



אוטומטים 1

מחברת סיכומים לקורס אוטומטים ושפות פורמאליות 2021

תוכן עניינים

2	מבוא
4	פעולות על מילים
5	פעולות על שפות
9	אוטומט סופי דטרמיניסטי - DFA
16	שפות רגולריות
19	סגירות שפות רגולריות תחת פעולות בוליאניות
24	אוטומט סופי לא-דטרמיניסטי - NFA
31	אסל"ד עם מסעי אפסילון - ENFA
35	סגירות תחת פעולות רגולריות
37	ביטויים רגולריים
42	למת הניפוח
46	שאלות ממבחנים ופתרונות מלאים

מוטיבציה

הכוח העיקרי של מחשב הוא היכולת לבצע את העיבוד של אינפורמציה שמגיעה כקלט, לאינפורמציה שיוצאת כפלט.

כשנרצה לתאר דבר שכזה בצורה פורמאלית, אנחנו רוצים שזה יהיה פשוט ככל האפשר כי יש המון סוגים שונים של אינפורמציה (מדיה, טקסט וכו').

כל הסוגים השונים של האינפורמציה בסופו של דבר מיוצגים במחשב באופן מאוד פשוט שהם ביטים (יחידת מידע בסיסית שהיא או 0 או 1).

לכן אפשר לחשוב על כל המידע שיש במחשב כסדרות של אפסים ואחדות וזה האופן שבו אנחנו רוצים לחשוב בקורס זה. כל המידע שאנו מתעסקים בו בקורס הולך להיות מיוצג על ידי סדרות של תווים כלשהם.

מבוא

הגדרה

אלפבית המיוצגת כסיגמא Σ היא קבוצה סופית כלשהי ולא ריקה.

דוגמה

$$\Sigma = \{1, 0\} \vee \Sigma = \{a, b, c\}$$

הגדרה

איבר של אלפבית Σ נקרא **אות** ולרוב נסמן אותיות אלה ב- σ (small sigma), τ (tau), γ (gamma).

הגדרה

מילה w מעל א"ב Σ היא סדרה סופית של אותיות מה- Σ .

דוגמה

למשל עבור $\Sigma = \{0, 1\}$ נכתוב אותיות ללא פסיקים ביניהן כלומר $w = 1010$.

הגדרה

שפה היא קבוצה של מילים והיא יכולה להיות סופית או אינסופית ונסמנה באות L .

דוגמה

למשל $L = \{aba, aab, b\}$ שפה סופית או $L = \{a, aa, aaa, \dots\}$ שפה אינסופית.

הגדרה

שפה ללא מילים נקראת **השפה הריקה** ומסומנת כך $\emptyset$.

הגדרה

לכל א"ב Σ , המילה שאין בה אותיות נקראת **המילה הריקה** ומסומנת באפסילון- ϵ .

נקודות חשובות

- ⊙ נציין ש- $\epsilon \neq \emptyset$ כלומר המילה הריקה היא לא השפה הריקה.
- ⊙ $L = \{\epsilon\}$ היא שפה שיש בה מילה אחת, שהיא המילה הריקה שולכן $\{\epsilon\} \neq \emptyset$.
- ⊙ מאחר ו- $\emptyset \notin \epsilon \vee \epsilon \notin \emptyset$ לא נמצאת בכל השפות.

הגדרה

אורך של מילה w מסומן ב- $|w|$ והוא מספר האותיות בה.

דוגמה

$$|01101| = 5, |\varepsilon| = 0, |a| = 1$$

הגדרה

תהי w מילה מעל א"ב Σ ויהא $\sigma \in \Sigma$ אות מתוך Σ אז **מספר המופעים** של האות σ במילה w מסומן $\#_{\sigma}(w)$.

דוגמה

$$\#_0(01111) = 1, \#_1(01111) = 5, \#_2(0111) = 0$$

הגדרה

Σ^* זה **שפת כל המילים מעל הא"ב Σ** .

דוגמה

$$\Sigma^* = \{\varepsilon, 0, 1, 01, 10, \dots, 1010, \dots\}$$

טענה

Σ^* אינסופית.

**נקודות חשובות**

- ⊙ המילה הריקה נמצאת בשפת כל המילים מעל האלפבית, $\varepsilon \in \Sigma^*$, מכיוון כי **כל המילים** נמצאות בה.
- ⊙ **האלפבית** לעולם **לא** יכול את ε (אפסילון סימן שמור אך ורק למילה הריקה ובאלפבית יש רק אותיות).
- ⊙ $\aleph_0$ היא קבוצת בת מניה אינסופית כאשר $\aleph_0 = |\mathbb{N} \text{ natural}|$ וגם $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ שוות עוצמה.
- ⊙ עוצמה הגדולה מבת מניה נקראת עוצמת הרצף ומסומנת $\aleph$ מהקבוצות $\mathbb{R}, P(\mathbb{Q}), P(\mathbb{Z}), P(\mathbb{N})$.
- ⊙ העוצמה של Σ^* , $|\Sigma^*| = \aleph_0$, סיגמא כוכבית, מורכבת מסדרות סופיות מעל קבוצה סופית.

⊙ לכל שפה L מתקיים ש- $\Sigma^* \subseteq L$ ולכן $|L| \leq \aleph_0$.

פעולות על מילים

הגדרה

בהינתן שתי מילים $w_1 = \sigma_1 \dots \sigma_n$ ו- $w_2 = \tau_1 \dots \tau_m$
השרשור של $w_1 \cdot w_2$ מוגדר להיות $w_1 \cdot w_2 := \sigma_1 \dots \sigma_n \cdot \tau_1 \dots \tau_m$.

⊙ הסימון $w_1 \cdot w_2$ נקרא השרשור (ולא כפל).

⊙ ייתכן ש- $w_1 \in L$ ו- $w_2 \in L$ אבל $w_1 \cdot w_2 \notin L$

למשל $L = \{aab, bba\}$ אבל $w_1 \cdot w_2 \notin L$ למרות ש- $w_1 \in L$ ו- $w_2 \in L$.

אבחנה

$$1. |w_1 \cdot w_2| = |w_1| + |w_2|$$

$$2. (w_1 \cdot (w_2 \cdot w_3)) = (w_1 \cdot w_2) \cdot w_3 \quad (\text{אסוציאטיביות/קיבוציות})$$

$$3. \varepsilon \cdot w = w \cdot \varepsilon = w$$

$$4. \text{לרוב } w_1 \cdot w_2 \neq w_2 \cdot w_1 \quad (\text{קומוטטיביות})$$

הגדרה

תהא w מילה כך ש- $w = u \cdot v$ אז u נקראת **רישא** של w ו- v נקראת **סיפא** של w .

הגדרה

חזקה של מילה w מוגדרת כשרשור שלה עם עצמה i פעמים (i טבעי) $w^i = w \cdot \dots \cdot w \Rightarrow (i \text{ times})$.

הגדרה רקורסיבית (פורמלית)

$$\text{עבור } i = 0 \text{ מתקיים } w^0 = \varepsilon$$

$$\text{עבור } i > 0 \text{ מתקיים } w^i = w \cdot w^{i-1}$$

אבחנה

$$w^1 = w^1 \cdot w^0 = w \cdot \varepsilon = w$$

הגדרה

תהי $w = a_1 \dots a_n$ מילה, **ההיפוך** (Reverse) של w מסומנת w^R ומתקיים $w^R = a_n \dots a_1$.

הגדרה רקורסיבית

$$w^R = w \text{ ואם } |w| = 0$$

$$w^R = a_n (a_1 \dots a_{n-1})^R \text{ ואם } |w| \neq 0$$

אבחנה

$$(abc)^R = c(ab)^R = cba^R = cba\varepsilon^R = cba$$

פעולות על שפות**הגדרה**

שפה L מעל האלפבית היא קבוצה של מילים מעל Σ .

תהיינה L_1, L_2 שפות מעל הא"ב Σ

$$1. \text{ איחוד: } L_1 \cup L_2 = \{w : w \in L_1 \vee w \in L_2\}$$

$$2. \text{ חיתוך: } L_1 \cap L_2 = \{w : w \in L_1 \wedge w \in L_2\}$$

$$3. \text{ שרשור: } L_1 \cdot L_2 = \{x \cdot y : x \in L_1 \wedge y \in L_2\}$$

$$4. \text{ היפוך: } L^R = \{w^R : w \in L\}$$

$$\text{למשל } L = \{aab, ba\}, L^R = \{baa, ab\}$$

$$5. \text{ חזקה: } L^i = L \cdot \dots \cdot L \Rightarrow (i \text{ times})$$

$$\text{הגדרה רקורסיבית: } L^0 = \{\varepsilon\}, L^i = L \cdot L^{i-1}$$

$$6. \text{ משלים: } \bar{L} = \Sigma^* \setminus L$$

דוגמה

$$L_1 = \{\varepsilon, 0, 11\}, L_2 = \{\varepsilon, 11\}$$

$$L_1 \cdot L_2 = \{\varepsilon \cdot \varepsilon, \varepsilon \cdot 11, 0 \cdot \varepsilon, 0 \cdot 11, 11 \cdot \varepsilon, 11 \cdot 11\} = \{\varepsilon, 11, 0, 011, 11, 1111\}$$

$$|L_1 \cdot L_2| = 6 \Rightarrow \text{שרשור}$$

$$|L_1| \cdot |L_2| = 3 \cdot 2 = 6 \Rightarrow \text{כפל}$$

כפל ושרשור הם **לא** אותו הדבר גם אם קיבלנו כאן אותו תוצאה!

טענה

$$L \cdot \emptyset = \emptyset \cdot L = \emptyset$$

הוכחה

אם $w \in \emptyset \cdot L$ אז קיימת $y \in L$, $x \in \emptyset$ כך ש- $w = x \cdot y$.
 אבל לא ייתכן ש- $x \in \emptyset$ לא קיימת אף מילה או בשפה הריקה ולכן קיבלנו סתירה.
 לכן אין w ששייכת ל- $\emptyset \cdot L$ ולכן $\emptyset \cdot L = \emptyset$.

I

טענה

$$L \cdot \{\varepsilon\} = \{\varepsilon\} \cdot L = L$$

הוכחה

אם $w \in L \cdot \{\varepsilon\}$ אז קיימות $y \in \{\varepsilon\}$, $x \in L$ כך ש- $w = x \cdot y$. מכאן $y = \varepsilon$ לכן $w = x \cdot \varepsilon$, כלומר לפי הגדרה על המילה הריקה, $x = w$ ולכן $w \in L$ ולכן $L \cdot \{\varepsilon\} \subseteq L$. - הכלה בכיוון השני מתקיים בדומה.

I

תכונות

- ⊙ **אסוציאטיביות:** $L_1 \cdot (L_2 \cdot L_3) = L_3 \cdot (L_1 \cdot L_2)$.
- ⊙ **הפרש:** $L_1 \setminus L_2 = \{w : w \in L_1 \wedge w \notin L_2\}$.
- ⊙ **דיסריטיביות:** $(L_1 \cup L_2) \cdot L_3 = L_1 \cdot L_3 \cup L_2 \cdot L_3$.

הגדרה

לכל שפה L נגדיר את **פעולת הכוכבית (סגור-קלייני / איטרציה)** של L בתור: $L^* = L^0 \cup L^1 \cup \dots$

$$L^* = L^0 \cup L^1 \cup L^2 \cup \dots = \{\varepsilon, a, a^2, \dots\}$$

$$\text{⊙ במקרה-1: } L^* = L^0 \cup \Sigma \cup \Sigma^1 \cup \dots = \Sigma^*$$

$$\text{⊙ במקרה-2: } L^* = \{\varepsilon\}, L^* = \emptyset, L^* \in \Sigma^* \text{ תמיד ולכן } L^* \neq \emptyset$$

$$\text{⊙ יתכן ש- } L_1^* \neq L_2^* \text{ אבל } L_1^* = L_2^*$$

$$\text{⊙ } \varepsilon \in L^*$$

טענה

$$(L^*)^* = L^*$$

$$(L^*)^* = \bigcup_{i=0}^{\infty} (L^*)^i = \bigcup_{i=0}^{\infty} \left(\bigcup_{j=0}^{\infty} L^j \right)^i = ? \quad \text{הסתכלות כללית (לא הוכחה- לצורך עזר)}$$

הוכחה נראה הכלה דו-כיוונית:

$$(L^*)^* \subseteq L^* \quad \text{לפי הגדרה של כוכבית (עבור } i = 1 \text{ ה-} L^* \text{ מופיעה באיחוד האינסופי } \bigcup_{i=0}^{\infty} (L^*)^i \text{).}$$

$$L^* \subseteq (L^*)^* \quad \text{תהי } w \in (L^*)^* \text{ לפי הגדרת האיטרציה, קיים } k \text{ טבעי כך ש- } w \in (L^*)^k.$$

$$\text{כלומר, קיימות } w_1, \dots, w_k \in L^* \text{ שכך ש- } w = w_1 \cdot \dots \cdot w_k.$$

$$\text{לכל } w_i \text{ מתקיים ש- } w_i \in L^* \text{, לכן קיימות } y_{i1}, y_{i2}, \dots, y_{il_i} \in L \text{ שכך ש- } w_i = y_{i1} \cdot y_{i2} \cdot \dots \cdot y_{il_i} \text{ (} w_i \in L^{l_i} \text{).}$$

$$\text{ולכן קיבלנו } w = w_1 \cdot w_2 \cdot \dots \cdot w_k = (y_{11} \cdot \dots \cdot y_{1l_1}) \cdot (y_{21} \cdot \dots \cdot y_{2l_2}) \cdot \dots \cdot (y_{k1} \cdot \dots \cdot y_{kl_k})$$

$$\text{ולפי קיבוציות, } w = y_{11} \cdot \dots \cdot y_{l_1} \cdot \dots \cdot y_{k1} \cdot \dots \cdot y_{kl_k} \text{, וולכן, } w \in L^{\sum_{i=1}^k l_i} \text{ ואז } w \in L^*.$$

I

הגדרה

$$\text{סגור-קלייני החיובי של } L, \text{ המסומן } L^+, \text{ מוגדר כך: } L^+ = \bigcup_{i=1}^{\infty} L^i. \quad \text{(כאן ה-} i \text{ מתחיל מ-1 ולא מ-0).}$$

$$L^+ \text{ מציינת שרשרת של אחד או יותר עותקים של } L. \text{ נשים לב שלכל שפה } L \text{ מתקיים } \varepsilon \in L^+.$$

$$\text{כמו כן, אפשר להוכיח שמתקיים } \varepsilon \in L^+ \Leftrightarrow \varepsilon \in L.$$

דוגמאות

$$\odot \quad \text{מהי שפת כל המילים מעל } \{a, b\} \text{ שכל מילה מתחילה ברצף } ab?$$

$$\text{לכן נשרשר: } \{ab\} \cdot \Sigma^*$$

$$\odot \quad \text{מהי שפת כל המילים מעל } \{a, b\} \text{ שמכילות את- } ab?$$

$$\text{לכן: } \Sigma^* \cdot \{ab\} \cdot \Sigma^*$$

$$\odot \quad \text{מהי שפת כל המילים מעל } \{a\} \text{ באורך זוגי?}$$

$$\bigcup_{i=0}^{\infty} \{a\}^{2i} = \{aa\}^* = \bigcup_{j=0}^{\infty} \{aa\}^j$$

⊙ מהי שפת כל המילים מעל $\Sigma = \{a\}$ באורך אי-זוגי?

$$\Sigma^* \setminus \{aa\}^* = \{a\} \cdot \{aa\}^* = \{a\} \cdot \bigcup_{i=0}^{\infty} \{aa\}^i = \bigcup_{i=0}^{\infty} \{a\}^{2i+1}$$

טעויות נפוצות עבור אי-זוגיים לא לעשות מן הסתם $\{aaa\}^*$ ולא לעשות $\{a\}^*$.

⊙ מהי שפת כל המילים מעל $\Sigma = \{a, b\}$ שמכילות a אחד בלבד?

$$\{b\}^* \cdot \{a\} \cdot \{b\}^*$$

⊙ מהי שפת כל המילים מעל $\Sigma = \{a, b\}$ שמכילות ab ולא bb?

$$(\Sigma^* \cdot \{ab\} \cdot \Sigma^*) \setminus (\Sigma^* \cdot \{bb\} \cdot \Sigma^*)$$

תרגיל

יהי Σ א"ב כלשהו, ותהיינה L, L_1, L_2 שפות כלשהן מעל הא"ב Σ .

עבור כל אחת מהטענות הבאות, קבעו האם מתקיים שוויון, או שמתקיימת הכלה בכיוון אחד בלבד, או שלא מתקיימת הכלה באף כיוון. עבור כל טענה של שוויון או הכלה עליכם להוכיח את תשובתכם.

עבור כל טענה שאין הכלה, יש לספק דוגמה נגדית.

א. $(L \cup \{\varepsilon\})^* = L^*$

ב. $L^* = (L^+)^+$

ג. $(L_1 \cup L_2)^* = (L_1^* \cdot L_2^*)^*$

ד. $\Sigma^* = \Sigma^* \cdot \Sigma^*$

פתרון

א. הוכחה:

$$\begin{aligned} (L \cup \{\varepsilon\})^* &= \bigcup_{i=0}^{\infty} (L \cup \{\varepsilon\})^i = \bigcup_{i=0}^{\infty} (L \cup L^0)^i = L^0 \cup \bigcup_{i=1}^{\infty} (L \cup L^0)^i = \\ &= L^0 \cup \bigcup_{i=1}^{\infty} (L - \{\varepsilon\})^i = L^0 \cup (L - \{\varepsilon\})^+ = L^0 \cup L^+ = L^* \end{aligned}$$

ב. צד $(L^+)^+ \subseteq L^*$:

$$\begin{aligned} w \in (L^+)^+ &\Rightarrow w \in \left(\bigcup_{i=1}^{\infty} L^i\right)^+ \Rightarrow w \in \bigcup_{j=1}^{\infty} \left(\bigcup_{i=1}^{\infty} L^i\right)^j \Rightarrow \forall k \geq 1 : k = i + j \Rightarrow \\ &\Rightarrow w \in \bigcup_{k=1}^{\infty} L^k \Rightarrow w \in \bigcup_{k=0}^{\infty} L^k \Rightarrow w \in L^* \end{aligned}$$

הצד השני **לא מתקיים** בהכרח וזאת מאחר כי לפי הגדרה - $\varepsilon \in L^*$ אבל $\varepsilon \notin (L^+)^+$.

ג. נוכיח באמצעות הכלה דו-כיוונית.

$$\begin{aligned} w \in (L_1 \cup L_2)^* &\Leftrightarrow w \in (L_1 \cdot \{\varepsilon\} \cup \{\varepsilon\} \cdot L_2)^* \Leftrightarrow w \in (L_1 \cdot L_2^0 \cup L_1^0 \cdot L_2)^* \Leftrightarrow \\ &\Leftrightarrow w \in (L_1^* \cdot L_2^* \cup L_1^* \cdot L_2^*)^* \Leftrightarrow w \in (L_1^* \cdot L_2^*)^* \end{aligned}$$

ד. הוכחה:

$$\begin{aligned} w \in \Sigma^* &\Leftrightarrow \exists k \in \mathbb{N} : w \in \Sigma^k \Leftrightarrow \\ &\Leftrightarrow \exists i, j \in \mathbb{N} : k = i + j \wedge w \in \Sigma^i \cdot \Sigma^j \Leftrightarrow w \in \Sigma^* \cdot \Sigma^* \end{aligned}$$

אוטומט סופי דטרמיניסטי (DFA)

הגדרה

אוטומט סופי דטרמיניסטי או באנגלית *Deterministic Finite Automaton* הוא מודל מתמטי שמגיב למילים, כלומר למחרוזות סופיות כקלט.

האוטומט מחזיר בתור פלט כן או לא על כל מילה באלפבית.

האוטומט "מזהה" שפות פורמליות, כלומר האוטומט יכול לזהות אם מילה היא בשפה הנתונה או לא.

האוטומט מורכב מ-

1. סרט - (שבו רשום הקלט, בכל תא בסרט כתובה אות אחת).
2. ראש קורא - (אפשר להסתכל עליו בתור "פוינטר" על כל תא בסרט).
3. בקרה מרכזית / מכונת מצבים - ("האלגוריתם" - מוגבל מאוד).
4. פלט.

אוטומט סופי דטרמיניסטי הוא חמישייה: $A = (Q, \Sigma, q_0, \delta, F)$

- Q - קבוצה סופית של מצבים.
- Σ - אלפבית כלשהו. (איזה אותיות עשויות להגיע בתור קלטים).
- $q_0 \in Q$ - "מצב התחלתי". (בפרט זה מבטיח ש- Q היא לא ריקה).
- $F \subseteq Q$ - "מצבים מקבלים".
- $\delta: Q \times \Sigma \rightarrow Q$ - "פונקציית מעברים" המוגדרת Q לרוב נוח לתאר את δ באמצעות טבלה.

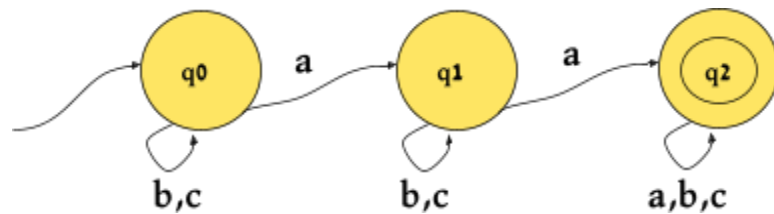
דוגמה

$$\Sigma = \{a, b, c\}$$

$$L = \{w : \#_a(w) \geq 2\}$$

$$Q = \{q_0, q_1, q_2\}$$

$$F = \{q_2\}$$

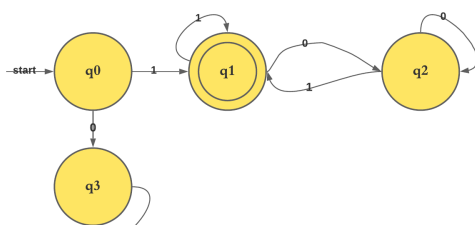


נתאר את פונקציית המעברים δ בטבלה:

	a	b	c
q_0	q_1	q_0	q_0
q_1	q_2	q_1	q_1
q_2	q_2	q_2	q_2

- חוגים עצמיים בגרף האוטומט הם כל q_i $\delta(q_i, \sigma) = q_i$.
- לדוגמה עבור הטבלה למעלה $\delta(q_0, c) = q_0$ הוא חוג עצמי.

יהי $\Sigma = \{0, 1\}$, נתבונן בתיאור גרפי של אוטומט A_0 :



אוטומט A_0 מקבל אך ורק מילים שמתחילות ומסתיימות באות 1.

אם המילה מתחילה באות 0, עובר האוטומט למצב q_3 ולא יוצא ממנו יותר - מצב כזה מכונה מצב "מלכודת".

העיגול הכפול בנמצא ב- q_1 נקרא מצב "מקבל" -

כלומר אם המצב האחרון שאליו הגיע האוטומט לאחר סיום קריאת מילת הקלט הינו מצב מקבל, אז אומרים כי האוטומט קיבל את מילת הקלט;

אחרת - דחה האוטומט את מילת הקלט.

המעברים בין q_1 ל- q_2 נקרא "מתנדנד".

תרגיל

Let

$L = \{w \in \{0, 1, 2\}^* : w \text{ is a ternary representation of an integer that is a multiple of } 5\}$

Give a DFA (Deterministic Finite Automaton) for L.

פתרון

בדומה למספר בינארי המיוצג בבסיס 2 עבור המספרים $\{0,1\}$, המספר הטרינרי הוא מספר המיוצג בבסיס 3 עבור קבוצת המספרים $\{0,1,2\}$ בלבד.

המרת מספר זה לייצוג עשרוני מתבצע כך לדוגמה:

$$(0212)_3 = 0 \cdot 27 + 2 \cdot 9 + 1 \cdot 3 + 2 \cdot 1 = (23)_{10}$$

אנו צריכים לקבל את המחרוזות שהם הכפולות של 5.

טיפ - תמיד שנפגוש תרגיל עם תנאי של כפולה של מספר כלשהו, נעזר באריתמטיקה מודולרית.

כאשר מתבקש לייצג DFA מה שאנו צריכים לעשות הוא לייצר מצבים עבור כל אחד מהאפשרויות,

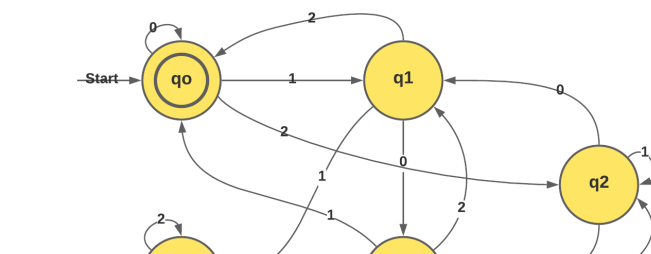
עבור מקרה של תרגיל זה, ישנם 5 מקרים : בכל פעם שנחלק ב-5 נקבל שארית כלשהי, 0,1,2,3 או 4

והם כל האפשרויות שלנו ולכן ניצור מצב עבור כל אחד מהאפשרויות האלה.

במידה וקיבלנו שארית 0 אז אנו במקרה טוב כלומר המספר אכן כפולה של 5.

טבלה לנוחות הפתרון עבור שרטוטי המעברים והמצבים:

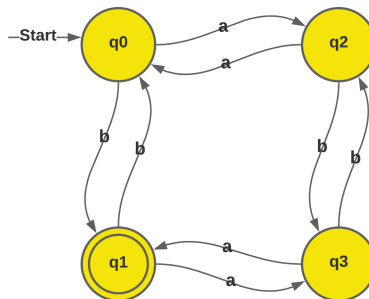
ביצוע	חישוב
$\Rightarrow q_0, B/c 3 \cdot 0 + 0 = 0$ $3 \cdot q_0 + 1 \Rightarrow q_1, B/c 3 \cdot 0 + 1 = 1$ $3 \cdot q_0 + 2 \Rightarrow q_2, B/c 3 \cdot 0 + 2 = 2$	
$\Rightarrow q_3, B/c 3 \cdot 1 + 0 = 3$ $3 \cdot q_1 + 1 \Rightarrow q_4, B/c 3 \cdot 1 + 1 = 4$ $3 \cdot q_1 + 2 \Rightarrow q_0, B/c 3 \cdot 1 + 2 = 5$	



$3 \cdot q_2 + 1$	$\Rightarrow q_1, B/c \ 3 \cdot 2 + 0 = 6 \rightarrow 5 + 1$
$3 \cdot q_2 + 2$	$\Rightarrow q_2, B/c \ 3 \cdot 2 + 1 = 7 \rightarrow 5 + 2$
	$\Rightarrow q_3, B/c \ 3 \cdot 2 + 2 = 8 \rightarrow 5 + 3$
$3 \cdot q_3 + 1$	$\Rightarrow q_4, B/c \ 3 \cdot 3 + 0 = 9 \rightarrow 5 + 4$
$3 \cdot q_3 + 2$	$\Rightarrow q_0, B/c \ 3 \cdot 3 + 1 = 10 \rightarrow 5 + 5$
	$\Rightarrow q_1, B/c \ 3 \cdot 3 + 2 = 11 \rightarrow 5 + 1$
$3 \cdot q_4 + 1$	$\Rightarrow q_2, B/c \ 3 \cdot 4 + 0 = 12 \rightarrow 5 + 5$
$3 \cdot q_4 + 2$	$\Rightarrow q_3, B/c \ 3 \cdot 4 + 1 = 13 \rightarrow 5 + 5$
	$\Rightarrow q_4, B/c \ 3 \cdot 4 + 2 = 14 \rightarrow 5 + 5$

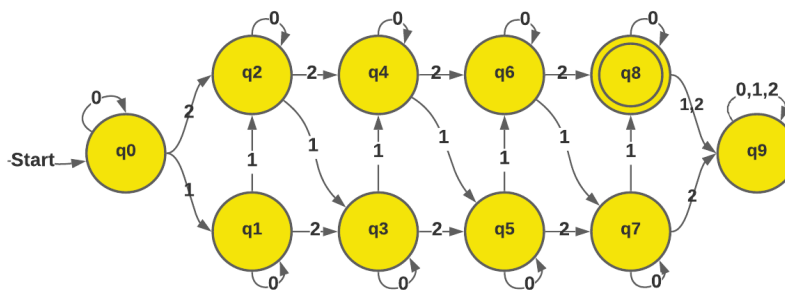
תרגיל

שפת כל המילים מעל $\{a, b\}$ שכך שמספר ה-a זוגי ומספר ה-b אי-זוגי.



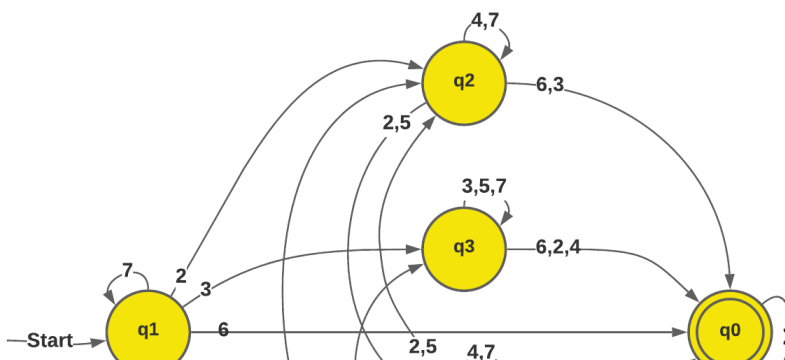
תרגיל

שפת כל המילים מעל $\{0, 1, 2\}$ שכך שסכום הספרות במילה שווה ל-8.



תרגיל

שפת כל המילים מעל $\{2, 3, 4, 5, 6, 7\}$ שכך שמכפלת הספרות במילה מתחלקת ב-6 ללא שארית.



הגדרה

בהינתן אוטומט A נגדיר את **פונקציית המעברים המורחבת** שלו $\widehat{\delta}_A: Q \times \Sigma^* \rightarrow Q$ באופן אינדוקטיבי:

$$\begin{aligned}\widehat{\delta}_A(q, \varepsilon) &\stackrel{\text{def}}{=} q \\ \widehat{\delta}_A(q, w \cdot \sigma) &\stackrel{\text{def}}{=} \widehat{\delta}_A(\widehat{\delta}_A(q, w), \sigma)\end{aligned}$$

טענה

עבור אס"ד A מתקיים שלכל $q \in Q_A$ ולכל $x, y \in \Sigma_A^*$ ש- $\widehat{\delta}_A(q, x \cdot y) = \widehat{\delta}_A(\widehat{\delta}_A(q, x), y)$

הוכחה

נוכיח את הטענה באינדוקציה על $|y|$.

בסיס: עבור $|y| = 0$, כלומר עבור $y = \varepsilon$, מתקיים: $\widehat{\delta}_A(q, x \cdot \varepsilon) = \widehat{\delta}_A(q, x) = \widehat{\delta}_A(\widehat{\delta}_A(q, x), \varepsilon)$

הנחה: נניח שהטענה נכונה עבור כל $|y| = n \geq 1$, נוכיח שהטענה נכונה לכל $|y| = n + 1$.

צעד: תהי מילה y כך ש: $|y| = n + 1$, נפרק את המילה בצורה הבאה: $y = t \cdot \sigma$ כאשר $|t| = n$

כלומר $t \in \Sigma^*$ וגם $\sigma \in \Sigma$ (תו יחיד $|\sigma| = 1$).

$$\begin{aligned}\widehat{\delta}_A(q, x \cdot y) &= \widehat{\delta}_A(q, x \cdot t \cdot \sigma) = \widehat{\delta}_A(\widehat{\delta}_A(q, x \cdot t), \sigma) = \widehat{\delta}_A(\widehat{\delta}_A(\widehat{\delta}_A(q, x), t), \sigma) = \\ &= \widehat{\delta}_A(\widehat{\delta}_A(q, x), t \cdot \sigma) = \widehat{\delta}_A(\widehat{\delta}_A(q, x), y)\end{aligned}$$

I

הערה

• $\varepsilon \in L(A)$ אם ורק אם המצב ההתחלתי הוא מצב מקבל.

• לעיתים נוח לדבר על שפת $L(q)$ - עבור מצב נתון q .

שפה זו היא שפת המילים המביאות את האס"ד A מהמצב ההתחלתי q_{0A} למצב נתון q , כאשר q לא בהכרח מצב מקבל, כלומר - עבור $q \in Q_A$ נגדיר:

$$L(q) = \{x: x \in \Sigma_A^*, \delta_A(q_{0A}, x) = q\}$$

הגדרה

בהינתן אוטומט A נגדיר את **שפת האוטומט** בצורה הבאה:

$$L(A) \stackrel{\text{def}}{=} \{w \in \Sigma^* : \widehat{\delta}(q_0, w) \in F_A\}$$

הגדרה

בהינתן אוטומט A ומצב כלשהו $q \in Q_A$ נגדיר את **שפת המצב** q בצורה הבאה:

$$L(q) \stackrel{\text{def}}{=} \{w \in \Sigma^* : \widehat{\delta}(q_0, w) = q\}$$

$$L(A) = \bigcup_{q \in F_A} L(q)$$

כעת ניתן להגדיר כי

דוגמה להוכחת שפת האוטומט

$$L(A) = \{w : |w| \text{ is even}\}$$

כדי להוכיח ששפת האוטומט A היא השפה המתוארת,

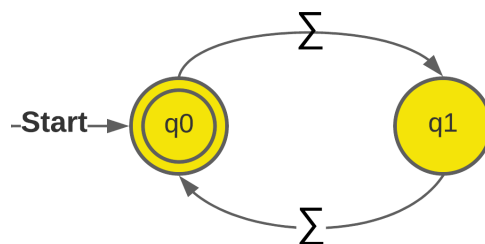
נוח להגדיר שתי שפות:

$$L_{\text{even}} = \{w \in \Sigma^* : |w| \bmod 2 = 0\}$$

$$L_{\text{odd}} = \{w \in \Sigma^* : |w| \bmod 2 = 1\}$$

ה- Σ באיור מוגדר כך שכל תו באלפבית יכול להשלח / להתקבל או במילים אחרות לכל $\sigma \in \Sigma$ סמתקיים

$$\delta_A(q_1, \sigma) = q_0 \in F_A \text{ וגם } \delta_A(q_0, \sigma) = q_1$$



טענה

$$[1] L(A) = L_{even}$$

הוכחה

נוכיח באינדוקציה על מבנה Σ^* .

👉 **בסיס האינדוקציה:** (במקרה ש- $\varepsilon = w$) עלינו להראות ש- $\varepsilon \in L(A) \Leftrightarrow \varepsilon \in L_{even}$.

$$\varepsilon \in L(A) \Leftrightarrow \widehat{\delta}_A(q_0, \varepsilon) \in F_A \Leftrightarrow q_0 \in F_A$$

ומאחר ו- q_0 היא המילה הריקה כלומר באורך 0 (זוגי) אזי $\varepsilon \in L_{even}$.

ולכן הוכחנו את מקרה הבסיס.

👉 **צעד האינדוקציה:** [2] נניח כי הטענה נכונה עבור w ונוכיח עבור $w \cdot \sigma$.

⇐ **כיוון א':** נניח כי $\sigma \in L_{even}$ ולכן [3] $w \in L_{odd}$ שכלומר מתקיים:

$$w \in L_{odd} \Rightarrow \widehat{\delta}_A(q_0, w) = q_1 \Rightarrow \widehat{\delta}_A(q_0, w \cdot \sigma) \stackrel{\text{def}}{=} \widehat{\delta}_A(\widehat{\delta}_A(q_0, w), \sigma) = \widehat{\delta}_A(q_1, \sigma) = q_0$$

ולכן קיבלנו $w \cdot \sigma \in L(q_0)$ שכמבוקש.

⇒ **כיוון ב':** נניח כי $w \cdot \sigma \in L(q_0)$ שכלומר, $\widehat{\delta}_A(q_0, w \cdot \sigma) = q_0$ ונוכיח ש $w \cdot \sigma \in L_{even}$.

$$\begin{aligned} \widehat{\delta}_A(q_0, w \cdot \sigma) = q_0 &\Rightarrow \widehat{\delta}_A(\widehat{\delta}_A(q_0, w), \sigma) = q_0 \Rightarrow [4] \Rightarrow \widehat{\delta}_A(q_0, w) = q_1 \\ &\Rightarrow \widehat{\delta}_A(q_1, \sigma) = q_0 \Rightarrow [5] \Rightarrow w \in L_{odd} \Rightarrow w \cdot \sigma \in L_{even} \end{aligned}$$

ולכן הוכחנו ש- $w \in L(q_0) \Leftrightarrow w \in L_{even}$.

ההוכחה עבור $w \in L(q_1) \Leftrightarrow w \in L_{odd}$ דומה [6].

[1] כדי להוכיח שוויון זה נוכיח בשיטת הכלה דו-כיוונית של קבוצות כלומר $w \in L(A) \Leftrightarrow w \in L_{even}$.

[2] לפי ההנחה $w \in L(A) \Leftrightarrow w \in L_{even}$ נרצה להוכיח ש- $w \cdot \sigma \in L(A) \Leftrightarrow w \cdot \sigma \in L_{even}$.

[3] הנחת האינדוקציה שלנו לא מספיק חזקה כי הרי הנחנו ש- $w \in L_{even}$ שולכן אנו צריכים לחזק את ההנחה.

לכן נשנה את הטענה $w \in L_{even} \Leftrightarrow w \in L(q_0) \Leftrightarrow w \in L_{odd}$ שוגם $w \in L(q_1) \Leftrightarrow w \in L_{odd}$.

בנוסף, אם $n = |w \cdot \sigma|$ באורך זוגי, אז ברור כי $n - 1 = |w|$ הוא באורך אי-זוגי.

[4] על-פי מבנה האוטומט נגיע למצב q_0 ממצב q_1 .

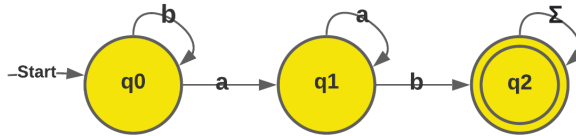
[5] לפי הנחת האינדוקציה אם w הוא מאורך אי-זוגי אז נוכל להסיק מיד ש- $w \cdot \sigma \in L_{\text{even}}$.

[6] בסיס $|w| = 1$, צעד $\sigma \in L_{\text{odd}}$, $w \cdot \sigma$ וכו'...

תרגיל

הוכח ששפת האוטומט היא שפת כל המילים המכילות ab .

פתרון



לפני ההוכחה "נכין את השטח", כלומר נצייר לנו אוטומט שמקיים את השפה הזו וננסה להבין איזה צורה של מילה w תתאים לנו להוכחה.

כדי שמילה תכיל את הרצף ab , נציג את w להיות:

$$w = x \cdot ab \cdot y. \text{ נרצה להוכיח ש- } L(A) = L, \text{ כלומר, אנחנו נוכיח ש- } w \in L(A) \Leftrightarrow w \in L.$$

הוכחה

⇒ כיוון א':

נציג את הפירוק של w באופן הבא: $w = x \cdot ab \cdot y$ ש- $x, y \in \Sigma^*$.

נניח ש- $w \in L$ שמילה המכילה את הרצף ab , נרצה להוכיח ש- $w \in L(A)$,

כלומר נוכיח שהמילה שייכת לשפת האוטומט.

לפי פונקצית המעברים של האוטומט A , עבור כל הערכים שבונים את המילה w ,

נבדוק את כל אחד מהמעבר למצבים הבאים לכל $0 \leq i \leq 2$ עם הפונקציה $\hat{\delta}(q_0, w) = q_i$

עבור מצב q_0 נקבל כאשר $\hat{\delta}(q_0, x) = q_0$ ש- x בנויה מכל תו שהוא לא a .

עבור מצב q_1 נקבל כאשר $\hat{\delta}(q_0, x \cdot a) = q_1$.

עבור מצב q_2 נקבל כאשר $\hat{\delta}(q_0, x \cdot ab) = q_2$.

ולכן לפי פונקציית המעברים נקבל ש:

$$\hat{\delta}(q_0, w) = \hat{\delta}(q_0, x \cdot ab \cdot y) = \hat{\delta}(\hat{\delta}(q_0, x \cdot ab), y) = \hat{\delta}(q_2, y) = q_2 \in F_A$$

ולכן $w \in L(A)$.

נניח ש $w \in L(A)$, כלומר $\hat{\delta}(q_0, w) = q_2 \in F_A$.

⇒ כיוון ב':

נוכיח ש- $w \in L$ כלומר נראה שהמילה w מכילה את הרצף ab .

טענה - אם $\hat{\delta}(q_0, w) = q_2$ אז w מכילה את הרצף ab . נוכיח טענה זו באינדוקציה על $|w|$:

☞ **בסיס האינדוקציה:** הטענה מתקיימת באופן ריק: $\varepsilon = w$ כלומר $|w| = 0$.

☞ **צעד האינדוקציה:** נניח שהטענה נכונה לכל $|w| = n \geq 0$, תהא $|w| = n + 1$ כאשר $\hat{\delta}(q_0, w) = q_2$, נפרק את w בצורה הבאה כך ש- $w = t \cdot \sigma$ עבור $t \in \Sigma^*$ וגם $\sigma \in \Sigma$.

לפי הנחת האינדוקציה, אם $\hat{\delta}(q_0, t) = q_2$ אז t מכילה את הרצף ab ולכן גם w עצמה.

לפי מבנה האוטומט, $\hat{\delta}(q_0, t) = q_1$ ומאחר ו- $w \in L(A)$ אז $\sigma = b$ ונעבור למצב מקבל $\hat{\delta}(q_0, t \cdot \sigma) = q_2$, ולפי הנחת האינדוקציה, w מכילה את הרצף ab .

ולכן $w \in L$.

I

שפות רגולריות

הגדרה

שפה L נקראת **שפה רגולרית** אם קיים אוטומט סופי דטרמיניסטי A כך שמתקיים $L(A) = L$.

הגדרה

אוסף כל השפות הרגולריות מעל אלפבית נתון Σ נקרא מחלקת השפות הרגולריות.

שאלה

האם יש שפות לא רגולריות?

תשובה

אם יש לנו רק מספר בן מניה של אוטומטים סופיים דטרמיניסטים וכל אוטומט מקבל רק שפה אחת אז יש רק מספר בן מניה של שפות רגולריות ויש מספר לא בן מניה של שפות בכלל.

אז לא רק שיש שפות לא רגולריות אלא **שכמעט כל השפות לא רגולריות**.

טענה

ישנן שפות לא רגולריות

הוכחה

אוטומט סופי דטרמיניסטי מקבל שפה אחת ויחידה.

לכן עוצמת קבוצת השפות הרגולריות $\geq$ עוצמת קבוצת כל האס"ד-ים.

אפשר להסתכל על כל אס"ד כעל סדרה סופית: $A = (Q, \Sigma, q_0, \delta, F)$

עם מספר סופי של סימנים: Σ, q_0, δ, F .

עוצמת קבוצת הסדרות הסופיות עם מספר סופי של סימנים היא בת מניה - $\aleph_0$.

לכן עוצמת קבוצת השפות הרגולריות $\geq \aleph_0$.

מהי עוצמת קבוצת השפות (כלשהן)?

ידוע לנו לפי הגדרה של שפה כי $\Sigma^* \subseteq L$ וגם ידוע לנו ש- $|\Sigma^*| = \aleph_0$.

אז קבוצת השפות מעל Σ נתון, שווה לקבוצת החזקה של Σ^* כלומר $P(\Sigma^*)$ (כל תתי הקבוצות של Σ^*).

ידוע לנו מתורת הקבוצות ש- $\aleph_0 < 2^{\aleph_0} = 2^{|\Sigma^*|} = |P(\Sigma^*)|$ כלומר יש לנו $2^{\aleph_0}$ שפות באופן כללי ויש רק $\aleph_0$ שפות רגולריות.

I

טענה

השפה $L = \{a^i \cdot b^j : i \geq 0\}$ לא רגולרית.

הוכחה

נניח בשלילה כי קיים אוטומט $A = (Q, \Sigma, q_0, \delta, F)$ כך ש- $L(A) = L$.

נתבונן בסדרת המצבים $q_0, q_1, \dots, q_i$ כך ש- $q_i = \delta(q_0, a^i)$.

כיוון שמספר המצבים סופי ואורך הסדרה אינסופי, אז קיימים $i < j$ כך ש- $q_i = q_j$ כלומר

$$\delta(q_0, a^i) = \delta(q_0, a^j) \text{ ועל-פי תכונות השרשור נקבל כי:}$$

$$\delta(q_0, a^i \cdot b^j) = \delta(\delta(q_0, a^i), b^j) = \delta(\delta(q_0, a^j), b^j) = \delta(q_0, a^j \cdot b^j)$$

אבל $\delta(q_0, a^i \cdot b^j) \in F_A$ ו- $\delta(q_0, a^j \cdot b^j) \notin F_A$. בסתירה לשוויון הנ"ל.

I

⊙ תמיד כשנרצה להוכיח ששפה כלשהי לא רגולרית, נוכיח בשלילה שהיא כן רגולרית.

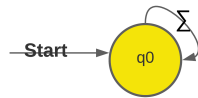
אבל איזו שפה היא כן שפה רגולרית?

טענה

השפה הריקה $\emptyset$ היא רגולרית.

הוכחה

שפה זו מתקבלת על-ידי כל אוטומט סופי דטרמיניסטי חסר מצבים מקבלים.

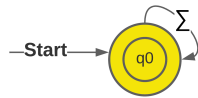


טענה

Σ^* היא שפה רגולרית.

הוכחה

נתבונן באוטומט A, שתיאורו הגרפי נתון כך:

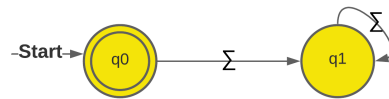


טענה

השפה המכילה את המילה הריקה בלבד, $L = \{\epsilon\}$, היא שפה רגולרית.

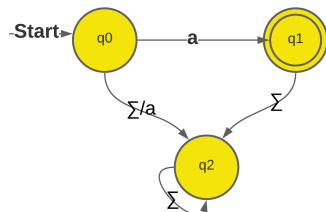
הוכחה

תיאור גרפי של אוטומט המקבל שפה זו:



טענה

לכל $\alpha \in \Sigma$ השפה $L_a = \{\alpha\}$ היא שפה רגולרית.



הוכחה

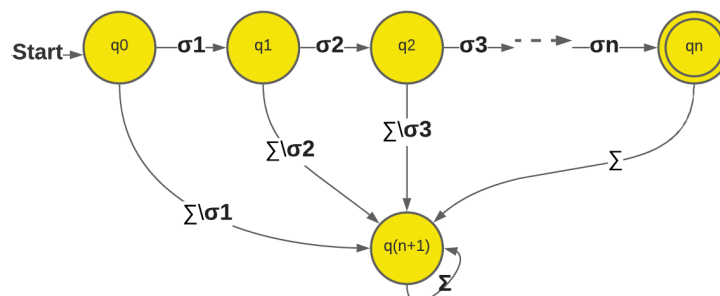
נראה תיאור גרפי של אוטומט המקבל את השפה הזאת:

טענה

לכל $w \in \Sigma^*$, השפה שהאיבר היחיד בה הוא מילה אחת $\{w\}$, היא שפה רגולרית.

הסבר

כתוב $w = \sigma_1 \cdot \sigma_2 \cdot \dots \cdot \sigma_n$ ונבנה אוטומט כך שכל מיקום של תו במילה הוא גם מצב מסוים באוטומט. כלומר האוטומט בעל צורת מצבים המורכבים לפי הסדר (משמאל לימין) של המילה w בדיוק. לבסוף התו האחרון במילה הינו מצב מקבל. כל תו "זר" למצב נוכחי כלשהו באוטומט עובר למצב בור (כולל המצב המקבל כי הרי אם מתקבל עוד תו אחרי המילה החוקית, היא כבר איננה מתקבלת באוטומט).



צורה פורמלית

נראה צורה פורמלית לאוטומט שהצגנו בהסבר האחרון.

קבוצת המצבים היא $Q = \{q_0, q_1, \dots, q_{n+1}\}$ קבוצת המצבים המקבלים היא $F = \{q_n\}$

פונקציית המעברים מוצגת בצורה הבאה,

לכל $1 \leq i \leq n$ נגדיר $\delta(q_{i-1}, \sigma_i) = q_i$ ולכל $\sigma_i \neq \sigma'_{i-1}$ נגדיר $\delta(q_{i-1}, \sigma') = q_{n+1}$.

כמו כן נגדיר לכל $\sigma \in \Sigma$: $\delta(q_{n+1}, \sigma) = q_{n+1}$, $\delta(q_n, \sigma) = q_{n+1}$.

נקודות חשובות

- כל שפה שהעוצמה שלה סופית, היא שפה רגולרית.
- אם L_2 רגולרית ו- $L_1 \subseteq L_2$, זה **לא מחייב** ש- L_1 היא גם רגולרית!.
- אם L רגולרית אז גם $\bar{L}$ רגולרית.
- אם L_1 ו- L_2 שפות רגולריות, אז גם $L_1 \cdot L_2$ שפות רגולריות.
- אם L_1 ו- L_2 שפות רגולריות, אז גם $L_1 \cap L_2$ שפות רגולריות וגם $L_1 \cup L_2$ שפות רגולריות.

סגירות שפות רגולריות תחת פעולות בוליאניות

להוכיח שפה רגולרית יכול להיות כאב ראש אחד גדול,

לכן נמצא דרכי קיצור שיעזור לנו להימנע מלהוכיח כל פעם מחדש ששפה היא רגולרית.

אנחנו רוצים למצוא דרך לבנות שפות רגולריות חדשות מתוך השפות הרגולריות שכבר יש לנו.

נרצה למצוא כללי בניה חדשים לשפות רגולריות ולהוכיח את הכללים האלו.

שיטה זו נקראת רדוקציה.

תכונות סגור של שפות

בהינתן פעולה R כלשהי המקבלת כקלט שפות $L_1, \dots, L_n$ ומחזירה כפלט שפה $L = R(L_1, \dots, L_n)$

נאמר ש- R היא **תכונת סגור רגולרית** אם $L_1, \dots, L_n$ רגולרית אז $R(L_1, \dots, L_n)$ רגולרית.

(זה לא קשור לסגור קלייני/איטרציה) אלא מדובר בסגירות לפעולות כמו פעולת המשלים.

ניקח תכונות שונות של שפות רגולריות ונבדוק אם תוצאת הפעלת הפעולה תהייה **בהכרח** שפה רגולרית כמו

משלים, חיתוך ואיחוד.

שאלה

האם שפה רגולרית סגורה תחת פעולת ההכלה? כלומר $L_1 \subseteq L_2$.

1. אם L_1 רגולרית אז האם בהכרח L_2 רגולרית?

2. אם L_2 רגולרית אז האם בהכרח L_1 רגולרית?

תשובה

עבור סעיף 2 נציג דוגמה נגדית: הוכחנו ש- Σ^* רגולרית, והוכחנו ש- $L_1 = \{0^n \cdot 1^n : n \geq 1\}$ לא רגולרית. קל לראות ש- $L_1 \subseteq L_2$ בסתירה לסעיף 2.

עבור סעיף 1 נציג דוגמה נגדית: הוכחנו ש- $L_2 = \{0^n \cdot 1^n : n \geq 1\}$ לא רגולרית והוכחנו ש- $L_1 = \emptyset$ היא שפה רגולרית בסתירה לסעיף 1.

לכן, השפה הרגולרית לא סגורה תחת פעולה ההכלה.

משפט

השפות הרגולריות סגורות תחת פעולת המשלים.

תיאור פורמלי

תהא L שפה רגולרית, נוכיח שגם $\bar{L} \stackrel{\text{def}}{=} \Sigma^* \setminus L$ שפה רגולרית.

מכיוון ש- L רגולרית קיים $A = (Q, \Sigma, q_0, \delta, F)$ כך ש- $L(A) = L$.

נגדיר אוטומט חדש שיש לו אותם תכונות כמו A חוץ מקבוצת המצבים המקבלים:

$$\bar{A} = (Q, \Sigma, q_0, \delta, Q \setminus F)$$

טענה

$$L(\bar{A}) = \bar{L}$$

הוכחה

$$w \in \bar{L} \Leftrightarrow w \notin L = L(A) \Leftrightarrow \delta(q_0, w) \notin F_A \Leftrightarrow \delta(q_0, w) \in Q \setminus F_A \Leftrightarrow w \in L(\bar{A})$$

I

⊙ אם L היא שפה רגולרית אז קיים עבודה אוטומט אס"ד כך ש: $L(A) = L$

ואם נרצה לקבל את השפה המשלימה $\bar{L}$ נהפוך את המצבים הלא מקבלים למצבים מקבלים ואת המצבים המקבלים ללא מקבלים.

משפט

אם L_1, L_2 שפות רגולריות אז $L_1 \cap L_2$ שפה רגולרית.

כלומר השפות הרגולריות סגורות תחת פעולת החיתוך.

הוכחה

[0] יהיו $A_2 = (Q_2, \Sigma, q_0, \delta_2, F_2)$, $A_1 = (Q_1, \Sigma, q_0, \delta_1, F_1)$ אוטומטים, כך ש: $L(A_2) = L_2$, $L(A_1) = L_1$. נבנה **אוטומט מכפלה** עבור $L_1 \cap L_2$:
 $A = (Q, \Sigma, q_0, \delta, F)$ כאשר $Q = Q_1 \times Q_2$ [1], $q_0 = (q_{0_1}, q_{0_2})$, $F = F_1 \times F_2$ [2], $\delta((q_1, q_2), \sigma) \stackrel{\text{def}}{=} (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma))$:
 ופונקציית המעברים מוגדרת כך: $\delta((q_1, q_2), \sigma) \stackrel{\text{def}}{=} (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma))$.
 נוכיח שהאוטומט מקבל את $L_1 \cap L_2$ [3].

טענת עזר [4]

לכל מילה $w \in \Sigma^*$, $q_1 \in Q_1$, $q_2 \in Q_2$, מתקיים $\hat{\delta}((q_1, q_2), w) = (\hat{\delta}_1(q_1, w), \hat{\delta}_2(q_2, w))$.

הוכחת טענת העזר

נוכיח טענה זו באינדוקציה על $|w|$.

בסיס האינדוקציה: נניח כי $w = \varepsilon$, $|w| = 0$.

מהגדרת פונקציית המעברים נקבל: $\delta((q_1, q_2), \varepsilon) = (q_1, q_2) = (\delta_1(q_1, \varepsilon), \delta_2(q_2, \varepsilon))$.

צעד האינדוקציה: נניח כי הטענה מתקיימת לכל $|w| = n$, נוכיח עבור $|w| = n + 1$.

נציג את הפירוק של w כך ש- $w = t \cdot \sigma$ כאשר $t \in \Sigma^n$ וגם $\sigma \in \Sigma$ כאשר $|\sigma| = 1$.

$$\begin{aligned} \delta_A((q_{A_1}, q_{A_2}), w) &= \delta_A((q_{A_1}, q_{A_2}), t \cdot \sigma) = [5] \delta_A(\delta_A((q_{A_1}, q_{A_2}), t), \sigma) = \\ &= [6] \delta_A((\delta_{A_1}(q_{A_1}, t), \delta_{A_2}(q_{A_2}, t)), \sigma) = [7] \delta_{A_1}(\delta_{A_1}(q_{A_1}, t), \sigma), \delta_{A_2}(\delta_{A_2}(q_{A_2}, t), \sigma) = \\ &= [8] \delta_{A_1}(q_{A_1}, w), \delta_{A_2}(q_{A_2}, w) \end{aligned}$$

מ.ש.ל טענת העזר.

נוכיח כי $L(A) = L_1 \cap L_2$.

בהינתן טענת העזר נקבל:

$$\begin{aligned} w \in L(A) [9] &\Leftrightarrow \hat{\delta}(q_0, w) \in F_A [10] \Leftrightarrow \hat{\delta}((q_{0_1}, q_{0_2}), w) \in F_1 \times F_2 \Leftrightarrow \\ &\Leftrightarrow [11] (\hat{\delta}_1(q_{0_1}, w), \hat{\delta}_2(q_{0_2}, w)) \in F_1 \times F_2 [12] \Leftrightarrow \\ &\Leftrightarrow \hat{\delta}_1(q_{0_1}, w) \in F_1 \wedge \hat{\delta}_2(q_{0_2}, w) \in F_2 \Leftrightarrow \\ &\Leftrightarrow w \in L_1 \wedge w \in L_2 \Leftrightarrow \end{aligned}$$

$$\Leftrightarrow w \in L_1 \cap L_2$$

I

[0] הסבר: אני לא יכול בכלל ליישם את דרך החשיבה הסדרתית הרגילה שלי כאן. אני רוצה לבנות עכשיו אוטומט שמבצע הרצה במקביל של האוטומטים עבור L_1 ושל L_2 . כל מצב באוטומט הוא זוג סדור של מצבים של האוטומט הראשון ושל האוטומט השני. כלומר אוסף כל הזוגות הסדורים מכפלה קרטזית של שתי קבוצות. האוטומט שנבנה נקרא **אוטומט המכפלה**. ההוכחה ארוכה, ונבנית בכמה בשלבים.

[1] זאת אומרת, כל מצב שאני בונה באוטומט המכפלה הוא בעצם זוג של מצב של A_1 ושל מצב של A_2 .

[2] כלומר מצב מקבל של האוטומט החדש הוא מצב שבו 2 האוטומטים נמצאים כרגע במצב מקבל.

[3] עד עכשיו לא הוכחנו את האוטומט רק הגדרנו אותו, צריך להוכיח שהאוטומט באמת עובד.

[4] עד עכשיו הגדרנו איך פונקצית המעברים מתנהגת עבור אות אחת בלבד אבל אני רוצה להוכיח שזה מתקיים גם עבור מילים כדי לפשט את ההוכחה.

[5] על-פי הגדרת פונקצית מעברים למילים.

[6] לפי הנחת האינדוקציה.

[7] לפי הגדרת δ .

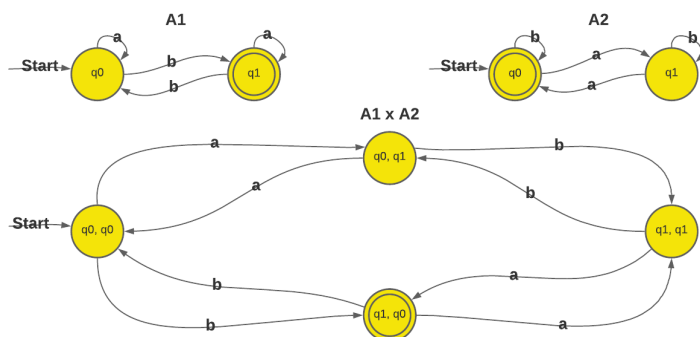
[8] הגדרת פונקצית מעברים למילים של שני האוטומטים.

[9] על-פי הגדרת שפה של אוטומט.

[10] על-פי הגדרת האוטומט.

[11] לפי טענת העזר שלנו.

[12] על-פי הגדרת מכפלה קרטזית.



איך נראה אוטומט מכפלה

נשים לב כי המצבים באוטומט המכפלה $A_1 \times A_2$ הם זוגות סדורים של המכפלה הקרטזית. יש חשיבות לסדר כלומר אם $A_1 \times A_2$ אז המצב המקבל יהיה (q_1, q_0) ולא (q_0, q_1) .

נתבונן לדוגמה במצב (q_0, q_0) , עבור A_1 במצב q_0 התו a מבצע חוג עצמי לכן יעבור ל- (q_0, q_1) ועבור התו b עובר למצב q_1 ולכן יעבור ל- (q_1, q_0) , נפעל באותו השיטה לשאר האוטומט.

משפט

אם L_1, L_2 שפות רגולריות אז $L_1 \cup L_2$ שפה רגולרית.
כלומר השפות הרגולריות סגורות תחת פעולת האיחוד.
נציג שתי הוכחות שונות לטענה זו:

הוכחה 1

לפי כלל דה-מורגן מתקיים: $\overline{\overline{L_1 \cup L_2}} = \overline{\overline{L_1} \cap \overline{L_2}} = \overline{\overline{L_1}} \cup \overline{\overline{L_2}} = L_1 \cup L_2$.
ידוע כי L_1, L_2 שפות רגולריות לכן:

☞ מסגירות שפות רגולריות למשלים נקבל ש- $\overline{L_1}, \overline{L_2}$ רגולריות.

☞ מסגירות שפות רגולריות לחיתוך נקבל ש- $\overline{L_1} \cap \overline{L_2}$ רגולריות.

☞ ושוב מסגירות שפות רגולריות למשלים נקבל ש- $\overline{\overline{L_1} \cap \overline{L_2}}$.

לכן לפי שוויון $\circledast$ נקבל כי $L_1 \cup L_2$ היא שפה רגולרית.

הוכחה 2

נשתמש בטכניקה של אוטומט המכפלה. נגדיר את האוטומט הבא: יהיו A_1, A_2 אס"ד-ים

כך ש- $L_1 = L(A_1), L_2 = L(A_2)$.

A יהיה זהה לאוטומט המכפלה של A_1, A_2 מלבד הגדרת F קבוצה:

$$\begin{aligned} F &= \{ (q_1, q_2) : q_1 \in F_1 \vee q_2 \in F_2 \} = \\ &= (F_1 \times Q_2) \vee (Q_1 \times F_2) \end{aligned}$$

נוכיח כי $L(A) = L_1 \cup L_2$:

$$\begin{aligned} w \in L(A) &\Leftrightarrow \\ &\Leftrightarrow \hat{\delta}(q_0, w) \in F \Leftrightarrow \\ &\Leftrightarrow \hat{\delta}((q_{0_1}, q_{0_2}), w) \in F \Leftrightarrow \\ &\Leftrightarrow [1](\hat{\delta}_1(q_{0_1}, w), \hat{\delta}_2(q_{0_2}, w)) \in F \Leftrightarrow \\ &\Leftrightarrow (\hat{\delta}_1(q_{0_1}, w), \hat{\delta}_2(q_{0_2}, w)) \in F_1 \times Q_2 \vee (\hat{\delta}_1(q_{0_1}, w), \hat{\delta}_2(q_{0_2}, w)) \in Q_1 \times F_2 \Leftrightarrow \\ &\Leftrightarrow \hat{\delta}_1(q_{0_1}, w) \in F_1, \hat{\delta}_2(q_{0_2}, w) \in Q_2 \vee \hat{\delta}_1(q_{0_1}, w) \in Q_1, \hat{\delta}_2(q_{0_2}, w) \in F_2 \Leftrightarrow [2] \end{aligned}$$

$$\begin{aligned}
&\Leftrightarrow \hat{\delta}_1(q_{0_1}, w) \in F_1 \vee \hat{\delta}_2(q_{0_2}, w) \in F_2 \Leftrightarrow \\
&\Leftrightarrow w \in L(A_1) = L_1 \vee w \in L(A_2) = L_2 \Leftrightarrow \\
&\Leftrightarrow w \in L_1 \cup L_2
\end{aligned}$$

I

[1] נשתמש בטענת העזר שהוכחנו בטענת סגירות לחיתוך שעדיין נכונה גם לטענה זו, אנו יכולים להשתמש בה גם פה כי בעצם כשהוכחנו אותה היא הסתמכה אך ורק על פונקצית המעברים של האוטומט לא היה שם שום שימוש בקבוצת המצבים המקבלים ובחיתוך שלה.

[2] ה-Q-ים לא אומרים לנו הרבה, הם לא נותנים לנו אינפורמציה חדשה לכן אין צורך לציין אותם יותר בהמשך ההוכחה.

אוטומט סופי לא-דטרמיניסטי (NFA)

המשמעות לאי-דטרמיניזם היא שעבור מילה כלשהי $w \in \Sigma^*$ שיכול להיות יותר ממסלול חישוב אחד של האוטומט על המילה w ; או לפעמים, לא קיים מסלול בכלל והאוטומט "נתקע" ללא יכולת לקרוא את כל הקלט. מעבר ממצב ואות נתונה יכול להיות לקבוצת מצבים.

ייתכן במיוחד, שעבור מילה כלשהי $w \in \Sigma^*$ יהיו חישובים המגיעים למצב מקבל, ואילו חישובים אחרים אינם מגיעים למצב מקבל כלל.

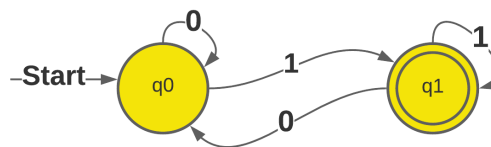
בתיאור הגרפי של אוטומט מעל Σ מתבטא האי-דטרמיניזם בכך שיש כמה קשתות היוצאות מאותו המצב ומעליהן מופיעה אותה אות. מאידך, קיימות אותיות מעל Σ שעבורן לא יוצאות קשתות מחלק מהמצבים.

יש מקרים בהם נוכל לצמצם את מספר המצבים מ-DFA ל-NFA.

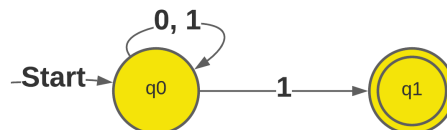
אם קיים מסלול חישוב המוביל למטרה רצויה, נתיר לאוטומט "לנחש" באופן מקומי את הצעד הבא מבין כל הצעדים הבאים האפשריים. באופן אינטואיטיבי אוטומט זה תמיד "מנחש נכון". כדי לראות את היתרון ביכולת "לנחש" נתבונן בדוגמה הבאה:

$$L = \{w \cdot 1 : w \in \{0, 1\}^*\}$$

עבור אוטומט DFA



עבור אוטומט NFA



נתבונן באוטומט NFA, כשהאוטומט נמצא במצב q_0 וקורא '1' בקלט ייתכנו 2 ניחושים:
 אם הניחוש הוא שזוהי האות האחרונה במילה, האוטומט עובר למצב המקבל q_1 .
 אחרת, האוטומט נשאר ב- q_0 .
 חשוב לשים לב שאם האוטומט מגיע במהלך החישוב ל- q_1 וזה עדיין לא סוף הקלט, האוטומט "נתקע" מאחר שאין לו מסעי המשך והחישוב לא מסתיים בקבלת המילה.

הגדרה פורמלית

אוטומט סופי לא-דטרמיניסטי A נתון על-ידי $A = (\Sigma, Q, q_0, F, \delta)$ כאשר Σ, Q, q_0, F כמו עבור האוטומט הסופי הדטרמיניסטי, ו- δ פונקציית המעברים תוגדר באופן הבא:

- בקריאת אות: $Q \times \Sigma \rightarrow 2^Q$: δ כלומר יכול להיות מצב שבו שלחנו יותר מפעם אחת אות ספציפית ולכן נקבל את קבוצת המצבים שהגענו אליהם לאחר קריאת האות.
 באופן כללי, $\delta(q_0, \sigma) = \{q_1, \dots, q_i\}$.
- בקריאת מילה: $Q \times \Sigma^* \rightarrow 2^Q$: $\hat{\delta}$ באותו האופן עבור קריאת מילה.

הגדרה

אם $w = u \cdot v$ אזי **פונקציית המעברים המורחבת** מבצעת את הפעולה הבאה בקריאת המילה w :

$$\hat{\delta}(q, w) = \hat{\delta}(q, u \cdot v) = \bigcup_{p \in \hat{\delta}(q, u)} \hat{\delta}(p, v)$$

כאשר P היא קבוצת המצבים אשר קיבלנו לאחר קריאת u .
 נבדוק מי מהקבוצות הללו קורא את הסיפא v ונקבל את קבוצת המצבים אשר סיימו לקרוא את w .

הגדרה

$\hat{\delta}$, **ההרחבה של δ** לקבוצת המצבים $2^Q \times \Sigma^* \rightarrow 2^Q$, מוגדרת כך:
 לכל $w \in \Sigma^*, P \subseteq Q$:

$$\hat{\hat{\delta}}(P, w) = \bigcup_{q \in P} \hat{\delta}(q, w)$$

(על-פי הסכם, הערך הוא $\emptyset$ במקרה ש- $P = \emptyset$).

הסבר הגדרה

$\hat{\hat{\delta}}$ עוקבת בבת-אחת אחרי כל תוצאות החישובים על המילה w מכל המצבים הנמצאים בקבוצת המצבים P .

הגדרה

$L(A)$, **השפה** המתקבלת על-ידי האוטומט הסופי הלא-דטרמיניסטי A , מוגדרת כך:

$$L(A) = \{ w \in \Sigma^* : \hat{\delta}(q_0, w) \cap F \neq \emptyset \}$$

משמעות ההגדרה היא כי השפה המתקבלת על-ידי אוטומט סופי **לא**-דטרמיניסטי היא אוסף כל המילים שעבורן קיים חישוב המסתיים **במצב מקבל**.

מילה w מתקבלת אם $\delta(q_0, w)$ מכיל לפחות מצב מקבל אחד, כלומר, קיים מסלול חישוב על w המתחיל ב- q_0 ומסתיים במצב מקבל.

שאלה

הוכח כי δ והרחבותיה $\hat{\delta}$ ו- $\hat{\hat{\delta}}$ מזדהות עבור זוגות (q, σ) כאשר $q \in Q$, $\sigma \in \Sigma$ ומזהים את $\{q\}$ עם q .

הוכחה

א. נראה כי ההרחבה $\hat{\delta}$ מזדהה עם δ עבור זוגות (q, σ) , כאשר $q \in Q$, $\sigma \in \Sigma$.

$$\begin{aligned} \hat{\delta}(q, \sigma) &= \\ &= \hat{\delta}(q, \varepsilon \cdot \sigma) = \\ &= \bigcup_{q' \in \hat{\delta}(q, \varepsilon)} \delta(q', \sigma) = \\ &= \bigcup_{q' \in \{q\}} \delta(q', \sigma) = \\ &= \delta(q, \sigma) \end{aligned}$$

ב. נראה כי ההרחבה $\hat{\hat{\delta}}$ מזדהה עם δ עבור זוגות (q, σ) כאשר $q \in Q$, $\sigma \in \Sigma$.

$$\begin{aligned} \hat{\hat{\delta}}(q, \sigma) &= \\ &= \hat{\hat{\delta}}(\{q\}, \sigma) = \\ &= \bigcup_{q' \in \{q\}} \hat{\delta}(q', \sigma) = \\ &= \hat{\delta}(q, \sigma) \end{aligned}$$

$$= \delta(q, \sigma)$$

שאלה

הוכח כי עבור NFA (בדומה ל-DFA) מתקיימת הטענה הזו:

לכל $q \in Q$ ולכל $x, y \in \Sigma^*$ מתקיים לפי ההרחבה:

$$\delta(q, x \cdot y) = \delta(\delta(q, x), y)$$

הוכחה

נוכיח את הטענה באינדוקציה על האורך של y כלומר $|y|$.

👉 **בסיס האינדוקציה:**

עבור $|y| = 0$ כלומר עבור $y = \varepsilon$, ועבור $q \in Q$, $x \in \Sigma^*$ על-פי ההרחבה $\hat{\delta}$ מתקיים:

$$\delta(q, x \cdot \varepsilon) = \delta(q, x) = \delta(\delta(q, x), \varepsilon)$$

👉 **הנחת האינדוקציה:**

נניח שהטענה נכונה לכל $y \in \Sigma^*$ ככל $|y| = n$.

👉 **צעד האינדוקציה:**

יהי $y \in \Sigma^*$ כך ש- $|y| = n + 1$.

נציג את הפירוק של y בצורה הבאה: $y = t \cdot \sigma$,

כאשר $t \in \Sigma^*$ וגם $|t| = n$ וכאשר $\sigma \in \Sigma$ כלומר $|\sigma| = 1$.

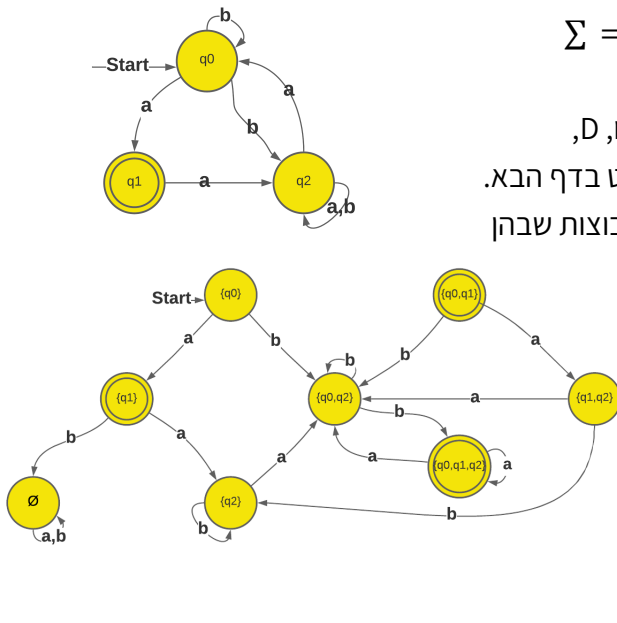
מכאן:

$$\begin{aligned} \delta(q, x \cdot y) &= \\ &= \delta(q, x \cdot t \cdot \sigma) = \\ &= \delta(\delta(q, x \cdot t), \sigma) = && \text{על-פי ההרחבה } \hat{\delta} \\ &= \delta(\delta(\delta(q, x), t), \sigma) = && \text{על-פי הנחת האינדוקציה} \\ &= \delta(\delta(q, x), t \cdot \sigma) = && \text{על-פי ההרחבה } \hat{\delta} \\ &= \delta(\delta(q, x), y) = \end{aligned}$$

אוטומט-החזקה

האי-דטרמיניזם אינו מגדיל את הכוח החישובי של האוטומט הסופי. נראה זאת על-ידי בניית אוטומט דטרמיניסטי (DFA) ששקול לאוטומט לא-דטרמיניסטי נתון. **האוטומט השקול** שנבנה נקרא **אוטומט-החזקה**, מאחר שמצביו הם איברי קבוצת החזקה של קבוצת מצבי האוטומט הלא-דטרמיניסטי. הרעיון בבנייה הוא לעקוב בבת-אחת אחר כל החישובים של האוטומט הלא-דטרמיניסטי, ולזכור כמצבים את אוספי המצבים המקוריים האפשריים במהלך כל אחד מהחישובים של האוטומט הלא-דטרמיניסטי. מאחר שמספר התת-קבוצות האפשריות הוא גם כן סופי, הרי אנו עדיין בגבולות הזיכרון החסום.

דוגמה - בניית אוטומט חזקה

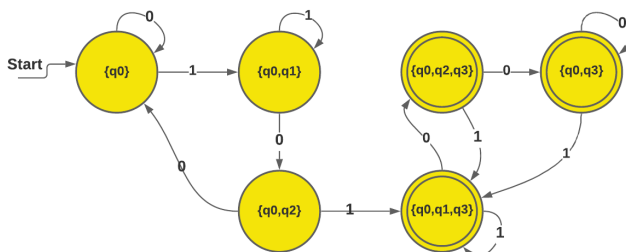


נציג כאן אוטומט סופי לא-דטרמיניסטי N מעל $\Sigma = \{a, b\}$ נבנה את האוטומט הסופי הדטרמיניסטי השקול ל- N . ל- D יש שלושה מצבים, ולכן לאוטומט החזקה השקול לו, D , יוגדר על-פי המעברים של N , כמתואר בהוכחת המשפט בדף הבא. המצבים המקבלים של D , על-פי ההגדרה, הם התת-קבוצות שבהן מופיע q_1 , שהוא המצב המקבל היחיד של N .

אוטומט החזקה המתקבל הוא: נשים לב, כי אי אפשר להגיע מ- q_0 אל המצבים $\{q_0, q_1\}$, $\{q_1, q_2\}$, ולמעשה ניתן להשמיט אותם מן הבנייה, מבלי לשנות את השפה המתקבלת. עדיף להתחיל מ- $\{q_0\}$ ולהפעיל את הבנייה רק על מצבים שבאמת נוצרים.

דוגמה נוספת לבניית אוטומט חזקה

$$L = \{w \in \{0, 1\}^* : w \text{ contains } 101\}$$



סד"פ לאוטומט חזקה

נמיר לאס"ד באופן הבא:

1. עבור כל מצב נבדוק אילו מצבים ניתן להגיע באמצעות קריאת מילה.
2. נשרטט את האוטומט.
3. במידה וישנו מצב שהוא תת-קבוצה נבדוק באילו מצבים ניתן להגיע עבור כל איבר בתת הקבוצה ונבצע איחוד ולשם נכווין את "החץ" (זוהי פונקצית המעברים).

נראה **שקילות** של NFA עם DFA בעזרת בניית אוטומט חזקה:

הוכחת השקילות עבור אוטומט חזקה

אנו הראנו שפונקציות המעברים מוגדרת באופן הבא: $\delta_N(q_0, \sigma) = \delta_D(\{q_0\}, \sigma)$. כלומר δ_D מציינת את פונקציות המעברים באוטומט החזקה בה כל מצב הוא קבוצת המצבים אליה ניתן להגיע לאחר קריאת σ מהאוטומט האי-דטרמיניסטי.

נוכיח באינדוקציה על **אורך** המילה $|w|$ ש: $\delta_N(q_0, w) = \delta_D(\{q_0\}, w)$.

מקרה בסיס: כאשר $|w| = 0$ כלומר $w = \epsilon$ נקבל לפי פונקציית המעברים:

$$\delta_N(q_0, \epsilon) = q_0 = \delta_D(\{q_0\}, \epsilon) = \{q_0\}$$

לפי הגדרת המצב ההתחלתי באוטומט החזקה ניתן לראות שהטענה נכונה עבור מקרה הבסיס מאחר ו-
 $q_0 = \{q_0\}$

הנחת האינדוקציה: נניח שהטענה נכונה עבור $|w| = n$ ונוכיח עבור $|w| = n + 1$.

צעד האינדוקציה: תהא $w = t \cdot \sigma$ כאשר $|t| = n$ ו- $|t| = 1$ כך ש- $\sigma \in \Sigma$ ו- $t \in \Sigma^*$. ולפיכך נקבל:

$$\delta_D(\{q_0\}, w) = \delta_D(\{q_0\}, t \cdot \sigma) = \delta_D(\delta_D(\{q_0\}, t), \sigma) =$$

לפי הנחת האינדוקציה אנו יודעים ש- $\delta_D(\{q_0\}, t) = \delta_N(q_0, t)$ ולפיכך:

$$= \delta_D(\delta_N(q_0, t), \sigma) = \delta_N(\delta_N(q_0, t), \sigma) =$$

$$= \delta_N(q_0, t \cdot \sigma) = \delta_N(q_0, w)$$

$$\delta_D(\{q_0\}, w) = \delta_N(q_0, w) \text{ ולכן}$$

I

נראה כעת ש: $L(N) = L(D)$ כלומר שתיהם מקבלים את אותה השפה בהכלה דו-כיוונית:

$$\begin{aligned} w \in L(D) &\Leftrightarrow \delta_D(\{q_0\}, w) \in F_D^{[1]} \Leftrightarrow \delta_D(\{q_0\}, w)^{[2]} \cap F_N \neq \emptyset \Leftrightarrow \\ &\Leftrightarrow \delta_N(q_0, w) \cap F_N \neq \emptyset^{[3]} \Leftrightarrow w \in L(N) \end{aligned}$$

- [1] - הכוונה F_D קבוצת תתי-הקבוצות המכילות איבר מ-F.
 [2] - נשים לב כי מצבים ב- F_D שייכים ל- F_N .
 [3] - על-פי הגדרת אסל"ד זוהי השפה המתקבלת ע"י האוטומט.

משפט

נאמר ששני אוטומטים הם **שקולים** אם הם מקבלים את אותה השפה.

טענה

לכל אוטומט סופי לא-דטרמיניסטי **קיים** אוטומט סופי דטרמיניסטי **השקול לו**.

הוכחה

נראה שהאוטומטים הללו שקולים בעזרת בניית אוטומט חזקה:
 $\Leftarrow$ **כיוון 1:** המרת אס"ד לאסל"ד:
 ניתן להפוך כל DFA ל-NFA שיקבל את אותה השפה באופן הבא:
 אם קראנו אות ב-אס"ד: $p = \delta_D(q, \sigma)$ הגענו למצב p כלשהו ולפיכך נגדיר:
 $\delta_N(q, \sigma) = \{p\}$ כלומר מאחר ולפונקציית המעברים עבור אסל"ד הגדרנו שבקריאת אות נקבל קבוצה של מצבים, אז קבוצת המצבים תהא המצב אליו הגענו מקריאת אות ב-DFA.
 $\Rightarrow$ **כיוון 2:** המרת אסל"ד לאס"ד:
 ניתן להפוך כל NFA ל-DFA שיקבל את אותה השפה באופן הבא:
 נבנה אוטומט חזקה אשר הוא יתאר את המעבר מ-NFA ל-DFA.
 קבוצת המצבים שלו היא: 2^Q .
 המצב ההתחלתי שלו הוא: $\{q_0\}$.
 האלפבית שלו היא: Σ .
 קבוצת המצבים המקבלים שלו היא: כל תתי-הקבוצות של Q אשר מכילות איבר מ-F.
 פונקציית המעברים שלו היא: $\delta_D(\{q_i, \dots, q_k\}, \sigma) = \{q_i, \dots, q_k\}$ שניתן להגיע אליהם בקריאת האות σ .

I

סיכום ביניים עבור אסל"ד

נגדיר את פונקציית המעברים המורחבת:

$$\hat{\delta}: Q \times \Sigma^* \rightarrow 2^Q$$

$$\hat{\delta}(q, \varepsilon) \stackrel{\text{def}}{=} \{q\}$$

נרחיב את $\hat{\delta}$:

$$\hat{\delta}: 2^Q \times \Sigma^* \rightarrow 2^Q$$

$$\hat{\delta}(p, w) \stackrel{\text{def}}{=} \bigcup_{q \in p} \hat{\delta}(q, w)$$

השפה של אוטומט אסל"ד מוגדרת כך: $L(A) = \{w \in \Sigma^* : \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$

אבחנות בסיסיות:

$$\hat{\delta}(\emptyset, w) = \emptyset$$

$$\hat{\delta}(\{q\}, w) = \hat{\delta}(q, w)$$

$$\hat{\delta}\left(\bigcup_{i=1}^n p_i, w\right) = \bigcup_{i=1}^n \hat{\delta}(p_i, w)$$

אסל"ד עם מסעי אפסילון (ENFA)

לאחר שהכרנו מודלים חישוביים כמו אס"ד ואסל"ד, ישנו מודל חדש שהוא **אוטומט סופי אי-דטרמיניסטי עם מסעי אפסילון**.

למודל זה יש תכונה מיוחדת - ניתן לעבור ממצב אחד למצב אחר **מבלי לקרוא שום מילה מהא"ב**.
היתרונות במודל זה הוא שבעזרת מעברי האפסילון ניתן לבנות אוטומטים אי-דטרמיניסטים יותר **פשוטים**.
למשל בעזרת מודל זה ניתן ליצור אוטומט יותר מופשט ממכפלה, כלומר בעזרת מסעי האפסילון ניתן **לאחד** בין שני אוטומטים המתארים שפות שונות, ובעזרתו ניתן לתאר את כל תכונות הסגור של שפות רגולריות: איחוד, חיתוך, איטרציה וכו'...

מאחר ובאוטומט הנ"ל ישנה הגדרה לפיה ניתן לעבור ממצב למצב מבלי לקרוא אפילו אות אחת, נגדיר תכונה נוספת שבעזרתה נבנה את פונקציית המעברים עבור אוטומט זה:

• נסמן '**סגור אפסילון**' באופן הבא: $CL^\varepsilon(q)$.

על-פי ההגדרה: $\delta: Q \times \{\{\varepsilon\} \cup \Sigma\} \rightarrow 2^Q$, כלומר איחדנו את השפה עם הסימון ε .
במילים אחרות, כאשר אנו מפעילים סגור ε על מצב, אנו בעצם מקבלים את תתי-הקבוצות שאלהן ניתן להגיע בקריאה של 0 או יותר מסעי אפסילון (תמיד המצב עצמו נכלל בקבוצה).

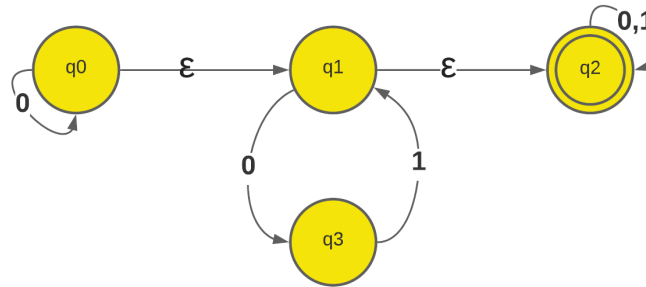
פונקציית המעברים: $\delta(q, \varepsilon) = CL^\varepsilon(q)$

ובעבור מילים: $\hat{\delta}(q, w \cdot \sigma) = \bigcup_{p \in \hat{\delta}(q, w)} CL^\epsilon(\delta(p, \sigma))$ כלומר בקריאת המילה נאחד את כל הקבוצות

עבורן ניתן להגיע ע"י קריאת w (בדיוק כמו באסל"ד) ולאחר מכן, נפעיל את סגור אפסילון על קבוצת המצבים שקראה את סאחרי w .

דוגמה

ניתן לראות בבירור שזהו NFA אך ישנם מעברים "ריקים" ואלו הם מעברי אפסילון.
נראה לדוגמה איך עובר סוגר אפסילון על המצב ההתחלתי: $CL^\epsilon(q_0) = \{q_0, q_1, q_2\}$



אסל"ד NFA לבין

שקילות בין

אסל"ד עם מסעי אפסילון ENFA

נראה שהאוטומטים הללו שקולים זה לזה:

← **כיוון 1:** המרת NFA ל-ENFA :

ניתן להמיר כל אסל"ד לאסל"ד עם מסעי אפסילון על-ידי כך שפשוט נוסיף מסעי אפסילון לאוטומט.
לכן הכיוון הנ"ל טריביאלי.

⇒ **כיוון 2:** המרת ENFA ל-NFA :

ניתן להפוך כל ENFA ל-NFA ע"י סילוק מסעי האפסילון. נעשה זאת ע"י כך שניצור אוטומט NFA חדש באופן הבא:

קבוצת המצבים של ה-NFA תהא אותה קבוצת המצבים כמו של ENFA.
המצב ההתחלתי יהיה זהה בשניהם.

פונקציית המעברים של ה-NFA תוגדר באופן הבא: $\delta(q, a) = \bigcup_{p \in CL^\epsilon(q)} CL^\epsilon(\delta(p, a))$

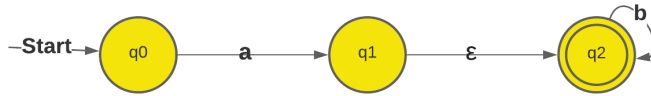
1. כלומר נבצע בדיקה עבור אילו מצבים ניתן להגיע בקריאת האות a עם 0 מעברי אפסילון או יותר.
 2. לאחר מכן, נחשב את הסוגר אפסילון עבור כל איבר בקבוצה הנ"ל ונאחד ביניהם.
 3. לכל איבר מהקבוצה הנ"ל, המצב q ישלח את a וכן הלאה...
- קבוצת המצבים שלו תלויה בכך:

- אם המילה הריקה אינה שייכת לשפת ה-ENFA אז קבוצת המצבים המקבלים שלהם שוות.

- אם המילה הריקה **שייכת** לשפת ה-ENFA אז נגדיר את המצב ההתחלתי במודל החדש NFA כמצב מקבל גם כן.

דוגמה

אסל"ד עם מסעי אפסילון:



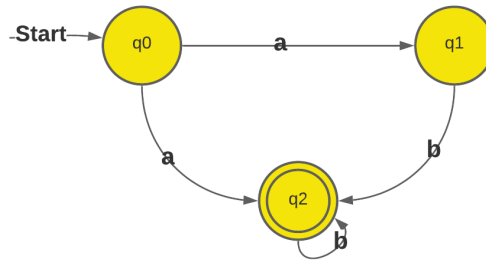
1. נשלח את המצבים לפי פונקצית המעברים:

עבור המצב q_0 : $CL^\epsilon(\delta(q_0, b)) = CL^\epsilon(\emptyset) = \emptyset$, $CL^\epsilon(\delta(q_0, a)) = CL^\epsilon(q_1) = \{q_1, q_2\}$

עבור המצב q_1 : $CL^\epsilon(\delta(q_1, b)) = CL^\epsilon(q_2) = \{q_2\}$, $CL^\epsilon(\delta(q_1, a)) = CL^\epsilon(\emptyset) = \emptyset$

עבור המצב q_2 : $CL^\epsilon(\delta(q_2, b)) = CL^\epsilon(q_2) = \{q_2\}$, $CL^\epsilon(\delta(q_2, a)) = CL^\epsilon(\emptyset) = \emptyset$

2. נשלח מכל מצב אות לעבר התת-קבוצה שקיבלנו בפונקצית המעברים:



משפט

אם A_E הוא אסל"ד עם מסעי

מסעי אפסילון כך שמתקיים:

$$L(A_{ND}) = L(A_E)$$

במילים אחרות: ENFA **שקול** ל-NFA.

אפסילון, אז קיים A_{ND} ללא

הוכחה

האוטומט ENFA מוגדר כך: $A_E = (Q_E, \Sigma, q_{0_E}, F_E, \delta_E)$

האוטומט NFA מוגדר כך: $A_{ND} = (Q_{ND}, \Sigma, q_{0_{ND}}, F_{ND}, \delta_{ND})$

כאשר:

$$Q_E = Q_{ND} \text{ (קבוצת המצבים יהיו אותם המצבים בדיוק).}$$

$$q_{0_E} = q_{0_{ND}} \text{ (המצב ההתחלתי יהיה אותו הדבר בדיוק).}$$

$$\delta_{ND}(q, \sigma) = \hat{\delta}_E(q, \sigma) = \bigcup_{p \in CL^\epsilon(q)} CL^\epsilon(\delta_E(p, \sigma))$$

$F_{ND} \Leftarrow$ מוגדרת באופן הבא:

- אם $\varepsilon \in L(A_E)$ אז נבצע: $F_{ND} \cup \{q_0\}$.
- אם $\varepsilon \notin L(A_E)$ אז לא נבצע כלום.

במילים אחרות, אם ב-ENFA המילה הריקה שייכת לשפה אז המצב ההתחלתי יהא מצב מקבל. לאחר שהגדרנו את פונקציית המעברים עבור האוטומט החדש, נוכיח באינדוקציה על **אורך** המילה w ונשתמש בהוכחה בטענת עזר להכלה דו-כיוונית:

$$\delta_{ND}(q, w) = \hat{\delta}_E(q, w) \text{ - טענת עזר}$$

❧ **בסיס האינדוקציה:** $|w| = 1$ כלומר $w = \sigma$: נלכן: $\delta_{ND}(q, \sigma) = \hat{\delta}_E(q, \sigma)$ ולפי הגדרת δ_{ND} השוויון מתקיים.

❧ **הנחת האינדוקציה:** נניח נכונות עבור $|w| = n$ ונוכיח עבור $|w| = n + 1$ כלומר $w = t \cdot \sigma$.

❧ **צעד האינדוקציה:**

$$\begin{aligned} \delta_{ND}(q_0, t \cdot \sigma) &= \delta_{ND}(\delta_{ND}(q_0, t), \sigma) = \delta_{ND}(\hat{\delta}_E(q_0, t), \sigma) = \\ &= \hat{\delta}_E(\hat{\delta}_E(q_0, t), \sigma) = \hat{\delta}_E(q_0, t \cdot \sigma) \end{aligned}$$

נראה ש- $L(A_{ND}) = L(A_E)$ כלומר: $\delta_{ND}(q_0, w) \cap F_{ND} \neq \emptyset \Leftrightarrow \delta_E(q_0, w) \cap F_E \neq \emptyset$

- עבור ε שהטענה נובעת מההגדרה.
- עבור $\varepsilon \neq$ שהטענה נובעת מטענת העזר.

I

תרגיל

נתבונן באוטומט הבא המקבל את השפה: $L(A) = \{0, 1\}^*$ הוכיח ש-A מקבל כל מילה בשפה.

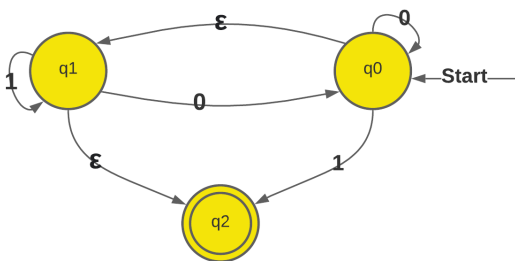
הוכחה

נוכיח באינדוקציה על **אורך** המילה.

טענה חזקה יותר: לכל $w \in \Sigma^*$ מתקיים ש-

$$F \cap \delta(q_1, w) = \{q_2\} \text{ וגם } F \cap \delta(q_0, w) = \{q_2\}$$

❧ **בסיס האינדוקציה:** $|w| = 0$ כלומר $w = \varepsilon$ מתקיים:



$$\delta(q_0, \varepsilon) = \{q_0, q_1, q_2\} \text{ ובפרט הקבוצה מכילה את } q_2.$$

$$\delta(q_1, \varepsilon) = \{q_1, q_2\} \text{ ובפרט הקבוצה מכילה את } q_2.$$

הנחת האינדוקציה: נניח נכונות בכל המילים באורך n ונוכיח עבור מילים באורך $n+1$.

צעד האינדוקציה: תהא $w = \sigma \cdot t$ כאשר $|t| = n$ וגם $\sigma \in \Sigma$, נחלק למקרים:

כאשר $\sigma = 0$ אז:

$$1. \delta(q_0, w) = \delta(q_0, 0 \cdot t) = \delta(\delta(q_0, 0), t) = \delta(\{q_0, q_1, q_2\}, t) \supseteq \delta(q_0, t)$$

ולפי הנחת האינדוקציה $F \cap \delta(q_0, t) = \{q_2\}$ ולכן $F \cap \delta(q_0, w) = \{q_2\}$.

$$2. \delta(q_1, w) = \delta(q_1, 0 \cdot t) = \delta(\delta(q_1, 0), t) = \delta(\{q_0, q_1, q_2\}, t) \supseteq \delta(q_1, t)$$

ולפי הנחת האינדוקציה $F \cap \delta(q_1, t) = \{q_2\}$ ולכן $F \cap \delta(q_1, w) = \{q_2\}$.

כאשר $\sigma = 1$ אז:

$$\delta(q_0, w) = \delta(q_1, w) = \delta(q_1, 1 \cdot t) = \delta(\delta(q_1, 1), t) = \delta(\{q_1, q_2\}, t) \supseteq \delta(q_1, t)$$

ולפי הנחת האינדוקציה היא מכילה את q_2 ולכן גם $\delta(q_1, w), \delta(q_0, w)$ מכילות את q_2 .

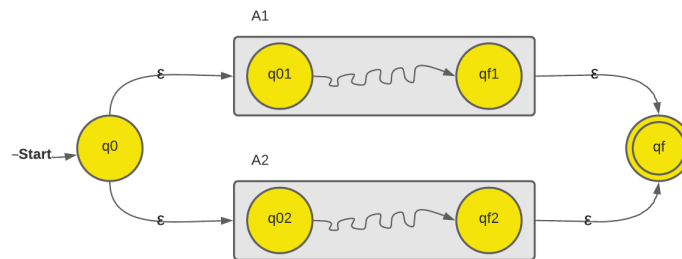
I

סגירות תחת פעולות רגולריות

לאחר שהגדרנו את ENFA נוכל לבנות בקלות אוטומטים אשר מתארים את תכונות הסגור של השפות הרגולריות. פעולות רגולריות: **איחוד, שירשור ואיטרציה**. מאחר והראינו כי ENFA שקול ל-NFA ששקול DFA, **אז כל פעולה רגולרית שמוצג בחלק זה, תקף באופן שקול וחוקי עבור כל שלושת המודלים.**

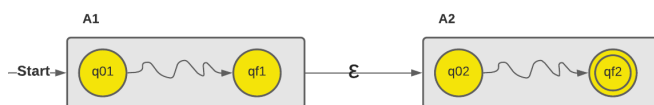
משפט

קבוצת השפות הרגולריות סגורה תחת פעולת **האיחוד**.
 יהיו L_1, L_2 שפות רגולריות מעל הא"ב, ונניח כי הן אינן ריקות.
 אז קיימים אוטומטים סופיים A_1, A_2 כך ש: $L_1 = L(A_1)$ ו- $L_2 = L(A_2)$.
 המצב המקבל היחיד הוא q_{f1} עבור A_1 ו- q_{f2} עבור A_2 כך ש- $Q_1 \cap Q_2 = \emptyset$ ומ- q_{f1}, q_{f2} אין קשתות יוצאות. ו- q_0 הם מצבים חדשים שאינם שייכים ל- $Q_1 \cup Q_2$
 נוודא שבהגדרת פונקציית המעברים עבור האוטומט הנ"ל, הפונקציה תתנהג ב- A_1 וב- A_2 בדיוק אותו הדבר כדי לא לאבד או לקבל מילים שאינם בשפה.
 בדרך כלל נגדיר אותה באופן הבא: $\forall q \in Q: q \neq q_f: \hat{\delta}(q, \sigma) = \delta_1(q, \sigma)$
 לגבי המצב המקבל המקורי נגדיר עבורו מעבר אפסילון ל- q_f .
 נבנה ENFA באופן הבא:



משפט

קבוצת השפות הרגולריות סגורה תחת פעולת **השירשור**.
 יהיו L_1, L_2 שפות רגולריות מעל הא"ב, אז קיימים אוטומטים סופיים A_1, A_2 כך ש: $L_1 = L(A_1)$ ו- $L_2 = L(A_2)$.
 נניח כי קבוצות המצבים של האוטומטים הינן זרות ולכל אחד מהאוטומטים מצב מקבל יחיד, אשר אין ממנו קשתות יוצאות.



נבנה אוטומט ENFA שיקבל $L_1 \cdot L_2$:

נוודא שבהגדרת פונקצית המעברים עבור האוטומט הנ"ל, הפונקציה תתנהג ב- A_1 וב- A_2 בדיוק אותו הדבר.

משפט

$$L^* = \bigcup_{i=0}^{\infty} L^i \text{ האטרציה}$$

הוכחה

תהי L שפה רגולרית מעל הא"ב, ונניח כי היא אינה ריקה. אזי קיים אוטומט סופי $A = (\Sigma, Q, q_0, \{q_f\}, \delta)$ המקיים $L = L(A)$, כך שמהמצב המקבל (היחיד) q_f אין קשתות יוצאות.

נגדיר אוטומט סופי A' באופן הזה:

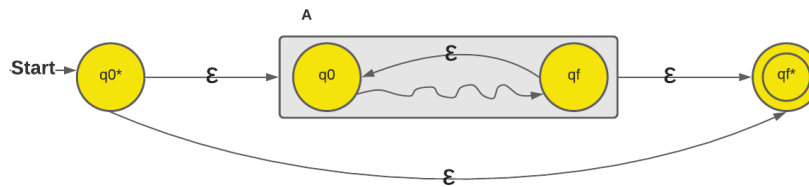
$$A' = (\Sigma, Q \cup \{q_0^*, q_f^*\}, q_0^*, \{q_f^*\}, \delta_{A'})$$

כאשר q_0^* ו- q_f^* הם מצבים חדשים שאינם שייכים ל- Q וכאשר:

$$1. \delta_{A'}(q_0^*, \varepsilon) = \delta_{A'}(q_f, \varepsilon) = \{q_0^*, q_f^*\}$$

$$2. \text{ לכל } q \in Q \setminus \{q_f\} \text{ ולכל } \sigma \in \Sigma \cup \{\varepsilon\} : \delta_{A'}(q, \sigma) = \delta(q, \sigma)$$

נוכל לתאר את A' באופן גרפי כך:



נשנה סימון $A' = A^*$, יש להוכיח כי: $L(A^*) = (L(A))^*$ בהכלה דו-כיוונית.

- **כיוון א':** ε מתקבל על-ידי האוטומט כיוון שיש מסע-אפסילון מ- q_0^* ל- q_f^* .

עבור כל מילה שהיא שרשור מספר סופי של מילים מ- L נעבור בתוך הלולאה הפנימית שבאזור האוטומט, ולבסוף נעבור עם מסע האפסילון ל- q_f^* .

כך ניתן להגדיר מסלול חישוב מקבל עבור כל $w \in L^+$. פורמלית, יש לבצע אינדוקציה על מספר השרשורים.

- **כיוון ב':** נניח ש- A^* מקבל את w ונתבונן במסלול חישוב מקבל.

מקרה א': המסלול מבצע מסע אפסילון מ- q_0^* ל- q_f^* (במקרה זה $w = \varepsilon$ ולכן $w \in L^*$).

מקרה ב': המסלול מגיע $k \geq 1$ פעמים ל- q_f ולאחר כל ביקור ב- q_f פרט לביקור האחרון מבצע מסע אפסילון ל- q_0^* . במקרה זה, ניתן לרשום $w = w_1 w_2 \dots w_k$, כאשר כל w_i מתקבלת ע"י A .

לכן במקרה זה $w \in L^*$.

I

ביטויים רגולריים

נסמן שפות באמצעות ביטויים באופן שכל שפה שניתן לציין בביטוי מתאים תהיה רגולרית. לכן מציאת ביטוי מתאים תהיה הוכחה עקיפה לרגולריות. באנגלית ביטוי רגולרית - regular expression או regex. במילים אחרות, **ביטוי רגולרי נועד לתאר שפה באופן פורמלי.**

הגדרה

אוסף הביטויים הרגולריים מעל הא"ב Σ המסומן R , הוא **הקבוצה המינימלית** המקיימת את ששת התכונות הבאות:

אטומים

(1) $\epsilon \in R$ (המילה הריקה).

(2) $\emptyset \in R$.

(3) לכל $\sigma \in \Sigma$ מתקיים $\sigma \in R$.

פעולות יצירה - אם $r_1, r_2 \in R$ אז:

(4) $r_1 + r_2 \in R$

(5) $r_1 \cdot r_2 \in R$

(6) אם $r \in R$ אז $(r)^* \in R$

הסימונים ϵ -ו- $\emptyset$ ואיברי Σ משמשים בשתי משמעויות:

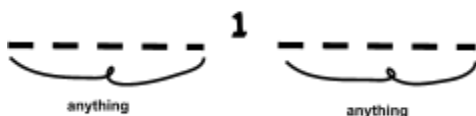
במשמעותם הרגילה כפי שאנחנו מכירים עד כה, ובמשמעותם כביטויים רגולריים.

דוגמאות בסיסיות

כעת נציג כמה דוגמאות של מה זה בכלל ביטוי רגולרי באופן הכי בסיסי שיש לפני שניגש לתיאוריה ולכל הבלאגן:

Ex1: all strings over $\Sigma = \{0, 1\}$ that contain a 1.

כדי לפתח חשיבה לתרגיל זה נחשוב בצורה הבאה, אם אנחנו מייצרים מחרוזת שהיא מכילה את התו '1', נצייר לנו בצד שרטוט



כללי ביותר עבור מחרוזת כזאת ותמיד כשנפגוש את המילה contain / מכיל וכו' בשאלות כאלה נשאל את עצמנו אילו איברים יכולים להיות לנו בין ה-1 הזה?

תמיד שיש מקרה שבו יכול להיות כל תו שנרצה בין ה-1 שלנו (anything) זה תמיד יהיה מצב של $(r_1 + r_2)^*$ ולכן הביטוי הרגולרי הזה מתואר כך:

$$regex = (0 + 1)^* 1 (0 + 1)^*$$

Ex2 : all strings having exactly two 1's.

תחילה נשרטט לנו דוגמה כללית למחרוזת כזאת כדי לסייע לפתרון השאלה הבסיסית הזאת.

בשונה מהתרגיל הקודם, יש לנו כאן הגבלה על מספר האחדים שלנו. במקרה זה, אנחנו לא יכולים להגיד שאפשר שכל תו באלפבית שלנו, יתקיים בין האחדים, מכיוון שאנחנו מוגבלים אך ורק ל-2 אחדים בלבד. לכן בשלושת החלקים המקווקים חייבים לקיים אך ורק תווים של אפסים בלבד, כמובן שלא משנה לנו כמה אפסים הם כי כל מה שמשנה לנו זה מספר האחדים. לכן הפתרון:

$$regex = 0^* 1 0^* 1 0^*$$

Ex3 : all even – length strings

במקרה כזה אפשר לתאר את הפתרון כך: $\varepsilon + 00 + 01 + 10 + 11 + \dots$

אבל זה לא דרך לכתיבה פורמלית, וחוץ מזה באופן הרבה יותר חריג, הביטוי הרגולרי שהצגנו הוא לא נכון בגלל שהוא אינסופי.

לכן אנחנו צריכים לחשוב על מה משותף לכל המקרים שהצגנו בדוגמא השגויה?

אנחנו יודעים שאם יש לנו מחרוזת באורך זוגי אז אפשר "לחלק" את המחרוזת לחלקים כך שכל אורך הוא באורך 2, אם המחרוזת הייתה אי-זוגית אז היינו "נתקעים" בסוף המחרוזת.

נסתכל על החלוקה הראשונה באופן אינדיבידואלי ואז לאפשר פעולת כוכב על החלוקה הזאת וכך נוכל להמשיך עם הפעולה הזאת על כל שאר החלוקות האחרות כמה שנרצה. כל אחד מהתווים בחלוקה הראשונה לא תלויים אחד בשני ולהפך, אז כל אחד מהם יכול להיות $(0 + 1)$ עבור התו הראשון וגם $(0 + 1)$ עבור התו השני כלומר נקבל מצב של $(0 + 1)(0 + 1)$ וככה נבטיח רק מילים באורך זוגי ולכן כל מה שנותר לעשות הוא:

$$regex = ((0 + 1)(0 + 1))^*$$

Ex4 : all strings start and end with the same character.

גם כאן אנחנו צריכים לחשוב מהם האפשרויות שמחרוזת כזאת יכולה להופיע?

מאחר ו- $\Sigma = \{0, 1\}$, אנחנו צריכים שאותו תו $\sigma \in \Sigma$

$\sigma \text{ ----- } \sigma$

יופיע בתחילת המחרוזת ובסופה, כמובן שכל מה שבין לבין יכול להיות כל תו שנרצה מתוך האלפבית שלנו. מאחר ומדובר על כל המחרוזות אנחנו צריכים לתת תשומת לב רבה שבאמת התייחסנו לכל התווים באלפבית שלנו כי $1 = \sigma$ או $0 = \sigma$ ולכן:

עבור $\sigma = 1$, נזכר בצורת החשיבה שלנו מהתרגיל הראשון בו ברגע שיש לנו אפשרות שכל תו יהיה בין התו הספציפי שלנו אנחנו נכתוב זאת בצורה הבאה $(0 + 1)^*$ ולכן עבור מקרה זה $1 (0 + 1)^*$.
עבור $\sigma = 0$ מקרה דומה, נקבל $0 (0 + 1)^*$.
ולכן הפתרון הוא:

$$regex = (1 (0 + 1)^* 1) + (0 (0 + 1)^* 0)$$

הביטויים הבאים הם ביטויים רגולרים מעל $\Sigma = \{a, b\}$:

- $\emptyset, \varepsilon, a, b$
- $(\varepsilon + b), ((\varepsilon + b) \cdot b)$
- $(\emptyset)^*, ((\varepsilon + b) \cdot (b)^*)$

הגדרה

עבור R אוסף הביטויים הרגולרים מעל Σ , נגדיר העתקה $L: R \rightarrow 2^{\Sigma^*}$,
באינדוקציית מבנה על R (במילים אחרות, נתאים לכל ביטוי רגולרי את השפה שלו):
1.

$$\begin{aligned} L[\emptyset] &= \emptyset \\ L[\varepsilon] &= \{\varepsilon\} \\ \forall \sigma \in \Sigma: L[\sigma] &= \{\sigma\} \end{aligned}$$

$$2. \text{ אם } r_1, r_2 \in R \text{ אז } r_1 + r_2 \in R$$

$$\begin{aligned} L[r_1 + r_2] &= L[r_1] \cup L[r_2] \\ L[r_1 \cdot r_2] &= L[r_1] \cdot L[r_2] \end{aligned}$$

$$3. \text{ אם } r \in R \text{ אז } r^* \in R$$

$$L[(r)^*] = (L[r])^*$$

באופן זה מותאמת לכל $r \in R$ שפה $L[r]$ והיא נקראת **השפה שמציין הביטוי הרגולרי r** .

סימונים

1. אם r ביטוי רגולרי נסמן ב- (r^+) את הביטוי הרגולרי $(r \cdot (r)^*)$.
2. נקבע סדר קדימויות כדי שאפשר יהיה להשמיט סוגריים:
* - קדימות גבוהה ביותר.
- - קדימות בינונית.

+ - קדימות נמוכה.

3. בדרך כלל נשמיט את אופרטור השרשור.

4. לעיתים נשתמש בביטוי הרגולרי לציון השפה שהוא מייצג.

הפעולות שכתובות בין סוגריים קודמות לפעולות בין האופרטורים שהצגנו ב-(2), כלומר: לסוגריים הקדימות הגבוהה ביותר.

לגבי ההשמטות נציג דוגמאות:

את הביטוי הרגולרי $((a^*) \cdot (b^*)) \cdot (c^*)$ אפשר לרשום בקיצור $a^* b^* c^*$.

את הביטוי הרגולרי $((a \cdot b)^*) \cdot (c^*)$ אפשר לרשום בקיצור $(ab)^* c^*$.

הערה

את שפת כל המילים מעל $\Sigma = \{0, 1\}$, כלומר את השפה $\{0, 1\}^*$, מציין הביטוי הרגולרי $(0 + 1)^*$. אפשר להשתמש בקיצור שנהגנו ולרשום את הביטוי הרגולרי הזה כ- Σ^* .
לכן עבור הפתרון של דוגמה 1 בדף הקודם שלנו אפשר להציג כך: $regex = \Sigma^* 1 \Sigma^*$.
ואת דוגמה 3 אפשר להציג כך: $regex = (\Sigma \Sigma)^*$.

נרצה להראות שביטויים רגולריים שקולים לאוטומטים.

משפט

לכל $r \in R$ מעל Σ , $L[r]$ היא שפה רגולרית.

הוכחה

נוכיח באינדוקציה מבנית על r .

❧ **בסיס האינדוקציה:**

$$L[\emptyset] = \emptyset$$

$$L[\varepsilon] = \{\varepsilon\}$$

$$\forall \sigma \in \Sigma: L[\sigma] = \{\sigma\}$$

❧ **צעד האינדוקציה:** נובע ישירות מסגירות השפות הרגולריות תחת הפעולות הרגולריות: איחוד, שירשור ואיטרציה (בהתאמה):

נניח כי הטענה נכונה עבור ביטויים r_1, r_2 ונוכיח כי היא נכונה עבור $r_1^*, r_1 \cdot r_2, r_1 + r_2$:

$$L[(r_1 + r_2)] = L[r_1] \cup L[r_2]$$

$$L[(r_1 \cdot r_2)] = L[r_1] \cdot L[r_2]$$

$$L[(r^*)] = (L[r])^*$$

רגולריות מסגירות לשרשור

רגולריות מסגירות לאיטרציה.

I

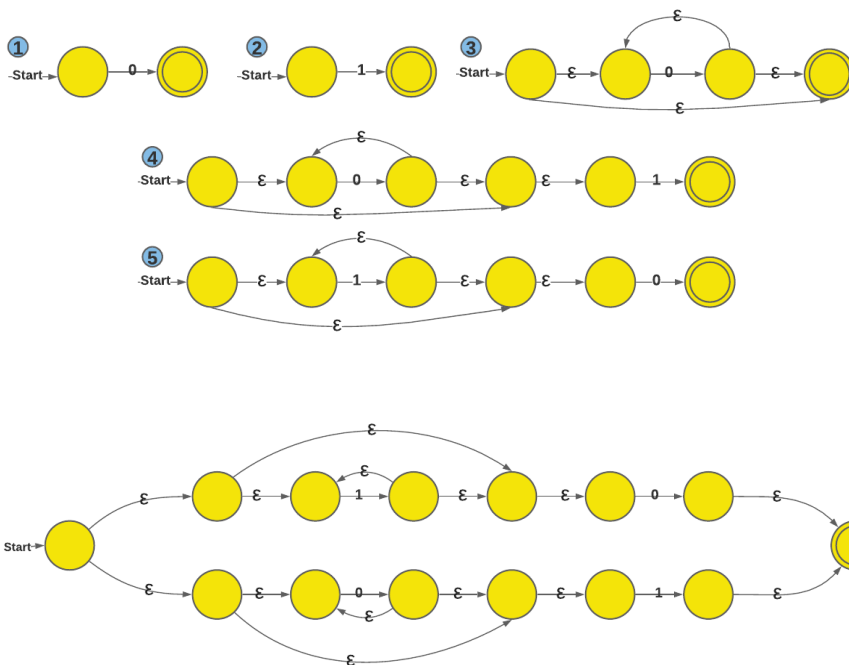
הוכחה זו משמשת דרך אפקטיבית לבניית אוטומט סופי המקבל את $L[r]$ עבור r נתון, נראה זאת בדוגמה שלפנינו:

דוגמה

נבנה אוטומט סופי המקבל את $L[r]$ כאשר $r = 0^* 1 + 1^* 0$.

נבנה בשלבים באמצעות

הבניות מהמשפטים שהוזכרו לעיל בלי לתת שמות למצבים.



עבור אוטומט:

$$r_1 = 0 \quad 1.$$

$$r_2 = 1 \quad 2.$$

$$r_3 = 0^* \quad 3.$$

$$r_4 = 0^* 1 \quad 4.$$

$$r_5 = 1^* 0 \quad 5.$$

ולבסוף $r = 0^* 1 + 1^* 0$

משפט

לכל שפה רגולרית $L \subseteq \Sigma^*$ קיים ביטוי רגולרי r כך ש- $L = L[r]$.

הוכחה

L היא רגולרית, לכן קיים DFA הבא $A = (\Sigma, \{q_1, \dots, q_m\}, q_1, \delta, F)$ כך ש- $L(A) = L$.

לכל i, j, k נסמן $L_{i,j}^k$ השפה הכוללת את המילים שמובילות את האוטומט מ- q_i ל- q_j בלי לעבור דרך מצב שמספרו גדול מ- k . (לעבור דרך = אינו כולל את המצב ממנו יוצאים והמצב אליו מגיעים).
פורמלית:

$$L_{i,j}^k = \{w : \delta(q_i, w) = q_j, \forall u, v \neq w, w = u \cdot v, \delta(q_i, u) = q_l \Rightarrow l \leq k\}$$

על-פי ההגדרה הקודמת, כיוון שאין מצב גדול מ- m הרי ש- $L_{i,j}^m$ כוללת את כל המילים המובילות את האוטומט

$$L(A) = \bigcup_{q_j \in F} L_{1,j}^m \text{, ובפרט, } q_j \text{ למצב } q_i \text{ ומצב } q_i \text{ למצב } q_j$$

נשים לב כי סימנו את המצב ההתחלתי ב- q_1 .

השפה $L(A)$ הינה איחוד של מספר סופי של שפות, לכן אם נוכל למצוא לכל $L_{i,j}^k$ ביטוי רגולרי נוכל למצוא ביטוי רגולרי עבור $L(A)$.

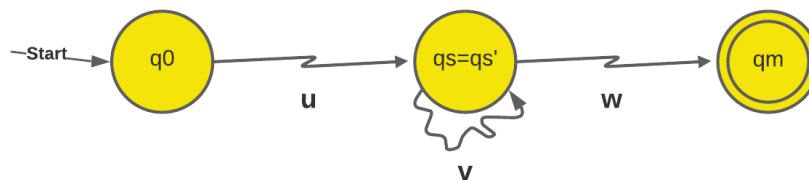
I

למת הניפוח

הדרך הנוחה להראות ששפה היא רגולרית היא בעזרת התכונות סגור שלה: איחוד, חיתוך, משלים, איטרציה, היפוך, חיסור, הכלה והפרש סימטרי. התכונות האלה מאפשרות לנו לבנות שפות רגולריות. אם נרצה להוכיח ששפה אינה רגולרית: נניח שהיא רגולרית נפעיל עליה תכונות סגור ונגיע לשפה שהיא אינה רגולרית.

עד עכשיו הכרנו את השפה הלא רגולרית: $L = \{a^n \cdot b^n : n \geq 0\}$, כעת ישנו כלי כללי יותר שיאפשר לנו להראות ששפות אינן רגולריות - והיא **למת הניפוח לשפות רגולריות**: זוהי למה **שכל** השפות הרגולריות מקיימות, ואם שפה אינה מקיימת את הלמה אז היא אינה רגולרית.

הרעיון הכללי: נקרא מילה z , נניח שזהו המסלול קריאה של המילה $z = u \cdot v \cdot w$:



נשים לב ש- v בעצם מבצע לולאה כלשהי, הלולאה הזאת יכולה להתבצע i פעמים, אך לבסוף נגיע למצב המקבל q_m , לכן, לפי פונקציית המעברים: $\hat{\delta}(q_0, u \cdot v^i \cdot w) \in F$, כלומר המילה: $z \in L$ כלומר $u \cdot v^i \cdot w \in L$.

אנו בעצם נרצה "**לנפח**" את החלק שבו ביצענו את הלולאה ואז נראה שעבור "**ניפוח**" i מסויים קיבלנו מילה **שאינה** שייכת לשפה ולפיכך השפה אינה רגולרית מאחר ולכל i הטענה צריכה להתקיים.

משפט

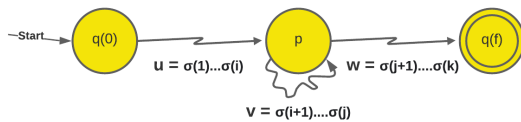
תהא L שפה רגולרית.

אז **קיים** מספר טבעי n כך **שלכל** מילה z בשפה L , שאורכה n , $|z| \geq n$, **קיים** פירוק עבורה $z = u \cdot v \cdot w$ כך ש:

$$1. |u \cdot v| \leq n$$

$$2. |v| \geq 1$$

$$3. \text{ לכל } i \geq 0 \text{ מתקיים ש: } u \cdot v^i \cdot w \in L$$

הסבר מפורט על המשפט וההוכחה**תהא L שפה רגולרית.**

ידוע כי עבור שפה רגולרית L קיים אס"ד שמקבל אותה.

אז קיים מספר טבעי n

n הוא בעצם מספר המצבים של האוטומט הזה: $|Q| = n$ (כאן באה הסופיות של האוטומט לידי ביטוי).

כך שלכל מילה z בשפה L , שאורכה n , $|z| \geq n$,

אם z היא מילה שאורכה לפחות n , אז כשנפעיל אותה על האוטומט, בכל צעד נקרא אות אחת מהמילה, האוטומט מתחיל מהמצב ההתחלתי q_0 , ובתום קריאת n האותיות - הוא ביקר בסך-הכל ב- $n+1$ מצבים. מאחר שמספר המצבים הוא $|Q| = n$, נובע מעקרון שובר היונים שקיים מצב אחד p , שבו האוטומט ביקר פעמיים (או יותר).

קיים פירוק עבורה $z = u \cdot v \cdot w$ כך ש:

כלומר z ניתנת לפירוק, ובתור:

u בוחרים את הרישא של המילה אותה קורא האוטומט עד שנגיע למצב p .

v בוחרים את המשך המילה אותה האוטומט קורא עד שהוא חוזר אל המצב p בפעם האחרונה.

w בוחרים את שאר המילה עד המצב המקבל.

$$1. |u \cdot v| \leq n$$

זה מתקיים באופן ברור כי u ו- v נבחרו שתיהן מתוך ה- n אותיות ראשונות של המילה z .

2. $|v| \geq 1$

מתקיים מכיוון שכדי לבקר פעמיים (או יותר) במצב p על האוטומט לקרוא לפחות אות אחת מהרגע בו ביקר במצב בפעם הראשונה ועד שהוא מבקר בו בפעם השניה.

3. לכל $i \geq 0$ מתקיים ש: $u \cdot v^i \cdot w \in L$

זה התנאי שמתאר את הניפוח עצמו, עבור כל מספר שכפולים (כולל 0) של קטע הניפוח, מקבלים מילה השייכת לשפה. כלומר, לא משנה כמה ננפח אותה עדיין היא תתקבל. מתקיים מכיוון שכל האותיות שהאוטומט קורא בזמן המעבר מ- p בחזרה ל- p , אין שום השפעה על הריצה של האוטומט. אפשר אפילו להשמיט את האותיות האלה, ואז האוטומט יבקר רק פעם אחת ב- p וימשיך ל- w .

אבחנה

לגבי תנאי 2: האורך של v הוא 1 מאחר ואם האורך היה 0 אז לא נתקענו באותו מצב. ומאחר ו- $|z| \geq n$ אנו חייבים להגיע לאותו מצב פעמיים.

הוכחה

L רגולרית ולכן קיים עבורה DFA כך ש: $L(A) = L$.
נסמן את מספר מצבי האוטומט באופן הבא: $|Q| = n$.
תהא L זכך שאורכה: $|z| \geq n$ נסמן $z = \sigma_1 \cdot \sigma_2 \cdot \dots \cdot \sigma_k$ ש: $k \geq n$.

נגדיר עבור האוטומט את סדרת המצבים $q_0, \dots, q_n$ ונגדיר $\hat{\delta}(q_0, \sigma_1 \cdot \sigma_2 \cdot \dots \cdot \sigma_i) = q_i$.
מעקרון שובך היונים מאחר ואורך המילה z גדול ממספר המצבים נקבל שבסדרת המצבים מופיע אותו מצב פעמיים כלומר קיים $p \in Q$ ו- $1 \leq i < j \leq n$, כך ש:

$$\begin{aligned}\hat{\delta}(q_0, \sigma_1 \cdot \dots \cdot \sigma_i) &= p \\ \hat{\delta}(q_0, \sigma_1 \cdot \dots \cdot \sigma_i \cdot \dots \cdot \sigma_j) &= p\end{aligned}$$

נגדיר את הפירוק של z באופן הבא:

$$\begin{aligned}u &= \sigma_1 \cdot \dots \cdot \sigma_i \\ v &= \sigma_{i+1} \cdot \dots \cdot \sigma_j \\ w &= \sigma_{j+1} \cdot \dots \cdot \sigma_k\end{aligned}$$

נבין בעצם איך הפירוק מתבצע על-ידי פונקצית המעברים:

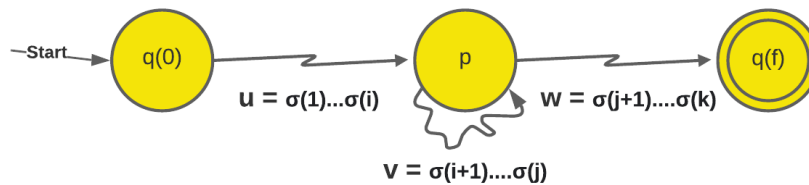
$$\hat{\delta}(q_0, u) = p = \hat{\delta}(q_0, u \cdot v) \text{ מאחר ונתקענו במצב } p \text{ אנו יודעים שנובע:}$$

נוכיח באינדוקציה ש: $\hat{\delta}(p, v^i) = p$ כלומר מתבצעת הלולאה $i \geq 0$ פעמים.
 ניתן לראות ש: $j = |u \cdot v|$ ולפי ההגדרה ראינו ש: $j \leq n$ ולכן $|u \cdot v| \leq n$.
 בנוסף מאחר ו- L זפירוקה היא: $w \cdot v \cdot u = z$ זוהרנו באינדוקציה ש: $\hat{\delta}(q_0, u \cdot v^i) = p$
 לכן שנקרא את w נגיע למצב מקבל כלומר:

$$\hat{\delta}(q_0, u \cdot v^i \cdot w) = \hat{\delta}(\hat{\delta}(q_0, u \cdot v^i), w) = \hat{\delta}(p, w) \in F$$

ולפיכך: $u \cdot v^i \cdot w \in F$ ולכן הפירוק מקיים את שלושת הדרישות.

I



להוכיח

שיטה כיצד

ששפה אינה רגולרית בעזרת למת הניפוח

הרעיון: ניקח n כלשהו, ניקח מילה z השייכת לשפה וכאשר ננפח את המילה אז נקבל מילה שאינה שייכת לשפה.

איך בעזרת למת הניפוח נוכיח ששפה היא **אינה** רגולרית? (התשובה ב- $\S$).
 נשים לב כי במשפט של הלמה, הכמתים מודגים בכוונה ויש לתת תשומת לב רבה לעניין הזה.
 תחילה, נציג שוב את הלמה לשפות רגולריות, לטובת הבנה טובה יותר של השיטה:

תהא L שפה רגולרית, אזי **קיים** $0 < n$ כך **שלכל** מילה $z \in L$ שהמקיימת שאורכה היא: $|z| \geq n$
קיים פירוק עבורה $z = u \cdot v \cdot w$ כך ש:
 1. $|u \cdot v| \leq n$
 2. $|v| \geq 1$
 3. **לכל** $i \geq 0$ מתקיים ש: $u \cdot v^i \cdot w \in L$.

$\S$ כעת, נפעיל שלילה על **כל** הכמתים בלמה:

תהא L שפה כלשהי אם **לכל** $0 < n$ **קיימת** מילה $z \in L$ כך שאורכה היא $|z| \geq n$
 כך **שלכל** פירוק עבורה $z = u \cdot v \cdot w$
 1. $|u \cdot v| \leq n$

$$2. |v| \geq 1$$

$$3. \text{קיים } i \geq 0 \text{ מתקיים ש: } w \notin L : u \cdot v^i \cdot w$$

אזי L שפה לא רגולרית.

כעת נשים לב שלא כל שפה שאינה רגולרית מקיימת את התכונה הנ"ל, ישנם שפות שהם לא רגולריות אשר ניתנות לניפוח בלבד.

הוכחת אי רגולרית ע"י שימוש בלמה

כדי להראות ששפה לא רגולרית – נראה כי היא לא מקיימת את למת הניפוח

1. נניח בשלילה כי השפה רגולרית, ויהי n הקבוע המובטח
2. נבחר מילה z כפונקציה של n
3. לכל פירוק של z ל uvw ע"פ תנאי הבלמה, נראה i שעבורו $w \cdot v^i \cdot u$ מהמילה z אינה שייכת לשפה L.
4. סתירה השפה אינה רגולרית.

שאלות של קביעת שפה רגולרית ולמת הניפוח

- איך פותרים?
 1. אם השפה רגולרית - מראים ביטוי רגולרי לשפה או מראים אוטומט.
 2. אם השפה לא רגולרית - מוכיחים באמצעות למת הניפוח.
- איך מזהים ששפה רגולרית או לא?

אם נדרשת ספירה של משהו שאינו מחזורי ואינו חסום - השפה לא רגולרית (בתנאי שהספירה לא "נבלעת" בשרשור או באיחוד).
- לפעמים השפה יכולה להראות לא רגולרית אבל היא כן, איך נדע? צריך להבין את "המבניות" של השפה.
- איפה נטעה? - לפעמים נחשוב שצריך לספור/לזכור צעדים (וזה אסור) ואז נסיק שהיא לא רגולרית. אבל יכול להיות שיש "מבניות" מסוימת שחוזרת על עצמה שמקיימת את "הספירה" מבלי "לזכור" כלומר השפה היא כן רגולרית. - זה דורש ניסיון בפתרון תרגילים בשביל להבין את זה.
- מומלץ לכתוב בפתרון לפני ההוכחה "אינטואיציה" בה נפרט למה הסקנו שהשפה היא רגולרית או לא רגולרית.

שאלה ממבחן מועד א' 2020 - פתרון של גיל לוי

לכל אחת מהשפות הבאות קבעו האם השפה רגולרית או לא. עליכם להוכיח את תשובתכם, אולם אם אתם בונים אוטומט או ביטוי רגולרי, אין צורך להוכיח את נכונות הבניה.
א. (9 נקודות) $L_1 = \{w \mid w \in \{0,1\}^*, \exists n \in \mathbb{N} \text{ such that } |w| = n^3\}$

פתרון א'

אינטואיציה: ישנה ספירה מדויקת של חזקה שלישית שהיא לא מחזורית ולא חסומה. לכן השפה **לא** רגולרית.
הוכחה: נניח בשלילה שהשפה L_1 רגולרית ולכן מקיימת את למת הניפוח.

יהי $n \in \mathbb{N}$ הקבוע המובטח מהלמה.

נבחר: $z = 0^{n^3} \in L_1$ ומתקיים:

$$|z| = n^3 \geq n$$

יהי $z = uvw$ פירוק כלשהו המקיים:

$$|uv| \leq n \text{ וגם } |v| \geq 1$$

נבחר: $i = 2$ ומכאן:

$$z' = 0^{n^3 + |v|} \notin L_1$$

כי:

$$n^3 < n^3 + |v| \leq n^3 + n < (n + 1)^3$$

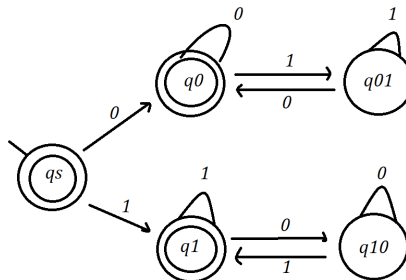
סתירה. ולכן השפה לא רגולרית.

ב. (9 נקודות) $L_2 = \{w \mid w \in \{0,1\}^*, \#_{01}(w) = \#_{10}(w)\}$, כלומר, L_2 היא שפת כל המילים מעל $\{0,1\}^*$ שבהן מספר המופעים של 01 שווה למספר המופעים של 10 (כולל מופעים חופפים).
לדוגמא במילה 010 יש מופע אחד של 01 ומופע אחד של 10.

פתרון ב'

אינטואיציה: לאורך כל המילה כמות ה-01 עולה בכל מעבר מ-0 ל-1 וכמות ה-10 עולה בכל מעבר מ-1 ל-0.
לכן, הכמויות עולות לסירוגין. דוגמא: 01010000011110. לכן השפה **כן** רגולרית.

הוכחה: נצייר אוטומט:



ג. (9 נקודות) $L_3 = \{w \mid w \in \{a,b,c\}^*, \exists \sigma, \sigma' \in \{a,b,c\} \text{ such that } \#_{\sigma}(w) \neq \#_{\sigma'}(w)\}$
כאשר $\#_{\sigma}(w)$ הוא מספר המופעים של האות σ במילה w . כלומר, L_3 היא שפת כל המילים מעל $\{a,b,c\}^*$ שבהן קיימות שתי אותיות שמספר המופעים שלהן במילה שונה.

פתרון ג'

אינטואיציה: חייבים למנות כמה פעמים קיבלנו כל תו כדי לדעת האם יש שני תווים שהם בכמות שונה וספירה זאת אינה חסומה. לכן השפה **לא** רגולרית.

הוכחה:

נניח בשלילה ש L_3 רגולרית ויהי $n \in N$ הקבוע המובטח מהלמה.

נבחר: $z = a^n b^{n+n!} c^{n+n!} \in L_3$ ומתקיים:

$$|z| \geq n + n + n! + n + n! = 3n + 2n! \geq n$$

יהא מילה z מפירוק כלשהו $z = uvw$ מהמקיים:

$$|uv| \leq n \text{ וגם } |v| \geq 1$$

$$z' = a^k \cdot a^{ti} \cdot a^{n-k-t} \cdot b^{n+n!} \cdot c^{n+n!}$$

נשים לב ש: $k + t \leq n$ ובנוסף $1 \leq t$.

$$z' = a^{n+t(i-1)} \cdot b^{n+n!} \cdot c^{n+n!}$$

נבחר: $i = \frac{n!}{t} + 1$ ונקבל:

$$\begin{aligned} z' &= a^{n+t \cdot (i-1)} \cdot b^{n+n!} \cdot c^{n+n!} = \\ &= a^{n+t \cdot \frac{n!}{t}} \cdot b^{n+n!} \cdot c^{n+n!} = \\ &= a^{n+n!} \cdot b^{n+n!} \cdot c^{n+n!} \notin L_3 \end{aligned}$$

כל התווים באותה הכמות.

סתירה. ולכן השפה לא רגולרית.

שאלה ממבחן מועד ב' 2019

שאלה 1 (27 נקודות)

לכל אחת מהשפות הבאות קבעו האם השפה רגולרית או לא. עליכם להוכיח את תשובתכם, אולם אם אתם בונים אוטומט או ביטוי רגולרי, אין צורך להוכיח את נכונות הבניה.

א. (9 נקודות) $\Sigma = \{a, b\}$, $L_1 = \{a^{2i}(ba)^i \mid i \geq 0\}$

ב. (9 נקודות) $L_2 = \{wuw^R \mid w \in \{0,1\}^+, u \in \{0,1\}^*\}$

ג. (9 נקודות) $L_3 = \{w1w^R \mid w \in \{0,1\}^+\}$

פתרון א'

אינטואיציה: אנחנו חייבים להבטיח שמספר המופעים של 'a' יהיה פי 2 ממספר המופעים של 'ba' וגם להבטיח שמספר ה-n מופעים של 'ba' צריך להופיע רק אחרי כל ה-2n מופעים של 'a'.

כלומר, אנחנו צריכים לספור/לזכור את ההתנהגות ולכן השפה **לא** רגולרית.

הוכחה: נניח בשלילה שהשפה L_1 רגולרית ולכן היא מקיימת את למת הניפוח.

יהא מספר טבעי n קבוע המובטח בלמה.

נבחר את המילה z להיות $z = a^{2n} \cdot (ba)^n \in L_1$ זומתקיים $|z| = 3n \geq n$.

יהא פירוק כלשהו של המילה z מן הצורה: $z = uvw$ זבאופן הבא:

$$u = a^k, v = a^{ti}, w = a^{2n-k-t} \cdot (ba)^n$$

נשים לב ש:

$$k + t \leq n \text{ וגם } 1 \leq t \leq n$$

כלומר:

$$z = a^k \cdot a^{ti} \cdot a^{2n-k-t} \cdot (ba)^n$$

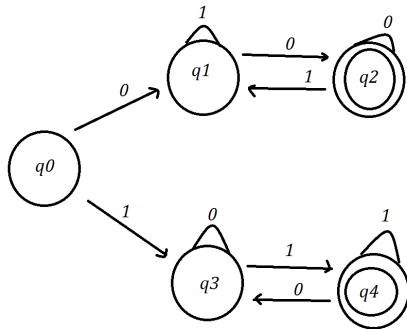
$$z = a^{2n+t(i-1)} \cdot (ba)^n$$

נבחר $i = 0$ ונקבל:

$$z = a^{2n-t} \cdot (ba)^n$$

מאחר ו- $t \geq 1$ אז בהכרח $z \notin L_1$ זזאת **סתירה** מאחר כי רצף ה- a הוא פחות מ- $2n$ ולכן המילה לא בשפה. לכן השפה לא רגולרית.

פתרון ב'



L_2 רגולרית כיוון שלכל מילה x , אם התו הראשון והאחרון זהים אז:

סעיף $x = w$ ולכן נוכל לבחור: $w = \sigma$ ואז $w^R = \sigma$ וכן: $u = y$. ואחרת, המילה לא בשפה. לכן L_2 היא שפת כל המילים שמתחילות

ומסתיימות באותו תו.

הוכחה: ציור אוטומט:

פתרון ג'

אינטואיציה: אין באפשרותינו לזכור את המילה w כדי לקבל את ההפוך של w לאחר שרשור עם ה- 1 . לכן השפה **לא** רגולרית.

הוכחה:

נניח בשלילה ש- L_3 היא שפה רגולרית ולכן היא מקיימת את למת הניפוח לשפות רגולריות.

יהא n מספר טבעי, קבוע המובטח בלמה.

נבחר את z להיות המילה: $z = 0^n \cdot 1 \cdot 0^n$ זומתקיים: $|z| = 2n + 1 \geq n$.

יהא פירוק של z מן הצורה: $z = uvw$ זכך ש:

$$u = 0^k, v = 0^{ti}, w = 0^{n-k-t} \cdot 1 \cdot 0^n$$

כאשר $1 \leq t \leq n$ וגם $k + t \leq n$.

כלומר z נראה כך:

$$z = 0^k \cdot 0^{ti} \cdot 0^{n-k-t} \cdot 1 \cdot 0^n$$

$$z = 0^{n+t(i-1)} \cdot 1 \cdot 0^n$$

נבחר $i = 0$ ונקבל:

$$z' = 0^{n-t} \cdot 1 \cdot 0^n$$

מאחר ו- $1 \leq t \leq n$ בהכרח $n - t < n$ כלומר $z' \notin L_3$, כיוון ש- $(0^n)^R \neq 0^{n-t}$.

סתירה - לכן השפה L_3 היא לא שפה רגולרית.

שאלה ממבחן מועד א' 2019

שאלה 1 (27 נקודות)

לכל אחת מהשפות הבאות קבעו האם השפה רגולרית או לא. עליכם להוכיח את תשובתכם, אולם אם אתם בונים אוטומט או ביטוי רגולרי, אין צורך להוכיח את נכונות הבניה.

- א. (9 נקודות) $\Sigma = \{a, b, c, d\}$, $L_1 = \{a^i b^j c d^k \mid i > 0, k \geq j > 0\}$
- ב. (9 נקודות) $\Sigma = \{a, b\}$, $L_2 = \{w \mid w = w^R, |w| \text{ is odd}\}$ (שפת כל המילים באורך אי זוגי בהן כל מילה שווה למילה ההפוכה)
- ג. (9 נקודות) $\Sigma = \{0, 1\}$, $L_3 = \{0^i 1^j \mid i, j \geq 2\}$

פתרון א'

אינטואיציה: אין אפשרות לזכור את j כדי להבטיח ש- k יהיה גדול או שווה אליו. לכן השפה לא רגולרית.

הוכחה: נניח בשלילה שהשפה L_1 היא שפה רגולרית, לכן היא מקיימת את למת הניפוח לשפות רגולריות.

יהא n מספר טבעי, קבוע המובטח בלמה.

נבחר את המילה z להיות: $z = a \cdot b^n \cdot c \cdot d^n$ כך ש- $|z| = 2n + 2 \geq n$.

יהא פירוק כלשהו של המילה z מן הצורה uvw כך ש:

$$u = a \cdot b^k, v = b^{ti}, w = b^{n-k-t} \cdot c \cdot d^n$$

כאשר $1 \leq t \leq n$ וגם $t + k \leq n$

המילה z תראה כך:

$$z = a \cdot b^k \cdot b^{ti} \cdot b^{n-k-t} \cdot c \cdot d^n$$

$$z = a \cdot b^{n+t(i-1)} \cdot c \cdot d^n$$

נבחר $i = 2$ ונקבל:

$$z' = a \cdot b^{n+t} \cdot c \cdot d^n$$

מאחר ו- $1 \leq t \leq n$ אז $z' \notin L_1$ וזאת סתירה. כיוון שבהכרח $n + t > n$ כלומר $j > k$.

ולכן השפה L_1 היא לא שפה רגולרית.

פתרון ב'

אינטואיציה: אם הא"ב בשפה היה רק תו אחד למשל $\Sigma = \{a\}$ אז מילה הייתה מכילה אך ורק סוג תו אחד, ומבטיחה ש- w מאורך אי זוגי ומתקיים למשל $w = aaa = w^R$. אבל זה לא המקרה כי $\Sigma = \{a, b\}$, אני צריך להציג פלינדרום כלומר מילה באורך אי-זוגי שווה להיפוך שלה, וידוע כי פלינדרום בשפות רגולריות ולכן השפה אינה רגולרית.

הוכחה: נניח בשלילה שהשפה L_2 היא שפה רגולרית, לכן היא מקיימת את למת הניפוח לשפות רגולריות. יהא n מספר טבעי וקבוע המובטח בלמה.

נבחר את המילה z להיות: $z = a^n b a^n$, כך ש- $|z| = 2n + 1 \geq n$.

יהא פירוק של המילה z מן הצורה הבאה $z = uv$ שמתקיים:

$$u = a^k, v = a^{ti}, w = a^{n-k-t} \cdot b \cdot a^n$$

כאשר $1 \leq t \leq n$ וגם $k + t \leq n$.

המילה z נראית כך:

$$z = a^k \cdot a^{ti} \cdot a^{n-k-t} \cdot b \cdot a^n$$

$$z = a^{n+t(i-1)} \cdot b \cdot a^n$$

נבחר $i = 0$ ונקבל:

$$z' = a^{n-t} \cdot b \cdot a^n$$

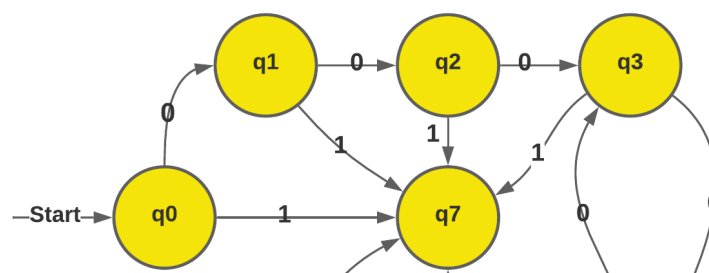
מאחר ו- $1 \leq t \leq n$ בהכרח $n - t < n$ כלומר $z' \notin L_2$ וזאת סתירה כיוון ש- $w^R \neq w$.

לכן השפה **אינה** רגולרית.

פתרון ג'

ה- i לא תלוי ב- n וגם להפך, שניהם תלויים במספר קבוע (2) שחייבים לקיים לכל מילה בשפה, יש כאן מחזוריות ברורה ותקנית עבור שפה רגולרית ולכן L_3 היא שפה רגולרית.

הוכחה:



שאלה ממבחן מועד ב' 2020

שאלה 1 (27 נקודות)

לכל אחת מהשפות הבאות קבעו האם השפה רגולרית או לא. עליכם להוכיח את תשובתכם, אולם אם אתם בונים אוטומט או ביטוי רגולרי, אין צורך להוכיח את נכונות הבניה.
א. (9 נקודות) $L_1 = \{w \in \{0,1\}^* \mid \#_0(w) > 2 \cdot \#_1(w)\}$, כאשר $\#_0(w)$ הוא מספר המופעים של האות 0 במילה w .

$$L_1 = \{w \in \{0,1\}^* \mid \#_0(w) > 2 \cdot \#_1(w)\} \text{ א.}$$

הוכחה: נניח בשלילה שהשפה L_1 שפה רגולרית ולכן היא מקיימת את למת הניפוח לשפות רגולריות.

יהא n מספר טבעי, קבוע המובטח בלמה.

נבחר מילה z להיות: $z = 0^{2n+1} \cdot 1^n \in L_1$ כאשר $|z| = 3n + 1 \geq n$.

יהא פירוק כלשהו של המילה z מן הצורה הבאה: $z = uvw$ וגם $u = 0^k, v = 0^{ti}, w = 0^{2n+1-k-t} \cdot 1^n$ כאשר $1 \leq t \leq n$ וגם $k + t \leq n$.

כלומר z מילה מהפירוק:

$$z = 0^k \cdot 0^{ti} \cdot 0^{2n+1-k-t} \cdot 1^n$$

$$z = 0^{2n+1+t(i-1)} \cdot 1^n$$

נבחר $i = 0$ ונקבל:

$$z' = 0^{2n+1-t} \cdot 1^n$$

מאחר ו- $t \geq 1$ נקבל מצב בו $2n + 1 - t < 2n + 1$ כלומר $z' \notin L_1$ סתירה ללמת הניפוח.

לכן השפה לא רגולרית.

ב. (18 נקודות) תהי L_2 השפה הבאה: $L_2 = \{w \in \{0,1\}^* \mid w = w^R\}$ (שפת כל המילים מעל $\{0,1\}$ בהן כל מילה שווה למילה ההפוכה).

- $L_2 \cdot L_2$ (9 נקודות)

- $(L_2)^*$ (9 נקודות)

שימו לב שאין צורך לקבוע האם L_2 רגולרית או לא, אלא האם $L_2 \cdot L_2$ ו- $(L_2)^*$ רגולריות או לא.

פתרון**עבור $L_2 \cdot L_2$ השפה לא רגולרית.****הוכחה:** נניח בשלילה שהשפה $L_2 \cdot L_2$ היא שפה רגולרית ולכן מקיימת את למת הניפוח לשפות רגולריות.יהא מספר טבעי n , קבוע המובטח מהלמה.נבחר את המילה z להיות $z = 0^n \cdot 1^n \cdot 0^n \cdot \varepsilon \in L_2 \cdot L_2$ כזו ש- $|z| = 3n \geq n$.יהא פירוק כלשהו של המילה z מן הצורה: $z = uvw$ כאשר $u = 0^k, v = 0^{ti}, w = 0^{n-k-ti} \cdot 1^n \cdot 0^n \cdot \varepsilon$ עבור $1 \leq t \leq n$ וגם $t + k \leq n$.כלומר המילה z מן הפירוק:

$$z = 0^k \cdot 0^{ti} \cdot 0^{n-k-ti} \cdot 1^n \cdot 0^n$$

$$z = 0^{n+t(i-1)} \cdot 1^n \cdot 0^n$$

נבחר $i = 0$ ונקבל:

$$z' = 0^{n-t} \cdot 1^n \cdot 0^n$$

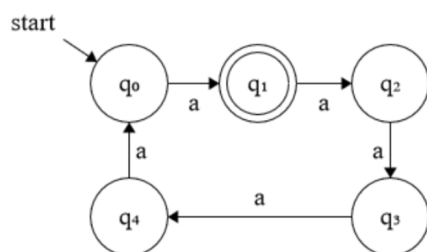
מאחר ו- $t \geq 1$ אז מתקיים ש- $t < n$ – כלומר הגענו למצב $z' \neq z$ כלומר, קיבלנו לאחר הניפוח מילה שלא מקיימת את תנאי השפה, $z' \notin L_2 \cdot L_2$ ובסתירה ללמת הניפוח.לכן השפה $L_2 \cdot L_2$ לא רגולרית.**עבור $(L_2)^*$: השפה רגולרית.****הוכחה:**תהא שפה זו שפת כל המילים מעל הא"ב $\Sigma = \{0, 1\}$ וזאת כיוון שהמילים "0" ו-"1" שייכות לשפה L_2 ולכן באיטרציה ל- L_2^* נוכל לבנות כל מילה המורכבת מהן כלומר $\{0, 1\}^*$.השפה רגולרית: $regex = (0 + 1)^*$.**שאלה**

קבעו והוכיחו האם השפה היא שפה רגולרית

$$L = \{w \in \{a\}^* : |w| \bmod 5 \equiv 1\}$$

פתרון

השפה רגולרית.

הוכחה

שאלות הוכחה על מודלים ושפות

שאלה ממועד א' 2020

שאלה 2 (27 נקודות)

הוכיחו או הפריכו:

א. (13 נקודות) בהינתן שפה L מעל Σ , נגדיר את השפה $DC(L)$ באופן הבא:

$$DC(L) = \{xy \mid \exists a \in \Sigma \text{ such that } xay \in L\}$$

אזי אם L רגולרית גם $DC(L)$ רגולרית.

ב. (14 נקודות) נגדיר מודל של אוטומט דטרמיניסטי אינסופי $M = (Q, \Sigma, q_0, \delta, F)$ בדומה לאוטומט סופי דטרמיניסטי (DFA, אס"ד) אבל ההבדל היחיד הוא ש- Q יכולה להיות בת מניה. אזי המודל DIA שקול למודל DFA .

פתרון של גיל לוי:

א. השפה רגולרית.

אינטואיציה: דוגמא: $L = \{abcc, abb\}$ ו- $DC(L) = \{bcc, acc, abc, bb, ab\}$.

מכיוון ש L רגולרית אז קיים לה אס"ד $A = (Q, \Sigma, q_0, \delta, F)$ כך ש: $L(A) = L$.

נגדיר אוטומט סופי עבור השפה $DC(L)$ באופן הבא:

ניצור 2 העתקים של האוטומט A כך שנתחיל במצב ההתחלתי של ההעתק הראשון, והמצבים המקבלים יהיו המקבלים רק בהעתק השני. מכל מצב בהעתק הראשון נוציא מעבר אפסילון במקום כל מעבר של תו באוטומט. כלומר אם: $\delta(q, a) = p$ אז נבצע מעבר אפסילון (בנוסף) מ- q (בהעתק הראשון) ל- p (בהעתק השני).

פורמאלי: $A_{DC} = (Q \cup \{q' : q \in Q\}, \Sigma, q_0, \delta', \{q' : q \in F\})$

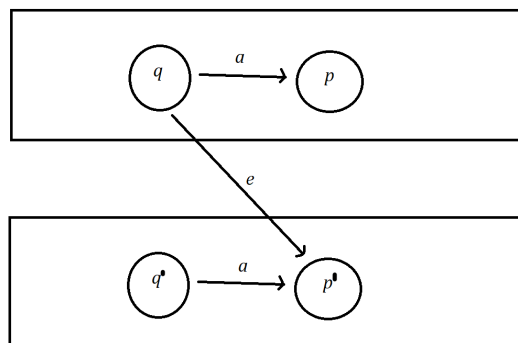
כאשר:

$$\delta'(q, a) = \{\delta(q, a)\}$$

$$\delta'(q', a) = \{\delta(q, a)'\}$$

$$\delta'(q, \varepsilon) = \{p' : \exists \sigma \in \Sigma \text{ s.t. } \delta(q, \sigma) = p\}$$

ציור להמחשה:



נוכיח שהאוטומט מקבל את השפה $DC(L)$.

כיוון 1: תהי $w \in DC(L)$. מכאן: קיים $x, y \in \Sigma^*$, $\sigma \in \Sigma$, כך ש: $x\sigma y \in L$.
 מכאן: $\delta(q_0, x\sigma y) \in F$. נסמן: $\delta(q_0, x) = q$, $\delta(q, \sigma) = p$, $\delta(p, y) = q_f \in F$.
 לפי בניית האוטומט, קיים מסלול שמקבל את xy .
 לפי בניית האוטומט: קיים מסלול כך ש: $q \in \delta'(q_0, x)$. בנוסף, $\delta'(q, \varepsilon) = p'$ כי $\delta(q, \sigma) = p$.
 ובנוסף, $\delta'(p', y) = q_f$ ומתקיים: q_f' הוא מצב מקבל.
 $\delta'(q_0, xy) \supseteq \delta'(q, y) \supseteq \delta'(p', y) \supseteq \{q_f'\}$ ולכן המילה מתקבלת.

כיוון 2: תהי w מילה שהתקבלה באוטומט A_{DC} .
 מכאן, מכיוון שחובה לעבור במסעי אפסילון להעתק השני, קיים מסלול $q \in \delta'(q_0, x)$ ומשם: $\delta'(q, \varepsilon) = p'$.
 ולאחר מכן: $\delta(p', y) = q_f$ כאשר q_f' הוא מצב מקבל.
 לפי בניית האוטומט, קיים $\sigma \in \Sigma$ כך ש: $\delta(q, \sigma) = p$ וכן: $\delta(q_0, x) = q$.
 בנוסף: $\delta(p, y) = q_f$ ולכן:
 $\delta(q_0, x\sigma y) = \delta(\delta(\delta(q_0, x), \sigma), y) = \delta(\delta(q, \sigma), y) = \delta(p, y) = q_f \in F$
 ולכן המילה מתקבלת ב A ולכן $x\sigma y \in L$ ולכן $xy \in DC(L)$.

ב. האוטומטים לא שקולים.

האוטומט DIA לא שקול ל DFA כי הוא מודל חזק יותר שיכול לקבל גם שפות לא רגולריות.

דוגמא לשפה לא רגולרית שכן יש לה אוטומט DIA . $L = \{a^n b^n : 0 < n \in N\}$.

נגדיר את האוטומט DIA הבא: $A = (Q, \Sigma, q_0, \delta, F)$ כאשר:

$$Q = \{q_{a=i} : i \in N \cup \{0\}\} \cup \{q_{b=i} : i \in N \cup \{0\}\} \cup \{q_x\}$$

$$\Sigma = \{a, b\}$$

$$q_0 = q_{a=0}$$

$$F = \{q_{b=0}\}$$

$$\delta(q_{a=i}, a) = q_{a=i+1}$$

$$i \neq 0 \text{ אם } \delta(q_{a=i}, b) = q_{b=i-1}$$

$$\delta(q_{a=0}, b) = q_x$$

$$\delta(q_{b=i}, a) = q_x$$

$$i \neq 0 \text{ אם } \delta(q_{b=i}, b) = q_{b=i-1}$$

$$\delta(q_{b=0}, b) = q_x$$

$$\delta(q_x, a) = q_x$$

$$\delta(q_x, b) = q_x$$

מתחילים במצב שזוכר שכמות ה a היא 0.

בכל קריאה של a עוברים למצב נוסף הזוכר את כמות ה a שקיבלנו עד כה.

ברגע שקוראים b אז עוברים למצב שזוכר שקיבלנו b והמונה יורד ב 1.

(הרעיון: כל a מגדיל מונה וכל b מקטין אותו ככה שאם חוזרים ל 0 אז הכמות זהה)

כל תו שאינו מהפורמט או שקראנו יותר b מעביר אותנו למצב בור.

מבחן 2019 סמסטר א מועד ב

שאלה 2 (27 נקודות)

הוכיחו או הפריכו:

א. (9 נקודות) לכל שפה L , אם L^* רגולרית אז L רגולרית.

ב. (9 נקודות) בהינתן שפה L נגדיר את השפה הבאה L' :

$$L' = \{x \mid \exists y \in L, \exists z \in L \text{ such that } x = yz^R\}$$

אם L רגולרית אז L' רגולרית.

ג. (9 נקודות) לכל שתי שפות L_1 ו- L_2 מעל אותו א"ב Σ מתקיים $(L_1 \cup L_2)^* = (L_1^* \cup L_2^*)^*$.

פתרון של גיל לוי

א. **הפרכה:** דוגמא נגדית: $\Sigma = \{0\}$, $L = \{0^n : n \in \mathbb{N}\}$ ומכאן: $L^* = \Sigma^*$.

קיבלנו ש L^* רגולרית אבל L לא רגולרית.

ב. נשים לב ש: $L' = L \cdot L^R$.

הוכחה: אם L רגולרית אז L^R רגולרית מסגירות שפות רגולריות להיפוך.

ולכן: $L \cdot L^R = L' = L$ רגולרית מסגירות שפות רגולריות לשרשור.

ג. **הפרכה:** דוגמא נגדית: $\Sigma = \{0, 1\}$, $L_1 = \{0\}$, $L_2 = \{1\}$ ומכאן: $\{0, 1\}^* \neq \{0\}^* \cup \{1\}^*$ כי לדוגמא:

המילה 01 נמצאת ב $\{0, 1\}^*$ אך לא ב $\{0\}^* \cup \{1\}^*$.

שאלה 2 (27 נקודות)

הוכיחו או הפריכו:

א. (9 נקודות) משפחת השפות הלא רגולריות סגורה תחת משלים (כלומר, משלים של שפה לא רגולרית הוא שפה לא רגולרית)

ב. (9 נקודות) משפחת השפות הלא רגולריות סגורה תחת איחוד סופי (כלומר, איחוד של מספר סופי של שפות לא רגולריות הוא שפה לא רגולרית)

ג. (9 נקודות) קיימת קבוצה אינסופית של שפות לא רגולריות השונות זו מזו שאיחודן הוא שפה רגולרית

פתרון:

א. הוכחה: תהי L שפה לא רגולרית ונניח בשלילה ש $\bar{L}$ רגולרית. מכאן: $\bar{\bar{L}} = L$ תהיה רגולרית (מסגירות של רגולריות למשלים). וזו סתירה.

ב. הפרכה: דוגמא נגדית: $L = \{a^n b^n : n \in N\}$ לא רגולרית, ולפי סעיף א', $\bar{L}$ לא רגולרית. אבל: $\bar{L} = \Sigma^*$ $L \cup \bar{L}$ שהיא רגולרית.

ג. הוכחה: דוגמא: $L = \{a^n b^n : n \in N\}$ לא רגולרית, ולפי סעיף א', $\bar{L}$ לא רגולרית. ניקח את: $\{L, \bar{L}\} \cup \{\{a^n b^n : i < n \in N\} : 1 < i \in N\}$ כל אחת מהשפות היא לא רגולרית והן שונות זו מזו והאיחוד שלהן הוא Σ^* שהיא רגולרית.

שאלה 3 (27 נקודות)

הוכיחו או הפריכו:

א. (13 נקודות) בהינתן שפה L מעל א"ב $\Sigma = \{a, b\}$, נגדיר את השפה $B(L)$ באופן הבא: $B(L) = \{ww^R \mid w \in L\}$. אזי אם L רגולרית גם $B(L)$ רגולרית.

ב. (14 נקודות) בהינתן שפה L מעל א"ב Σ , נגדיר את השפה L' באופן הבא: $L' = \{xyz \mid xy^Rz \in L\}$. אזי אם L רגולרית גם L' רגולרית.

פתרון של גיל לוי:

א. אינטואיציה: דוגמא: $L = \{ab, aab, bb\}$ אז: $B(L) = \{abba, aabbaa, bbbb\}$. צריך לדעת את סדר התווים בתחילת המילה כדי לדעת את הסדר ההפוך בסוף המילה. השפה $B(L)$ אינה בהכרח רגולרית.

דוגמא נגדית: $L = \{a^n b : n \in N\}$. מכאן: $B(L) = \{a^n bba^n : n \in N\}$. נוכיח ש $B(L)$ אינה רגולרית.

הוכחה: נניח בשלילה שהשפה רגולרית ולכן מקיימת את למת הניפוח. יהי $n \in N$ הקבוע המובטח מהלמה.

נבחר: $z = a^n bba^n \in B(L)$. מתקיים: $|z| = 2n + 2 \geq n$.

יהי $z = uv$ פירוק כלשהו המקיים: $|u| \leq n, |v| \geq 1$.

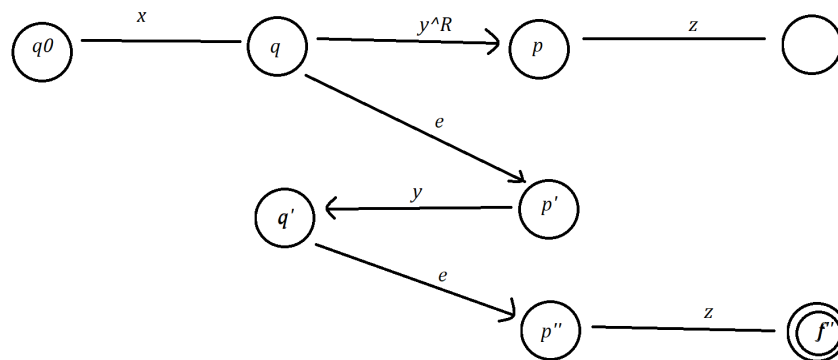
נבחר: $i = 2$ ונקבל: $z' = a^{n+|v|} bba^n \notin B(L)$ כי כמות ה- a בתחילת המילה לא זהה לכמות בסוף המילה. סתירה. ולכן השפה לא רגולרית.

ב. אינטואיציה: דוגמא: $L = \{abc, aacb\}$ אז:

$L' = \{abc, aacb, bac, acb, cba, acab, aabc, abca, bcaa\}$

בהינתן L רגולרית. השפה L' רגולרית.

יהא A האוטומט של L .
 נגדיר את האוטומט A' של L' באופן הבא:
 נתחיל משכפול ראשון של A (מהמצב ההתחלתי שלו). מכל מצב q נאפשר מעבר אפסילון לכל מצב בהעתק של האוטומט A^R עבור השפה L^R שגם היא רגולרית מסגירות שפות רגולריות להיפוך.
 כאשר ניצור העתק כזה לכל מצב מקור q ומצב יעד p . כלומר, מכל מצב q בהעתק הראשון נוציא מסע אפסילון למצב p' בהעתק: A_{qp}^R שזוכר שהגענו ממצב q וצריכים לסיים במצב p (בהעתק הנ"ל).
 מכל העתק כזה, נוציא מסע אפסילון אך ורק ממצב q' בהעתק הנ"ל: A_{qp}^R למצב p'' בהעתק נוסף (שני) של A .
 ההעתק השני של A יהיה ללא מסעי אפסילון חדשים והמצבים המקבלים שלו יהיו המצבים המקבלים של A .
 (בשאר ההעתיקים לא יהיו מצבים מקבלים).



שאלה 3 (27 נקודות)

- א. (13 נקודות) הוכיחו או הפריכו: תהי L_1 שפה רגולרית, ותהי L_2 שפה כך ש- $L_1 \Delta L_2$ רגולרית. (תזכורת: $L_1 \Delta L_2$ הוא ההפרש הסימטרי של שתי השפות כלומר, $(L_1 \Delta L_2) = (L_1 \setminus L_2) \cup (L_2 \setminus L_1)$). אזי L_2 רגולרית.
- ב. (14 נקודות) תהינה L_1 ו- L_2 שתי שפות רגולריות מעל אותו א"ב Σ . נגדיר את השפה הבאה L' :
 $L' = \{v \in \Sigma^* \mid \exists u \in L_2 \text{ such that } vu \in L_1\}$.
 רמז: בהינתן אס"ד ל- L_1 הראו כיצד ניתן לשנות אותו כדי לקבל אס"ד ל- L' .

פתרון:

- א. הוכחה: $(L_1 \Delta L_2) \Delta L_1 = L_2$ ומכיון ש- $L_1 \Delta L_2$ רגולרית וגם L_1 רגולרית אז מסגירות להפרש סימטרי של שפות רגולריות, גם L_2 רגולרית.
- ב. הוכחה: מכיון ש-2 השפות רגולריות אזי קיימים A_1, A_2 אוטומטים סופיים לכל אחת מהשפות.

נבנה אוטומט ל L' באופן הבא: נתחיל מהעתק ראשון של A_1 (מהמצב ההתחלתי שלו) ומכל מצב q בו נוציא מסע אפסילון למצב המקביל: (q, p_0) בהעתק חדש שהוא אוטומט מכפלה של A_1, A_2 . כאשר p_0 הוא המצב ההתחלתי של A_2 המצבים המקבילים יהיו זוג המצבים (q, p) בהעתק שהוא אוטומט מכפלה כך ש q הוא מצב מקבל ב A_1 וגם p הוא מצב מקבל ב A_2 .
 פורמאלית: אם $A_1 = (Q, \Sigma, q_0, \delta_1, F_1)$, $A_2 = (P, \Sigma, p_0, \delta_2, F_2)$ אז $A' = (Q \cup (Q \times P), \Sigma, q_0, \delta', F')$ כאשר:
 $F' = \{(q, p) : q \in F_1 \text{ and } p \in F_2\}$
 $\delta'(q, \varepsilon) = \{(q, p_0)\}$ ובנוסף: $\delta'(q, \sigma) = \{\delta_1(q, \sigma)\}$
 $\delta'((q, p), \sigma) = \{(\delta_1(q, \sigma), \delta_2(p, \sigma))\}$

שאלה 3 (27 נקודות)

- א. (13 נקודות) תהי L שפה רגולרית. נגדיר את השפה L' להיות שפת המילים z השייכות ל- L שעבורן לא קיים אף פירוק $z=uvw$ המקיים $|v| \geq 1$ וגם לכל $i \geq 0$. $uv^i w \in L$. הוכיחו או הפריכו: L' רגולרית.
- ב. (14 נקודות) שפה רגולרית L נקראת **יחידת קבלה** אם "ם קיים אוטומט סופי דטרמיניסטי A בעל מצב מקבל יחיד כך ש- $L=L(A)$. תנו דוגמא לשפה רגולרית לא ריקה שאינה יחידת קבלה. הוכיחו את תשובתכם.

פתרון

א. **הוכחה:** מכיוון ש L רגולרית אז היא מקיימת את למת הניפוח ולכן קיים $n \in \mathbb{N}$,

כך שעבור כל מילה $z \in L$ כך ש $|z| \geq n$ קיים פירוק $z = uvw$ המקיים:

$$|uv| \leq n, |v| \geq 1, \text{ וגם לכל } i \geq 0 \text{ מתקיים: } uv^i w \in L$$

מכאן, המילים ב L שלא מקיימות את התנאי הן לכל **היותר אלה באורך קטן מ n** ומכאן:

L' מכילה מספר סופי של מילים באורך הקטן או שווה ל n .

מכאן: L' סופית ולכן רגולרית כי כל שפה סופית היא רגולרית.

ב. **דוגמא:** $L = \{\varepsilon, 0\}$.

נניח בשלילה שקיים אוטומט A עם מצב מקבל יחיד כך ש: $L(A) = L$.

אבל מכאן: $\varepsilon \in L$ ולכן: $q_0 \in F$ ו $\delta(q_0, \varepsilon) = q_0$.

ומכיוון ש $0 \in L$ אז: $\delta(q_0, 0) \in F$ ומכיוון שיש רק מצב מקבל יחיד אז: $\delta(q_0, 0) = q_0$.

אבל מכאן: $q_0 \in F$ ו $\delta(q_0, 0) = \delta(\delta(q_0, 0), 0) = \delta(q_0, 0) = q_0 \in F$ ולכן המילה 00 התקבלה למרות שהיא

לא בשפה. סתירה.

שאלות של ביטויים רגולריים

הרעיון הוא לבנות אוטומט, להוריד את המצבים ותוך כדי שמורידים את המצבים, לבנות את הביטוי הרגולרי.

אלגוריתם לבניית ביטוי רגולרי

1. בונים לשפה אוטומט כמה עם כמה שפחות מצבים (כלומר לשאוף לבניית אוטומט אי-דטרמיניסטי).
2. בבניית האוטומט ניצור מצב מקבל יחיד.
3. האלגוריתם עצמו: בכל שלב, מוחקים מצב שאינו התחלתי ואינו המקבל ובודקים מה המסלול באורך 2 שמגיע מכל מצב לכל מצב (ששונה ממה שמחקנו) דרך אותו מצב שמחקנו. שאר המצבים אותו הדבר.

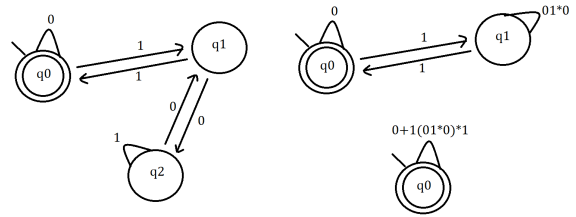
שאלה 4 (27 נקודות)

- כתבו ביטוי רגולרי עבור השפות הבאות. אין צורך להוכיח את תשובתכם אך הסבר קצר יועיל (במיוחד במקרה של טעות "טכנית" בביטוי).
- א. (9 נקודות) שפת כל המילים מעל $\{0,1\}$ שהן ייצוג בינארי של מספרים המתחלקים ב-3.
 - ב. (9 נקודות) שפת כל המילים מעל $\{0,1\}$ שאורכן מתחלק ב-3.
 - ג. (9 נקודות) שפת כל המילים מעל $\{0,1\}$ שמכילות לפחות שלושה 1-ים.

פתרון:

א.

הסבר על הציור:



תשובה סופית: $r_1 = (0 + 1(01^*0)^*1)^*$

ב. $r_2 = ((0 + 1)^3)^*$

בכל איטרציה חובה 3 תווים שהם 0 או 1.

ג. $r_3 = 0^*10^*10^*1(0 + 1)^*$

שאלה 4 (27 נקודות)

א. (13 נקודות) יהי r ביטוי רגולרי כלשהו. תהי L שפה מעל Σ כך ש- $L = \Sigma^* \cdot L[r]$. הוכיחו שאם $L \neq \emptyset$ וגם $L \neq \Sigma^*$ (כלומר L שונה מ- Σ^* וגם L אינה ריקה) אז לא קיים ביטוי רגולרי s כך ש- $\bar{L} = \Sigma^* \cdot L[s]$.

ב. (14 נקודות) כיתבו ביטוי רגולרי עבור השפה הבאה- $\{w \mid \#_a(w) \bmod 2 = \#_b(w) \bmod 2\}$, כאשר $\#_a(w)$ הוא מספר ה- a ים במילה w , $\#_b(w)$ הוא מספר ה- b ים במילה w . אין צורך להוכיח את תשובתכם אך הסבר קצר יועיל (במיוחד במקרה של טעות "טכנית" בביטוי).

פתרון:

א.

הסבר: הכוונה שהמילים ב- L : כל תחילת מילה יכולה להיות הכל, כלומר היא לא מוגבלת (כי היא Σ^*) אבל הסיומת של המילה חייב להיות עם ביטוי שהוא מהפורמט של r . $L[r] =$ זה השפה של הביטוי הרגולרי r .

הוכחה: נניח בשלילה שקיים ביטוי רגולרי s כך ש: $\bar{L} = \Sigma^* \cdot L[s]$.

ונניח ש $L \neq \emptyset$ וגם $L \neq \Sigma^*$.

נוכיח ש $L[r] = \bar{L}[s]$.

תהי $w \in L[r]$ ומכאן: $w = \varepsilon \cdot w \in L$ ולכן: $w \notin \bar{L}$ ולכן בהכרח $w \notin L[s]$ כי אחרת: $\varepsilon \cdot w \in \bar{L}$.

תהי $w \in L[s]$ ומכאן: $w = \varepsilon \cdot w \in \bar{L}$ ולכן בהכרח $w \notin L[r]$ כי אחרת: $\varepsilon \cdot w \in L$.
 מכאן, $\varepsilon \in L[r]$ או $\varepsilon \in L[s]$.
 אבל אם $\varepsilon \in L[r]$ אז $L = \Sigma^* - L$ - סתירה.
 ואם $\varepsilon \in L[s]$ אז $\bar{L} = \Sigma^* - L = \Phi$ ומכאן: $L = \Sigma^*$ - סתירה.
 ולכן לא קיים ב"ר s כזה.

ב. השפה פשוט מתארת את כל המילים באורך זוגי ולכן נקבל:

$$r_2 = ((a + b)^2)^*$$



שאלה 4 (27 נקודות)

א. (13 נקודות) לפניכם זוג ביטויים רגולריים. קבעו האם הם מציינים את אותה שפה. הוכיחו את תשובתכם

$$r_2 = (b^* a)^* (a^* b)^*, r_1 = (a + b)^*$$

ב. (14 נקודות) כיתבו ביטוי רגולרי עבור שפת כל המילים מעל $\Sigma = \{a, b\}$ שבהן מספר ה-a-ים שווה למספר ה-b-ים וכן בכל רישא (תחילית) ההפרש בין מספר ה-a-ים למספר ה-b-ים אינו עולה בערכו המוחלט על 2: $\{w \mid \#_a(w) = \#_b(w), \forall u \text{ which is a prefix of } w \mid \#_a(u) - \#_b(u) \leq 2\}$
 כאשר $\#_a(w)$ הוא מספר ה-a-ים במילה w .
 אין צורך להוכיח את תשובתכם אך הסבר קצר יועיל (במיוחד במקרה של טעות "טכנית" בביטוי).

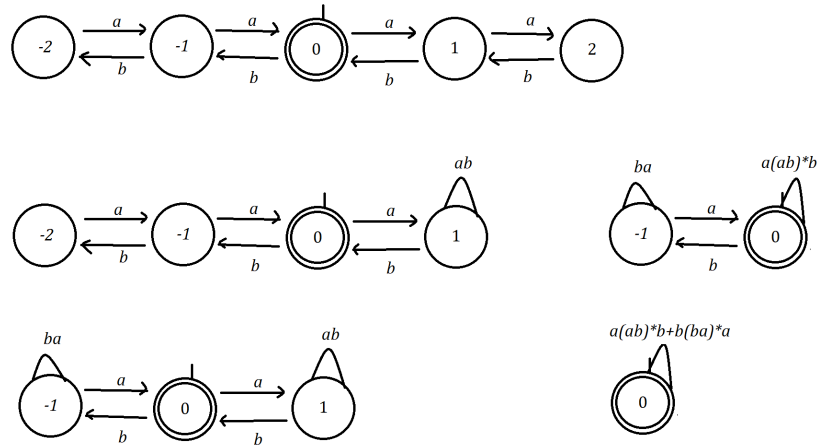
פתרון:

א. הוכחה:

כיוון 1: מכיוון ש $\Sigma = \{a, b\}$ אז $L[r_1] = \Sigma^*$ ולכן בהכרח: $L[r_2] \subseteq L[r_1]$.
 כיוון 2: תהי $w \in L[r_1]$. מכאן נחלק את w באופן הבא: $w = x \cdot y$ כך ש: x מסתיימת ב a , y מכילה רק b .
 נשים לב שאם ב w אין a אז $w = y$ ואם יש בו a אבל הוא מסתיים ב a אז $w = x$.
 לכן כיסינו את כל המקרים.

מכאן: $y \in L[(a^* b)^*]$ כאשר לא נבחר בכלל את a וניקח רק את b .
 את x נחלק באופן הבא: $x = u_1 u_2 \dots u_k$ כך ש: u_i הוא רצף (אפילו של 0) של b המסתיים ב a בודד.
 ולכן: $u_i \in L[(b^* a)^*]$ ומכאן: $x \in L[(b^* a)^*]$ ולכן: $w \in L[r_2]$.

ב.



תשובה סופית: $r = (a(ab)^*b + b(ba)^*a)^*$

סגירות שפות תחת פעולות

כל שפה שהעוצמה שלה סופית, היא שפה רגולרית. (לא בהכרח להפך).
אם L_1 ו- L_2 רגולריות אז גם הפעולות הבאות הם שפות רגולריות:

$$\begin{aligned} & L_1^R \\ & \overline{L_1} \\ & L_1 \cdot L_2 \\ & L_1 \cup L_2 \\ & L_1 \cap L_2 \end{aligned}$$

אם L_2 רגולרית ו- $L_1 \subseteq L_2$, זה לא מחייב ש- L_1 היא גם רגולרית!.

קבוצת השפות הרגולריות סגורה תחת פעולת האיטרציה $L^* = \bigcup_{i=0}^{\infty} L^i$.

אם L לא רגולרית אז רק הפעולות הבאות הם גם שפות לא רגולריות: L^R , $\overline{L}$.
הוכחה עבור אם L לא רגול' אז $\overline{L}$ לא רגול': תהי L שפה לא רגולרית ונניח בשלילה ש $\overline{L}$ רגול'.
מכאן: $\overline{\overline{L}} = L$ תהיה רגולרית (מסגירות של רגולריות למשלים). וזו סתירה.

שקילות אוטומטים

שני אוטומטים הם שקולים אם ורק אם הם מקבלים את אותה השפה.

A_1, A_2 הם אס"דים והם שקולים אם $A_1 = A_2$.

כדי להוכיח שקילות של אוטומטים, יש להוכיח הכלה דו כיוונית כלומר:

$$L(A_1) \subseteq L(A_2) \text{ and } L(A_2) \subseteq L(A_1)$$

אוטומטים לא שקולים - מודלים שקולים יכולים לקבל את אותה כמות של שפות. מספיק נראה שפה אחת שמודל אחד יכול לקבל והשני לא יכול לקבל, אז זה מוכיח שהמודלים לא שקולים.

שפות לא רגולריות שהוכחו בהרצאה

$$L = \{a^n b^n : n \geq 1\} \quad || \quad L = \{w \in \{a, b\}^* : \#_a(w) = \#_b(w)\}$$

$$L = \{ww : w \in \{a, b\}^*\} \quad || \quad L = \{a^{k^2} : k \geq 1\}$$

הוכח בהרצאה - שפה **לא** רגולרית **שכן מקיימת** את למת הניפוח: $L = \{a\}^* \cup \{b^j \cdot a^{k^2} : j, k \geq 1\}$

פעולות על שפות - תהיינה L_1, L_2 שפות מעל הא"ב Σ :

$$\bar{L} = \Sigma^* \setminus L \quad || \quad L_1 \cup L_2 = \{w : w \in L_1 \vee w \in L_2\} \quad || \quad L_1 \cap L_2 = \{w : w \in L_1 \wedge w \in L_2\}$$

$$L_1 \cdot L_2 = \{x \cdot y : x \in L_1 \wedge y \in L_2\} \quad || \quad L^R = \{w^R : w \in L\} \quad || \quad L^i = L \cdot \dots \cdot L \Rightarrow (i \text{ times})$$

$$L_1 \setminus L_2 = \{w : w \in L_1 \wedge w \notin L_2\} \text{ הפרש: } L_1 \cdot (L_2 \cdot L_3) = L_3 \cdot (L_1 \cdot L_2) \text{ אסוציאטיביות: } \\ (L_1 \cup L_2) \cdot L_3 = L_1 \cdot L_3 \cup L_2 \cdot L_3 \text{ דיסריטיביות:}$$

סדר פעולות פתרון לשאלות מבחן

שאלות של לקבוע האם השפה רגולרית והוכחות בלמת הניפוח (בדרך כלל בשאלה 1)

איך פותרים?

1. אם השפה רגולרית - מראים ביטוי רגולרי לשפה או מראים ציור אוטומט.
2. אם השפה לא רגולרית - מוכיחים באמצעות למת הניפוח.

איך מזהים ששפה רגולרית או לא?

אם נדרשת ספירה של משהו שאינו מחזורי ואינו חסום - השפה **לא** רגולרית (בתנאי שהספירה לא "נבלעת" בשרשרת או באיחוד).

מומלץ לכתוב בפתרון לפני ההוכחה "**אינטואיציה**" בה נפרט למה הסקנו שהשפה היא רגולרית או לא רגולרית.

שאלות הוכחה על שפות ומודלים (בדרך כלל בשאלות 2 ו-3)

1. שאלות על פעולות סגור מסוג "אם L_1 רגולרית ו L_2 רגולרית אז $L_1 \Delta L_2$ רגולרית":

אם מוכיחים: אז ננסה לפתור את השאלה בעזרת פעולות סגור לשפות רגולריות שהוכחנו בכיתה:

אם L_1 ו- L_2 רגולריות אז גם הפעולות הבאות הם שפות רגולריות:

$$\begin{aligned} L_1^R \\ \overline{L_1} \\ L_1 \cdot L_2 \\ L_1 \cup L_2 \\ L_1 \cap L_2 \end{aligned}$$

אם מפריכים:

נתן דוגמה נגדית: כלומר נתן דוגמא לשפה לא רגולרית שמקיימת / לא מקיימת (תלוי בשאלה).
אם נרצה להראות דוגמא לשפה לא רגולרית צריך להוכיח קודם שהיא לא רגולרית אלא אם היא שפה שנלמדה והוכחה בהרצאה כמו למשל:

$$L = \{a^n b^n : n \geq 1\}$$

$$L = \{w \in \{a, b\}^* : \#_a(w) = \#_b(w)\}$$

$$L = \{ww : w \in \{a, b\}^*\}$$

$$L = \{a^{k^2} : k \geq 1\}$$

הוכח בהרצאה - שפה **לא** רגולרית **שכן מקיימת** את למת הניפוח: $L = \{a\}^* \cup \{b^j \cdot a^{k^2} : j, k \geq 1\}$

2. הוכחת פעולות סגור (שקילות) כמו למשל $(L^*)^* = L^*$:

צריך לפתור בעזרת הוכחת אם"ם ובדרך כלל נוכיח באינדוקציה. למשל:

כיוון ראשון: תהי $w \in L^*$ שנראה כי $w \in (L^*)^*$ ומומשיכים...

כיוון שני: תהי $w \in (L^*)^*$ שנראה כי $w \in L^*$ ומומשיכים...

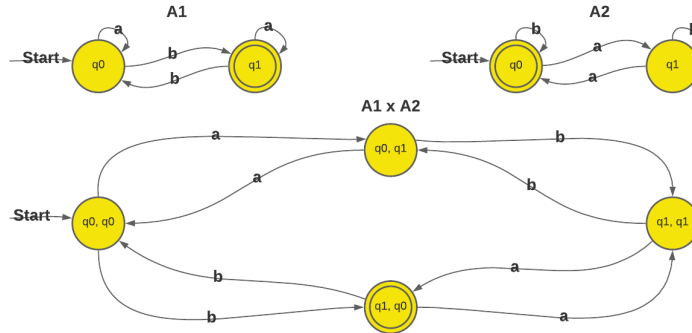
3. שאלות בשימוש פתרון בעזרת אוטומט מכפלה:

איך נראה אוטומט מכפלה

נשים לב כי המצבים באוטומט המכפלה $A_1 \times A_2$ הם זוגות סדורים של המכפלה הקרטזית.

יש חשיבות לסדר כלומר אם $A_1 \times A_2$ אז המצב המקבל יהיה (q_1, q_0) ולא (q_0, q_1) .

נתבונן לדוגמה במצב (q_0, q_0) , עבור A_1 במצב q_0 התו a מבצע חוג עצמי לכן יעבור ל- (q_0, q_1) ועבור התו b עובר למצב q_1 ולכן יעבור ל- (q_1, q_0) , נפעל באותו השיטה לשאר האוטומט.



סד"פ שאלה בעזרת שימוש באוטומט מכפלה:

1. נצייר בצד שרטוט פשוט על איך שאמור לתאר את האוטומט (לא לצרף לפתרון).
2. להסביר במילים (זה חלק מההוכחה) את התהליך (בנייה) של האוטומט מהמצב ההתחלתי שלו עד למקבל וכל המעברים הקריטיים שבדרך (בעיקר מסעי אפסילון).
3. נכתוב פורמאלית את האוטומט מכפלה שיצרנו (בדרך כלל רק 4 נק' מכלל השאלה).

שאלה לדוגמה:

ב. (14 נקודות) תהינה L_1 ו- L_2 שתי שפות רגולריות מעל אותו א"ב Σ . נגדיר את השפה הבאה L' :
 $L' = \{v \in \Sigma^* \mid \exists u \in L_2 \text{ such that } vu \in L_1\}$
 רמז: בהינתן אס"ד ל- L_1 הראו כיצד ניתן לשנות אותו כדי לקבל אס"ד ל- L' .

הוכחה: מכיוון ש 2 השפות רגולריות אזי קיימים A_1, A_2 אוטומטים סופיים לכל אחת מהשפות. נבנה אוטומט ל- L' באופן הבא: נתחיל מהעתק ראשון של A_1 (מהמצב ההתחלתי שלו) ומכל מצב q בו נוציא מסע אפסילון למצב המקביל: (q, p_0) בהעתק חדש שהוא אוטומט מכפלה של A_1, A_2 . כאשר p_0 הוא המצב ההתחלתי של A_2 המצבים המקבלים יהיו זוג המצבים (q, p) בהעתק שהוא אוטומט מכפלה כך ש q הוא מצב מקבל ב- A_1 וגם p הוא מצב מקבל ב- A_2 .

פורמאלית (החלק הזה מקבל בדרך כלל רק כ-4 נק' מהשאלה):

אם $A_1 = (Q, \Sigma, q_0, \delta_1, F_1)$, $A_2 = (P, \Sigma, p_0, \delta_2, F_2)$ אס"דים אז:

$A' = (Q \cup (Q \times P), \Sigma, q_0, \delta', F')$ כאשר:

$$F' = \{(q, p) : q \in F_1 \text{ and } p \in F_2\}$$

$$\delta'(q, \varepsilon) = \{(q, p_0)\} \text{ ובנוסף: } \delta'(q, \sigma) = \{\delta_1(q, \sigma)\}$$

$$\delta'((q, p), \sigma) = \{(\delta_1(q, \sigma), \delta_2(p, \sigma))\}$$

4. שאלות בשימוש פתרון בעזרת העתק של אוטומטים

סד"פ:

- נצייר לנו בצד שרטוט של האוטומט (לא חלק מהוכחה).
- נגדיר את האוטומט (נסביר במילים את האוטומט מהמצב ההתחלתי עד למקבל).

ב. (14 נקודות) בהינתן שפה L מעל Σ , נגדיר את השפה L' באופן הבא:
 $L' = \{xyz \mid xy^Rz \in L\}$. אזי אם L רגולרית גם L' רגולרית.

אינטואיציה: דוגמא: $L = \{abc, aacb\}$ אז $L' = \{abc, aacb, bac, acb, cba, acab, \dots\}$.

בהינתן L רגולרית. השפה L' רגולרית. יהא A האוטומט של L .

נגדיר את האוטומט A' של L' באופן הבא:

נתחיל משכפול ראשון של A (מהמצב ההתחלתי שלו). מכל מצב q נאפשר מעבר אפסילון לכל מצב

בהעתק של האוטומט A^R עבור השפה L^R שגם היא רגולרית מסגירות שפות רגולריות להיפוך.

כאשר ניצור העתק כזה לכל מצב מקור q ומצב יעד p . כלומר, מכל מצב q בהעתק הראשון נוציא מסע

אפסילון למצב p' בהעתק: A_{qp}^R שזוכר

שהגענו ממצב q וצריכים לסיים במצב p (בהעתק הנ"ל).

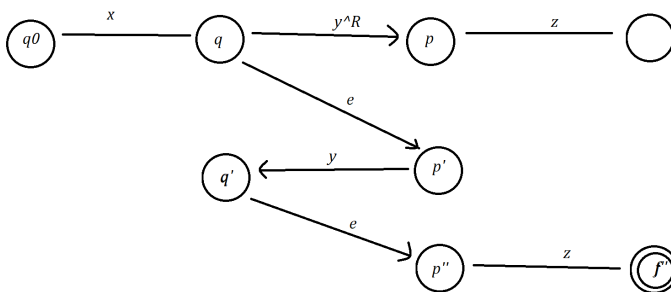
מכל העתק כזה, נוציא מסע אפסילון אך

ורק ממצב q' בהעתק הנ"ל: A_{qp}^R למצב

p'' בהעתק נוסף (שני) של A .

ההעתק השני של A יהיה ללא מסעי

אפסילון חדשים והמצבים המקבלים שלו יהיו המצבים המקבלים של A . (בשאר ההעתיקים לא יהיו מצבים מקבלים).



5. שקילות אוטומטים

שני אוטומטים הם שקולים אם ורק אם הם מקבלים את אותה השפה.

A_1, A_2 הם אס"דים והם שקולים אם $A_1 = A_2$.

כדי להוכיח שקילות של אוטומטים, יש להוכיח הכלה דו כיוונית כלומר:

$$L(A_1) \subseteq L(A_2) \text{ and } L(A_2) \subseteq L(A_1)$$

אוטומטים לא שקולים - מודלים שקולים יכולים לקבל את אותה כמות של שפות. מספיק נראה שפה אחת שמודל אחד יכול לקבל והשני לא יכול לקבל, אז זה מוכיח שהמודלים לא שקולים.

ההבדל בין אוטומט מכפלה לאוטומט חזקה-

- אוטומט מכפלה - אוטומט שכל מצב הוא זוג סדור של מצבי אוטומטים אחרים.
- אוטומט חזקה - אוטומט שכל מצב הוא קבוצת החזקה של קבוצת מצבי אוטומט לא דטרמיניסטי. (משתמשים בו להוכחת שקילות בין מודלים)

שאלות הוכחה על ביטויים רגולריים (בדרך כלל בשאלה 4)

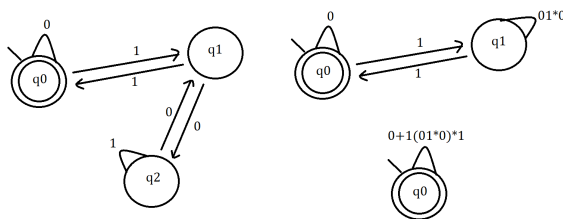
הרעיון הוא לבנות אוטומט, להוריד את המצבים ותוך כדי שמורידים את המצבים, לבנות את הביטוי הרגולרי.

אלגוריתם לבניית ביטוי רגולרי

1. בונים לשפה אוטומט כמה עם כמה שפחות מצבים (כלומר לשאוף לבניית אוטומט אי-דטרמיניסטי).
2. בבניית האוטומט ניצור מצב מקבל יחיד.
3. האלגוריתם עצמו: בכל שלב, מוחקים מצב שאינו התחלתי ואינו המקבל ובודקים מה המסלול באורך 2 שמגיע מכל מצב לכל מצב (ששונה ממה שמחקנו) דרך אותו מצב שמחקנו. שאר המצבים אותו הדבר.

לדוגמה

א. (9 נקודות) שפת כל המילים מעל $\{0,1\}$ שהן ייצוג בינארי של מספרים המתחלקים ב-3.



פתרון: נצייר ונפעל לפי הסד"פ:

$$r_1 = (0 + 1(01^*0)^*1)^*$$

חלאס.