תקציר מודלים חישוביים

ערך יגאל הינדי

<u>**תוכן עניינים**</u>

חלק 1 – האוטומט הסופי

פרק 1 – אוטומט סופי דטרמיניסטי .

מושגים ומילות מפתח :

אוטומט סופי דטרמיניסטי הוא מודל של מערכת המגיבה על סדרות של קלטים . בזמן נתון , מערכת כזו נמצאת במצב אחד מתוך קבוצה סופית של מצבים , והמעבר ממצב בו נמצאת המערכת למצב חדש מתרחש בהתאם לסימן הבא שמגיע מהקלט .

סופי – משום שמספר מצביו הוא סופי וגם מספר המעבריםשבו .

דטרמיניסטי – משמעו שאין לאוטומט זה "אפשרות בחירה". בהימצאו במצב מסוים ובהגיע קלט מסוים ישנו בדיוק מעבר אפשרי אחד .

מסלול חישוב סדרת המעברים המתבצעים באוטומט עבור סדרת קלט מסוימת (מהמצב ההתחלתי ועד למצב שאליו מגיעים עם תום קריאת הסדרה) .
מסלול חישוב יכול להיות **מקבל** או **לא מקבל** .

טבלת מעברים בטבלת מעברים אנו מתאימים לכל מצב (שורה) ולכל אות קלט (עמודה) מצב חדש .

פונקצית מעברים טבלת מעברים מייצגת למעשה התאמה , המתאימה לכל זוג של מצב ואות קלט - מצב חדש . מאחר שהתאמה זו מתאימה לכל זוג כזה מצב אחד ויחיד , הרי היא התאמה חד ערכית , כלומר , <u>פונקציה</u> .

מילת קלט סדרה של קלטים .

א"ב כל אותיות הקלט – סימנים כלשהם - האפשריות עבור האוטומט . (אותיות , אלפבית , א"ב) .

שפה אוסף של מילים מתקבלות . השפה מכילה את כל המילים שבקריאתן מגיע האוטומט הסופי הדטרמיניסטי למצב מקבל .

שפה פורמלית השפות שמקבלים אוטומטים . להבדיל משפות טבעיות .

מצב	מעבר	קלט	מצב התחלתי	מצב מקבל
מצב מלכודת	לולאה עצמית			

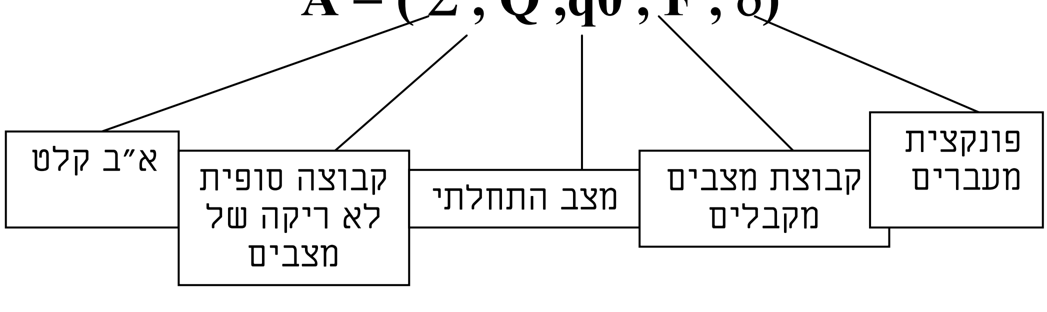
<u>הגדרת אוטומט סופי דטרמיניסטי</u>

לאוטומט סופי דטרמיניסטי יש חמישה מרכיבים :

1. **א"ב** כל אותיות הקלט – סימנים כלשהם - האפשריות עבור האוטומט. מספר אותיות זה חייב להיות סופי וגדול מ –0 .
2. **מצבים** כל המצבים בהם יכול האוטומט להימצא . מספר המצבים חייב להיות סופי וגדול מ –0 .
3. **מצב התחלתי** אחד המצבים , שממנו מתחיל האוטומט את מסלול החישוב על כל מילת קלט .
4. **קבוצת מצבים מקבלים** קבוצה מתוך קבוצת המצבים , המכילה 0 מצבים או יותר .
5. **פונקצית המעברים** פונקציה שמציינת עבור כל זוג – מצב של האוטומט ואות מא"ב – מצב (אחד ויחיד) שאליו עוברים תוך קריאת האות המסוימת מהמצב המסוים .

* ההגדרה דורשת במפורש כי קבוצת המעברים של האוטומט היא סופית אך אינה מתייחסת לסופיות קבוצת המעברים . סופיות קבוצת המעברים נובעת מסופיות קבוצת המצבים וסופיות קבוצת הקלט .

$$A = (\Sigma , Q ,q0 , F , \delta)$$



בנית אוטומט בבניית אוטומט ניתן לתאר את האוטומט בשתי דרכים :
א – על ידי תיאור גרפי .
ב – על ידי טבלת מעברים , מצב התחלתי וקבוצת מצבים מקבלים .

דוגמה 1:

לפניך השפה L מעל הא"ב {a,b} :

$$L = \{a^n b^m \mid n,m > 0\}$$

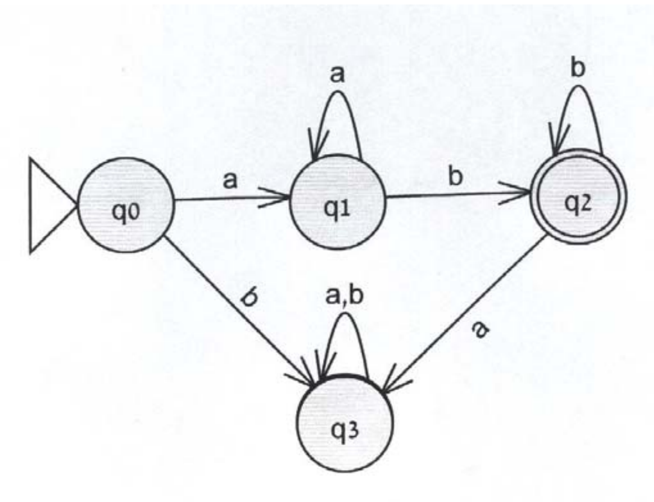
בנה אוטומט סופי דטרמניסטי המקבל את השפה L .

תשובה :

הא"ב : {a,b}
המצבים : q0,q1,q2,q3
המצב ההתחלתי : q0
קבוצת המצבים המקבלים : q2

טבלת מעברים

	a	b
Q0	Q1	Q3
Q1	Q1	Q2
Q2	Q3	Q2
Q3	Q3	Q3



פונקציית מעברים

f(q0,a)=q1 , f(q0,b)=q3, f(q1,a)=q1 ,
f(q1,b)=q2 , f(q2,a)=q3, f(q2,b)=qb ,
f(q3,a)=q3, f(q3,b)=q3

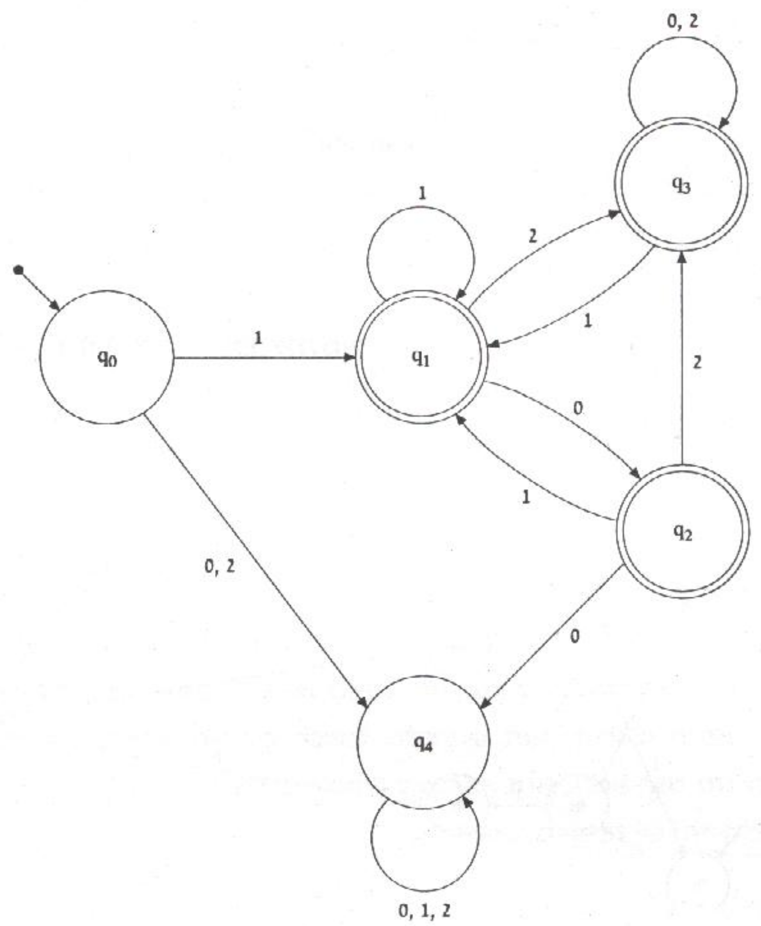
q0 – מצב התחלתי
q1 – זוכר כי המילה התחילה ב- a וכי האות האחרונה שנקראה היא a .
q2 – זוכר כי המילה התחילה ברצף כלשהו של a וכי האות האחרונה שנקראה היא b
q3 – זוכר כי המילה תחילה ב- b , או שהתקבלה a לאחר b זהו מצב מלכודת .

דוגמה 2:

הא"ב הוא 0,1,2 והשפה היא שפת כל המילים שמתחילות ב-1 ולא מופיע בהן רצף האותיות 100.
בנה אוטומט סופי דטרמניסטי המקבל את השפה .

תשובה :

- q_0 – המצב ההתחלתי.
- q_1 – זוכר כי המילה התחילה באות 1 וכי האות האחרונה שנקראה היא 1.
- q_2 – זוכר כי המילה התחילה באות 1 וכי שתי האותיות האחרונות שנקראו הן 10.
- q_3 – זוכר כי המילה התחילה באות 1 וכי האותיות האחרונות שנקראו אינן מהוות התחלה של הרצף.
- q_4 – מצב מלכודת שאליו מגיעים אם המילה לא התחילה ב-1 או אם נקרא הרצף 100.



פרק 3 – מילים ושפות פורמליות .

א"ב כל אותיות הקלט – סימנים כלשהם - האפשריות עבור האוטומט.

מילים :

מילה סדרה של אותיות מא"ב נתון הרשומות משמאל לימין .למשל כשהא"ב הוא {a,b} אזי aa , aba , abbaa הן מילים המורכבות האותיות הא"ב . נהוג לומר שאלו מילים **מעל** א"ב זה .

אורך מילה אורך מילה הוא מספר האותיות בה . הוא מסומן ב- $|\mathbf{w}|$ למשל $|abba| = 4$.

מילה ריקה מילה ריקה היא סדרה באורך של 0 אותיות . תפקידה מקביל לתפקידו של המספר 0 במספרים הטבעיים .

פעולות על מילים

שרשור של שתי מילים שרשורן של שתי מילים הוא מילה הנוצרת מהדבקת המילה השניה מימין למילה הראשונה . למשל , שרשור של ab עם abb יוצר את המילה ababb .
באופן כללי אם w1 ו- w2 הן שתי מילים , נסמן את שרשורן כך :
$\mathbf{w1 \cdot w2}$

חזקה של מילה הוא שרשור של מילה ,w, לעצמה . מסומן ב- $\mathbf{w}^3$. לדוגמה $(aba)3 = aba\cdot aba\cdot aba = abaabaaba$.
w^0 משמעו 0 של מופעים של w, כלומר - מילה ריקה $\mathbf{w=\varepsilon}$.

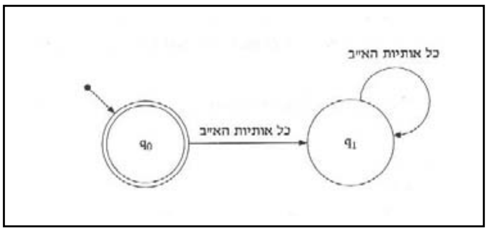
היפוך של מילה היפוך של מילה הוא המילה הנוצרת מהיפוך סדר האותיות . למשל היפוך של המילה aaba הוא המילה abaa . היפוך של המילה הריקה הוא המילה הריקה .
נסמן את היפוך של מילה w כך : $\mathbf{R(w)}$.
לדוגמה : $R(aab) = baa$.

שפות פורמליות :

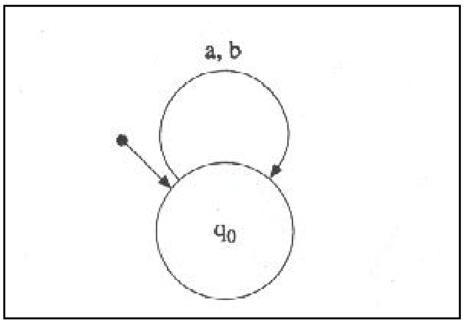
שפה פורמלית
שפה פורמלית היא קבוצה של מילים מעל א״ב מסוים . אנו אומרים כי גם גם השפה היא מעל א״ב זה .
שפה יכולה להכיל מספר כלשהו של מילים – מספר סופי של מילים , אינסוף מילים או אף לא מילה אחת .

דוגמאות :
השפה {aba} היא שפה סופית המכילה רק מילה אחת - aba .
השפה {ε} היא שפה בת מילה אחת – המילה הריקה .
השפה Φ מסמן שפה שאינה מכילה אף מילה .
שפת כל המילים מעל א״ב {a,b} שאורכן זוגי (זו שפה אינסופית)
רישום פורמלי של שפה זו כקבוצה יכול להיות - {|w| זוגי | w}

אוטומט המקבל רק את המילה הריקה



אוטומט המקבל את השפה הריקה



פעולות על שפות

שרשור של שפות
שרשור של שתי שפות הוא השפה המכילה את **כל** אפשרויות השרשור של מילה מהשפה הראשונה עם מילה מהשפה השניה . גם כאן – בדומה לשרשור מילים – יש חשיבות **לסדר** השרשור .

{ab,abb} · {ba,bb,aba} = {abba,abbb,ababa,abbba,,abbbb,abbaba}

חזקה של שפה
שרשור של שפה L לעצמה מסומן ב- L^2 . L^0 מוגדרת כשפה {∈} ולא כ- Φ .

$L = \{ab,a\}$ $\quad L^2 = \{abab,aba,aab,aa\}$

היפוך של שפה
היפוך של שפה L הוא השפה הנוצרת מהיפוך של כל המילים בשפה .
שפה זו מסומנת ב- R(L) .

R{abb,ab} = {R(abb),R(ab)} = {bba,ba}

דוגמה 3 :

נתונות השפות הבאות מעל הא״ב {a,b}:

$$L_3=\{a^n b^m \mid n \ge m+2\} \qquad L_4=\{b^n \mid n \ge 2\}$$

מהי השפה $L_4 \cdot R(L_3)$?

תשובה :

$$L_4 \cdot R(L_3)=\{ b^k \mid k \ge 2\}\cdot\{ b^m a^n \mid n \ge m+2\}=\{ b^k b^m a^n \mid k \ge 2, n \ge m+2\}=$$
$$\{ b^{k+m} a^n \mid k \ge 2, n \ge m+2\}$$

שפה רגולרית ואוטומט סופי דטרמיניסטי

שפה רגולרית — שפה נקראת רגולרית אם אפשר לבנות אוטומט סופי דטרמיניסטי שמקבל אותה .

לאוסף כל השפות הרגולריות קוראים משפחת השפות הרגולריות .

תכונות של משפחת השפות הרגולריות :

שפה חלקית — לא בהכרח רגולרית

שפה L_2 נקראת חלקית לשפה L_1 אם כל המילים שב- L_2 נמצאות גם ב- L_1 . מסמנים זאת כך : $L_2 \subseteq L_1$

$L_1=\{aba,ab,bbb,aab\}$, $L_2=\{ab,bbb\}$. L_2 חלקית ל- L_1 .

אם L_1 שפה רגולרית מעל א"ב מסוים ו- $L_2 \subseteq L_1$ - כלומר L_2 חלקית ל- L_1 - לא בהכרח ש L_2 גם רגולרית .

> על מנת להוכיח כי הטענה נכונה בכל מקרה , עלינו להראות כי לכל א"ב ולכל שתי שפות L1 ו- L2 כך ש- $L_2 \subseteq L_1$ ו- L_1 רגולרית , חייב להתקיים ש- L_2 היא שפה רגולרית . כדי להפריך את הטענה די למצוא דוגמה אחת שבה הטענה אינה מתקיימת , כלומר , טענה נגדית .
>
> $L_1=\{a^n b^m\}$ $\quad$ $L_2=\{a^n b^n\}$
>
> L_1 רגולרית L_2 חלקית ל- L_1 אך היא אינה רגולרית

שפת המשלים — רגולרית

שפת המשלים של L (L היא שפה רגולרית מעל א"ב כלשהו) היא שפת כל המילים שאינן ב- L . מסמנים זאת כך : $\overline{L}$.

אם L היא שפה רגולרית מעל א"ב מסוים אזי שפת המשלים של L אף היא רגולרית.

שפת חיתוך — רגולרית

אם קיימות ,לדוגמה , שתי שפות L_1 ו- L_2 שפת החיתוך שלהן היא קבוצת כל המילים ששייכות **גם** ל- L1 **וגם** ל- L_2 . מסמנים זאת כך : $L_2 \cap L_1$

אם L1 היא שפה רגולרית וגם L2 היא שפה רגולרית שפת החיתוך שלהן גם היא רגולרית .

שפת האיחוד — רגולרית

אם קיימות ,לדוגמה , שתי שפות L_1 ו- L_2 שפת האיחוד שלהן היא קבוצת כל המילים ששייכות **או** ל- L_1 **או** ל- L_2 . מסמנים זאת כך : $L_2 \cup L_1$.

אם L1 היא שפה רגולרית וגם L2 היא שפה רגולרית שפת האיחוד שלהן גם היא רגולרית .

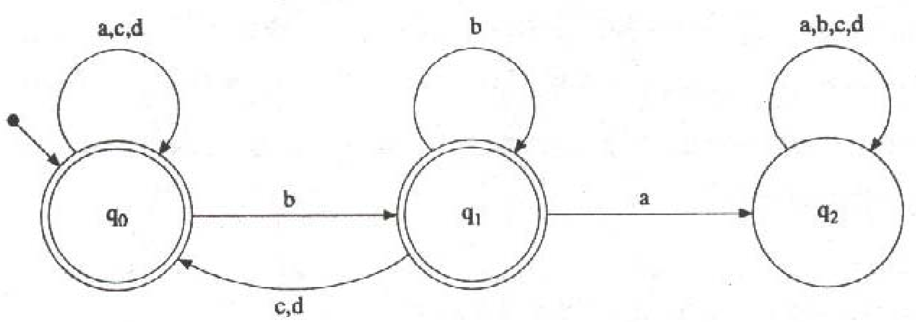
דוגמה 4 : אוטומט מכפלה – טכניקת הבנייה

הא"ב {a,b,c,d}   בנה אוטומט המקבל את שפת כל המילים שנמצאות ב- L1 וגם ב- L2 וכן אוטומט המקבל את שפת כל המילים שנמצאות ב-L1 או ב-L2 .

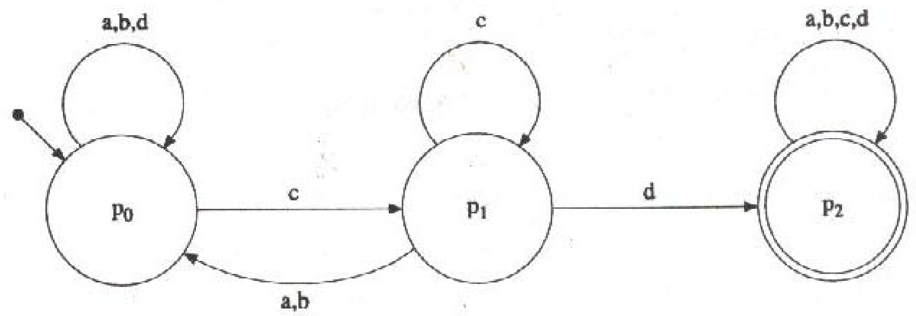
L1=ba שפת כל המילים שאינן מכילות את הרצף
L2 = cd שפת כל המילים שמכילות את הרצף

תשובה :

ב. ראשית, נבנה אוטומטים A_1 ו-A_2 המקבלים בהתאמה את השפות L_1 ו-L_2 :



איור 3.12



נרשום את טבלאות המעברים המתאימות :

A_1:

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

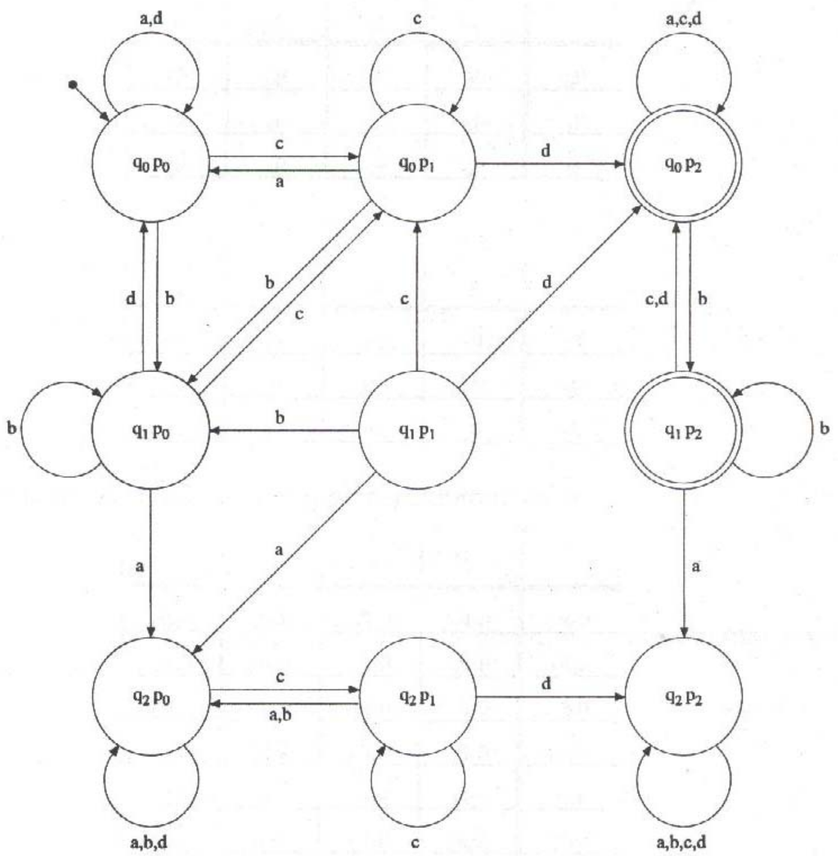
A_2:

	a	b	c	d
p_0	p_0	p_0	p_1	p_0
p_1	p_0	p_0	p_1	p_2
p_2	p_2	p_2	p_2	p_2

נחשב את טבלת המעברים המתאימה לאוטומטים החדשים :

	a	b	c	d
q_0p_0	q_0p_0	q_1p_0	q_0p_1	q_0p_0
q_0p_1	q_0p_0	q_1p_0	q_0p_1	q_0p_2
q_0p_2	q_0p_2	q_1p_2	q_0p_2	q_0p_2
q_1p_0	q_2p_0	q_1p_0	q_0p_1	q_0p_0
q_1p_1	q_2p_0	q_1p_0	q_0p_1	q_0p_2
q_1p_2	q_2p_2	q_1p_2	q_0p_2	q_0p_2
q_2p_0	q_2p_0	q_2p_0	q_2p_1	q_2p_0
q_2p_1	q_2p_0	q_2p_0	q_2p_1	q_2p_2
q_2p_2	q_2p_2	q_2p_2	q_2p_2	q_2p_2

זהו האוטומט המקבל את שפת החיתוך :



ההבדל בין האוטומט המקבל את שפת החיתוך לבין האוטומט המקבל את שפת האיחוד הוא רק בציון המצבים המקבלים .

המצבים המקבלים שיהיו באוטומט האיחוד הם כל המצבים שבהם מופיע q0,q1 או p2. כלומר יהיו **7** מצבים מקבלים .

דוגמה 5 : הוכחת שפה רגולרית באמצעות תכונת הסגירות של המשפחות הרגולריות

נתבונן בשפה מעל הא"ב $\{a,b,c\}$ המכילה את כל המילים שיש בהן מספר זוגי של אותיות 0 ומספר אי זוגי של אותיות 1 , ואינן מכילות את הרצף 0,1 , ואף לא את הרצף 20. האם שפה זו היא רגולרית ? הוכח את תשובתך . (תרגיל 3.25 עמוד 101)

תשובה :

תחילה, נשים לב כי אפשר להציג את השפה הנדונה בתרגיל גם באופן הבא:
$L = (L_1 \cap L_2) \cap (\overline{L_3} \cap \overline{L_4})$, כאשר

$L_1 = \{w \mid$ w מכילה מספר זוגי של 0 $\}$

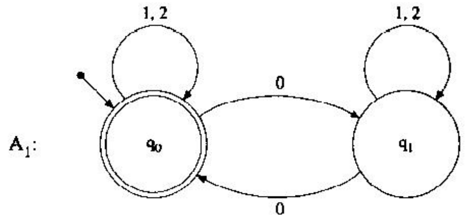
$L_2 = \{w \mid$ w מכילה מספר אי-זוגי של 1 $\}$

$L_3 = \{w \mid$ w מכילה את הרצף 01 $\}$

$L_4 = \{w \mid$ w מכילה את הרצף 20 $\}$

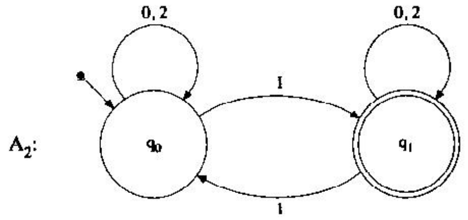
L_1, L_2, L_3 ו-L_4 הן שפות רגולריות, כי ניתן לבנות אוטומט מתאים עבור כל אחת מהן;

A_1 לקבלת L_1:



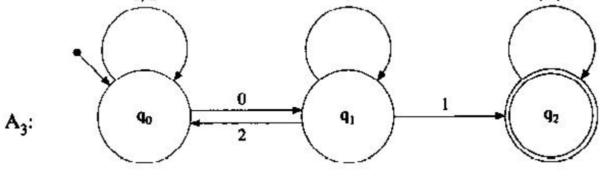
איור 3.29

A_2 לקבלת L_2:



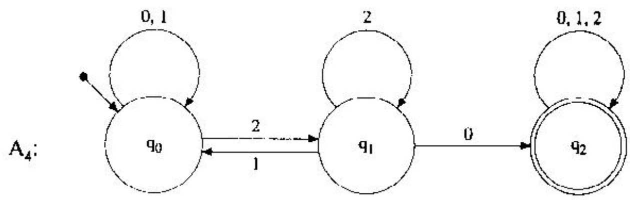
איור 3.30

A_3 לקבלת L_3:



איור 3.31

A_4 לקבלת L_4:



איור 3.32

מאחר שמשפחת השפות הרגולריות סגורה תחת פעולת המשלים, גם השפות $\overline{L_3}$ ו- $\overline{L_4}$ הן רגולריות. מסגירות משפחת השפות הרגולריות תחת חיתוך, נובע כי השפה $\overline{L_3} \cap \overline{L_4}$ היא רגולרית וגם השפה $L_1 \cap L_2$ היא רגולרית. ולבסוף, מסגירות משפחת השפות הרגולריות תחת חיתוך, נובע כי גם השפה $L = (L_1 \cap L_2) \cap (\overline{L_3} \cap \overline{L_4})$ היא רגולרית.

בכך הוכחנו כי השפה שהוצגה בדוגמה זו היא רגולרית, בלי לבנות אוטומט שמקבל אותה.

דוגמה להוכחת אי-רגולריות של שפה

שאלה :

האם השפה $\{a^nb^n \mid n\geq 0\}$ מעל הא"ב $\{a,b\}$ היא רגולרית? הוכח את תשובתך .

הערה :

אינטואיטיבית , היא אינה רגולרית כי אוטומט שצריך לבדוק אם במילה נתונה מספר אותיות a שווה למספר אותיות b , צריך לספור את אותיות a ולהשוות את התוצאה למספר אותיות b וספירה זו מחייבת מספר אין סופי של מצבים . אולם טיעון אינטואיטיבי אינו הוכחה .

תשובה :

נוכיח שהשפה L אינה רגולרית על ידי הוכחה בדרך השלילה .
נניח שהשפה L היא רגולרית . לכן קיים אוטומט סופי שמקבל אותה , ונניח ששמו A .
נסתכל על קבוצת המילים $\{an \mid n\geq 0\}$ (כלומר על הקבוצה $\{\varepsilon, a, aa, aaa, aaaa, \ldots\}$) . אנו טוענים שעל כל מילה בקבוצה זו , האוטומט A מגיע למצב שונה . גם את טענה זו נוכיח בדרך שלילה :
אנו מניחים שקיימות בקבוצה שתי מילים ai ו- aj שעליהן האוטומט A מגיע לאותו מצב q .
המילה aibi שייכת לשפה L , כי מספר אותיות a שבה שווה למספר אותיות b . לכן , האוטומט צריך להגיע למצב מקבל , בקבלו מילה זו כקלט .הרישא של המילה ai מביאה את האוטומט למצב q, ולכן הגעת המילה bi במצב q מביאה את האוטומט למצב מקבל .
אך , מכאן נובע שאוטומט זה מקבל גם את המילה ajbi : על פי הנחת השלילה , קריאת הרישא aj מביאה את האוטומט למצב q , וכאמור הגעת המילה bi במצב q מביאה את האוטומט למצב מקבל . אבל המילה ajbi אינה שייכת לשפה L , כי מספר אותיות a שבה אינו שווה למספר אותיות b .
קבלת מילה שאינה שייכת לשפה סותרת את הנחתנו הראשונית – שהאוטומט A מקבל את השפה L - והמסקנה היא שהנחת השלילה שהנחנו (קיום שתי מילים מתוך הקבוצה שעליהן מגיע האוטומט A לאותו מצב q) הייתה שגויה , והאוטומט אכן מגיע למצב שונה על כל מילה בקבוצה.
אבל , בקבוצה זו יש אין ספור מילים , מכאן נובע שלאוטומט זה יש אין ספור מצבים , בסתירה להיותו אוטומט סופי.
מסתירה זו אנו מסיקים , שהנחתנו הראשונית הייתה שגויה , כלומר , לא קיים אוטומט במקבל את השפה L .

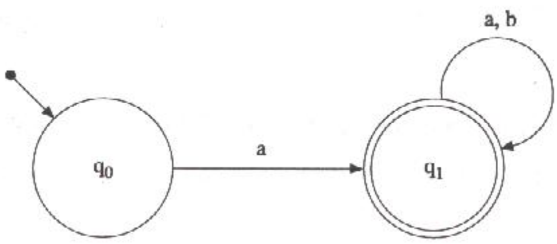
פרק 4 – מודלים נוספים של אוטומט סופי .

אוטומט סופי דטרמיניסטי לא מלא

אוטומט שבו לכל מצב ולכל אות קלט יהיה מעבר אחד לכל היותר (או אחד או אף אחד). כלומר, באוטומט זה מאפשרים קיום משבצות ריקות בטבלת המעברים . יתכן , כי באוטומט כזה, עבור מצב מסוים ואות קלט מסוימת , לא מותאם מעבר . במצב כזה האוטומט "נתקע" .

דוגמה 6 : אוטומט סופי דטרמניסטי המקבל את שפת כל המילים המתחילות ב- a

נתייחס לא"ב {a, b}. באיור 4.3 מתואר אוטומט סופי דטרמיניסטי לא מלא, המקבל את שפת כל המילים המתחילות באות a.



איור 4.3

באוטומט זה, אם אות הקלט הראשונה היא a, מתבצע מעבר למצב המקבל q_1, ובו נשאר האוטומט עד סיום קריאת מילת הקלט. כלומר, כל מילה שמתחילה באות a מתקבלת. אם האות הראשונה היא b, האוטומט נתקע, ולכן מילה שמתחילה ב-b אינה מתקבלת.

הנה טבלת המעברים של האוטומט :

	a	b
q_0	q_1	
q_1	q_1	q_1

כפי שיכולנו לצפות, לא כל המשבצות בטבלה מלאות. משבצת ריקה מציינת היתקעות של האוטומט.

הגדרת אוטומט סופי דטרמיניסטי לא מלא

לאוטומט סופי דטרמיניסטי לא מלא יש חמישה מרכיבים :

1. **א"ב** – כל אותיות הקלט – סימנים כלשהם - האפשריות עבור האוטומט. מספר אותיות זה חייב להיות סופי וגדול מ –0 .
2. **מצבים** – כל המצבים בהם יכול האוטומט להימצא . מספר המצבים חייב להיות **סופי** וגדול מ –0 .
3. **מצב התחלתי** – אחד המצבים , שממנו מתחיל האוטומט את מסלול החישוב על כל מילת קלט .
4. **קבוצת מצבים מקבלים** – קבוצה מתוך קבוצת המצבים , המכילה 0 מצבים או יותר .
5. **קבוצת המעברים** – קבוצת שלשות . כל שלשה מורכבת ממצב , אות קלט , ומצב . (משמעות שלשה (qi,x,qj) היא כאשר האוטומט נמצא במצב qi ונקראת האות x הוא עובר למצב qj) בקבוצה זו לא קיימות שתי שלשות (qi,x,qj) ו- (qk,y,qe) כך שמתקיים qi=qk ו- x=y (כלומר אין יותר ממעבר אחד לכל זוג של מצב ואות קלט)

כאן אין אפשרות לדבר על פונקצית מעברים (כמו בהגדרה של אוטומט דטרמיניסטי מלא) וזאת משום שייתכן כי יהיו זוגות עבורם לא יותאם מצב .

הגדרה : קבלה ודחייה של מילים על ידי אוטומט סופי דטרמיניסטי לא מלא

א. אוטומט סופי דטרמיניסטי לא מלא מקבל מילה כאשר היא נקראת עד סופה , ובתום קריאתה נמצא האוטומט במצב מקבל .

ב. אוטומט סופי דטרמיניסטי לא מלא דוחה מילה במקרים הבאים :
1. המילה נקראת עד סופה ובתום קריאתה נמצא האוטומט במצב לא מקבל .
2. במהלך קריאת המילה מגיע האוטומט למצב אשר ממו אין מעבר המתאים לאות הקלט הבאה (האוטומט **נתקע**) .

אוטומט סופי לא דטרמניסטי

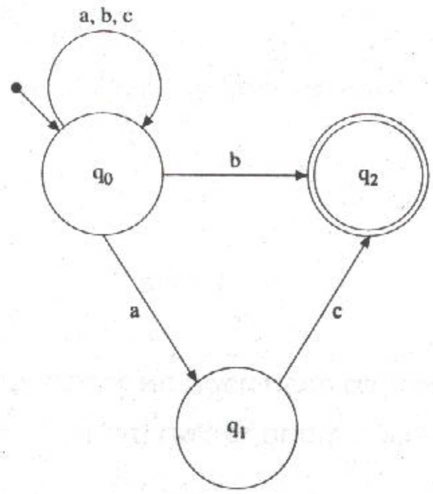
באוטומט סופי לא דטרמיניסטי יכולים להיות למילת קלט <u>כמה</u> מסלולי חישוב אפשריים. וכדי להחליט אם מלה מתקבלת על ידי אוטומט כזה , אנו בוחנים את כל המסלולים ורואים אם יש לפחות אחד מהם שהוא מסלול חישוב מקבל . לפני האוטומט עומדות שתי אפשרויות ועליו ל"נחש" איזו מהן טובה יותר עבורו .

<u>דוגמה 7 : אוטומט סופי לא דטרמניסטי</u>

תאר אוטומט לא דטרמניסטי שמקבל את שפת כל המילים מעל הא"ב $\{a,b,c\}$ שמסתיימות ב- b או ברצף ac

לאוטומט שלושה מצבים :

- q_0 – המצב ההתחלתי.
- q_1 – זוכר כי האות האחרונה שנקראה היתה a, וכי האוטומט ניחש שאות זו מתחילה את הרצף ac ושרצף זה מסיים את מילת הקלט.
- q_2 – זוכר כי האות האחרונה שנקראה היתה b, או כי שתי האותיות האחרונות שנקראו היו ac. האוטומט "מעוניין" להגיע למצב זה בד בבד עם סיום מילת הקלט.



<u>הגדרת אוטומט סופי לא דטרמיניסטי</u>

לאוטומט סופי לא דטרמיניסטי יש חמישה מרכיבים :

1. **א"ב** — כל אותיות הקלט – סימנים כלשהם - האפשריות עבור האוטומט. מספר אותיות זה חייב להיות סופי וגדול מ –0 .

2. **מצבים** — כל המצבים בהם יכול האוטומט להימצא . מספר המצבים חייב להיות סופי וגדול מ –0 .

3. **מצב התחלתי** — אחד המצבים , שממנו מתחיל האוטומט את מסלול החישוב על כל מילת קלט .

4. **קבוצת מצבים מקבלים** — קבוצה מתוך קבוצת המצבים , המכילה 0 מצבים או יותר .

5. **קבוצת המעברים** — קבוצת שלשות . כל שלשה מורכבת ממצב , אות קלט , ומצב . (משמעות שלשה (qi,x,qj) היא כאשר האוטומט נמצא במצב qi ונקראת האות x הוא עובר למצב qj) .
גם כאן – כמו בהגדרת המרכיב החמישי האוטומט דטרמיניסטי לא מלא – אין אפשרות לדבר על פונקצית מעברים (כמו בהגדרה של אוטומט דטרמיניסטי מלא) וזאת משום שלא וכל זוג של מצב ואות קלט מותאם מצב .

הגדרה <u>: קבלה ודחייה של מילים על ידי אוטומט סופי לא דטרמיניסטי</u>

א. אוטומט סופי לא דטרמיניסטי מקבל מילה אם <u>קיים</u> מסלול חישוב המתאים למילה זו ומסתיים (עם תום המילה) במצב מקבל .
ב. אוטומט סופי לא דטרמיניסטי דוחה מילה אם לא קיים מסלול חישוב מקבל המתאים למילה זו . בכל מסלול חישוב אפשרי למילה זו האוטומט נתקע , או שהמסלול מסתיים במצב שאינו מקבל .

כוחם של המודלים החדשים

האם ישנה שפה **<u>רגולרית</u>** עבורה **<u>לא קיים</u>** אוטומט סופי לא דטרמיניסטי המקבל אותה ?
<u>לא</u> . מכיוון שלמעשה אוטומט דטרמיניסטי הוא מקרה פרטי של אוטומט לא דטרמיניסטי .

האם ישנה שפה **<u>לא רגולרית</u>** עבורה **<u>קיים</u>** אוטומט סופי לא דטרמיניסטי המקבל אותה ?
<u>לא</u> . בכל אוטומט נתון ניתן להיפתר מהאי-דטרמיניסטים ולהפוך את האוטומט לדטרמיניסטי .

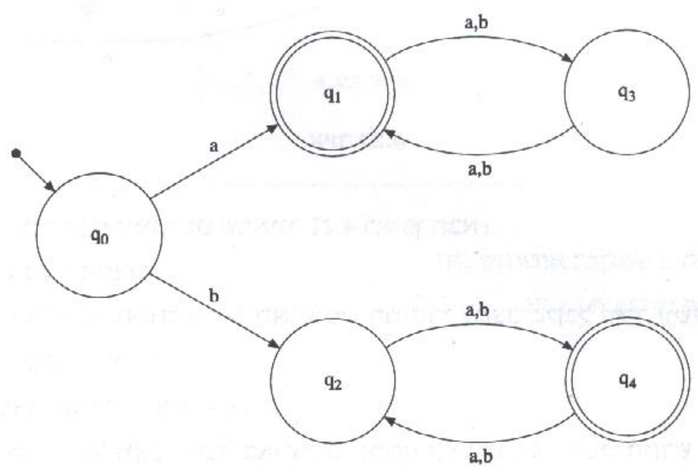
תכונות נוספות של משפחת השפות הרגולריות

משפחת השפות הרגולריות סגורה ל- : משלים
חיתוך
איחוד
<u>היפוך</u>
<u>שרשור</u>

דוגמה לבניית אוטומט היפוך

א. מהי השפה L שמקבל האוטומט הסופי הדטרמיניסטי שבאיור 4.24?

ב. תאר אוטומט סופי שמקבל את R(L).



תשובה :

א. השפה שאוטומט זה מקבל היא :

L = { a·w | w מילה מעל הא"ב {a, b} שאורכה זוגי } ∪

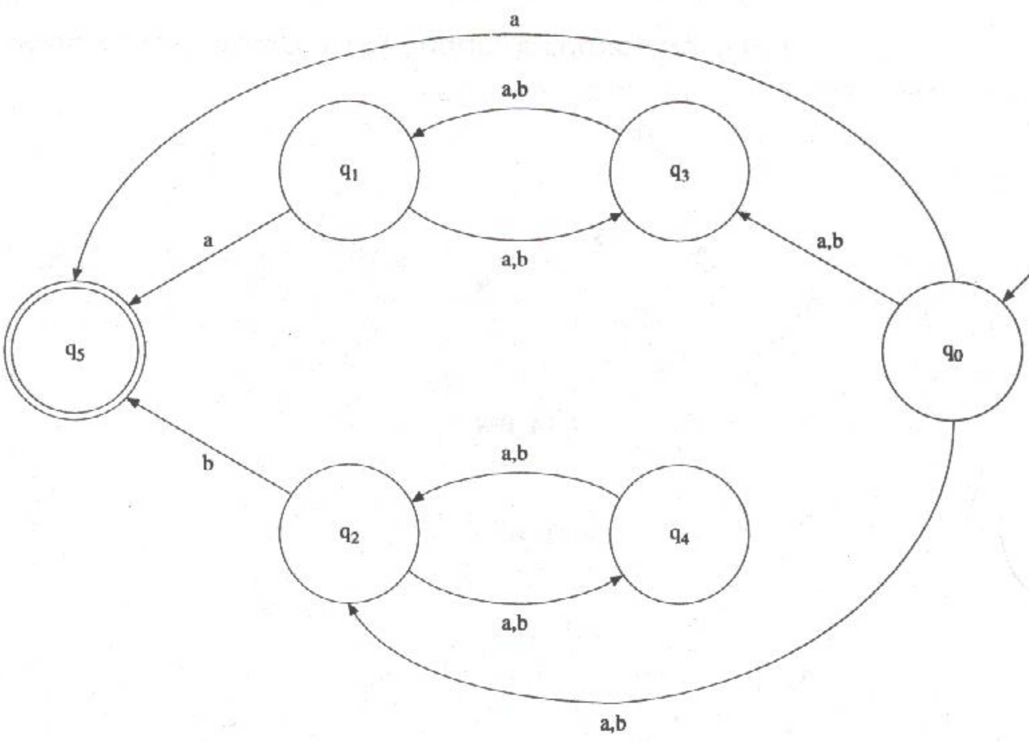
{ b·w | w מילה מעל הא"ב {a, b} שאורכה אי-זוגי }

ב. השפה ההפוכה היא :

R(L) = { w·a | w מילה מעל הא"ב {a, b} שאורכה זוגי } ∪

{ w·b | w מילה מעל הא"ב {a, b} שאורכה אי-זוגי }

נפעיל את הטכניקה לבניית אוטומט שמקבל את שפת ההיפוך, כדי לבנות אוטומט שמקבל את R(L) :



דוגמה לבניית אוטומט שרשור

נתונות השפות $L_1 = \{$ w | aba מכילה את הרצף w $\}$ מעל הא״ב {a, b}
$L_2 = \{$ w | באורך זוגי w $\}$ מעל הא״ב {b, c}

מהי שפת השרשור $L_1 \cdot L_2$, ומעל איזה א״ב?

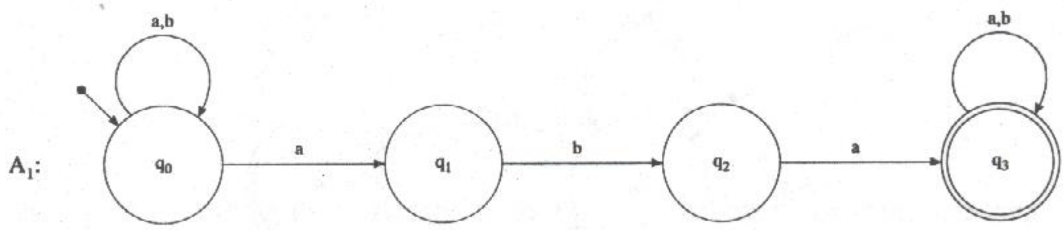
בנה אוטומט המקבל את שפת השרשור.

תשובה

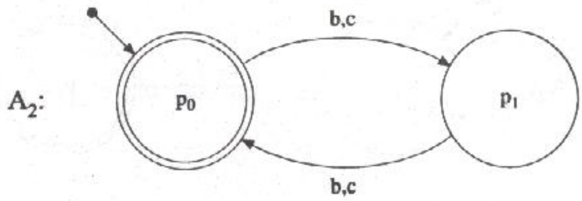
שפת השרשור היא מעל הא״ב {a, b, c} וזוהי השפה הבאה:

$L_1 \cdot L_2 = \{$ $w_1 \cdot w_2$ | w_1 מעל {a, b} מכילה את הרצף aba, w_2 מעל {b, c} באורך זוגי $\}$

ראשית, נתאר אוטומטים A_1 ו-A_2, המקבלים בהתאמה את L_1 ו-L_2:

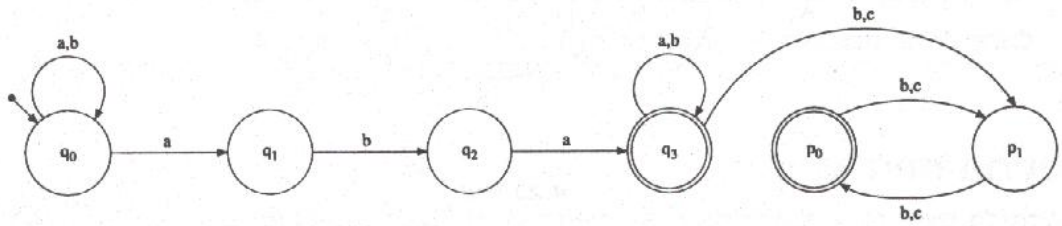


איור 4.18



איור 4.19

נפעיל את הטכניקה לבניית אוטומט המקבל את שפת השרשור:



איור 4.20

שימו לב, כי באוטומט זה גם q_3, שהיה מצב מקבל ב-A_1, הוא מצב מקבל באוטומט השרשור (כי ב-A_2 המצב ההתחלתי הוא מצב מקבל), וכי באוטומט השרשור המצב p_0 והמעברים היוצאים ממנו אינם מיותרים (ואכן, ב-A_2 יש מעברים הנכנסים אל p_0).

חלק 2 - אוטומט המחסנית

פרק 5 – אוטומט המחסנית

אוטומט מחסנית לא דטרמיניסטי

אוטומט מחסנית לא דטרמיניסטי , אף הוא כמו אוטומטיים סופיים , מודל של מערכת המגיבה על סדרות סופיות של קלטים הנקראות מילים .
בזמן נתון נמצאת מערכת זו במצב אחד מתוך קבוצה סופית של מצבים , ובמחסנית יש סדרה (אולי ריקה) של אותיות . עם קריאת אות קלט מתבצע מעבר ממצב למצב , תוך כדי ביצוע שינוי במחסנית .

הגדרה : אוטומט מחסנית לא דטרמיניסטי

לאוטומט מחסנית לא דטרמיניסטי יש שישה מרכיבים :

1. **א"ב הקלט** – כל אותיות הקלט – סימנים כלשהם - האפשריות עבור האוטומט. מספר אותיות זה חייב להיות סופי וגדול מ –0 .
2. **א"ב המחסנית** – האותיות שאותן יכול האוטומט לדחוף למחסנית . זהו מרכיב חדש , שלא היה קיים בהגדרת אוטומט סופי . גם מספר אותיות אלה סופי .
3. **מצבים** – כל המצבים בהם יכול האוטומט להימצא . מספר המצבים חייב להיות סופי וגדול מ –0 .
4. **מצב התחלתי** – אחד המצבים , שממנו מתחיל האוטומט את מסלול החישוב על כל מילת קלט .
5. **קבוצת מצבים מקבלים** – קבוצה מתוך קבוצת המצבים , המכילה 0 מצבים או יותר .
6. **קבוצת המעברים** – קבוצת חמישיות . כל חמישיה מורכבת ממצב , אות קלט , אות מא"ב המחסנית (או סימן שהמחסנית ריקה) מצב ומילה מעל א"ב המחסנית.

<u>דוגמה לתשובה בנושא אוטומט מחסנית</u>

<u>שאלה</u>

נתונה השפה L מעל הא"ב {a,b,c} : $L= (a^n b^m c^m a^k \mid n>0 , n>k , m>0 \}$

הגדר אוטומט מחסנית המקבל את השפה

<u>תשובה</u>

1) **א"ב הקלט** - {a,b,c}

2) **א"ב המחסנית** - { S ,A , T , B , ⊥ }

⊥ - מסמן שהמחסנית ריקה .

S - כדי לסמן קריאת ה-a הראשונה .

A - כדי לסמן קריאת a שאינה ראשונה .

T - כדי לסמן קריאת ה-b הראשונה .

B - כדי לסמן קריאת b שאינה ראשונה .

3) **המצבים** - q0 , q1 , q2 , q3

4) **המצב ההתחלתי** q0

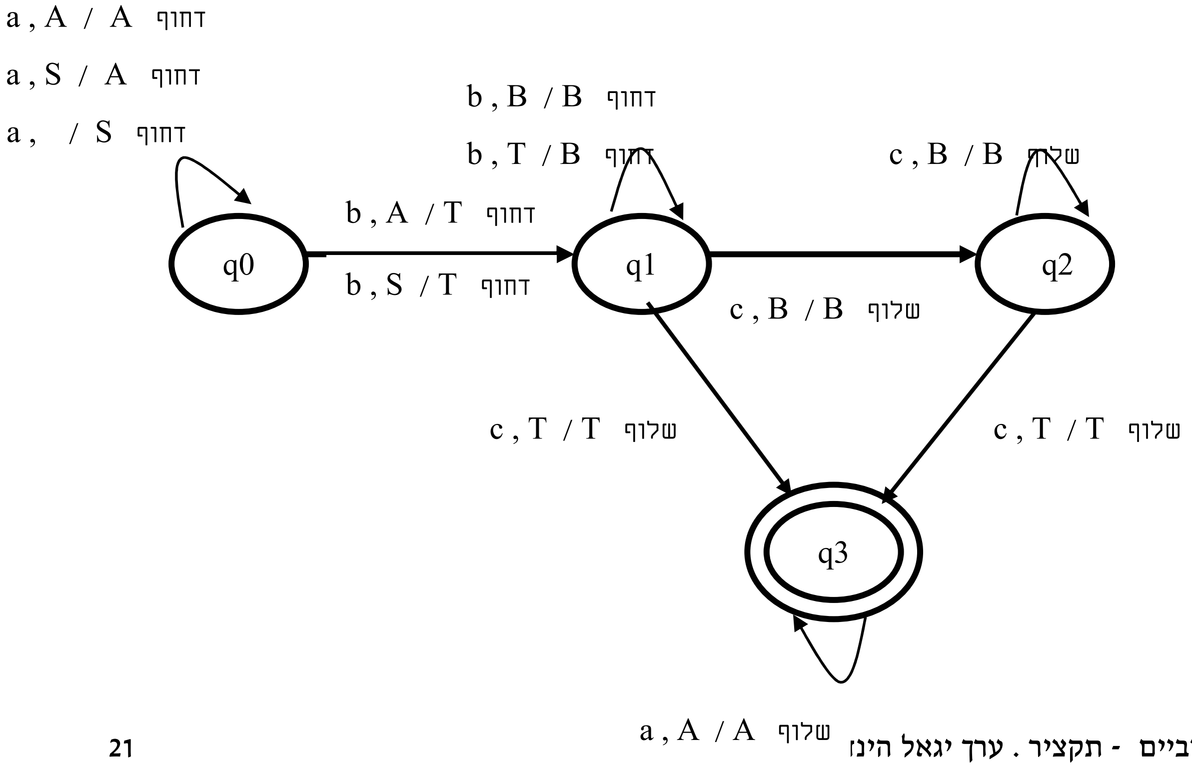
5) **המצבים המקבלים** q3

6) **קבוצת המעברים:**

(q0 , a , ⊥ , q0 , S)
(q0 , a , S , q0 , SA)
(q0 , a , A , q0 , AA)
(q0 , b , S , q1 , ST)
(q0 , b , A , q1 , AT)
(q1 , c , T , q3 , ε)
:
:

השלם את קבוצת המעברים

הערה : ללא שינוי יסומן כך :
(q5 , a , A , q6 , A)



<u>פרק 6 – כוחו ומגבלותיו של אוטומט המחסנית</u>

כוח החישוב של אוטומט המחסנית הלא דטרמיניסטי גדול מזה של האוטומט הסופי – ניתן לעשות בעזרת מודל אוטומט המחסנית הלא דטרמיניסטי <u>כל מה שניתן לעשות</u> בעזרת מודל האוטומט הסופי <u>ויותר</u> .

לדוגמה : $\{ a^n b^n | n \geq 0 \}$

כוח החישוב של אוטומט חישוב דטרמיניסטי ואוטומט חישוב לא דטרמיניסטי <u>**אינו זהה**</u> . לאוטומט מחסנית דטרמיניסטי יש כוח חישוב גדול יותר מזה של האוטומט הסופי , אך לאוטומט מחסנית לא דטרמיניסטי כוח חישוב גדול עוד יותר.

לדוגמה : { w מילה מעל הא"ב {a,b} | $w.R(w)$ }

גם מודל אוטומט המחסנית הינו מוגבל , אפילו בגרסתו הלא דטרמיניסטית . כלומר יש שפות שלא ניצן לקבל בעזרתו .

לדוגמה : $\{ a^n b^n c^n | n \geq 0 \}$

תכונות של משפחת השפות חופשיות ההקשר

שפה חופשית הקשר היא שפה שמתקבלת על ידי אוטומט מחסנית לא דטרמיניסטי .

משפחת השפות חופשיות ההקשר - היא קבוצת כל השפות חופשיות ההקשר .

אופי השפה	האם רגולרית ?		האם חופשית הקשר ?
שפה חלקית	לא		לא
שפת המשלים	כן		לא
שפת החיתוך	כן		לא
שפת האיחוד	כן		כן
שפת ההיפוך	כן		כן
שפת השרשור	כן		כן

חלק 3 - מכונת טיורינג

פרק 7 – מכונת טיורינג

מודל מכונת טיורינג מתקבל על ידי הרחבת מודל אוטומט מחסנית. ההרחבה מתבטאת בגישה חופשית יותר לזיכרון: איננו מוגבלים רק בגישה לתא העליון של המחסנית , וניתן לראות את תוכנו של כל תא ולשנות את תוכנו של כל תא .

הגדרה : מכונת טיורינג

למכונת טיורינג יש שישה מרכיבים :

1. **א"ב הקלט** – כל אותיות שיכולות להיות רשומות על הסרט לפני תחילת תהליך החישוב. מספר האותיות בא"ב זה חייב להיות סופי וגדול מ –0 .

2. **א"ב המכונה** – אותיות נוספות שבהן יכולה המכונה להיעזר במהלך פעולתה . גם מספר אותיות אלה סופי .

3. **מצבים** – כל המצבים בהם יכולה המכונה להימצא . מספר המצבים חייב להיות סופי וגדול מ –0 .

4. **מצב התחלתי** – המצב שממנו מתחילה תמיד המכונה את מסלול החישוב .

5. **קבוצת מצבים מקבלים** – קבוצה מתוך קבוצת המצבים , המכילה 0 מצבים או יותר .

6. **קבוצת המעברים** – קבוצת חמישיות . כל חמישיה מורכבת מהאיברים הבאים :
   - א. מצב
   - ב. אות מא"ב הקלט או מא"ב המכונה או אחד הסימנים Δ $\vdash$.
   - ג. מצב
   - ד. אות מא"ב הקלט , או א"ב המכונה או מאחד הסימנים
   - ה. אחת מההוראות ***ימין*** או ***שמאל*** .

השינויים והתוספות למודל מכונת טיורינג :

א. אנו מציירים את הזיכרון כשהוא פרוס הצידה .

ב. אנו מסמנים במפורש תאים ריקים בעזרת הסימן המיוחד Δ .

. קצה הסרט מסומן על ידי הסימן $\vdash$. לא ניתן למחוק סימן זה ולא ניתן לכתוב אותו בשום מקום אחר על פני הסרט .

ד. יש ראש קורא וכותב , שמצביע על התא שאליו מתייחסים כרגע. אנו מסמנים את מיקום הראש על ידי חץ קטן $\uparrow$.

ה. בתחילת תהליך החישוב, מילת הקלט רשומה על פני הסרט , כשהיא צמודה לקצהו השמאלי .

ו. מכונת טיורינג עוצרת רק כאשרהצעד הבא אינו מוגדר . כאשר המכונה עוצרת , ניתן לקבוע אם מילת הקלט התקבלה או נדחתה , כתלות במצב שבו המכונה נמצאת בזמן עצירתה .

אי עצירה של מכונת טיורינג

תהליכי החישובים באוטומטים סופיים ובאוטומטי מחסנית הם סופיים . האוטומט נעצב בהגיעו לסוף מילת קלט . מכונת טיורינג יכולה לסרוק את מילת הקלט כמה וכמה פעמים ועשוי להיווצר מצב שבו תהיה סריקה אין סופית של הסרט .

חישובים בעזרת מכונות טיורינג

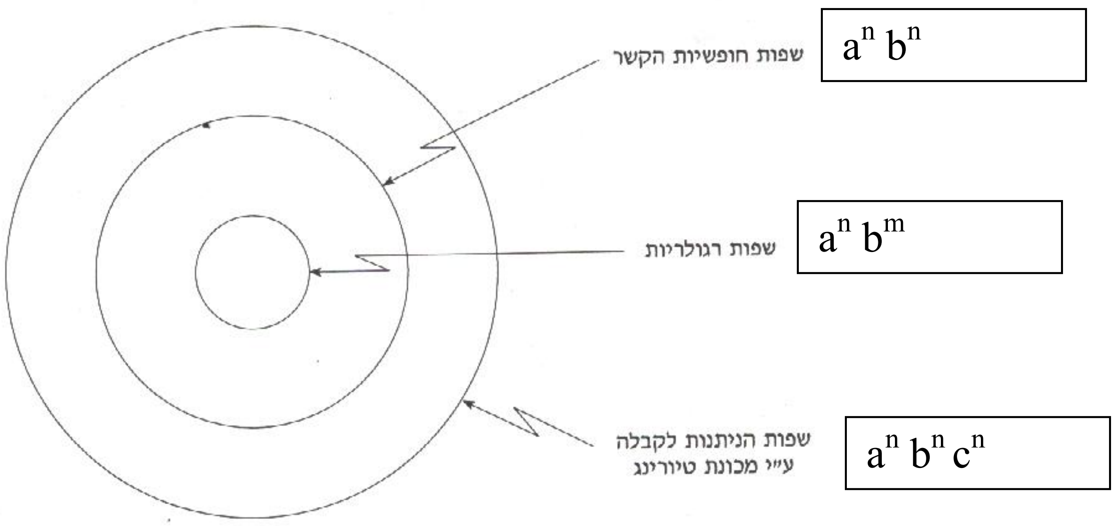
מכונת טיורינג היא כלי שבעזרתו ניתן גם לחשב פונקציות ולא רק להכריע שייכות של מילים לשפות פורמליות .כאשר מכונת טיורינג מחשבת פונקציה , היא מקבלת את הפרמטרים לחישוב על פני הסרט כמילת קלט, ורושמת את התוצאה בסיום החישוב על פני הסרט . שיטת הכתיבה היא אונרית – שימוש בספרה 1 בלבד (להבדיל מבינארית- 0..1 ועשרונית 0..9). בסיום התהליך הפלט יופיע על הסרט בין שני סימני $.

כוחו ומגבלותיו של אוטומט טיורינג

__כוחו של מודל מכונת טיורינג בהשוואה לכוחם של המודלים הקודמים :__

כוח החישוב של מכונת טיורינג גדול מזה של אוטומט המחסנית – ניתן לעשות בעזרת מכונת טיורינג __כל מה שניתן לעשות__ בעזרת מודל אוטומט המחסנית __ויותר__ .

לדוגמה : $\{ a^n b^n c^n | n \geq 0 \}$



__כוחו של מודל מכונת טיורינג לכוחו של מחשב כללי :__

כוחו של מודול מכונת טיורינג שקול למחשב אידיאלי , שאין לו מגבלות זיכרון – זוהי התיזה של צ'רץ וטיורינג . כל בעיה הניתנת לפתרון בעזרת מכונת טיורינג שעוצרת תמיד , קיימת תוכנית מחשב שפותרת אותה , ולכל בעיה בניתנת לפתרון בעזרת תכנית מחשב , קיימת מכונת טיורינג העוצרת תמיד שפותרת אותה .

__המחשב אינו כל יכול :__

לא ניתן לכתוב תוכנית מחשב שמזהה אם תוכנית מחשב נתונה עוצרת כאשר היא רצה על קלט נתון . בעיה זו נקראת בעיית העצירה .