

## הרצאה 2: אוטומטים סופיים

Based on previous iterations of this course, given by Nir Bitansky, Rotem Oshman, Iftach Haitner, and Omer Paneth.

מרצה: אורי שטמר

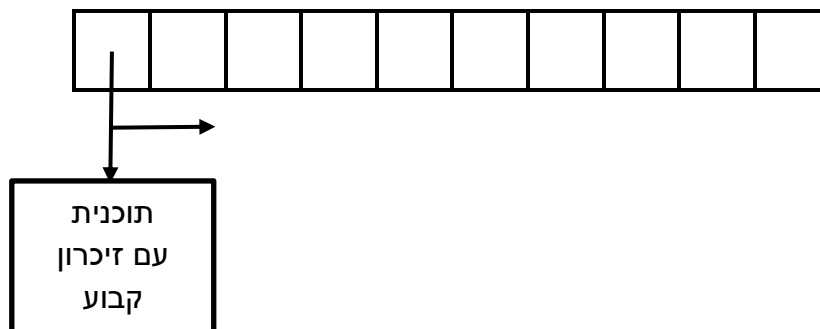
בהרצאה הקודמת דיברנו על מעגלים בוליאניים, שזה מודל חישובי מאוד טבעי ופשוט, אבל גם לא ראיסטי: כדי לפתור בעיות כלליות (על קלטים באורך משתנה) צריך מעגל נפרד לכל אורך קלט. למשל כדי לפתור בעיה כמו "האם מספר הוא מספר ראשוני" נצטרך משפחה אינסופית של מעגלים שזה לא ראיסטי.

המטרה שלנו בסופו של דבר היא לדבר על היכולות והמגבלות של "חישוב" כמו שאנחנו מכירים אותו, למשל תוכניות בפיתון. שם אנחנו יודעים שאותם כמה שורות קוד יודעות לטפל בקלטים באורך שרירותי.

בתור חימום לקראת תוכניות כפי שאנחנו מכירים אותם, בשיעורים הקרובים אנחנו נדבר על מודל הרבה יותר פשוט שנקרא אוטומטים סופיים. זהו מודל חישוב שתופס תוכניות עם זיכרון מאוד קטן (קבוע, לא תלוי באורך הקלט). המודל הזה לא מספיק כללי כדי לתפוס את כל היכולות של פיתון למשל. אבל לפחות זה יהיה מודל יוניפורמי, כלומר מודל שבו יש לנו תוכנית אחת שיודעת לטפל בקלטים באורך שרירותי (לבעיות שאפשר לפתור במודל הזה...).

## אינטואיציה:

איך נראית תוכנית עם זיכרון קבוע? תוכנית כזאת לא יכולה להחזיק את כל הקלט בזיכרון. היא אפילו לא יכולה "לדעת" מהו אורך הקלט... נרצה למדל את זה באמצעות תוכנית פשוטה שקוראת את ביטי הקלט אחד-אחד משמאל לימין ובדרך "זוכרת" משהו בגודל קבוע<sup>1</sup>.



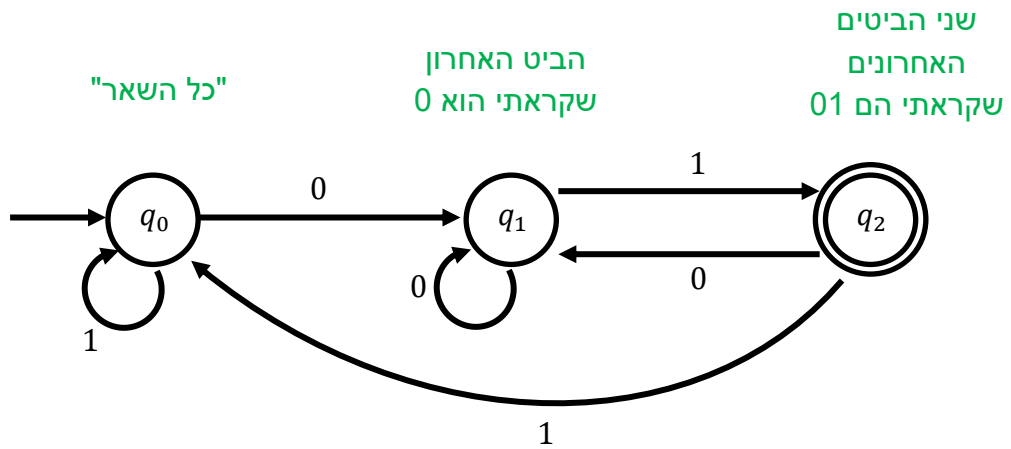
איך נמדל תוכניות כאלה? נתחיל עם דוגמה

דוגמה: נרצה להכריע את השפה של מילים בינאריות המסתיימות ב 01:  
 $L = \{w01 : w \in \{0,1\}^*\}$

מוטיבציה לדוגמה הזאת: חיפוש קובץ במחשב שנמר בסיומת כלשהי

<sup>1</sup> היינו יכולים לחשוב גם על מודל חישוב שבו לתוכנית מותר לנוע גם ימינה וגם שמאלה, ולכן יכולה לבצע יותר ממעבר אחד על הקלט. אנחנו נתרכז במודל הפשוט יותר בו מבצעים מעבר אחד בלבד. "ד", "א", מסתבר שהמודלים האלה "שקולים", עד כדי גידול אקספוננציאלי בזיכרון הדרוש...

נתאר תוכנית כזאת באופן סכמטי:



תוכנית כזאת נקראת אוטומט ומרכיביה:

- קבוצת מצבים  $(q_0, q_1, q_2)$
- מצב תחילי  $(q_0)$
- קבוצת המצבים המקבלים  $(q_2)$
- מעברים בין מצבים

ריצה על קלט:

- מתחילים: במצב ההתחלתי
- חוזרים: קוראים את התו הבא מהקלט משמאל לימין ועוברים מצב לפי הקשת המתאימה
- בסוף: אם סיימנו במצב מקבל אז מקבלים. אחרת דוחים.

**טענה:** (לא פורמלית, אנחנו לא מוכיחים את זה, נראה בהמשך איך מוכיחים כאלה טענות)

האוטומט הנ"ל מקבל בדיוק את השפה  $\{w01 : w \in \{0,1\}^*\}$

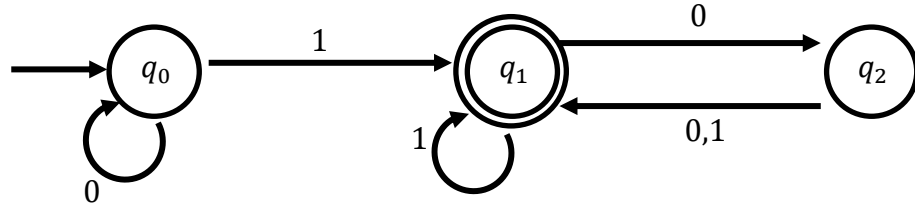
(לא פורמלית כי עוד לא הגדרנו מזה השפה שאוטומט "מקבל")

**קלטים לדוגמה:**

✓  $q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_1 \xrightarrow{0} q_2$  101 •

X  $q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_1$  010 •

(\*) נסתכל על אוטומט נוסף לדוגמה:



אנחנו נראה בהמשך (ואפילו נוכיח) מה השפה שהאוטומט הזה מקבל. כרגע רק נבדוק כמה דוגמאות לקלטים:

**קלטים לדוגמה:**

- ✓  $q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_1$  1101 •
- ✗  $q_0 \xrightarrow{0} q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2$  0010 •

**הגדרה:** אוטומט סופי דטרמיניסטי (אס"ד) הוא חמישייה  $A = (Q, \Sigma, \delta, q_0, F)$  כאשר:

- $Q$  קבוצה סופית (לא ריקה) של מצבים
- $\Sigma$  אלפאבית
- $\delta: Q \times \Sigma \rightarrow Q$  פונקציית מעברים
- $q_0 \in Q$  מצב תחילי
- $F \subseteq Q$  קבוצת מצבים מקבלים

בדוגמה האחרונה:

- $Q = \{q_0, q_1, q_2\}$
- $\Sigma = \{0,1\}$
- $\delta$  מוגדרת כך:

$\delta$	0	1
$q_0$	$q_0$	$q_1$
$q_1$	$q_2$	$q_1$
$q_2$	$q_1$	$q_1$

- $q_0$
- $F = \{q_1\}$

שימו לב: פונק' המעברים  $\delta$  מקבלת "מצב נוכחי" + "אות אחת" ואומרת לאן אנחנו זזים. יהיה לנו נח לחשוב על הכללה של זה אשר לוקחת "מצב נוכחי" + "מילה" ואומרת לאן נגיע אם נתחיל מהמצב הנוכחי ונפעיל את פ' המעברים תו תו. זאת הרחבה טבעית של פ' המעברים.

**הגדרה:** יהי  $A = (Q, \Sigma, \delta, q_0, F)$  אוטומט. פונקציית המעברים המורחבת  $\hat{\delta}: Q \times \Sigma^* \rightarrow Q$  מוגדרת בצורה רקורסיבית:

- עבור המילה הריקה  $\varepsilon$ , לכל מצב  $q \in Q$  מתקיים  $\hat{\delta}(q, \varepsilon) = q$
  - לכל  $n \geq 1$  ולכל מילה  $x \in \Sigma^n$  מתקיים  $\hat{\delta}(q, x) = \delta(\hat{\delta}(q, x_1 \dots x_{n-1}), x_n)$
- כלומר, נריץ קודם את פונק' המעברים המורחבת  $\hat{\delta}$  על  $n-1$  התווים הראשונים, ואח"כ נבצע עוד צעד לפי פונק' המעברים  $\delta$

עכשיו בעזרת פ' המעברים המורחבת נוכל להגדיר מה זה אומר "שאוטומט מקבל מילה":

**הגדרה:** אוטומט  $A$  מקבל מילה  $x \in \Sigma^*$  אם  $\hat{\delta}(q_0, x) \in F$ .  
 באופן שקול:  $A$  מקבל מילה  $x \in \Sigma^n$  אם קיימים  $q_1, \dots, q_n \in Q$  כך ש-  
 (א) לכל  $i \in [n]$  מתקיים  $\delta(q_{i-1}, x_i) = q_i$   
 (ב)  $q_n \in F$

אז לכל אוטומט יש את אוסף המילים שהוא מקבל. לזה אנחנו קוראים "השפה של האוטומט":

**הגדרה:** השפה של  $A$  היא אוסף המילים ש-  $A$  מקבל. מסומנת כ-  $L(A)$ .

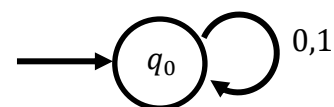
עכשיו אנחנו יכולים לשאול את עצמנו: מהו אוסף השפות שאפשר לקבל במודל הפשוט הזה, כלומר מהו אוסף השפות עבורן קיים אוטומט המקבל אותן? כפי שנראה בהמשך, לא לכל שפה יש אוטומט ולכן האוסף הזה הוא לא כל השפות בעולם.

**הגדרה:** שפה היא רגולרית אם קיים אוטומט המקבל אותה.

זה מושג שניגע בו עוד הרבה בהמשך. עכשיו כדי להפנים את המושגים הנ"ל, נראה עוד כמה דוגמאות לאוטומטים וננסה להבין לכל אחד מהם מה השפה שהוא מקבל

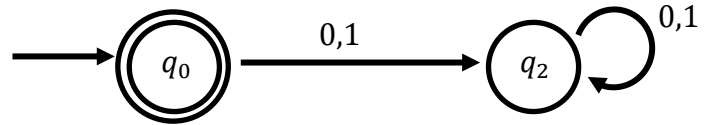
**דוגמה 1:**

$$L(A) = \emptyset$$

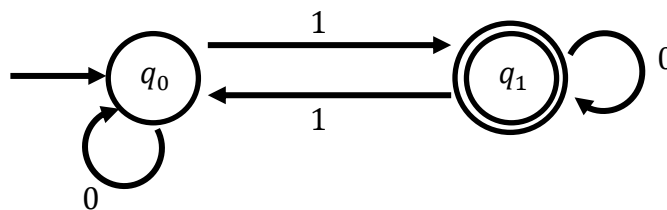


## דוגמה 2:

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



## דוגמה 3:



טענה:  $L(A) = \{x : \#_1(x) \bmod 2 = 1\}$   
נסמן גם כ  $\oplus x$

איך מוכיחים טענה כזאת פורמלית? באופן כללי, אנחנו יודעים שקבלה מוגדרת ע"י פונק' המעברים המורחבת שהיא נוסחה רקורסיבית ולכן בד"כ טענות כאלה יוכחו באינדוקציה. נראה כמה הוכחות כאלה. זה יהיה דיי פשוט ולא מפתיע במיוחד אבל זה משהו שאנחנו צריכים לשחות בו.

### הוכחה:

אבחנה: לכל  $b, c \in \{0,1\}$  מתקיים  $\delta(q_b, c) = q_{b \oplus c}$   
 (האבחנה הזאת נובעת מההגדרה של האוטומט)

כעת נוכיח באינדוקציה על אורך הקלט  $x$  כי

$$\hat{\delta}(q_0, x) = q_{\oplus x}$$

מדוע זה מספיק? משום ש  $q_1$  הוא מצב מקבל ו-  $q_0$  דוחה ולכן  $\hat{\delta}$  מובילה למצב מקבל אם "ם מס' האחדות בקלט הוא אי-זוגי.

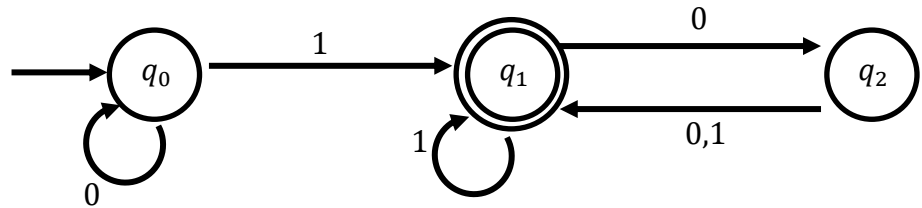
בסיס:  $x = \varepsilon$  ובפרט  $\hat{\delta}(q_0, \varepsilon) = q_0 = q_{\oplus \varepsilon}$

צעד אינדוקציה: יהי  $x = yb$  עבור  $y \in \{0,1\}^n$  ועבור  $b \in \{0,1\}$ . מתקיים:

$$\hat{\delta}(q_0, x) = \underset{\text{הגדרת } \hat{\delta}}{\delta(\hat{\delta}(q_0, y), b)} = \underset{\text{אינדוקציה}}{\delta(q_{\oplus y}, b)} = \underset{\text{מתחילת ההוכחה}}{q_{(\oplus y) \oplus b}} = q_{\oplus x}$$

מ.ש.ל.

**דוגמה 4: נחזור לאוטומט שראינו קודם, בדוגמה שסימנו כ- (\*)**



מהי השפה של האוטומט הזה?

אנחנו נוכיח עכשיו שהשפה הזאת מכילה את כל המילים שמסתיימות ב 1 ולאחריו רצף זוגי (אולי ריק) של אפסים. פורמלית,

**טענה:**

$$L(A) = \{y 1 0^{2k} : y \in \{0,1\}^*, k \geq 0\}$$

**הוכחה:**

נוכיח באינדוקציה על אורך הקלט  $x$  שמתקיים:

$$\delta(q_0, x) = \begin{cases} q_1 & , x \in \{y 1 0^{2k}\} \\ q_0 & , x \in \{0^k\} \\ q_2 & , x \in \{y 1 0^{2k+1}\} \end{cases}$$

נשים לב כי כל קלט  $x$  משתייך לבדיקת אחת משלוש הקבוצות הנ"ל. לכן בפרט האוטומט מקבל קלט  $x$ , כלומר  $\delta(q_0, x) = q_1$ , אם הקלט הזה הוא מהצורה הנכונה, כלומר  $x \in \{y 1 0^{2k}\}$

שאלה: מדוע שלושת המקרים האלו מהווים חלוקה של מרחב הקלטים שלנו?

- או שלא מופיעים אחדים בכלל (מקרה 2), או שכן מופיעים אחדים.
- אם מופיעים אחדים, נסתכל על האחד האחרון ונספור כמה אפסים יש אחריו. אם יש מספר זוגי אז אנחנו במקרה 1 ואם יש מספר אי-זוגי אז במקרה 3.

**בסיס:**

$$\delta(q_0, \varepsilon) = q_0 \text{ ואכן מתקיים } x = 0^0$$

**צעד אינדוקציה:** נראה עבור  $x \in \{0,1\}^{n+1}$

**מקרה 1:**  $x \in \{y 1 0^{2k}\}$  באורך  $n + 1$

**מקרה א1:**  $k = 0$  ולכן  $x = y1$ . במקרה זה,

$$\delta(q_0, x) = \delta(\delta(q_0, y), 1) = q_1$$

מכיוון ש- 1 תמיד מעביר ל  $q_1$  ללא קשר למצב בו נמצאים.

**מקרה 1:**  $k > 0$  ובפרט  $x = y 1 0^{2\ell+1} 0$  עבור  $\ell = k - 1 \geq 0$ . במקרה זה,

$$\hat{\delta}(q_0, x) = \delta(\hat{\delta}(q_0, y 1 0^{2\ell+1}), 0) = \delta(q_2, 0) = q_1$$

הנחת  
האינדוקציה

**מקרה 2:**  $x \in \{0^k\}$  באורך  $n + 1$

$$\hat{\delta}(q_0, x) = \delta(\hat{\delta}(q_0, 0^{k-1}), 0) = \delta(q_0, 0) = q_0$$

הנחת  
האינדוקציה

**מקרה 3:**  $x \in \{y 1 0^{2k+1}\}$  באורך  $n + 1$

$$\hat{\delta}(q_0, x) = \delta(\hat{\delta}(q_0, y 1 0^{2k}), 0) = \delta(q_1, 0) = q_2$$

הנחת  
האינדוקציה

מ.ש.ל.

עכשיו אנחנו עוברים לדבר על הנושא הבא שקשור לאוטומטים ובאופן כללי יותר לשפות רגולריות ותכונות שלהם

## תכונות סגירות של שפות רגולריות

תזכורת: שפה רגולרית היא שפה שיש אוטומט שמקבל אותה

שאלה: אם מישהו נותן לנו שפה, איך נוכיח שהיא רגולרית? איזה כלים עומדים לרשותנו? אפשרות 1, שכבר ראינו, נבנה אוטומט ונוכיח שהוא מקבל את השפה הנתונה. אפשרות 2, שנדבר עליה עכשיו, זה באמצעות סגירות של שפות רגולריות לתכונות/פעולות מסוימות.

יהי  $\Sigma$  אלפאבית ויהיו  $L, L' \subseteq \Sigma^*$  שפות. נגדיר:

• איחוד:

$$L \cup L' = \{x : x \in L \vee x \in L'\}$$

• חיתוך:

$$L \cap L' = \{x : x \in L \wedge x \in L'\}$$

• משלים (מוגדר ביחס ל- $\Sigma$ ):

$$\bar{L} = \Sigma^* \setminus L = \{x : x \in \Sigma^* \wedge x \notin L'\}$$

• שרשור:

$$L \parallel L' = L L' = \{x y : x \in L \wedge y \in L'\}$$

השרשור הוא אסוציאטיבי, כלומר  $L_1(L_2 L_3) = (L_1 L_2)L_3$

• חזקה:

$$L^0 = \{\varepsilon\}, \quad \text{and } \forall i \geq 1: L^i = L L^{i-1}$$

• סגור קליני:

$$L^* = \bigcup_{i \geq 0} L^i$$

הערה:  $L^*$  אינסופית לכל  $L$  מלבד  $\{\varepsilon\}, \emptyset$ .

### דוגמאות

- עבור  $L = \{1,3,5,7,9,11,13, \dots\}$  מתקיים  $L^* = \{\varepsilon\} \cup L$  למה? כי אם  $a, b, c$  הם אי-זוגיים אז גם  $abc$  הוא אי-זוגי ולכן הוא כבר נמצא ב  $L$ .
- תהי  $L_{\text{even}}$  אוסף המילים בעלות אורך זוגי ותהי  $L_{\text{odd}}$  אוסף המילים בעלות אורך אי-זוגי. אזי:

$$\begin{aligned} L_{\text{even}}^m &= L_{\text{even}}, \quad m \geq 1 \\ L_{\text{even}}^0 &= L_{\text{odd}}^0 = \{\varepsilon\} \\ L_{\text{odd}}^2 &= L_{\text{even}} \setminus \{\varepsilon\} \\ L_{\text{even}} L_{\text{odd}} &= L_{\text{odd}} = L_{\text{odd}} L_{\text{even}} \end{aligned}$$

**משפט:** אוסף השפות הרגולריות סגור לכל הפעולות הנ"ל.

### דוגמאות לשימוש:

1. האם השפה  $L = \{x : \#_1(x) \bmod 2 = 0\}$  רגולרית? כן, כי ראינו ש- $\bar{L}$  רגולרית.
2. קל להשתכנע שכל שפה המכילה מילה בודדת (שפה כזאת נקראת סינגלטון) היא רגולרית. מדוע? נוכל לבנות אוטומט "שבדוק" את התווים במילה הזאת אחד-אחד. אם יש התאמה מלאה נגיע למצב מקבל ואחרת דוחה.
3. לכן, לפי סגירות לאיחוד, גם כל שפה סופית היא רגולרית.
4. האם השפה  $\{0^n 1^m : n, m \geq 0\}$  היא רגולרית? כן, נובע מסגירות לשרשור מכיוון ש  $\{0\}^*$  ו- $\{1\}^*$  רגולריות (זה קל מאוד להראות)

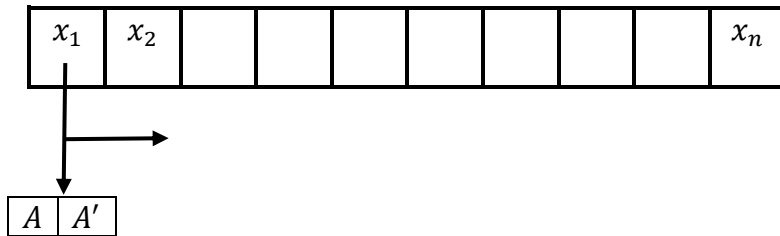
נתחיל מלהוכיח סגירות תחת איחוד

**משפט:** אם שפות  $L, L'$  רגולריות אז גם  $L \cup L'$  רגולרית

איך נוכיח? אנחנו יודעים שיש לנו אוטומט  $A$  שמקבל את  $L$  ויש לנו אוטומט  $A'$  שמקבל את  $L'$ . אם היה מדובר בפיתון אז היינו מריצים למשל קודם את  $A$  ומקבלים אם מקבלת, ואם לא אז מריצים את  $A'$ .

אבל עם אוטומטים אנחנו לא ממש יכולים לעשות את זה, כי יש לנו רק "one shot" לקרוא את הקלט.

הרעיון: נריץ את  $A, A'$  במקביל. בכל רגע, המצב בו נמצאים יקודד את שני המצבים בהם היינו נמצאים בכ"א מהאוטומטים



### הוכחה:

יהיו  $A = (Q, \Sigma, \delta, q_0, F)$  ו-  $A' = (Q', \Sigma, \delta', q'_0, F')$  אוטומטים המקבלים את  $L$  ו-  $L'$  בהתאמה. נגדיר באופן הבא:  $A_U = (Q_U, \Sigma, \delta_U, q_U, F_U)$

- $Q_U = Q \times Q'$
- לכל  $(q, q') \in Q \times Q'$  ולכל  $\sigma \in \Sigma$  נגדיר  $\delta_U((q, q'), \sigma) = (\delta(q, \sigma), \delta'(q', \sigma))$
- $q_U = (q_0, q'_0)$
- $F_U = \{(q, q') \in Q \times Q' : q \in F \vee q' \in F'\}$

על מנת להוכיח נכונות נשתמש בטענה הבאה:

**טענה:** לכל  $x \in \Sigma^*$  ולכל  $(q, q') \in Q \times Q'$  מתקיים

$$\hat{\delta}_U((q, q'), x) = (\hat{\delta}(q, x), \hat{\delta}'(q', x))$$

הוכחה באינדוקציה על אורך  $x$  (תוודאו)

הנה צעד האינדוקציה:

$$\begin{aligned} \hat{\delta}_U((q, q'), x\sigma) &\stackrel{\text{הגדרת } \hat{\delta}_U}{=} \delta_U(\hat{\delta}_U((q, q'), x), \sigma) \stackrel{\text{ה.א.}}{=} \delta_U((\hat{\delta}(q, x), \hat{\delta}'(q', x)), \sigma) \stackrel{\text{הגדרת } \delta_U}{=} \\ &= (\delta(\hat{\delta}(q, x), \sigma), \delta'(\hat{\delta}'(q', x), \sigma)) \stackrel{\text{הגדרות } \hat{\delta}, \hat{\delta}'}{=} (\hat{\delta}(q, x\sigma), \hat{\delta}'(q', x\sigma)) \end{aligned}$$

נכונות האוטומט  $A_U$  נובעת מהטענה:

$$x \in L(A_U)$$

אם"ם (הגדרת האוטומט  $A_U$ )

$$\hat{\delta}_U((q_0, q'_0), x) \in F_U$$

אם"ם (לפי הטענה האחרונה)

$$(\hat{\delta}(q_0, x), \hat{\delta}'(q'_0, x)) \in F_U$$

אם"ם (הגדרת  $F_U$ )

$$\hat{\delta}(q_0, x) \in F \quad \text{או} \quad \hat{\delta}'(q'_0, x) \in F'$$

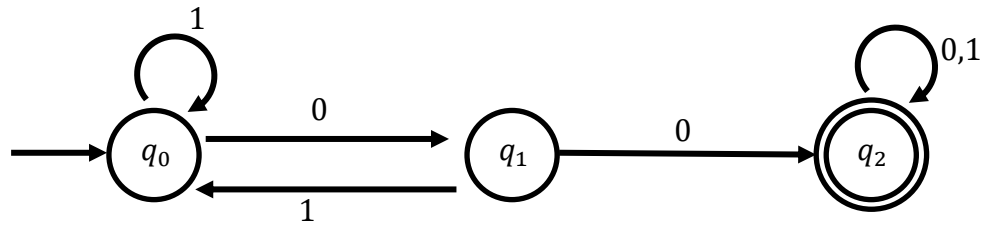
אם"ם (הגדרת האוטומטים  $A, A'$ )

$$x \in L(A) \cup L(A')$$

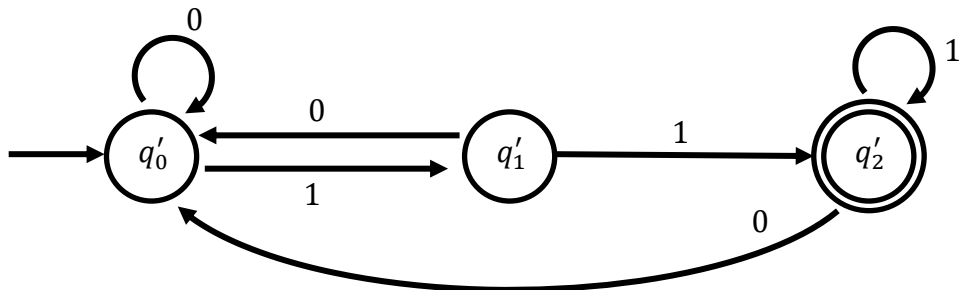
מ.ש.ל.

דוגמה: נסו לבנות את "אוטומט האיחוד" המתקבל לאחר אם מתחילים משני האוטומטים הבאים:

מילים המכילות 00



מילים המסתיימות ב 11



נקודה למחשבה: האוטומט המתקבל גדול משמעותית מהאוטומטים שהתחלנו איתם (גידול ריבועי...).  
בתרגול תוכיחו סגירות למשלים וחיתוך.



זה נקרא "עץ הריצה" על הקלט. כל אוטומט אי-דטרמיניסטי, בהינתן קלט, מגדיר לנו עץ (=אוסף של מסלולים/ריצות אפשריות שונות). נרצה לדעת "האם יש דרך להגיע למצב מקבל".

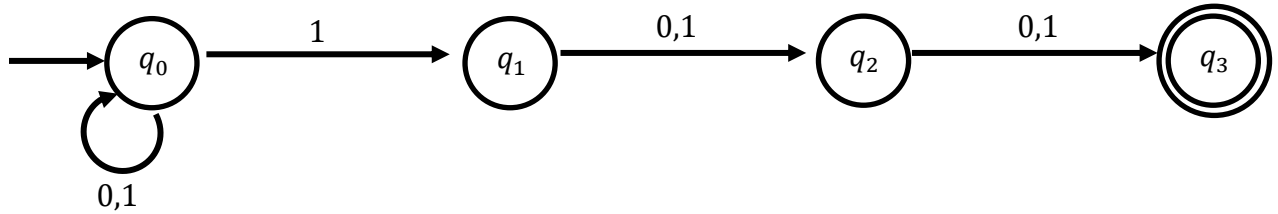
**משפט:** (נוכיח בשיעור הבא)

לכל שפה  $L$  קיים אס"ד שמקבל אותה אם"ם קיים אסל"ד שמקבל אותה.

אז למה בכלל שנדבר על אוטומטים אי-דטרמיניסטיים? לפעמים יהיה הרבה יותר פשוט לבנות כאלה...

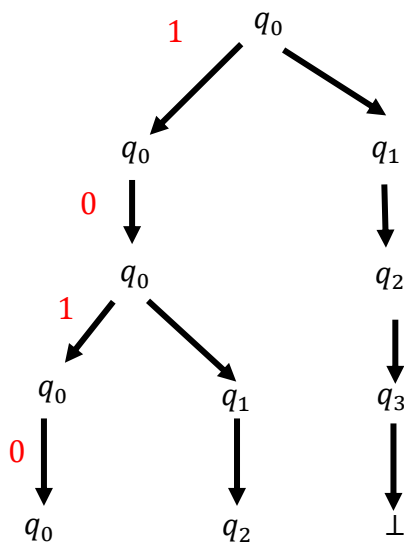
**דוגמה:** בנו אוטומט לא-דטרמיניסטי לשפה  $\{0,1\}^2 \{1\} \{0,1\}^*$  "כל המחרוזות הבינאריות בהן מופיע '1' בתו השלישי מהסוף"

אפשר לבנות אוטומט דטרמיניסטי לשפה הזאת, אבל הוא יהיה יחסית גדול ומסובך (אולי תעשו משהו דומה בשיעורי בית). עכשיו נראה אוטומט לא-דטרמיניסטי מאוד פשוט לשפה הזאת

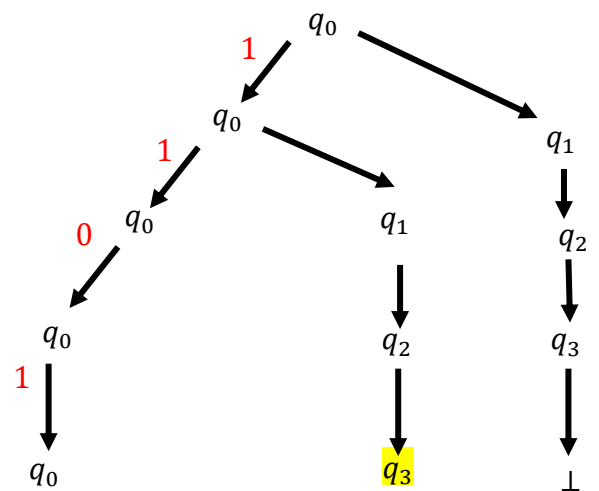


אפשר לחשוב על האוטומט הזה כאילו הוא מנסה לנחש מתי הוא הגיע לאות השלישית מהסוף. בהתחלה הוא קורא איזשהו string וחושב לעצמו שהוא עוד לא הגיע לסוף. מתישהו אנחנו נקרא תו '1' "וננחש" שהוא שלישי מהסוף וניכנס ל mode של בדיקה שזה המצב. אם המילה תימשך אחרי זה אז "ניחשנו לא נכון".

ריצה על קלט 1010:



ריצה על קלט 1100:



**תזכורת:** עבור קבוצה  $Q$  אנחנו מסמנים את אוסף תתי הקבוצות של  $Q$  (נקרא גם "קבוצת החזקה") באופן הבא:

$$2^Q = P(Q) = \{S : S \subseteq Q\}$$

עכשיו נגדיר אוטומטים כאלה בצורה כמעט פורמלית (כמעט כי אנחנו עוד לא מטפלים במעברים "ספונטניים"; נציין זאת בהגדרה כ- "מינוס").

**הגדרה:** אוטומט סופי לא-דטרמיניסטי מינוס (אסל"ד-)

הוא חמישייה  $N = (Q, \Sigma, \delta, S, F)$  כאשר:

- $Q$  קבוצה סופית (לא ריקה) של מצבים
- $\Sigma$  אלפאבית
- $\delta: Q \times \Sigma \rightarrow 2^Q$
- $S \subseteq Q$  קבוצת מצבים תחיליים
- $F \subseteq Q$  קבוצת מצבים מקבלים

הערה: בניגוד לאס"ד ששם היה לנו מצב תחילי יחיד כאן אנחנו מרשים אוסף של מצבים תחיליים. בינתיים לא דיברנו על כאלה דוגמאות אבל באופן כללי נרשה זאת באסל"ד. איך התמונה הייתה משתנה? היו לנו כמה עצים – עץ לכל מצב התחלתי.

בדוגמה לעיל  $N = (Q, \Sigma, \delta, S, F)$ :

$$Q = \{q_0, q_1, q_2, q_3\}, \quad \Sigma = \{0,1\}, \quad S = \{q_0\}, \quad F = \{q_3\}$$

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

בפעם הבאה ניזכר בהגדרה הזאת ואז נרצה להגדיר פורמלית את פ' המעברים המורחבת באסל"ד ומהי ריצה