

חישוביות וסיבוכיות

תוכן העניינים

1	מכונות טיריניג
4	הגדרה של מכונת טיריניג
4	טבלת המעברים
17	חישוב פונקציות
21	
2	מודלים חישוביים שקולית
24	
3	מכונות טיריניג מרובת סרטים
28	מכונות טיריניג מרובת סרטים: הגדרה היוריסטית
28	מכונות טיריניג מרובת סרטים: הגדרה פורמלית
29	كونפיגורציה של מכונת טיריניג מרובה סרטים
31	שקלות בין מטם"ס למ"ט עם סרט יחיד
4	מכונת טיריניג אי-דטרמיניסטיבית
36	הגדירה של מכונת טיריניג אי-דטרמיניסטיבית
36	ע"ז החישוב של מ"ט א"ד
38	שקלות בין מכונת טיריניג אי-דטרמיניסטיבית למכונת טיריניג דטרמיניסטיבית
39	
5	התזה של צרצ טיריניג ודקדוקים כלליים
43	היחס בין הכרעה וקבלה
43	שקלות של מכונת טיריניג ותוכנית מחשב
44	SIMPLE
44	דקדוקים כלליים
50	דקדוקים כלליים ומכונת טיריניג
56	היררכיה של חומסקי
57	כל שפה חסרת הקשר הינה כריעה
57	
6	תכונות סגורות של R ו- RE
59	הגדירה של השפות R ו- RE
59	היחס בין הכרעה וקבלה
59	סגורות של שפות כריעות ושפות קבילות
60	סגורות של שפות כריעות תחת $DROPOUT$
65	קידוד של מ"ט דטרמיניסטיבית
66	מ"ט אוניברסלית U
66	
7	אי-כריעות
69	השפות L_{halt}, L_{acc} לא כריעות
69	השפה L_E לא כריעת
73	השפה L_{EQ} לא כריעת
75	סיכום: כריעות וקבילות של שפות
78	

79	8 רדוקציה
79	טבלה של רדוקציות
79	מכונת טיורינג המחשבת את פונקציה
81	רדוקציות
88	דוגמאות בשימוש של משפט הרדוקציה בין שפות משלימות (משפט 8.2)
88	דוגמאות בשימוש של משפט הרדוקציה (משפט 8.1)
95	9 מבוא לסיבוכיות זמן
95	הגדרה של סיבוכיות זמן
102	יחס בין הסיבוכיות של מכונת טיורינג עם סרט יחיד ומכונת טיורינג מרובת סרטים
103	דטרמיניסטי אי טיורינג ומכונת דטרמיניסטית טיורינג מכונת של הסיבוכיות בין יחס P המחלקה
104	PATH בעיית
107	RELPRIME בעיית
108	שימושיים משפטים של *הוכחות
110	
10 NP והמחלקה P המחלקה	113
113	P המחלקה
114	P - בעיות דוגמאות
114	HAMPATH בעית
115	אימות אלגוריתם
115	NP המחלקה
118	אי-דטרמיניסטי טיורинг למוכנת NP בין הקשר
119	הקשר בין המחלקה P ו- NP
11 NP שלמות	11
119	NPH ו- NPC המחלקות
120	בעית הספריות
120	SAT בעית
121	משפט קוק לוין
121	$kSAT$ גרסאות של
124	3SAT בעית
126	הוכחת משפט קוק לוין*
12 רדוקציות פולינומיאליות	12
126	NP היא $CLIQUE$ שלמה
127	בעית הקבוצה הבלתי תליה
128	בעית הכיסוי בקודקודים
129	VC בעיה
129	$PARTITION$
130	רדוקציות פולינומיאליות
130	שפות NP שלמות
13 סיבוכיות מקום ושלמות ב PSPACE	13
130	הגדרה של סיבוכיות מקום
133	משפט סבי'
134	PSPACE המחלקה
135	שלמות ב- $PSPACE$
136	נוסחאות בוליאניות עם כמותים
137	אסטרטגיות מנצחות במשחקים
137	המחלקה Σ

154	המחלקה	NL
154	שלמות ב-	NL
154	שיויון NL ו-	coNL

שיעור 1

מכונות טיורינג

1.1 הגדרה של מכונת טיורינג

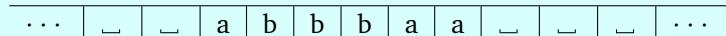
הגדרה 1.1 מבנות טיריניג (הגדרה היוריסטית)

הקלט והסרט

- מוכנות טירוג'ינג (מ"ט) קורא קלט.
 - הקלט עצמו נמצא נמצוא על סרט אינסופי מחולק למשבצות.
 - כל תו של הקלט כתוב במשבצת אחת של הסרט.
 - במכונת טירוג'ינג אנחנו מניחים שהסרט אינסופי לשני הצדדים.

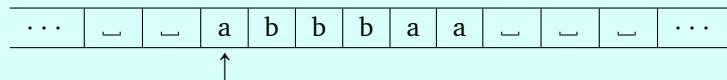
* משמאלי לתחילת הקלט יש רצף אינסופי של תווים רווח " ".

* מימין לסוף הקלט יש רצף אינסופי של תווים רווח " ".



הראש

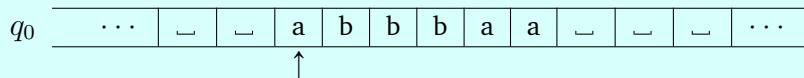
- במצב ההתחלתי הראש בקצת השמאלי של הקלט.



- הראש יכול לזרז 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

- בהרבה החלטות השרות כאשר מימינו נמצא רצף אינסופי של תווים – ים.
 - בראש מכביע על התא הראשון בשרות והמכונה נמצאת במצב התחלתי q_0 .



- בכל צעד חישוב, בהתחסן במצב הנוכחי ולאות שמתוחת לראש (הטו הנקרה), המכונה מחייבת:
 - * לאיזה מצב בעבר
 - * מה לכתוב מתוחת לראש (הטו הנכתב)
 - * لأن להזיז את הראש (תא אחד ימינה, או תא אחד שמאלה, או להישאר במקום).
 - למcona ישם שני מצבים מיוחדים:
 - * אם במשך הריצה המכונה מגיעה ל- q_{acc} היא עוברת ומקבלת.
 - * אם במשך הריצה המכונה מגיעה ל- q_{rej} היא עוברת ודוחה.
 - אם המכונה לא מגיעה ל- q_{acc} או q_{rej} היא תמשיך לרווץ לנצח.

הגדרה 1.2 מכונת טיורינג

מכונת טיורינג (מ"ט) היא שביעה

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

כasher:

Q	קובוצת מצבים סופית ולא ריקה
Σ	אלפבית הקלט
Γ	אלפבית הסרט
δ	פונקציית המעברים
q_0	מצב התחלתי
q_{acc}	מצב מקבל יחיד
q_{rej}	מצב דוחה יחיד

דוגמה 1.1

בנייה מכונת טיורינג אשר מקבלת מילה אם היא בשפה

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

ז"א השפט כל המילים עם מספר שווה אותיות a ו b . הפאודו-קוד של המכונה, כדלקמן.**פאודו-קוד**

1) סורקים את הקלט משמאול לימין.

- אם לא מצאנו a וגם לא מצאנו $b \Leftarrow$ מקבלת.
- אם האות הראשונה שהראש הוא מצא a , כתובים עליו ✓, וועברים לשלב (2).
- אם האות הראשונה שהראש הוא מצא b , כתובים עליו ✓, וועברים לשלב (3).

2) ממשיכים לוז ימינה עד שנמצא b תואם.

- אם לא מצאנו $b \Leftarrow$ דוחה.
- אם מצאנו b כתובים עליו ✓, חוזרים לתחלת הקלט וחוזרים לשלב (1).

3) ממשיכים לוז ימינה עד שנמצא a תואם.

- אם לא מצאנו $a \Leftarrow$ דוחה.
- אם מצאנו a כתובים עליו ✓, חוזרים לתחלת הקלט וחוזרים לשלב (1).

כעת נתן הגדרה פורמלית של המכונה טיורינג שמבצעת את האלגוריתם זהה.

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

כasher Q הקבוצה המצביעים הבאה:

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

המשמעותם של כל המצביעים נרשמים בטבלה למטה:

q_0	ה מצב ההתחלתי. אליו נחזיר אחרי כל סבב התאמת של זוג אותיות.
q_a	מצב שבו ראיינו a ומחפשים b תואם.
q_b	מצב שבו ראיינו b ומחפשים a תואם.
q_{back}	מצב ששנשתמש בו כדי לחזור לказה השמאלי של הקלט ולהתחל את הסריקה הבאה (סבב ההתאמת הבא).
q_{acc}	מצב מקבל.
q_{rej}	מצב דוחה.

האלפבית של הקלט, Σ , והלפבית של השרת, Γ , הינם:

$$\Sigma = \{a, b\}, \quad \Gamma = \{a, b, _, \checkmark\}.$$

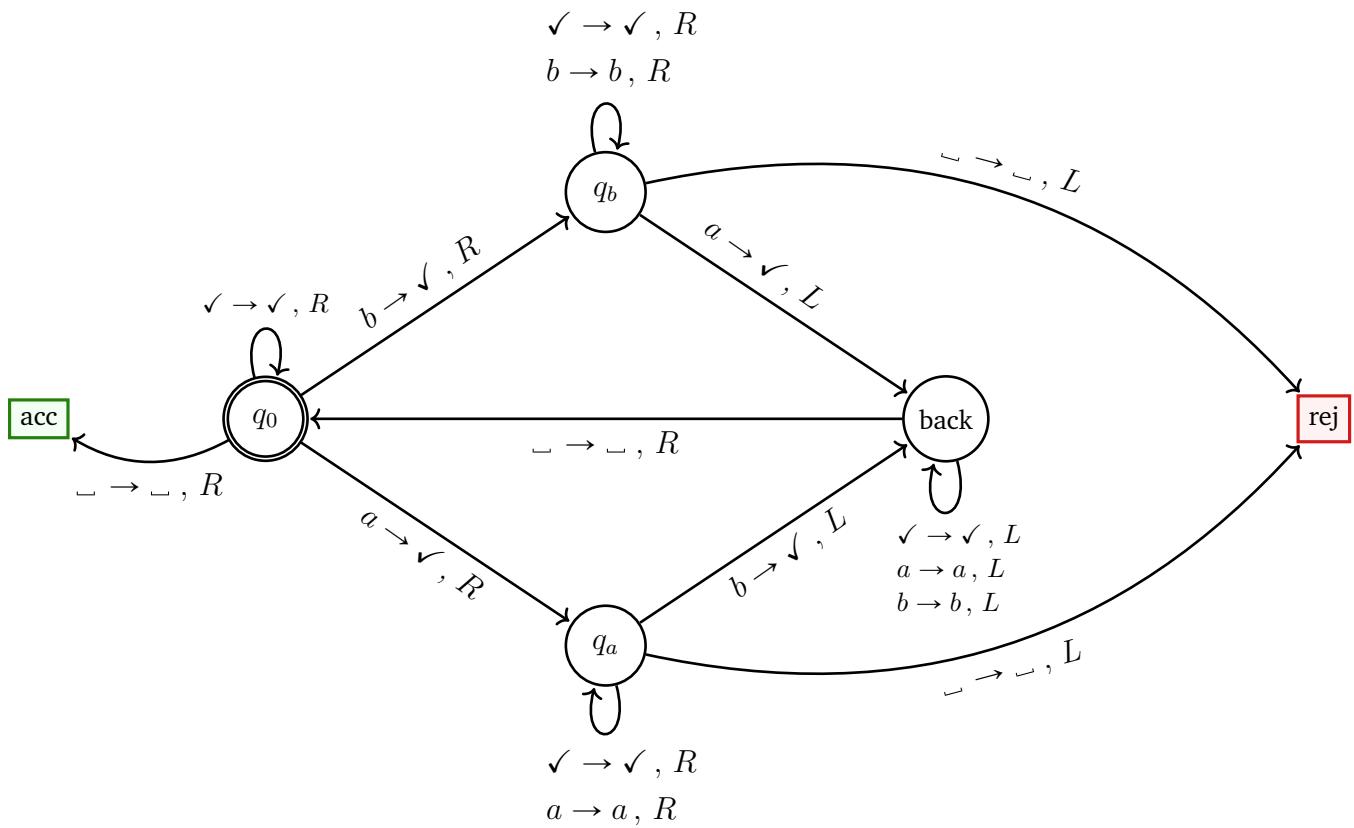
הפונקציית המעברים δ היא מוגדרת כדלקמן.

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

לעתים קל יותר לרשום את פונקציית המעברים δ בטבלה:

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

תרשים מצבוי



דוגמה 1.2

בדקו אם המכוון טיורינג של הדוגמה 1.1 מקבלת את המילה `aab`.

פתרון:

-	q_0	a	a	b	-
-	✓	q_a	a	b	-
-	✓	a	q_a	b	-
-	✓		q_{back}	a	✓
-	q_{back}	✓	a	✓	-
q_{back}	-	✓	a	✓	-
-	q_0	✓	a	✓	-
-	✓	q_0	a	✓	-
-	✓	✓	q_a	✓	-
-	✓	✓	✓	q_a	-
-	✓	✓	rej	✓	-

דוגמה 1.3

בדקו אם המכוון טיורינג של הדוגמה 1.1 מקבלת את המילה `.abbbbaaa`.

פתרון:

-	q_0	a	b	b	b	a	a	-
-	✓	q_a	b	b	b	a	a	-
-	q_{back}	✓	✓	b	b	a	a	-
q_{back}	-	✓	✓	b	b	a	a	-
-	q_0	✓	✓	b	b	a	a	-
-	✓	q_0	✓	b	b	a	a	-
-	✓	✓	q_0	b	b	a	a	-
-	✓	✓	✓	q_b	b	a	a	-
-	✓	✓	✓	b	q_b	a	a	-
-	✓	✓	✓	✓	q_{back}	b	✓	a
-	✓	✓	✓	q_{back}	✓	b	✓	a
-	✓	q_{back}	✓	✓	b	✓	a	-
q_{back}	-	✓	✓	✓	b	✓	a	-
-	q_0	✓	✓	✓	b	✓	a	-
-	✓	q_0	✓	✓	b	✓	a	-
-	✓	✓	q_0	✓	b	✓	a	-
-	✓	✓	✓	q_0	b	✓	a	-
-	✓	✓	✓	✓	q_b	✓	a	-
-	✓	✓	✓	✓	✓	q_b	a	-
-	✓	✓	✓	✓	q_{back}	✓	✓	-
-	✓	✓	✓	q_{back}	✓	✓	✓	-

—	✓	✓	q_{back}	✓	✓	✓	✓	—
—	✓	q_{back}	✓	✓	✓	✓	✓	—
—	q_{back}	✓	✓	✓	✓	✓	✓	—
q_{back}	—	✓	✓	✓	✓	✓	✓	—
—	q_0	✓	✓	✓	✓	✓	✓	—
—	✓	q_0	✓	✓	✓	✓	✓	—
—	✓	✓	q_0	✓	✓	✓	✓	—
—	✓	✓	✓	q_0	✓	✓	✓	—
—	✓	✓	✓	✓	q_0	✓	✓	—
—	✓	✓	✓	✓	✓	q_0	✓	—
—	✓	✓	✓	✓	✓	✓	q_{acc}	—

הגדרה 1.3 קונפיגורציה

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

$uq\sigma v$

כאשר משמעות:

$$u, v \in \Gamma^*, \quad \sigma \in \Gamma, \quad q \in Q.$$

מצב המכוונה,	q
הסימן במקומות הראש	σ
תוכן הסרט משמאלי לראש,	u
תוכן הסרט מימין לראש.	v

דוגמה 1.4 (המשך של דוגמה 1.2)

u	q	σ	v
—	q_0	a	a b —
— ✓	q_a	a	b —
— ✓ a	q_a	b	—
— ✓	q_{back}	a	✓ —
—	q_{back}	✓	a ✓ —
—	q_{back}	—	✓ a ✓ —
—	q_0	✓	a ✓ —
— ✓	q_0	a	✓ —
— ✓ ✓	q_a	✓	—
— ✓ ✓ ✓	q_a	—	—
— ✓ ✓ ✓	q_{rej}	✓	—

דוגמה 1.5

בנו מכונת טיריניג אשר מקבלת כל מילה בשפה

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

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

פתרונות:

ראשית נשים לב למשפט הבא:

1.1 משפט

מספר שלם n שווה לחזקה אי-שלילית של 2, כלומר ($k \geq 0$) אם ורק אם קיימים שלם m ובראשו חילוק של $n - 2^m$ נקבע נותן.

הוכחה:
↳ כיוון

$\cdot \frac{n}{2^k} = 1$ $\Leftrightarrow n = 2^k$ ($k \geq 0$)

כיוון \Rightarrow

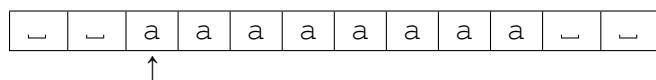
אם קיימים $m \geq 0$ עבורו $n = 2^m$ אז $\frac{n}{2^m} = n$ ולכן n שווה לחזקה אי-שלילית של 2.

לאורו המשפט הזה נבנה אלגוריתם אשר מחלק את מספר האותיות במילה ב- 2 שוב ושוב באורה איטרטיבית.

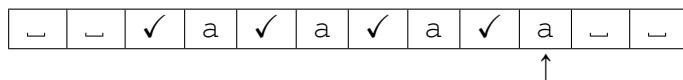
- אם אחרי סיבוב מסוים נקלט מספר אי-זוגי שונה מ-1, אז אין מצב שמספר האותיות a הוא חזקה של 2.
 - בצד שני אם אחרי סיבוב כלשהו נקלט בדיקות a אחת הנשארת, z^a אחרי מספר מסוים של חילוקים של המספר אותיות a קיבלנו 1, אז מובטח לנו שהמספר של אותיות a הוא שווה לחזקה של 2.

כעת נסביר כיצד המכונית טירינג מבצעת את השיטה הזאת בפועל כלהלן.

¹⁾ במאכג התחלתי יש מחרוזת של רצף אותיות *a* כתובות על הסרט והראש נמצא מתחת האות הראשוונה.



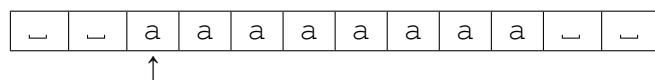
2) עוברים על הקלט משמאלי לימיון ומבצעים מחייקה לשירוגין של האות a. כלומר, אותן אחת נמחק ואות אחת נשאיר וכן הלאה, עד שmaguius לקצתה הימין של המילה.



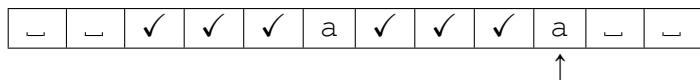
3) אחרי שהראש הגיע לסוף המילה:

- אם מצאנו את a אחות בדיק \Leftarrow המכונה מקבל.
 - אם כתוב ✓ בתו האחרון \Leftarrow המכונה תדחה.
 - אחרת, אם כתוב a בתו האחרון הרש charozot וחוורים לשלב 2).

כדוגמה של מילה המתקבלת על ידי האלגוריתם, למטה רשומות האיטרציות של האלגוריתם זהה על המילה $w = aaaaaaaaa$. במצב התחלתי הסרט נראה כדלקמן.

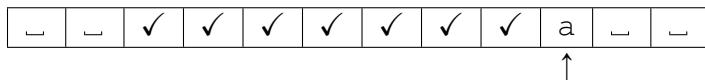


התו האחרון a אז ממשיכים לאיטרציה הבא.



איטרציה 2) בסוף האיטרציה $i = 2$ הסדרת נראית כך:

התו הראשון הוא a אז ממשיכים לאיטרציה הבאה.

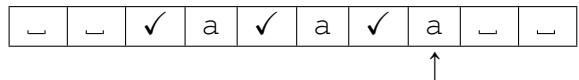
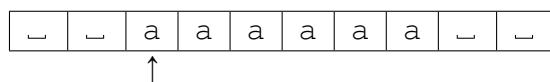


איטרציה 3) לאחר האיטרציה $i = 3$ הסדרת נראית כך:

התו האחרון הוא a אז ממשיכים לאיטרציה הבאה.

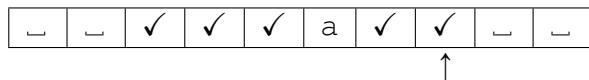
איטרציה 4) באיטרציה $i = 4$ יש אות a אחת בדיק אז המכונה מקבלת.

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



איטרציה 1) לבסוף האיטרציה $i = 1$ הסדרת נראית כך:

התו האחרון a אז ממשיכים לאיטרציה הבאה.



איטרציה 2) לבסוף האיטרציה $i = 2$ הסדרת נראית כך:

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

מצב `none`: מצב ההתחלתי. עדין לא קראנו a כתוצאה זה.

מצב `one`: קראנו a בודד.

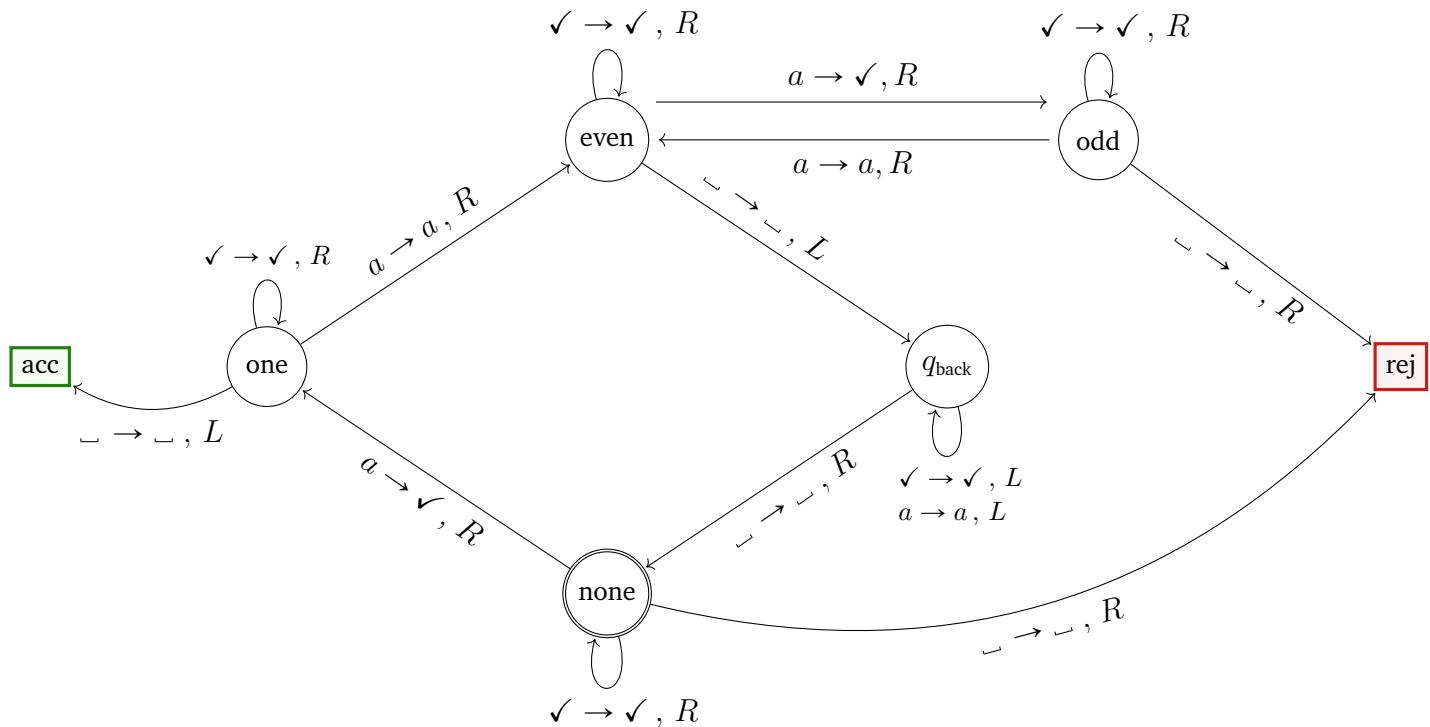
הפונקציית המעברים מתוארת על ידי התרשים

מצב `even`: קראנו מספר זוגי של a .

מצב `odd`: קראנו מספר אי-זוגי של a .

מצב q_{back} : חזרה שלמה.

מצבים למטה.

**דוגמה 1.6**

בדקו אם המילה $aaaa$ מתתקבלת על ידי המכונת טיורינג בדוגמה 1.5.

פתרונות:

[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	✓]
[back	✓	a	✓	a]
[none	✓	a	✓	a]
[✓	none	a	✓	a]
[✓	✓	one	✓	a]
[✓	✓	✓	one	a]
[✓	✓	✓	a	even]
[✓	✓	✓	back	a]
[✓	✓	back	✓	a]
[back	✓	✓	✓	a]
[none	✓	✓	✓	a]
[✓	none	✓	✓	a]
[✓	✓	none	✓	a]

✓	✓	✓	none	a	—
✓	✓	✓	✓	one	—
✓	✓	✓	acc	✓	—

<i>u</i>	<i>q</i>	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 —
—	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.

פתרונות:

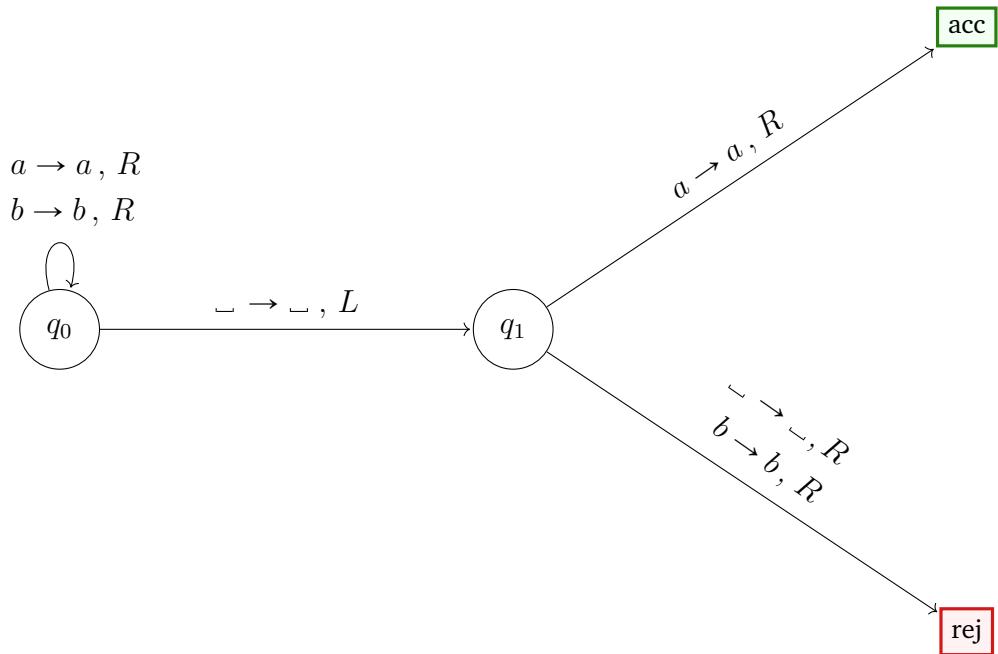
none	a	a	a	—
✓	one	a	a	—
✓	a	even	a	—
✓	a	✓	odd	—
✓	a	✓	—	rej

<i>u</i>	<i>q</i>	σ	v
—	none	a	aa —

— ✓	one	a	a —
— ✓ a	even	a	—
— ✓ a ✓	odd	—	—
— ✓ a ✓ —	rej	—	—

דוגמה 1.8

מהי השפה של המוכנה למטה:

**פתרון:**

1) סורקים את הקלט משמאל לימין.

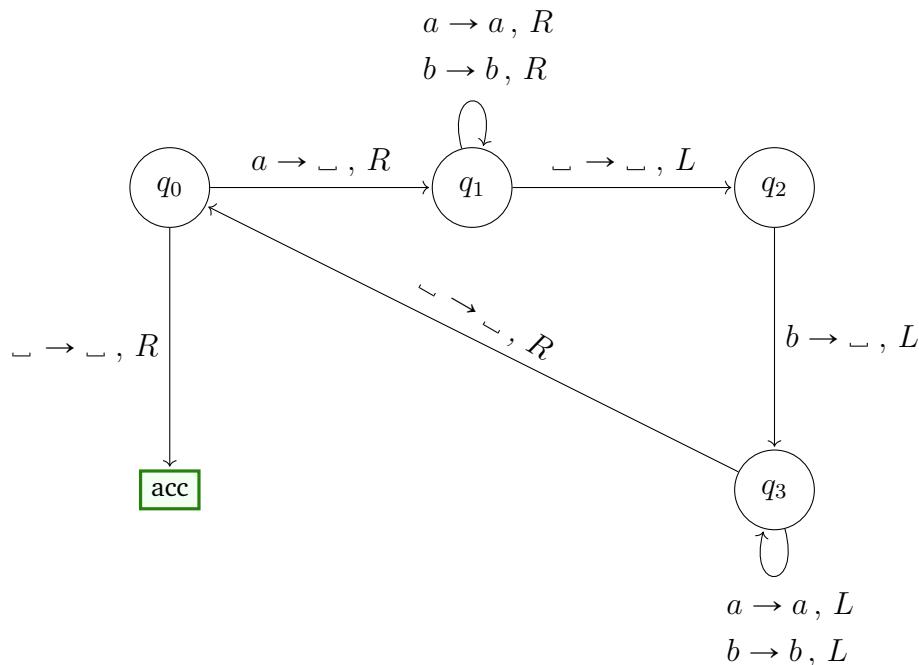
- אם הtau הנקרא a או b עוברים לו ימינה הבא וחוזרים לשלב 1).
- אם הtau הנקרא $—$ אז הגענו לסוף הקלט, ועוברים לשלב 2).

2) עוברים שמאלה לתוך הארון של המילה.

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

לכן המוכנה מקבלת שפת המילים המסתויימות באות a .**דוגמה 1.9**

מהי השפה של המוכנה למטה:



פתרונות:

1) במצב ההתחלתי:

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

2) עוברים ימינה עד שמנגנים 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 וחזרת ומורידה תו b תואם בסוף המילה. בכל איטרציה אם המכונה לא מוצאת b תואם בסוף המילה היא דוחה המילה. אחרת אם המכונה לא דוחה המילה וכל האותיות נמחקוות אז המילה מתתקבלת. לכן המכונה מקבלת שפת המילים

$$\{a^n b^n | n \geq 0\} .$$

הגדרה 1.4 גירה בצעד אחד

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

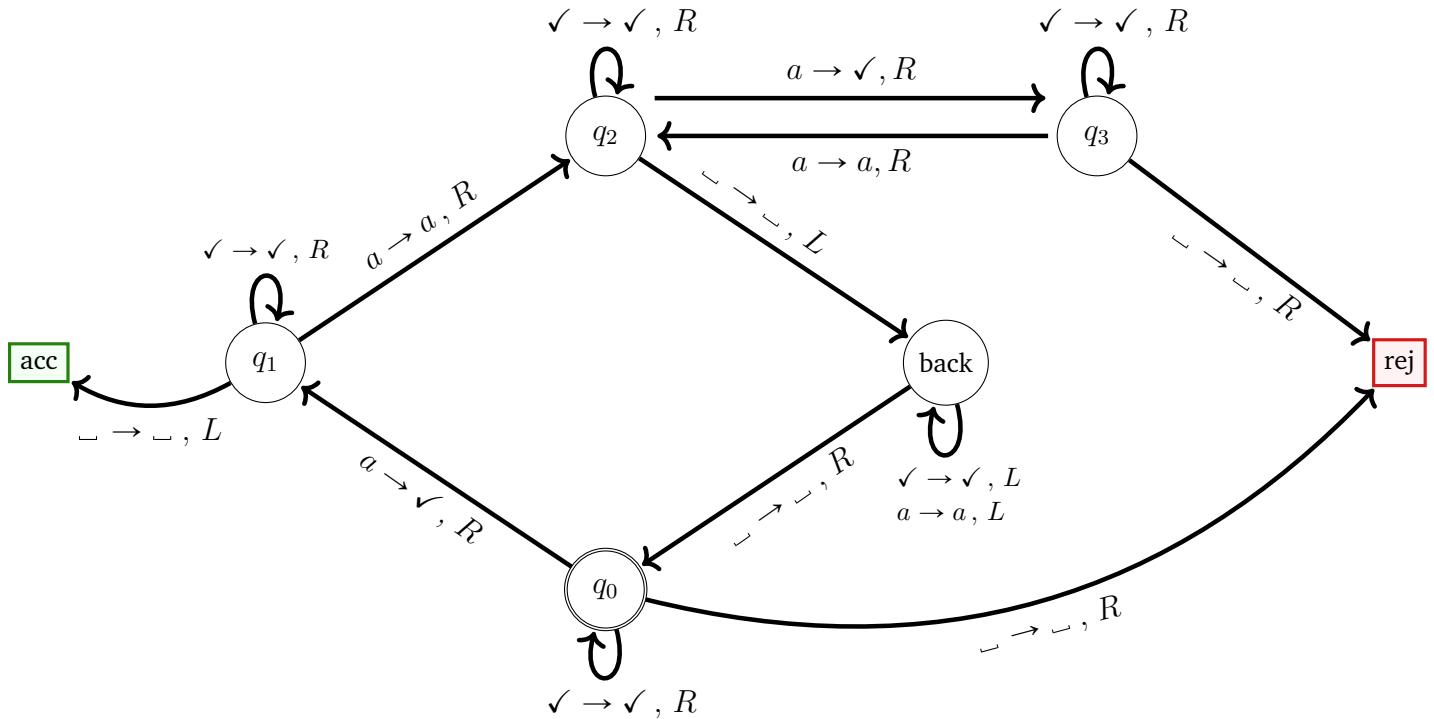
$$c_1 \vdash_M c_2$$

(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 המكونת טיורינג שמתוארת בתרשים למטה מתקיים

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

**הגדרה 1.5 גירירה בכללי**

תהי $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}})$ מكونת טיורינג, ותהינה 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 המكونת טיורינג שמתוארת בתרשים למטה מתקיים

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

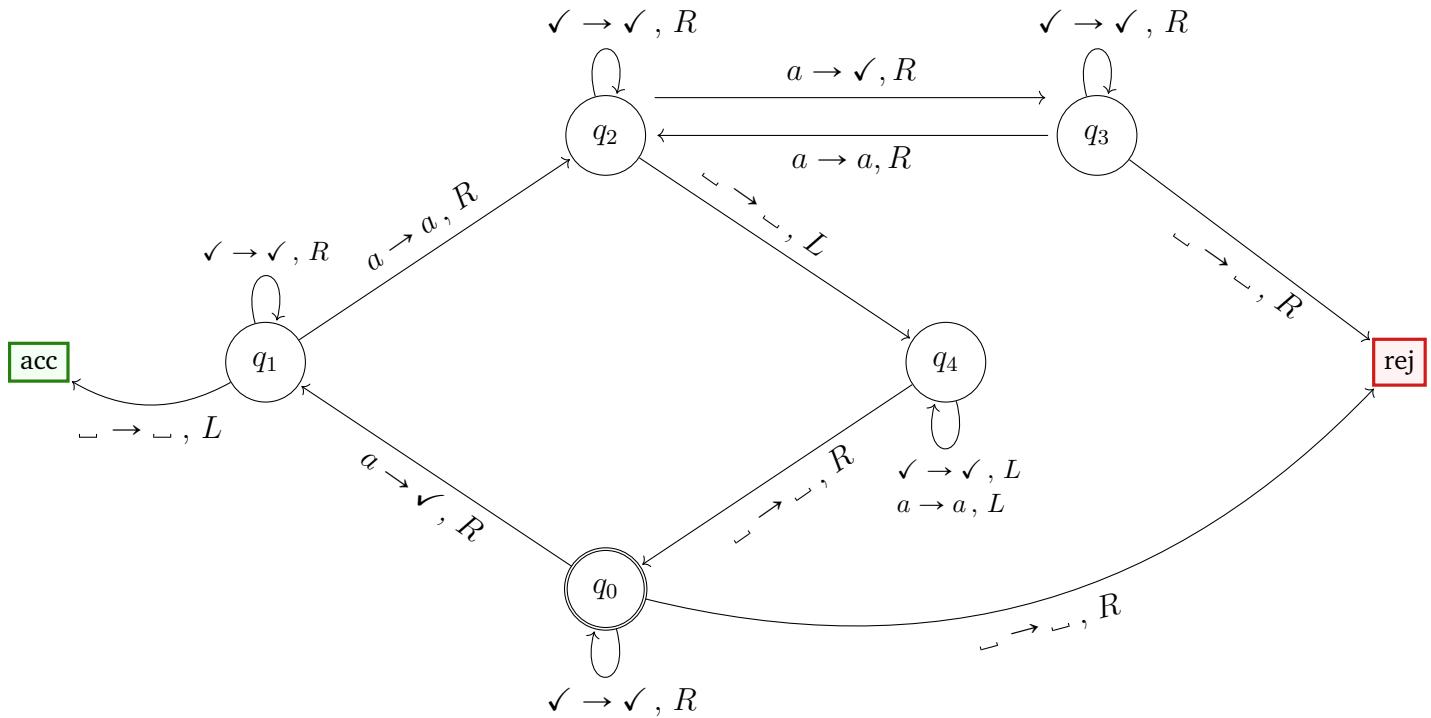
בגלל ש:

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

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

$$\vdash_M \checkmark \checkmark a q_2 _$$

$$\vdash_M \checkmark \checkmark q_4 a .$$



הגדירה 1.6 קבלת ודוחיה של מחרוזות

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

- **מקבלת את w אם**

$$q_0 w \vdash_M^* u q_{\text{acc}} \sigma v$$

כאשר $\Gamma^* v, \sigma \in \Gamma$ כלשהם.

- **דוחה את w אם**

$$q_0 w \vdash_M^* u q_{\text{rej}} \sigma v$$

כאשר $\Gamma^* v, \sigma \in \Gamma$ כלשהם.

הגדירה 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:

- M מקבלת את w .

- M דוחה את w .

הגדירה 1.8 קבלת 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 $w \in L$ מקבלת את w .

- אם $w \notin L$ לא מקבלת את w .

במקרה זה כאשר M מקבלת את השפה L , נכתב ש-

$$L(M) = L .$$

1.2 טבלת המעברים

דוגמה 1.12

בנו מכונה טיורינג שמכריעת את השפה

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

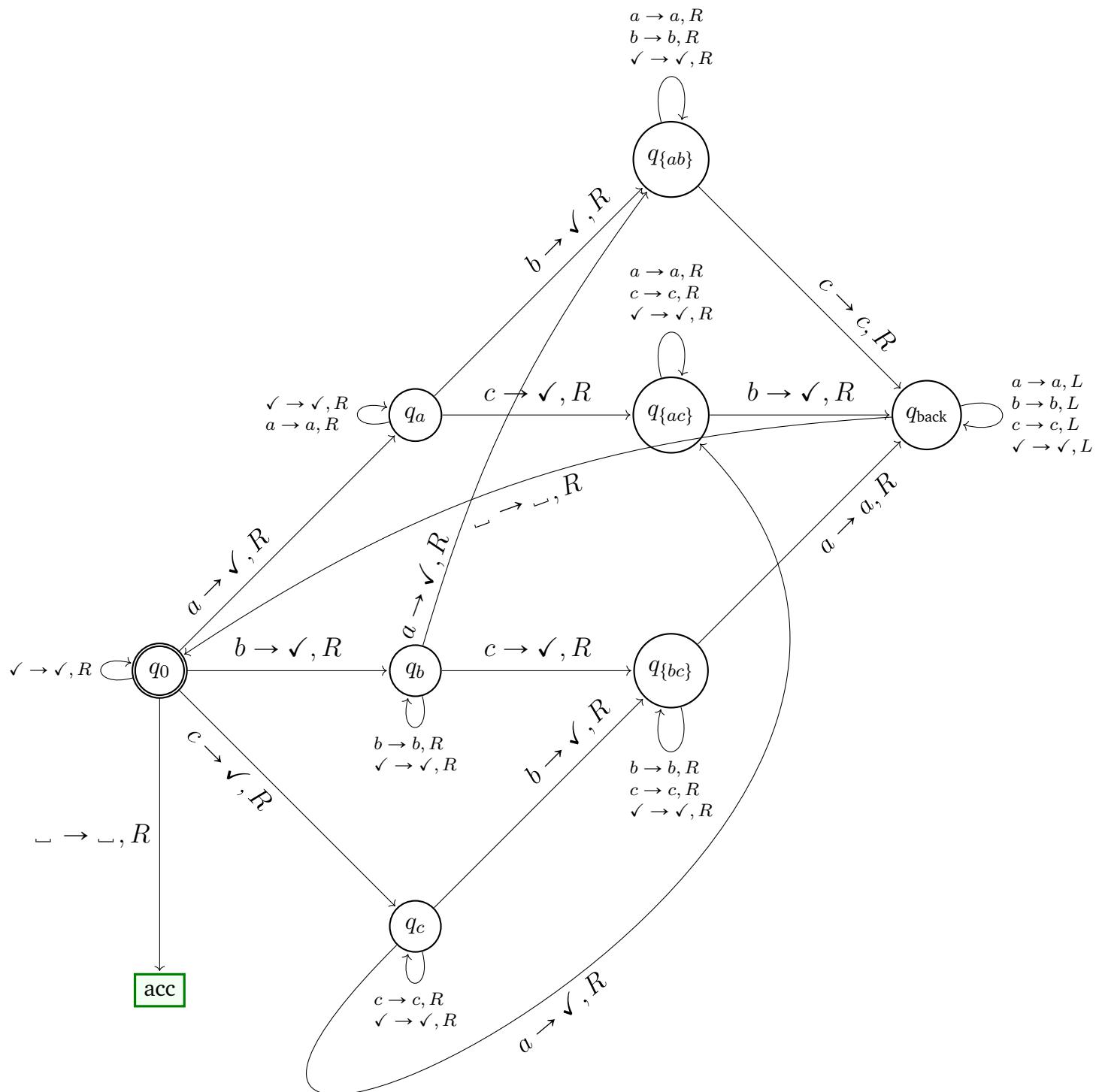
פתרון:

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

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

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

כעת נתאר את המכונה על ידי תרשימים המצביעים של המכונה:



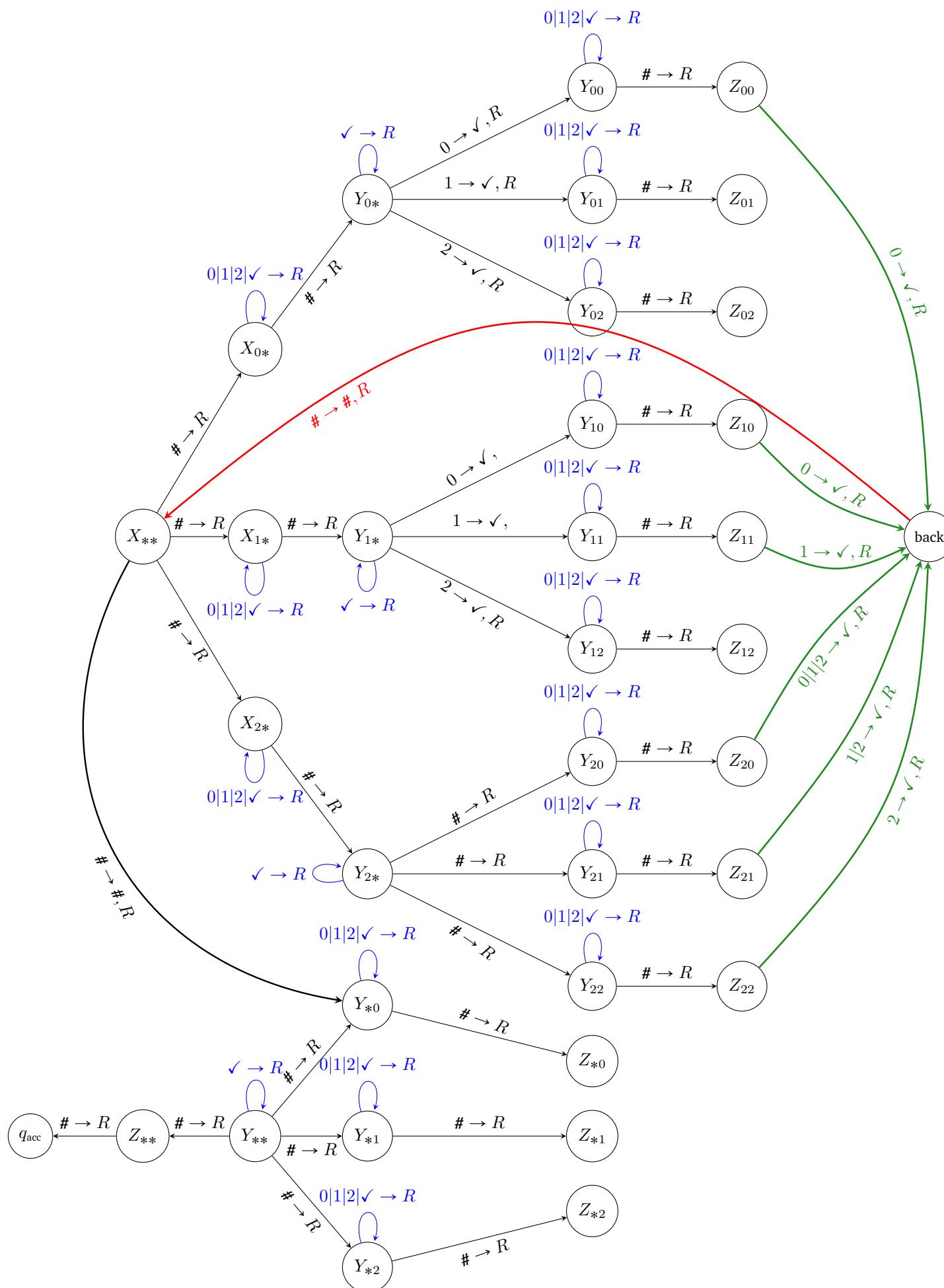
דוגמה 1.13

בנו מכונת טיורינג שמקreira את השפה

$$\{x_1 \dots x_k \# y_1 \dots y_k \# z_1 \dots z_k \mid x_i, y_i, z_i \in \{0, 1, 2\}, \forall i, x_i \geq y_i \geq z_i\}$$

פתרונות:

מצב	סימון בסרט	מצב חדש	כתיבה	תזואה	תנאי
$X * *$	σ	$X\sigma*$	✓	R	
$X * *$	✓	$X * *$	✓	R	
$X\sigma*$	0, 1, 2, ✓	$X\sigma*$	∅	R	
$X\tau*$	#	$Y\tau*$	∅	R	
$Y\tau*$	σ	$Y\tau\sigma$	∅	R	
$Y\tau*$	✓	$Y\tau*$	∅	R	
$Y\tau\sigma$	0, 1, 2, ✓	$Y\tau\sigma$	∅	R	
$Y\tau_1\tau_2$	#	$Z\tau_1\tau_2$	∅	R	
$Z\tau_1\tau_2$	✓	$Z\tau_1\tau_2$	∅	R	
$Z\tau_1\tau_2$	σ	q_{back}	✓	L	
$Z * *$	—	q_{acc}	∅	R	$\tau_1 \geq \sigma \geq \tau_2$
q_{back}	0, 1, 2, ✓	q_{back}	∅	L	
q_{back}	—	$X * *$	∅	R	



1.3 חישוב פונקציות

הגדרה 1.9 מכונת טיורינג שמחשבת פונקציה f

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

- $\Sigma_2 \subset \Gamma$ ו- $\Sigma = \Sigma_1$.
- לכל $q_0 w \vdash q_{\text{acc}}, f(w) \in \Sigma_1^*$ מתקיים.

דוגמה 1.14 חיבור אונרי

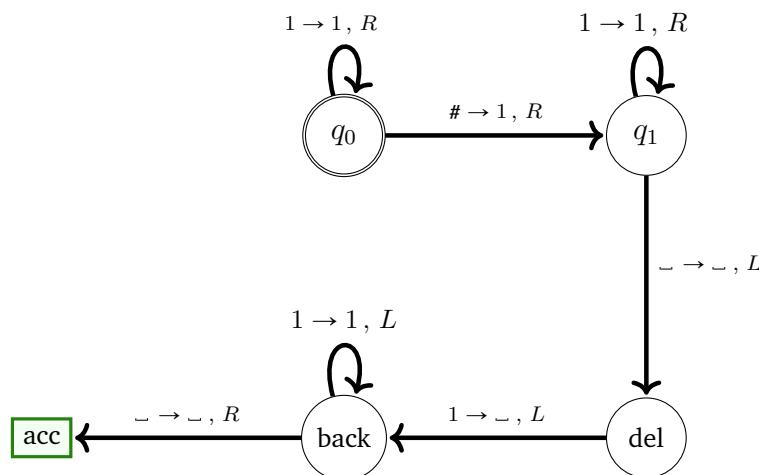
בנו מכונת טיורינג אשר מקבלת את הקלט

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

ומחזירה את פלט

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

פתרונות:



דוגמה 1.15 כפל אונרי

בנו מכונת טיורינג אשר מקבלת את הקלט

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

ומחזירה את פלט

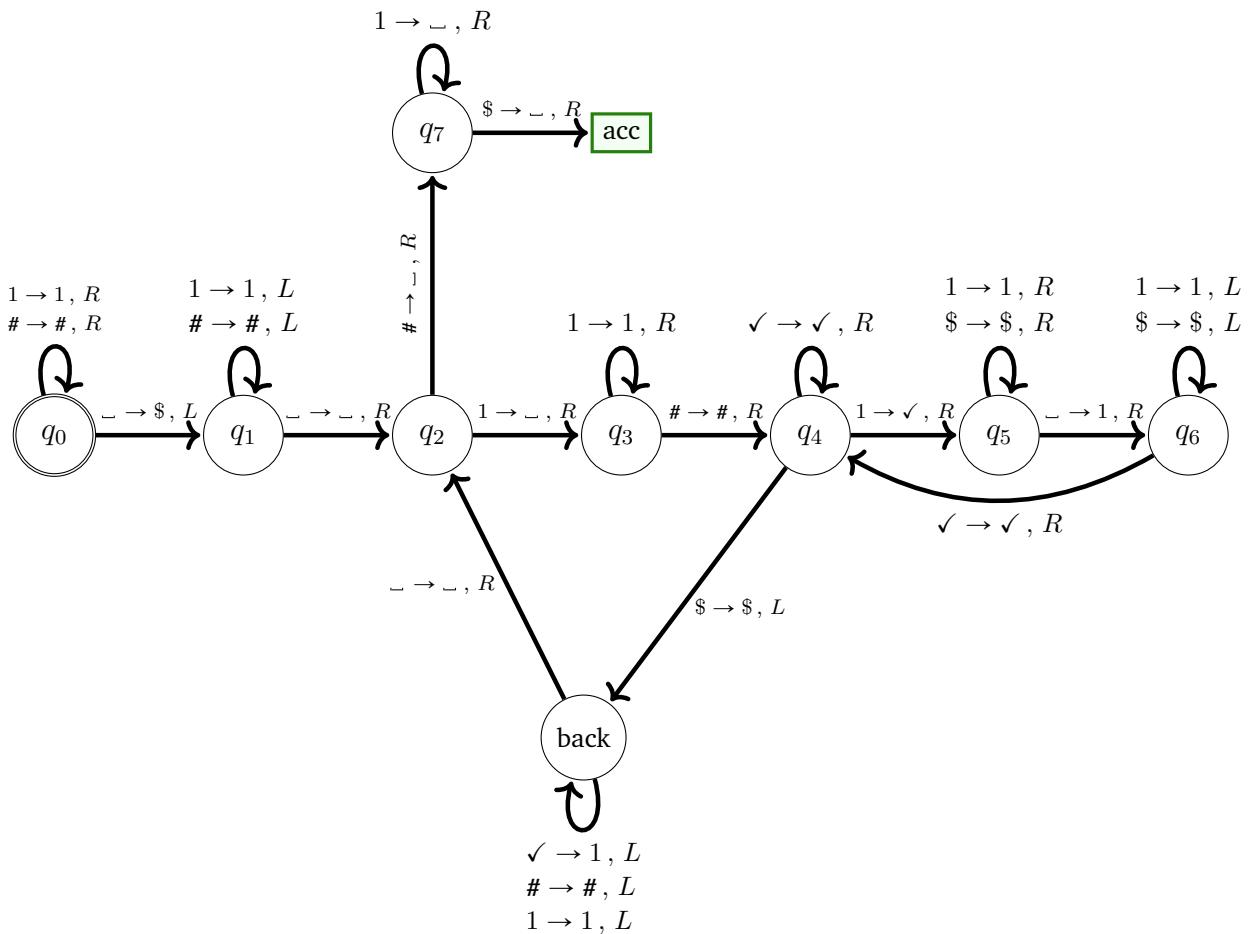
$$1^{i \cdot j}.$$

פתרונות:

- 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 ונוסף שם את התו \$.
לאחר מכן נחזור לתחילת הקלט.
- על כל אות 1 במילה השמאלית נעתק את המילה הימנית לאחר סימן ה- \$.
- לאחר מכן נשאיר רק את התווים שלאחר סימן ה \$. כמובן, נמחק את כל מה שאינו פלט.



μ	q	σ	ν
$\underline{\quad}$	q_0	1	$1\#11_\underline{\quad}$
$\underline{\quad}11\#11$	q_1	$\underline{\quad}$	$\underline{\quad}$
$\underline{\quad}11\#11$	q_1	\$	$\underline{\quad}$
$\underline{\quad}$	q_1	$\underline{\quad}$	$11\#11\$$
$\underline{\quad}$	q_2	1	$1\#11\$$
$\underline{\quad}\underline{\quad}$	q_3	1	$\#11\$$
$\underline{\quad}\underline{\quad}1\#$	q_4	1	$1\$$
$\underline{\quad}\underline{\quad}1\#\checkmark$	q_5	1	\$
$\underline{\quad}\underline{\quad}1\#\checkmark 1\$$	q_5	$\underline{\quad}$	$\underline{\quad}$
$\underline{\quad}\underline{\quad}1\#\checkmark 1\1	q_6	$\underline{\quad}$	$\underline{\quad}$
$\underline{\quad}\underline{\quad}1\#$	q_6	\checkmark	$1\$1\underline{\quad}$
$\underline{\quad}\underline{\quad}1\#\checkmark$	q_4	1	$\$1\underline{\quad}$
$\underline{\quad}\underline{\quad}1\#\checkmark\checkmark$	q_5	\$	$1\underline{\quad}$
$\underline{\quad}\underline{\quad}1\#\checkmark\checkmark\1	q_5	$\underline{\quad}$	$\underline{\quad}$
$\underline{\quad}\underline{\quad}1\#\checkmark\checkmark\11	q_6	$\underline{\quad}$	$\underline{\quad}$

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

שיעור 2

מודלים חישוביים שקולית

הגדירה 2.1 מודל חישובי

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

הגדירה 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 ו רק קיימת מ"ט במודל B שמכריעת את L .
- 2) קיימת מ"ט במודל A שמקבלת את L אם ו רק קיימת מ"ט במודל B שמקבלת את L .

דוגמה 2.1

נסמן ב- T את מודל המכונה הטיורינג הבסיסי.

במודל זה בכלל צעד ניתן לוז ימינה או שמאלה. אך לא ניתן להישאר במקום, באותה המשבצת הסרט. במודל זה, הסרט הוא אינסופי לשני הצדדים. בתחילת החישוב הראש נמצא בתחילת הקלט.

נסמן ב- O את מודל המכונה הטיורינג עם סרט ימינה בלבד.

במודל זה בכלל צעד ניתן לוז ימינה או שמאלה. אך לא ניתן להישאר במקום, באותה המשבצת הסרט. במודל זה, הסרט הוא אינסופי לכיוון אחד בלבד - ימינה. בתחילת החישוב, הקלט ממוקם בקצתה השמאלי של הסרט והראש נמצא בתחילת הקלט. החישוב מתנהל כמו במכונה במודל T , למעט כאשר הראש נמצא באותה המשבצת השמאלית ביותר הסרט וצריך לוז שמאלה - במקרה זה הראש נשאר במקום ולא זו.

הוכיחו כי המודל T והמודל O שקולים חישובית.

פתרון:

יש להוכיח ש:

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

כיוון ראשון

נווכיח כי לכל מ"ט במודל O קיימת מ"ט שcolaה במודל T . כלומר:

נתונה $(Q^O, \Sigma^O, \Gamma^O, \delta^O, q_0^O, acc^O, rej^O)$ במודל O .

نبנה $M^T = (Q^T, \Sigma^T, \Gamma^T, \delta^T, q_0^T, acc^T, rej^T)$.

- רכיבי המ"ט M^T זהים לאלו של המ"ט M^O , מלבד מהתוכנה שהראש של M^O לעולם לא זו מעבר לказחה השמאלי של הקלט.
- נבודד רק עם צד ימין של הסרט האינסופי של M^T ואז M^T תהיהcolaה ל- M^O .

- כדי לדאוג שהראש של המוכונה הדו-כיוונית M^T לא יעבור לכמה השמאלי של הקלט, נוסיף מצבים חדשים וגם מעברים חדשים לפונקציית המעברים של M^T , שmbטיחים שהראש של M^T לא יעבור לכמה השמאלי של הקלט, באופן הבא.
- בתחילת כל חישוב, המוכונה M^T מסמנת את המשבצת מצד שמאל וליד המשבצת הראשונה של הקלט בסימן מיוחד \$.
לכן, המוכונה M^T השקולה למוכונה M^O היא
- ונדר את הפונקציית המעברים של M^T כך שכל פעם שהראש מגע למשבצת המסומנת \$, הראש חוזר ימינה למשבצת הראשונה של הקלט, כמפורט בטבלה המעברים למטה.

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

והפונקציית המעברים מתוארת בטבלה למטה.

מצב	סימון	מצב חדש	כתיבה	позזה	תנאי
q_0^T	σ	$q_\T	\emptyset	L	
$q_\T	-	q_0^O	\$	R	
q	\$	q	\$	R	$\forall q \in Q^O$

הוכחנו את הכיוון הראשון:
ראינו מוכנה דו-כיוונית השקולה למוכנה חד-כיוונית.

כעת נוכיח את הטענה בכיוון השני:
נראה מוכנה חד-כיוונית השקולה למוכנה דו-כיוונית.

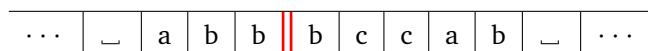
כיוון שני

נוכיח כי לכל מ"ט במודול T קיימת מ"ט שקופה במודול O . כלומר:

נתונה $M^T = (Q^T, \Sigma^T, \Gamma^T, \delta^T, q_0^T, acc^T, rej^T)$ במודול T .

נבנה $M^O = (Q^O, \Sigma^O, \Gamma^O, \delta^O, q_0^O, acc^O, rej^O)$ שקופה במודול O .

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



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

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

\$	b	b	a	—	—	—	—	...
b	c	c	a	b	—	—	—	...
...								

- * 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 :

—	a \rightarrow	b \rightarrow		b \rightarrow	b \rightarrow	c \rightarrow	c \rightarrow	a \rightarrow	— \rightarrow
---	-----------------	-----------------	--	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

תזוזה שකולה ב- M^O :

\$	\leftarrow b	\leftarrow b	\leftarrow a	—	—	—	—
b \rightarrow	c \rightarrow	c \rightarrow	a \rightarrow	b \rightarrow	— \rightarrow	—	—

- * 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 :

—	a \leftarrow	b \leftarrow		b \leftarrow	b \leftarrow	c \leftarrow	c \leftarrow	a \leftarrow	— \leftarrow
---	----------------	----------------	--	----------------	----------------	----------------	----------------	----------------	----------------

תזוזה שתקולה ב- M^O :

\$	\rightarrow b	\rightarrow b	\rightarrow a	—	—	—	—
b \leftarrow	c \leftarrow	c \leftarrow	a \leftarrow	b \leftarrow	— \leftarrow	—	—

לכן, המכונה M^O השકולה למכונה M^T היא

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

סביר את כל הרכיבים של M^O :

- לכל מצב $q \in Q^T$ נגיד q_U ו- q_D של Q^O , כדי לבדוק בין המצביעים שבהם הראש נמצא בחלק העליון לבין המצביעים שבהם הראש נמצא בחלק התיכון של הסרט.

$$\cdot \Sigma^O = \Sigma^T \bullet$$

$$\cdot \Gamma^O \subseteq (\Gamma^T \times \Gamma^T) \cup \{\$\} \bullet$$

$$\cdot q_{\text{acc}}^O = q_{\text{acc}}^T \bullet$$

$$\cdot q_{\text{rej}}^O = q_{\text{rej}}^T \bullet$$

- הפונקציות המעבירים δ מותוארכות בטבלת המעברים למטה. בטבלה, הסימנים π, σ, τ מסמנים כל תוארכות אלפביתית Γ^T :

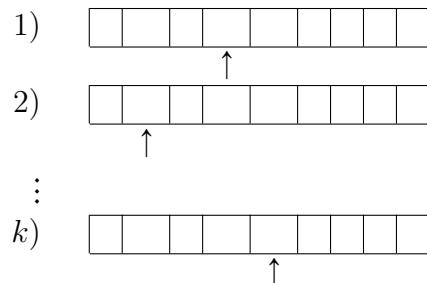
תנאי	תזואה	כתיבה	מצב חדש	סימון	מצב
אותחול					
q_0^O	τ	q_τ	\$	R	$\tau \in \Sigma \cup \{_\}$ $\sigma \in \Sigma$
q_σ	τ	q_τ	$_\sigma$	R	
$q._$	$_$	back	$_$ $_$	L	
q_{back}	$_\tau$	q_{back}	\cap	L	
q_{back}	\$	$q_0^T.D$	\cap	R	
תזואה מקורית שמאלת					
q_D	π σ	p_D	π τ	L	תזואה שמאלת: $(q, \sigma) \xrightarrow{M^T} (p, \tau, L)$
q_U	σ π	p_U	τ π	R	
q_D	$_$	p_D	$_$ τ	L	תזואה שמאלת: $(q, _) \xrightarrow{M^T} (p, \tau, L)$
q_U	$_$	p_U	τ $_$	R	
תזואה מקורית ימינה					
q_D	π σ	p_D	π τ	R	תזואה ימינה: $(q, \sigma) \xrightarrow{M^T} (p, \tau, R)$
q_U	σ π	p_U	τ π	L	
q_D	$_$	p_D	$_$ τ	R	תזואה ימינה: $(q, _) \xrightarrow{M^T} (p, \tau, R)$
q_U	$_$	p_U	τ $_$	L	
פגיעה בקצת					
q_D	\$	q_U	\cap	R	
q_U	\$	q_D	\cap	R	
כל השאר עוברים ל-rej					

שיעור 3

מכונות טיורינג מרובת סרטים

1.3. מכונת טיורינג מרובת סרטים: הגדרה היוריסטית

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



- לכל סרט יש ראש שלו.
- בתחילת העבודה הקלט w כתוב בתחילת הסרט הראשון וכל הסרטים ריקים. הראשים בכל סרט מוצבאים על התא הראשון הסרט, והמכונה נמצאת במצב התחלתי q_0 .
- בכל צעד חישוב, לפי המצב הנוכחי ול- k התווים שמתוחת ל- k הראשים, המכונה מחליט לאיזה מצב לעבור, מה לכתוב מתחת לכל אחד מ- k הראשים ולאן להזיא את הראש בכל אחד מ- k הסרטים.
- הראשים של הסרטים יכולים לנוע באופן בלתי- תלוי בהתאם לפונקציית המעברים של המטמ"ס.

2. מכונת טיורינג מרובת סרטים: הגדרה פורמלית

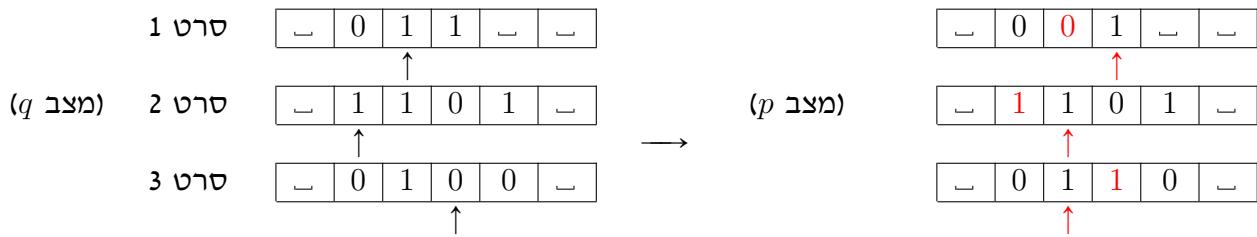
הגדרה 3.1 מכונת טיורינג מרובת סרטים

מכונת טיורינג מרובת סרטים היא שבייעה:

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

כאשר $Q, \Sigma, \Gamma, q_0, q_{\text{rej}}, q_{\text{acc}}$ מוגדרים כמו מ"ט עם סרט יחיד (ראו הגדרה 1.2).
ההבדל היחידי בין מ"ט עם סרט יחיד לבין מטב"ס הוא הפונקציית המעברים. עבור מטמ"ס הפונקציית המעברים היא מצויה הבא:

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

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 סרטיים

הכללה של קונפיגורציה של מ"ט עם סרט יחיד:

$$\begin{pmatrix} u_1 q \ v_1 \\ u_2 q \ v_2 \\ \vdots \\ u_k q \ v_k \end{pmatrix}$$

דוגמה 3.2

בנו מטמ"ס שמכריעת את השפה:

$$L_{w^R} = \{w = \{a, b\}^* \mid w = w^R\}.$$

כלומר שפת הפלינדרומים.

פתרון:

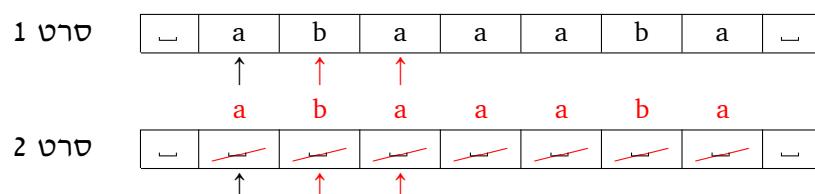
בנייה מ"ט עם שני סרטיים:

תאואר המכונה:

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

על הקלט $w = M_2$:

(1) מעתיקת את w לסרט 2.



(2) מזיהה את הראש בشرط 1 לתו הראשון ב- w ואת הראש בشرط 2 לתו האחרון ב- w .

(3) משווה בין התווים שמתוחת לראשים:

- אם התו שמתוחת לראש 1 הוא $__$

$$\text{.acc} \Leftarrow$$

- אם התווים שמתוחת לראשים שונים $\text{.rej} \Leftarrow$

- אחרת מזיהה את הראש בشرط 1 ימינה ואת הראש בشرط 2 שמאליה, וחוזרת לשלב (3).

הfonקציית המעברים של M_2 היא:

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

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

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

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

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

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

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

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

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

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

תאור המcona:

נסמן M_1 המcona עם סרט ייחיד שמכריעת את השפה L_{w^R}

על הקלט $w = M_1$

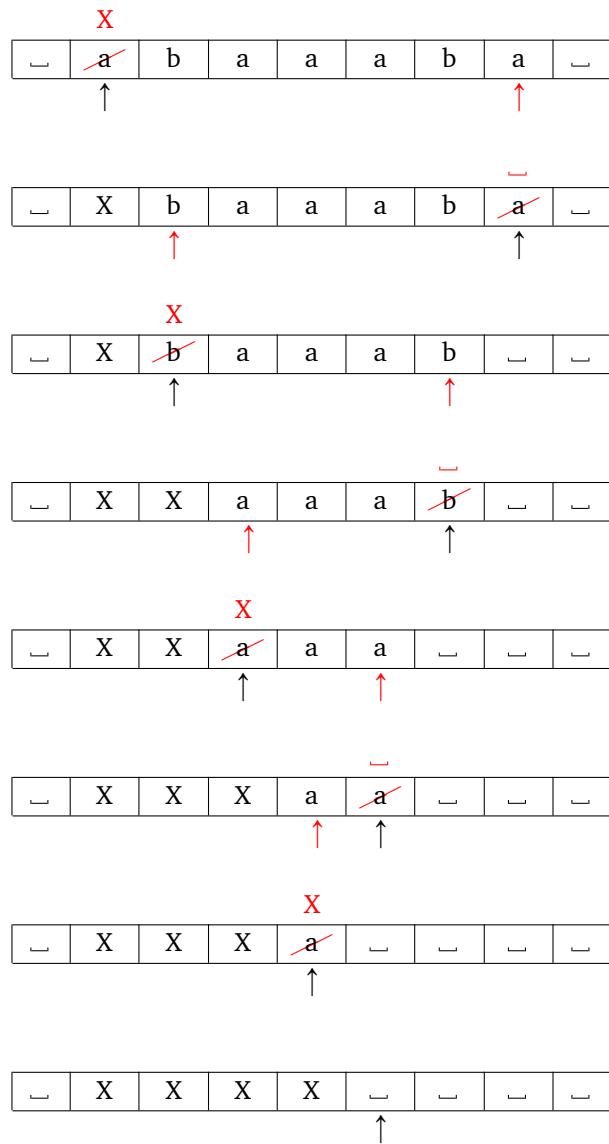
(1) אם התו שמתוחת לראש הוא $__$ אז $.acc \leftarrow M_1$

(2) זוכרת את התו שמתוחת לראש ומוחקתו אותו ע"י X .

(3) מזיהה את הראש ימינה עד התו הראשון משמאל ל- $__$.

• אם התו שמתוחת לראש הוא X $.acc \Leftarrow X$

- אם הトー שונה מהתו שזכורנו $\Leftarrow \text{ rej}$.
- מוחק את הトー שמתהדר בראש ע"י $_$, מזיה את הראש שמאלוה עד הトー הראשון מימין לו $- X$ וחוזרת לשלב (1).



3.4 שיקולות בין מטמ"ס למ"ט עם סרט יחיד

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

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

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

כלומר, לכל קלט $w \in \Sigma^*$:

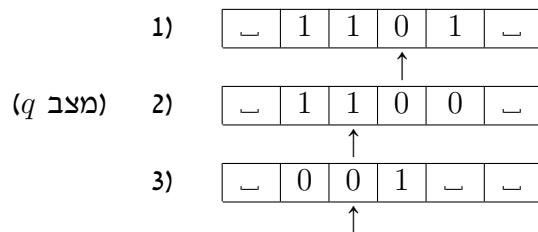
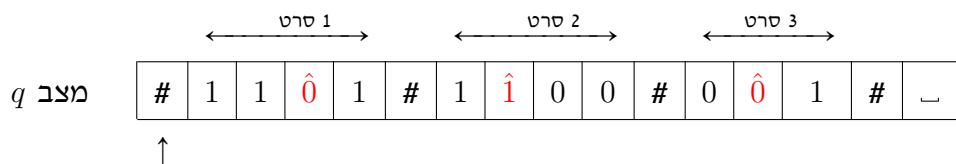
- אם M מקבלת את w $\Leftarrow M'$ מקבלת את w .
- אם M דוחה את w $\Leftarrow M'$ דוחה את w .
- אם M לא עוצרת על w $\Leftarrow M'$ לא עוצרת על w .

הוכחה:

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

רעיון הבנייה:

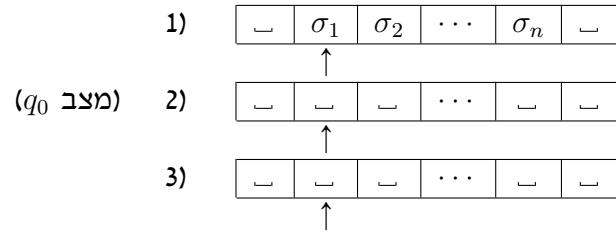
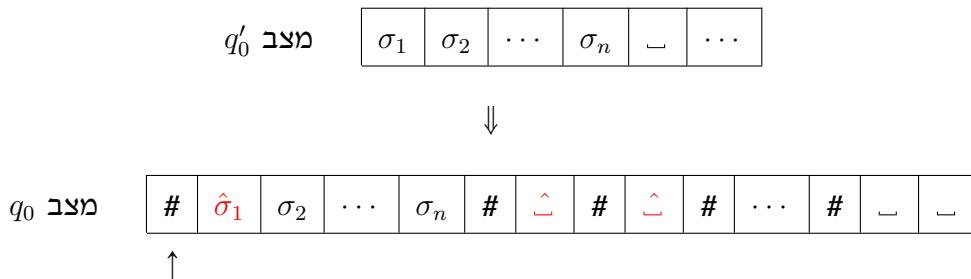
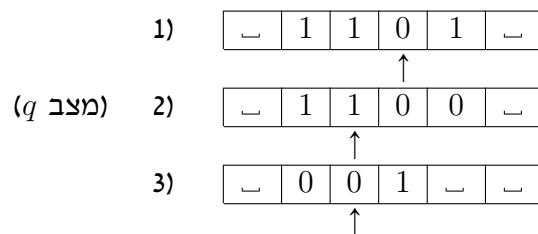
בhinint קלט M' , $w \in \Sigma^*$ תבצע "סימולציה" של ריצה M על w .

 M -ב M' -ב

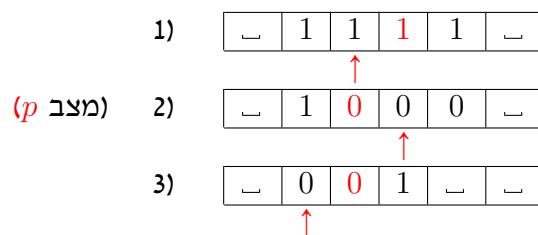
- M' תשמור את התוכן של k הסרטים של M על הסרט, רק שההתוכן של סרט i יופיע בין $\#_i$ ל- $\#_{i+1}$.
- M' תשמור את המיקום של הראשיים של M ע"י הכפלת הא"ב Γ .
כלומר, לכל אות $\alpha \in \Sigma$, M' תשמור שתי אותיות α ו- $\hat{\alpha}$ ב- Γ' , כך ש- $\hat{\alpha}$ תסמן את התו שמתוחת בראש בכל סרט.
- בכל צעד חישוב, M' סורקת את הסרט שלה משמאלי לيمין כדי ללמידה מהם התווים שמתוחת בראשים (התווים שמשמעותם ב- $\hat{\alpha}$).
- M' משתמש בפונקציית המעברים δ_k של M כדי לחשב את המעבר הבא.
- M' סורקת את הסרט שלה משמאלי לימין כדי לעדכן את הסרטים ואת המיקום הראשיים בהם.

תאור הבנייה של M' :**1) שלב האיתחול**

בhinint קלט M' , $w = \sigma_1 \sigma_2 \cdots \sigma_n$ מתחילה את הקונFIGורציה ההתחלתית של M על הסרט שלה.

M -ב M' -ב(2) תאור צעד חישוב של M M -ב

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



M' ב-

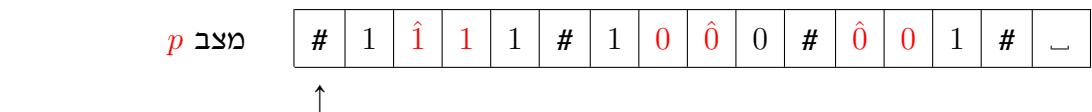
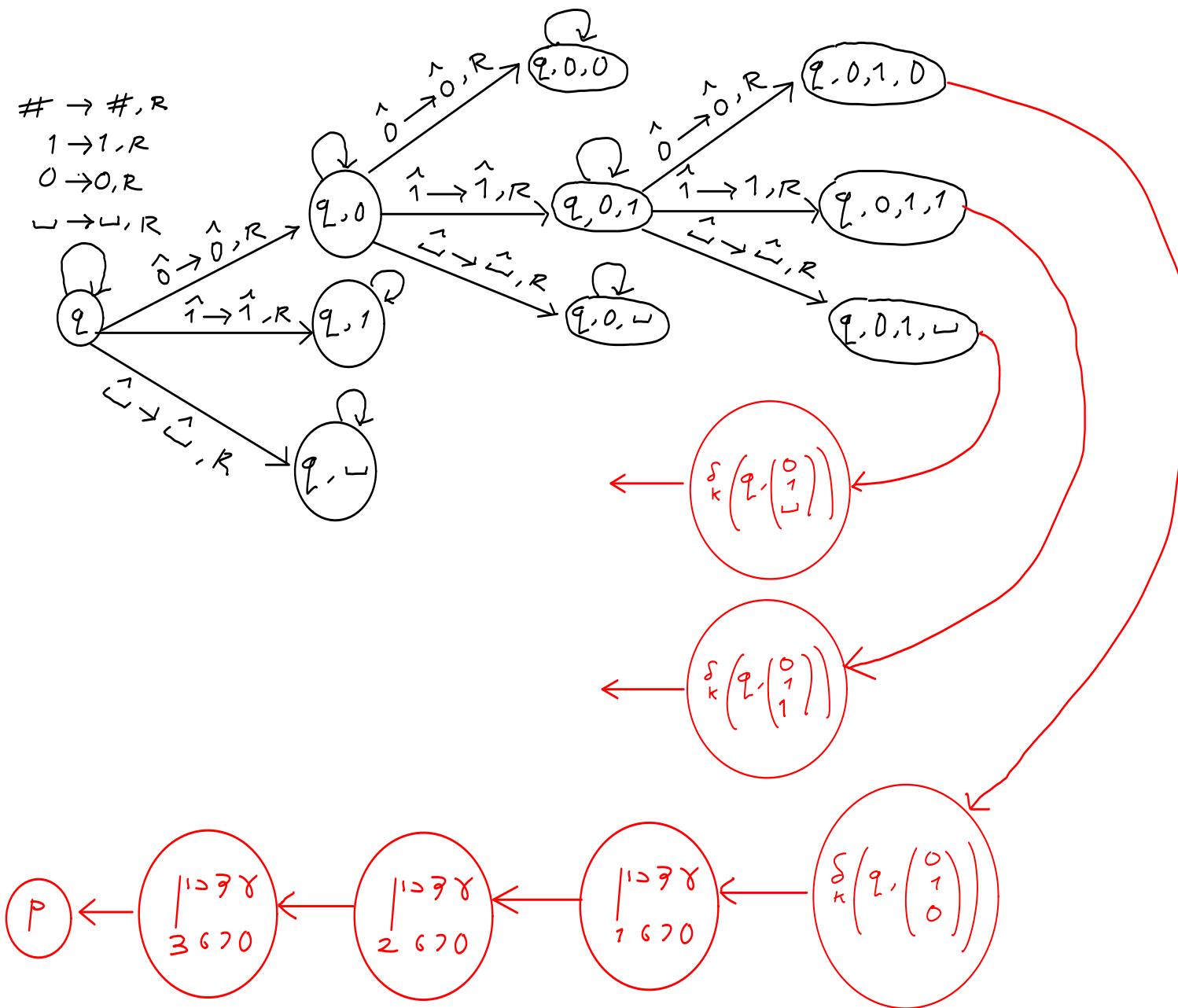
↓

- איסוף מידע
 - M' סורקת את הסרט שלה משמאלי לימיון ומזהה את התווים שמסומנים ב- \hat{a} . מידע זה ניתן לשמר במצבים. לדוגמה:

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

זה אפשרי מכיוון שמספר המצבים הנדרש הוא סופי:

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



• עדכון הסרטים

- אם בכלל שלב M מזיהה אחד מן הראשים הויירוטואליים ימינה אל סימן $\#$, פעולה זו מצינית שמקונת M הזיהה את הראש המתאים אל החלק הריק שטרם נקרא של הסרט. لكن M כותבתתו על המשבצת היזו ומזיהה את כל התוכן של הסרט בין התא הזה לבין התא $\#$ הימני ביותר בתחום אחד ימינה. לאחר מכן מושיכת את הסימולציה כרגע.

שיעור 4

מכונת טיריניג אי-דטרמיניסטיבית

4.1 הגדרה של מכונת טיריניג אי-דטרמיניסטיבית

הגדרה 4.1 מכונת טיריניג אי-דטרמיניסטיבית

מכונת טיריניג אי-דטרמיניסטיבית (מ"ט א"ד) היא שביעיה

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

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

Δ היא פונקציה המעבירים

$$\Delta : (Q \setminus \{q_{\text{acc}}, q_{\text{rej}}\}) \times \Gamma \rightarrow P(Q \times \Gamma \times \{L, R, S\}) .$$

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

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

- קונפיגורציה של מ"ט א"ד זהה לקונפיגורציה של מ"ט דטרמיניסטיבית.
- לכל קונפיגורציה יתכן מספר קונפיגורציות עוקבות.

- לכל מילה $w \in \Sigma^*$ יתכן מספר ריצות שונות:

* ריצות שמנגיעה ל- q_{acc} .

* ריצות שמנגיעה ל- q_{rej} .

* ריצות שלא עוזרות.

* ריצות שנתקעות.

הגדרה 4.2

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

השפה של מ"ט א"ד M היא

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

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

דוגמה 4.1

נתונה השפה

$$L = \{1^n \mid n \text{ איינו ראשוני}\}, \quad \Sigma = \{1\}.$$

בנו מ"ט המכריעה את השפה L .**פתרון:**הרעיוןنبנה מ"ט א"ד N המכריעה את L . N תבחר באופן א"ד מספר $n < t < n$ ותבדוק האם t מחלק את n .

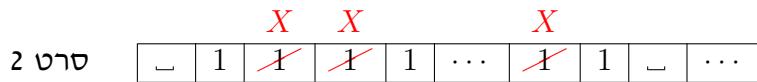
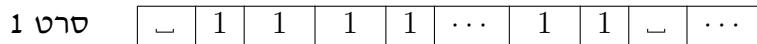
1 סרט 1	$\boxed{_ 1 1 1 1 1 1 1 1 \dots}$	n
---------	--	-----

2 סרט 2	$\boxed{_ 1 1 1 _ \dots}$	t
---------	---------------------------------------	-----

תאור הבניה $w = 1^n$ על קלט $:w = 1^n$ **שלב 1)**

- N בוחרת באופן א"ד מספר $n < t < n$.
- מעתקה את w לסרט 2.
- עוברת על העותק משמאליימין, 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 כמוות ה- 1 -ים שלא נמחקו.



שלב 2) N בודקת האם t שנבחר מחלק את n .

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

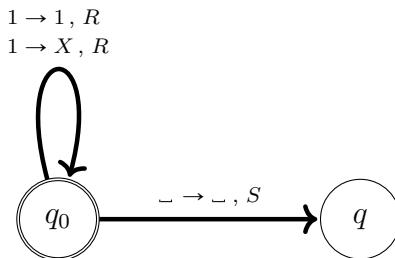
4.2 עץ חישוב של מ"ט א"ד

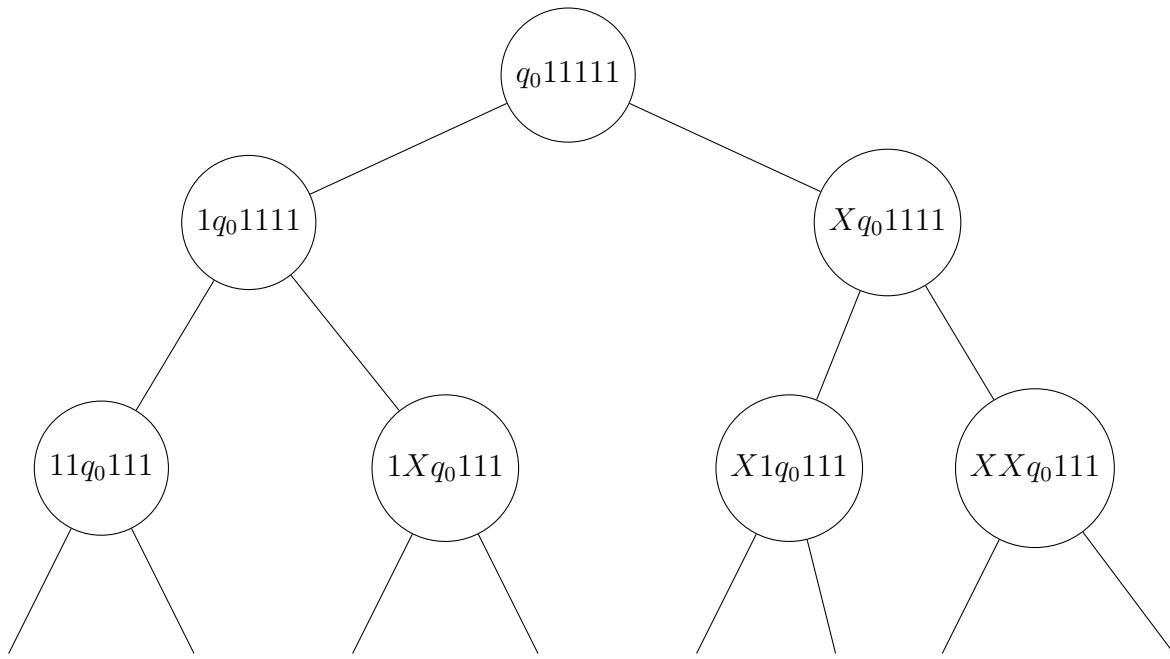
הגדרה 4.5 עץ חישוב של מ"ט א"ד

בהתנון מ"ט א"ד M ומילה $w \in \Sigma^*$, עץ חישוב 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 w_0 .
- 3) לכל קדקוד v בעץ הבנים של v הם כל הקדקודים הנובעים מהקונפיגורציה המתוארת ע"י v .

דוגמה 4.2





4.3 שיקילות בין מכונת טירינג אי-טרמיניניסטיית למכונת טירינג דטרמיניניסטיית

משפט 4.1 שיקולות בין מ"ט א-דטרמיניסטי למ"ט דטרמיניסטי ב-*RE*

לכל מכונת טיריניג אי-דטרמיניסטיבית N קיימת מכונת טיריניג דטרמיניסטיבית D כך ש-

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

כלומר לכל $w \in \Sigma^*$:

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

הובחה: בהינתן מכונת טוירינג אי-דטרמיניניסטית N נבנה מכונת טוירינג דטרמיניניסטית D ונוכיח כי

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

רעיון הוכחה

בהתנition קלט Σ^* , $w \in \Sigma^*$, D תבע ריצה של כל החישובים האפשריים של N על w , ואם אחד החישובים מסוימים ב- D תעדיר ותתקבל.

מכיוון שיתכנו חישובים אינסופיים, לא יוכל לסרוק את עץ החישוב לעומק. במקום זה נסrok את העץ לרוחב. ככלומר, נבדוק את כל החישובים באורך 1, ולאחר מכן נבדוק את כל החישובים באורך 2, וכן הלאה. אם אחד החישובים הסטיים ב- D תעוצר ותתקבל.

תאור הבניה

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

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

אז

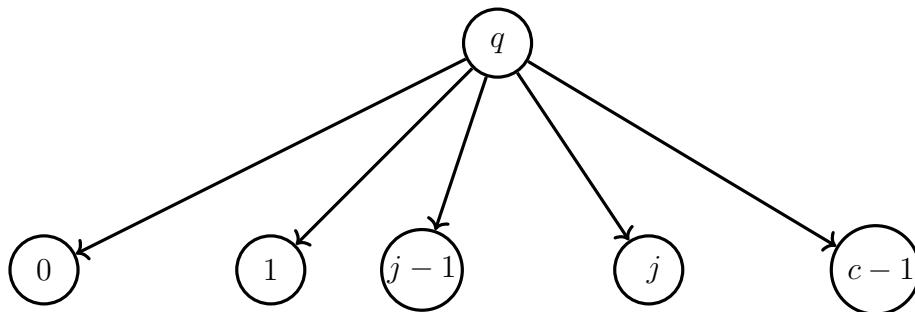
$$|\Delta(q, \alpha)| \leq |Q| \cdot |\Gamma| \cdot |\{L, R, S\}| = 3|Q| \cdot |\Gamma| .$$

נסמן:

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

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

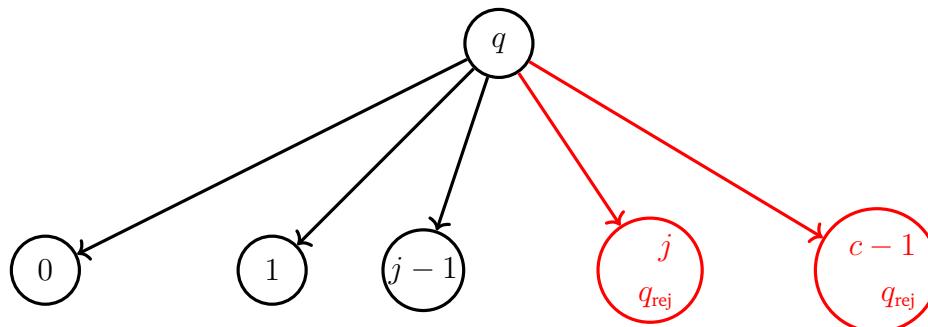
$$\{0, 1, 2, \dots, C - 1\} .$$



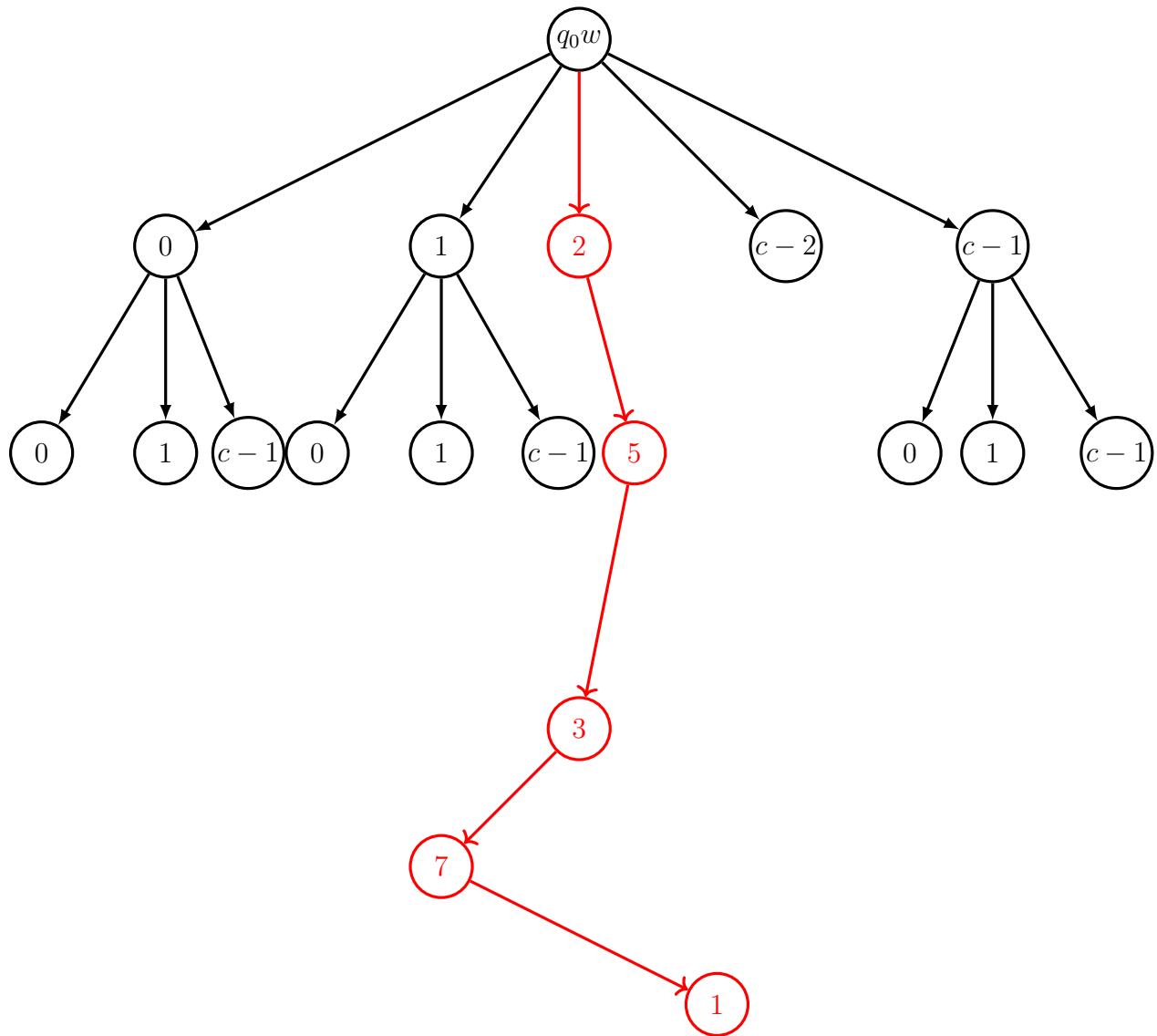
$$, |\Delta(q, \alpha)| = j < C \text{ וא}$$

אי לכל $j \leq k \leq C - 1$

. $k = (q_{\text{rej}}, \alpha, S)$ קבוע



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

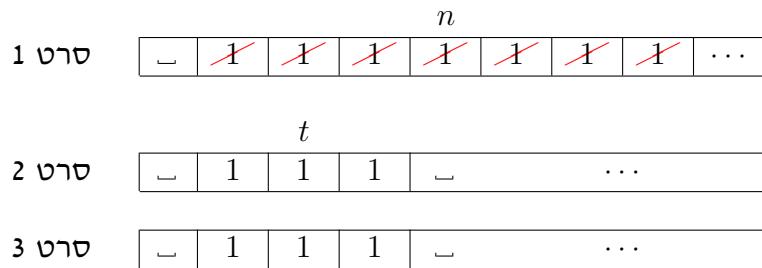


קידום לקסיקוגרפי:

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

הבנייה של D

D מכילה 3 סרטים:



w על קלט D

(1) מתחילה את המחרוזת בסרט 3 ל-0.

(2) מעתקה את w לסרט 2.

(3) מרים את N על w לפי המחרוזת בסרט 3.

• אם N קיבלה את $w \Leftarrow D$ עוצרת ומקבלת.

• אחרת D מוחק את סרט 2, מקדמת את המחרוזת בסרט 3 לקסיקוגרפיה וחזרת לשלב 2)



שיעור 5

התזה של צרץ טירינג ודקזוקים כלליים

5.1 היחס בין הכרעה וקבלה

משפט 5.1 כל שפה כריעה היא גם קבילה

כל שפה כריעת היא גם קבילה.

■ **הוכחה:** המכונה טירינג שמכריעה את L גם מקבלת אותה. נשאל שאלה. האם כל שפה קבילה היא גם כריעת? זאת שאלה שכרגע אין לנו מספיק כלים לענות עליה. נחזור לשאלת זו בפרק הבא. לבינתיים נוכיח טענה חלשה יותר.

משפט 5.2

תהי L שפה.

אם גם L וגם \bar{L} קבילות אז L כריעת.

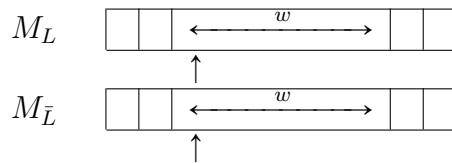
הוכחה: תהי M_L מ"ט שמקבלת את L , ותהי $M_{\bar{L}}$ מ"ט שמקבלת את \bar{L} .
נבנה מ"ט D_L שמכריעת את L .

כיצד תעבוד המ"ט D_L המכריעת?

- נרים במקביל את M_L ואת $M_{\bar{L}}$.

- אם M_L מקבלת את המילה אז נועבור ל-.acc.

- אם $M_{\bar{L}}$ מקבלת את המילה אז נועבור ל-.re.



- הסימולציה מתבצעת ע"י סימולץ צעד צעד.

- * צעד במכונה M_L .

- * צעד במכונה $M_{\bar{L}}$.

- נמשיך בסימולציה המקבילה עד שאחת המכונות מגיעה למצב acc.

- * אם M_L מקבלת $\text{acc} \leftarrow$.
- * אם $M_{\bar{L}}$ מקבלת $\text{rej} \leftarrow$.
- לא יכול להיות מצב כי אף אחת מהמכונות לא מגיעה למצב acc כי כל מחרוזת $L \in w$ או $\bar{L} \in w$.

5.2 שיקולות של מכונת טיורינג ותוכנית מחשב

- מכונת טיורינג היא מודל חישובי למחשב.
- מחשב = תוכנית מחשב.
- תוכנית מחשב כתובה בשפת תכנות, למשל
 - * ג'אווה
 - * פיתון
 - C *
 - SIMPLE *

- המרכיבים של שפת תכנות הם
 - * משתנים
 - * פעולות
 - * תנאים
 - * זרימה

נוכיח כי מכונת טיורינג ותוכנית מחשב שקולים חישובי.

SIMPLE 5.3

משתנים

¹ i , j , k , . . .

- טבעיות:

מקבילים כערך מספר טבעי.

מערכות

- המערכתים אין-סופיים

¹ A [] ,
² B [] ,
³ C [] ,

- בכל תא ערך מתוק א"ב Γ.

אתחול

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

פעולות

- השמה בקבוע:

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

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

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

- פעולות חשבו:

תנאים

- $B[i]==A[j]$ (מערכים).

- $y \geq x$ (משתנים טבעיות).

זרימה

- סדרה פקדות ממושפרות.

- goto : מותנה ולא מותנה.

- עזירה עם ערך חזרה.

```
1 one = 1
2 zero = 0
3 B[zero] = "0"
4 i=0
5 j=i
6 if A[i] == B[zero] goto 9
7 i=j + one
8 goto 3
9 C[one] = A[j]
10 if C[one] == A[zero] goto 12
11 stop(0)
12 stop(1)
```

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

הגדרה 5.1 קבלה ודחיה של מחרוזות 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 ו渴לה של מחרוזות בשפה SIMPLE

עבור שפה L ותוכנית P בשפת SIMPLE. אומרים כי

- P מבירעה את L אם היא מקבלת את המילים שב- L ודוחה את אלה שלא ב- L .
- P מקבלת את L אם היא מקבלת את כל וრק המילים ב- L .

משפט 5.3

המודלים של מכונת טיריניג ותוכנית SIMPLE שקולים.

הוכחה:

כיוון ראשון:

נוכיח כי לכל מ"ט M קיימת תוכנית P שcolaה.
בוצע סימולציה של מ"ט M במחשב P .

בלי להכנס פרטים, די ברור שבשפה עילית, כגון ג'אווה, ניתן להציג מבני נתונים עברור כל מרכיבי מכונת טיריניג:

- הסרט.
- המצביעים.
- מיקום הראש.
- טבלת המעברים.

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

כיוון שני:

נוכיח כי לכל תוכנית P בשפה SIMPLE קיימת מ"ט Shcola.

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

הרכיבים הם:

- משתנים.
- פעולה.
- תנאים.
- זרימה.

משתנים

לכל משתנה יהיה סרט משלו.
המספר שהמשתנה יחזק יוצג בסיס אונרי.
בהתחלת הסרט יהיה רק עם רוחחים, זה מייצג את המספר אפס בסיס אונרי.

לכל מערך יהיה סרט משלו.
בכל תא הסרט המערך תהיה אותן.
בהתחלת כל המערכות יהיו מאופסים למעט הסרט הראשון, שיחזק את הקלט.

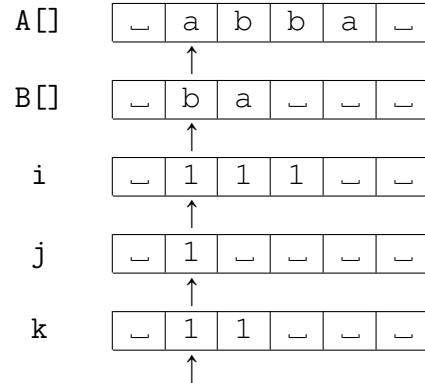
למשל ההשמה הבאה של משתנים בשפה SIMPLE:

```

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

```

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

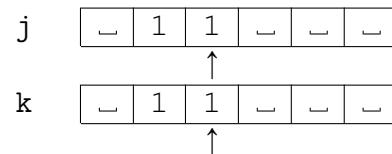
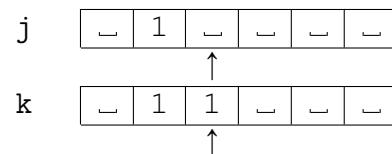
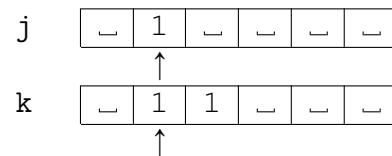


פעולות

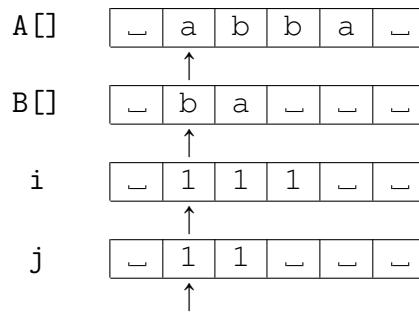
כעת נניח שנשים

```
1 j = k
```

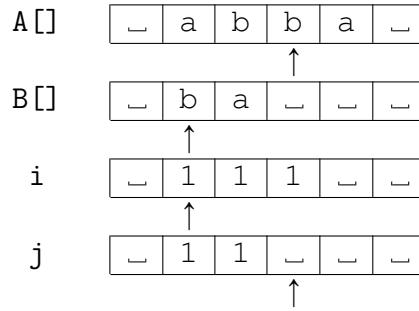
אפשר למשם את ההשמה הזאת על ידי להעתיק את תוכן הסרט של המשתנה k לסרט של המשתנה j.



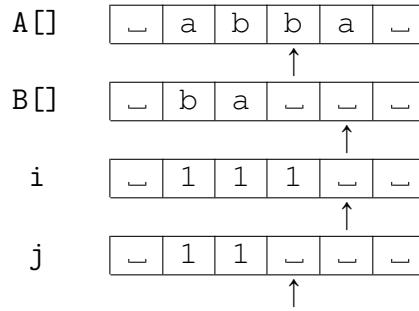
כעת נניח שנשים $[j]=A[i], [B[3]]=A[2], \dots$.
המשם זה במ"ט ע"י להעתיק את תוכן משבצת 2 בסרט של $[A]$ למשבצת 3 בסרט של $[B]$.



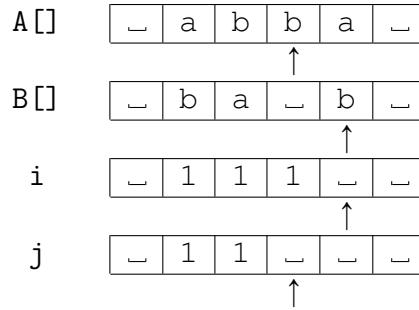
שלב 2



שלב 3



שלב 4



נניח עכשו שאנו רוצים לשים

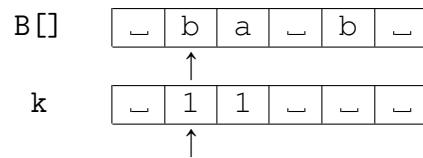
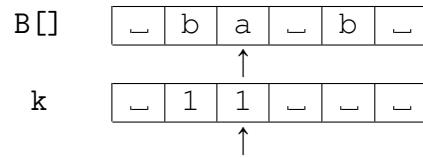
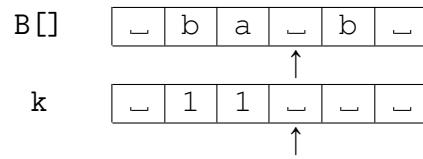
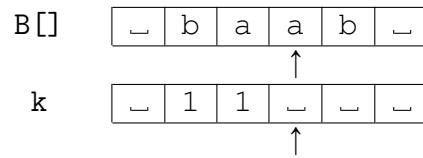
¹ B [k] <- "a"

א"

¹ B [3] <- "a"

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

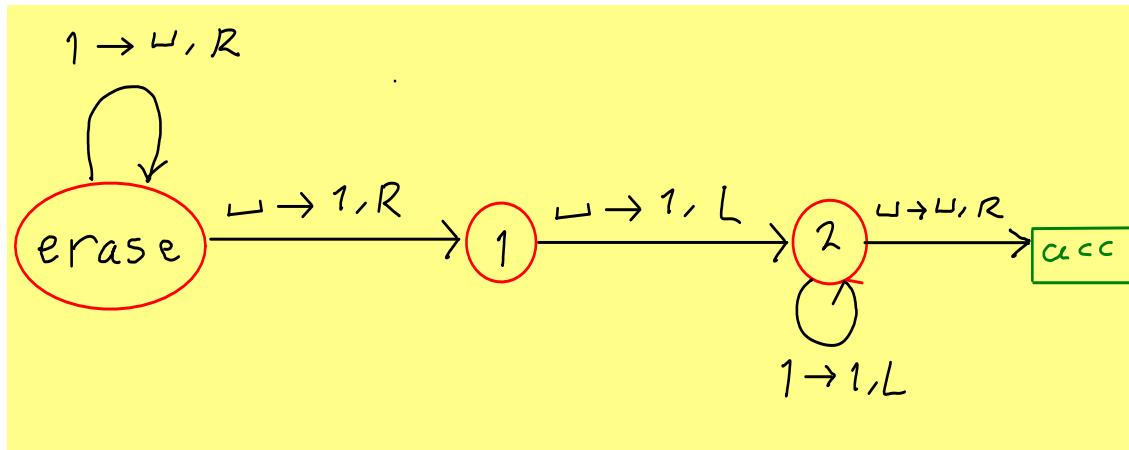
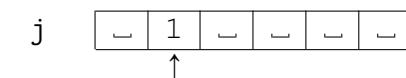
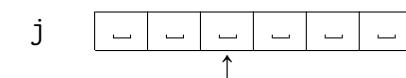
שלב 1

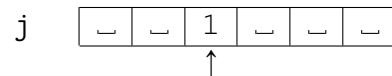
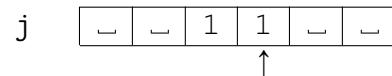
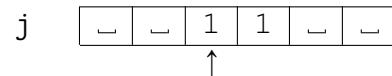
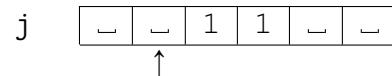
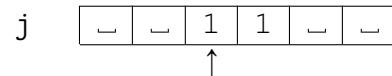
**שלב 2****שלב 3****שלב 4**

cut נניח שאנו רוצים לשים

1 j=2

אז נממש זה במת עם הפעולות הבאות:

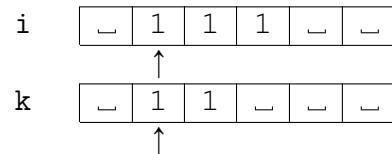
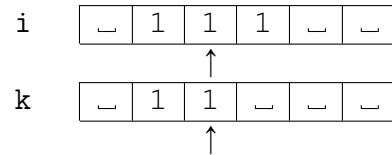
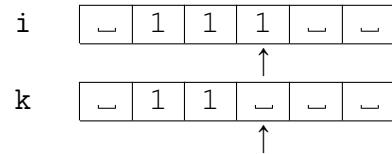
**שלב 1****שלב 2**

שלב 3)**שלב 4)****שלב 5)****שלב 6)****שלב 7)**תנאים

נניח שאנו רוצים למש את התנאי

`i >= k`

נitin לבדוק את התנאי במת' על ידי הפעולות הבאות:

שלב 1)**שלב 2)****שלב 3)****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 אלףיבית.
- Σ קבוצה סופית של **טרמינלים** שמורכב מאותיות קטנות וסימנים של אלףיבית.
- R קבוצה של כללים. כל כלול הוא מצורה

$$\gamma \rightarrow u$$

כאשר $V \in \gamma$ משתנה בודד בצד שמאל ו- $u \in \Sigma^*$ מחרוזת של משתנים וטרמינלים 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 $\{0, 1, \#\}$, המשתנה ההתחלתי הוא $S = A$ והכללים של הדקדוק הם

$$R = \begin{cases} A \rightarrow 0A1 \\ A \rightarrow B \\ B \rightarrow \# . \end{cases}$$

הגדרה 5.4 יצירה של מילה על ידי דקדוק חסר קשר

- 1) כתבו את המשתנה ההתחלתי S .
- 2) מצאו משתנה וכלל אשר מתחליל אם המשתנה זה, והחליפו אותו עם המחרוזת בצד ימין של הכלל.
- 3) חזרו על שלבים 1 ו- 2 עד שלא נשאר אף משתנים של V .

דוגמה 5.2הדקוק G_1 יוצר את המחרוזת $000\#\underline{111}$

$$A \xrightarrow{A \rightarrow 0A1} 0A1 \xrightarrow{A \rightarrow 0A1} 00A11 \xrightarrow{A \rightarrow 0A1} 000A111 \xrightarrow{A \rightarrow B} 00B11 \xrightarrow{B \rightarrow \#} 000\#\underline{111}$$

דוגמה 5.3

נתון את הדקדוק

$$G_2 = (\{S, T, F\}, \{(,), +, \times, a\}, R, S)$$

כאשר הכללים הם

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

: $a + a$ יוצר את המילה: G_2

$$S \xrightarrow{S \rightarrow S+T} S + T \xrightarrow{S \rightarrow T} T + T \xrightarrow{T \rightarrow F} F + F \xrightarrow{F \rightarrow a} a + a$$

בדקdock כללי, גם בצד ימין וגם בצד שמאל יכולה להופיע מחרוזת של משתנים וטרמינליים. פורמלי:

הגדעה 5.5 דקדוקים כלליים

דקדוק חסר קשר הוא קבוצה

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

כאשר

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

$$\gamma \rightarrow u$$

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

דוגמה 5.4

נתון את הדקדוק

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

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

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

:aaaa יוצר את המילה: G

$$\begin{array}{ccccccc} S & \xrightarrow{S \rightarrow [S]} & [S] & \xrightarrow{S \rightarrow [S]} & [[S]] & \xrightarrow{S \rightarrow a} & [[a]] \\ & \xrightarrow{[\rightarrow \varepsilon } & [aa] & \xrightarrow{[a \rightarrow aa[} & aa[a] & \xrightarrow{[a \rightarrow aa[} & aa aa[] \\ & & & & & \xrightarrow{[\rightarrow \varepsilon } & aaaa \end{array}$$

דוגמה 5.5

נתון את הדקדוק

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

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

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

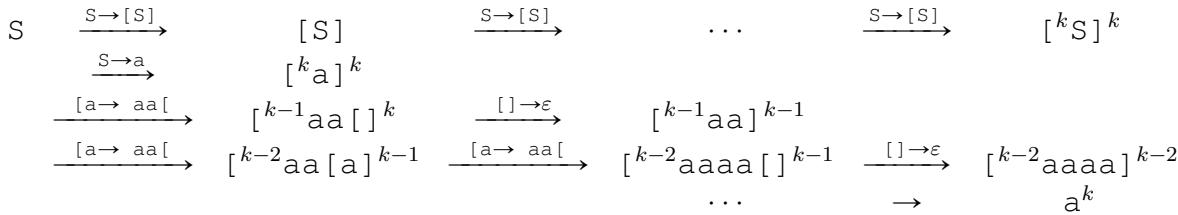
מהן המילים שנitin נוצר בעזרת הדקדוק הזה,
או במילים אחרות: מהי השפה של הדקדוק?

פתרון:

תשובה:

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

הסבר:



דוגמה 5.6

בנו דקדוק כללי אשר יוצר את הפסה

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

פתרון:

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

$$S \rightarrow abS , \tag{1}$$

$$ab \rightarrow ba , \tag{2}$$

$$ba \rightarrow ab , \tag{3}$$

$$S \rightarrow \varepsilon . \tag{4}$$

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

שימוש לב: בדקדוק כללי אנו מאפשרים גם כליל'יצרה בהם מצד שמאל יש רק טרמינלים.
לכן, ניתן גם שנמשיך ונפתח מחזורות של טרמינלים. למשל

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

נשאל שאלה כללית:

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

דוגמה 5.7

בנו דקדוק כללי שיוצר את השפה

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

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.

נעזר את האותיות c, a, b יחד.

נעsha זאת בצורה כזו שכדי לסיים את תהליך הגזירה יש לסדר את האותיות בסדר הרצוי:

תחילה a ,אחר מכן b ,ובסוף c .

$$S \rightarrow S'] \quad (1)$$

$$S' \rightarrow aS'bC \mid \varepsilon \quad (2)$$

$$Cb \rightarrow bC \quad (3)$$

$$C] \rightarrow]c \quad (4)$$

$$] \rightarrow \varepsilon \quad (5)$$

$$\begin{array}{llllll} S & \xrightarrow{1} & S'] & \xrightarrow{2} & aS'bC] & \xrightarrow{2} aaS'bCbC] & \xrightarrow{2} aaaS'bCbCbC] \\ & \xrightarrow{3} & aaaS'bbCCbc] & \xrightarrow{3} & aaaS'bbCbCC] & \xrightarrow{3} & aaaS'bbbCCC] \\ & \xrightarrow{4} & aaaS'bbbCC]c & \xrightarrow{4} & aaaS'bbbC]cc & \xrightarrow{4} & aaaS'bbb]ccc \\ & \xrightarrow{5} & aaaS'bbbccc & \xrightarrow{1} & aaabbccc & & \end{array}$$

דוגמה 5.8

בנו דקדוק כללי אשר יוצר את שפת המיללים

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

פתרון:

דוגמא זאת תמחיש כיצד דקדוק כללי יכול "לפעול בדומה" למוכנת טיריניג.

דקודוק נשתמש במשתנים וכלי גירה שיאפשרו מעין תנעה על גבי המחרוזות הנוצרת, בדומה לתנועת הראש של מוכנת טיריניג על גבי הסרט.

S → [H {	כל גזירה ייחיד מהמשתנה ההתחלתי. המשתנה H ידמה את הראש של המ"ט ש"יזוז" מצד לצד על המחרוזות הנגזרות. הסוגר המרובע] מסמן את הקצה השמאלי של המילה השמאלית. הסוגר המסלול } מסמן את הקצה השמאלי של המילה הימנית.	1
[H → [aH _a	כל זה מאפשר הוספת אות a לקצה השמאלי של המילה השמאלית. המשתנה H מוחלף במשתנה H _a כדי "לזכור" שיש עכשו להוסיף a גם במחוזות הימנית. (בדומה לזכרון של מ"ט).	2
H _a a → aH _a	כל זה מאפשר בראש "ליזוז" ימינה.	3
H _a { → H{a	כאשר המשתנה H _a "גינע" לסוגר המסלול, הוא יجوز אות a נוספת מימין לסוגר, שהוא הקצה השמאלי של המחרוזות הימנית. כך אפשרנו להוסיף שתי אותיות a: אחת מימין לסוגר] ואחת תואם ימין לסוגר } . כלומר אותן a בקצת השמאלי של כל אחת המחרוזות.	4
aH → Ha	cut ציריך לאפשר למשתנה H לחזור שמאלה על גבי האותיות שבין הסוגרים, עד הסוגר].	5

ברגע "שהראש" H חזר לתחילת המחרוזות ועומד ליד הסוגר] עברים על שלבים 5-2 שוב. בסבב הבא נחק במחשב גם יקרה של שתי אותיות b.

[H → [bH _b	כל זה מאפשר הוספת אות b לקצה השמאלי של המילה השמאלית. המשתנה H מוחלף במשתנה H _b כדי "לזכור" שיש עכשו להוסיף b גם במחוזות הימנית.	2'
H _a a → aH _a H _a b → bH _a H _b a → aH _b H _b b → bH _b	כללים האלהאפשרים בראש "ליזוז" ימינה.	3'
H _b { → H{b	כאשר המשתנה H _b "גינע" לסוגר המסלול, הוא יجوز אות b נוספת מימין לסוגר, שהוא הקצה השמאלי של המחרוזות הימנית. כך אפשרנו להוסיף שתי אותיות b: אחת מימין לסוגר] ואחת תואם ימין לסוגר } . כלומר אותן b בקצת השמאלי של כל אחת המחרוזות.	4'
bH → Ha	Cut ציריך לאפשר למשתנה H לחזור שמאלה על גבי האותיות שבין הסוגרים, עד הסוגר].	5'

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

H → ε [→ ε { → ε	הכללים האלה אפשרים להעלים את המשתנים } , [, H	6
-------------------------	--	---

למשל:

$$\begin{array}{ccccccccc}
 S & \xrightarrow{1} & [H \{ & \xrightarrow{2} & [aH_a \{ & \xrightarrow{4} & [aH \{ a & \xrightarrow{5} & [Ha \{ a \\
 & & \xrightarrow{2} & & [aH_a a \{ a & \xrightarrow{3} & [aaH_a \{ a & \xrightarrow{4} & [aaH \{ aa & \xrightarrow{5} & [Haa \{ aa \\
 & & \xrightarrow{2} & & [bH_b a a \{ aa & \xrightarrow{3} & [baaH_b \{ aa & \xrightarrow{4} & [baaH \{ baa & \xrightarrow{5} & [Hbaa \{ baa \\
 & \xrightarrow{6} & baabaa & & & & & &
 \end{array}$$

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- $L = L(G)$.

הוכחה: **ביוון ראשון.**

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

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

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

$w=S$ (1)

:repeat (2)

- פצל באופן לא דטרמיניסטי את w ל- xyz .
- בחר באופן לא דטרמיניסטי גזירה $v \rightarrow t$ של G .
- אם $t \neq y$ דחיה.
- $zv=x$
- אם $w==v$ קבל.

ביוון שני.

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

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

נתונה מ"ט M בעלת הtablת המעברים להלן. נבנה דקדוק כללי G שممמש אותם צעדים.

תואזה	כתביה	מצב חדש	סימן	מצב
R	a	q_0	q_0	a
R	b	q_1	q_1	b
L	-	acc	acc	-
L	a	q_0	q_0	a
L	b	q_1	q_1	b

לפי הtablת המעברים קיימים הצעדים

$q q_0 b a b \vdash_M aaq_1 ab$

נניח שבדוק כללי G קיים אותו הצעד

$$q q_0 bab \xrightarrow{G} aaq_1 ab$$

ניתן למעשה צעד זה על ידי הכלל

$$q_0 \xrightarrow{b} a \quad q_1$$

באופן כללי,

- עבור כל פונקציית המעברים של M שגוררת תזוזה ימינה מצורה

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

נமמש מעבר זה על ידי כלל של הדקדוק G מצורה

$$q\sigma \rightarrow \pi p .$$

- עבור כל פונקציית המעברים של M שגוררת תזוזה שמאליה מצורה

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

או כלל $\Gamma \in \tau$ ב- G נממש מעבר זה על ידי הכלל

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

■

5.6 היררכיה של חומסקי

מודל חישובי	דקדוק	משפחה שפות
מכונת טיורינג	כללי	קבילות
אוטומט מחסנית	חסירות הקשר	אוטומטים רולריים וمتקובלות על ידי
אוטומט סופי	רגולרי	רגולריות

- היררכיה של חומסקי קושرت לנו בין משפחות של שפות דקדוקים ומודלים חישוביים.
- בתחום ההיררכיה נמצאות השפות הרגולריות שנוצרות על ידי דקדוקים רולריים וمتקובלות על ידי אוטומטים סופיים.
- מעלהן נמצאות השפות חסירות הקשר שנוצרות על ידי דקדוקים חסרי הקשר וمتקובלות על ידי אוטומטי מחסנית.
- מעלהן נמצאות השפות הקבילות שנוצרות על ידי דקדוקים כלליים וمتקובלות על ידי מכונות טיורינג.
- כל רמה בהיררכיה מכילה ממש את הרמה שמתחתה.
- * כל שפה רגולרית היא גם חסירת הקשר, אבל יש שפות חסירות הקשורין רגולריות.
- * כל שפה חסירת הקשר היא קבילה, אבל יש שפות קבילות שאין חסירות הקשר.

5.7 כל שפה חסירת הקשר הינה קריאה

לפי היררכיה של חומסקי אנחנו יודעים לקבוע שכל שפה חסירת הקשר היא קבילה.

האם כל שפה חסירת הקשר הינה קריאה?

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

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

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

מתקיים ש- s היא תת-מחוזת של (u_i) . אומרים שקודקוד u_i הוא משמעותי אם (u_i) מכיל ממש את s .

כל קודקוד משמעותי מוסיף לפחות אחת ל- w .
 לכן, ישנו לכל היותר $|w|$ קודקודים משמעותיים.
 לכן, ברגע הקודקודים הפנימיים (u_1, u_2, \dots, u_m) שאורכן לפחות $(|V| + 1)(|w| + 1)$, בהכרח ישנו תת רצף $(u_i, u_{i+1}, \dots, u_{i+|V|+1})$, שבו כל הקודקודים לא משמעותיים.
ברצף זה בהכרח ישנים 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 .
 בסתיו להנחה המינימליות של העץ.

משפט 5.6

כל שפה חסרת הקשר היא כריעה.

הוכחה: בהינתן דקדוק חסר הקשר (V, Σ, S, R) , התוכנית הלא דטרמיניס הבאה מכריעת 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 מסוימים. לכן, זו תוכנית להכרעה. ישנו חישוב שמחזיר "כן" אם ורק אם $w \in L(G)$.

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

$$R = \{L \subseteq \Sigma^* : \text{קיים מ"ט המכ裏עה את } L\}.$$

הגדרה 6.2 RE

אוסף השפות הקבילות מסומן RE ומוגדר

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

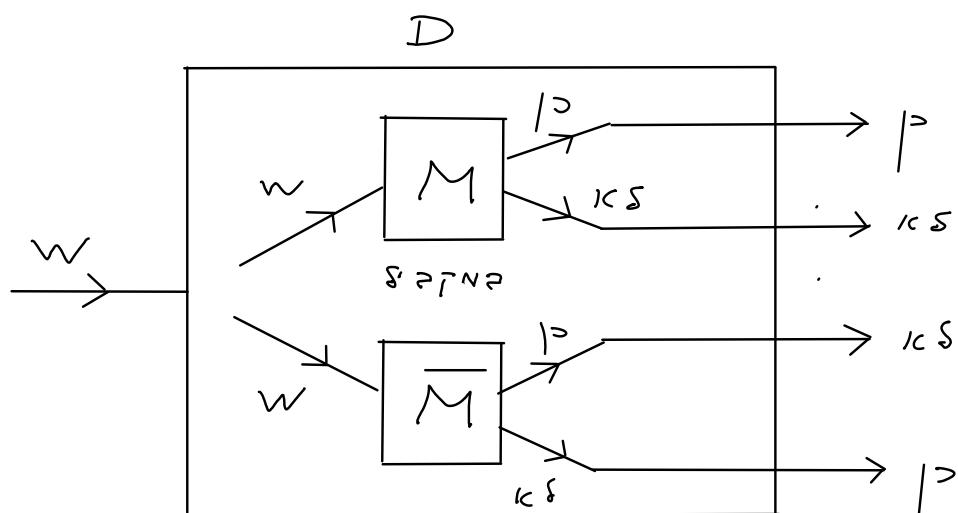
6.2 היחס בין הכרעה וקבלה

למה 6.1 היחס בין הכרעה וקבלה

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

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

בנה מ"ט D המכ裏עה את L .



על קלט w : D

1) D מעתקה את w לסרט נסף.

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

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

נוכיח כי D מכריעה את L .

אם $w \in L$

$w \in L(M) \Leftarrow$

מקבלת את w או (w דוחה את M) \Leftarrow

עוצרת ומתקבלת את w .

אם $w \notin L$

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

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

מקבלת את w או (w דוחה את M) \Leftarrow

עוצרת ודוחה את w .



6.3 סגירות של שפות כריעות ושפות קבילות

משפט 6.1 סגירות של השפות הקריעות

סגורה תחת: R

1) איחוד

2) חיתוך

3) משלימים

4) שרשור

5) סגור קלין

משפט 6.2 סגירות של השפות הקבילות

סגורה תחת: RE

1) איחוד

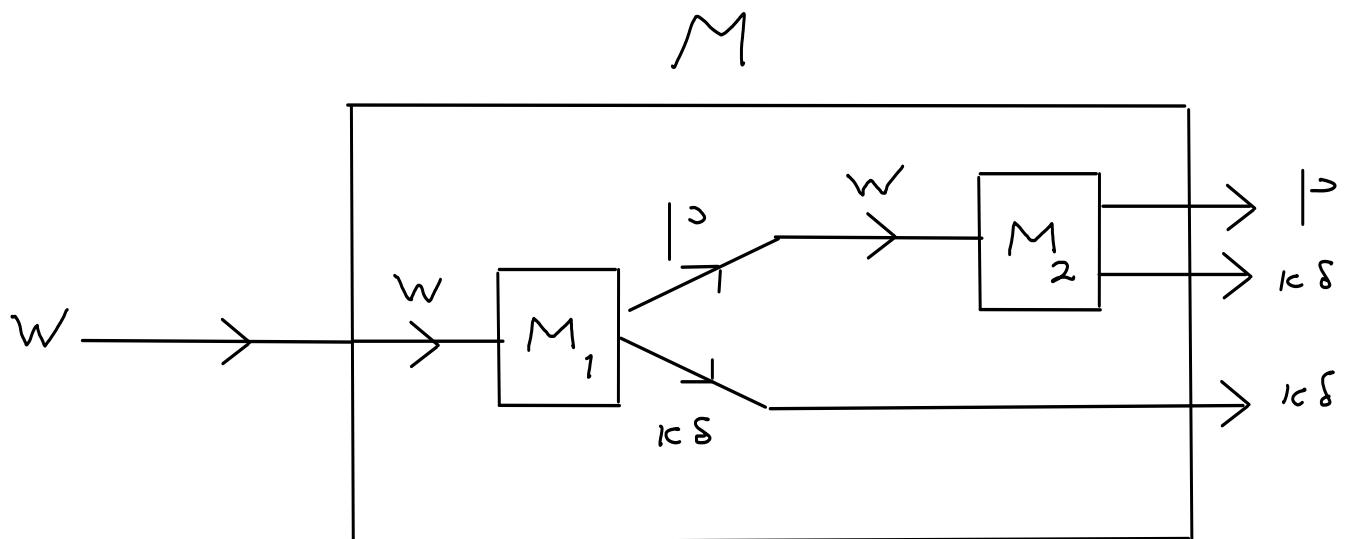
(2) חיתוך

(3) שרשור

(4) סגור קלין

הוכחה:

(1) חיתוך:

(א) סגורה תחת חיתוך R נוכיח כי לכל שתי שפות $L_1, L_2 \in R$ מתקיים $L_1 \cap L_2 \in R$.תהי $M_1 \cap M_2 \in R$ המככירה את $L_1 \cap L_2$ בהתאם. נבנה מ"ט המככירה את L_1 ו- L_2 .תאור הבנייהעל קלט $w = M$:1) מעתקה את w לסרט נוספת.2) מרכיב את M_1 על w .• אם M_1 דוחה $\Leftarrow M$ דוחה.• אחרת M מרכיב את M_2 על העותק של w ועונה כמוותה.נקודות:נוכיח כי M מככירה את $L_1 \cap L_2$.אם $w \in L_1 \cap L_2$ $w \in L_2$ וגם $w \in L_1 \Leftarrow$

w מקבלת את M_1 ו- M_2 מקבלת את $w \Leftrightarrow$

M מקבלת את w .

אם

$w \notin L_1 \cap L_2 \Leftrightarrow$

w דוחה את M_2 או w דוחה את M_1 .

w דוחה את M .

(ב) סגורה תחת חיתוך RE

nocih ci lcl shi shfot $L_1, L_2 \in RE$ matkym $L_1 \cap L_2 \in RE$

tahyina M_1 M₂ shi mconot tyorign makkblot at L_1 - L_2 bhetama.

nbnha m't M makkblat at $L_1 \cap L_2$ baotu open cmo (a).

(2) **איחוד:**

(א) סגורה תחת איחוד R

nocih ci ld l shi shfot $L_1, L_2 \in R$ matkym $L_1 \cup L_2 \in R$

tahyina M_1 m't mckriya at L_1 - M_2 m't mckriya at L_2 .

nbnha m't M mckriya at $L_1 \cup L_2$.

תאור הבנייה

על קלט $w = M$:

(1) M mutika at w lsrut nosf.

(2) mericha at M_1 ul w .

• am M_1 makkblat $\Leftarrow M$ makkblat.

• achrat, M mericha at M_2 ul houtek shel w wouna cmoha.

(ב) סגורה תחת איחוד RE

nocih ci lcl shi shfot $L_1, L_2 \in RE$ matkym $L_1 \cup L_2 \in RE$

tahyina M_1 m't makkblat at L_1 - M_2 m't makkblat at L_2 .

nbnha m't M az M makkblat at $L_1 \cup L_2$.

תאור הבנייה

על קלט $w = M$:

(1) M bochrot baopen az'd $i \in \{1, 2\}$.

(2) M mericha at M_i ul w wouna cmoha.

(3) **שרשור:**

(א) סגורה תחת שרשור R

nocih ci lcl shi shfot $L_1, L_2 \in R$ matkym $L_1 \cdot L_2 \in R$ caasher

$$L_1 \cdot L_2 = \{w = w_1 w_2 \mid w_1 \in L_1 \wedge w_2 \in L_2\} .$$

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 $L_1 \cdot L_2$.

תאור הבנייה

על קלט $w = M$

(1) בוחרת באופן א"ד חלוקה של w ל- $w = w_1 w_2$

(2) מרים את M_1 על w_1 M

- אם D דוחה $M_1 \Leftarrow D$

- אחרת, M מרים את M_2 על w_2 ועונה כמוות.

(ב) סגורה תחת שרשור

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

4) * קליני

א) סגורה תחת * קליני

nocich ci lal shfa L:

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

כasher

$$L^* = \{w = w_1 w_2 \cdots w_k \mid \forall 1 \leq i \leq k, w_i \in L\}.$$

תהי M מ"ט המכריעה את L .

נבנה מ"ט א"ד M^* המכריעה את L^* .

תאור הבנייה

על קלט $w = M^*$

(1) אם $w = \varepsilon$ מתקבלת.

(2) אחרת M^* בוחרת באופן א"ד חלוקה של w ל- $w = w_1 \cdots w_k$

(3) לכל $1 \leq i \leq k$

w_i מרים את M על M^*

- אם D דוחה את w_i $M^* \Leftarrow D$

- אחרת חוזרים לשלב (3).

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

(ב) RE סגורה תחת * קליני

5) משלים

א) RE סגורה תחת המשלים

nocich ci

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

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} .

$w = \bar{M}$ על קלט:

(1) מרכיב את M על w .

- אם M מקבלת \bar{M} דוחה.

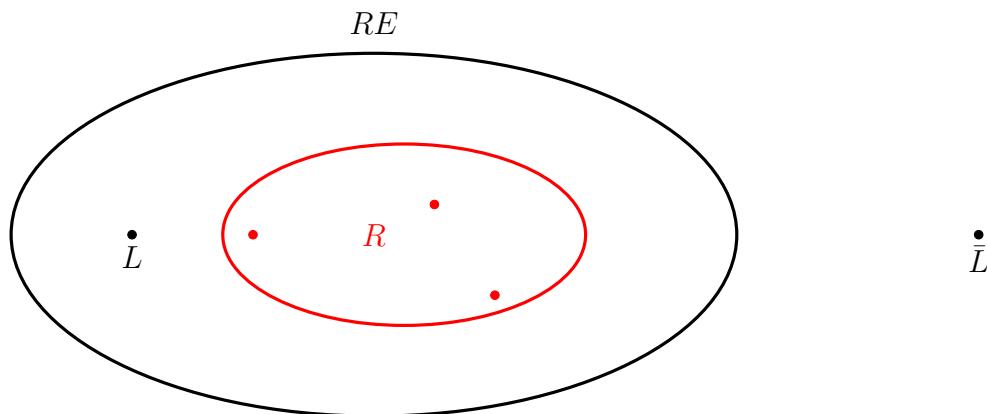
- אם דוחה M מקבלת.

ב) אינה סגורה תחת המשלים



משפט 6.3 אינה סגורה תחת המשלים

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



הוכחה:

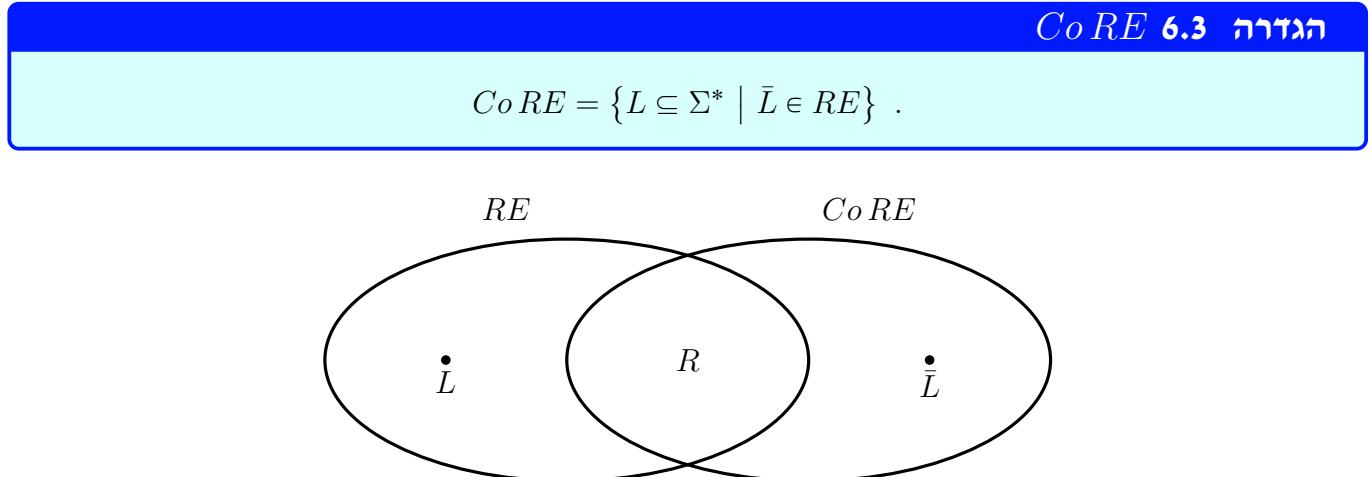
נניח כי $\bar{L} \in RE$ ונניח בשילילה כי $L \in RE \setminus R$.

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



הגדרה 6.3 Co RE

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



אבחנה

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 שפה כריעה ותהי $DROPOUT(L)$ השפה הבאה:

$$DROPOUT(L) = \{uv \mid uv \in L, \sigma \in \Sigma^* : u\sigma v \in L\}$$

אזי השפה $DROPOUT(L)$ גם כריעה.

הוכחה:

השפה L כריעה אזי קיימת מכונת טיריניג M_L המכריעה אותה. נבנה מכונת טיריניג $M_{DROPOUT}$ המכריעה את $DROPOUT(L)$ באופן הבא:

בנייה המכונות טיריניג

על כל קלט w :

1) בוחרת חילוק של המילה w לשורש של שני מילים באופן א-דטרמיניסטי כך ש: $w = uv$.

2) בוחרת אות σ^* באופן א-דטרמיניסטי ובונה את המילה $u\sigma v$.

3) מריצה M_L על המילה $u\sigma v = w$ ועונה כמוות.

הוכת הנכונות

נניח ש- $w \in DROPOUT(L)$.

\Leftarrow קיימים מילים $u, v \in L$ וקיים $\sigma^* \in \Sigma^*$ כך ש- $w = uv$.

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

\Leftarrow $M_{DROPOUT}$ קיבל את w .

נניח ש- $w \notin DROPOUT(L)$.

\Leftarrow לא קיימים מילים $u, v \in L$ ו- $\sigma^* \in \Sigma^*$ כך ש- $w = uv$.

\Leftarrow כל ריצה של $M_{DROPOUT}$ עוברת ל- q_{rej} .

\Leftarrow $M_{DROPOUT}$ תדחה את w .

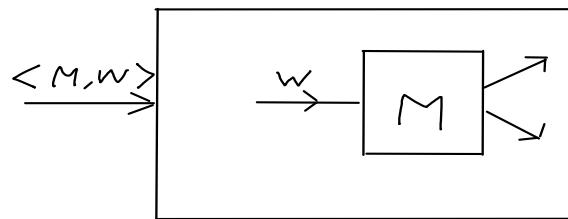


6.5 קידוד של מ"ט דטרמיניסטית

הגדרה 6.4 קידוד של מ"ט

בاهינתן קבוצה O של עצמים מופשטים (למשל מכונת טיורינג, תוכנית מחשב, גרפ'). הקידוד של O , מסומן $\langle O \rangle$, הוא מיפוי של O אל מהירות מעל אלףית סופי ששי 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 מילה $\langle w \rangle$ וקידוד של מ"ט $\langle M \rangle$, וביצעת סימולציה של ריצה של M על w ועונה בהתאם.

תאור הפעולה של U

על קלט x :

(1) בודקת אם 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 :

1	6	7	0	$\langle M \rangle$	-	$\langle w \rangle$...
---	---	---	---	---------------------	---	---------------------	-----

2	6	7	0	$\langle \rangle$	-	$\langle w \rangle$...
---	---	---	---	-------------------	---	---------------------	-----

- רושמת את הקוניגורציה ההתחלתית w_{q_0} על סרט 2.
- מחשבת את הקוניגורציה הבאה בעזרת טבלת המעברים.
- בסוף כל מעבר בין שתי קוניגורציות, U בודקת אם המצב הנוכחי הוא q_{acc} .
 - * אם כן U עוצרת ומקבלת.

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

מהי השפה של U ?

לכל x :

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

: $x = \langle M, w \rangle$ (2)

- אם M מקבלת w מקבלת את x .
- אם M דוחה את w דוחה את x .
- אם M לא עוצרת על w לא עוצרת על x .

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

הגדרה 6.5 L_{acc}

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

הגדרה 6.6 L_{halt}

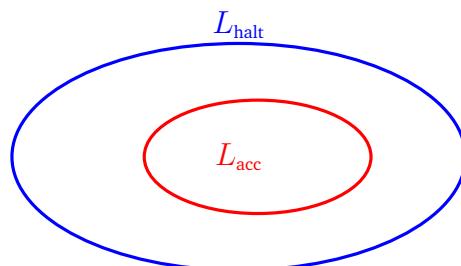
$$L_{\text{halt}} = \{\langle M, w \rangle \mid w \text{ עוצרת על } M\} \in \text{RE} \setminus \text{R}$$

הגדרה 6.7 L_{d}

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

אבחנה:

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



משפט 6.5

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

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

משפט 6.6

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

הוכחה: נבנה מ"ט U' שהוא למעשה U פרט למקום שבו U עצמה ומחטה, U' תעצור ותקבל.

נווכיח כי U' מקבלת את L_{halt} :

אם $x \in L_{\text{halt}}$

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

x עוצרת ומתקבלת את $U' \Leftarrow$

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

. x דוחה את $U' \Leftarrow x \neq \langle M, w \rangle$ •

. M לא עוצרת על $U' \Leftarrow w$ לא עוצרת על $x = \langle M, w \rangle$ •



שיעור 7

אי-כריעות

7.1 השפות לא כריעות $L_d, L_{\text{halt}}, L_{\text{acc}}$

הגדרה 7.1 L_{acc}

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

הגדרה 7.2 L_{halt}

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

הגדרה 7.3 L_d

$$L_d = \{\langle M \rangle \mid \langle M \rangle \notin L(M)\} \notin RE$$

משפט 7.1 $L_{\text{acc}} \in RE$

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

הוכחה: מכיוון ש- $L(U) = L_{\text{acc}}$, כאשר U המכונת טיריניג 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' שהוא למעשה U פרט למקום שבו U עצמה ומחטה, U' תעוצר ותחזור ותקבל.

נווכיח כי U' מקבלת את L_{halt} :

אם $x \in L_{\text{halt}}$

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

x עוצרת ומקבלת את $U' \Leftarrow$

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

• U' דוחה את x .

• M לא עוצרת על U' לא עוצרת על w .

משפט 7.3 $L_d \notin RE$

$$L_d \notin RE .$$

הוכחה:

נניח בשלילה כי $L_d \in RE$.

• אם M_d מקבלת את L_d ⇐

$$. L(M_d) = L_d \Leftarrow$$

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

• אם $L(M_d) \neq L_d$ ⇐ $\langle M_d \rangle \notin L_d$ ⇐ $\langle M_d \rangle \in L(M_d)$.

• אם $L(M_d) \neq L_d$ ⇐ $\langle M_d \rangle \in L_d$ ⇐ $\langle M_d \rangle \notin L(M_d)$.

בשני המקרים קיבלנו סתיירה לכך ש- $L_d \notin RE$ ולכן $L(M_d) = L_d$.

משפט 7.4 $L_{\text{acc}} \text{ לא כריעה}$

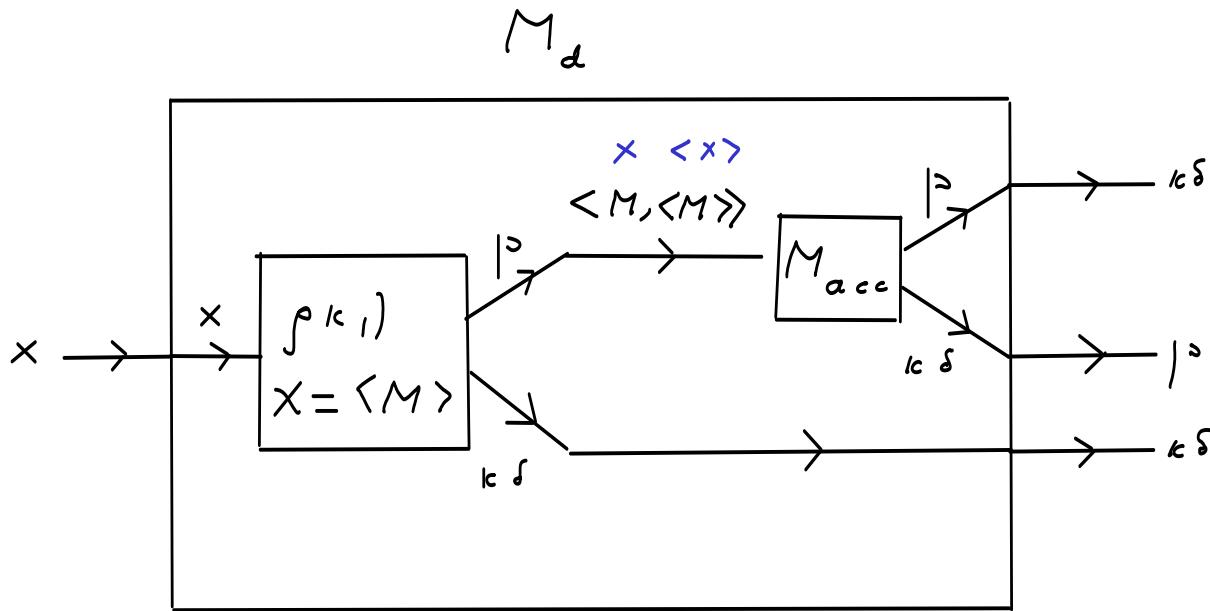
$$L_{\text{acc}} = \{\langle M, w \rangle \mid w \in L(M)\} \notin R .$$

הוכחה:

נניח בשלילה כי $L_{\text{acc}} \in R$ ותהי M_{acc} המכריעה את L_{acc} .

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

$$L_d = \{\langle M, w \rangle \mid \langle M \rangle \notin L(M)\} .$$



התאור של M_d

: x על קלט $= M_d$

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

2) מחשבת את $\langle x \rangle = \langle\langle M \rangle\rangle$

3) מרייצה את $\langle M, \langle M \rangle \rangle$ על הזוג M_{acc}

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

- אם M_{acc} דוחה \Leftarrow מקבלת.

כעת נוכיח כי M_d מכירעה את L_d :

אם $x \in L_d$

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

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

x מקבלת את $M_d \Leftarrow$

אם $x \notin L_d$ שני מקרים:

x דוחה את $M_d \Leftarrow x \neq \langle M \rangle$:1)

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

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

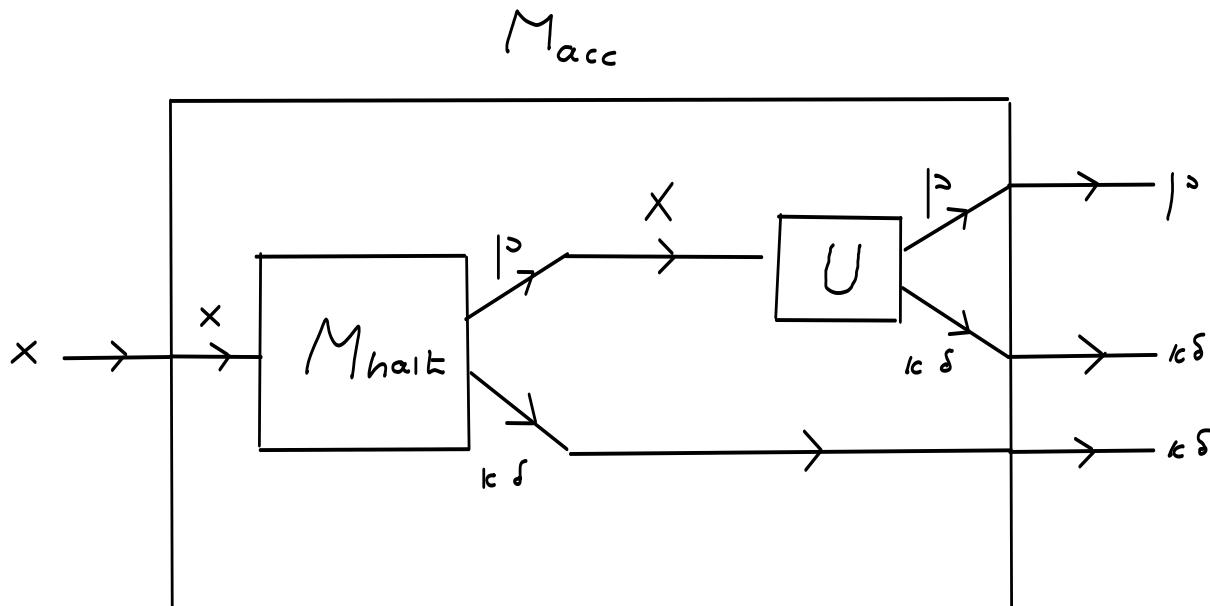
x דוחה את $M_d \Leftarrow$

משפט 7.5 לא כריעה L_{halt}

$$L_{\text{halt}} = \{\langle M, w \rangle \mid w \text{ עצרת על } M \} \notin R .$$

הוכחה:

נניח בsvilleה כי $L_{\text{halt}} \in R$ ותהי M_{halt} מ"ט המכריעה את $L_{\text{halt}} \in R$.
נשתמש בו כדי לבנות מ"ט המכריעה את L_{acc} (בסתירה לכך שגם $L_{\text{acc}} \notin R$ כפי שהוכחנו במשפט 7.4).

התאור של M_{acc} על קלט $x = M_{\text{acc}}$ 1) מרים את M_{halt} על x .• אם M_{acc} דוחה M_{halt} .• אם M_{acc} מקבלת M_{halt} מרים את U על x ועונה כמוה.אבחןנוכיח כי M_{acc} מכריעה את L_{acc} אם $x \in L_{\text{acc}}$

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

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

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

מקרה (1) $\langle M, w \rangle$:

דוחה את $M_{\text{halt}} \Leftarrow$
דוחה את $M_{\text{acc}} \Leftarrow$

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

מקרה (א): M לא עוצרת על w דוחה את x .

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

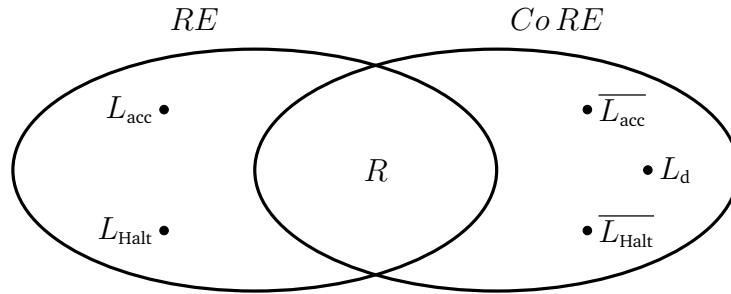
הראנו כי M_{acc} מכיריעת L_{acc} בסתייה לכך ש- $L_{\text{acc}} \notin R$.
 $L_{\text{halt}} \notin R$.

משפט 7.6

$$L_{\text{acc}} \in RE \setminus R \Rightarrow \bar{L}_{\text{acc}} \notin RE ,$$

$$L_{\text{halt}} \in RE \setminus R \Rightarrow \bar{L}_{\text{halt}} \notin RE ,$$

$$L_d \notin RE \setminus R .$$



7.2 השפה L_E לא כריעה

הגדרה 7.4 השפה L_E

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

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} המכיריעת L_{acc} באופן הבא.

בנייה של M_w

ראשית נגדיר את המ"ט M_w :

על כל קלט $x = M_w$:

(1) אם $x \neq w$ ⇐ דוחה.

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

אבחנה

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

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

בנייה של M_{acc}

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

על כל קלט $x = M_{\text{acc}}$:

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

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

(3) מריצה M_E על M_w :

(4) • אם M_E מקבלת ⇐ דוחה.

• אם M_E דוחה ⇐ מקבלת.

נכונות

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

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

מקרה 1: M_{acc} מקבלת M_E ⇐ $L(M_w) = \emptyset$ ⇐ $x \neq \langle M, w \rangle$

מקרה 2: M_{acc} ⇐ $\langle M_w \rangle$ M_E ⇐ $L(M_w) = \emptyset$ ⇐ $w \notin L(M)$ ו- $x = \langle M, w \rangle$

לסיכום:

אם L_E כרעה אז אפשר לבנות מ"ט M_{acc} המכrica את L_{acc} בסתירה לכך ש- $L_E \notin R$.

משפט 7.8 $L_E \notin RE$

$L_E \notin RE$

הוכחה:
הרעיון

בנייה מכונת טיריניג אי-דטרמיניסטיבית N מקבלת את

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

: x על קלט $= N$

1 אם $x \neq \langle M \rangle$ דוחה.

2 אם $x \in N$ בוחרת מילה Σ^* w באופן אי-דטרמיניסטיבי.

3 מרים M על w .

- אם M מקבלת $\Rightarrow N$ מקבלת.

- אם M דוחה $\Rightarrow N$ דוחה.

הוכחת הנכונות

אם $x \in \bar{L}_E$

$L(M) \neq \emptyset \wedge x = \langle M \rangle \Leftarrow$

$w \in L(M) \wedge \Sigma^* w \text{ כך ש } \Leftarrow$

$\exists \text{ ניחוש } \Sigma^* w \text{ כך ש } M \text{ מקבלת את } w \Leftarrow$

$x = \langle M \rangle \Leftarrow \text{קיים חישוב של } N \text{ מקבל את}$

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

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

כעת נוכיח כי $\bar{L}_E \notin RE$

$.L_E \in R$, 6.1 הוכחנו לעלה ש- $\bar{L}_E \in RE$. לכן $\bar{L}_E \in RE$ הוכחה כי $L_E \in RE$.
או בסתרה לכך ש- $L_E \notin R$.
 $L_E \notin RE$ לכן ■

7.3 השפה L_{EQ} לא כריעה

הגדרה 7.5

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

משפט 7.9 $L_{EQ} \notin R$

$L_{EQ} \notin R$

השפה L_{EQ} לא כריעה.

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

בנייה של M_E

על כל קלט $x = M_E$

• אם $x \neq \langle M \rangle$ דוחה. **(1)**

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

• אם M_{EQ} מקבלת M_E מתקבלת. **(3)**

• אם M_{EQ} דוחה דוחה. **(4)**

נכונות

אם $x \in L_E$

$L(M) = \emptyset \wedge x = \langle M \rangle \Leftarrow$

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

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

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

M_E מקבל. **(5)**

אם $x \notin L_E$ שני מקרים:

• אם $x \neq \langle M \rangle$ דוחה. $M_E \Leftarrow x \neq \langle M \rangle$ **(1)**

• אם $x = \langle M \rangle$ דוחה. **(2)**

$L(M) \neq \emptyset \wedge x = \langle M \rangle \Leftarrow$

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

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

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

M_E דוחה. **(6)**

לסיכום:

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

משפט 7.10 $L_{EQ} \notin RE$

$L_{EQ} \notin RE$

לא קבילה. L_{EQ}

הוכחה:

נניח בשלילה כי L_{EQ} קבילה. תהי M_{EQ} מ"ט המכבלת את L_E באופן הבא.

בנייה של M_E

x על כל קלט $= M_E$

1 אם $x \neq \langle M \rangle$ דוחה.

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

• אם M_{EQ} מקבלת \Leftarrow מקבלת.

נכונות

אם $x \in L_E$

$L(M) = \emptyset \wedge x = \langle M \rangle \Leftarrow$

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

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

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

. M_E מקבל.

לסיכום:

אם $L_E \notin RE$ קבילה אז אפשר לבנות מ"ט M_E מקבלת את L_E בסתייה למשפט 7.8 האומר ש-
■ $L_{EQ} \notin RE$ לכן

משפט 7.11 $\bar{L}_{EQ} \notin RE$

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

הוכחה:

נניח בsvilleה כי \bar{L} קבילה. תהי $M_{\bar{acc}}$ מ"ט מקבלת את \bar{L}_{EQ} . אז נבנה מ"ט M_{EQ} מקבלת את \bar{L} .
באופן הבא.

בנייה של M_1

ראשית נגדיר מ"ט M_1 באופן הבא:

x על כל קלט $= M_1$

1) מריצה M על w ועונה כמוות.

בנייה של $M_{\bar{acc}}$

x על כל קלט $= M_{\bar{acc}}$

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

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

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

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

נכונות

אם $x \in L_{\overline{\text{acc}}}$

לא מקבלת $M \Leftarrow$

$L(M_1) = \emptyset \Leftarrow$

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

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

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

סיכום:

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

7.4 סיכום: כרייעות וקבילות של שפות

קבילה	כרייעה	
✓	✗	L_{acc}
✗	✗	$\overline{L}_{\text{acc}}$
✗	✗	L_d
✓	✗	L_{Halt}
✗	✗	$\overline{L}_{\text{Halt}}$
✗	✗	L_E
✓	✗	\overline{L}_E
✗	✗	L_{EQ}
✗	✗	\overline{L}_{EQ}
✗	✗	L_{REG}
✗	✗	L_{NOTREG}

שיעור 8

רדוֹקצִיה

8.1 טבלה של רדוֹקצִיות

טבלה של רדוֹקצִיות

עמוד	רדוקציה
דוגמה 8.6 עמוד 84	$L_{\text{HALT}} \leq L_{\text{acc}}$
דוגמה 8.11 עמוד 88	$\bar{L}_{\text{acc}} \leq L_{\text{NOTREG}}$
דוגמה 8.12 עמוד 89	$L_{\text{acc}} \leq L_{\text{NOTREG}}$
דוגמה 8.13 עמוד 90	$L_{\text{HALT}} \leq L_{\text{NOTREG}}$
דוגמה 8.15 עמוד 92	$L_{\text{acc}} \leq L_{\text{REG}}$
דוגמה 8.14 עמוד 91	$\bar{L}_{\text{acc}} \leq L_{\text{REG}}$
דוגמה 8.16 עמוד 93	$\bar{L}_{\text{acc}} \leq L_{M_1 \cap !M_2}$ 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 8.17 עמוד 93	$\bar{L}_{\text{acc}} \leq L_{M_1 \subset M_2}$ כאשר $.L_{M_1 \subset M_2} = \{\langle M_1, M_2 \rangle \mid L(M_1) \subset L(M_2)\}$

8.2 מכונת טיורינג המחשבת פונקציה

הגדרה 8.1 מכונת טיורינג המחשבת פונקציה

בහינתן פונקציה $\Sigma^* \rightarrow \Sigma^*$: f 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

בහינתן פונקציה $\Sigma^* \rightarrow \Sigma^*$: f אומרים כי f חסיבה אם קיימת מכונת טיורינג המחשבת את f .

דוגמה 8.1

$$f_1(x) = xx . \quad (8.1)$$

$f_1(x)$ חסיבה.

דוגמה 8.2

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

$f_2(x)$ חסיבה.

דוגמה 8.3

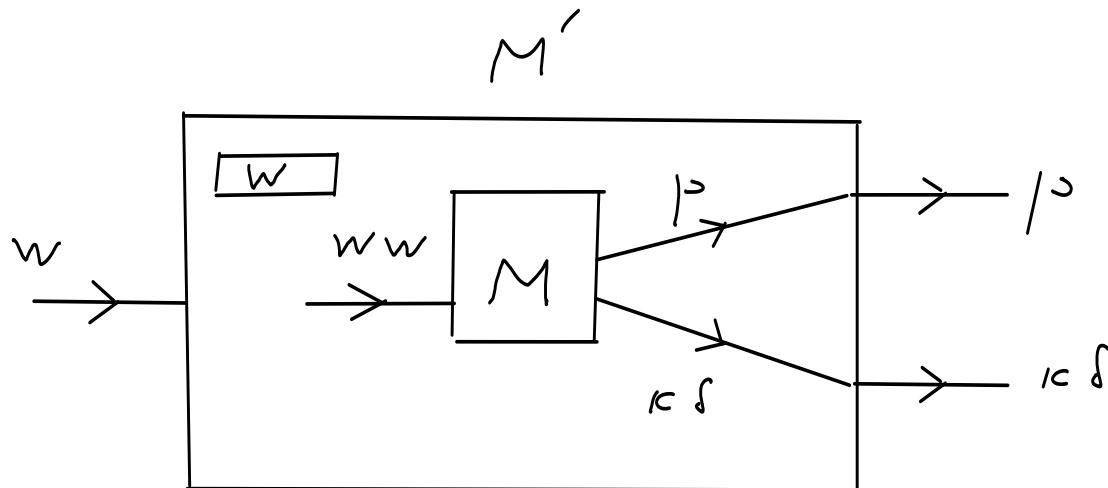
$$f_3(x) = \begin{cases} \langle M' \rangle & x = \langle M \rangle \\ \langle M^* \rangle & x \neq \langle M \rangle \end{cases} . \quad (8.3)$$

כאשר

M^* מ"ט שמקבלת כל קלט. •

M' מ"ט המכבלת את השפה

$$L(M') = \{w \in \Sigma^* \mid ww \in L(M)\} .$$



חסיבה כי ניתן לבנות מ"ט שבודקת האם $\langle M \rangle = x$. 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

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

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

8.3 רדוקציות**הגדרה 8.3 רדוקציות**

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

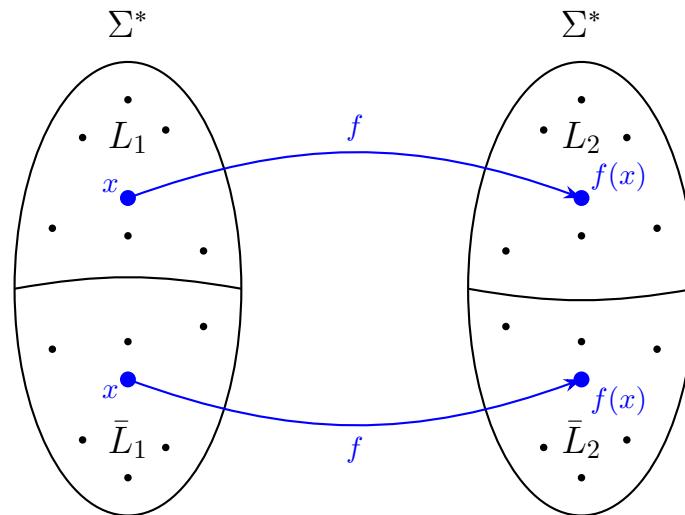
$$L_1 \leq L_2 ,$$

אם \exists פונקציה $f : \Sigma^* \rightarrow \Sigma^*$ המקיים:

(1) f חשיבה

(2) לכל $x \in \Sigma^*$

$$x \in L_1 \iff f(x) \in L_2 .$$

**דוגמה 8.5**

נתונות השפות

$$L_1 = \{x \in \{0,1\}^* \mid \text{זוגי } |x|\} ,$$

$$L_2 = \{x \in \{0,1\}^* \mid \text{אי-זוגי } |x|\} .$$

הוכיחו כי

$$L_1 \leq L_2 .$$

פתרונות:

נדיר את הפונקציה

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

הוכחת הנכונות:

$$\cdot f(x) \in L_2 \Leftarrow |f(x)| \Leftarrow f(x) = 1 \Leftarrow |x| \Leftarrow x \in L_1$$

$$\cdot f(x) \notin L_2 \Leftarrow |f(x)| \Leftarrow f(x) = 10 \Leftarrow |x| \Leftarrow x \notin L_1$$

משפט 8.1 משפט הרדוקציה

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

$$L_1 \leq L_2$$

אזי התנאים הבאים מתקיימים:

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

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

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

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

הוכחה: מכיוון ש-

$$L_1 \leq L_2$$

קיימת פונקציה f חסיבה המקיים:

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

לכל $x \in \Sigma^*$ תהי M_f מ"ט המחשבת את f .

$$(1) \quad \underline{\text{נוכיח}} \quad L_1 \in R \Leftarrow L_2 \in R$$

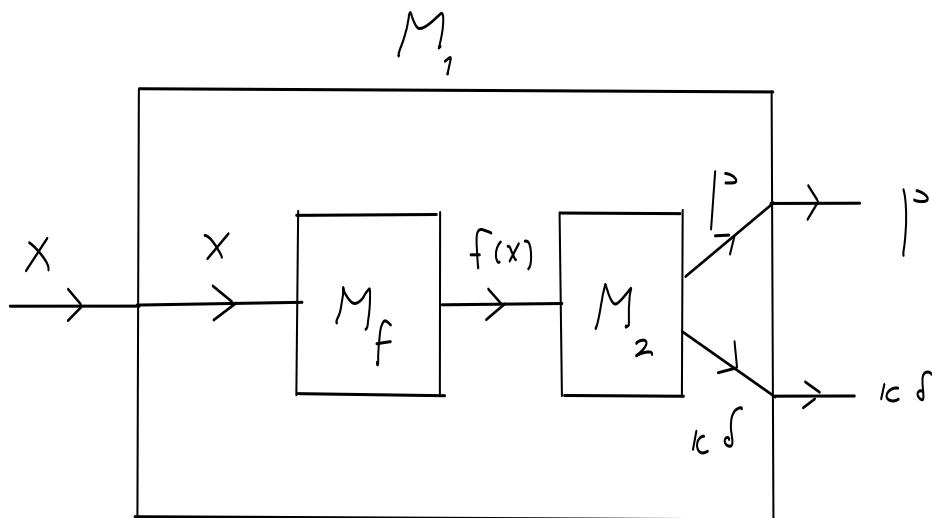
תהי M_2 מ"ט המכריעה את L_2 .נבנה מ"ט M_1 המכריעה את L_1 .התאור של M_1 $= \text{על קלט } x = M_1$ 1. מחשבת את $f(x)$ בעזרת M_f .2. מריצה את M_2 על $f(x)$ ועונה כמורה.נוכיח כי M_1 מכריעה את L_1 .

$$x \in L_1 \Leftarrow f(x) \in L_2 \Leftarrow M_2 \text{ מקבלת את } M_1 \Leftarrow f(x) \in L_2 \Leftarrow x \in L_1 \text{ אם }$$

$$x \in M_1 \iff f(x) \in M_2 \text{ דוחה את } M_2 \iff f(x) \notin L_2 \iff x \notin L_1 \bullet$$

$$\underline{L_1 \in RE \iff L_2 \in RE} \quad (2)$$

תהי M_2 מ"ט המקבלת את L_2
نبנה מ"ט M_1 המקבלת את L_1 .



התאור של M_1

$$x = \text{על קלט} : M_1$$

1. מחשבת את $f(x)$ בעזרת M_f .

2. מרים את M_2 על $f(x)$ ועונה כמוה.

נוכיח כי M_1 מקבלת את L_1 :

$$x \in M_1 \iff f(x) \in M_2 \text{ מקבלת את } x \iff f(x) \in L_2 \iff x \in L_1 \bullet$$

$$x \in M_1 \iff f(x) \notin M_2 \text{ לא מקבלת את } x \iff f(x) \notin L_2 \iff x \notin L_1 \bullet$$

(3)

(4)

כל 8.1

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

$$L \leq L' .$$

לדוגמה:

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

(כנ"ל לגבי R)

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

$$L' \leq L .$$

לדוגמה

$$L_d \leq L$$

(כנ"ל לגבי R).

8.6 דוגמה

$L_{\text{halt}} = \{\langle M, w \rangle \mid w \text{ עצמן על } M\}$ ו $L_{\text{acc}} = \{\langle M, w \rangle \mid w \in L(M)\}$ הוכיחו כי $L_{\text{acc}} \leq L_{\text{halt}}$ ע"י רדוקציה $L_{\text{acc}} \notin R$

פתרונות:

בנייה פונקציית f חשיבה ומקיימת

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

w מתקבל על M' מתקבל את w \Leftarrow M

w דוחה על M' \Leftarrow w דוחה את M

w לא מתקבלת על M' \Leftarrow w לא מתקבלת על M .

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

כאשר

M' מ"ט שלא עצמת על אף קלט.

M' מ"ט המתנהגת כמו M פרט למקומות בהם M עצמה ומחתה, M' תיכנס ללולאה אינסופית.

nocionot_haradoktsia

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

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

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

נכיח כי $x \in L_{\text{acc}} \iff f(x) \in L_{\text{halt}}$

: $x \in L_{\text{acc}}$ אם

$$w \in L(M) \text{ ו } x = \langle M, w \rangle \Leftarrow$$

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

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

אם x איז שני מקרים:

מקרה 1:

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

מקרה 2:

$$f(x) = \langle M', w \rangle \Leftarrow w \notin L(M) \text{ ו } x = \langle M, w \rangle$$

מקרה א: M לא עוצרת על w $w \notin L(M)$

מקרה ב: M דוחה את w $w \notin L(M)$

לסיום, הוכחנו רזוקציה $L_{\text{acc}} \not\subseteq R$. ומכיון ש- (משפט 7.4) איז ממשט הרזוקציה 8.1, מתקיים $L_{\text{halt}} \not\subseteq R$.

8.7 דוגמה

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:

$$L_{\Sigma^*} \not\subseteq R$$

פתרון:

nocich ci $L_{\Sigma^*} \not\subseteq R$ ע"י רזוקציה

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

בנה פונקציה חשיבה f המקיים

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

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

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

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

כאשר

מ"ט שדוחה כל קלט. •

• M' היא מ"ט שלל כל קלט x , מתעלמת מ- x ומריצה את M על w ועונה כמוות.

אבחנה:

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

נכונות הרדוקציה:

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

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

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

נוכיח כי

$$x \in L_{\text{acc}} \iff f(x) \in L_{\Sigma^*}$$

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

אם $x \in L_{\text{acc}}$ שני מקרים:

מקרה 1: $f(x) \notin L_{\Sigma^*} \Leftarrow L(M_\emptyset) = \emptyset \wedge f(x) = \langle M_\emptyset \rangle \Leftarrow x \neq \langle M, w \rangle$

מקרה 2: $f(x) \notin L_{\Sigma^*} \Leftarrow L(M') = \emptyset \wedge f(x) = \langle M' \rangle \Leftarrow w \notin L(M) \wedge x = \langle M, w \rangle$

לסיום, הוכחנו רדוקציה $L_{\text{acc}} \leq L_{\Sigma^*} \leq R$. ומכיון ש- $L_{\text{acc}} \leq L_{\Sigma^*}$ (משפט 7.4) אז ממשט הרדוקציה 8.1 מתקיים $L_{\Sigma^*} \notin R$.

8.8 דוגמה

נתונה השפה

$$L_d = \{\langle M \rangle \mid \langle M \rangle \notin L(M)\}$$

ונתונה השפה

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

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

$$L_d \leq \bar{L}_{\text{acc}} .$$

פתרון:

نبנה פונקציה חשיבה f המקיים

$$x \in L_d \iff f(x) \in L_{\text{acc}} .$$

$$w' \notin L(M') \iff \langle M \rangle \notin L(M)$$

$$w' \in L(M') \iff \langle M \rangle \in L(M)$$

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

כאשר M^* המ"ט שמקבלת כל קלט.

נכונות הרדוקציה:

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

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

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

נוכיח כי

$$x \in L_d \Leftrightarrow f(x) \in \bar{L}_{\text{acc}}$$

$$\Leftrightarrow \langle M \rangle \notin L(M) \text{ ו } f(x) = \langle M, \langle M \rangle \rangle \Leftrightarrow \langle M \rangle \notin L(M) \text{ ו } x = \langle M \rangle \Leftrightarrow x \in L_d \text{ אם } f(x) \in \bar{L}_{\text{acc}}$$

אם שני מקרים:

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

$$. f(x) \notin \bar{L}_{\text{acc}} \Leftrightarrow \langle M \rangle \in L(M) \text{ ו } f(x) = \langle M, \langle M \rangle \rangle \Leftrightarrow \langle M \rangle \in L(M) \text{ ו } x = \langle M \rangle \quad : \underline{\text{מקרה 2}}$$

לסיכום, הוכחנו רדוקציה $L_d \leq \bar{L}_{\text{acc}}$, ומכיון ש- $L_d \leq RE$ (משפט 7.3) אז ממשט הרדוקציה 8.1, מתקיים $\bar{L}_{\text{acc}} \notin RE$.

משפט 8.2 ממשפט הרדוקציה בין שפות משלימות

אם קיימת רדוקציה $\bar{L}_1 \leq \bar{L}_2$, אז קיימת רדוקציה $L_1 \leq L_2$

הוכחה:

אם \exists רדוקציה

$$L_1 \leq L_2$$

אז \exists פונקציה חשיבה f המקיים $x \in L_1 \Leftrightarrow f(x) \in L_2$

ולכן עבור אותה פונקציה f היא גם חשיבה וגם מקיימת

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

ולכן

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



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

דוגמה 8.9

הוכחנו בדוגמה 8.7 רדוקציה

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

לכן לפי משפט 8.2 קיימת רדוקציה

$$\bar{L}_{\text{acc}} \leq \bar{L}_{\Sigma^*} .$$

מכיוון ש- $\bar{L}_{\Sigma^*} \notin RE$, אז ממשפט הרדוקציה 8.1 מתקיים

מכיוון ש- $\bar{L}_{\text{acc}} \notin RE$, אז ממשפט הרדוקציה 8.1 מתקיים

דוגמה 8.10

הוכחנו בדוגמה 8.8 רדוקציה

$$L_d \leq \bar{L}_{\text{acc}} .$$

לכן לפי משפט 8.2 קיימת רדוקציה

$$\bar{L}_d \leq L_{\text{acc}} .$$

מכיוון ש- $\bar{L}_d \in RE$, אז ממשפט הרדוקציה 8.1 מתקיים

8.5 דוגמאות בשימוש של משפט הרדוקציה (משפט 8.1)

דוגמה 8.11

תהי L_{NOTREG} השפה

$$L_{\text{NOTREG}} = \{\langle M \rangle \mid L(M) \text{ לא רגולרית}\} .$$

הוכחו כי השפה L_{NOTREG} לא כרעה על ידי רדוקציה מ-

פתרונות:

השפה \bar{L}_{acc} מוגדרת

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

והשפה L_{NOTREG} מוגדרת

$$L_{\text{NOTREG}} = \{\langle M \rangle \mid L(M) \text{ לא רגולרית}\} .$$

נגידר 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:

$M' = \text{על כל קלט } y$

(1) אם $y \in PAL \iff$ מקבלת.

(2) אחרת מריצה M על w ועונה כמוות.

הוכחת נכונות הרדוקציה

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

מקרה 1: $x = \langle M, w \rangle$

w לא מקבלת M ⇐

$L(M') \in PAL$ ⇐

$\langle M' \rangle \in PAL$ ⇐

$f(x) \in PAL$ ⇐

$.f(x) \in L_{\text{NOTREG}}$ ⇐

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

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

לכן הוכחנו כי $f(x)$ היא רדוקציה מ- L_{acc} ל- L_{NOTREG} ⇔ $f(x) \in NOTERG$.

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

דוגמה 8.12

תהי L_{NOTREG} השפה

$$L_{\text{NOTREG}} = \{\langle M \rangle \mid \text{לא רגולרית } L(M)\}.$$

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

פתרון:

השפה L_{acc} מוגדרת $L_{\text{acc}} = \{\langle M, w \rangle \mid w \text{ מקבלת } M\}$.

והשפה L_{NOTREG} מוגדרת $L_{\text{NOTREG}} = \{\langle M \rangle \mid \text{לא רגולרית } L(M)\}$.

נגדיר הפונקציה הבאה:

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

כאשר M' מ"ט הבאה:

$M' = M'$ על כל קלט w :

(1) M' מריצה M על w .

(2) אם M דוחה ⇐ M' דוחה.

- אם M מקבלת $\Rightarrow M'$ בודקת אם y פלינדרום.

* אם כן \Rightarrow מקבלת.

* אם לא \Rightarrow דוחה.

הוכחת נכונות הרדוקציה

$$\begin{aligned} f(x) \in L_{\text{NOTREG}} &\Leftrightarrow f(x) \in PAL \Leftrightarrow L(M') = PAL \Leftrightarrow M \text{ מקבלת } w \Leftrightarrow x \in L_{\text{acc}} \\ &\quad .f(x) \notin L_{\text{NOTREG}} \Leftrightarrow \text{שניהם מקרים.} \end{aligned}$$

$$\begin{aligned} \langle M_\emptyset \rangle \notin L_{\text{NOTREG}} &\Leftrightarrow L(M_\emptyset) = \emptyset \text{ ו } f(x) = \langle M_\emptyset \rangle \Leftrightarrow x \neq \langle M, w \rangle \\ &\quad .f(x) \notin L_{\text{NOTREG}} \Leftrightarrow \end{aligned}$$

$$\begin{aligned} \langle M' \rangle \notin L_{\text{NOTREG}} &\Leftrightarrow L(M') = \emptyset \Leftrightarrow M \text{ לא מקבלת } w \text{ ו } x = \langle M, w \rangle \\ &\quad .f(x) \notin L_{\text{NOTREG}} \Leftrightarrow \end{aligned}$$

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

תהי השפה L_{NOTREG} הטענה
 $L_{\text{NOTREG}} = \{\langle M \rangle \mid L(M) \text{ לא רגולרית}\}$.

הוכחו כי השפה L_{NOTREG} לא כרעה על ידי רדוקציה מ- L_{HALT} .

פתרון:

השפה L_{HALT} מוגדרת
 $L_{\text{HALT}} = \{\langle M, w \rangle \mid w \text{ עצרת על } M\}$.

והשפה L_{NOTREG} מוגדרת
 $L_{\text{NOTREG}} = \{\langle M \rangle \mid L(M) \text{ לא רגולרית}\}$.

נדיר הפונקציה הבאה:

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

כאשר M' מ"ט הבאה:

: y על כל קלט $z = M'$

(1) M' מרכיבת M על w .

• אם M דוחה \Rightarrow דוחה. (2)

• אם M מקבלת \Rightarrow ממשיכה לשלב (3).

(3) $y \in PAL$ אם M' בודקת אם y פלינדרום.

• אם כן \Rightarrow מקבלת.

• אם לא \Rightarrow דוחה.

הוכחת נכונות

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

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

מקרה 1: $\langle M_\emptyset \rangle \notin L_{\text{NOTREG}} \iff L(M_\emptyset) = \emptyset \wedge f(x) = \langle M_\emptyset \rangle \iff x \neq \langle M, w \rangle \wedge f(x) \notin L_{\text{NOTREG}} \iff$

מקרה 2: $\langle M_\emptyset \rangle \notin L_{\text{NOTREG}} \iff L(M_\emptyset) = \emptyset \iff w \text{ לא עוצרת על } M \wedge x = \langle M, w \rangle \wedge f(x) \notin L_{\text{NOTREG}} \iff$

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

תהי L_{REG} השפה

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

הוכחו כי השפה L_{REG} לא כריעה על ידי רדוקציה מ-

פתרונות:

השפה \bar{L}_{acc} מוגדרת

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

והשפה L_{REG} מוגדרת

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

נגדיר הפונקציה הבאה:

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

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

$M' = \text{על כל קלט } y$:

(1) מריצה M על w .

(2) • אם M דוחה \iff דוחה.

• אם M מקבלת \iff בודקת אם y פלינדרום:

• אם כן \iff מקבלת.

• אם לא \iff דוחה.

אבחנה

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

הוכחת נכונות הרדוקציה

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

מקרה 1: $f(x) \in L_{\text{REG}} \Leftarrow \langle M_{\emptyset} \rangle \in L_{\text{REG}} \Leftarrow L(M_{\emptyset}) = \emptyset \wedge f(x) = \langle M_{\emptyset} \rangle \Leftarrow x \neq \langle M, w \rangle$

מקרה 2: $\langle M_{\emptyset} \rangle \in L_{\text{REG}} \Leftarrow L(M') = \emptyset \wedge f(x) = \langle M' \rangle \Leftarrow x \notin L(M) \wedge x = \langle M, w \rangle \wedge f(x) \in L_{\text{REG}} \Leftarrow$

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

דוגמה 8.15

תהי L_{REG} השפה

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

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

פתרונות:

השפה L_{acc} מוגדרת $L_{\text{acc}} = \{\langle M, w \rangle \mid w \text{ מקבלת } M\}$.

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

נדיר 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 שמכריע את השפה של פלינדרומים, ו- M' מ"ט הבאה:

$M' = \text{על כל קלט } y$:

(1) M' בודקת אם y פלינדרום:

- אם כן \Leftarrow מקבלת.
- אם לא מרים M על w ועונה כמוות.

הוכחת נכונות הרדוקציה

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

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

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

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

דוגמה 8.16

נתונה השפה הבאה:

$$L = \{\langle M_1, M_2, w \rangle \mid w \in L(M_1) \wedge w \notin L(M_2)\} .$$

הוכחנו כי $L \notin RE$ ע"י רדוקציה מ-**פתרון:**פונקציית הרדוקציה:

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

כאשר

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

נכונות הרדוקציה:

ראשית, f חסיבה כי ניתן לבנות מ"ט שתבדוק האם $x = \langle M, w \rangle$ או לא, תחזר קידוד קבוע $\langle M^*, M_\emptyset, \varepsilon \rangle$. אם לא, תחזר קידוד $\langle M^*, M, w \rangle$.

נוכיח כי

$$x \in \bar{L}_{\text{acc}} \iff f(x) \in L_{M_1 \rightarrow M_2} .$$

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

מקרה 1: $f(x) \in \dots \iff \varepsilon \notin L(M_\emptyset) \wedge \varepsilon \in L(M^*) \wedge f(x) = \langle M^*, M_\emptyset, \varepsilon \rangle \iff x \neq \langle M, w \rangle \iff x \in \bar{L}_{M_1 \rightarrow M_2}$.

מקרה 2: $w \notin L(M) \wedge w \in L(M^*) \wedge f(x) = \langle M^*, M, w \rangle \iff w \notin L(M) \wedge x = \langle M, w \rangle \iff f(x) \in L_{M_1 \rightarrow M_2} \iff$

$w \notin L(M) \wedge w \in L(M^*) \wedge f(x) = \langle M^*, M, w \rangle \iff w \in L(M) \wedge x = \langle M, w \rangle \iff x \notin \bar{L}_{\text{acc}}$ אם $f(x) \notin L_{M_1 \rightarrow M_2} \iff$

לסיכום, הוכחנו רדוקציה $\bar{L}_{\text{acc}} \notin RE$, ומכיון ש- $\bar{L}_{\text{acc}} \leq L_{M_1 \rightarrow M_2}$ ממשפט הרדוקציה מתקיים

דוגמה 8.17

נתונה השפה הבאה:

$$L = \{\langle M_1, M_2, w \rangle \mid w \in L(M_1) \wedge w \notin L(M_2)\} .$$

הוכחנו כי $L \notin RE$ ע"י רדוקציה מ-**פתרון:**פונקציית הרדוקציה:

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

כאשר

- M_\emptyset היא מ"ט שדועה כל קלט.
- M' היא מ"ט של קלט y מתעלמת מ- y ומריצה M על w ועונה כמוות.

אבחנה:

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

נכונות הרדוקציה:

ראשית, f חשיבה כי ניתן לבנות מ"ט שתבדוק האם $x = \langle M, w \rangle$. אם לא, תחזיר קידוד קבוע $\langle M_\emptyset, M_\emptyset \rangle$. אם כן, תחזיר קידוד קבוע $\langle M'_\emptyset, M'_\emptyset \rangle$, כאשר M'_\emptyset הוסיף kod ל- $\langle M \rangle$ המוחק את הקלט y ורושם w במקומו.

נוכיח כי

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

$$L(M') = \Sigma^* f(x) = \langle M_\emptyset, M'_\emptyset \rangle \iff w \in L(M) \text{ ו } x = \langle M, w \rangle \iff x \in L_{\text{acc}}$$

$$\cdot f(x) \in L_{M_1 \subset M_2} \iff L(M_\emptyset) \subset L(M') \iff$$

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

$$\text{מקרה 1: } .f(x) \notin L_{M_1 \subset M_2} \iff L(M_\emptyset) = L(M'_\emptyset) \text{ ו } f(x) = \langle M_\emptyset, M'_\emptyset \rangle \iff x \neq \langle M, w \rangle$$

$$\text{מקרה 2: } L(M') = \emptyset f(x) = \langle M_\emptyset, M'_\emptyset \rangle \iff w \notin L(M) \text{ ו } x = \langle M, w \rangle$$

$$\cdot f(x) \notin L_{M_1 \subset M_2} \quad L(M') = L(M_\emptyset) \iff$$

לסיכום, הוכחנו רדוקציה מתקיים $L_{\text{acc}} \leq L_{M_1 \subset M_2}$, ומכיון רדוקציה מתקיים $L_{M_1 \subset M_2} \leq L_{\text{acc}}$, $L_{\text{acc}} = L(M'_\emptyset)$.

שיעור 9

מבוא לסיבוכיות זמן

9.1 הגדרה של סיבוכיות זמן

עד כה כל הבעיות החישוביות שעשכנו בהן הניחו שהמשאים שעומדים לרשות מכונת הטיורינג שפותרת אותן הם **בלתי מוגבלים**. כתע נ עבור לעסוק בשאלת מה קורה כאשר אנחנו מוגבלים חלק ממשאים אלו. יש סוגים רבים של ממשאים שנייתן לעסוק בהם, אבל שני הנפוצים ביותר בתיאוריה של מדעי המחשב הם

- **זמן חישוב**,

- **הזיכרון שנדרש לצורך החישוב**.

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

אבל במחשבים שונים האלגוריתם ירוץ זמן זנים שונים בשל

- **יעילות המעבד**,

- **אופטימיזציות שמתבצעות ברמת המעבד**,

- **אופטימיזציות בזמן הקומפליציה**,

וכיווץ בהן.

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

הערה 9.1

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

הגדרה 9.1 זמן הריצה של מכונת טיורינג

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

הגדרה 9.2 סיבוכיות זמן של בעיה/שפה

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

דוגמה 9.1 סיבוכיות זמן של השפה של מחרוזות האוניריות

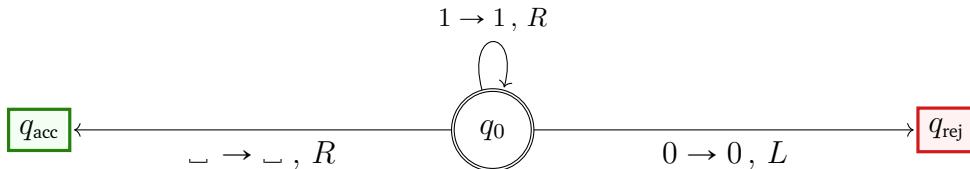
נתבונן על השפה של מחרוזות האוניריות הבאה:

$$L = \{1^n \mid n \geq 0\}.$$

נבנה מכונת טיורינג הבאה שמכריעה אותה:

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

כאשר $\Sigma = \{0, 1\}$ ו- $\Gamma = \{0, 1\}$. המ מצבים והמעברים מתוארים בהתרשים מצבים של מטפה:



בכדי לחשב את הסיבוכיות זמן של L בהתאם למוגנה בפסאודוקוד. על כל קלט w :

- (1) • אם המילה היא מילת הריקה מקבל.
- אחרת ממשיכה לשלב (2).
- (2) • אם התו הנזכר 0 תדחה.
- אחרת אם התו הנזכר הוא 1 מקבל.
- אחרת חוזרת לשלב (2).

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

$$\begin{aligned} &\Leftarrow \text{חסם העליון של מספר צעדי חישוב של } M \text{ על קלט } w \text{ הוא } n \text{ כאשר } |w| = n. \\ &\Leftarrow M \text{ עוצרת בזמן } O(n). \\ &\Leftarrow L \text{ כריעה בזמן } O(n). \\ &\Leftarrow L \in TIME(n). \end{aligned}$$

באופן כללי, אם מכונה על קלט w מבצעת פחותה מ- $|w|$ צעדי חישוב, המשמעות היא המכונה כלול לא קראה את כל הקלט, וזה אינו נקרה נפוץ במיוחד (אם כי הוא בח初恋 קיים). אם כן ברור שמידות זמן חריצת היא תמיד **ביחס לאורך הקלט**.

דוגמה 9.2

בדוגמה זו נבנה מכונת טיורינג M עם סרט ייחיד שמכריע את השפה $L = \{a^n b^n \mid n \geq 0\}$ ואז נחשב את הסיבוכיות זמן שלה.

בנייה המכונת טיורינג

"על קלט w :

- (1) אם התו הנזכר הוא 1 מקבל.

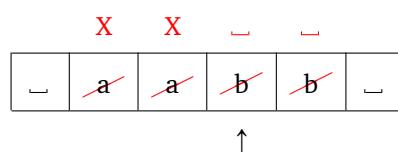
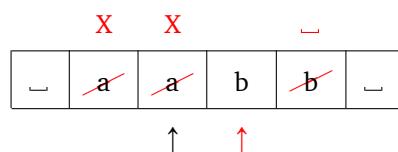
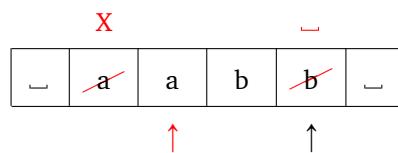
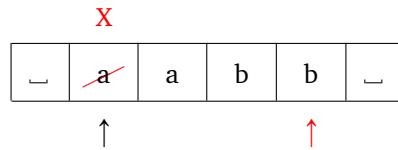
(2) אחרת אם הtau הנקרא הוא $M \Leftarrow b$ דוחה.

(3) אחרת מוחקמת את הtau הנקרא ע"י X .

(4) מזיהה את הראש ימינה עד הtau הראשון משמאלו ל- \sqsubset בקצבה הימין של הקלט.

- אם הtau הוא a או $M \Leftarrow X$ דוחה.

• אחרת מוחקמת את הtau הנקרא ע"י \sqsubset , מזיהה את הראש שמאלה עד הtau הראשון מימין ל- $\sqsubset X$ 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

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

דוגמה 9.3

כעת נבנה מכונת טיורינג מרובת סרטים' M' שמכריעת השפה $L = \{a^n b^n \mid n \geq 0\}$ ונראה שהסיבוכיות זמן שונה מזו של המכונת טיורינג עם סרט יחיד שראינו בדוגמה הקודמת.

בנייה המכונה

w על קלט: M' $O(n)$ (1) מעתיקה את ה- b -ים לסדרת 2 (ותוך כדי לבדוק האם w מהצורה a^*b^*).(2) מוחק את ה- b -ים מסרט 1. $O(n)$

(3) מזיהה את הראשים לתחילת הסרטיים.

(4) אם שני הראשים מצביעים על $_ \leftarrow$ מקבלת.(5) אם אחד הראשים מצביע על $_ \leftarrow$ והשני לא \leftarrow תדחה. $O(n)$ שלבים (3-5):

(6) מזיהה את שני הראשים ימינה וחזרת לשלב (3).

סדרת (1)



סדרת (2)

סיבוכיות זמןנסמן את אורך הקלט ב- $n = |w|$ זמן הריצה של M' הוא $O(n)$ **הגדרה 9.3** $TIME(f(n))$

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

$$TIME(f(n)) = \{L \mid O(f(n)) \text{ זמן } L \text{ בזמן } f(n)\}.$$

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 זו ראיינו בדוגמה 1.5. נבנה מכונת טיריניג M עם סרט ייחיד שמכריעה את השפה $L = \{a^{2^n} \mid n \geq 0\}$ ואז נחשב את הסיבוכיות זמן שלה.

הרעיווןהמכונה תסרק את הסרט שלה משמאל לימין ובכל איטרציה תמחק חצי מה- a -ים שנשארו.

בנייה המכונה

" על קלט $w = M$

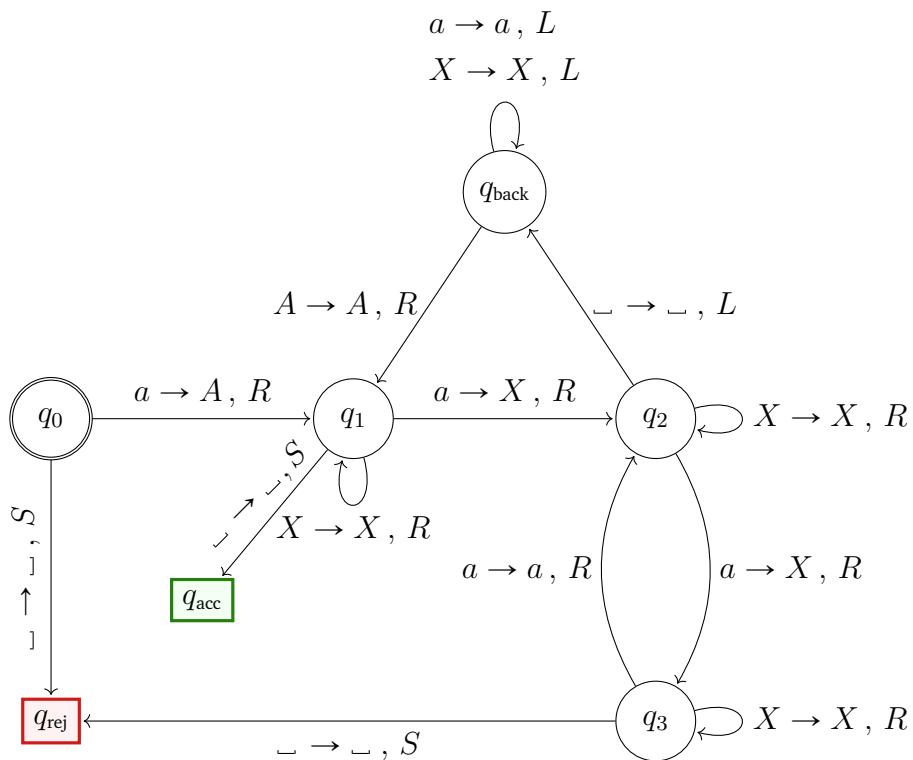
(1) אם הטו הנקרה הוא $_ \Leftarrow$ דוחה.

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

(3) סורקת את הסרט משמאלה לימין ומוחקמת כל a שני:

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

התרשים מציבים של המבנה מתואר באירור למטה:

סיבוכיות זמן

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

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

דוגמה 9.7 חישור ביןארי

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

בנייה המכונה

$\text{SUBTRACT} = \langle \text{על קלט } y, x \rangle$ כאשר y, x שלמים בבסיס ביןארי ו- $y > x$:

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

2) מעמידה את ראש הסרטים 3, 2, 1 על התאים המיושרים של הקצוות הימניים של הקלטים. בפרט, ראש הסרטים 1 ו-2 נמצאים על הספרות הפחות משמעותית של x ו- y , וראש הסרט 3 נמצא בתא שמתוחתם.

3) שלב זה מבצע חישור ללא חוב.

יהי $\sigma_1 \in \{0, 1\}^*$ תו הנקרה הסרט 1 והוא $\sigma_2 \in \{0, 1\}^*$ תו הנקרה הסרט 2.

- אם $(\sigma_1, \sigma_2) = (_, _)$ אז M מקבלת.
- אם $(0, 0, 0), \text{כותבת}, (\sigma_1, \sigma_2) = (0, 0)$ מזיהה את הראשים (L, L, L) וחזרת על שלב זה.
- אם $(1, 0, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (1, 0)$ מזיהה את הראשים (L, L, L) וחזרת על שלב זה.
- אם $(1, 1, 0), \text{כותבת}, (\sigma_1, \sigma_2) = (1, 1)$ מזיהה את הראשים (L, L, L) וחזרת על שלב זה.
- אם $(_, 0, 0), \text{כותבת}, (\sigma_1, \sigma_2) = (_, 0)$ מזיהה את הראשים (S, L, L) וחזרת על שלב זה.
- אם $(_, 1, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (_, 1)$ מזיהה את הראשים (S, L, L) וועברת לשלב הבא.
- אם $(0, _, 0), \text{כותבת}, (\sigma_1, \sigma_2) = (0, _)$ מזיהה את הראשים (L, S, L) וחזרת על שלב זה.
- אם $(1, _, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (1, _)$ מזיהה את הראשים (L, S, L) וחזרת על שלב זה.
- אם $(0, 1, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (0, 1)$ מזיהה את הראשים (L, L, L) וועברת לשלב הבא.

4) שלב זה מבצע חישור כאשר קיים חוב.

- אם $(1, 0, 0), \text{כותבת}, (\sigma_1, \sigma_2) = (1, 0)$ מזיהה את הראשים (L, L, L) וועברת לשלב הקודם.
- אם $(0, 1, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (0, 1)$ מזיהה את הראשים (L, L, L) וחזרת על שלב זה.
- אם $(1, 1, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (1, 1)$ מזיהה את הראשים (L, L, L) וחזרת על שלב זה.
- אם $(0, 0, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (0, 0)$ מזיהה את הראשים (L, L, L) וחזרת על שלב זה.
- אם $(_, 0, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (_, 0)$ מזיהה את הראשים (S, L, L) וחזרת על שלב זה.
- אם $(_, 1, 0), \text{כותבת}, (\sigma_1, \sigma_2) = (_, 1)$ מזיהה את הראשים (S, L, L) וחזרת על שלב זה.
- אם $(0, _, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (0, _)$ מזיהה את הראשים (L, S, L) וחזרת על שלב זה.
- אם $(1, _, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (1, _)$ מזיהה את הראשים (L, S, L) וועברת לשלב הקודם.

סיבוכיות זמן

יהי $|x|, |y| = n$. המכונה SUBTRACT את השני הסרטים 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 $O(n)$ (lienari).

דוגמה 9.8 האלגוריתם החילוק של אוקלידס

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

$$x = qy + r, \quad 0 \leq r < |y|.$$

במקרה הפרטי ש- $x < y < 0$ אז:

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

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

בנייה האלגוריתם

: $0 < y < x$ כאשר y, x שלמים בבסיס בינארי ו- $x = DIVISION$

$$\left. \begin{array}{l} q \leftarrow 0 \\ r \leftarrow x \end{array} \right\} \text{1) מתחילה}$$

2) כל עוד ש- $r \geq y$

$$r \leftarrow r - y \quad \text{3}$$

$$q \leftarrow q + 1 \quad \text{4}$$

5) פלט: q, r .

סיבוכיות זמן

נסמן $| \langle x, y \rangle |$ אורץ הקלט.

• מבצע r איטרציות לכל היותר.

• $y < r$ 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

$$O(n^2).$$

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

משפט 9.1

לכל מכונת טיורינג מרובה סרטים M הרצה בזמן $t(n)$ קיימת מכונת טיורינג סרט יחיד M' השköלה ל- M ורצה בזמן $O(t^2(n))$.

הוכחה:

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

نبנה מכונת טיורינג S עם סרט אחד שרצה בזמן $O(t^2(n))$.

- ראשית נרשום את התוכן של k הסרטים 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:

המכונת טיורינג M

0	1	0	1	0	-	-
↑						

a	a	a	-	-	-	-
↑						

a	b	a	b	-	-	-	-
↑							

המכונת טיורינג S

#	0	1	0	1	0	#	a	a	a	#	a	b	a
↑				↑			↑				↑		

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

בנויות המכונה S

- 1) תחילת S מתחילה את הסרט שלה בלבסוף את התכנים של k הסרטים על הסרט היחיד שלה, עם תו '#' להפריד בין סרטים שונים של M .
- 2) בנוסף S רושמת תו '#' בקצה השמאלי כדי לסמנו את התחלת הקלט ותו '#' בקצת הימין כדי לסמנו את סוף הקלט.

3) ב כדי לסייע צעד חישוב אחד של M בהמוניה S , המכונה S سورקת את הסרט מ- # הראשון ל- # האחרון. ב巡视ה זו S זכרת את המיקומים של ה- k ראשים על פי התאים שמסומנים עם חצים, באמצעות k תא זכרון.

4) אחר כך S מבצעת סיריקה שנייה של הסרט. ב巡视ה זו, לפי הפונקציית המעברים של M , המכונה S מבצעת, לכל $1 \leq i \leq k$

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

5) במצב שהראש של אחד הסרטים של M זו ימינה לתו רוח מצד ימין של סוף הקלט, אז S תוסיף תא אחד עםתו רוח מצד שמאל של ה- # המפריד בין סרטים, ולאחר כך היא תזוז את כל התאים שבצד ימין של התא המוסף מקום אחד ימינה.

סיבוכיות זמן של המכונה

יהי n האורך של התוכן הכי ארוך מthan 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 $O(t(n))$.

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

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

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

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

\Leftarrow הסיבוכיות זמן של S היא:

$$O(t(n)) O(t(n)) = O(t^2(n))$$

9.3 יחס בין הסיבוכיות של מכונת טירינג דטרמיניסטיבית ומכונת טירינג אי דטרמיניסטיבית

משפט 9.2

לכל מכונת טירינג אי-דטרמיניסטיבית N הרצה בזמן $f(n)$, קיימת מכונת טירינג דטרמיניסטיבית D השකולה לה- N ורצה בזמן $2^{f(n)}$.

הוכחה:
בhinintן מכונת טירינג אי-דטרמיניסטיבית N הרצה בזמן $f(n)$ מכונת טירינג דטרמיניסטיבית D באותו אופן כמו בהוכחת השקילות במשפט 4.1.

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

בhinintן 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

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

ולכן זמן הריצה של D חסום ע"י

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

נתיחס כאן לשני החסמים הבאים:



1) חסם פולינומיAli הוא חסם מהצורה n^c עבור $0 < c$ כלשהו.

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

9.4 המחלקה P

הגדרה 9.4 בעיית הכרעה

בעיית הכרעה מוגדרת באופן הבא:

"בhinintן קלט כלשהו, האם הקלט מקיים תנאי מסוים"

דוגמה 9.9

בhinintן מספר n , האם n ראשוני?

כל בעיית הכרעה ניתנת לתאר כשפה שקולה:

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

משפט 9.3

. שפה \equiv בעיית הכרעה .

הגדרה 9.5 אלגוריתם זמן פולינומיAli

אומרים כי אלגוריתם A מכריעה בעיה בזמן פולינומיAli אם קיימים קבוע $0 < c$ כך שזמן הריצה של A על כל קלט w חסום ע"י $O(|w|^c)$.

משפט 9.4 התזה של צירץ' (Church Thesis)

אם קיים אלגוריתם המכריע בעיה בזמן פולינומי, אז קיימת מכונת טיורינג דטרמיניסטיבית המכריע את השפה השקולה לבעיה זו בזמן פולינומי.

. מכונת טיורינג ≡ אלגוריתם מכריע .

הגדרה 9.6 המחלקה P

המחלקה P היא אוסף כל הבעיות (השפות) שקיימים עבורן אלגוריתם (מכונת טיורינג דטרמיניסטיבית) המכריע אותן בזמן פולינומי.

דוגמה 9.10

בדוגמה 9.2 הוכחנו כי קיימת מכונת טיורינג דטרמיניסטיבית שמכריעת עת השפה

$$L = \{a^n b^n \mid n \geq 0\} \in P$$

בזמן $O(n^2)$. לכן $L \in P$.

דוגמה 9.11

תהי L_{prime} השפה הבאה:

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

הוכיחו: $L_{\text{prime}} \in P$

פתרונות:

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

האלגוריתם עצמו עוזה שימוש של המשפט הבא:

משפט: למספר פריך יש מחלק הקטן מ- או שווה לשורשו

יהי n מספרשלם. אם n פריך אז קיים שלם $a \leq \sqrt{n}$ שמחלק את n .

הוכחה:

- נניח בsvilleה כי n פריך ולא קיים מחלק d של n כך ש- $\sqrt{n} < d \leq n$.
- n פריך אז קיימים שלמים a, b כך ש- $n = ab$, כאשר $1 < a, b < n$.
- נניח כי $n < a \leq b$.

על פי ההנחה שלנו, אם a מחלק n אז $\sqrt{n} > a$. בנוסף: $b \geq a > \sqrt{n}$.

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

ז"א $n > n$ וזה סתירה!

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

בנייה האלגוריתם

: $x = PRIME$ על קלט

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

- אם לא \Leftarrow דוחה.

2) אם $n = 1 \Leftarrow$ דוחה.

3) אם $n = 2 \Leftarrow$ מקבל.

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

- מחשב $n \bmod d$

$\Leftarrow n \bmod d = 0$ • מקבל.

5) מקבל.

סיבוכיות זמן

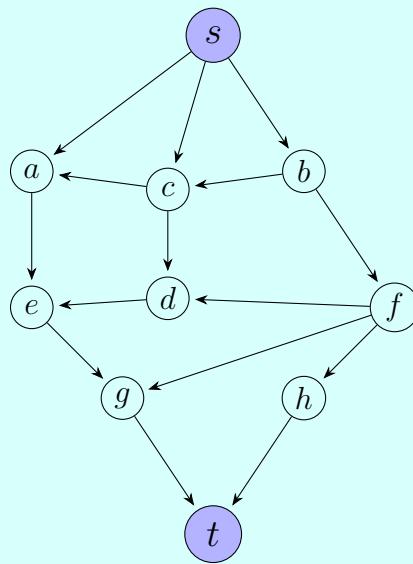
- $PRIME$ מבצע $O(n)$ איטרציות.
- בכל איטרציה $PRIME$ מחשב $n \bmod d$ באמצעות האלגוריתם $DIVISION$, אשר הוכחנו בדוגמה 9.8 שהוא רץ בזמן $O(n^2)$.
- לכן הסיבוכיות זמן של $PRIME$ היא $O(n^3)$.

לפיכך:

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

9.5 בעיית PATH

הגדרה 9.7 בעיית המסלול בגרף מכוון



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

פלט: האם קיים מסלול ב- G מ- s ל- t ?

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

משפט 9.5

$$PATH \in P.$$

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

בנייה האלגוריתם

: $\langle G, s, t \rangle = A$

(1) צובע את s .

(2) מבצע $|V| - 1$ פעמים:

- לכל צלע $(u, v) \in E$:
 - * אם u צבוע \Leftarrow צבע את v .
 - אם t צבוע \Leftarrow מקבל.
 - אחרת \Leftarrow דוחה.
- (3)

הוכחת הנכונות

הוכחה של הכוון \Leftarrow

: $x \in PATH$ אם

. t גרף מכוון וגם $s, t \in V$ כך קיימים מסלול מ- s ל- t .

\Leftarrow קיימים קבוצת צלעות

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

כasher $.k \leq |V| - 1$ $u_1 = s$, $u_k = t$

\Leftarrow לכל $1 \leq i \leq k - 1$, באיטרציה i הקודקוד u_{i+1} יצובע.

\Leftarrow מכיוון ש- $|V| \leq k$ איזי עד סוף הלולאה הקודקוד $u_k = t$ יצובע.

\Leftarrow מכיוון שקודקוד t צובע אז A מקבל.

הוכחה של הטעון \Rightarrow

: $x \notin PATH$ \Leftarrow 2 מקרים:

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

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

לאחר האיטרציה ה- i כל קודקוד שצובע הוא קודקוד שנייתן להגעה אליו מ- s ע"י מסלול מכוון של אורך i לכל היותר.

לא ניתן להגעה לקודקוד ע"י אף מסלול בגרף G .

\Leftarrow לא קיימים סדרת קודקודיים $t \rightarrow \dots \rightarrow u_2 \rightarrow u_3 \rightarrow \dots \rightarrow s$ כך שבסוף הלולאה t יהיה צובע. דוחה. $A \Leftarrow$

סיבוכיות זמן

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

■ $.PATH \in P \Leftarrow PATH \in TIME(N^2) \Leftarrow O(N^2)$ בזמן $PATH$

9.6 הבעייה 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?

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

אנחנו נוכיח כי ניתן להכrie את RELPRIME בזמן פולינומי, כלומר נוכיח $\text{RELPRIME} \in P$ 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 שלמדו בקורסים קודמים:

משפט 9.6 המשפט של האלגוריתם של אוקלידס

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

אם x, y שלמים אז

■ **הוכחה:** ההוכחה היא לא חלק של הקורס ומופיע בסעיף האחרון "הוכחות של משפטי שימושיים" בדף 110. המשפט של אוקלידס הוא אלגוריתם, שמקבל כקלט שני מספרים y, x וpollut את $\gcd(x, y)$. הוא מתבסס על המשפט 9.6. האלגוריתם עצמו הוא כדלקמן:

$x, y = \text{על קלט}$

(1) כל עוד $y \neq 0$:

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

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

(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 $\text{RELPRIME} \in P$ נctrיך למשפט עזר הבא:

משפט 9.7 (משפט עזר)

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

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

משפט 9.8

$$RELPRIME \in P .$$

הוכחה:

نبנה אלגוריתם A המכירע את $RELPRIME$ בזמן פולינומיאי. $RELPRIME$ היא השפה של הבעיה, שמקבלת כקלט שני מספרים שלמים x, y ומחייבת תשובה לשאלת, האם y זרימ. ככלומר:

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

לכן A משתמש באלגוריתם של אוקלידס ($EUCLID(x, y)$) כדי לחשב בנייה האלגוריתם A המכירע:

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

- אם $\gcd(x, y) = 1$ אז A מקבל.

- אחרת A דוחה."

הוכחת הנכונות

הנכונות של A מנובעת ישיר מהנכונות של האלגוריתם האוקלידס, $EUCLID$.

סיבוכיות זמן

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

- לפי משפט ערך 9.7: $\frac{x}{2} < x \bmod y$.

בכל איטרציה, בשלב (2) המשטנה x מקבל את הערך החדש $y \bmod x$.

ניתן לחשב את $y \bmod x$ בעורת האלגוריתם החילוק של אוקלידס $DIVISION$ שרצה בזמן $O(n^2)$ (ראו דוגמה 9.8).

לכן בכל איטרציה הערך החדש של x קטן ממחצית הערך הקודם של x .

לכן אחרי כל איטרציה, x קטן לפחות חצי.

בשלב (3), A מחלייף בין x ו- y , או אחריו כל 2 איטרציות, גם x קטן לפחות חצי וגם y קטן לפחות חצי.

לכן המספר הפעמים המקסימלי שאפשר לבצע שלבים (2) ו- (3) היא $m \triangleq \min(2 \lfloor \log_2 x \rfloor, 2 \lfloor \log_2 y \rfloor)$.

לכן m הוא חסם עליון של מספר האיטרציות ש- $EUCLID$ מבצע.

אם $n \leq m$ הוא האורך של הקלט $\langle x, y \rangle$ לנכון $EUCLID$ מבצע $O(n)$ איטרציות.

כל איטרציה מבצעת $DIVISION$ כדי לחשב $x \bmod y$ אשר רץ בזמן $O(n^2)$.

רץ בזמן $O(n^3)$. ככלומר:

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

לכן:

$$RELPRIME \in P .$$



9.7 *הוכחות של משפטי שימושיים

משפט 9.9 המשפט של האלגוריתם של אוקלידסאם 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 זו היא רק להוסיף הבנה להוכחה של משפט 9.8 לסיבוכיות זמן של *RELPRIME* למטרה. היא לא הוכחה שאותם תיבחנו עלייה ואפשר לדלג עליה.

נתחיל אם משפט החלוק של אוקלידס, שאומר שאם x, y שלמים אז קיימים שלמים q ו- $r \leq 0$ כך ש:

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

נגיד $d \triangleq \gcd(x, y)$.
מכיוון ש- d הוא מחלק משותף של x ו- y אז $d \mid x$ ו- $d \mid y$ וגם $d \mid (x \bmod y)$. לכן בaczות מסוואה (1*):

$$(d \mid x) \wedge (d \mid y) \xrightarrow{\text{משוואת (1*)}} d \mid (x \bmod y)$$

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

$$d \mid \gcd(y, x \bmod y). \quad (2*)$$

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

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

וז"א וגם $\bar{d} \mid x$ אז בהכרח:

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

לסיכום, לפי משוואות (2*) ו- (3*):
 $d \mid \bar{d} \wedge \bar{d} \mid d$.

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

משפט 9.10 (משפט עזר)אם $x \bmod y < \frac{x}{2}$ אז $x > y$.**הוכחה:** יש שני מקרים:

$$y \leq \frac{x}{2} \quad (1)$$

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

נווכיח את הטענה עבור שני המקרים.

מקרה 1: $y \leq \frac{x}{2}$

לפי משפט החלוק של אוקלידס אם x, y שלמים Überom $x > y$ אז קיימים $q = x \text{ mod } y$ ו- $r = qy + r = \left\lfloor \frac{x}{y} \right\rfloor y + (x \text{ mod } y)$.

$x \text{ mod } y < y \leq \frac{x}{2}$ לפיכך $y \leq \frac{x}{2}$ ובפרט $r < y$ וגם $0 \leq r < y$.

מקרה 2: $y > \frac{x}{2}$

לפי משפט החלוק של אוקלידס אם x, y שלמים Überom $x > y$ אז קיימים $q = x \text{ mod } y$ ו- $r = qy + r = \left\lfloor \frac{x}{y} \right\rfloor y + (x \text{ mod } y)$.

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

$$x = qy + r = (1)(y) + ry + (x \text{ mod } y).$$

מכאן

$$x - y = x \text{ mod } y.$$

כעת נציב את ההנחה ההתחלטית $x - y < \frac{x}{2}$ ונקבל $y > \frac{x}{2}$

$$x \text{ mod } y < \frac{x}{2}.$$



שיעור 10

המחלקה P והמחלקה NP

10.1 המחלקה P

הגדרה 10.1 בעיית הכרעה

בעיית הכרעה מוגדרת באופן הבא:

"בhinatan קלט כלשהו, האם הקלט מקיים תנאי מסוימים ? "

דוגמה 10.1 דוגמה של בעיית הכרעה

לדוגמה, בהינתן מספר n , האם n ראשוני?

משפט 10.1 שקיים בין בעיה לשפה

כל בעיה הכרעה ניתן לתאר כשפה שוקולה:

. בעיה הכרעה \equiv שפה

דוגמה 10.2

לדוגמה, הבעיה הכרעה הבאה:

"בhinatan מספר n , האם n ראשוני? "

ניתנת לרשום כשפה הבאה:

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

הגדרה 10.2 אלגוריתם זמן פולינומייאלי

אומרים כי אלגוריתם A מכרייע בעיה בזמן פולינומייאלי אם קיימים קבוע $c > 0$ כך שזמן הריצה של A על קלט w חסום ע"י $O(|w|^c)$.

התזה של צרצ' טירונג אומר שאם קיימים אלגוריתם המכרייע בעיה בזמן פולינומייאלי, אז קיימת מכונת טירונג דטרמיניסטיבית המכרייע את השפה השוקולה בעיה או בזמן פולינומייאלי.

. אלגוריתם מכרייע \equiv מכונת טירונג דטרמיניסטיבית

הגדרה 10.3 המחלקה P

המחלקה P היא אוסף כל הבעיה (השפות) שקיימים עבורן אלגוריתם (מכונת טירונג דטרמיניסטיבית) המכרייע אותן בזמן פולינומייאלי.

10.2 דוגמאות לבעיות ב- P

(1)

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

(2)

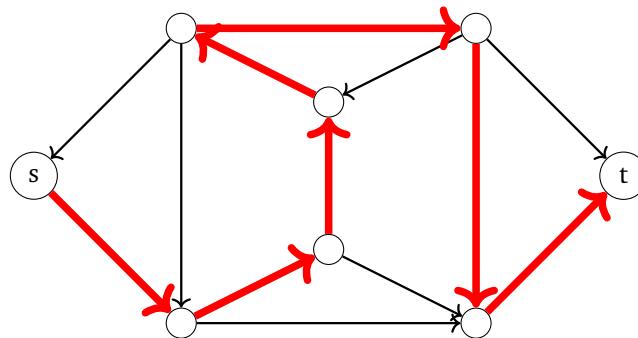
$$RELPRIME = \{ \langle x, y \rangle \mid x \text{ ו- } y \text{ זרים} \} \in P$$

10.3 בעיית המסלול המילטוני $HAMPATH$

הגדירה 10.4 בעיית $HAMPATH$

בاهינתן גרף מכון $G = (V, E)$ 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 בדיקת פעם אחת.

לדוגמה:



הגדירה 10.5 בעיית $HAMPATH$

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

פלט: האם G מכיל מסלול המילטוני מ- s ל- t ?

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

נשאל שאלת: האם $HAMPATH \in P$?

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

- בהינתן קלט $\langle G, s, t \rangle \in HAMPATH$, האם $\langle G, s, t \rangle \in ?$

• נעה על שאלת אחרת:

בاهינתן קלט $\langle G, s, t \rangle$, ומחרוזת y , האם $\langle G, s, t \rangle \in HAMPATH$?

- ניתן לבדוק האם y היא מסלול המילטוני מ- s ל- t ב- G בזמן פולינומיAli ולענות בהתאם.
- במקרה זה, אומרם כי $HAMPATH$ ניתנת לאימות בזמן פולינומיAli.

10.4 אלגוריתם אימות

הגדרה 10.6 אלגוריתם אימות

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

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

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

הערה 10.1

זמן ריצה של אלגוריתום אימות נמדד ביחס לגודל הקלט $|w|$.

אלגוריתם אימות פולינומיAli אם הוא רץ בזמן פולינומיAli.

10.5 המחלקה NP

הגדרה 10.7 המחלקה NP

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

משפט 10.2 $HAMPATH \in NP$

בעיית המסלול ההAMILTONI : $HAMPATH$

קלט: גראף מכוכון $G = (V, E)$ ושני קודקודים $s, t \in V$

פלט: האם G מכיל מסלול המילוטוני מ- s ל- t ?

$$HAMPATH = \{ \langle G, s, t \rangle \mid G \text{ גראף מכוכון המכיל מסלול המילוטוני מ- } s \text{ ל- } t \}$$

הוכחה: הוכחנו כי $HAMPATH \in NP$

הוכחה: נבנה אלגוריתם אימות V עבור $HAMPATH$.

: על קלט $\langle G, s, t \rangle, y$:

1) בודק האם y היא סדרה של

$$u_1, u_2, \dots, u_n$$

השונים זה מזה.

- אם לא \Leftarrow דוחה.

2) בודק האם $u_n = t$ ו- $u_1 = s$.

- אם לא \Leftarrow דוחה.

(3) בודק שכל הצלעות (u_i, u_{i+1} $1 \leq i \leq n$) קיימות ב- G .

- אם כן \Leftarrow מקבל.
- אם לא \Leftarrow דוחה.

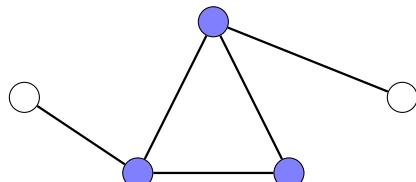
נכונות

- זמן הריצה של האלגוריתם הוא פולינומילי בגודל הקלט.
- אם 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 G לא מכיל מסלול המילטוני מ- s ל- t \Leftarrow לכל y , האלגוריתם ידחה את הזוג $(\langle G, s, t \rangle, y)$.

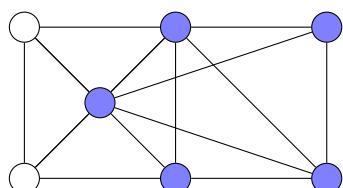
■

הגדרה 10.8 קליקה

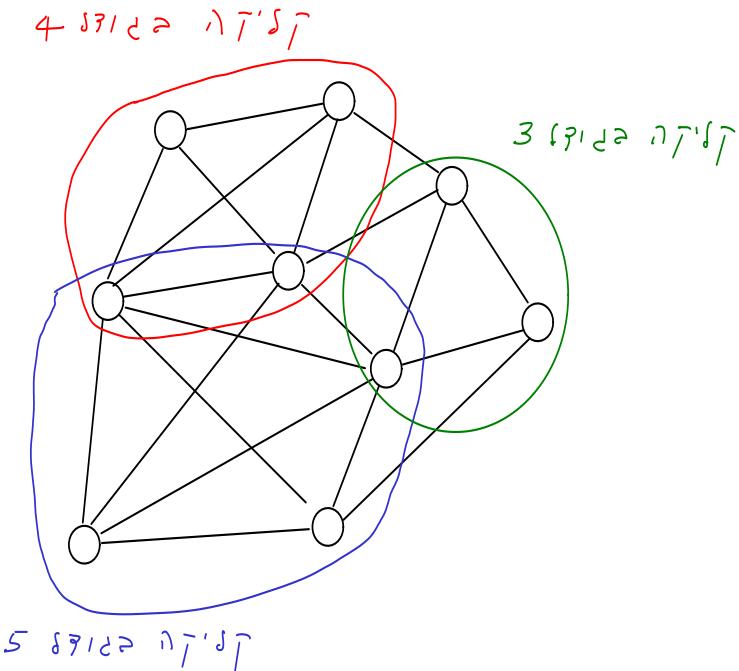
בහינתן גרף לא מכוון $G = (V, E)$, קליקה ב- G היא תת-קובוצה של קודקודים $C \subseteq V$ כך שלכל שני קודקודים $u, v \in C$ מתקיים $(u, v) \notin E$.



קליקה בגודל 3



קליקה בגודל 5

**הגדרה 10.9 בעיית הקליקה**

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

פלט: האם G קליקה בגודל k ?

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

משפט 10.3 $CLIQUE \in NP$

$$CLIQUE \in NP.$$

הוכחה: נבנה אלגוריתם אimotoת V עבור $CLIQUE$.

:($\langle G, k \rangle, y$) על קלט V

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 $S = \{x_1, x_2, \dots, x_n\}$ ומספר t .

פלט: האם קיימת תת-קבוצה של S שסכום איבריה שווה t ?

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

משפט 10.4 $\text{SubSetSum} \in NP$

$\text{SubSetSum} \in NP$.

הוכחה: נבנה אלגוריתם אimotoות 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 למוכנות טיריניג א-דטרמיניסטיבית

NP=Non-deterministic polynomial-time.

משפט 10.5

לכל בעיה A :

אם ורק אם קיימת מוכנות טיריניג א-דטרמיניסטיבית המכ裏עה את A בזמן פולינומילי.

דוגמה 10.3

نبנה מוכנות טיריניג א-דטרמיניסטיבית M המכ裏עה את $CLIQUE$ בזמן פולינומילי.

$\langle G, k \rangle$ על קלט $= M$:

- בוחרת באופן א-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 אimotoות \equiv מ"ט א"ד.

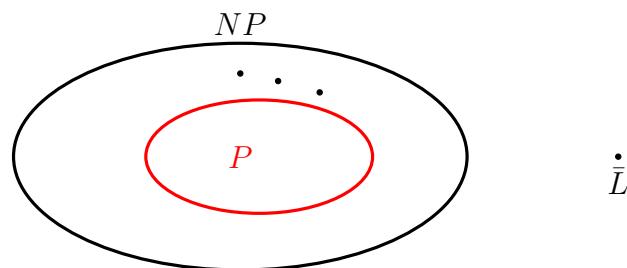
10.7 הקשר בין המחלקה P ו- NP

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 $\bar{A} \in P$ אז גם $A \in P$.

הגדרה 10.11

$$Co\,NP = \{A \mid \bar{A} \in NP .\}$$

לדוגמה:

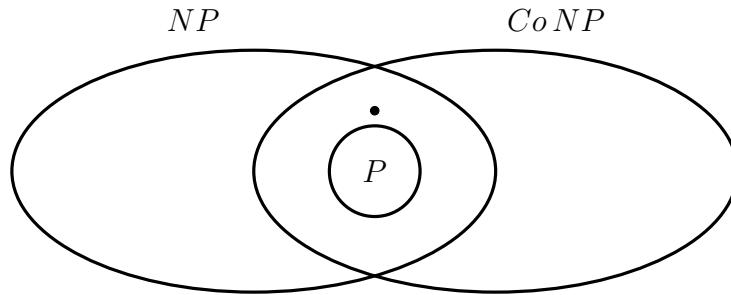
$$\overline{HAMPATH} \in Co\,NP .$$

$$\overline{CLIQUE} \in Co\,NP .$$

שאלה פתוחה: האם $?NP = Co\,NP$

משפט 10.8

$$P \subseteq NP \cap Co\,NP .$$



שאלה פתוחה: האם $?P = NP \cap Co\,NP$

נדון בשאלה המרכזית: האם $?P = NP$

הגדרה 10.12 פונקציה פולינומיאלית

בاهינתן פונקציה $\Sigma^* \rightarrow \Sigma^*$, אומרים כי f חסיבה בזמן פולינומיאלי אם קיים אלגוריתם (מ"ט דטרמיניסטי) המחשב את f בזמן פולינומיאלי.

הגדרה 10.13 רדוקציה פולינומיאלית

בاهינתן שתי בעיות A ו- B . אומרים כי A ניתנת לרדוקציה פולינומיאלית ל- B , ומסומנים $B \leqslant_P A$, אם קיימת פונקציה $\Sigma^* \rightarrow \Sigma^*$ המקיים:

(1) f חסיבה בזמן פולינומיאלי

(2) לכל $w \in \Sigma^*$:

$$w \in A \Leftrightarrow f(w) \in B .$$

משפט 10.9 משפט הרדוקציה

לכל שתי בעיות A ו- B , אם $A \leqslant_P B$ אז

(1) אם $A \in P$ אז $B \in P$.

(2) אם $A \in NP$ אז $B \in NP$.

מסקנה מ- (1) ו- (2):

(3) אם $B \notin P$ אז $A \notin P$.

(4) אם $B \notin NP$ אז $A \notin NP$.

הוכחה: מכיוון שקיימת רדוקציה D של A ל- B , קיימת פונקציה f חסיבה בזמן פולינומיאלי המקיים, לכל $w \in \Sigma^*$,

$$w \in A \Leftrightarrow f(w) \in B .$$

יהי M_f האלגוריתם שמחשבת את f בזמן פולינומיאלי.

(1) נוכח כי אם $B \in P$ אז $A \in P$.

יהי M_B האלגוריתם שמכריע עת B בזמן פולינומיאלי. נבנה אלגוריתם M_A המכrüע את A בזמן פולינומיאלי.

התאור של M_A

= על כל קלט w :

1. מחשב את $f(w)$ ע"י M_f .

2. מריץ את M_B על $f(w)$ ועונה כמותה.

נוכח כי M_A מכrüע את A בזמן פולינומיאלי:

- אם $M_A \Leftarrow f(w) \Leftarrow M_B \Leftarrow f(w) \in B \Leftarrow w \in A$ מקבל את w .

- אם $M_A \Leftarrow f(w) \Leftarrow M_B \Leftarrow f(w) \notin B \Leftarrow w \notin A$ דוחה את w .

נוכח כי זמן הריצה של M_A הוא פולינומיאלי 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

$$P_f(|w|) + P_B(|f(w)|)$$

מכיוו ש- $|f(w)| \leq P_f(|w|)$, הזמן הריצה של M_A על w חסום ע"י

$$P_f(|w|) + P_B(P_f(|w|)) = P_f(|w|) + (P_B \circ P_f)(|w|)$$

כאשר $P_B \circ P_f$ מסמן את ההרכבה של שני פולינומים. לכן M_A רץ בזמן פולינומייאלי בגודל $|w|$.



שיעור 11

NP שלמות

11.1 המחלקות NPH ו- NP

הגדרה 11.1 NP-hard

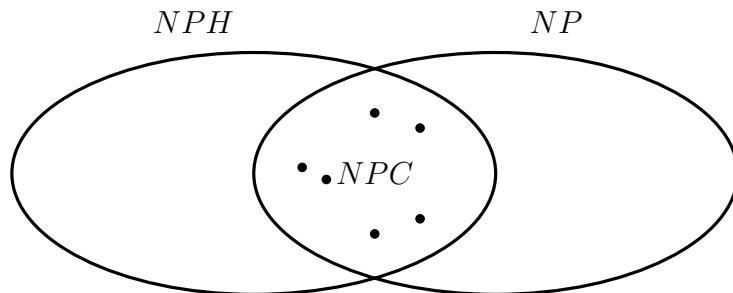
בשפה B נקראת NP קשה אם לכל בעיה $A \in NP$ קיימת רדוקציה . $A \leqslant_P B$

הגדרה 11.2 NP-complete

בשפה B נקראת NP שלמה אם

$$B \in NP \quad (1)$$

(2) לכל בעיה $A \in NP$ קיימת רדוקציה . $A \leqslant_p B$



משפט 11.1

אם B בעיה NP שלמה וגם $P = NP$ אז $A \in P$ $\forall A \in NP$

הוכחה:

- הוכחנו כבר שה- . $P \subseteq NP$
- נוכח כי $NP \subseteq P$

לכל בעיה $A \subseteq NP$ קיימת רדוקציה $A \leqslant_P B$ ומכיון ש- $B \in P$, מושפט הרדוקציה מתקיים . $A \leqslant_P B$

מסקנה 11.1

אם $\bar{A} \leqslant_P \bar{B}$ אז $A \leqslant_P B$

משפט 11.2

אם $A \leqslant_p C$ ואו $B \leqslant_p C$ וגם $A \leqslant_p B$

הוכחה:

משפט 11.3

תהי B בעיה NP -שלמה. אז לכל בעיה $C \in NP$, אם $B \leq_p C$ אז גם C היא NP -שלמה.

הוכחה: מכיוון ש- B היא NP -שלמה, לכל בעיה $A \in NP$ קיימת רדוקציה $B \leq_p A$. מכיוון ש- מהטרנסיטיביות מתקיים $A \leq_p C \leq_p B$ לכל בעיה $A \in NP$ ולכן C היא NP -שלמה.

11.2 בעית הספיקות**הגדרה 11.3**

נוסחת ϕ היא נוסחה בוליאנית מעל n משתנים x_1, x_2, \dots, x_n המכילה m פסוקיות כאשר כל פסוקית מכילה אוסף של ליטרלים ($x_i \setminus \bar{x}_i$) המוחברים ע"י OR (\vee) בוליאני והפסוקיות מחוברים ע"י AND (\wedge) בוליאני.

לדוגמה

$$\phi = \left(x_1 \vee \bar{x}_2 \vee x_4 \vee \bar{x}_7 \right) \wedge \left(x_3 \vee x_5 \vee \bar{x}_8 \right) \wedge \dots$$

הגדרה 11.4 נוסחת CNF ספיקה

נוסחת ϕ היא ספיקה אם קיימת השמה למשתנים x_1, x_2, \dots, x_n ע"י $T \setminus F$ כך ש- ϕ מקבלת ערך T , כלומר בכל פסוקית ישנו לפחות ליטרל אחד שקיבל ערך T .

11.3 בעית SAT**הגדרה 11.5 בעית SAT**

קלט: נוסחת ϕ , פלט: האם ϕ ספיקה?

$$SAT = \{ \langle \phi \rangle \mid \text{נוסחת CNF ספיקה } \phi \}$$

משפט 11.4

$$SAT \in NP$$

הוכחה: בניית אלגוריתם אimotoת V עבור SAT .

: על קלט $V = \langle \phi \rangle, y$

(1) בודק האם y היא השמה למשתנים x_1, x_2, \dots, x_n .

- אם לא $\Leftrightarrow 3CNF$ דוחה.

(2) בודק האם השמה זו מספקת את ϕ .

- אם כן \Leftrightarrow מקבל.
- אם לא \Leftrightarrow דוחה.

■

11.4 משפט קוק לוין

משפט 11.5 (1973) משפט קוק לוין

הבעית SAT היא NP - שלמה.

רעיון ההוכחה:

$.A \leq_p SAT, A \in NP$ לכל $:w \in \Sigma^*$ לכל

$$w \in A \Leftrightarrow f(w) \in SAT,$$

$$\text{כאן } f(w) = \langle \phi_w \rangle$$

מסקנה 11.2

$$P = NP \Leftrightarrow SAT \in P.$$

11.5 גרסאות של $kSAT$

ישנן לכל היותר k ליטרלים בכל פסוקית:

$$.1SAT \in P \bullet$$

$$\phi = x_1 \wedge \bar{x}_2 \wedge x_3 \wedge \dots$$

$$.2SAT \in P \bullet$$

$$\phi = (x_1 \vee \bar{x}_2) \wedge (x_2 \vee x_4) \wedge \dots$$

$$.3SAT \in P \text{ - שלמה.} \bullet$$

3SAT בעית 11.6

הגדרה 11.6 בעית 3SAT

קלט: נוסחת $\phi, 3CNF$

פלט: האם ϕ ספיקת?

$$3SAT = \{\langle \phi \rangle \mid \phi \text{ נוסחת } 3CNF \text{ ספיקת}\}$$

משפט 11.6 $3SAT \in NP$ שלמה. $3SAT \in NP$ שלמה.

הוכחה:

ישקיימים את השני תנאים הבאים:

(1) $3SAT \in NP$

ניתן לבנות אלגוריתם אimoto עבור $SAT \in NP$ דומה לאלגוריתם האimoto עבור SAT שבנו בהוכחה של המשפט קוק-ליין 11.5 לעילו.

(2) $3SAT \in NP$ קשה ע"י רדוקציה

$SAT \leq_p 3SAT$.

ואז בגלל ש- $SAT \in NP$ שלמה (לפי משפט קוק-ליין 11.5) ומכיון ש- $3SAT \in NP$ אז לפי משפט האסימפטוטית 11.2 גם $3SAT \in NP$ - שלמה.

קיים פונקציה הרדוקציה $SAT \leq_p 3SAT$

כעת נבנה את פונקציה הרדוקציה מ- SAT ל- $3SAT$.ראשית נזכיר כי כל נוסחה בוליאנית ϕ ניתנת לרשום בצורה CNF בזמן פולינומיAli.

בහינתנו נוסחת CNF , ϕ (הקלט של SAT) נבנה בזמן פולינומיAli נוסחת ϕ' (הקלט של $3SAT$) ואז נוכחים מתקיים

$\langle\phi'\rangle \in 3SAT \Leftrightarrow \langle\phi\rangle \in SAT$.

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 a_k המכיל $k > 3$ ליטרלים, ניצור אוסף C' של פסוקיות שבו כל פסוקית מכילה 3 ליטרלים, ע"י הוספה 3 משתנים y_1, y_2, \dots, y_{k-3} :

$$C' = (a_1 \vee a_2 \vee y_1) \wedge (\bar{y}_1 \vee a_3 \vee y_2) \wedge \dots \wedge (\bar{y}_{i-2} \vee a_i \vee y_{i-1}) \wedge \dots \wedge (\bar{y}_{k-3} \vee a_{k-1} \vee a_k)$$
.

בפרט, עבור כל פסוקית $(a_1 \vee a_2 \vee \dots \vee a_k)$ נניח $a_i = 1$ הוא הלiteral הראשון שווה ל- 1. אז

• נשים $y_j = 1$ לכל $1 \leq j \leq i-2$

• ונסים $y_j = 0$ לכל $i-1 \leq j \leq k-3$

סימנו להגדר את הפונקציה הרדוקציה.

כעת נוכחים כי הפונקציה הזאת מקיימת את התנאי ההכרחי

$$\langle\phi'\rangle \in 3SAT \Leftrightarrow \langle\phi\rangle \in SAT$$
.

כיון \Leftarrow :

נניח כי $\langle\phi\rangle \in SAT$ ותהי X השמה המספקת את ϕ .
nocich שקיימת השמה X' מתאימה המספקת את ϕ' .

- בכל פסוקית C של ϕ , עבור הליטרלים a_1, a_2, \dots, a_k ניתן אותם ערכים כמו ב- X .
- מכיוון ש- X מספקת את ϕ , בכל פסוקית $C = (a_1 \vee a_2 \vee \dots \vee a_k)$ יש לפחות ליטרל אחד שקיבל ערך 1. נניח $a_i = 1$. אז על פי ההגדרה של פונקציית הרודוקציה:

$$\begin{aligned} * \text{ נשים } 1 = y_j \text{ לכל } 2 \leq j \leq i-1 \\ * \text{ ונשים } 0 = y_j \text{ לכל } i-1 \leq j \leq k-3 \end{aligned}$$

באופן זהה אנחנו נזכיר אוסף C' של פבוקיות עם המבנה הבא:

$$\begin{aligned} & \left(a_1 \vee a_2 \vee y_1 \right) \wedge \left(\bar{y}_1 \vee a_2 \vee y_2 \right) \wedge \dots \wedge \left(\bar{y}_{i-3} \vee a_{i-1} \vee y_{i-2} \right) \wedge \left(\bar{y}_{i-2} \vee a_i \vee y_{i-1} \right) \wedge \left(\bar{y}_{i-1} \vee a_{i+1} \vee y_i \right) \\ & \wedge \dots \wedge \left(\bar{y}_{k-3} \vee a_{k-1} \vee a_k \right) \end{aligned}$$

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 שקיימות השמה X המספקת את ϕ .

נסתכל על פסוקית $C = (a_1 \vee a_2 \vee \dots \vee a_k)$ נניח בשלילה שלא קיימת השמה X המספקת את C . אז בהכרח

$$a_1 = a_2 = \dots = a_k = 0$$

לפי זה, באוסף פסוקיות C' שנתקבל על פי ההגדרה של פונקציית הרודוקציה, $y_j = 1$ לכל $1 \leq j \leq k-3$ $y_j = 0$ לכל $y_1 = y_2 = \dots = y_{k-3} = 0$. כלומר מתקיים

$$C' = \left(a_1 \vee a_2 \vee y_1 \right) \wedge \left(\bar{y}_1 \vee a_3 \vee y_2 \right) \wedge \dots \wedge \left(\bar{y}_{i-2} \vee a_i \vee y_{i-1} \right) \wedge \dots \wedge \left(\bar{y}_{k-3} \vee a_{k-1} \vee a_k \right)$$

הפסוקית האחרונה $\left(\bar{y}_{k-3} \vee a_{k-1} \vee a_k \right)$ אינה מסופקת.
לכן C' אינה מסופקת, בסתיו לכך X' מספקת את ϕ' .

ולכן $\langle \phi \rangle \in SAT$

$SAT \leq 3SAT$

כעת נוכיח כי הרודוקציה זו היא זמן פולינומיAli.

סיבוכיות

הчисוב של הפונקציה מתבצע בזמן פולינומיAli. ספציפי, אם האורך של הנוסחה ϕ הוא n אז הרודוקציה היא $O(n)$.



11.7 הוכחת משפט קוק לוין*

משפט 11.7 משפט קוק לוין

הבעית SAT היא NP - שלמה.

הוכחה:

חישפה מלאה: ההוכחה הבאה מتبוססת על ההוכחה שנותונה בספר של Sipser.

על פי הגדירה 11.2 יש להוכיח שני התנאים הבאים מתקיימים:

תנאי 1: $SAT \in NP$

תנאי 2: $A \in NP \leq_p SAT$

ראשית נוכיח כי $SAT \in NP$:

כדי להוכיח כי SAT שייכת ל- NP , נוכיח כי אישור המרכיב מהשמה מספקת עבור נוסחת קלט ϕ ניתן לאיומות בזמן פולינומייאלי.

נניח כי $n = |\phi|$. כאמור ב- ϕ מופיעים n ליטרלים. "א" השמה כלשהי דורשת n משתני בוליאניים לכל היוטר.

- אלגוריתם האimotoות מחליף כל משתנה בנוסחה בערך המתאים לו על פי ההשמה. השלב זהה הוא $(n)O$.

- אחר כך האלגוריתם מחשב את ערכו של הביטוי:

* נניח כי הנוסחה ϕ מכילה k דורות של סוגרים בתוך סוגרים.

* החישוב מתחילה עם החישובים של הביטויים בתוך הסוגרים הכל בפנים.

* יש n סוגרים הći-בפנים לכל היוטר, וכל אחד של סוגרים אלה מכיל n ליטרלים לכל היוטר. לכן החישוב זהה הוא $(n^2)O$.

* יש k דורות של סוגרים לכן החישוב כולל הוא $(kn^2)O$.

- בסה"כ הסיבוכיות זמן הריצה היא

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

לפיכך אישור של השמה כלשהי מatabase בזמן פולינומייאלי.

- אם ערכו של הביטוי הוא 1 הנוסחה ספיקה.

הוכחנו כי $SAT \in NP$. עכשו נוכיח כי $SAT \leq_p A$.

תהי N מ"ט אי-דטרמיניסטי זמני-פולינומייאלית 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, כאמור אורץ הקלט הוא n .

הסימנים w_n, \dots, w_1 מסמנים את התווים של הקלט.

- בתא הראשון בכל שורה יש #, ולאחר מכן רשומה הקונפיגורציה של N . בסוף הקונפיגורציה בכל שורה יש #.
 - אחרי #- בקצתה הימין של המילה, בכל תא ישתו רוח עד הסוף של השורה. התוויי רוח לפני המשבצת הראשונה של הקונפיגורציה לא מופיעים בטבלה.
 - האורך של כל שורה הוא בדיק n^k תאים.
 - בטבלה יש בדיק n^k שורות לסייע בהאה:
 - המכונת טירינג מבצעת n^k צעדים לכל היותר.
 - בכל צעד המ"ט עוברת לkonfigurציה חדשה.
 - בכל שורה רשומה konfigurציה.
 - בסה"כ יש n^k שורות עבור ה- n^k konfigurציות שונות האפשריות.

אנו אומרים כי טבלה כלשהי היא **טבלה המבלטת** אם באחת השורות יש קו ניגורציה אשר N מקבלת אותה.

בעזרת הטבלה נתאר את הרדוקציה זמו-פולינומיאלית f משפה A כלשוי L - SAT .

הפונקציה הרדוקציה f מקבלת קלט w ומחזירה נוסחה $\phi = f(w)$, אשר לפי ההגדרה של פונקציית הרדוקציה, עומדת בתנאי הבא:

$$w \in A \quad \Leftrightarrow \quad f(w) \in SAT .$$

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 (i, j) של הטבלת הקונפיגורציות נגידר משתנה בוליאני $x_{i,j,s}$ לכל המסתנה x_{ijs} מוגדר על פי התנאי

$$x_{ijs} = 1$$

אם בתא ה- j,i של הטבלה יש $s \in C$. למשל, אם בתא ה- $(2,5)$ של הטבלה מופיע התו a אז

$$x_{2,5,a} = 1$$

בעוד

$$x_{2,5,b} = 0 .$$

במובן זהה, התכנים של כל התאים של הטבלה מסומנים על ידי המשתנים של ϕ .

עכשו נבנה נוסחה ϕ על סמך התנאי שהשמה מספקת של ϕ תהיה מתאימה לטבלה המתקבלת של N . נגיד:

$$\phi = \phi_{\text{cell}} \wedge \phi_{\text{start}} \wedge \phi_{\text{move}} \wedge \phi_{\text{acc}} . \quad (11.1)$$

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

• הנוסחה ϕ_{cell}

כפי שמצוין לעיל, אם המשתנה $x_{i,j,s}$ "דולק", כלומר אם $x_{i,j,s} = 1$, זאת אומרת שיש סימן s בתא ה- i,j של הטבלה. אנחנו רוצים להבטיח שהשמה כלשהי בנוסחה אשר מתאימה לקונפיגורציה של הטבלה, מדliquה בדיק משותה אחד לכל תא של הטבלה. למטרה זו נגיד ϕ_{cell} כך:

$$\phi_{\text{cell}} = \bigwedge_{1 \leq i, j \leq n^k} \left[\left(\bigvee_{s \in C} x_{i,j,s} \right) \wedge \left(\bigwedge_{\substack{s,t \in C \\ s \neq t}} (\bar{x}_{i,j,s} \vee \bar{x}_{i,j,t}) \right) \right] \quad (11.2)$$

* האיבר הראשון בסוגרים מרובעים, מבטיח שלכל תא של הטבלה, לפחות משתנה אחד דולק.

* האיבר השני מבטיח שעבור כל תא של הטבלה, משתנה אחד לכל היוטר דולק.

לפיכך כל השמה מספקת עומדת בתנאי שיהיה בדיק סימן אחד, s , בכל תא של הטבלה.

• הנוסחה ϕ_{start}

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

$$\begin{aligned} \phi_{\text{start}} = & x_{1,1,\#} \wedge x_{1,2,q_0} \wedge x_{1,3,w_1} \wedge x_{1,4,w_2} \\ & \wedge \dots \wedge \\ & x_{1,n+2,w_n} \wedge x_{1,n+3,_} \wedge \dots \wedge x_{1,n^k-1,_} \wedge x_{1,n^k,\#} \end{aligned} \quad (11.3)$$

• הנוסחה ϕ_{acc}

הנוסחה ϕ_{acc} מבטיחה שקיים טבלה קונפיגורציה אשר המ"ט N מקבל 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:

$$\phi_{\text{acc}} = \bigvee_{1 \leq i, j \leq n^k} x_{i,j,q_{\text{acc}}} \quad (11.4)$$

• הנוסחה ϕ_{move}

הנוסחה ϕ_{move} מבטיחה שכל שורה של הטבלה היא "שורה חוקית".
כלומר בכל שורה, הקונפיגורציה היא כך שאפשר הגיע אליה על ידי תזוזה חוקית של N מהקונפיגורציה הקודמת שMOVEDה בשורה אחרת מעלה.

תזוזה חוקית בין כל שתי קונפיגורציות נקבעת על ידי הפונקציה 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 $1 \leq i \leq n^k - 1$ מתקיים

$$c_i \vdash_N c_{i+1} .$$

במנוחי הטבלה, אפשר להגיד תזוזה חוקית בין כל שתי שורות על ידי תת-טבלה מסדר 3×2 שמכילה 3
תאים מתאימים של שתי שורות שכנות.
 مكان ואילך אנחנו נקרא לתת-טבלה כזאת "חלון".

למטה יש דוגמאות של חלונות חוקיים:

<table border="1"><tr><td>a</td><td>q_1</td><td>b</td></tr><tr><td>q_2</td><td>a</td><td>c</td></tr></table>	a	q_1	b	q_2	a	c	<table border="1"><tr><td>a</td><td>q_1</td><td>b</td></tr><tr><td>a</td><td>a</td><td>q_2</td></tr></table>	a	q_1	b	a	a	q_2	<table border="1"><tr><td>a</td><td>a</td><td>q_1</td></tr><tr><td>a</td><td>a</td><td>b</td></tr></table>	a	a	q_1	a	a	b
a	q_1	b																		
q_2	a	c																		
a	q_1	b																		
a	a	q_2																		
a	a	q_1																		
a	a	b																		
<table border="1"><tr><td>#</td><td>b</td><td>a</td></tr><tr><td>#</td><td>b</td><td>a</td></tr></table>	#	b	a	#	b	a	<table border="1"><tr><td>a</td><td>b</td><td>a</td></tr><tr><td>a</td><td>b</td><td>q_2</td></tr></table>	a	b	a	a	b	q_2	<table border="1"><tr><td>b</td><td>b</td><td>b</td></tr><tr><td>c</td><td>b</td><td>b</td></tr></table>	b	b	b	c	b	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:

<table border="1"><tr><td>a</td><td>b</td><td>a</td></tr><tr><td>a</td><td>a</td><td>a</td></tr></table>	a	b	a	a	a	a	<table border="1"><tr><td>a</td><td>q_1</td><td>b</td></tr><tr><td>q_1</td><td>a</td><td>a</td></tr></table>	a	q_1	b	q_1	a	a	<table border="1"><tr><td>b</td><td>q_1</td><td>b</td></tr><tr><td>q_2</td><td>b</td><td>q_2</td></tr></table>	b	q_1	b	q_2	b	q_2
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 מהוות חלון חוקי. ז"א

$$\phi_{\text{move}} = \bigwedge_{\substack{1 \leq i \leq n^k \\ 1 \leq j \leq n^k}} (\text{חלון } i, j \text{ חוקי}) \quad (11.5)$$

אנטו מציירים בטקסט "חלון ה- i, j חוקי" את הנוסחה הבאה, כאשר a_6, \dots, a_1 מסמנים את התכנים של ה-6 תאים של כל חלון:

$$\bigvee_{\substack{\{a_1, a_2, a_3, a_4, a_5, a_6\} \\ \text{חלון חוקי}}} (x_{i,j-1,a_1} \wedge x_{i,j,a_2} \wedge x_{i,j+1,a_3} \wedge x_{i+1,j-1,a_4} \wedge x_{i+1,j,a_5} \wedge x_{i+1,j+1,a_6}) \quad (11.6)$$

עד כה הוכחנו שקיים רדוקציה מכל שפה $.SAT \rightarrow A \in NP$ כעת נוכיח כי הרדוקציה זו חישובית בזמן-פולינומייאלי.

הטבלה של N היא מסדר $n^k \times n$ ולכן היא מכילה n^{2k} תאים.

נחשב את הסיבוכיות של כל הנוסחים $\phi_{\text{move}}, \phi_{\text{acc}}, \phi_{\text{start}}, \phi_{\text{cell}}$.

• הנוסחה ϕ_{cell}

הנוסחה (11.2) של ϕ מכילה n^{2k} נוסחאות עם 3 ליטרלים. לכן

$$\phi_{\text{cell}} = O(n^{2k}) .$$

• הנוסחה ϕ_{start}

הנוסחה (11.3) של ϕ מכילה בדיק n^k ליטרלים. לכן

$$\phi_{\text{start}} = O(n^k) .$$

• הנוסחה ϕ_{acc}

הנוסחה (11.4) של ϕ מכילה בדיק n^k ליטרלים. לכן

$$\phi_{\text{acc}} = O(n^k) .$$

• הנוסחה ϕ_{move}

הנוסחה (11.6,11.5) של ϕ מכילה n^{2k} נוסחאות עם 6 ליטרלים. לכן

$$\phi_{\text{move}} = O(n^{2k}) .$$

לכן בסה"כ

$$\phi = O(n^{2k}) + O(n^k) + O(n^k) + O(n^{2k}) = O(n^{2k}) .$$

לפיכך קיימת רדוקציה חישובית בזמן פולינומייאלי מכל שפה L - SAT $\in NP$



שיעור 12

רذוקציות פולינומיאליות

CLIQUE 12.1 - NP היא - שלמה

משפט 12.1 $CLIQUE \in NPC$

הבעית CLIQUE היא (ראו הגדרה 10.8):

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

היא NP - שלמה CLIQUE

הוכחה:

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

2) נוכח כי $NP \leq_P CLIQUE$ קשה ע"י רذוקציה.

פונקציית הרזוקציה

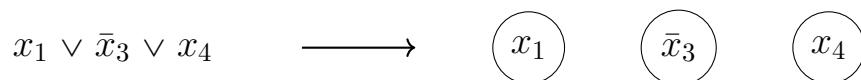
בhinintן נוסחת ϕ 3CNF מעל n משתנים x_1, x_2, \dots, x_n המכילה m פסוקיות, ניצור זוג $\langle G, k \rangle$ ונוכח כי

$$\langle \phi \rangle \in 3SAT \Leftrightarrow \langle G, k \rangle \in CLIQUE.$$

בנייה את הגרף G באופן הבא:

הקודקודים של G :

לכל פסוקית C_i ב- ϕ המכילה 3 ליטרלים ניצור שלשה t_i המכילה 3 קודקודים המתאימים להחטறלים של C_i :



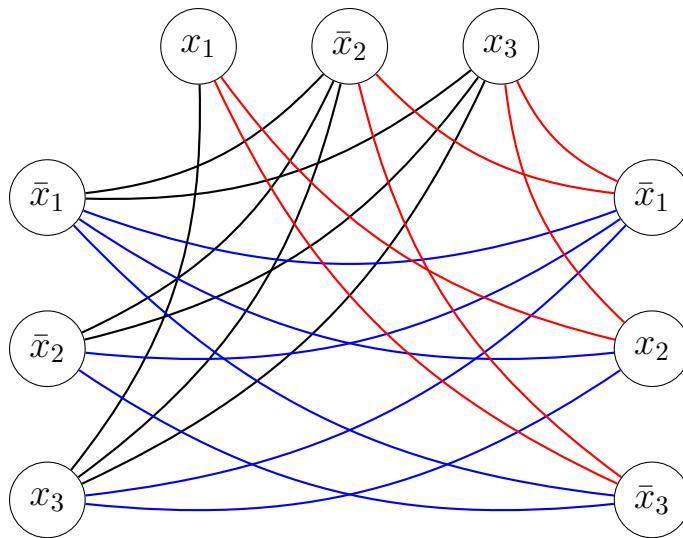
הצלעות של G :

נבחר בין כל שני קודקודים פרט לזוגות הבאים:

- זוג קודקודים המתאימים למשתנה ומשלים שלו.
- זוג קודקודים שנמצאים באותו שלושה.

לדוגמא:

$$\phi = \left(\begin{array}{c} \textcolor{red}{T} \\ x_1 \vee \bar{x}_2 \vee x_3 \\ C_1 \end{array} \right) \wedge \left(\begin{array}{c} \textcolor{red}{T} \\ \bar{x}_1 \vee \bar{x}_2 \vee x_3 \\ C_2 \end{array} \right) \wedge \left(\begin{array}{c} \textcolor{red}{T} \\ \bar{x}_1 \vee x_2 \vee \bar{x}_3 \\ C_3 \end{array} \right)$$



נקבע $k = m$.

נכונות הרדוקציה

1) ניתן לבנות את G בזמן פולינומיائي בגודל ϕ .

2) נוכיח כי

$$\langle \phi \rangle \in 3SAT \Leftrightarrow \langle G, k \rangle \in CLIQUE .$$

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 לשינוי 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 בדיזוק קודקוד אחד מכל שלשה t_i . ניתן השמה למשתנים של ϕ כך שהליטרל שמתאים לקודקוד שנמצא בклיקה יקבל ערך T .
- השמה זו אפשרית מכיוון שהקליקה לא מכילה שני קודקודים המתאימים לשינוי ומשלים שלו.

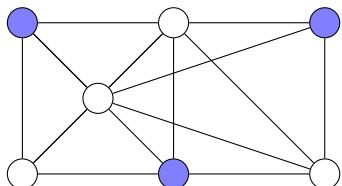
- בנוסף לשם זו מספקת את ϕ מכיוון שהקליקה מכילה קודקוד מכל שלשה t_i ולכן הliterל המתאים לקודקוד בשלה t_i קיבל ערך T ולכן הוא מספק את הפסוקית C_i .
- לכן ϕ ספיקה.

■

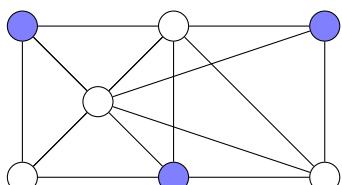
12.2 בעיית הקבוצה הבלתי תלויות

הגדרה 12.1 קבוצה בלתי תלויות

בاهינתן גרף לא מכוון $G = (V, E)$, קבוצה בלתי תלויות ב- G היא תת-קבוצה של קודקודיים $S \subseteq V$ כך שלכל שני קודקודיים S $u, v \in S$ מתקיים $(u, v) \notin E$.



קבוצת בלתי תלויות בגודל 3: $k = 3$



קבוצת בלתי תלויות בגודל 3: $k = 3$

הגדרה 12.2 בעיית IS

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

פלט: האם קיימת קבוצה בלתי תלויות ב- G בגודל k ?

$$IS = \{ \langle G, k \rangle \mid G \text{ גרף לא מכוון המכיל קבוצה בלתי תלויות בגודל } k \}$$

משפט 12.2 $IS \in NPC$

הבעיה IS היא NP - שלמה.

הוכחה:

(1) נומינצי $IS \in NP$

בנייה אלגוריתם אimoto V עבור IS .
 $V = \text{על קלט } (\langle G, k \rangle, y)$:

- בודק האם y היא קבוצה של k קודקודיים מ- G השוניים זה מזה.
 - אם לא \Rightarrow דוחה.
- בודק האם כל שני קודקודיים מ- y לא מחוברים בצלע ב- G .
 - אם כן \Rightarrow מקבל.

- אם לא \Rightarrow דוחה.

(2) נוכחות CIQUE \leq_P IS

פונקציית הרדוקציה:

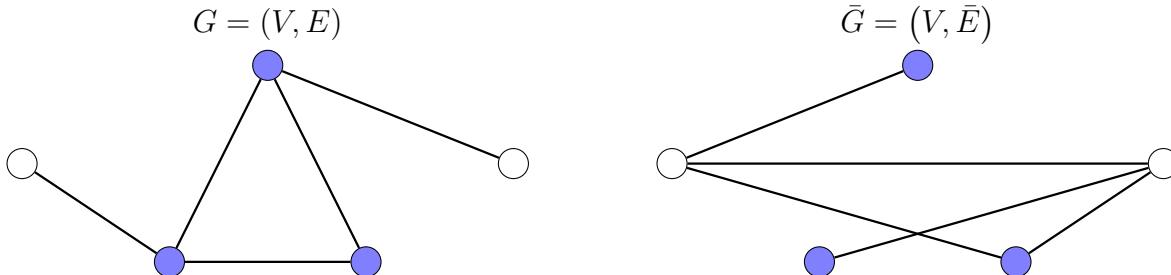
בhinתן זוג $\langle G, k \rangle$ הקלט של CLIQUE, ניצור זוג $\langle G', k' \rangle$, הקלט של IS, ונוכח כי:

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

הfonקציית הרדוקציה מוגדרת כך שהתנאים הבאים מתקייםים:

- 1) נניח שהגרף הוא $G = (V, E)$. אז הגרף G' הוא הגרף המשלים של $G = (V, E)$ כאשר $G' = \bar{G} = (V, \bar{E})$ ו $\bar{E} = \{(u_1, u_2) \mid (u_1, u_2) \notin E\}$.
- 2) $k' = k$

לדוגמה, בהינתן הגרף $G = (V, E)$ שמכיל קliquה בגודל $3, k = 3$, הפונקציית הרדוקציה R מוחירה את הגרף $\bar{G} = (V, \bar{E})$ ואת המספר $3, k' = k = 3$, כמתואר בתרשימים למטה:



nocחות הרדוקציה

- 1) ניתן לבנות G' בזמן פולינומיائي בגודל G .
- 2) נוכחה כי $\langle G, k \rangle \in CLIQUE \Leftrightarrow \langle G', k' \rangle \in IS$.

כיוון

- בhinתן גרף $G = (V, E)$ ושלם נניח כי $\langle G, k \rangle \in CLIQUE$.
- אם G מכיל קliquה S בגודל k , אז $(u_1, u_2) \in E$ (אם $u_1, u_2 \in S$ שני קודוקדים בקliquה S או $u_1, u_2 \in S$ כלומר, כל שני קודוקדים ב- S מחוברים בצלע של G).
- אם $(u_1, u_2) \notin \bar{E}$ (אם $u_1, u_2 \in S$ אז שני קודוקדים ב- S לא מחוברים בצלע של הגרף \bar{G} , זהינו G').

\Leftarrow אותה הקבוצה S היא קבוצה בלתי תלوية ב- G' בגודל $k' = k$.

\Leftarrow G' מכיל קבוצה בלתי תלوية בגודל k' .

$\langle G', k' \rangle \in IS \Leftarrow$

כיוון \Rightarrow

בhinתן גרף G' ושלם k' .

$\langle G', k' \rangle \in IS$.

$\Leftarrow G'$ מכיל קבוצה בלתי תלوية S בגודל k' .

\Leftarrow אם $u_1, u_2 \in S$ איזי $(u_1, u_2) \notin \bar{E}$.

כלומר, כל שני קדוקודים ב- S לא מחוברים בצלע של G' .

\Leftarrow אם $u_1, u_2 \in S$ איזי $(u_1, u_2) \in E$.

כלומר, כל שני קדוקודים 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 .

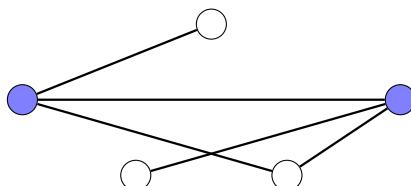
$\Leftarrow \langle G, k \rangle \in CLIQUE$.

■

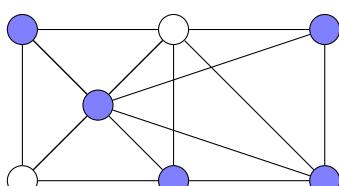
12.3 בעית הcisוי בקדוקודים

הגדרה 12.3 cisוי בקדוקודים

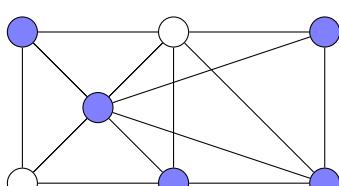
בhinתן גרף לא מכון $G = (V, E)$, cisוי בקדוקודים ב- G הוא תת-קבוצה של קודוקודים $C \subseteq V$ כך שלכל צלע $S \in C$, $u \in C$ וותקיים $v \in S$ $u \neq v$.



cisוי בקדוקודים בגודל 2: $k = 2$



cisוי בקדוקודים בגודל 5: $k = 5$



cisוי בקדוקודים בגודל 5: $k = 5$

12.4 הבעיה VC

הגדירה 12.4 בעיית VC

קלט: גראף לא מכוון $G = (V, E)$ ומספר k .

פלט: האם קיימים כיסוי בקודקודים ב- G בגודל k ?

$$VC = \{\langle G, k \rangle \mid \text{גראף לא מכוון המכיל כיסוי בקודקודים בגודל } k\}$$

משפט 12.3 $VC \in NPC$

הבעיה VC היא NP - שלמה.

הוכחה:

$$\underline{VC \in NP}$$

בנייה אלגוריתם אימות VC עבור $.VC$ על קלט $(\langle G, k \rangle, y) = V$:

- בודק האם כל צלע ב- G מכילה לפחות קצה אחד ב- y .

○ אם כן \Rightarrow מקבל.

○ אם לא \Rightarrow דוחה.

$$\underline{\text{נוכיח כי } VC \text{ היא } NP\text{-קשה ע"י רדוקציה}}$$

פונקציית הרדוקציה:

בהתנחת זוג $\langle G, k \rangle$ הקלט של IS , נוצר זוג $\langle G', k' \rangle$ הקלט של VC ונוכיח כי

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

הפונקציית הרדוקציה מוגדרת כך שהתנאים הבאים מתקיימים:

$$\begin{aligned} 1) \text{ נניח שהגרף הוא } &G = (V, E) \\ \text{ אז הגרף } G' \text{ הוא אותו גראף } &G = (V, E) \end{aligned}$$

$$.k' = |V| - k \quad (2)$$

נכונות הרדוקציה

1) ניתן לבנות G' בזמן פולינומיאלי בגודל G .

$$2) \text{ נוכיח כי } \langle G, k \rangle \in IS \Leftrightarrow \langle G', k' \rangle \in VC .$$

כיוון

בהתנחת גראף $G = (V, E)$ ושלם k .

$$\text{נניח כי } \langle G, k \rangle \in IS .$$

$\Leftarrow G$ מכיל קבוצה בלתי תלויות S בגודל k .
 \Leftarrow אם $u_1 \in S$ וגם $u_2 \in S$ אז $(u_1, u_2) \notin E$.
 כלומר, כל שני קדוקודים ב- S לא מחוברים בצלע ב- G .

\Leftarrow השלילה הלוגית של הגרירה הזאת היא:
 אם $u_2 \notin S$ אז $u_1 \notin S$ או $(u_1, u_2) \in E$.
 \Leftarrow אם $u_2 \in V \setminus S$ אז $u_1 \in V \setminus S$ או $(u_1, u_2) \in E$.
 \Leftarrow $k' = |V| - k$ היא כיסוי קדוקודים ב- G בגודל k' .
 \Leftarrow הגרף $G' = G$ מכיל כיסוי קדוקודים בגודל k' .
 $\Leftarrow \langle G', k' \rangle \in VC$ \Leftarrow

כיוון \Rightarrow

בהינתן גרף G' ושלם k' .
 $\langle G', k' \rangle \in VC$.

$\Leftarrow G''$ מכיל כיסוי בקדוקודים C בגודל k'' .
 \Leftarrow אם $u_2 \in C$ אז $u_1 \in C$ או $(u_1, u_2) \in E$.
 \Leftarrow השלילה הלוגית של הגרירה הזאת היא:
 אם $(u_1, u_2) \notin E$ אז $u_2 \notin C$ ו $u_1 \notin C$.
 \Leftarrow אם $(u_1, u_2) \notin E$ וגם $u_2 \in V \setminus C$ אז $u_1 \in V \setminus C$.
 \Leftarrow כל שני קדוקודים ב- $V \setminus C$ לא מחוברים בצלע ב- G'' .
 \Leftarrow $k = |V| - k'$ היא קבוצת בלתי תלויות ב- G'' בגודל k .
 \Leftarrow הגרף $G = G'$ מכיל קבוצה בלתי תלויות בגודל k .



PARTITION 12.5

הגדרה 12.5 בעית PARTITION

קלט: קבוצת מספרים שלמים $.S = \{x_1, x_2, \dots, x_n\}$
פלט: האם קיימת תת-קבוצה $Y \subseteq S$ כך ש-

$$\sum_{y \in Y} y = \sum_{y \in S \setminus Y} y$$

$$PARTITION = \left\{ S \mid \sum_{y \in Y} y = \sum_{y \in S \setminus Y} y \text{ קבוצת שלמים, וקיימת תת-קבוצה } Y \subseteq S \text{ כך ש-} \right\}$$

12.6 רדוקציות פולינומיאליות

משפט 12.4 רדוקציות פולינומיאליות

$$\begin{aligned}
 SAT &\leqslant_P 3SAT \\
 3SAT &\leqslant_P CLIQUE \\
 CLIQUE &\leqslant_P IS \\
 IS &\leqslant_P VC \\
 SubSetSum &\leqslant_P PARTITION \\
 HAMPATH &\leqslant_P HAMCYCLE
 \end{aligned}$$

משפט 12.7 שפות NP שלמות**משפט 12.5 שפות NP- שלמות**

(משפט קוק לוין) SAT - שלמה. $3SAT$ - שלמה. $HAMPATH$ - שלמה. $CLIQUE$ - שלמה. IS - שלמה. VC - שלמה.

שיעור 13

סיבוכיות מקום ושלמות ב PSPACE

1.3.1 הגדרה של סיבוכיות מקום

המשמעות המרכזי שדברנו עליו בקורס היה זמן החישוב שמדדנו באמצעות מספר הצעדים שמבצעת מכונת טיורינג. מدد חשוב נוסף הוא **כמota הziיכרו** שבה החישוב משתמש. למעשה, המודל של מכונת טיורינג משתמש ב"אינסופי" זיכרון, מכיוון שהסרט שעליו כותבת המכונה הוא אינסופי. לצורך לנו הגדרנו מכונת טיורינג כך שבסרט יש רצף אינסופי של תווים רוח מחד שמאלית ומצד ימין של הקלט. אבל בפועל ניתן לדרוש שהמכונה לא תגעה לתאים שאינם שמימיין לתא ספציפי כלשהו, שמהווה את חסם הזיכרון של המכונה.

הגדרה 13.1 סיבוכיות מקום של מכונת טיורינג

הסיבוכיות מקום של מכונת טיורינג M על קלט w היא פונקציה $f(|w|)$ שבהם M עושה שימוש בחישוב על w .

אם הסיבוכיות מקום של מכונה M היא $f(n)$ אז אומרם כי M רצה מקום (n) .

הגדרה 13.2 המחלוקת $SPACE(f(n))$

אומרים כי שפה L שיכת למחלוקת $SPACE(f(n))$ אם קיימת מכונת טיורינג דטרמיניסטיבית שמכריעה את L במקום $O(f(n))$. פורמלי:

$$SPACE(f(n)) = \{L \mid O(f(n)) \text{ מקום } L \text{ שמכריעה } M\}.$$

דוגמה 13.1

נראה כי ניתן לפתור את הבעיה SAT ע"י אלגוריתם שרצ' במקום לינארי. כולל:

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

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

$$M = \langle \phi \rangle \text{ על כל קלט } \langle \phi \rangle :$$

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

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

א) M רושמת את מחרוזת של ההשמה $a_1 a_2 \dots a_m$ על סרט העבודה.

ב) M מחשבת את הערך של ϕ עבור ההשמה הנוכחית a_1, a_2, \dots, a_m ע"י סריקה של הקלט $\langle \phi \rangle$ רשום על סרט הקלט.

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

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

M רצה במקום ליניארי. בפרט:

- M שומרת על סרט העבודה את ההשמה $a_m \dots a_1$ זה נדרש $O(m)$ תאים.
- המספר המשתנים, m הוא n לכל היותר.
- לכן M רצה במקום $O(n)$.

לפיכך:

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

הגדלה 13.3 המחלוקת $NSPACE(f(n))$

אומרים כי שפה L שיכת למחלוקת $NSPACE(f(n))$ אם קיימת מכונת טירינג א-דטרמיניסטית שמכריעה את L במקום $O(f(n))$. פורמלי:

$$NSPACE(f(n)) = \{L \mid O(f(n)) \text{ שמכריעה } L \text{ במקום} \}.$$

דוגמה 13.2

תהי השפה הבאה:

$$ALL_{NFA} = \{\langle A \rangle \mid L(A) = \Sigma^* \text{ עבورو NFA } A \text{ הוא} \}.$$

הוכיחו כי $ALL_{NFA} \in NSPACE(n)$

פתרון:

הפתרון מtbוסס על זה שהוא פשוט יותר לבנות אלגוריתם המכريع את השפה המשלימה:

$$\overline{ALL_{NFA}} = \{\langle A \rangle \mid \text{קיימת } \Sigma^* \text{ עבורה } A \text{ דוחה } w\}.$$

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

משפט 13.1

אם $M = (Q, \Sigma, \delta, q_0, F)$ הוא NFA וקיים מילה w שנדחה ע"י M , או האורך המילה $|w| \leq 2^q$, כאשר q הוא המספר המצביע של M , וקיימים אינסוף מילים שנדחות ע"י $M = |Q|$

כעת נגדיר סימון חדש שיעזור לנו לבנות את האלגוריתם. נניח ש- $M = (Q, \Sigma, \delta, q_0, F)$ היא מכונת NFA כלשהי. תהי $P(Q)$ הקבוצת החזקה של Q . עבור כל NFA הפונקציית המעברים היא מהצורה

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

בhinתן מילה $a_n \dots a_1$ אשר $\Sigma = a_i \in \Sigma$ הוא התו ה- i של המילה, $1 \leq i \leq n$. נגדיר את הסדרה הבאה:

$$S_0, S_1, S_2, \dots, S_n$$

כך אשר

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

כך אשר $S_i \in P(Q)$ לכל $0 \leq i \leq n$

בנייה האלגוריתם

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

$x \in N$ = על כל קלט:

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

2) ייְהֵי $|Q| = q$ מספר המ מצבים של M . נגדיר $S_0 = \{q_0\}$.

- 3) N מבצעת את הלולאה הבאה:
 $i = 0 \leq i \leq 2^q - 1$

a) בוחרת באופן איזדרמיניסטי TWO קלט $a_i \in \Sigma$.

b) מחשבת

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

c) אם $S_{i+1} \cap F \neq \emptyset$ תדחה.

בפועל N בודקת את התנאי הזה ע"י לסמן את כל המ מצבים שב- S_i . אם אחד מה מצבים המסומנים הוא מצב קבלה אז N תדחה.

- 4) אם בסיום הלולאה לא הייתה מצב-קבלה באף אחת מן הקבוצות S_i אז N מקבל.

$x \in \overline{ALL_{NFA}}$ אם

$\langle A \rangle = x$, כאשר A היא מכונת NFA . וקיים מילה $w \in \Sigma^*$ כך ש- A תדחה.
 \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 מקבל.

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

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

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

\Leftarrow לכל מילה $w \in \Sigma^*$, קיים שלב שבו A נמצא במצב קבלה.
 \Leftarrow בכל ריצה של N , קיימת איטרציה i עבורה $S_i \cap F \neq \emptyset$
 \Leftarrow באיטרציה זו N תדחה.
 \Leftarrow בכל ריצה N תדחה.
 \Leftarrow דוחה את x .

סיבוכיות מקומ

- נסמן ב- $n = |\langle M \rangle|$ את אורך הקלט, וב- $q = |Q|$ את מספר המ מצבים 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 הנווכה $Q \subseteq S_i$ של מצבים אפשריים.
 - * פעולה N שומרת S_i בוקטור ביטים באורך q לכל היוטר.
 - * מונה של האטרציות הלולאה עד 2^q , המאוחסן ביצוג בינארי ודורש $O(q)$ ביטים.
 - *תו קלט אחד הנבחר באופן אי-דטרמיניסטי בכל אטרציה, ומשתני עזר לחישוב S_{i+1} , הדורשים מקום קבוע או בינארי ב- q .

לפיכך סיבוכיות המקום הכלול של N היא

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

לפיכך האלגוריתם N פועל במקום לינארי.

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

13.2 משפט סביז'

הגדרה 13.4 CANFIELD

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

$$\langle N, c_1, c_2, t \rangle = \text{על קלט } CANFIELD$$

(1) רושם את c_2 ו- t על מחסנית.

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

• אם לא אז הוא דוחה.

(3) אם $t = 1$:

• אם 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 הריצה של N על w אשר משתמשת במקום

(כאשר w היא המילה הנקרה של הקונפיגורציה (c_k)):

$$\text{CANFIELD} \left(N, c_1, c_k, \left\lfloor \frac{t}{2} \right\rfloor \right) \text{ מרץ} \quad (5)$$

כאשר $\left\lfloor \frac{t}{2} \right\rfloor$ הוא השלים הכי הקרוב ל- $\frac{t}{2}$ וקטן מ- $\frac{t}{2}$ או שווה ל- $\frac{t}{2}$.

$$\text{CANFIELD} \left(N, c_k, c_2, \left\lceil \frac{t}{2} \right\rceil \right) \text{ מרץ} \quad (6)$$

כאשר $\left\lceil \frac{t}{2} \right\rceil$ הוא השלים הכי הקרוב ל- $\frac{t}{2}$ וגדול מ- $\frac{t}{2}$ או שווה ל- $\frac{t}{2}$.

- (7) אם שתי הרצות בשלבי 4) ו- 5) הסתיימו בקבלה \Leftarrow מקבל.
 (8) אחרת אם לא התקבלה תשובה קבלה \Leftarrow דוחה.

משפט 13.2 משפט סבץ'

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

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

הוכחה:

הריעון של ההוכחה:

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

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

בנייה המוכנה:

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

- אם ניתן לעبور מ- c_1 ל- c_2 בכל היתר t צעדים $\Leftarrow CANYIELD(N, c_1, c_2, t)$ מקבל.
- אחרת $CANYIELD(N, c_1, c_2, t) \Leftarrow$ דוחה.

נגידר מכונת טיורינג דטרמיניסטית M שמסמלצת את המוכנה האי-דטרמיניסטית N באופן הבא.

ראשית נסמן ב- n את אורך הקלט w של N .

תהי c_0 הקונפיגורציה ההתחלתית.

נתנו את N כך שלאחר כל ריצה הראש חוזר לקצה השמאלי של תוכן הסרט ו- N עוברת לkonfiguracji c_{acc} .

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

המוכנה טיורינג הדטרמיניסטי M תוגדר כך:

על קלט w :

1) מרים $CANYIELD(N, c_0, c_{acc}, 2^{df(n)})$ על הקלט w ועונה כמוות.

הוכחת הנכונות:

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

\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 $\langle N, c_0, c_{\text{acc}}, 2^{df(n)} \rangle$ על הקלט $CANYIELD$ קיבל.
 $M \Leftarrow$ קיבל w .
 $N \in NSPACE(f(n))$ ו- $w \notin L(N)$
 \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 $\langle N, c_0, c_{\text{acc}}, 2^{df(n)} \rangle$ על הקלט $CANYIELD$ ידחה.
 $M \Leftarrow$ ידחה w .

סיבוכיות מקומ:

- כל פעם ש- $CANYIELD$ מפעיל את עצמו באופן רקורסיבי, הוא רושם את c_1, c_2 ו- t על מחסנית, כך שניתן יהיה לשזר אותו לאחר הקריאה הרקורסיבית הבאה.
- בכלל ש- $(N \in NSPACE(f(n)))$ איזי 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 $2^{df(n)}$ שכן העומק של הרקורסיה הוא $O(\log_2(2^{df(n)})) = O(f(n))$.
- לכן המקום הכלול ש- M דורש הוא $O(f^2(n))$.

לפיכך

$$M \in SPACE(f^2(n)) .$$

לסיכום: הוכחנו שהינתן מכונת אי-דטרמיניסטיבית N כלשהי שמכריעה שפה A כלשהי עברית

$$N \in NSPACE(f(n)) ,$$

קייםת מכונת טיריניג דטרמיניסטיבית M שמכריעה A במקום $O(f^2(n))$, כלומר:

$$M \in SPACE(f^2(n)) .$$

לפיכך:

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



13.3 המחלוקת PSPACE

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

הגדרה 13.5 אלגוריתם מקומ פולינומיAli

אומרים כי אלגוריתם A מカリע בעיה במקום פולינומיAli אם קיים קבוע $c > 0$ כך שהזמן הריצה של A על קלט w חסום ע"י $O(|w|^c)$.

התזה של צרץ' טירונג אומר שאם קיים אלגוריתם המカリע בעיה במקום פולינומיAli, אז קיימת מכונת טירונג דטרמיניסטיבית המכreira את השפה השקולה לבעיה זו במקום פולינומיAli.

. אלגוריתם מカリע \equiv מכונת טירונג דטרמיניסטיבית .

הגדרה 13.6 המחלוקת $PSPACE$

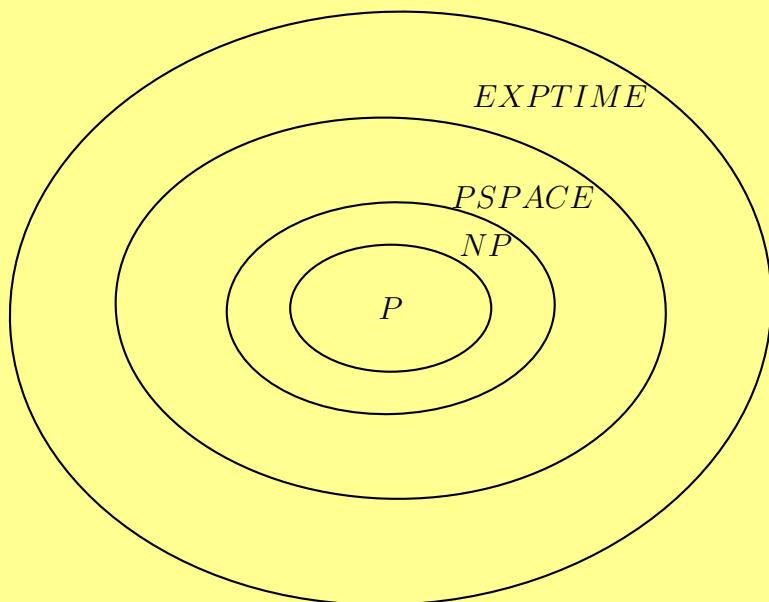
אומרים כי שפה L שיכת ל- $PSPACE$ אם קיימת מכונת טירונג דטרמיניסטיבית שמכreira אותה במקום פולינומיAli.

הגדרה 13.7 המחלוקת $NPSPACE$

אומרים כי שפה L שיכת ל- $NPSPACE$ אם קיימת מכונת טירונג אי-דטרמיניסטיבית שמכreira אותה במקום פולינומיAli.

משפט 13.3

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

**13.4 שלמות ב- $PSPACE$** **הגדרה 13.8 $PSPACE$ קשה**

בעיה B נקראת $PSPACE$ קשה אם לכל בעיה $A \in PSPACE$ קיימת רדוקציה בזמן פולינומיAli מ- A ל- B . כלומר: $A \leq_P B$.

הגדירה 13.9 שלמות $PSPACE$

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

$$B \in PSPACE \quad (1)$$

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

13.5 נוסחאות בولיאניות עם כמתים

אפשר לחשב על בעיות NP בתור סוג של "משחק" לשחקן יחיד: בהינתן שפה $L \in NP$ קיים לה יחס R_L ואפשר לחשב ה"משחק" שבו בהינתן x מטרת השחקן היא למצוא y כך ש- $y \in R_L(x)$. למשל, בהינתן לוח סודוקו x , מטרת השחקן היא למצוא מילוי y של הלוח. דוגמה נוספת, בהינתן אוסף x של צורות, מטרת השחקן היא למצוא דרך y שסדר אותו כך שירכיבו ריבוע. עוד דוגמה היא, בהינתן קוביה הונגרית במצב x , מטרת השחקן היא למצוא סדרת מלחכים y שתחזיר אותה במצב מסוים, וכן הלאה. כל המשחקים הללו מאופיינים בכך שהם משחקים לשחקן יחיד ועם ידע מלא (אין אלמנט של אקרואיות, מרגע המשחק קיבל את הקלט).

בעיה ה- $-NP$ - שלמה המייצגת ביותר היא SAT . אפשר לחשב על SAT בתור בעיה של בדיקת מחרוזת ביצוג בינארי: השמה היא מילה $w \in \{0, 1\}^n$ ופסוק ה- ϕ בודק האם המילה w עומה על קריטריון מסוים ϕ "מקודדת". כפי שראינו, לכל המשחקים שלעיל אנחנו יכולים לבנות נוסחה ϕ שתקודד את המשחק עצמו, ותבודוק האם w היא פתרון של המשחק.

אפשר להכליל את הרעיוןות הללו למשחקים בשני שחקנים. המחלוקת המתתקבלת היא $PSPACE$ והשפה המקבילה לה- SAT בחלוקת זו הוא $TQBF$.

בפרק 11 ו- 12 הגדרנו נוסחה בוליאנית כביטוי מתמטי שבוני מהמרכיבים הבאים:

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

ונג	\wedge
או	\vee
לא	\neg

אפשר להכליל את ההגדירה זו לנוסחה בוליאנית **עם כמתים**. כמתים הם סימנים המוצמדים לפסוק ומוסיפים לו משמעות של כלל או של **קיים**. בהקשר שלנו "פסוק" יכול משתנים, קשרים (אופרטורים בוליאניים עיקריים), **קבועים** (0 ו- 1) וכמתים, כאשר כמתים מתווסףיס בצורה הבאה:

הגדירה 13.10 כמתים

אם ϕ הוא פסוק, אז $\phi x \forall$ (קרי "לכל x ϕ ") הוא פסוק.

אם ϕ הוא פסוק, אז $\phi x \exists$ (קרי "קיים x כך ש- ϕ ") הוא פסוק.

הסימנים \exists , \forall נקראים **כמתים** ונשתמש בסימון Q כדי לציין כמה מבין \forall , \exists באופן כללי.

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

• אמרים שבפסוק $Qx\phi$, כל מופיע של x בתחום ϕ נופל תחת הכמת Q .

• משתנה שנופל תחת כמת כלשהו נקרא **משתנה הקשור** ומשתנה שלא נופל תחת אף כמת נקרא **משתנה**

חופשי.

דוגמה 13.3 דוגמאות של נוסחאות בוליאניות עם כמתים

בדוגמאות הבאות y, x הם משתנים בוליאניים. קלומר $\{0, 1\}$.

$$\phi = \forall x \exists y [(x \vee y) \wedge (\bar{x} \vee \bar{y})] . \quad (1)$$

בדוגמה זו $\phi = T$.

$$\phi = \forall x [x \vee \bar{x}] \rightarrow \phi = T . \quad (2)$$

$$\phi = \exists x (x \wedge \bar{x}) \rightarrow \phi = F . \quad (3)$$

אם בפסוק אין משתנים חופשיים, ערך האמת שלו נקבע באופן חד משמעי, בצורה הבאה:

הגדרה 13.11 ערך אמת של פסוק שלא כולל משתנים חופשיים

ערך האמת של פסוק שכולל רק קבועים וקשרים נקבע בהתאם לטבלאות האמת של הקשרים. נסמן ב- (x) פסוק עם המשתנה חופשי x ונסמן ב- (a) את מה שמתකבל מהפסוק כמשמעותם כל מופיע של x בקבוע $a \in \{0, 1\}$. איז:

- ערך האמת של (x) הוא T אם ורק אם ערך האמת של (0) ϕ **ו גם** של (1) ϕ הוא T .
- ערך האמת של $(\exists x)$ הוא T אם ורק אם ערך האמת של (0) ϕ או של (1) ϕ הוא T .

הגדרה 13.12 QBF

אומרים שפסוק הוא *QBF* (Quantified Boolean Formula) אם הוא מהצורה

$$Qx_1 Qx_2 \dots Qx_n \phi(x_1, x_2, \dots, x_n)$$

כך ש- ϕ הוא פסוק המשתנים היחידים שלו הם x_1, \dots, x_n . לפסוק זה יש ערך אמת מוגדר: או T או F .

הגדרה 13.13 TQBF

$\langle \phi \rangle$ השפה *TQBF* כוללת את כל פסוקי ה- *QBF* שערך האמת שלהם הוא T .

$$TQBF = \{\langle \phi \rangle \mid \phi = T \text{ QBF } \phi\} .$$

הערה 13.1

נשים לב שאפשר לישוב על *SAT* בתור מעין מכירה פרטיה של *TQBF* שבו כל ה- $-Q$ - אם הם \exists , ואם קיימות לפסוק השמה מספקת היא למעשה השאלה האם הפסוק עם כמתים הוא T .

משפט 13.4

$$SAT \subseteq TQBF .$$

הוכחה: תרגיל בית.

משפט 13.5

השפה

$$TQBF \in PSPACE .$$

הוכחה:

נראה אלגוריתם רקורסיבי שבלם קרייה רקורסיבית מורייד את אחד מהכמתים של הנוסחה. תנאי העצירה הוא נוסחה ϕ ללא כמתים. בנוסחה זו אין משתנים כלל (כי ב- QBF אין משתנים חופשיים) אלא רק קבועים, וחישוב ערך האמת של הפסוק הוא פולינומייאלי בגודל הפסוק.

בנייה האלגוריתם

$$:QBF \text{ על קלט } \psi \text{ כאשר } \psi = A$$

$$\psi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \phi(x_1, \dots, x_n)$$

A מבצע כך:

1) מקרה הבסיס: אם $n = 0$ אז ψ הוא קבוע.

- אם $\psi = 1$ מקבל.
- אם $\psi = 0$ דוחה.

2) מקרה הרקורסיבי: אם $n > 0$ אז A מבצע את הרקורסיה הבאה.

• אם $\exists : Q_1 =$

- מרים A על $. \psi' = Q_2 x_2 \dots Q_n x_n \phi(0, x_2, \dots, x_n)$
- אם התקבל שערך האמת של ψ' הוא T אז מקבל.
- אחרת מרים A על $Q_2 x_2 \dots Q_n x_n \phi(1, x_2, \dots, x_n)$ ועונה כמוות.

• אם $\forall : Q_1 =$

- מרים A על $. \psi' = Q_2 x_2 \dots Q_n x_n \phi(0, x_2, \dots, x_n)$
- אם התקבל שערך האמת של ψ' הוא F אז דוחה.
- אחרת מרים A על $Q_2 x_2 \dots Q_n x_n \phi(1, x_2, \dots, x_n)$ ועונה כמוות.

הוכחת נכונות

ניתן להוכיח הנכונות של A ע"י אינדוקציה.

שלב הבסיס: $n = 0$.

אם $\psi = 1 \Leftarrow$ בשלב (1) A מקבל.

אם $\psi = 0 \Leftarrow$ בשלב (1) A דוחה.

שלב המעבר: $n = k$

נניח ש- A מכיר QBF באורך k : $\psi = Q_1x_1Q_2x_2 \dots Q_kx_k\phi(x_1, \dots, x_n)$. זהה ההנחה האינדוקציה שלנו. נוכיח כי A גם מכיר QBF באורך $k+1$ באופן הבא:

אם $\Leftarrow Q_1x_1Q_2x_2 \dots Q_kx_kQ_{k+1}x_{k+1}\phi(x_1, \dots, x_k, x_{k+1}) = T$

מקרה 1: $Q_1 = \exists$

$Q_2x_2 \dots Q_kx_kQ_{k+1}x_{k+1}\phi(0, \dots, x_k, x_{k+1}) = T \Leftarrow$

או $Q_2x_2 \dots Q_kx_kQ_{k+1}x_{k+1}\phi(1, \dots, x_k, x_{k+1}) = T$

\Leftarrow מכיוון ש- A מכיר QBF באורך k אז מקבל.

מקרה 2: $Q_1 = \forall$

$Q_2x_2 \dots Q_kx_kQ_{k+1}x_{k+1}\phi(0, \dots, x_k, x_{k+1}) = T \Leftarrow$

וגם $Q_2x_2 \dots Q_kx_kQ_{k+1}x_{k+1}\phi(1, \dots, x_k, x_{k+1}) = T$

\Leftarrow מכיוון ש- A מכיר QBF באורך k אז מקבל.

אם $\Leftarrow Q_1x_1Q_2x_2 \dots Q_kx_kQ_{k+1}x_{k+1}\phi(x_1, \dots, x_k, x_{k+1}) = F$

מקרה 1: $Q_1 = \exists$

$Q_2x_2 \dots Q_kx_kQ_{k+1}x_{k+1}\phi(0, \dots, x_k, x_{k+1}) = F \Leftarrow$

או $Q_2x_2 \dots Q_kx_kQ_{k+1}x_{k+1}\phi(1, \dots, x_k, x_{k+1}) = F$

\Leftarrow מכיוון ש- A מכיר QBF באורך k אז דוחה.

מקרה 2: $Q_1 = \forall$

$Q_2x_2 \dots Q_kx_kQ_{k+1}x_{k+1}\phi(0, \dots, x_k, x_{k+1}) = F \Leftarrow$

וגם $Q_2x_2 \dots Q_kx_kQ_{k+1}x_{k+1}\phi(1, \dots, x_k, x_{k+1}) = F$

\Leftarrow מכיוון ש- A מכיר QBF באורך k אז דוחה.

סיבוכיות מקומ של האלגוריתם

- עומק הרקורסיה בהרצה של האלגוריתם על פסוק עם n כמתים הוא n .
- כל רמה של הרקורסיה דורשת תא בודד של מקום למעט שלב תנאי העצירה שדורש מקום פולינומיAli.
- לכן האלגוריתם כולו דורש רק מקום פולינומיAli.



משפט 13.6 $TQBF$ היא $PSPACE$ שלמה.

השפה $TQBF$ היא NP שלמה.

הוכחה: נראה כי

$TQBF \in PSPACE$ (1)

2) לכל שפה $A \in PSPACE$ מתקיים $.A \leqslant_P TQBF$

הוכיחנו כי $TQBF \in PSPACE$ במשפט 13.5. נשאר להוכיח כי לכל שפה $A \in PSPACE$ מתקיים $.A \leqslant_p TQBF$.

בנייה הרדוקציה.

נראה כי לכל שפה $L \in PSPACE$ מתקיים

$$L \leqslant_P TQBF .$$

תהי L שפה השויכת ל- $PSPACE$ ותהי M מכונה טיורינג המכריעה את L במקומות פולינומילי (n). נוכיח שקיימת פונקציה f כך ש:

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

ז"א ψ כאשר ψ היא נוסחה בولיאנית עם כמתים כך ש:

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

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

$$\phi(c, c', t) \triangleq \begin{cases} 1 & \text{אם } t \text{ צעדים לכל היותר :} \\ 0 & \text{אחרת :} \end{cases} .$$

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

המקרה 1

לכל $t > 1$ ולכל קונפיגורציות c, c' :

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

כאן w מסמן קונפיגורציה ביןים בין הקונפיגורציה c לקונפיגורציה c' . היחס הרקורסיבי זהה של $\phi(c, c', t)$ אומר ש- M יכול לעבור מ- c ל- c' ב- t צעדים לכל היותר אם קיימת קונפיגורציה ביןים w כך ש- M יכול לעבור מ- c ל- w בכל היותר $\left\lfloor \frac{t}{2} \right\rfloor$ צעדים, ולאחר מכן M יכול לעבור מ- w ל- c' בכל היותר $\left\lfloor \frac{t}{2} \right\rfloor$ צעדים. הנוסחה $\phi(w, c', \left\lfloor \frac{t}{2} \right\rfloor)$ הינה נבנתה בשלב הבא של הרקורסיה באותו אופן. התתLIMIT הזה ממשיך עד שנגיע לנוסחה $\phi(x, y, 1)$ או $\phi(x, y, 0)$.

המקרה 2

$$\phi(c, c', t = 1) = \begin{cases} 1 & \text{בצעד חישוב בודד :} \\ 0 & \text{אחרת :} \end{cases}$$

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

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

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

$$x_{ijs} = \begin{cases} 1 & \text{если } s \in C \text{ записан в ячейке } ij \\ 0 & \text{если } s \in C \text{ не записан в ячейке } ij \end{cases}$$

למשל אם בתא $(2, 5)$ כתובה a אז הנוסחה $\phi(c, c', t = 1)$ מוגדרת:

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

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

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

$$\begin{aligned} & \text{Множество ячеек та для которых значение ячейки единично.} * \\ & \bigwedge_{\substack{s,t \in C \\ s \neq t}} (\overline{x_{i,j,s}} \vee \overline{x_{i,j,t}}) * \end{aligned}$$

לפיכך ϕ_{cell} מספקת אם ורק אם יש בדיק סימן אחד כתוב בכל תא. בנווגוד זה אם שני משתנים של אותו תא דולקים בগল שהסימן s וגם הסימן t כתובים בו זמנית בתא ה- ij אז $x_{i,j,s} = 1$ וגם $x_{i,j,t} = 1$ ו- ϕ_{cell} תיהה שווה ל- 0.

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

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

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 סהפוןקציית המעברים של M . במנוחה הטבלה, אפשר להגיד תזוזה חוקית בין השתי שורות על ידי תת-טבלה מסדר 3×2 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

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- $i + 2$ תקרה החלון- i . לכן:

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

המקרה 0אם $t = 0$ אז

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

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

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

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

הוכחת הנכונותאם $x \in L$ x מקבל $M \Leftarrow$ M יכולה לעבור מ- c_{acc} ל- c_{start} במספר צעדים פחות מ- או שווה ל- h $\psi = 1 \Leftarrow$ $\psi \in TQBF \Leftarrow$ אם $x \notin L$ x תדחה $M \Leftarrow$ לא קיימים צעדים של M מ- c_{start} ל- c_{acc} . $\psi = 0 \Leftarrow$ $\psi \notin TQBF \Leftarrow$ סיבוכיות זמן של הרדוקציה

בפונקציית הרדוקציה מחשבת $f(x) = \psi = \phi(c_{\text{start}}, c_{\text{acc}}, h)$, באופן רקורסיבי. לרקורסיה יש מספר שלבים השווה ל:

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

- הנוסחה (13.1) של ϕ_{cell} מכילה n^{2k} נוסחאות עם 3 ליטרלים. לכן הסיבוכיות זמן של ϕ_{cell} היא:

$$O(n^{2k}).$$

- הנוסחאות ϕ_c ו- $\phi_{c'}$ במשואה (13.2) מכילות בדיקות n^k ליטרלים. לכן הסיבוכיות זמן של ϕ_c ושל $\phi_{c'}$ הן: $O(n^k)$.

- הנוסחה (13.3) של ϕ_{move} מכילה n^{2k} נוסחאות עם 6 ליטרלים. לכן הסיבוכיות זמן של ϕ_{move} היא $O(n^{2k})$.

לפיכך בסה"כ הרדוקציה ניתנת לחישוב בזמן $O(n^{2k})$ ולכן הרדוקציה היא רדוקציה פולינומיאלית.

13.6 אסטרטגיות מנחות במשחקים

משחק בערך מוגדר תחרות בו שני שחקנים מתחרים להשיג מטרה כלשהי לפי כללי המשחק. משחקים באים במגוון צורות, כולל משחק לוח, כגון שחמט, משחקי כלכלה, משחקי מלחמה, משחקים שמודלים תחרות עסקית ותחרות חברתית.

יש קשר בין משחקים וcms. לכל פסוק עם cms קיים משחק מתאים, ולהפך כל משחק ניתן לתאר ע"י נוסחה עם cms. ראשוני תיאור פסוק עם cms כמשמעותו הבנה אינטואטיבית של המשמעות של הפסוק. מצד שני ביטוי של משחק ב邏輯 פסוק עם cms עוזר לנו להבין את האינטואיציות בין השחקנים, ההשפעה של התור של שחקן אחד על השני והتوزאה של המשחק.

למטה נגדיר משחק כזה שנראה **נוסחה בוליאנית FORMULAGAME**

הגדרה 13.14 FORMULAGAME

תהי

$$\psi = \exists x_1 \forall x_2 \exists x_2 \dots Q x_k \phi(x_1, \dots, x_k)$$

נוסחה בוליאנית עם cms בצורת PNF כאשר Q מסמן \forall או \exists . נגדיר משחק FORMULAGAME באופן הבא:

- יש שני שחקנים: אליס (A) ובוב (B).
- אליס ובוב בוחרים בתורים ערך של אחד המשתנים x_1, \dots, x_k .
- שחקן A בוחר את הערכים של אילו משתנים הנופלים תחת cms \forall .
- שחקן B בוחר את הערכים של אילו משתנים הנופלים תחת cms \exists .
- התורים של אליס ובוב בתורי המשחק הם לפי סדר הופעת cms' בנוסחה ψ .
- כיוון שלא נשרים משתנים בלי ערכיהם, בודקים את הערךאמת של הנוסחה ψ ללא cms' בתחילת הנוסחה. ככלומר בודקים את הערך של $\phi(x_1, \dots, x_k)$ עם הערכים של המשתנים שאليس ובוב בחרו.
- אם $\phi = 1$ אז בוב (שחקן B) מנצת.
- אם $\phi = 0$ אז אליס (שחקן A) מנצת.

13.7 המחלקה L

13.8 המחלקה NL

13.9 שלמות ב- NL

13.10 SHIWIION NL ו- coNL