא.
- בתחילת כל חישוב, המכונה M^T מסמנת את המשבצת מצד שמאל וליד המשבצת הראשונה של הקלט בסימן מיוחד $\$$.
- נגדיר את הפונקצית המעברים של M^T כך שכל פעם שהראש מגיע למשבצת המסומנת $\$$, הראש חוזר ימינה למשבצת הראשונה של הקלט, כמפורט בטבלת המעברים למטה.

לכן, המכונה M^T השקולה למכונה M^O היא

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

כאשר

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

והפונקצית המעברים מתוארת בטבלה למטה.

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

הוכחנו את הכיוון הראשון:
ראינו מכונה דו -כיוונית השקולה למכונה חד -כיוונית.

כעת נוכיח את הטענה בכיוון הראשון:
נראה מכונה חד -כיוונית השקולה למכונה דו -כיוונית.

<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.

- נסמן "קו המפריד" על הסרט של המכונה הדו-כיוונית M^T.

$\cdots$	$\sqcup$	a	b	b	b	c	c	a	b	$\sqcup$	$\cdots$

- נסמן את המשבצת הראשונה של הסרט של המכונה החד-כיוונית M^O עם $\$$.
- כל שאר המשבצות של הסרט של M^O נחלק לשני חצאים: חצי העליון U וחצי התחתון D.
- תוכן הסרט של המכונה הדו־כיוונית M^T נכתב על סרטה של המכונה החד־כיוונית M^O כך:
  * החלק של הסרט שמצד שמאל של קו המפריד נכתב בשורה העליונה של סרט M^O בכיוון הפוך (מימין לשמאל).

∗ החלק של הסרט שמצד ימין של קו המפריד נכתב בשורה התחתונה של סרט M^O בכיוון הרגיל (משמאל לימין).

\$	b	b	a	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$
	b	c	c	a	b	$\sqcup$	$\sqcup$	$\cdots$

- ∗ תזוזה ימינה של M^T **מצד ימין של קו המפריד** $\Leftarrow$ תזוזה ימינה בשורה התחתונה של M^O.
  ∗ תזוזה ימינה של M^T **מצד שמאל של קו המפריד** $\Leftarrow$ תזוזה שמאלה בשורה העליונה של M^O.

תזוזה ימינה ב- M^T:

$\sqcup$	a $\rightarrow$	b $\rightarrow$	‖ b $\rightarrow$	b $\rightarrow$	c $\rightarrow$	c $\rightarrow$	a $\rightarrow$	$\sqcup$ $\rightarrow$

תזוזה שקולה ב-M^O:

\$	$\leftarrow$ b	$\leftarrow$ b	$\leftarrow$ a	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
	b $\rightarrow$	c $\rightarrow$	c $\rightarrow$	a $\rightarrow$	b $\rightarrow$	$\sqcup$ $\rightarrow$	$\sqcup$

∗ תזוזה שמאלה של M^T **מצד ימין של קו המפריד** $\Leftarrow$ תזוזה שמאלה בשורה התחתונה של M^O.
∗ תזוזה שמאלה של M^T **מצד שמאל של קו המפריד** $\Leftarrow$ תזוזה ימינה בשורה העליונה של M^O.

תזוזה שמאלה ב-M^T:

$\sqcup$	a $\leftarrow$	b $\leftarrow$	‖ b $\leftarrow$	b $\leftarrow$	c $\leftarrow$	c $\leftarrow$	a $\leftarrow$	$\sqcup$ $\leftarrow$

תזוזה שקולה ב-M^O:

\$	$\rightarrow$ b	$\rightarrow$ b	$\rightarrow$ a	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
	b $\leftarrow$	c $\leftarrow$	c $\leftarrow$	a $\leftarrow$	b $\leftarrow$	$\sqcup$ $\leftarrow$	$\sqcup$

לכן, המכונה M^O השקולה למכונה M^T היא

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

נסביר את כל הרכיבים של M^O:

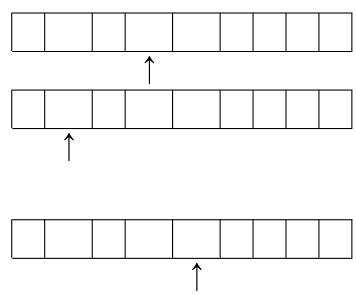
- לכל מצב $q \in Q^T$ נגדיר q_U ו- q_D של Q^O, כדי להבחין בין המצבים שבהם הראש נמצא בחלק העליון לבין המצבים שבהם הראש נמצא בחלק התחתון של הסרט.
- $\Sigma^O = \Sigma^T$.
- $\Gamma^O \subseteq \left(\Gamma^T \times \Gamma^T\right) \cup \{\$\}$.
- $q_{\text{acc}}^O = q_{\text{acc}}^T$.
- $q_{\text{rej}}^O = q_{\text{rej}}^T$.
- הפונקצית המעברים δ^O מתוארת בטבלת המעברים למטה. בטבלה, הסימנים $\tau, \sigma, \pi \in \Gamma^T$ מסמנים כל תו שבאלפבית Γ^T:

מצב	סימון	מצב חדש	כתיבה	תזוזה	תנאי
אתחול					
q_0^O	τ	q_τ	$\$$	R	$\tau \in \Sigma \cup \{\sqcup\}$ $\sigma \in \Sigma$
q_σ	τ	q_τ	$\sqcup$ σ	R	
$q_{\cdot\sqcup}$	$\sqcup$	back	$\sqcup$ $\sqcup$	L	
q_{back}	$\sqcup$ τ	q_{back}	$\circlearrowleft$	L	
q_{back}	$\$$	$q_0^T.D$	$\circlearrowleft$	R	
תזוזה מקורית שמאלה					
q_D	π σ	p_D	π τ	L	תזוזה שמאלה: $(q,\sigma) \xrightarrow{M^T} (p,\tau,L)$
q_U	σ π	p_U	τ π	R	
q_D	$\sqcup$	p_D	$\sqcup$ τ	L	תזוזה שמאלה: $(q,\sqcup) \xrightarrow{M^T} (p,\tau,L)$
q_U	$\sqcup$	p_U	τ $\sqcup$	R	
תזוזה מקורית ימינה					
q_D	π σ	p_D	π τ	R	תזוזה ימינה: $(q,\sigma) \xrightarrow{M^T} (p,\tau,R)$
q_U	σ π	p_U	τ π	L	
q_D	$\sqcup$	p_D	$\sqcup$ τ	R	תזוזה ימינה: $(q,\sqcup) \xrightarrow{M^T} (p,\tau,R)$
q_U	$\sqcup$	p_U	τ $\sqcup$	L	
פגיעה בקצה					
q_D	$\$$	q_U	$\circlearrowleft$	R	
q_U	$\$$	q_D	$\circlearrowleft$	R	
כל השאר עוברים ל-rej					

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

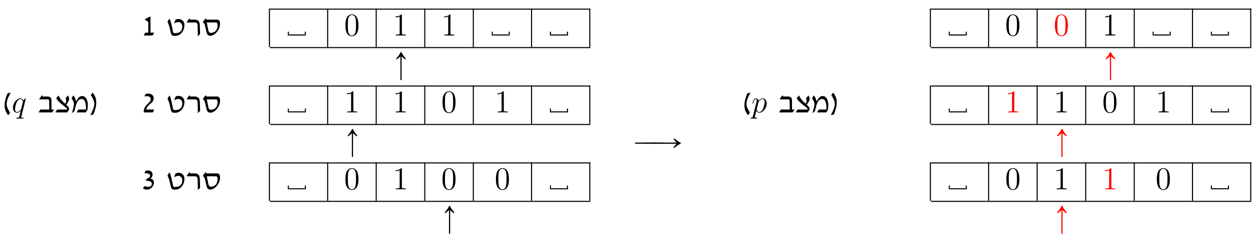
מכונת טיורינג מרובת סרטים היא שביעייה:

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

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 מכונת טיורינג מרובת סרטים

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:

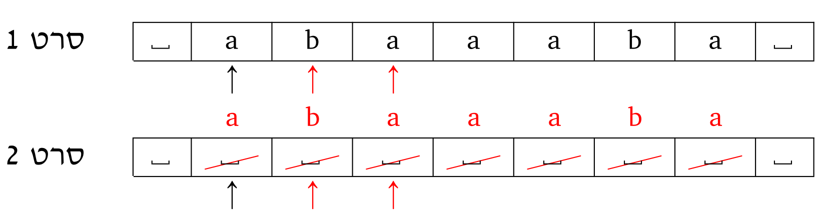
נבנה מ"ט עם שני סרטים:

<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:

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

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

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

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

$$\delta\left(q_{\text{back}},\begin{pmatrix}\sqcup\\\sqcup\end{pmatrix}\right)=\left(q_{\text{check}},\begin{pmatrix}\sqcup\\\sqcup\end{pmatrix},\begin{pmatrix}R\\S\end{pmatrix}\right)\ ,$$

$$\delta\left(q_{\text{check}},\begin{pmatrix}\sigma\\\sigma\end{pmatrix}\right)=\left(q_{\text{check}},\begin{pmatrix}\sigma\\\sigma\end{pmatrix},\begin{pmatrix}R\\L\end{pmatrix}\right)\ ,$$

$$\delta\left(q_{\text{check}},\begin{pmatrix}\sigma\\\tau\end{pmatrix}\right)=\left(q_{\text{rej}},\begin{pmatrix}\sigma\\\tau\end{pmatrix},\begin{pmatrix}S\\S\end{pmatrix}\right)\ ,$$

$$\delta\left(q_{\text{check}},\begin{pmatrix}\sqcup\\\sqcup\end{pmatrix}\right)=\left(q_{\text{acc}},\begin{pmatrix}\sqcup\\\sqcup\end{pmatrix},\begin{pmatrix}S\\S\end{pmatrix}\right)\ .$$

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:

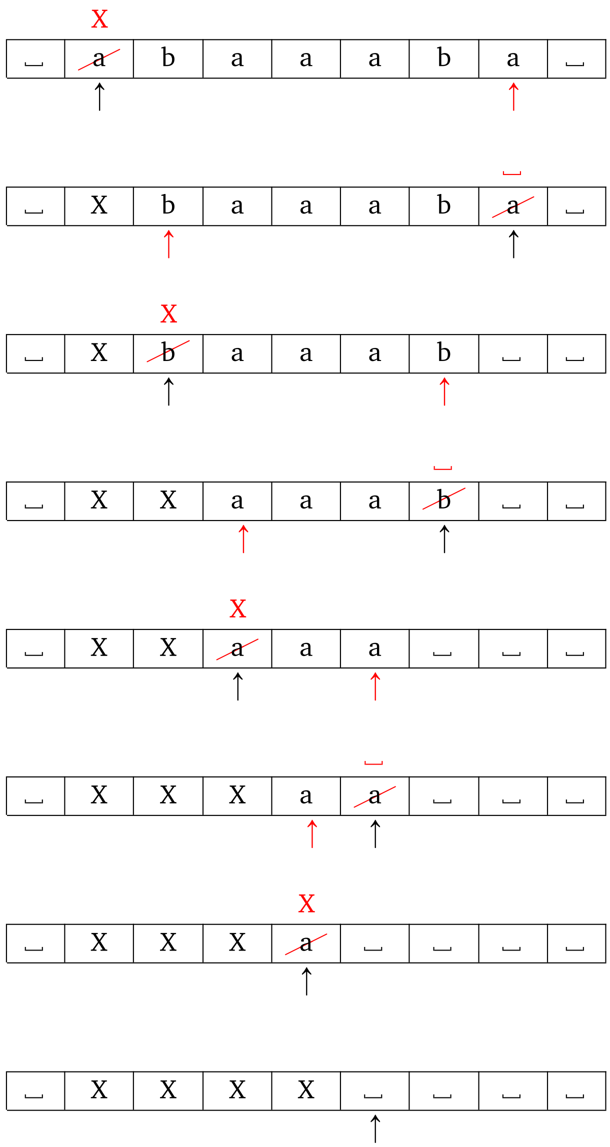
(1) אם התו שמתחת לראש הוא $\sqcup$ אז $M_1 \leftarrow$ acc.

(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.

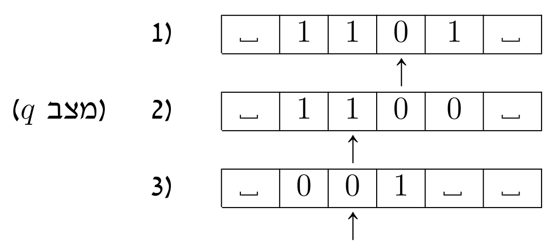
הוכחה:

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 $M' = \left(Q', \Sigma, \Gamma', \delta', q_0', q_{\text{acc}}', q_{\text{rej}}'\right)$ השקולה ל- M באופן הבא:

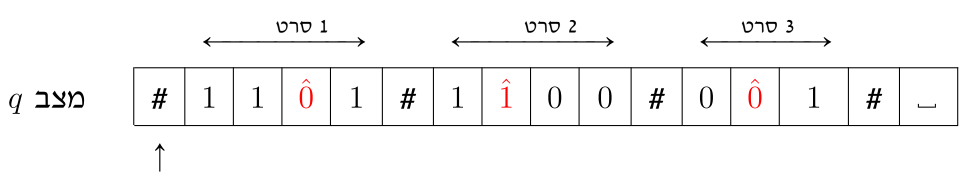
<u>רעיון הבנייה:</u>

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.

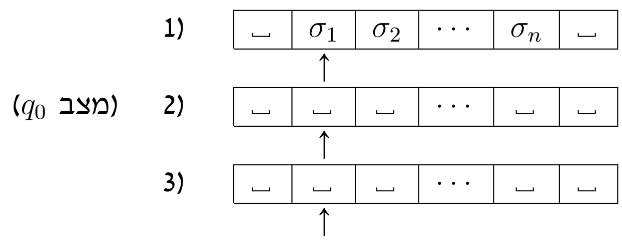
- בכל צעד חישוב, M' סורקת את הסרט שלה משמאל לימין כדי ללמוד מהם התווים שמתחת לראשים (התווים שמסומנים ב- $\hat{\alpha}$).
- M' משתמשת בפונקצית המעברים δ_k של M כדי לחשב את המעבר הבא.
- M' סורקת את הסרט שלה משמאל לימין כדי לעדכן את הסרטים ואת המיקום הראשים בהם.

<u>תאור הבנייה של M':</u>

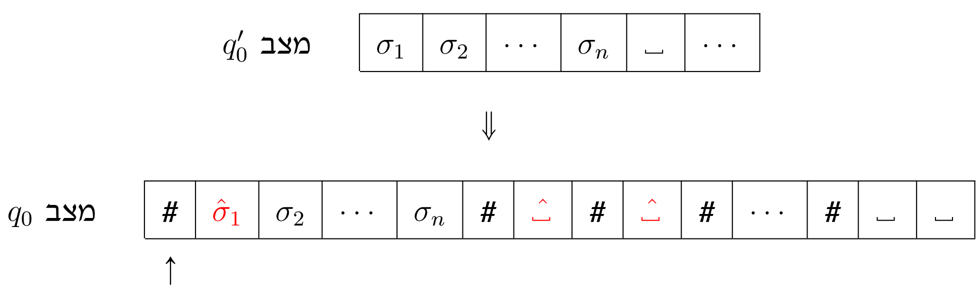
1) <u>שלב האיתחול</u>

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

ב- M

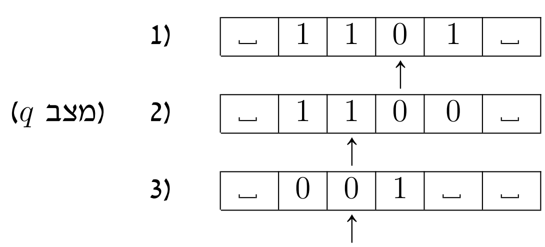


ב- M'

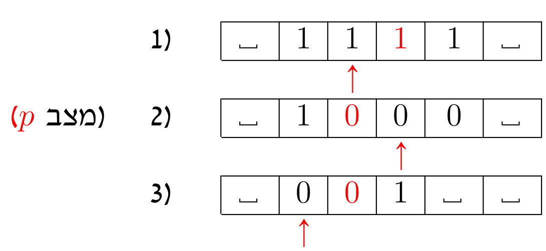


2) תאור צעד חישוב של M

ב- M



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



ב- M'

מצב q

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

$$\Downarrow$$

מצב p

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

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

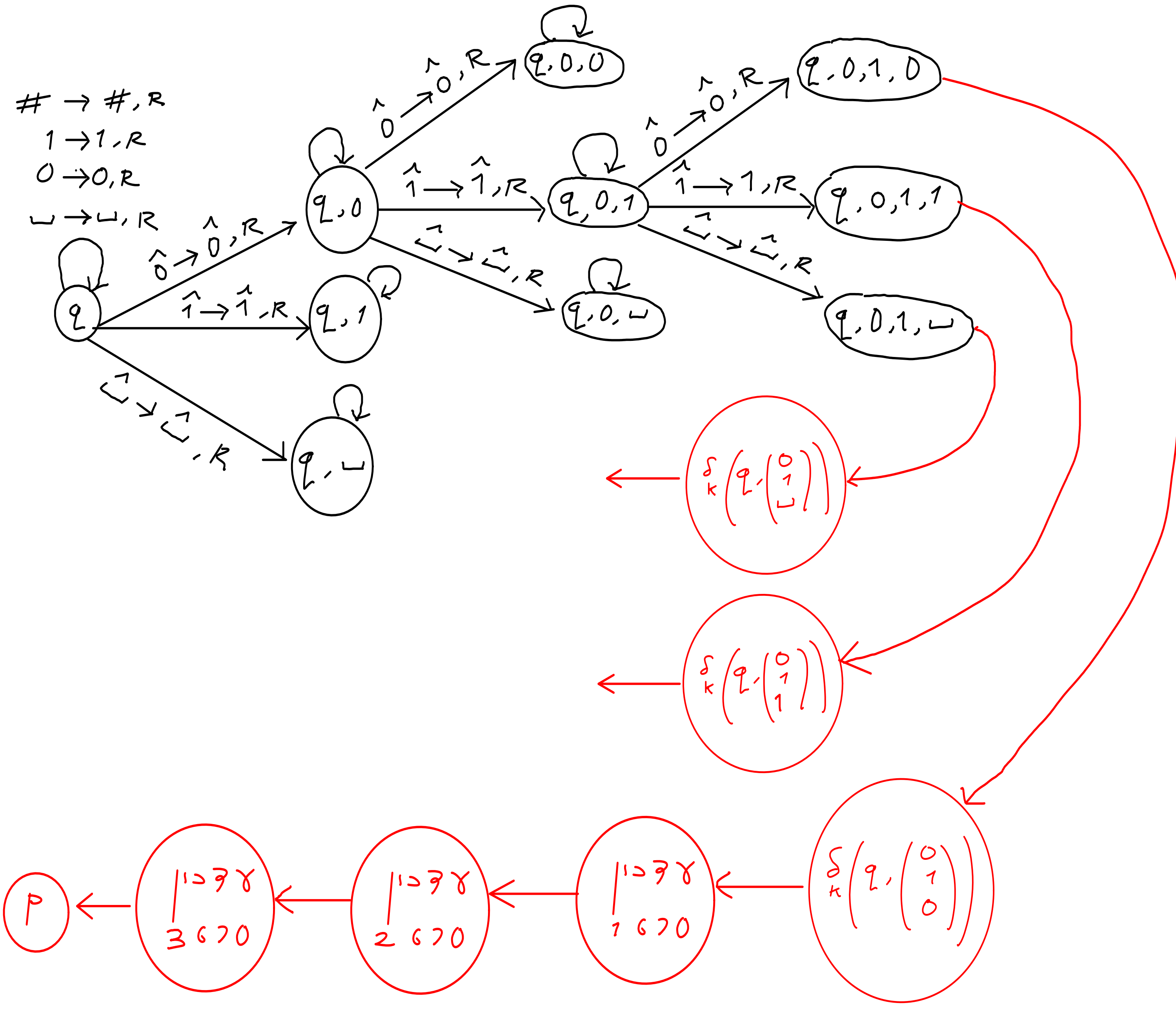
$$q\ ,\ \begin{pmatrix}0\\1\\0\end{pmatrix}\ .$$

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	#	$\sqcup$
$\uparrow$															



מצב p

#	1	$\hat{1}$	1	1	#	1	0	$\hat{0}$	0	#	$\hat{0}$	0	1	#	$\sqcup$
↑															

- <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.

- אם בכל שלב M' מזיזה אחד מן הראשים הווירטואליים ימינה אל סימן ה־ #, פעולה זו מציינת שמכונת M הזיזה את הראש המתאים אל החלק הריק שטרם נקרא של הסרט. לכן M' כותבת תו $\sqcup$ על המשבצת הזו ומזיזה את כל התוכן של הסרט בין התא הזה לבין ה־ # הימני ביותר בתא אחד ימינה. לאחר מכן היא ממשיכה את הסימולציה כרגיל.

■

שיעור 4
מכונת טיורינג אי דטרמיניסטית

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

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

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:

הרעיון

נבנה מ"ט א"ד N המכריעה את L.

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

						n			
סרט 1	$\sqcup$	$\cancel{1}$	$\cancel{1}$	$\cancel{1}$	$\cancel{1}$	$\cancel{1}$	$\cancel{1}$	$\cancel{1}$	$\cdots$

			t			
סרט 2	$\sqcup$	1	1	1	$\sqcup$	$\cdots$

תאור הבניה

$N =$ 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.

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

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

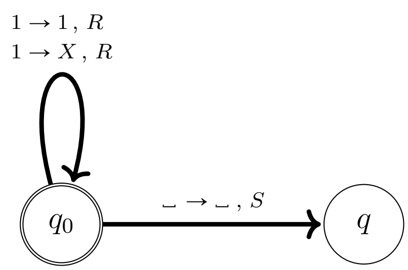
בהינתן מ"ט א"ד M ומילה $w \in \Sigma^*$, עץ החישוב של M ו- w הוא עץ מושרש שבו:

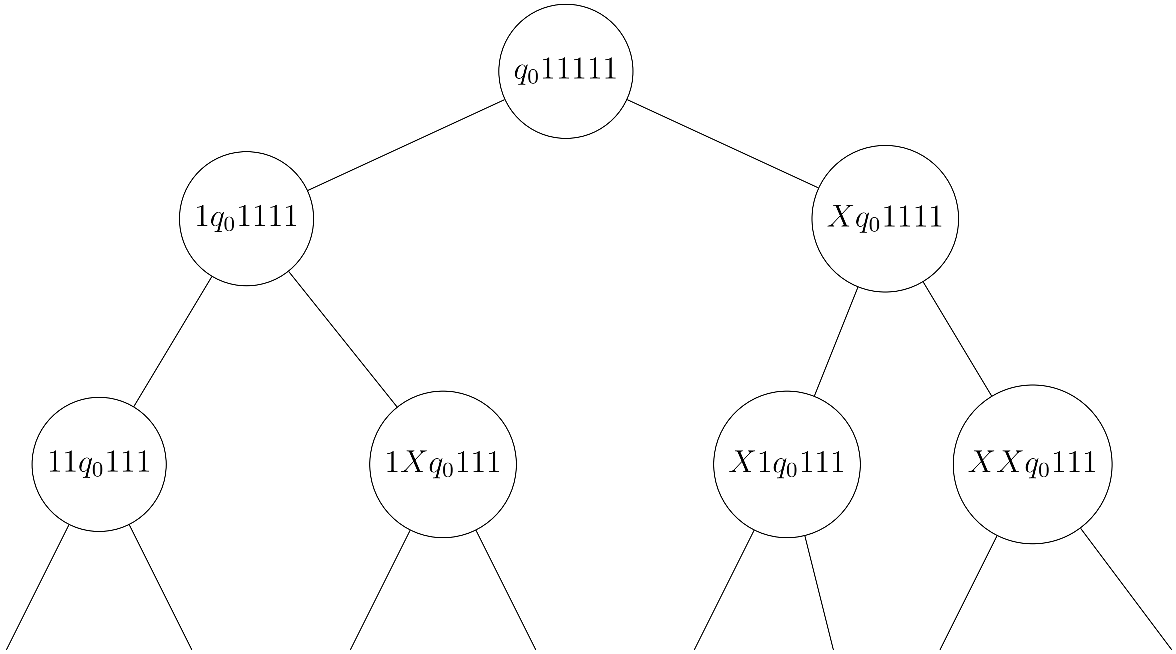
1) כל קדקוד בעץ מתאר קונפיגורציה בחישוב של M על w.

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 מכונת טיורינג אי-דטרמיניסטית למכונת טיורינג דטרמיניסטית

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

לכל מכונת טיורינג אי-דטרמיניסטית N קיימת מכונת טיורינג 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 מכונת טיורינג אי-דטרמיניסטית N נבנה מכונת טיורינג דטרמיניסטית D ונוכיח כי

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

<u>רעיון ההוכחה</u>

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.

<u>תאור הבניה</u>

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

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

אזי

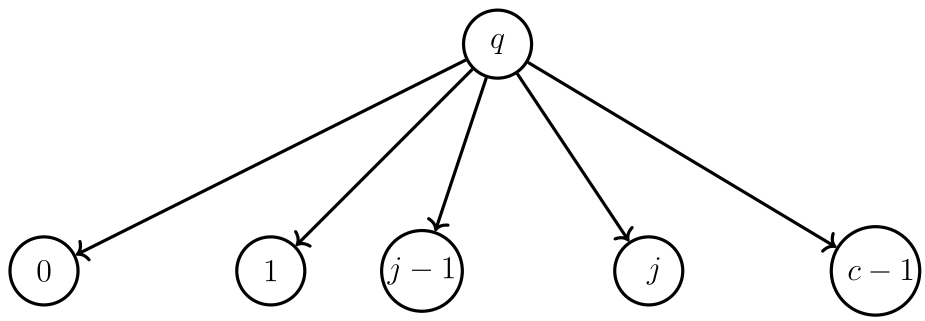
$$|\Delta(q,\alpha)|\leqslant|Q|\cdot|\Gamma|\cdot|\{L,R,S\}|=3|Q|\cdot|\Gamma|\ .$$

נסמן:

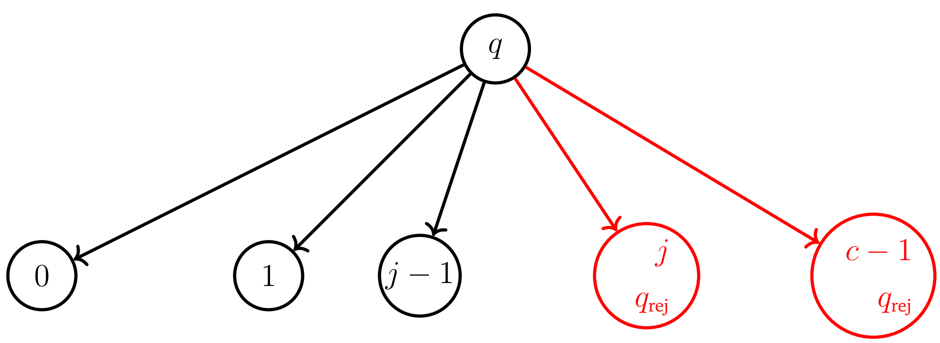
$$C=3|Q|\cdot|\Gamma|\ .$$

- לכל מצב $q\in Q$ ולכל אות $\alpha\in\Gamma$ נמספר את המעברים ב- $\Delta(q,\alpha)$ שרירותית

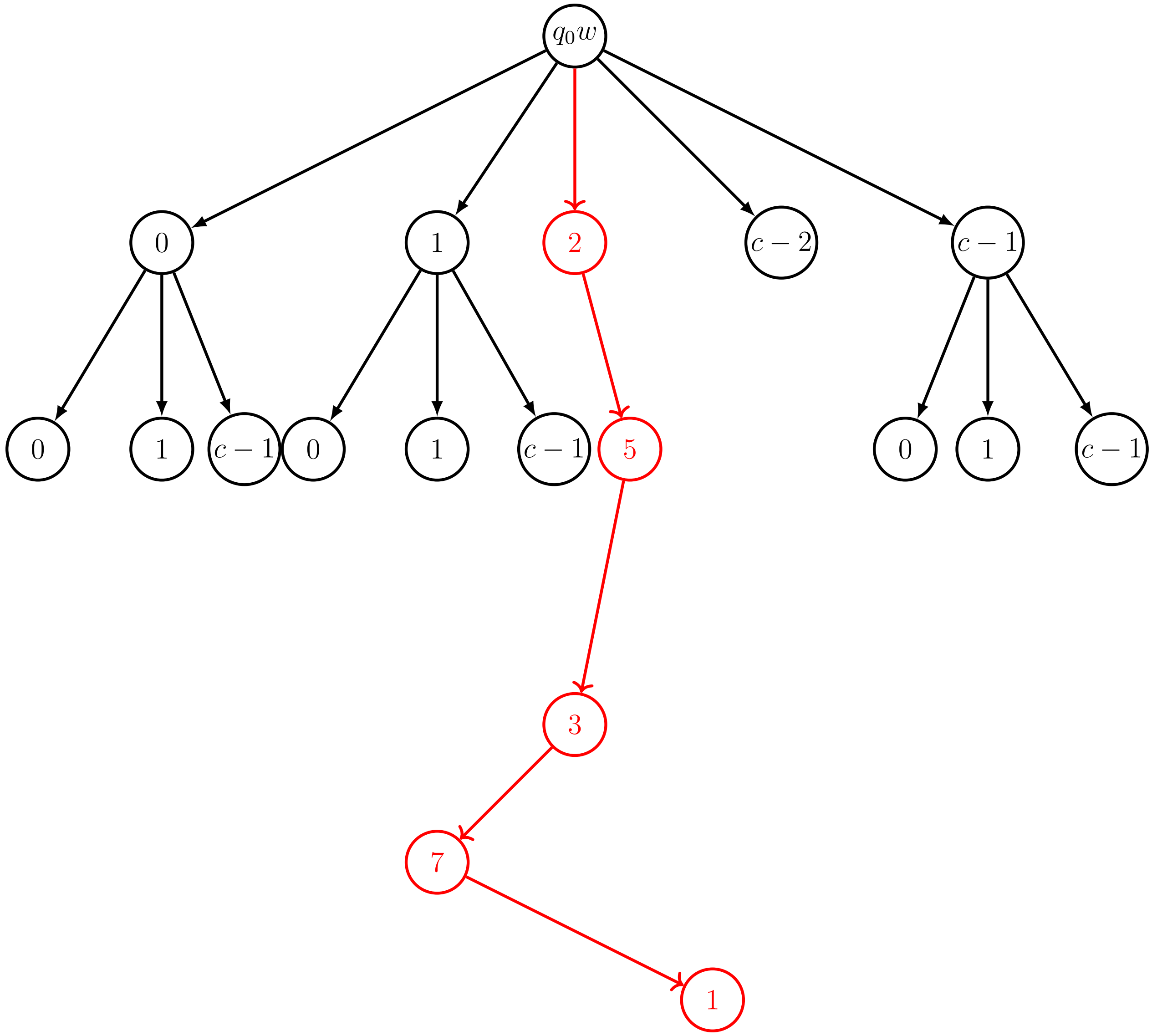
$$\{0,1,2,\cdots,C-1\}\ .$$



- 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.

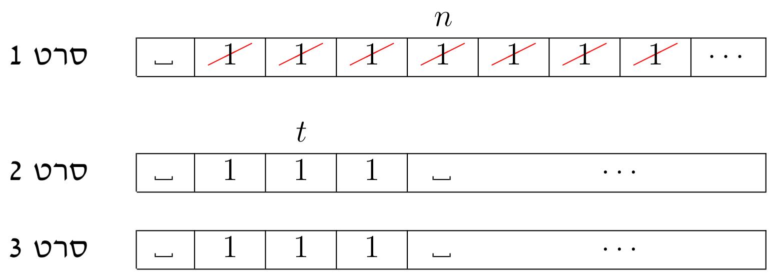


<u>קידום לקסיקוגרפי:</u>

$$\begin{array}{llllllll}
0 & 00 & 10 & \cdots & (C-1)0 & 000 & \\
1 & 01 & 11 & \cdots & (C-1)1 & 001 & \\
2 & 02 & 12 & \cdots & (C-1)2 & 002 & \cdots \\
\vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \\
C-1 & 0(C-1) & 1(C-1) & \cdots & (C-1)(C-1) & 00(C-1) &
\end{array}$$

<u>הבנייה של D</u>

D מכילה 3 סרטים:



$D =$ " 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

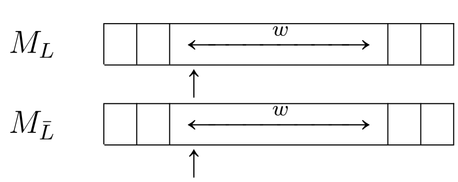
תהי L שפה.

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

הוכחה: תהי M_L מ"ט שמקבלת את L, ותהי $M_{\bar{L}}$ מ"ט שמקבלת את $\bar{L}$.
נבנה מ"ט D_L שמכריעה את L.

<u>כיצד תעבוד המ"ט D_L המכריעה?</u>

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



■

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

* אם M_L מקבלת $\leftarrow$ acc.
* אם $M_{\bar{L}}$ מקבלת $\leftarrow$ 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 מחשב כתובה בשפת תכנות, למשל
  * ג'אווה
  * פייתון
  * C
  * SIMPLE
- המרכיבים של שפת תכנות הם
  * משתנים
  * פעולות
  * תנאים
  * זרימה

נוכיח כי מכונט טיורינג ותוכנית מחשב שקולים חישובי.

5.3 SIMPLE

<u>משתנים</u>

- טבעיים:

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

מקבלים כערך מספר טבעי.

<u>מערכים</u>

- המערכים אין-סופיים

```
A[],
B[],
C[],
```

- בכל תא ערך מתוך א"ב Γ.

<u>אתחול</u>

- כל המשתנים מאותחלים ל- 0.
- הקלט נמצא בתאים הראשונים של `A[]`.

<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
```

<u>תנאים</u>

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

<u>זרימה</u>

- 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:

<u>כיוון ראשון:</u>

נוכיח כי לכל מ"ט M קיימת תוכנית P שקולה.
נבצע סימולציה של מ"ט M במחשב P .

בלי להכנס לפרטים, די ברור שבשפה עילית, כגון ג'אווה, ניתן להגדיר מבני נתונים עבור 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.

<u>כיוון שני:</u>

נוכיח כי 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.

<u>משתנים</u>

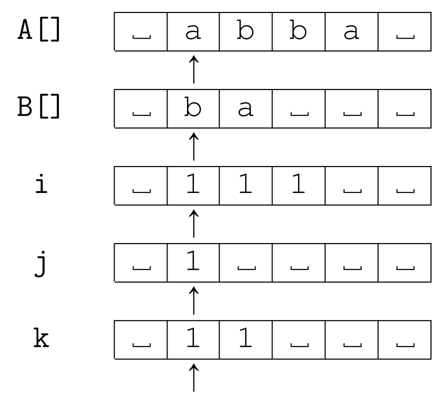
לכל משתנה יהיה סרט משלו.
המספר שהמשתנה יחזיק ייוצג בבסיס אונרי.
בהתחלה הסרט יהיה רק עם רווחים, זה מייצג את המספר אפס בבסיס אונרי.

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:

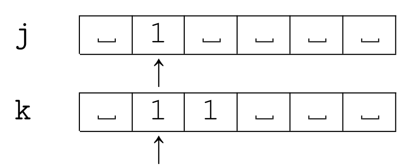


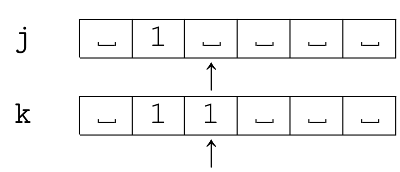
<u>פעולות</u>

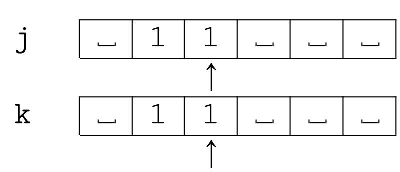
כעת נניח שנשים

```
j = k
```

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

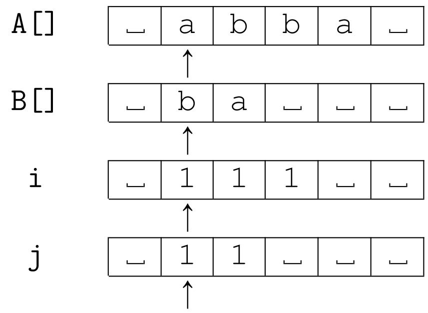




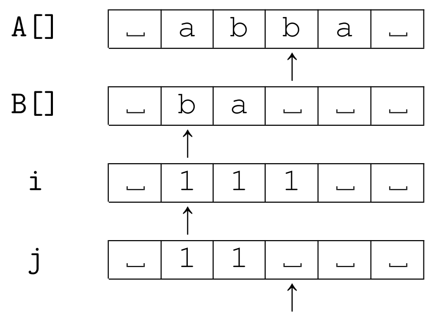


כעת נניח שנשים B[i]=A[j], ז״א B[3]=A[2].
נממש זה במ״ט ע״י להעתיק את תוכן משבצת 2 בסרט של A[] למשבצת 3 בסרט של B[].

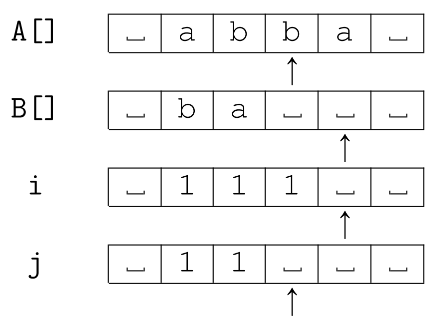
שלב 1)



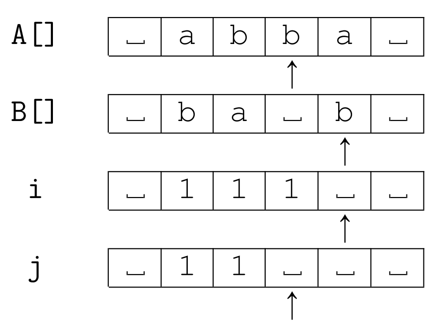
שלב 2)



שלב 3)



שלב 4)



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	␣	␣	␣
				↑		

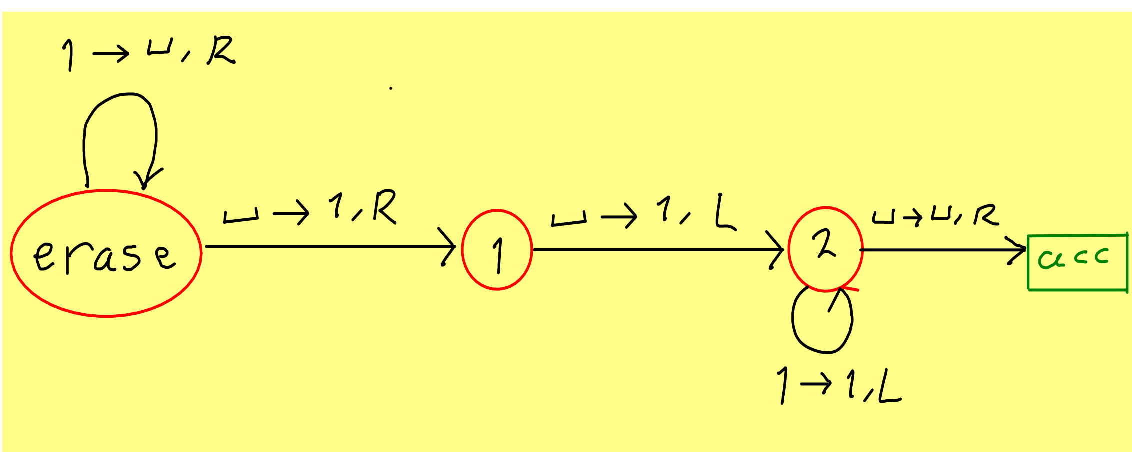
שלב 4)

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

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

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

פתרון:

תשובה:

$$L(G) = \{\mathtt{a}^n \mid n = 2^k \ , \ k \geqslant 1\} \ .$$

הסבר:

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

דוגמה 5.6

בנו דקדוק כללי אשר יוצר את הפשה

$$L = \{w \in \{a,b\}^* \mid \#a_w = \#b_w\} \ .$$

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.

$$\mathtt{S} \to \mathtt{S'}] \tag{1}$$
$$\mathtt{S'} \to \mathtt{aS'bC} \mid \varepsilon \tag{2}$$
$$\mathtt{Cb} \to \mathtt{bC} \tag{3}$$
$$\mathtt{C]} \to \mathtt{]c} \tag{4}$$
$$\mathtt{]} \to \varepsilon \tag{5}$$

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

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 המחרוות הנגזרות. הסוגר המרובע [מסמן את הקצה השמאלי של המילה השמאלית. הסוגר המסולסל { מסמן את הקצה השמאלי של המילה הימינית.	$S \rightarrow [H\{$
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 הימינית. (בדומה לזיכרונות של מ"ט).	$[H \rightarrow [aH_a$
3	כלל זה מאפשר לראש "לזוז" ימינה.	$H_a a \rightarrow aH_a$
4	כאשר המשתנה H_a "יגיע" לסוגר המסולסל, הוא יגזור אות a נוספת מימין לסוגר, שהוא הקצה השמאלי של המחרוזת הימינית. כך אפשרנו להוסיף שתי אותיות a: אחת מימין לסוגר [ואחת תואם ימין לסוגר {. כלומר אות a בקצה השמאלי של כל אחת המחרוזות.	$H_a\{ \rightarrow H\{a$
5	כעת צריך לאפשר למשתנה H לחזור שמאלה על גבי האותיות שבין הסוגרים, עד הסוגר [.	$aH \rightarrow Ha$

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 הימינית.	$[H \rightarrow [bH_b$
'3	כללים האלה מאפשרים לראש "לזוז" ימינה.	$H_a a \rightarrow aH_a$ $H_a b \rightarrow bH_a$ $H_b a \rightarrow aH_b$ $H_b b \rightarrow bH_b$
'4	כאשר המשתנה H_b "יגיע" לסוגר המסולסל, הוא יגזור אות b נוספת מימין לסוגר, שהוא הקצה השמאלי של המחרוזת הימינית. כך אפשרנו להוסיף שתי אותיות b: אחת מימין לסוגר [ואחת תואם ימין לסוגר {. כלומר אות b בקצה השמאלי של כל אחת המחרוזות.	$H_b\{ \rightarrow H\{b$
'5	כעת צריך לאפשר למשתנה H לחזור שמאלה על גבי האותיות שבין הסוגרים, עד הסוגר [.	$bH \rightarrow Hb$

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

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

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 $\text{v} \to \text{t}$ של G.
- אם $\text{t} \neq \text{y}$ 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 שממשמ 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

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

נניח שבדקדוק כללי G קיים אותו הצעד

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

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

- הירכיה 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

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

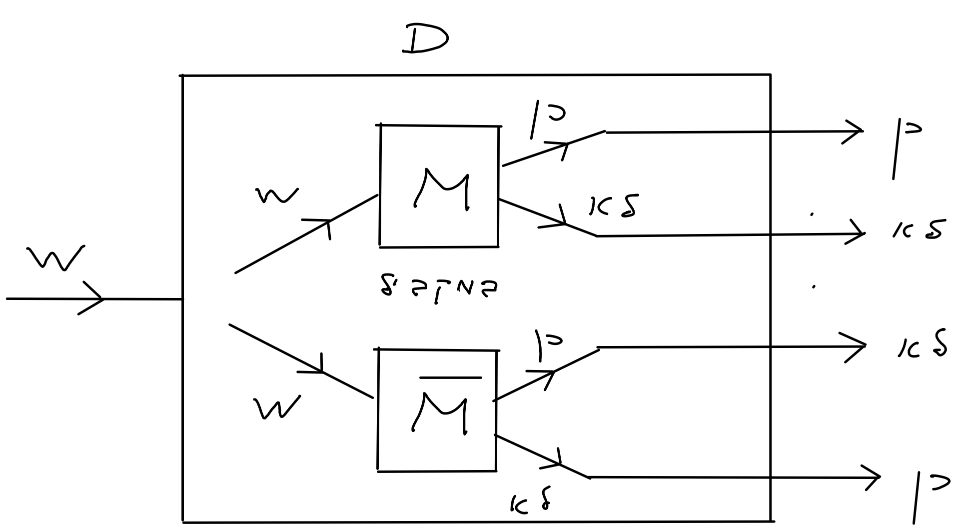
6.2 היחס בין הכרעה וקבלה

למה 6.1 היחס בין הכרעה וקבלה

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.

■

6.3 סגירות של שפות כריעות ושפות קבילות

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

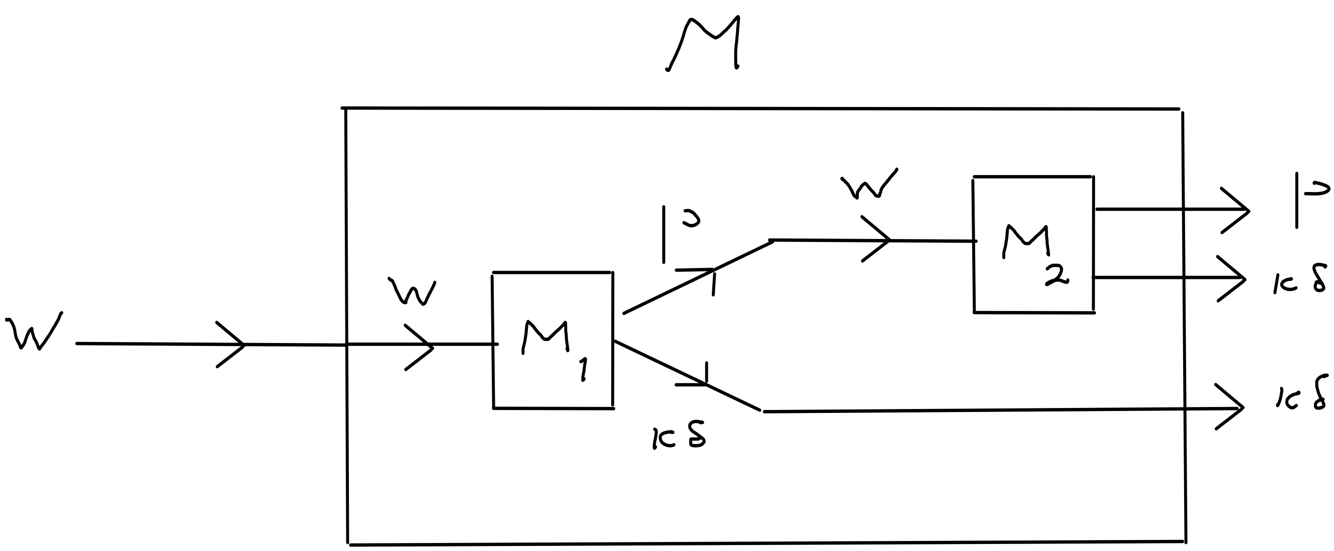
הוכחה:

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) <u>RE **סגורה תחת שרשור**</u>

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

4) קליני $*$

א) <u>R **סגורה תחת** $*$ **קליני**</u>

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^*.

<u>תאור הבנייה</u>

$M^* =$ 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) <u>RE **סגורה תחת** $*$ **קליני**</u>

5) משלים

א) <u>R **סגורה תחת המשלים**</u>

נוכיח כי

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

כאשר

$$\bar{L} = \{ w \in \Sigma^* \mid w \notin L \} \ .$$

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.

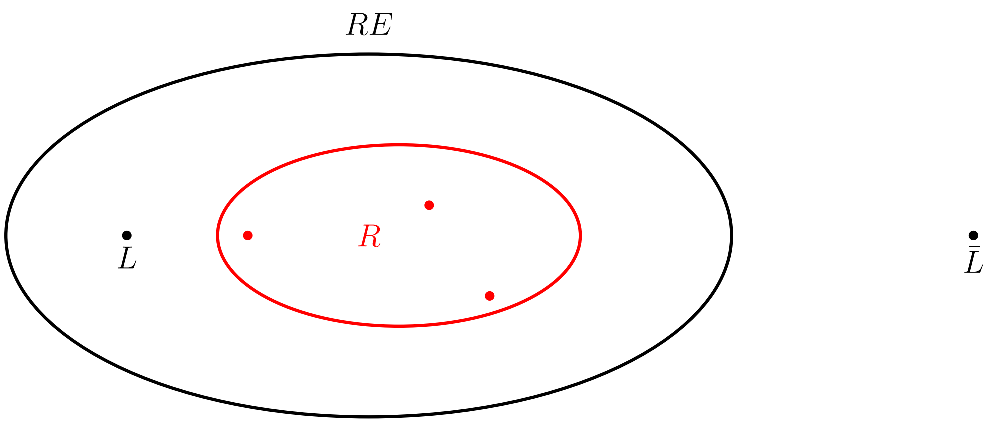
- אם M מקבלת $\Leftarrow$ $\bar{M}$ דוחה.
- אם M דוחה $\Leftarrow$ $\bar{M}$ מקבלת.

ב) <u>RE אינה סגורה תחת המשלים</u>

■

משפט 6.3 RE אינה סגורה תחת המשלים

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



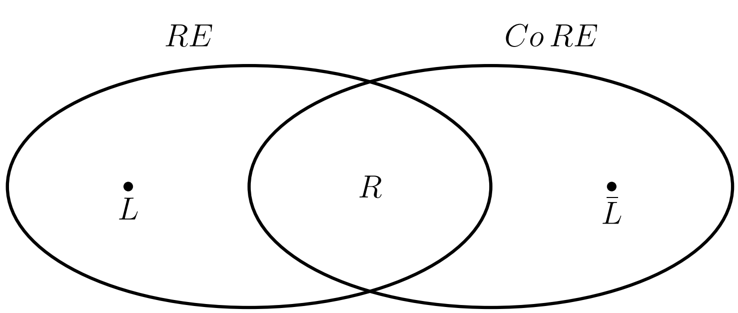
הוכחה:
נניח כי $L \in RE \backslash R$ ונניח בשלילה כי $\bar{L} \in RE$.

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

■

הגדרה 6.3 $Co\,RE$

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



<u>אבחנה</u>

לפי למה 6.1:

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

6.4 סגירות של שפות כריעות תחת $DROPOUT$

משפט 6.4 סגירות של שפות כריעות תחת $DROPOUT$

תהי L שפי כריעה ותהי $\mathrm{DROPOUT}(L)$ השפה הבאה:

$$\mathrm{DROPOUT}(L) = \left\{u\mathrm{v}\ \middle|u\mathrm{v} \in L\ ,\ \sigma \in \Sigma^*\ :\ u\sigma\mathrm{v} \in L\right\}$$

אזי השפה $\mathrm{DROPOUT}(L)$ גם כריעה.

הוכחה:
השפה L כריעה אזי קיימת מכונת טיורינג M_L המכריעה אותה. נבנה מכונת טיורינג M_{DROPOUT} המכרעיה את $\mathrm{DROPOUT}(L)$ באופן הבא:

<u>בניית המכונת טיורינג</u>

$M_{\mathrm{DROPOUT}} =$ על כל קלט w:

(1 בוחרת חילוק של המילה w לשרשור של שני מילים באופן אי-דטרמיניסטי כך ש: $w = u\mathrm{v}$.

(2 בוחרת אות $\sigma \in \Sigma^*$ באופן אי-דטרמיניסטי ובונה את המילה $w = u\sigma\mathrm{v}$.

(3 מריצה M_L על המילה $w = u\sigma\mathrm{v}$ ועונה כמוה.

<u>הוכת הנכונות</u>

נניח ש- $w \in \mathrm{DROPOUT(L)}$.

⇐ קיימים מילים $u,\mathrm{v} \in L$ וקיים $\sigma \in \Sigma^*$ כך ש- $u\sigma\mathrm{v} \in L$.

⇐ קיימת ריצה של M_{DROPOUT} עבורה M_{DROPOUT} תבחר חילוק $w = uv$ ו- M_{DROPOUT} תבחר $\sigma \in \Sigma^*$ כך ש- $u\sigma\mathrm{v} \in L$.

⇐ M_{DROPOUT} תקבל את w.

נניח ש- $w \notin \mathrm{DROPOUT(L)}$.

⇐ לא קיימים מילים $u,\mathrm{v} \in L$ ו- $\sigma \in \Sigma^*$ כך ש- $u\sigma\mathrm{v} \in L$.

⇐ כל ריצה של M_{DROPOUT} עוברת ל- q_{rej}.

⇐ M_{DROPOUT} תדחה את w.

■

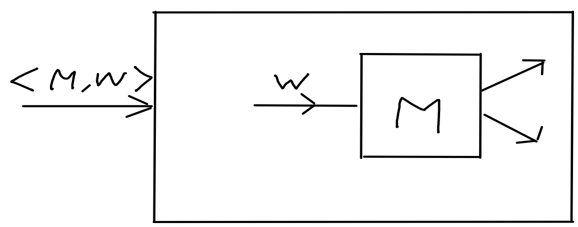
6.5 קידוד של מ"ט דטרמיניסטית

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 $O_1,\dots,O_k$ נסמן את הקידוד שלהם $\langle O_1,O_2,\dots,O_k\rangle$.

6.6 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.

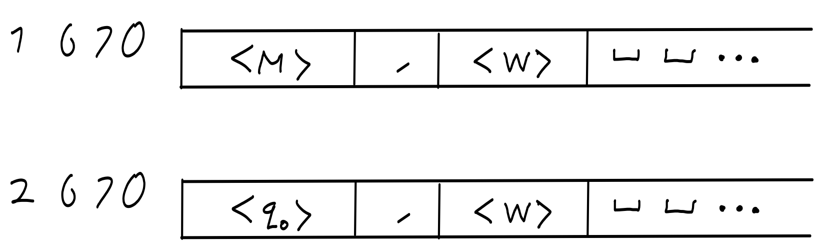
<u>תאור הפעולה של U</u>

$U =$ 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}

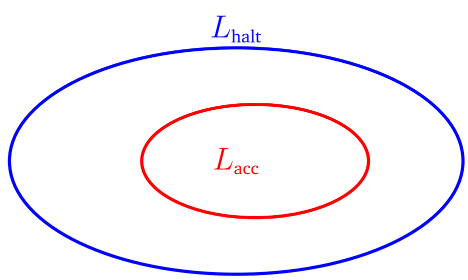
$$L_{\text{halt}} = \{\langle M, w\rangle \mid \text{עוצרת על } w \ M\} \in RE \backslash R$$

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 6.5

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

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

משפט 6.6

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

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}

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

הגדרה 7.2 L_{halt}

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

הגדרה 7.3 L_{d}

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

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

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

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

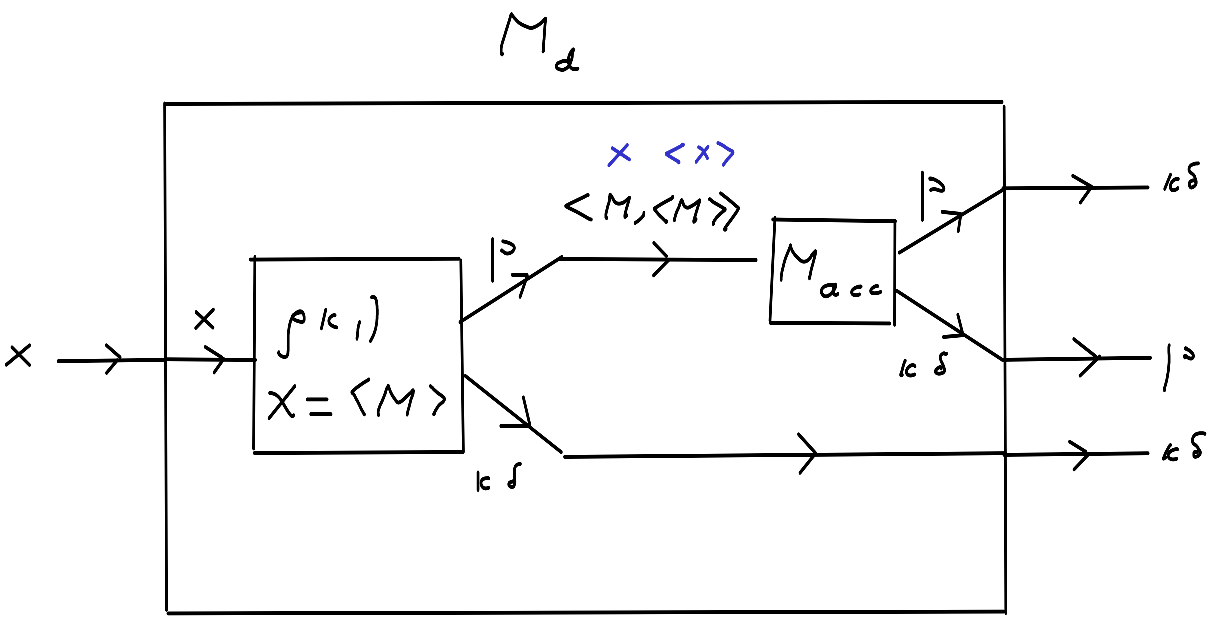
משפט 7.4 L_{acc} **לא כריעה**

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

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>

$M_{\text{d}} =$ על קלט x:

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

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

(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 $x \in L_{\text{d}}$

$\Leftarrow$ $x = \langle M\rangle$ 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 $x \notin L_{\text{d}}$ שני מקרים:

מקרה (1): $x \neq \langle M\rangle$ $\Leftarrow$ M_{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.

$\blacksquare$

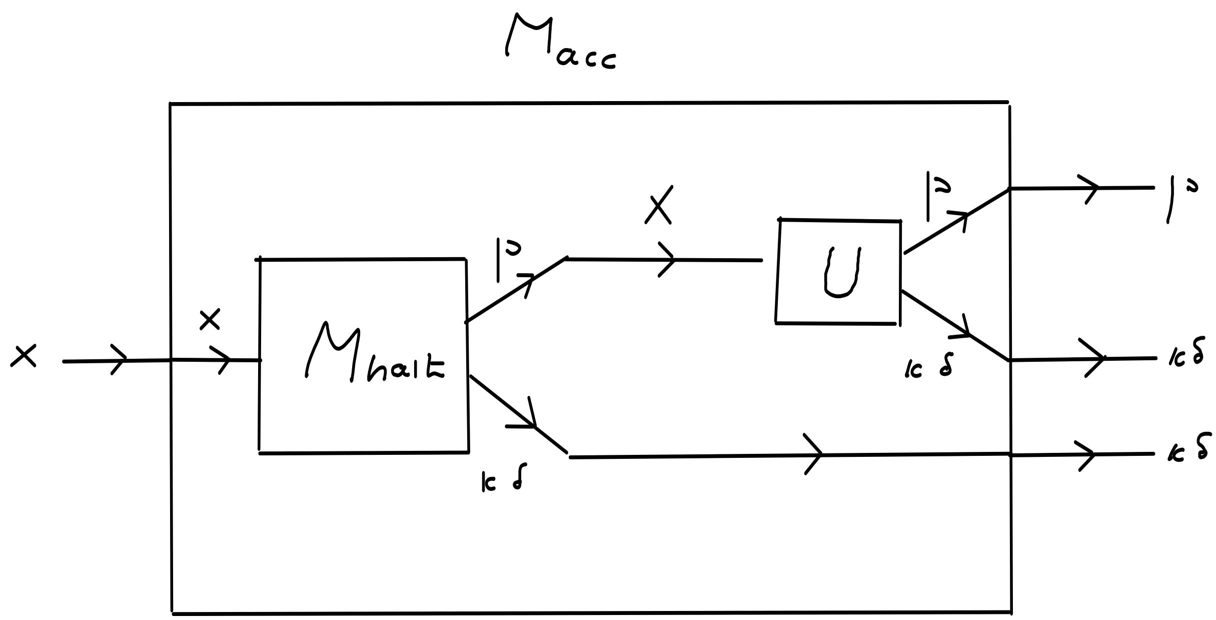
משפט 7.5 L_{halt} לא כריעה

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

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_{halt} 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:

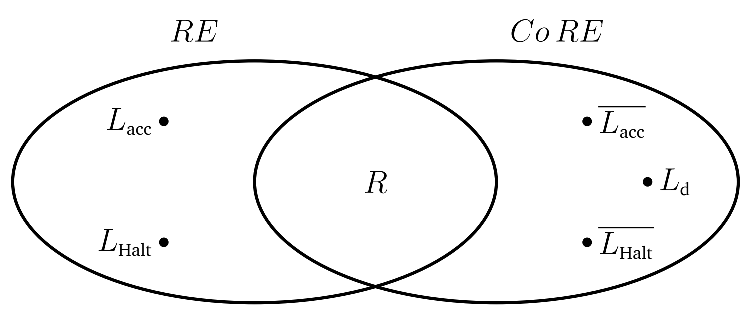
מקרה (א): M לא עוצרת על w $\Leftarrow$ M_{halt} דוחה את x $\Leftarrow$ M_{acc} דוחה את x.

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 מכונת טיורינג אי-דטרמיניסטית 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 אי-דטרמיניסטית.

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.

בנייה של M_E

$M_E =$ 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.

נכונות

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:

מקרה 1: $x \neq \langle M \rangle$ $\Leftarrow$ M_E דוחה.

מקרה 2: $\Leftarrow$ $x = \langle M \rangle$ 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 $M_{\bar{EQ}}$ 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}	$\times$	$\checkmark$
$\overline{L_{\text{acc}}}$	$\times$	$\times$
L_{d}	$\times$	$\times$
L_{Halt}	$\times$	$\checkmark$
$\overline{L_{\text{Halt}}}$	$\times$	$\times$
L_{E}	$\times$	$\times$
$\overline{L_{\text{E}}}$	$\times$	$\checkmark$
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 84
$\bar{L}_{\text{acc}} \leqslant L_{\text{NOTREG}}$	דוגמה 8.11 עמוד 88
$L_{\text{acc}} \leqslant L_{\text{NOTREG}}$	דוגמה 8.12 עמוד 89
$L_{\text{HALT}} \leqslant L_{\text{NOTREG}}$	דוגמה 8.13 עמוד 90
$L_{\text{acc}} \leqslant L_{\text{REG}}$	דוגמה 8.15 עמוד 92
$\bar{L}_{\text{acc}} \leqslant L_{\text{REG}}$	דוגמה 8.14 עמוד 91
$\bar{L}_{\text{acc}} \leqslant L_{M_1 \cap !M_2}$ כאשר $L_{M_1 \cap !M_2} = \{\langle M_1, M_2, w\rangle \mid w \in L(M_1) \wedge w \notin L(M_2)\}$.	דוגמה 8.16 עמוד 93
$\bar{L}_{\text{acc}} \leqslant L_{M_1 \subset M_2}$ 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 93

8.2 מכונת טיורינג המחשבת את פונקציה

הגדרה 8.1 מכונת טיורינג 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

מכונת טיורינג שמחשבת פונקציה עוצרת תמיד.

הגדרה 8.2 פונקציה חשיבה

בהינתן פונקציה $f : \Sigma^* \to \Sigma^*$ אומרים כי f חשיבה אם קיימת מכונת טיורינג 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.

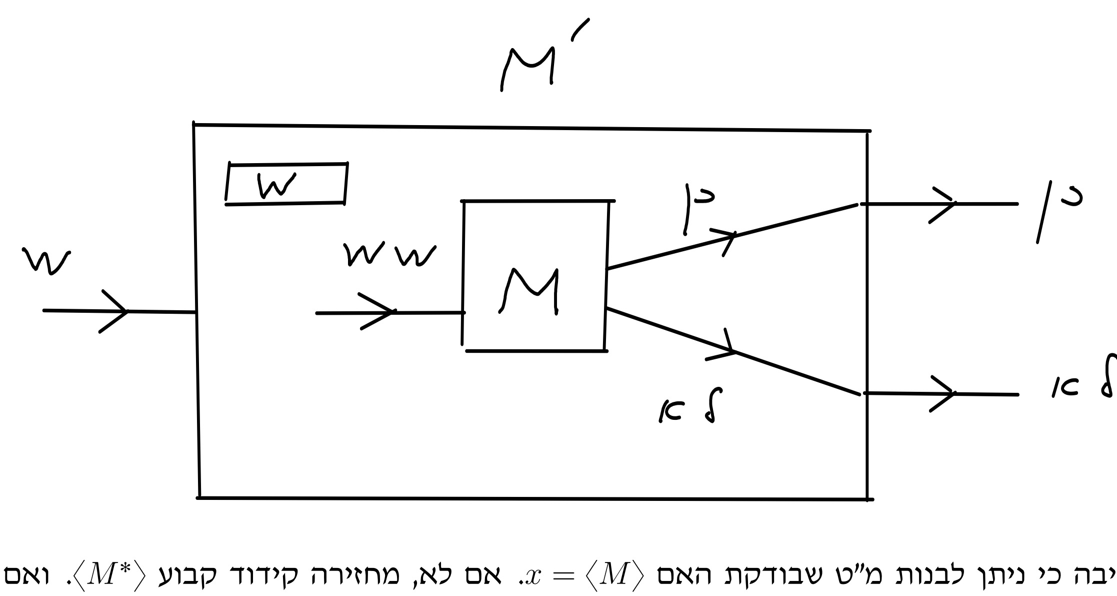
דוגמה 8.3

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

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

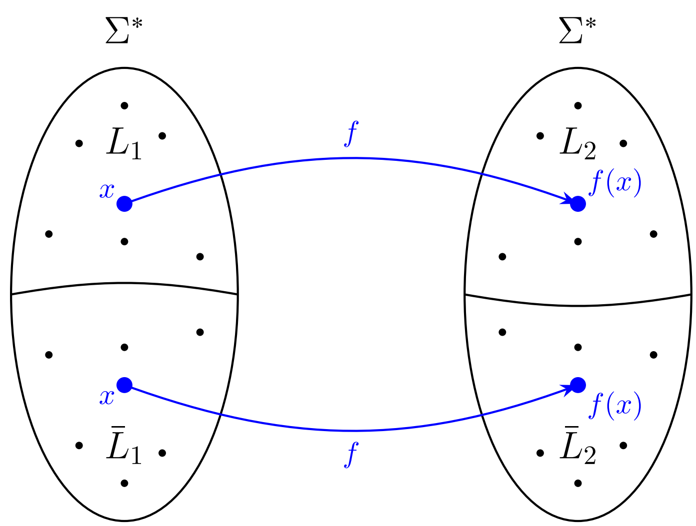
$$L_1 \leqslant L_2 \ ,$$

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>

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

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

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:

(1) $L_2 \in R \Rightarrow L_1 \in R$

(2) $L_2 \in RE \Rightarrow L_1 \in RE$

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

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

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 $L_2 \in R \Rightarrow L_1 \in R$</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.

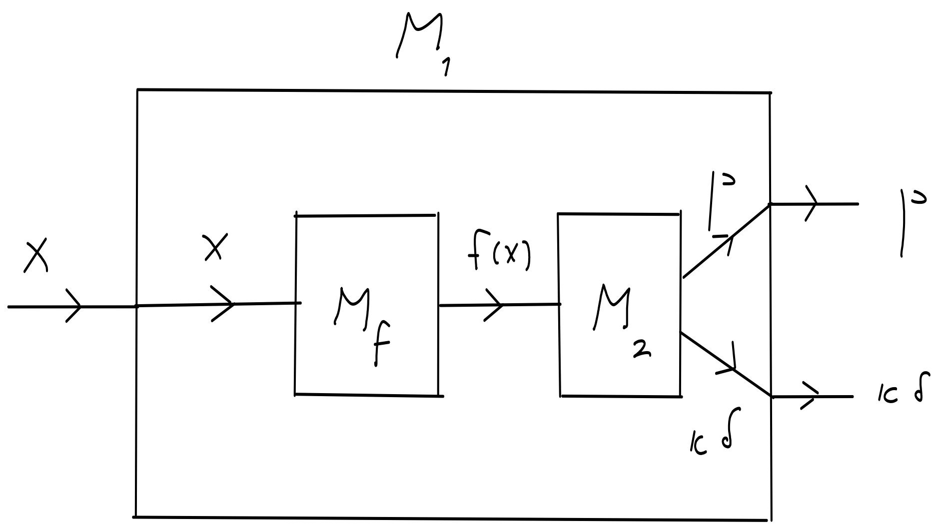
נוכיח כי M_1 מכריעה את L_1.

- אם $x \in L_1 \quad \Rightarrow \quad f(x) \in L_2 \quad \Rightarrow \quad M_2$ מקבלת את $f(x) \quad \Rightarrow \quad M_1$ 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 $L_{\text{acc}} \leqslant L_{\text{halt}}$. 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:

$L_{\Sigma^*} \notin R$

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.

<u>אבחנה:</u>

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

<u>נכונות הרדוקציה:</u>

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

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

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:

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

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

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

דוגמה 8.8

נתונה השפה

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

ונתונה השפה

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

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

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

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

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

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

כאשר M^* המ"ט שמקבלת כל קלט.

<u>נכונות הרדוקציה:</u>

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

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

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:

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

<u>מקרה 2):</u> $x=\langle M\rangle$ 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 ממשפט 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
$$L_{\text{NOTREG}} = \left\{\langle M\rangle \;\middle|\; L(M) \text{ לא רגולרית}\right\}\ .$$
הוכיחו כי השפה L_{NOTREG} לא כריעה על ידי רדוקציה מ- $\bar{L}_{\text{acc}}$.

פתרון:

השפה $\bar{L}_{\text{acc}}$ מוגדרת
$$\bar{L}_{\text{acc}} = \left\{\langle M,w\rangle \;\middle|\; M \text{ לא מקבלת } w\right\} \cup \{x \neq \langle M,w\rangle\}\ .$$
והשפה L_{NOTREG} מוגדרת
$$L_{\text{NOTREG}} = \left\{\langle M\rangle \;\middle|\; L(M) \text{ לא רגולרית}\right\}\ .$$
נגדיר הפונקציה הבאה:
$$f(x) = \begin{cases} \langle M'\rangle & x = \langle M,w\rangle \\ w' \in PAL & x \neq \langle M,w\rangle \end{cases}$$
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 $x \in \bar{L}_{\text{acc}} \quad \Leftarrow \quad$ 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 $x \notin \bar{L}_{\text{acc}} \quad \Leftarrow \quad M$ מקבלת $w \quad \Leftarrow \quad L(M') = \Sigma^* \quad \Leftarrow \quad f(x) \in \Sigma^* \quad \Leftarrow \quad f(x) \in L_{\text{NOTREG}}$.

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 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_{HALT}.

פתרון:

השפה L_{HALT} מוגדרת

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

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 $\Leftarrow$ ממשיכה לשלב 3).

3) M' בודקת אם $y \in PAL$.
- אם כן $\Leftarrow$ מקבלת.
- אם לא $\Leftarrow$ דוחה.

<u>הוכחת הנכונות</u>

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

$x \notin L_{\text{HALT}}$ $\Leftarrow$ 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 עוצרת על w $\Leftarrow$ $L(M_\varnothing) = \varnothing$ $\Leftarrow$ $\langle M_\varnothing\rangle \notin L_{\text{NOTREG}}$
$\Leftarrow$ $f(x) \notin L_{\text{NOTREG}}$.

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

תהי L_{REG} השפה

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

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

פתרון:

השפה $\bar{L}_{\text{acc}}$ מוגדרת

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

והשפה L_{REG} מוגדרת

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

נגדיר הפונקציה הבאה:

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

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.

<u>אבחנה</u>

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

<u>הוכחת נכונות הרדוקציה</u>

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

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

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

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

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.

הוכחת נכונות הרדוקציה

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

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

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

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

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

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- $\bar{L}_{\mathrm{acc}}$.

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

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

אם $x \in \bar{L}_{\mathrm{acc}}$ $\Leftarrow$ 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 $x \notin \bar{L}_{\mathrm{acc}}$ $\Leftarrow$ $x = \langle M, w \rangle$ 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 $\bar{L}_{\mathrm{acc}} \leqslant L_{M_1 \neg M_2}$, ומכיוון ש- $\bar{L}_{\mathrm{acc}} \notin RE$ ממשפט הרדוקציה מתקיים $L_{M_1 \neg M_2} \notin RE$.

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

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

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

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

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

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 זמן

9.1 הגדרה של סיבוכיות זמן

עד כה כל הבעיות החישוביות שעסקנו בהן הניחו שהמשאבים שעומדים לרשות מכונת הטיורינג שפותרת אותן הם **בלתי מוגבלים**. כעת נעבור לעסוק בשאלה מה קורה כאשר אנחנו מגבילים חלק ממשאבים אלו. יש סוגים רבים של משאבים שניתן לעסוק בהם, אבל שני הנפוצים ביותר בתיאוריה של מדעי המחשב הם

- **זמן החישוב**,
- **הזיכרון שנדרש לצורך החישוב**.

אחת מהבעיות שבהן נתקלים:
כשמעוניינים למדוד את צריכת המשאבים הללו של אלגוריתם מסויים היא שלא ברור כיצד למדוד אותם.
האם זמן חישוב נמדד בשניות?
אם כן, כיצד ניתן לחשב את זמן החישוב עבור אלגוריתם נתון?
האם עלינו לקודד ולהריץ אותו על מחשב מסוים?

אבל במחשבים שונים האלגוריתם ירוץ זמנים שונים בשל

- יעילות המעבד,
- אופטימיזציות שמתבצעות ברמת המעבד,
- אופטימיזציות בזמן הקומפליצה,

וכיוצא בהן.

אפילו תנאים חיצוניים כמו החום בסביבת המעבד עשויים להשפיע על זמן הריצה. מכאן הרצון למצוא הגדרה **תיאורטית** של זמן ריצה, שאינה תלויה בחומרה זו או אחרת.

הערה 9.1

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

הגדרה 9.1 זמן הריצה של מכונת טיורינג

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

הגדרה 9.2 סיבוכיות זמן של בעיה/שפה

בהינתן קלט w באורך $n=|w|$. 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 $f\left(|w|\right)$.

דוגמה 9.1 סיבוכיות זמן של השפה של מחרוזות האוניריות

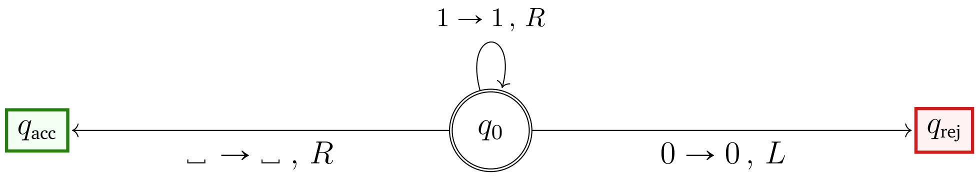
נתבונן על השפה של מחרוזות האוניריות הבאה:

$$L = \left\{1^n \;\middle|\; n \geqslant 0\right\} \ .$$

נבנה מכונת טיורינג הבאה שמכריעה אותה:

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

כאשר $\Sigma = \{0,1\}$ ו- $\Gamma = \{0,1,\sqcup\}$. המצבים והמעברים מתוארים בהתרשים מצבים שלמטה:



בכדי לחשב את הסיבוכיות זמן של L נתאר את המכונה בפסאודוקוד.
$M =$ על כל קלט w:

(1
- אם המילה היא מילת הריקה תקבל.
- אחרת ממשיכה לשלב 2).

(2
- אם התו הנקרא 0 תדחה.
- אחרת אם התו הנקרא הוא $\sqcup$ תקבל.
- אחרת חוזרת לשלב 2).

ככל שהקלט ארוך יותר, כך M תבצע צעדי חישוב רבים יותר.
בפרט המספר הצעדים המקסימלי הוא במקרה שבקלט w יש רק תווי 1 ולכן המ"ט תבצעת $n = |w|$ צעדי חישוב. בכל מקרה אחר היא תבצעת פחות מ- n צעדים.

$\Leftarrow$ חסם העליון של מספר צעדי חישוב של M על קלט w הוא n כאשר $n = |w|$.

$\Leftarrow$ M עוצרת בזמן $O(n)$

$\Leftarrow$ L כריעה בזמן $O(n)$.

$\Leftarrow$ $L \in TIME(n)$.

באופן כללי, אם מכונה על קלט w מבצעת פחות מ־ $|w|$ צעדי חישוב, המשמעות היא המכונה כלל לא קראה את כל הקלט, וזה אינו מקרה נפוץ במיוחד (אם כי הוא בהחלט קיים). אם כן ברור שמדידת זמן הריצה היא תמיד **ביחס לאורך הקלט**.

דוגמה 9.2

בדוגמה הזו נבנה מכונת טיורינג M עם סרט יחיד שמכריעה את השפה $L = \left\{a^n b^n \;\middle|\; n \geqslant 0\right\}$ ואז נחשב את הסיבוכיות זמן שלה.

<u>**בניית המכונת טיורינג**</u>

$M =$ "על קלט w:

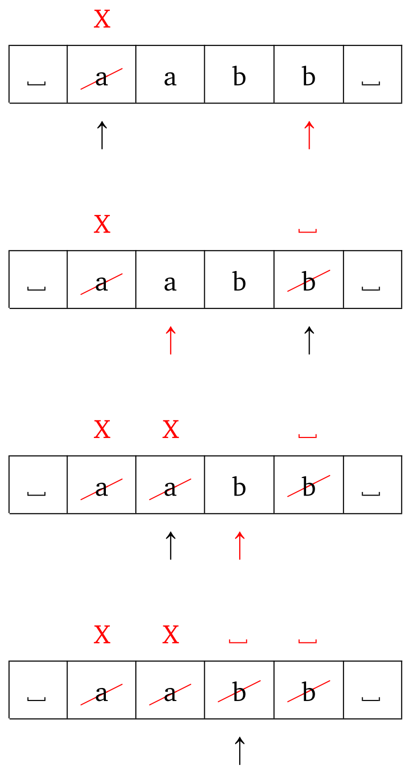
(1) אם התו הנקרא הוא $\sqcup$ $\Leftarrow$ M תקבלת.

(2) אחרת אם התו הנקרא הוא $b \Leftarrow M$ דוחה.

(3) אחרת מוחקת את התו הנקרא ע"י X.

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

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



זמן הריצה

נסמן את אורך הקלט ב- $n = |w|$.

- M מבצעת $\left\lfloor \frac{n}{2} \right\rfloor$ איטרציות.
- בכל איטרציה M סורקת את הסרט פעמיים וזה עולה $O(n)$ צעדים.
- לכן סה"כ הזמן הריצה של M חסום ע"י

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

דוגמה 9.3

כעת נבנה מכונת טיורינג מרובת סרטים M' שמכריעה את השפה $L = \left\{a^n b^n \,\middle|\, n \geqslant 0\right\}$ ונראה שהסיבוכיות זמן שונה מזה של המכונת טיורינג עם סרט יחיד שראינו בדוגמה הקודמת.

בניית המכונה

$M' =$ "על קלט w:

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

(2) מזיזה את הראשים לתחילת הסרטים. $O(n)$.

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

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

(5) מזיזה את שני הראשים ימינה וחוזרת לשלב (3)." שלבים (3-5): $O(n)$.

סרט (1)

␣	a	a	b	b	␣

סרט (2)

␣	b	b			␣

סיבוכיות זמן

נסמן את אורך הקלט ב- $n = |w|$.

הזמן הריצה של M' הוא $O(n)$.

הגדרה 9.3 $TIME(f(n))$

נגדיר הקבוצת השפות $TIME(f(n))$ כך שלכל שפה $L \in TIME(f(n))$ קיימת מכונת טיורינג דטרמיניסטית שמכריעה את L בזמן לכל היותר $f(n)$, כאשר n הוא האורך של הקלט:

$$TIME(f(n)) = \{L \mid \text{קיימת מ"ט דטרמיניסטית המכריעה } L \text{ בזמן } O(f(n))\} .$$

דוגמה 9.4

עבור השפה בדוגמה 9.2:

$$L \in TIME(n^2) .$$

דוגמה 9.5

עבור השפה בדוגמה 9.3:

$$L \in TIME(n) .$$

דוגמה 9.6

הדוגמה הזו ראינו בדוגמה 1.5. נבנה מכונת טיורינג M עם סרט יחיד שמכריעה את השפה $L = \{a^{2^n} \mid n \geqslant 0\}$ ואז נחשב את הסיבוכיות זמן שלה.

הרעיון

המכונה תסרוק את הסרט שלה משמאל לימין ובכל איטרציה תמחק חצי מה- a -ים שנשארו.

בניית המכונה

$M=$ " על קלט w:

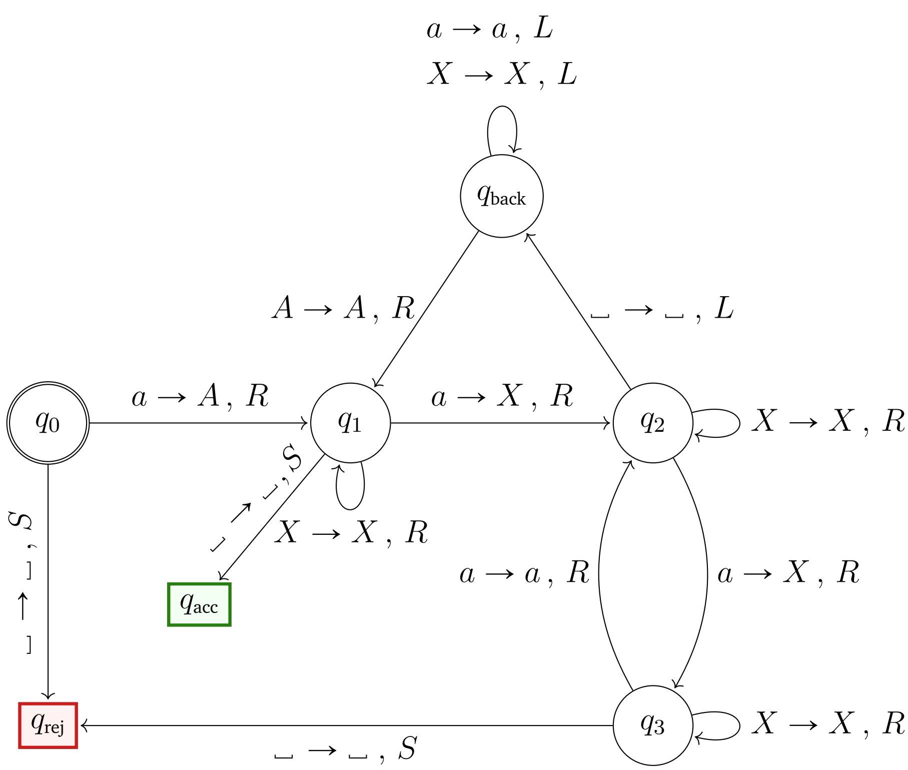
(1) אם התו הנקרא הוא $\sqcup$ $\Leftarrow$ דוחה.

(2) אחרת מחליפה את ה- a הראשון ב- A (כדי לסמן את תחילת הסרט).

(3) סורקת את הסרט משמאל לימין ומוחקת כל a שני:

- אם הסרט מכיל a יחיד $\Leftarrow$ מקבלת.
- אם הסרט מכיל מספר אי-זוגי של a -ים $\Leftarrow$ דוחה.
- מזיזה את הראש שמאלה לתחילת הסרט וחוזרת לשלב (3)."

התרשים מצבים של המכונה מתואר באיור למטה:



<u>סיבוכיות זמן</u>

- שלב (3) דורש n צעדים לכל היותר.
- המכונה חוזרת על שלב (3) לכל היותר n פעמים.
- לפיכך הסיבוכיות זמן של השפה L היא $O\left(n^2\right)$. כלומר:

$$L \in TIME\left(n^2\right)\ .$$

דוגמה 9.7 חיסור בינארי

בדוגמה זו נבנה מכונת טיורינג מ- 3 סרטים שמקבלת כקלט שני מספרים שלמים x,y $(x>y)$ בבסיס בינארי ומחשבת את החיסור $x-y$.

<u>בניית המכונה</u>

$SUBTRACT =$ " על קלט $\langle x,y\rangle$ כאשר x,y שלמים בבסיס בינארי ו- $x>y$:

1) רושמת את x בסרט 1 ואת y בסרט 2 משמאל לימין כך שהתאים עם הקצוות הימניות של x ו-y מיושרים זה מתחת לזה. בתחילה סרט 3 ריק.

2) מעמידה את ראשי סרטים $1,2,3$ על התאים המיושרים של הקצוות הימניים של הקלטים. בפרט, ראשי סרטים 1 ו-2 נמצאים על הספרות הפחות משמעותיות של x ו-y, וראש סרט 3 נמצא בתא שמתחתם.

3) שלב זה מבצע חיסור ללא חוב.

יהי $\sigma_1\in\{0,1,\sqcup\}$ תו הנקרא בסרט 1 ויהי $\sigma_2\in\{0,1,\sqcup\}$ תו הנקרא בסרט 2.

- אם $(\sigma_1,\sigma_2)=(\sqcup,\sqcup)$ אז M מקבלת.
- אם $(\sigma_1,\sigma_2)=(0,0)$, כותבת $(0,0,0)$, מזיזה את הראשים (L,L,L) וחוזרת על שלב זה.
- אם $(\sigma_1,\sigma_2)=(1,0)$, כותבת $(1,0,1)$, מזיזה את הראשים (L,L,L) וחוזרת על שלב זה.
- אם $(\sigma_1,\sigma_2)=(1,1)$, כותבת $(1,1,0)$, מזיזה את הראשים (L,L,L) וחוזרת על שלב זה.
- אם $(\sigma_1,\sigma_2)=(\sqcup,0)$, כותבת $(\sqcup,0,0)$, מזיזה את הראשים (S,L,L) וחוזרת על שלב זה.
- אם $(\sigma_1,\sigma_2)=(\sqcup,1)$, כותבת $(\sqcup,1,1)$, מזיזה את הראשים (S,L,L) ועוברת לשלב הבא.
- אם $(\sigma_1,\sigma_2)=(0,\sqcup)$, כותבת $(0,\sqcup,0)$, מזיזה את הראשים (L,S,L) וחוזרת על שלב זה.
- אם $(\sigma_1,\sigma_2)=(1,\sqcup)$, כותבת $(1,\sqcup,1)$, מזיזה את הראשים (L,S,L) וחוזרת על שלב זה.
- אם $(\sigma_1,\sigma_2)=(0,1)$, כותבת $(0,1,1)$, מזיזה את הראשים (L,L,L) ועוברת לשלב הבא.

4) שלב זה מבצע חיסור כאשר קיים חוב.

- אם $(\sigma_1,\sigma_2)=(1,0)$, כותבת $(1,0,0)$, מזיזה את הראשים (L,L,L) ועוברת לשלב הקודם.
- אם $(\sigma_1,\sigma_2)=(0,1)$, כותבת $(0,1,1)$, מזיזה את הראשים (L,L,L) וחוזרת על שלב זה.
- אם $(\sigma_1,\sigma_2)=(1,1)$, כותבת $(1,1,1)$, מזיזה את הראשים (L,L,L) וחוזרת על שלב זה.
- אם $(\sigma_1,\sigma_2)=(0,0)$, כותבת $(0,0,1)$, מזיזה את הראשים (L,L,L) וחוזרת על שלב זה.
- אם $(\sigma_1,\sigma_2)=(\sqcup,0)$, כותבת $(\sqcup,0,1)$, מזיזה את הראשים (S,L,L) וחוזרת על שלב זה.
- אם $(\sigma_1,\sigma_2)=(\sqcup,1)$, כותבת $(\sqcup,1,0)$, מזיזה את הראשים (S,L,L) וחוזרת על שלב זה.
- אם $(\sigma_1,\sigma_2)=(0,\sqcup)$, כותבת $(0,\sqcup,1)$, מזיזה את הראשים (L,S,L) וחוזרת על שלב זה.
- אם $(\sigma_1,\sigma_2)=(1,\sqcup)$, כותבת $(1,\sqcup,1)$, מזיזה את הראשים (L,S,L) ועוברת לשלב הקודם."

סיבוכיות זמן

יהי $n=\max(|x|,|y|)$. המכונה $SUBTRACT$ סורקת את השני סרטים שבהם כתובים המספרים x ו- y במקביל, ותוך כדי רושמת את הפלט על סרט 3. לכן $SUBTRACT$ מבצעת $(O(n))$ צעדים.

לכן הסיבוכיות זמן של $SUBTRACT$ היא $O(n)$ (ליניארי).

דוגמה 9.8 האלגוריתם החילוק של אוקלידס

המשפט החילוק של אוקלידס אומר שבהינתן שני מספרים שלמים x, y, אזי קיימים שלמים q, r כך ש

$$x = qy + r \ , \qquad 0 \leqslant r < |y| \ .$$

במקרה הפרטי ש- $0 < y < x$ אז:

$$q = \left\lfloor \frac{x}{y} \right\rfloor \ , \qquad r = x \bmod y \ .$$

קיים אלגוריתם שמקבל כקלט x, y ונותן כפלט q ו- r. האלגוריתם עובד לכל שלמים x, y (בלי קשר לסימן שלהם). אנחנו נסתכל על האלגוריתם במקרה הפרטי ש- $0 < y < x$, כמתואר למטה.

<u>בניית האלגוריתם</u>

$DIVISION =$ " על קלט $\langle x, y\rangle$ כאשר x, y שלמים בבסיס בינארי ו- $0 < y < x$:

1) מאתחל $\left.\begin{matrix} q \leftarrow 0 \\ r \leftarrow x \end{matrix}\right\}$

2) כל עוד ש $r \geqslant y$:

3) $r \leftarrow r - y$

4) $q \leftarrow q + 1$

5) פלט: q, r."

<u>סיבוכיות זמן</u>

נסמן $n = |\langle x, y\rangle|$ אורך הקלט.

- $DIVISION$ מבצעת r איטרציות לכל היותר.
- $r < y$ לכן y הוא חסם עליון של המספר האיטרציות המקסימלי של $DIVISION$.
- לכן $DIVISION$ מבצע $O(n)$ איטרציות לכל היותר.
- בכל איטרציה $DIVISION$ מבצע חיסור בינארי עם $SUBTRACT$ אשר (כפי שראינו בדוגמה 9.7) הוא $O(n)$.
- לכן הסיבוכיות זמן של $DIVISION$ היא

$$O\left(n^2\right) \ .$$

9.2 יחס בין הסיבוכיות של מכונת טיורינג עם סרט יחיד ומכונת טיורינג מרובת סרטים

משפט 9.1

לכל מכונת טיורינג מרובת סרטים M הרצה בזמן $t(n)$ קיימת מכונת טיורינג סרט יחיד M' השקולה ל- M ורצה בזמן $O\left(t^2(n)\right)$.

הוכחה:

תהי M מכונת טיורינג כלשהי עם k סרטים הרצה בזמן $O\left(t(n)\right)$.

נבנה מכונת טיורינג S עם סרט אחד שרצה בזמן $O\left(t^2(n)\right)$.

- ראשית נרשום את התוכן של ה- k סרטים של M על הסרט היחיד של S.
- התכנים של הסרטים מופרדים על ידי תו # על הסרט היחיד.
- המיקומים של הראשים של כל הסרטים של M מסומנים על הסרט היחיד על ידי חצים.

להלן יש דוגמה לכך של מכונת טיורינג עם 3 סרטים:

<u>המכונת טיורינג M</u>

0	1	0	1	0	␣	␣
			↑			

a	a	a	␣	␣	␣	␣
		↑				

a	b	b	␣	␣	␣	␣
↑						

<u>המכונת טיורינג S</u>

#	0	1	0	1 ↑	0	#	a	a	a ↑	#	a	b ↑	b

אפשר לסמלץ צעד חישוב אחד של M במכונת טיורינג S באופן הבא:

<u>**בניית המכונה S**</u>

1) תחילה S מאתחלת את הסרט שלה בלכתוב את התכנים של ה- k סרטים על הסרט היחיד שלה, עם תו # להפריד בין סרטים שונים של M.

2) בנוסף S רושמת תו # בקצה השמאלי כדי לסמן את התחילת הקלט ותו # בקצה הימין כדי לסמן את סוף הקלט.

3) בכדי לסמלץ צעד חישוב אחד של M בהמכונה S, המכונה S סורקת את הסרט מ- # הראשון ל- # האחרון. בסריקה זו S זוכרת את המיקומים של ה- k ראשים על פי התאים שמסומנים עם חצים, באמצעות k תאי זכרון.

4) אחר כך S מבצעת סריקה שנייה של הסרט. בסריקה זו, לפי הפונקצית המעברים של M, המכונה S מבצעת, לכל $1 \leqslant i \leqslant k$:

- כתיבה של הסימן החדש בסרט ה-i במיקום הנוכחי של הראש של סרט ה-i,
- תזוזה של הראש של סרט ה- i.

5) במצב שהראש של אחד הסרטים של M זז ימינה לתו רווח מצד ימין של סוף הקלט, אז S תוסיף תא אחד עם תו רווח מצד שמאל של ה- # המפריד בין סרטים, ואחר כך היא תזיז את כל התאים שבצד ימין של התא המוסף מקום אחד ימינה.

סיבוכיות זמן של המכונה S

יהי n האורך של התוכן הכי ארוך מתוך כל התכנים של ה- k סרטים של M.

- האורך של התוכן של S שווה לסכום הארכים של התכנים של ה- k סרטים של M.
- נתון שהזמן הריצה של M הוא $O(t(n))$.

$\Leftarrow$ M עושה שימוש של $O(t(n))$ תאים.

$\Leftarrow$ אורך הקלט של S חסום מלמעלה ע"י $O(t(n))$.

$\Leftarrow$ סריקה שלמה של הקלט של S מתבצעת ב- $O(t(n))$ צעדים.

$\Leftarrow$ מכיוון שכל סריקה דורשת זמן $O(t(n))$ וכדי לדמות צעד אחד של M, המכונה S מבצעת שתי סריקות, אז S לוקחת זמן $O(t(n))$ לבצע צעד חישוב אחד של M.

$\Leftarrow$ הסיבוכיות זמן של S היא:

$$O(t(n))\ \ O(t(n)) = O\left(t^2(n)\right)$$

■

9.3 יחס בין הסיבוכיות של מכונת טיורינג דטרמיניסטית ומכונת טיורינג אי דטרמיניסטית

משפט 9.2

לכל מכונת טיורינג אי-דטרמיניסטית N הרצה בזמן $f(n)$, קיימת מכונת טיורינג דטרמיניסטית D השקולה ל- N ורצה בזמן $2^{(f(n))}$.

הוכחה:

בהינתן מכונת טיורינג אי-דטרמיניסטית N הרצה בזמן $f(n)$ מכונת טיורינג דטרמיניסטית D באותו אופן כמו בהוכחת השקילות במשפט 4.1.

כלומר, בהינתן קלט w, D תסרוק 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 עבור $c > 0$ כלשהו.

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

9.4 המחלקה P

הגדרה 9.4 בעיית הכרעה

בעיית הכרעה מוגדרת באופן הבא:

"בהינתן קלט כלשהו, האם הקלט מקיים תנאי מסוים"

דוגמה 9.9

בהינתן מספר n, האם n ראשוני?

כל בעיית הכרעה ניתן לתאר כשפה שקולה:

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

משפט 9.3

שפה $\equiv$ בעיית הכרעה .

הגדרה 9.5 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 מכונת טיורינג דטרמיניסטית המכריעה את השפה השקולה לבעייה זו בזמן פולינומיאלי.

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

הגדרה 9.6 המחלקה P

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

דוגמה 9.10

בדוגמה 9.2 הוכחנו כי קיימת מכונת טיורינג דטרמייססטית שמכריעה עת השפה

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

בזמן $O\left(n^2\right)$. לכן $L \in P$.

דוגמה 9.11

תהי L_{prime} השפה הבאה:

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

הוכיחו: $L_{\text{prime}} \in P$

פתרון:

כדי להוכיח כי $L_{\text{prime}} \in P$, נבנה אלגוריתם שמכריע את L_{prime} ואחר כך נראה כי האלגוריתם רץ בזמן פולינומיאלי.

האלגוריתם עצמו עושה שימוש של המשפט הבא:

<u>משפט: למספר פריק יש מחלק הקטן מ- או שווה לשורשו</u>

יהי n מספר שלם. אם n פריק אז קיים שלם $\sqrt{n} \leqslant a$ שמחלק את n.

<u>הוכחה:</u>

- נניח בשלילה כי n פריק ולא קיים מחלק d של n כך ש- $d \leqslant \sqrt{n}$.
- n פריק אז קיימים שלמים ab כך ש- $n = ab$, כאשר $1 < a, b < n$.
- נניח כי $1 < a \leqslant b < n$.
- על פי ההנחה שלנו, אם a מחלק n אז $a > \sqrt{n}$. בנוסף: $b \geqslant a > \sqrt{n}$.
- לכן

$$n = ab > \left(\sqrt{n}\right)\left(\sqrt{n}\right) = n \ ,$$

ז"א $n > n$ וזו סתירה!

■

על סמך המשפט הזה נבנה אלגוריתם $PRIME$ המכריע את L_{prime} בזמן פולינומיאלי. הרעיון של האלגוריתם הוא, בהינתן קלט n, לבדוק אם קיים מחלק של n אשר קטן מ- או שווה ל- $\sqrt{n}$. אם כן אז n פריק ואם לא אז n ראשוני.

בניית האלגוריתם

$PRIME =$ על קלט x:

1) בודק אם $x = \langle n \rangle$ ו- n מספר שלם חיובי.

- אם לא $\Leftarrow$ דוחה.

2) אם $n = 1$ $\Leftarrow$ דוחה.

3) אם $n = 2$ $\Leftarrow$ מקבל.

4) אחרת לכל $3 \leqslant d \leqslant \lfloor \sqrt{n} \rfloor$:

- מחשב $n \bmod d$.
- $n \bmod d = 0$ $\Leftarrow$ מקבל.

5) מקבל.

סיבוכיות זמן

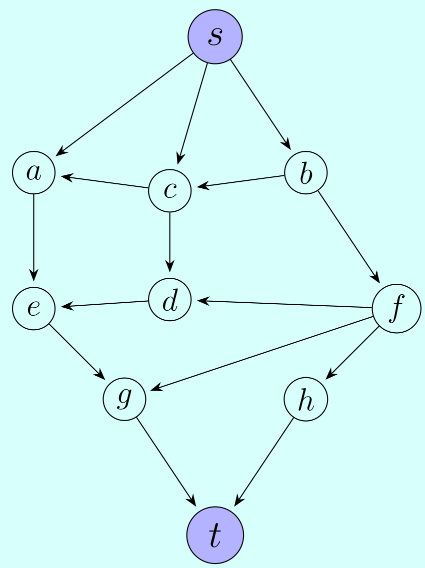
- $PRIME$ מבצע $O(n)$ איטרציות.
- בכל איטרציה $PRIME$ מחשב $n \bmod d$ באמצעות האלגוריתם $DIVISION$, אשר הוכחנו בדוגמה 9.8 שהוא רץ בזמן $O\left(n^2\right)$.
- לכן הסיבוכיות זמן של $PRIME$ היא $O\left(n^3\right)$

לפיכך:

$$L_{\text{prime}} \in P \ .$$

9.5 בעיית PATH

הגדרה 9.7 בעיית המסלול בגרף מכוון



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

<u>פלט:</u> האם קיים מסלול ב- G מ- s ל- t?

$$PATH=\{\langle G,s,t\rangle \mid \text{קיים מסלול ב- } G \text{ מ- } s \text{ ל- } t\}$$

משפט 9.5

$$PATH\in P\ .$$

הוכחה: נבנה אלגוריתם A שמכריע את $PATH$ בזמן פולינומיאלי.

<u>בניית האלגוריתם</u>

$A=$ על קלט $\langle G,s,t\rangle$:

(1 צובע את s.

(2 מבצע $|V|-1$ פעמים:

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

(3
- אם t צבוע $\Leftarrow$ מקבל.
- אחרת $\Leftarrow$ דוחה.

<u>הוכחת הנכונות</u>

<u>הוכחה של הכיוון $\Leftarrow$</u>

אם $x \in PATH$:

$\Leftarrow$ $x = \langle G, s, t\rangle$ ו- $G = (V, E)$ גרף מכוון וגם $s, t \in V$ כך שקיים מסלול מ- s ל- t.

$\Leftarrow$ קיים קבוצת צלעות

$$C = \{u_1u_2,\ u_2u_3,\ \ldots\ ,\ u_{k-2}u_{k-1}\ ,\ u_{k-1}u_k\} \subseteq E\ ,$$

כאשר $u_1 = s\ ,\ u_k = t$ ו- $k \leqslant |V|$.

$\Leftarrow$ לכל $1 \leqslant i \leqslant |V| - 1$, באיטרציה i הקודקוד u_{i+1} יצובע.

$\Leftarrow$ מכיוון ש- $k \leqslant |V| - 1$ אזי עד סוף הלולאה הקודקוד $u_k = t$ יצובע.

$\Leftarrow$ מכיוון שהקודקוד t צובע אז A מקבל.

הוכחה של הכיוון $\Rightarrow$

$x \notin PATH \Leftarrow$ 2 מקרים:

מקרה 1) $x \neq \langle G, s, t\rangle$ ואז A דוחה.

מקרה 2) אחרת $x = \langle G, s, t\rangle$ כאשר $G = (V, E)$ גרף מכוון וגם $s, t \in V$ כך שלא קיים מסלול מ- s ל- t.

$\Leftarrow$ לכל קבוצה

$$C = \Big\{su_2,\ u_1u_2,\ \ldots\ ,\ u_{k-2}u_{k-1}\ ,\ u_kt \Big| \ \{u_i\} \in V\ , 2 \leqslant k \leqslant |V| - 1\Big\}\ ,$$

קיים j כך ש- u_ju_{j+1} לא צלע של G.

$\Leftarrow$ עבור כל מסלול של הקבוצה C, הקודקודים $u_{j+1}, \ldots, u_k, t$ לא יהיו צבועים.

$\Leftarrow$ מכיוון ש- t לא צבוע אז A דוחה.

סיבוכיות זמן

- נסמן אורך הקלט $N = \langle G, s, t\rangle$.
- נסמן $n = |V|$ ו- $m = |E|$.
- A מבצע לולאה לכל $1 \leqslant i \leqslant |V| - 1$ שדורש $O(|V|)$ צעדים.
- בתוך הלולאה A מבצע לולאה מעל צלעות עבורן הנקודה ההתחלתית צובעת. $\Leftarrow$ שלב זה דורש $O(|E|)$ צעדים.
- סבה"כ הסיבוכיות זמן של A היא $O(|E| \cdot |V|) = O(nm)$.
- בנסוף: $N \leqslant n$ וגם $N \leqslant m$ לכן A דורש N^2 צעדי חישוב לכל היותר.
- לכן A רץ בזמן $O\left(N^2\right)$

מצאנו אלגוריתם שמכריע את $PATH$ בזמן $O\left(N^2\right)$ $\Leftarrow$ $PATH \in TIME\left(N^2\right)$ $\Leftarrow$ $PATH \in P$. ■

9.6 הבעיית $RELPRIME$

הגדרה 9.8 מספרים זרים (Relatively prime)

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

הגדרה 9.9 בעיית $RELPRIME$

קלט: שני מספרים x ו- y.

פלט: האם x ו- y זרים?

$$RELPRIME = \{\langle x, y\rangle \mid \gcd(x, y) = 1\} \ .$$

אנחנו נוכיח כי ניתן להכריע את $RELPRIME$ בזמן פולינומיאלי, כלומר נוכיח $RELPRIME \in P$ במשפט 9.8 למטה. לפניכן נסביר את האלגוריתם של אוקלידס למציאת ה- gcd של שני שלמים, ומתוך זה נוכל לחשב את הסיבוכיות זמן של $RELPRIME$. ראשית נזכיר משפט שלמדנו בקורסים קודמים:

משפט 9.6 השמפט של האלגוריתם של אוקלידס

אם x, y שלמים אז

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

הוכחה: ההוכחה היא לא חלק של הקורס ומופיע בסעיף האחרון "הוכחות של משפטים שימושיים" בדף 110. ■

האלגוריתם של אוקלידס הוא אלגוריתם, שמקבל כקלט שני מספרים x, y ופולט את $\gcd(x, y)$. הוא מתבוסס על המשפט 9.6. האלגוריתם עצמו הוא כדלקמן:

$EUCLID =$ על קלט x, y:

(1) כל עוד $y \neq 0$:

(2) $x \leftarrow x \bmod y$

(3) $\text{swap}(x, y)$

(כלומר מחליפים בין x ו- y).

(4) מחזירים את x.

לדוגמה:

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

כדי להוכיח כי $RELPRIME \in P$ נצטרך למשפט עזר הבא:

משפט 9.7 (משפט עזר)

אם $x > y$ אז $x \bmod y < \frac{x}{2}$.

הוכחה: ההוכחה היא לא חלק של הקורס ומופיע בסעיף האחרון "הוכחות של משפטים שימושיים" בדף 111. ■

משפט 9.8

$$RELPRIME \in P \ .$$

הוכחה:
נבנה אלגוריתם A המכריע את $RELPRIME$ בזמן פולינומיאלי. $RELPRIME$ היא השפה של הבעיה, שמקבלת כקלט שני מספרים שלמים x,y ומחזירה תשובה לשאלה, האם x,y זרים. כלומר:

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

לכן A משתמש בהאלגוריתם של אוקלידס $EUCLID(x,y)$ כדי לחשב $\gcd(x,y)$.

בניית האלגוריתם A המכריע $RELPRIME$:

$A=$ " על קלט $\langle x,y\rangle$: A מריץ את $EUCLID$ על x ו- y.

- אם $EUCLID(x,y)$ מחזיר $\gcd(x,y)=1$ אז A מקבל.
- אחרת A דוחה."

הוכחת הנכונות

הנכונות של A מנובעת ישר מהנכונות של האלגוריתם האוקלדס, $EUCLID$.

סיבוכיות זמן

נראה כי A רץ בזמן פולינומיאלי בגודל הקלט $\langle x,y\rangle$. נסמן את אורך הקלט $n=|\langle x,y\rangle|$.

- לפי משפט עזר 9.7: $x \bmod y < \dfrac{x}{2}$.
- בכל איטצרציה, בשלב (2) המשתנה x מקבל את הערך החדש $x\leftarrow x \bmod y$.
  ניתן לחשב את $x \bmod y$ בעזרת האלגוריתם החילוק של אוקלידס $DIVISION$ שרץ בזמן $O\left(n^2\right)$ (ראו דוגמה 9.8).
- לכן בכל איטצרציה הערך החדש של x קטן ממש מחצי של הערך הקודם של x.
- לכן אחרי כל איטרציה, x קטן בלפחות חצי.
- בשלב (3), A מחליף בין x ו- y, אז אחרי כל 2 איטרציות, גם x קטן בלפחות חצי וגם y קטן בלפחות חצי.
- לכן המספר הפעמים המקסימלי שאפשר לבצע שלבים (2) ו- (3) היא $m \triangleq \min\left(\ 2\lfloor \log_2 x\rfloor \ ,\ 2\lfloor \log_2 y\rfloor\ \right)$.
- לכן m הוא חסם עליון של מספר האיטרציות ש $EUCLID$ מבצע.
- $m\leqslant n$ כאשר n הוא האורך של הקלט $\langle x,y\rangle$ לכן $EUCLID$ מבצע $O(n)$ איטרציות.
- כל איטרציה מבצעת $DIVISION$ כדי לחשב $x \bmod y$ אשר רץ בזמן $O\left(n^2\right)$.

  $\Leftarrow$ A רץ בזמן $O\left(n^3\right)$. כלומר:

$$RELPRIME \in TIME\left(n^3\right) \ .$$

לכן:

$$RELPRIME \in P \ .$$

■

9.7 *הוכחות של משפטים שימושיים

משפט 9.9 השמפט של האלגוריתם של אוקלידס

אם x, y שלמים אז

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

הוכחה: (להעשרה בלבד)

המטרה של ההוכחה הזו היא רק להוסיף הבנה להוכחה של משפט 9.8 לסיבוכיות זמן של $RELPRIME$ למטה.
היא לא הוכחה שאתם תיבחנו עליה ואפשר לדלג עליה.
נתחיל אם משפט החילוק של אוקלידס, שאומר שאם x, y שלמים אז קיימים שלמים q ו- $0 \leqslant r < y$ כך ש:

$$x = qy + r = \left\lfloor \frac{x}{y} \right\rfloor y + (x \bmod y) \ . \qquad (1*)$$

נגדיר $d \triangleq \gcd(x, y)$.
מכיוון ש- d הוא מחלק משותף של x ו- y אז $d \mid x$ וגם $d \mid y$. לכן בזכות משוואה (1*):

$$(d \mid x) \wedge (d \mid y) \xRightarrow{\text{משוואה (1*)}} d \mid (x \bmod y)$$

ז"א $d \mid y$ וגם $d \mid (x \bmod y)$ אז בהכרח:

$$d \mid \gcd\left(y, x \bmod y\right) \ . \qquad (2*)$$

כעת נגדיר $\bar{d} \triangleq \gcd(y, y \bmod x)$.
מכיוון ש- $\bar{d}$ הוא מחלק משותף של y ו- $x \bmod y$ אז $\bar{d} \mid y$ וגם $\bar{d} \mid x \bmod y$. לכן בזכות משוואה (1*):

$$(\bar{d} \mid y) \wedge (\bar{d} \mid x \bmod y) \xRightarrow{\text{משוואה (1*)}} \bar{d} \mid x$$

ז"א $\bar{d} \mid y$ וגם $\bar{d} \mid x$ אז בהכרח:

$$\bar{d} \mid \gcd\left(x, y\right) \ . \qquad (3*)$$

לסיכום, לפי משוואות (2*) ו- (3*):

$$d \mid \bar{d} \quad \wedge \quad \bar{d} \mid d \ .$$

מכיוון ש- $d, \bar{d} > 0$ אז בהכרח $d = \bar{d}$, ז"א $\gcd(x,y) = \gcd(y, x \bmod y)$. ■

משפט 9.10 (משפט עזר)

אם $x > y$ אז $x \bmod y < \frac{x}{2}$.

הוכחה: יש שני מקרים:

1) $y \leqslant \frac{x}{2}$

2) $y > \frac{x}{2}$

נוכיח את הטענה עבור שני המקרים.

מקרה 1: $y \leqslant \dfrac{x}{2}$.

לפי משפט החילוק של אוקלידס אם x,y שלמים עבורם $x > y$ אז קיימים $q = \left\lfloor \dfrac{x}{y} \right\rfloor$ ו- $r = x \bmod y$ כך ש $0 \leqslant r < y$ ו-

$$x = qy + r = \left\lfloor \frac{x}{y} \right\rfloor y + (x \bmod y) \ .$$

בפרט $r < y$ וגם $y \leqslant \dfrac{x}{2}$ לפיכך $x \bmod y < y \leqslant \dfrac{x}{2}$.

מקרה 2: $y > \dfrac{x}{2}$.

לפי משפט החילוק של אוקלידס אם x,y שלמים עבורם $x > y$ אז קיימים שלם $q = \left\lfloor \dfrac{x}{y} \right\rfloor$ ושלם $r = x \bmod y$ כך ש $0 \leqslant r < y$ ו-

$$x = qy + r = \left\lfloor \frac{x}{y} \right\rfloor y + (x \bmod y) \ .$$

בפרט אם $y > \dfrac{x}{2}$ אז $x < 2y$. אז בהכרח $q < 2$. מכיוון ש- $x > y$ ו- $q = \left\lfloor \dfrac{x}{y} \right\rfloor$ אז הערך המינימלי של q הוא $q = 1$. לכן אם $q < 2$ בהכרח $q = 1$. לכן יש לנו

$$x = qy + r = (1)(y) + ry + (x \bmod y) \ .$$

מכאן

$$x - y = x \bmod y \ .$$

כעת נציב את ההנחה ההתחלתית $y > \dfrac{x}{2} \Leftarrow x - y < \dfrac{x}{2}$ ונקבל

$$x \bmod y < \frac{x}{2} \ .$$

■

שיעור 10
המחלקה P והמחלקה NP

10.1 המחלקה P

הגדרה 10.1 בעיית הכרעה

בעיית הכרעה מוגדרת באופן הבא:

"בהינתן קלט כלשהו, האם הקלט מקיים תנאי מסוים ? "

דוגמה 10.1 דוגמה של בעיית הכרעה

לדוגמה, בהינתן מספר n, האם n ראשוני?

משפט 10.1 שקיות בין בעייה לשפה

כל בעייה הכרעה ניתן לתאר כשפה שקולה:

$$\text{בעיית הכרעה} \equiv \text{שפה} \ .$$

דוגמה 10.2

לדוגמה, הבעיית הכרעה הבאה:

"בהינתן מספר n, האם n ראשוני? "

ניתנת לרשום כשפה הבאה:

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

הגדרה 10.2 אלגוריתם זמן פולינומיאלי

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

התזה של צרץ' טיורינג אומר שאם קיים אלגוריתם המכריע בעייה בזמן פולינומיאלי, אזי קיימת מכונת טיורינג דטרמיניסטית המכריעה את השפה השקולה לבעייה זו בזמן פולינומיאלי.

$$\text{אלגוריתם מכריע} \equiv \text{מכונת טיורינג דטרמיניסטית} \ .$$

הגדרה 10.3 המחלקה P

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

10.2 דוגמאות לבעיות ב- P

(1

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

(2

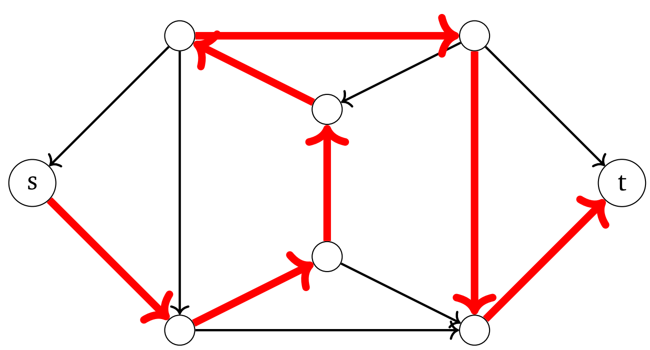
$$RELPRIME = \{\langle x, y\rangle \mid x \text{ ו- } y \text{ זרים}\} \in P$$

10.3 בעיית המסלול ההמילטוני $HAMPATH$

הגדרה 10.4 $HAMPATH$

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 10.5 בעיית 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?

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

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 10.6 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 $w \in A$ $\Leftarrow$ קיים y כך ש: $V(w,y)=T$.
- אם $w \notin A$ $\Leftarrow$ לכל y מתקיים $V(w,y)=F$.

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 10.7 המחלקה NP

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

משפט 10.2 $HAMPATH \in NP$

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?

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

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>

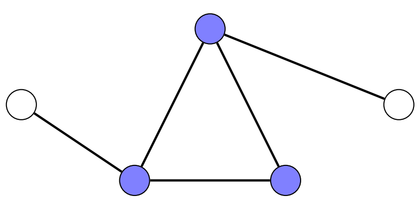
- זמן הריצה של האלגוריתם הוא פולינומיאלי בגודל הקלט.
- אם $\langle G, s, t\rangle \in HAMPATH$ $\Leftarrow$ G מכיל מסלול המילטוני מ- s ל- t $\Leftarrow$ עבור y שהוא קידוד של מסלול זה, V יקבל את הזוג $(\langle G, s, t\rangle, y)$.
- אם $\langle G, s, t\rangle \notin HAMPATH$ $\Leftarrow$ G לא מכיל מסלול המילטוני מ- s ל- t $\Leftarrow$ לכל y, האלגוריתם ידחה את הזוג $(\langle G, s, t\rangle, y)$.

■

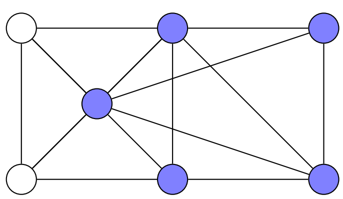
הגדרה 10.8 קליקה

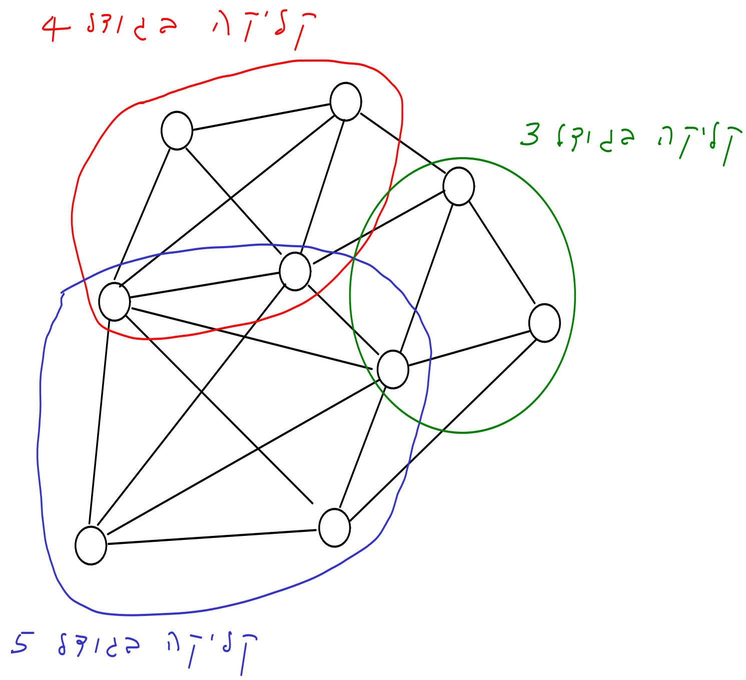
בהינתן גרף לא מכוון $G = (V, E)$, קליקה ב- G היא תת-קבוצה של קודקודים $C \subseteq V$ כך שלכל שני קודקודים $u, \mathrm{v} \in C$ מתקיים $(u, \mathrm{v}) \in E$.

קליקה בגודל $k = 3$:



קליקה בגודל $k = 5$:





הגדרה 10.9 בעיית הקליקה

<u>קלט:</u> גרף לא מכוון $G = (V, E)$ ומספר k.

<u>פלט:</u> האם G קליקה בגודל k?

$$CLIQUE = \{\langle G, k\rangle \mid G \text{ גרף לא מכוון המכיל קליקה בגודל } k\}$$

משפט 10.3 $CLIQUE \in NP$

$CLIQUE \in NP$.

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 10.10 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?

$$SubSetSum = \left\{ \langle S,t\rangle \;\middle|\; \text{קיימת } Y\subseteq S \text{ כך ש- } \sum_{x\in Y} x = t \right\}$$

משפט 10.4 $SubSetSum \in NP$

$$SubSetSum \in NP \ .$$

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 למכונת טיורינג אי-דטרמיניסטית

NP=Non-deterministic polynomial-time.

משפט 10.5

לכל בעייה A:

$A\in NP$ אם ורק אם קיימת מכונת טיורינג אי-דטרמיניסטית המכריעה את A בזמן פולינומיאלי.

דוגמה 10.3

נבנה מכונת טיורינג אי-דטרמיניסטית M המכריעה את $CLIQUE$ בזמן פולינומיאלי.

$M=$ על קלט $\langle G,k\rangle$:

- בוחרת באופן אי-דטרמיניסטי 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.

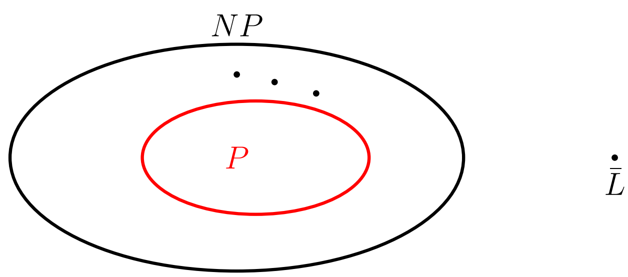
10.7 הקשר בין המחלקה P ו- NP

$P =$ כל הבעיות שניתן להכריע בזמן פולינומיאלי.

$NP =$ כל הבעיות שניתן לאמת בזמן פולינומיאלי.

משפט 10.6

$$P \subseteq NP\ .$$



שאלה פתוחה: האם $P = NP$?

משפט 10.7

P סגורה תחת משלים.

הוכחה: אם $A \in P$ אזי גם $\bar{A} \in P$. ∎

הגדרה 10.11 $CoNP$

$$CoNP = \{A \mid \bar{A} \in NP\ .\}$$

לדוגמה:

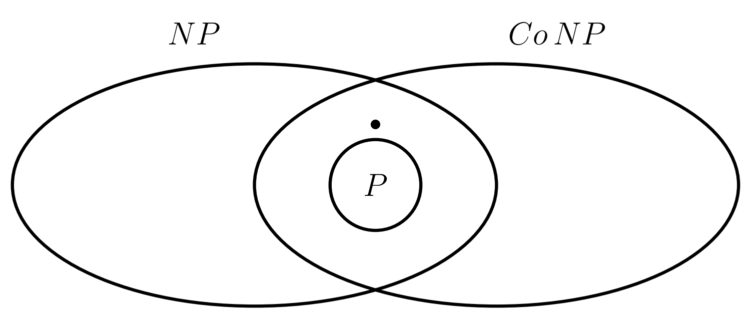
$$\overline{HAMPATH} \in CoNP\ .$$

$$\overline{CLIQUE} \in CoNP\ .$$

שאלה פתוחה: האם $NP = CoNP$?

משפט 10.8

$$P \subseteq NP \cap CoNP\ .$$



שאלה פתוחה: האם $P = NP \cap CoNP$?

נדון בשאלה המרכזית: האם $P = NP$?

הגדרה 10.12 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 10.13 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 10.9 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 $w\in A \Leftarrow f(w)\in B \Leftarrow M_B$ מקבל את $f(w) \Leftarrow M_A$ מקבל את w.
- אם $w\notin A \Leftarrow f(w)\notin B \Leftarrow M_B$ דוחה את $f(w) \Leftarrow M_A$ 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

$$P_f\left(|w|\right)+P_B\left(|f(w)|\right)$$

מכיוון 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

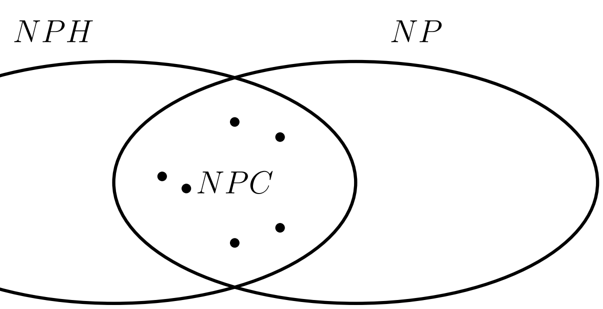
בעייה B נקראת NP קשה אם לכל בעייה $A \in NP$ קיימת רדוקציה $A \leqslant_P B$.

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 $B \leqslant_P A$ אזי $\bar{B} \leqslant_P \bar{A}$.

משפט 11.2

אם $B \leqslant_p A$ וגם $C \leqslant_p B$ אזי $C \leqslant_p A$.

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

<u>קלט:</u> נוסחת CNF, ϕ.
<u>פלט:</u> האם ϕ ספיקה?

$$SAT = \left\{ \langle \phi \rangle \;\middle|\; \text{ספיקה } CNF \text{ נוסחת } \phi \right\}$$

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 $A \in NP$, $A \leqslant_p SAT$.
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 $3SAT$

קלט: נוסחת $3CNF$, ϕ.
פלט: 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 $\langle\phi'\rangle \in 3SAT$.

כיוון $\Rightarrow$:

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.

סיבוכיות

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: $A \leqslant_p SAT$ 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

$$O(n) + O(kn^2) = O(n^2)$$

  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 $O(n^k)$ 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 קונפיגורציות שונות האפשריות.

#	q_0	w_1	w_2	$\ldots$	w_n	$\sqcup$	$\ldots$	$\sqcup$	#
#	q_0								#
#	q_0								#
#									#

אנחנו אומרים כי טבלה כלשהי 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:

$$\phi = \phi_{\text{cell}} \wedge \phi_{\text{start}} \wedge \phi_{\text{move}} \wedge \phi_{\text{acc}} \ . \tag{11.1}$$

נסביר את כל הנוסחאות ϕ_{cell}, ϕ_{start}, ϕ_{move} ו- ϕ_{acc} אחד אחד למטה.

• <u>הנוסחה ϕ_{cell}</u>

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.

• <u>הנוסחה ϕ_{start}</u>

נוסחה ϕ_{start} מבטיחה ששורה הראשונה של הטבלה היא הקונפיגורציה ההתחלתית של N על הקלט w:

$$\begin{aligned}\phi_{\text{start}} =& x_{1,1,\#} \wedge x_{1,2,q_0} \wedge x_{1,3,w_1} \wedge x_{1,4,w_2} \\ & \wedge \ldots \wedge \\ & x_{1,n+2,w_n} \wedge x_{1,n+3,\sqcup} \wedge \ldots \wedge x_{1,n^k-1,\sqcup} \wedge x_{1,n^k,\#}\end{aligned} \tag{11.3}$$

• <u>הנוסחה ϕ_{acc}</u>

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	q_1	b
q_2	a	c

a	q_1	b
a	a	q_2

a	a	q_1
a	a	b

#	b	a
#	b	a

a	b	a
a	b	q_2

b	b	b
c	b	b

החלונות האלה למטה הם דוגמאות לחלונות לא חוקיים:

a	b	a
a	a	a

a	q_1	b
q_1	a	a

b	q_1	b
q_2	b	q_2

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 10.8):

$$CLIQUE = \{\langle G,k\rangle \mid G \text{ מכיל קליקה בגודל } k\} .$$

$CLIQUE$ היא NP- שלמה

הוכחה:

1) הוכחנו כי $CLIQUE \in NP$ במשפט 10.3.

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:

$$x_1 \vee \bar{x}_3 \vee x_4 \qquad \longrightarrow \qquad x_1 \qquad \bar{x}_3 \qquad x_4$$

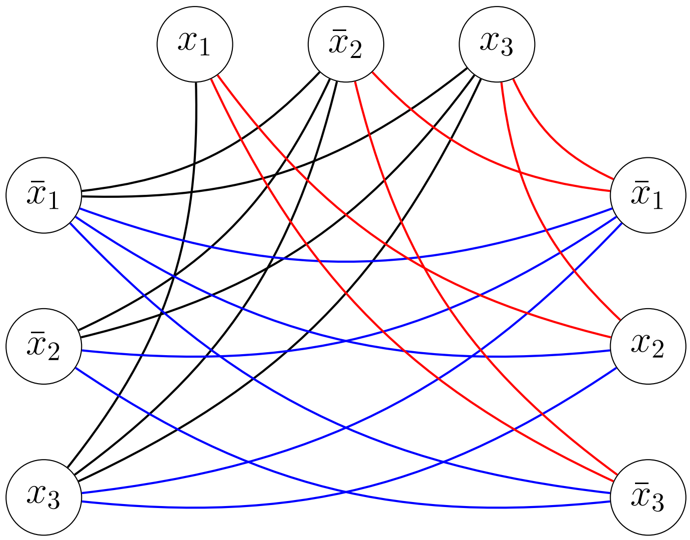
<u>הצלעות של G:</u>

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.

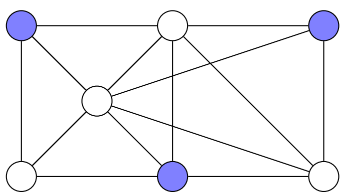
■

12.2 בעיית הקבוצה הבלתי תלויה

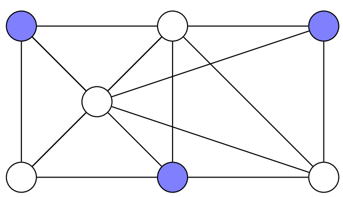
הגדרה 12.1 קבוצה בלתי תלויה

בהינתן גרף לא מכוון $G=(V,E)$, קבוצה בלתי תלויה ב- G היא תת-קבוצה של קודקודים $S\subseteq V$ כך שלכל שני קודקודים $u,\text{v}\in S$ מתקיים $(u,\text{v})\notin E$.

קבוצה בלתי תלוייה בגודל $k=3$:



קבוצה בלתי תלוייה בגודל $k=3$:



הגדרה 12.2 בעיית IS

<u>קלט</u>: גרף לא מכוון $G=(V,E)$ ומספר k.
<u>פלט</u>: האם קיימת קבוצה בלתי תלויה ב- G בגודל k?

$$IS=\left\{\langle G,k\rangle \;\middle|\; \text{גרף לא מכוון המכיל קבוצה בלתי תלויה בגודל } k \;\; G\right\}$$

משפט 12.2 $IS\in NPC$

הבעייה IS היא NP - שלמה.

הוכחה:

<u>**(1) נוכיח כי $IS\in NP$**</u>

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:

$$\langle G,k\rangle \in CLIQUE \quad \Leftrightarrow \quad \langle G',k'\rangle \in IS \ .$$

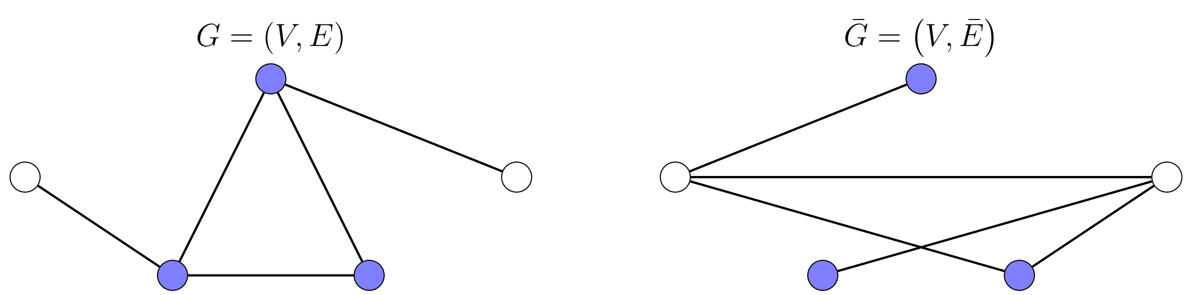
הפונקציית הרדוקציה מוגדרת כך שהתנאים הבאים מתקיימים:

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 $\langle G',k'\rangle \in IS \quad \Leftrightarrow \quad \langle G,k\rangle \in CLIQUE$.

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.

$\Leftarrow$ $\langle G, k\rangle \in CLIQUE$.

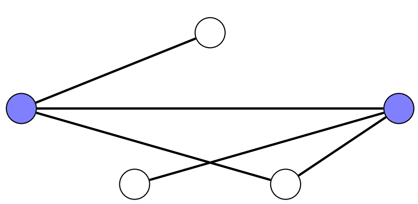
■

12.3 בעיית הכיסוי בקודקודים

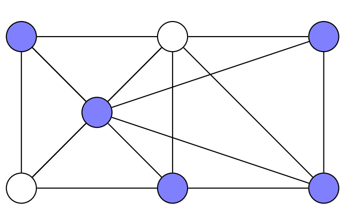
הגדרה 12.3 כיסוי בקודקודים

בהינתן גרף לא מכוון $G = (V, E)$, כיסוי בקודקודים ב- G הוא תת-קבוצה של קודקודים $C \subseteq V$ כך שלכל צלע $u, \mathrm{v} \in S$ מתקיים $u \in C$ או $\mathrm{v} \in C$.

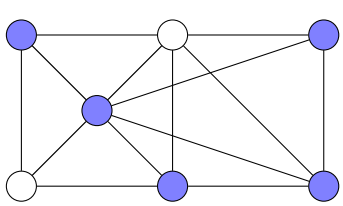
כיסוי בקדקודים בגודל $k = 2$:



כיסוי בקדקודים בגודל $k = 5$:



כיסוי בקדקודים 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?

$$VC = \left\{ \langle G,k\rangle \;\middle|\; \text{גרף לא מכוון המכיל כיסוי בקודקודים בגודל } k \text{ } G \right\}$$

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 . $\langle G,k\rangle \in IS \quad \Leftrightarrow \quad \langle G',k'\rangle \in VC$

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'.

$\Leftarrow$ $\langle G', k' \rangle \in VC$.

כיוון $\Rightarrow$

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 $PARTITION$

קלט: קבוצת מספרים שלמים $S = \{x_1, x_2, \ldots, x_n\}$.
פלט: 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 הגדרה של סיבוכיות מקום

הגדרה 13.1 סיבוכיות מקום של מכונת טיורינג

הסיבוכיות מקום של מ"ט M על קלט w היא פונקציה $f(|w|)$ השווה למספר התאי סרט לכל היותר של המכונה M שבהם נעשה שימוש בחישוב של M על w.

הגדרה 13.2 המחלקה $SPACE(f(n))$

מחלקת $SPACE(f(n))$ היא אוסף כל השפות L עבורן קיימת מכונת טיורינג דטרמיניסטית M שמכריעה אותה כך ש:
על כל קלט w באורך $n=|w|$, המכונה M משתמשת לכל היותר $O(f(n))$ תאי סרט.

$$SPACE(f(n)) = \{L \mid \text{מ"ט } M \text{ שמכריעה } L \text{ ומשתמשת לכל היותר ב- } O(f(n)) \text{ תאי סרט} \ \exists\} \ .$$

דוגמה 13.1

נראה כי ניתן לפתור את הבעיה SAT ע"י אלגוריתם שהוא רץ במקום לינארי. כלומר:

$$SAT \in SPACE(n) \ .$$

תהי ϕ נוסחה בוליאנית כלשהי. נסמן $n=|\phi|$ ונסמן ב- m את מספר המשתנים ב- ϕ. נגדיר מכונה M שפועלת כך:

$M =$ על כל קלט $\langle\phi\rangle$:

(1 M רושמת את המחרוזת $\langle\phi\rangle$ על סרט הקלט.

(2 לכל השמה $a_1,a_2,\ldots,a_m$ (כאשר $a_i\in\{0,1\}$ הוא הערך הנוכחי של x_i):

א) M רושמת את מחרוזת של ההשמה $a_1a_2\cdots a_m$ על סרט העבודה.

ב) M מחשבת את הערך של ϕ עבור ההשמה הנוכחית $a_1,\ldots a_m$ ע"י סריקה של הקלט $\langle\phi\rangle$ שרשום על סרט הקלט.

ג) אם מתקבל $\phi(a_1,\ldots,a_m)=1$ אז M מקבלת.

(3 אם עבור כל ההשמות התקבל $\phi(a_1,\ldots,a_m)=0$ אז M דוחה.

מכאן אנחנו רואים כי המכונה M_1 רצה במקום ליניארי. בפרט:

- M שומרת על סרט העבודה את ההשמה $a_1\ldots a_m$ וזה נדרש $O(m)$ תאים.
- המספר המשתנים, m הוא n לכל היותר.

- לכן M רצה במקום $O(n)$.

לפיכך:

$$SAT \in SPACE(n) \ .$$

הגדרה 13.3 המחלקה $NSPACE(f(n))$

מחלקת $NSPACE(f(n))$ היא אוסף כל השפות L עבורן קיימת מכונת טיורינג אי-דטרמיניסטית N שמכריעה אותה כך ש:
על כל קלט w באורך $n = |w|$ המכונה N משתמשת לכל היותר $O(f(n))$ תאי סרט מתוך כל המסלולי חישוב של N.

$$NSPACE(f(n)) = \left\{ L \ \middle| \ \text{קיים מ"ט א"ד } N \text{ שמכריעה } L \text{ ומשתמשת לכל היותר ב- } O(f(n)) \text{ תאי סרט.} \right\} .$$

דוגמה 13.2

תהי ALL_{NFA} השפה הבאה:

$$ALL_{NFA} = \left\{ \langle A \rangle \ \middle| \ A \text{ הוא } NFA \text{ עבורו } L(A) = \Sigma^* \right\} .$$

הוכיחו כי $ALL_{NFA} \in NSPACE(n)$.

פתרון:
הפתרון מתבוסס על זה שזה פשוט יותר לבנות אלגוריתם המכריע את השפה המשלימה:

$$\overline{ALL_{NFA}} = \left\{ \langle A \rangle \ \middle| \ \text{קיימת } w \in \Sigma^* \text{ עבורה } A \text{ דוחה } w \right\} .$$

נשתמש במשפט מרכזי כדי לבנות אלגוריתם שמכריע את $\overline{ALL_{NFA}}$:

משפט 13.1

אם $M = (Q, \Sigma, \delta, q_0, F)$ הוא NFA וקיים מילה w שנדחת ע"י M אז האורך המילה $|w| \leqslant 2^q$ כאשר $q = |Q|$ הוא המספר המצבים של M, וקיימים אינסוף מלים שנדחות ע"י M.

לפני שנתאר את האלגוריתם עצמו, נגדיר סימון שנשתמש בו בבניית האלגוריתם. נניח ש- $M = (Q, \Sigma, \delta, q_0, F)$ היא מכונת NFA כלשהי. תהי $P(Q)$ הקבוצת החזקה של Q. עבור כל NFA הפונקצית המעברים היא מהצורה

$$\delta : Q \times \Sigma \to P(Q) \ .$$

בהינתן מילה $w = a_1 a_2 \cdots a_n$ כאשר $a_i \in \Sigma$ הוא התו ה- i של המילה, $1 \leqslant i \leqslant n$. נגדיר את הסדרה הבאה:

$$S_0 \ , \ S_1 \ , \ S_2 \ , \quad \ldots \quad , \ S_n$$

כאשר

$$S_0 \triangleq \{q_0\} \ , \quad S_{i+1} \triangleq \delta(S_i, a_i) \ ,$$

כאשר $S_i \in P(Q)$ לכל $0 \leqslant i \leqslant n$.

<u>בניית האלגוריתם</u>

נבנה אלגוריתם לא-דטרמיניסטי, N המכריע את $\overline{ALL_{NFA}}$ באופן הבא:

$N =$ "על כל קלט x:

1) בודקת אם $x = \langle M \rangle$, כאשר $M = (Q, \Sigma, \delta, q_0, F)$ היא מכונת NFA.

- אם לא $\Leftarrow N$ תדחה.

2) יהי $q = |Q|$ מספר המצבים של M. נגדיר $S_0 = \{q_0\}$.

3) N מבצעת את הלולאה הבאה:

לכל $0 \leqslant i \leqslant 2^q - 1$:

א) בוחרת באופן אי־דטרמיניסטי תו קלט $a_i \in \Sigma$.

ב) מחשבת

$$S_{i+1} = \delta\left(S_i, a_i\right).$$

ג) אם $S_{i+1} \cap F \neq \varnothing \Leftarrow N$ תדחה.

בפועל N בודקת את התנאי הזה ע"י לסמן את כל המצבים שב- S_{i+1}. אם אחד מהמצבים המסומנים הוא מצב קבלה אז N תדחה.

4) אם בסיום הלולאה לא היה מצב-קבלה באף אחת מן הקבוצות S_i אז N תקבל. "

אם $x \in \overline{ALL_{NFA}}$:

$\Leftarrow$ $x = \langle A \rangle$, כאשר A היא מכונת NFA. וקיימת מילה $w \in \Sigma^*$ כך ש־ A תדוחה.

$\Leftarrow$ קיימת מילה w' באורך לכל היותר 2^q ש־A תדחה.

$\Leftarrow$ קיימת ריצה של N שבה N בוחרת את התווים של w' בלולאה.

$\Leftarrow$ במהלך הריצה של A על w', אין מצב קבלה באף אחת מן הקבוצות S_i.

$\Leftarrow$ N לא דחתה עד סוף הלולאה.

$\Leftarrow$ בסופה N תקבל.

אם $x \notin \overline{ALL_{NFA}}$ אז שני מקרים:

מקרה 1) $x \neq \langle A \rangle \Leftarrow N$ תדחה בשלב 1.

מקרה 2) $x = \langle A \rangle$ ו- $L(A) = \Sigma^*$.

$\Leftarrow$ לכל מילה $w \in \Sigma^*$, קיים שלב שבו A נמצא במצב קבלה.

$\Leftarrow$ בכל ריצה של N, קיימת איטרציה i עבורה $S_i \cap F \neq \varnothing$.

$\Leftarrow$ באיטרציה זו N תדחה.

$\Leftarrow$ בכל ריצה N תדחה.

$\Leftarrow$ N דוחה את x.

סיבוכיות מקום

- נסמן ב־ $n = |\langle M \rangle|$ את אורך הקלט, וב־ $q = |Q|$ את מספר המצבים של ה־ NFA.
- כל מצב וכל מעבר של M מופיעים בקידוד, מתקיים $q = O(n)$.
- במהלך כל ריצה, N שומרת רק את המידע הבא:

* הקבוצה הנוכחית $S_i \subseteq Q$ של מצבים אפשריים.
  לפועל N שומרת S_i בוקטור ביטים באורך q לכל היותר.
* מונה של האיטרציות הלולאה עד 2^q, המאוחסן בייצוג בינארי ודורש $O(q)$ ביטים.
* תו קלט אחד הנבחר באפן אי-דטרמיניסטי בכל איטרציה, ומשתני עזר לחישוב S_{i+1}, הדורשים מקום קבוע או לינארי ב־q.

לפיכך סיבוכיות המקום הכוללת של N היא

$$O(q) = O(n) \ .$$

לפיכך האלגוריתם N פועל במקום לינארי.

שימו לב: N לינארי במקום אף על פי שזמן הריצה שלו עלול להיות אקספוננציאלי.

13.2 משפט סביץ'

הגדרה 13.4 $CANYIELD$

בהינתן מכונת טיורנג אי-דטרמיניסטית N, מספר טבעי חיובי t, ושתי קונפיגורציות c_1, c_2 של N (ראו את ההגדרה של קונפיגורציה בהגדרה 1.3). האלגוריתם $CANYIELD$ הוא אלגוריתם דטרמיניסטי הבודק אם ניתן לעבור מ- c_1 ל- c_2 על ידי לכל היותר t צעדי חישוב של N.
התאור פסאודוקוד של $CANYIELD$ הוא כדלקמן:

$CANYIELD =$ על קלט $\langle N, c_1, c_2, t\rangle$:

1) רושם את c_1, c_2 ו- t על מחסנית.

2) בודק אם N היא מכונת טיורינג, c_1, c_2 קונפיגורציות ו- t מספר טבעי חיובי.

* אם לא אז הוא דוחה.

3) אם $t = 1$:

* אם $c_1 = c_2$ אז הוא מקבל.
* אחרת אם $c_1 \vdash_N c_2$ (אם אפשר לעבור מ- c_1 ל- c_2 בצעד אחד [ראו הגדרה 1.4]) מקבל.
* אחרת הוא דוחה.

4) אם $t > 1$, לכל קונפיגורציה c_k של הרצה של N על w אשר משתמשת במקום $f(n)$ (כאשר w היא המילה הנקראת של הקונפיגורציה c_k):

5) מריץ $CANYIELD\left(N, c_1, c_k, \left\lfloor \dfrac{t}{2} \right\rfloor\right)$
כאשר $\left\lfloor \frac{t}{2} \right\rfloor$ הוא השלם הכי הקרוב ל- $\frac{t}{2}$ וקטן מ- או שווה ל- $\frac{t}{2}$.

6) מריץ $CANYIELD\left(N, c_k, c_2, \left\lceil \dfrac{t}{2} \right\rceil\right)$
כאשר $\left\lceil \frac{t}{2} \right\rceil$ הוא השלם הכי הקרוב ל- $\frac{t}{2}$ וגדול מ- או שווה ל- $\frac{t}{2}$.

7) אם שתי ההרצות בשלבי 4) ו- 5) הסתיימו בקבלה $\Leftarrow$ מקבל.

8) אחרת אם לא התקבלה תשובת קבלה $\Leftarrow$ דוחה.

משפט 13.2 משפט סביץ'

לכל פונקציה $f:\mathbb{N}\to\mathbb{R}^+$, אם $f(n)\geqslant n$ אז

$$NSPACE\left(f(n)\right)\subseteq SPACE\left(f^2(n)\right)\ .$$

הוכחה:

הריעון של ההוכחה:

תהי N מ"ט אי-דטרמיניסטית שמכריעה את השפה A במקום $O\left(f(n)\right)$, כאשר n אורך הקלט w של N.
נבנה מכונת טיורינג דטרמיניסטית, M שמכריעה את A במקום $O\left(f^2(n)\right)$.
כלומר, בהינתן $N\in NSPACE\left(f(n)\right)$ המכריעה שפה A, נבנה $M\in SPACE\left(f^2(n)\right)$ המכריעה A.
כלומר, אנחנו נראה שלכל $N\in NSPACE\left(f(n)\right)$ קיימת $M\in SPACE\left(f^2(n)\right)$.
באופן הזה אנחנו נוכיח כי

$$NSPACE\left(f(n)\right)\subseteq SPACE\left(f^2(n)\right)\ .$$

בניית המכונה:

תהי N מכונת טיורינג אי-דטרמיניסטית שמכריעה השפה A.
תהי w מחרוזת שהיא הקלט של N.
בהינתן שתי קונפיגורציות c_1,c_2 של N ומספר טבעי חיובי t.

- אם ניתן לעבור מ- c_1 ל- c_2 בלכל היותר t צעדים $\Leftarrow$ $CANYIELD\left(N,c_1,c_2,t\right)$ מקבל.
- אחרת $\Leftarrow$ $CANYIELD\left(N,c_1,c_2,t\right)$ דוחה.

נגדיר מכונת טיורינג דטרמיניסטית M שמסמלצת את המכונה האי-דטרמיניסטית N באופן הבא.

ראשית נסמן ב- n את אורך הקלט w של N.

תהי c_0 הקונפיגורציה ההתחלתית.

נתקן את N כך שלאחר כל ריצה הראש חוזר לקצה השמאל של תוכן הסרט ו- N עוברת לקונפיגורציה c_{acc}.

נגדיר d כך ש- $2^{df(n)}$ הוא חסם עליון של מספר הקונפיגורציות שקיימות בריצות של N שדורשות $O\left(f(n)\right)$ מקום.

המכונת טיורינג הדטרמיניסטית M תוגדר כך:

$M=$ "על קלט w:

1) מריצה $CANYIELD$ על הקלט $\langle N,c_0,c_{\text{acc}},2^{df(n)}\rangle$ ועונה כמוה."

הוכחת הנכונות:

נניח $w\in L(N)$ ו- $N\in NSPACE\left(f(n)\right)$.

$\Leftarrow$ לפי ההגדרה של d, ל- N יש $2^{df(n)}$ לכל היותר.

$\Leftarrow$ קיים מסלול חישוב N על w מ- c_0 ל- c_{acc}.

$\Leftarrow$ האלגוריתם $CANYIELD$ על הקלט $\langle N,c_0,c_{\text{acc}},2^{df(n)}\rangle$ יקבל.

$\Leftarrow$ M יקבל w.

נניח $w \notin L(N)$ ו- $N \in NSPACE(f(n))$.

$\Leftarrow$ לפי ההגדרה של d, ל- N יש $2^{df(n)}$ לכל היותר.

$\Leftarrow$ לא קיים מסלול חישוב של N על w מ- c_0 ל- c_{acc}.

$\Leftarrow$ האלגוריתם $CANYIELD$ על הקלט $\langle N, c_0, c_{\text{acc}}, 2^{df(n)}\rangle$ ידחה.

$\Leftarrow$ M ידחה w.

סיבוכיות מקום:

- כל פעם ש- $CANYIELD$ מפעיל את עצמו באופן רקורסיבי, הוא רושם את c_1, c_2 ו- t על מחסנית, כך שניתן יהיה לשחזר אותם לאחר הקריאה הרקורסיבית הבאה.
- בגלל ש- $N \in NSPACE(f(n))$ אזי הכתיבה של c_1, c_2 ו- t על המחסנית דורשת $O(f(n))$ מקום.
- בכל שלב של הרקורסיה, האלגוריתם $CANYIELD$ מחלק את t ב- 2.
- הערך ההתחלתי של t הוא $2^{df(n)}$ לכן העומק של הרקורסיה הוא
$$O\left(\log_2\left(2^{df(n)}\right)\right) = O(f(n)) \ .$$
- לכן המכום הכולל ש- M דורש הוא $O(f^2(n))$.

לפיכך
$$M \in SPACE\left(f^2(n)\right) \ .$$
לסיכום: הוכחנו שבהינתן מכונת אי-דטרמיניסטית N כלשהי שמכריעה שפה A כלשהי עבורה
$$N \in NSPACE(f(n)) \ ,$$
קיימת מכונת טיורינג דטרמיניסטית M שמכריעה A במקום $O(f^2(n))$, כלומר:
$$M \in SPACE\left(f^2(n)\right) \ .$$
לפיכך:
$$NSPACE(f(n)) \subseteq SPACE\left(f^2(n)\right) \ .$$

■

13.3 המחלקה PSPACE

ההגדרה הבאה היא האנלוג של ההגדרה 10.2 של אלגוריתם זמן פולינומיאלי.

הגדרה 13.5 אלגוריתם מקום פולינומיאלי

אומרים כי אלגוריתם A מכריע בעייה במקום פולינומיאלי אם קיים קבוע $c > 0$ כך שהמקום הריצה של A על קלט w חסום ע"י $O(|w|^c)$.

התזה של צרץ' טיורינג אומר שאם קיים אלגוריתם המכריע בעייה במקום פולינומיאלי, אזי קיימת מכונת טיורינג דטרמיניסטית המכריעה את השפה השקולה לבעייה זו במקום פולינומיאלי.

$$\text{אלגוריתם מכריע} \equiv \text{מכונת טיורינג דטרמיניסטית} \ .$$

הגדרה 13.6 המחלקה $PSPACE$

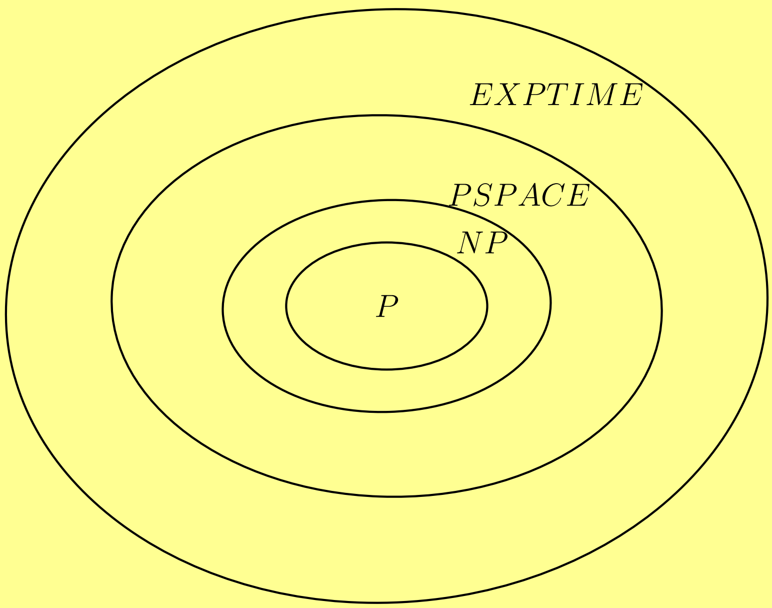
המחלקה $PSPACE$ היא אוסף כל הבעיות (השפות) שקיים עבורן אלגוריתם (מכונת טיורנג) דטרמיניסטי המכריע אותן במקום פולינומיאלי.

הגדרה 13.7 המחלקה $NPSPACE$

$NPSPACE$ היא אוסף כל הבעיות (השפות) שקיים עבורן אלגוריתם (מכונת טיורנג) אי-דטרמיניסטי המכריע אותן במקום פולינומיאלי.

משפט 13.3

$$P \subseteq NP \subseteq PSPACE \subseteq EXPTIME \ .$$



13.4 שלמות ב- $PSPACE$

הגדרה 13.8 $PSPACE$ קשה

בעייה B נקראת $PSPACE$ קשה אם לכל בעייה $A \in PSPACE$ קיימת רדוקציה זמן פולינומיאלית מ- A ל- B. כלומר: $A \leqslant_P B$.

הגדרה 13.9 $PSPACE$ שלמות

בעייה B נקראת $PSPACE$ שלמה אם השני התנאים הבאים מתקיימים:

1) $B \in PSPACE$

2) לכל בעייה $A \in PSPACE$ קיימת רדוקציה זמן פולינומיאלית מ- A ל- B. כלומר: $A \leqslant_p B$.

13.5 נוסחאות בוליאניות עם כמתים

בפרקים 11 ו- 12 הגדרנו נוסחה בוליאנית כביטוי מתמטי שבנוי מהמרכיבים הבאים:

- משתנים בוליניאיים, שמקבלים את הערכים 0 ו- 1 (לעתים מסומנים F ו- T).
- אופרטורים בוליאניים עיקריים

$\wedge$	וגם
$\vee$	או
$\neg$	לא

כעת נכליל את ההגדרה הזו לסוג היותר מורחב של נוסחה בוליאנית: נוסחה בוליאנית עם כמתים.

הגדרה 13.10 נוסחה בוליאנית עם כמתים - QBF

בנוסחת בוליאנית עם כמתים מופיעה אחת מהשני כמתים העיקריים:

$\forall$	"לכל" (נקרא גם "כמת כולל")
$\exists$	"קיים" (נקרא גם "כמת ישי")

דוגמה 13.3 דוגמאות של נוסחאותת בוליאניות עם כמתים

בדוגמאות הבאות x,y הם משתנים בוליאנים. כלומר $x,y\in\{0,1\}$.

(1

$$\phi=\forall\ x\ \exists\ y\ \ [(x\vee y)\wedge(\bar{x}\vee\bar{y})]\ .$$

בדוגמה זו $\phi=1$.

(2

$$\phi=\forall\ x\ \ [x\vee\bar{x}]\qquad\rightarrow\qquad\phi=1\ .$$

(3

$$\phi=\exists\ x\ \ (x\wedge\bar{x})\qquad\rightarrow\qquad\phi=0\ .$$

הגדרה 13.11 $TQBF$

$\langle\phi\rangle$ בשפה $TQBF$ אם ϕ נוסחת בוליאנית עם כמתים והנוסחה מעורכת לאמת.

$$TQBF=\left\{\langle\phi\rangle\ \middle|\ \phi\ \text{נוסחת בוליאנית עם כמתים ו-}\ \phi=1\ \right\}\ .$$

הערה 13.1

בניגוד ל- SAT עבורה השאלה היא האם קיימת הצבת אמת, ב- $TQBF$ לכל נוסחה יש ערך אמת או שקר יחיד, מכיוון שאין משתנים חופשיים.

משפט 13.4

$$SAT \subseteq TQBF \ .$$

הוכחה: תרגיל בית. ■

משפט 13.5

השפה $TQBF$ היא NP שלמה.

הוכחה: נראה כי

1) $TQBF \in PSPACE$
2) לכל $A \in PSPACE$ מקתיים $A \leqslant_P TQBF$.

נבנה אלגוריתם רקורסיבי $A \in PSPACE$ שמכריע את $TQBF$ באופן הבא.

<u>בניית האלגוריתם</u>

$A =$ "על קלט $\langle \psi \rangle$ כאשר

$$\psi = Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \phi\left(x_1, \ldots, x_n\right) \ ,$$

כאשר לכל $1 \leqslant i \leqslant n$, Q_i הוא כמת $\forall$ או $\exists$, x_i משתנה בוליאני ו- $\phi\left(x_1, \ldots, x_n\right)$ נוסחה בוליאנית בלי כמתים:

1) <u>מקרה הבסיס:</u>

אם $n=0$ אז $\phi\left(x_1, \ldots, x_n\right)$ הוא קבוע ומעריך אותו.

2) <u>מקרה הרקורסיבי:</u>

- אם $Q_1 = \exists$:
  - מריץ $A\Big(\psi\left(x_1 = 0\right)\Big)$.
  - מריץ $A\Big(\psi\left(x_1 = 1\right)\Big)$.
  - אם **לפחות אחד מהם** התקבל אז מקבל.
- אם $Q_1 = \forall$:
  - מריץ $A\Big(\psi\left(x_1 = 0\right)\Big)$.
  - מריץ $A\Big(\psi\left(x_1 = 1\right)\Big)$.
  - אם **שניהם** התקבלו אז מקבל."

<u>הוכחת הנכונות</u>

ניתן להוכיח הנכונות של A ע"י אינדוקציה.

<u>שלב הבסיס: $n=0$.</u>

אם $\psi = 1 \Leftarrow$ בשלב (1) A מחשב את $\phi(x_1, \ldots, x_n) = 1 \Leftarrow A$ מקבל.

אם $\psi = 0 \Leftarrow$ בשלב (1) A מחשב את $\phi(x_1, \ldots, x_n) = 0 \Leftarrow A$ דוחה.

<u>שלב המעבר: $n=k$.</u>

נניח ש- A מכריע נוסחה כלשהי עם כמתים $\psi = Q_1x_1Q_2x_2\ldots Q_kx_k\phi\left(x_1,\ldots,x_n\right)$ במקרה $n=k$. זוהי ההנחת האינדוקציה שלנו. נוכיח כי A גם מכריע נוסחה ψ כלשהי במקרה ש- $n=k+1$ באופן הבא:

אם $\psi\left(x_1,x_2,\ldots,x_k,x_{k+1}\right)=1 \Leftarrow$ שני מקרים:

מקרה 1: $Q_1=\exists$

$\Leftarrow$ $\psi\left(x_1=0\right)=1$ או $\psi\left(x_1=1\right)=1$ ושתיהן QBF בעלי k משתנים וכמתים.

$\Leftarrow$ $A\big(\psi\left(x_1=0\right)\big)$ התקבל או $A\big(\psi\left(x_1=1\right)\big)$ התקבל.

$\Leftarrow$ A מקבל $\psi\left(x_1,\ldots,x_{k+1}\right)$.

מקרה 2: $Q_1=\forall$

$\Leftarrow$ $\psi\left(x_1=0\right)=1$ וגם $\psi\left(x_1=1\right)=1$ ושתיהן QBF בעלי k משתנים וכמתים.

$\Leftarrow$ $A\big(\psi\left(x_1=0\right)\big)$ התקבל וגם $A\big(\psi\left(x_1=1\right)\big)$ התקבל.

$\Leftarrow$ A מקבל $\psi\left(x_1,\ldots,x_{k+1}\right)$.

אם $\psi\left(x_1,x_2,\ldots,x_k,x_{k+1}\right)=0 \Leftarrow$ שני מקרים:

מקרה 1: $Q_1=\exists$

$\Leftarrow$ $\psi\left(x_1=0\right)=0$ וגם $\psi\left(x_1=1\right)=0$ ושתיהן QBF בעלי k משתנים וכמתים.

$\Leftarrow$ $A\big(\psi\left(x_1=0\right)\big)$ דוחה וגם $A\big(\psi\left(x_1=1\right)\big)$ דוחה.

$\Leftarrow$ A דוחה $\psi\left(x_1,\ldots,x_{k+1}\right)$.

מקרה 2: $Q_1=\forall$

$\Leftarrow$ $\psi\left(x_1=0\right)=0$ או $\psi\left(x_1=1\right)=0$ ושתיהן QBF בעלי k משתנים וכמתים.

$\Leftarrow$ $A\big(\psi\left(x_1=0\right)\big)$ דוחה או $A\big(\psi\left(x_1=1\right)\big)$ דוחה.

$\Leftarrow$ A דוחה $\psi\left(x_1,\ldots,x_{k+1}\right)$.

סיבוכיות מקום של האלגוריתם

כדי לחשב את הסיבוכיות מקום, נסמן השימוש מקום ב- $S\left(n,m\right)$ כאשר n המספר המשתנים של ψ ו- $m=|\psi|$ הוא האורך של ψ. אז יש לנו את היחס רקורסיבי הבא:

$$S(0,m)=O\left(m\right)\ , \qquad S(n,m)=S(n-1,m)+O(m)\ .$$

מכאן

$$S(n,m)=O(nm)\ .$$

הוכחה כי השפה $TQBF$ היא $PSPACE$ קשה: בניית הרדוקציה $L\leqslant_P TQBF$.

נראה כי לכל שפה $L\in PSPACE$ מתקיים

$$L\leqslant_P TQBF\ .$$

תהי L שפה השייכת ל- $PSPACE$ ותהי M מכונת טיורינג המכריעה את L במקום פולינומיאלי $S(n)$. נוכיח שקיימת פונקציה f כך ש:

$$x\in L \iff f(x)\in TQBF\ .$$

ז"א $f(x)=\psi$ כאשר ψ היא נוסחה בוליאנית עם כמתים כך ש:

$$\begin{aligned} x\in L &\quad\Rightarrow\quad \psi=1 \quad , \\ x\notin L &\quad\Rightarrow\quad \psi=0 \quad . \end{aligned}$$

כדי להגדיר את הפונקצית הרדוקציה, קודם נגדיר את הנוסחאה הבוליאנית הבאה. יהיו c,c' שתי קונפיגורציות של המכונה M ויהי t מספר שלם חיובי.

$$\phi\left(c,c',t\right)\triangleq\begin{cases}1 & : \text{אפשר לעבור מ- } c \text{ ל- } c' \text{ עם } t \text{ צעדים לכל היותר} \\ 0 & : \text{אחרת}\end{cases} \ .$$

הצורה המפורשת של הנוסחה $\phi\left(c,c',t\right)$ עצמה בנויה רקורסיבי באופן הבא.

המקרה $t>1$

לכל $t>1$ ולכל קונפיגורציות c,c':

$$\phi\left(c,c',t\right)=\exists w\ \forall x,y \quad (x=c\wedge y=w)\vee(x=w\wedge y=c') \quad\Rightarrow\quad \phi\left(x,y,\left\lfloor\frac{t}{2}\right\rfloor\right) \ .$$

כאן w מסמן קונפיגורציה ביניים בין הקונפיגורציה c לקונפיגורציה c'. היחס הרקורסיבי הזה של $\phi\left(c,c',t\right)$ אומר ש- M יכול לעבור מ- c ל- c' ב- t צעדים לכל היותר אם קיימת קונפיגורציה ביניים w כך ש- M יכול לעבור מ- c ל- w בכל היותר $\lfloor\frac{t}{2}\rfloor$ צעדים, ואחר כך M יכול לעבור מ- w ל- c' בכל היותר $\lfloor\frac{t}{2}\rfloor$ צעדים. הנוסחאות $\phi\left(c,w,\lfloor\frac{t}{2}\rfloor\right)$ ו- $\phi\left(w,c',\lfloor\frac{t}{2}\rfloor\right)$ הן נבנות בשלב הבא של הרקורסיה באותו אופן. התתליך הזה ממשיך עד שנגיע לנוסחה $\phi(x,y,t)$ שבה $t=0$ או $t=1$.

המקרה $t=1$

$$\phi(c,c',t=1)=\begin{cases}1 & : \text{אם } M \text{ יכול לעבור מ- } c \text{ ל- } c' \text{ בצעד חישוב בודד} \\ 0 & : \text{אחרת}\end{cases}$$

הצורה המפורשת של $\phi(c,c',t=1)$ היא כדלקמן. תהיינה c,c' שתי קונפיגורציות כלשהן. נגדיר טבלת הקונפיגורציות:

#	c_1	c_2	$\ldots$	c_{N-1}	c_N	#
#	c'_1	c'_2	$\ldots$	c'_{N-1}	c'_N	#

נניח כי $M=(Q,\Sigma,\Gamma,\delta,q_0,q_{\text{acc}},q_{\text{rej}})$ ותהי $C=Q\cup\Gamma\cup\{\#\}$. עבור כל תא ה- (i,j) של הטבלת הקונפיגורציות נגדיר משתנה בוליאני $x_{i,j,s}$ לכל $1\leqslant i\leqslant 2$ ולכל $1\leqslant j\leqslant N$ כך ש:

$$x_{ijs}=\begin{cases}1 & : \text{הסימן } s\in C \text{ כתוב בתא } ij \\ 0 & : \text{הסימן } s\in C \text{ לא כתוב בתא } ij\end{cases}$$

למשל אם בתא $(2,5)$ כתוה a אז $x_{2,5,\text{a}}=1$ בעוד $x_{2,5,\text{b}}=0$. הנוסחה $\phi\left(c,c',t=1\right)$ מוגדרת:

$$\phi\left(c,c',t=1\right)=\phi_{\text{cell}}\wedge\phi_c\wedge\phi_{\text{move}}\wedge\phi_{c'} \ .$$

- $\phi_{\text{cell}}=1$ אם ורק אם יש רק סימן אחד בדיוק שכתוב בכל תא ו- 0 אחרת. בפרט:

$$\phi_{\text{cell}}=\bigwedge_{\substack{1\leqslant j\leqslant 2\\1\leqslant j\leqslant n^k}}\left[\left(\bigvee_{s\in C}x_{i,j,s}\right)\wedge\left(\bigwedge_{\substack{s,t\in C\\s\neq t}}\left(\overline{x_{i,j,s}}\vee\overline{x_{i,j,t}}\right)\right)\right] \tag{13.1}$$

* $\bigvee_{s\in C} x_{i,j,s}$ מבטיח שלכל תא לפחות משתנה אחד דולק.
* $\bigwedge_{\substack{s,t\in C\\ s\neq t}} \left(\overline{x_{i,j,s}} \vee \overline{x_{i,j,t}}\right)$ מבטיח שלכל תא משתנה אחד לכל היותר דולק.

לפיכך ϕ_{cell} מסופקת אם ורק אם יש בדיוק סימן אחד כתוב בכל תא. בניגוד זה אם שני משתנים של אותו תא דולקים בגלל שהסימן s וגם הסימן t כתובים בו זמנית בתא ה- ij אז $x_{i,j,s}=1$ וגם $x_{i,j,t}=1$ ואז ϕ_{cell} תהיה שווה ל- 0.

- הנוסחה ϕ_c ($\phi_{c'}$) קובעת כי המשתנים הספציפים של הקונפיגורציות c (c') הם דולקים:

$$\begin{aligned}\phi_c =& x_{1,0,\#} \wedge x_{1,1,c_1} \wedge x_{1,2,c_2} \wedge \ldots \wedge x_{1,N-1,c_{N-1}} \wedge x_{1,N,c_N} \wedge x_{1,N+1,\#} \ ,\\ \phi_{c'} =& x_{2,0,\#} \wedge x_{2,1,c'_1} \wedge x_{2,2,c'_2} \wedge \ldots \wedge x_{2,N-1,c'_{N-1}} \wedge x_{2,N,c'_N} \wedge x_{2,N+1,\#} \ .\end{aligned} \tag{13.2}$$

- $\phi_{\text{move}}=1$ אם אפשר להגיע מ- c ל- c' על ידי תזוזה חוקית אחת של M ו- 0 אחרת. תזוזה חוקית בין כל שתי קונפיגורציות מוגדרת על פי סהפונקצית המעברים של M. במונחי הטבלה, אפשר להגדיר תזוזה חוקית בין השתי שורות על ידי תת-טבלה מסדר 2×3 שמכילה 3 עמודות.

נקרא תת-טבלה כזה "חלון". אלה הם דוגמאות של חלונות חוקיים:

a	q_1	b
q_2	a	c

a	q_1	b
a	a	q_2

a	a	q_1
a	a	b

#	b	a
#	b	a

a	b	a
a	b	q_2

b	b	b
c	b	b

בעוד אלה הם דוגמאות של חלונות לא חוקיים:

a	b	a
a	a	a

a	q_1	b
q_1	a	a

b	q_1	b
q_2	b	q_2

$\phi_{\text{move}}=1$ אם כל חלון של הטבלה חוקי ו- 0 אם לא. בפרט, כל חלון מכיל 6 תאים. החלון המכיל את ה- 3 עמודות i, $i+1$ ו- $i+2$ תקרא החלון-i. לכן:

$$\begin{aligned}\phi_{\text{move}} &= \bigwedge_{1\leqslant i\leqslant N-2} \left(\text{חלון-} i \text{ חוקי}\right)\\ &= \bigwedge_{1\leqslant i\leqslant N-2} \left(x_{1,i,c_i} \wedge x_{1,i+1,c_{i+1}} \wedge x_{1,i+2,c_{i+2}} \wedge x_{2,i,c'_i} \wedge x_{2,i+1,c'_{i+1}} \wedge x_{2,i+2,c'_{i+2}}\right)\end{aligned} \tag{13.3}$$

המקרה $t=0$

אם $t=0$ אז

$$\phi\left(c,c',t=0\right)=\begin{cases}1 & : c=c'\\ 0 & : c\neq c'\end{cases} \ .$$

כעת נחזור להגדרה עצמה של הרדוקציה פולינומיאלית. תהי L שפה כריעה ע"י מכונת טיורינג M במקום $O\left(n^k\right)$, אז קיימת רדוקציה $f(x)=\psi$ כך ש:

$$\psi=\phi\left(c_{\text{start}},c_{\text{acc}},h\right)$$

כאשר $h=2^{df(n)}$ ו- $d>0$ מספר ממשי חיובי הנבחר כך של- M יש לכל היותר $2^{df(n)}$ קונפיגורציות בהינתן קלט של אורך n, ו- $f(n)=n^k$.

הוכחת הנכונות

אם $x \in L$

$\Leftarrow$ M תקבל x

$\Leftarrow$ M יכולה לעבור מ- c_{start} ל- c_{acc} במספר צעדים פחות מ- או שווה ל- h

$\Leftarrow$ $\psi = 1$

$\Leftarrow$ $\psi \in TQBF$.

אם $x \notin L$

$\Leftarrow$ M תדחה x

$\Leftarrow$ לא קיימים צעדים של M מ- c_{start} ל- c_{acc}.

$\Leftarrow$ $\psi = 0$

$\Leftarrow$ $\psi \notin TQBF$.

סיבוכיות זמן של הרדוקציה

בפונקצית הרדוקציה מחשבת $f(x) = \psi = \phi(c_{\text{start}}, c_{\text{acc}}, h)$ כאשר $h = 2^{df(n)}$ ו- $f(n) = n^k$, באופן רקורסיבי. לרקורסיה יש מספר שלבים השווה ל:

$$\log_2\left(2^{df(n)}\right) = df(n) = dn^k \ .$$

- הנוסחה (13.1) של ϕ_{cell} מכילה n^{2k} נוסחאות עם 3 ליטרלים. לכן הסיבוכיות זמן של ϕ_{cell} היא:
$$O\left(n^{2k}\right) \ .$$
- הנוסחאות ϕ_c ו- $\phi_{c'}$ במשואה (13.2) מכילות בדיוק n^k ליטרלים. לכן הסיבוכיות זמן של ϕ_c ושל $\phi_{c'}$ הן:
$$O\left(n^k\right) \ .$$
- הנוסחה (13.3) של ϕ_{move} מכילה n^{2k} נוסחאות עם 6 ליטרלים. לכן הסיבוכיות זמן של ϕ_{move} היא
$$O\left(n^{2k}\right) \ .$$

לפיכך בסה"כ הרדוקציה ניתנת לחישוב בזמן $O\left(n^{2k}\right)$ ולכן $f \in TIME\left(n^{2k}\right)$ ולכן הרדוקציה היא רדוקציה פולינומיאלית.

■

13.6 המחלקה L

13.7 המחלקה NL

13.8 שלמות ב- NL

13.9 שיוויון NL ו- coNL