מבחן 2010

מבחן בגרות 2010

פרטים כלליים

מועד הבחינה: בכל זמן

מספר השאלון: 1

משך הבחינה: 3 שעות

חומר עזר בשימוש: הכל (ספרים ומחברות)

המלצות: קרא המלצות לפני הבחינה ובדיקות אחרונות לפני מסירה (עמודים 8-11)

מבנה השאלון

פרק ראשון - עיצוב תוכנה	2 שאלות (בחירה מ-4) לכל שאלה – 25 נקודות	סה"כ 50 נקודות
פרק שני - מודלים חישוביים	2 שאלות (בחירה מ - 4) לכל שאלה – 25 נקודות	סה"כ 50 נקודות

תוכן עניינים של פתרון המבחן

פתרון פרק ראשון - ע"צוב תוכנה 2010

פתרון שאלה 1

נושא מרכזי: מחסנית
סוג השאלה: פיתוח פעולה

מספר פתרונות

נתוני השאלה: כתוב פעולה המקבלת שתי מחסניות st1,st2 שמכילות מספרים שלמים וגדולים מאפס.
הפעולה מחזירה את סכום זוג האיברים הסמוכים מ- st1, הקרובים ביותר לראש המחסנית, הגדול מכל זוגות האיברים הסמוכים במחסנית st2.

גרסת פתרון 1

```
/*
                                    הפעולה מקבלת מחסנית st המכילה מספרים שלמים וגדולים מאפס.
                                    הפעולה מחזירה את סכום זוג האיברים הסמוכים הגדול ביותר ב-st.
                                    הנחה: במחסנית יש לפחות שני איברים
*/
public static int maxStPairSum (Stack< Integer > st)
{
      int a = st.Pop();
      int b = st.Pop();
      int sum = a + b;
      while (!st.isEmpty())
      {
            a = b;
            b = st.Pop();
            if (a + b > sum)
                  sum = a + b;
      }
      return sum;
}
/*  פעולה זו מחזירה את סכום זוג האיברים הסמוכים הקרוב ביותר לראש המחסנית st1 הגדול
מהסכום של כל זוג איברים סמוכים ב-st2.  */

public static int bigPairSum(Stack<integer> st1, Stack<integer> st2)
{
      int max = maxStPairSum(st2);
      int a = st1.Pop();
      int b = st1.Pop();
      if (a + b > max)
            return a + b;
      while (!st1.isEmpty())
      {
            a = b;
            b = st2.Pop();
            if (a + b > max)
                  return a + b;
      }
      return 0;
}
```


גרסת פתרון 2 לפעולה MaxStPairSum

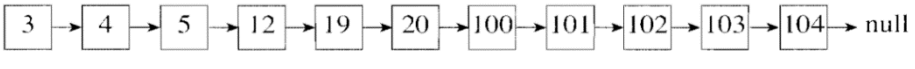
```
public static int  maxStPairSum (Stack<integer> st)
{
      int val1 = st.Pop();
      int val2 = st.Pop();
      int max = val1 + val2;
      while ( ! st.isEmpty() )
     {
           if  (  val2 + st.Top() > max )
               max = val2 + st.Top();
          val2 = st.Pop();
      }
      return max;
}
```

פתרון שאלה 2

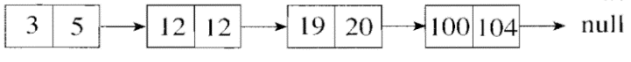
נושא מרכזי: רשימות
סוג השאלה: פיתוח פעולה

נתוני השאלה: L היא רשימה המכילה מספרים שלמים שונים זה מזה וממוינים בסדר עולה.
רשימת הטווחים ב- L היא רשימה חדשה שנבנית באופן הזה: בעבור כל רצף של מספרים עוקבים ב- L יהיה ברשימת הטווחים איבר אחד שמכיל שני מספרים. מספר אחד הוא המספר הקטן ביותר ברצף, והמספר השני הוא המספר הגדול ביותר ברצף.
רצף יכול להיות באורך 1 או יותר. אם הרצף הוא באורך 1, הוא מיוצג ברשימת הטווחים על-ידי איבר ששני המספרים בו שווים.

לדוגמה, בעבור הרשימה L שלפניך:



רשימת הטווחים של L תהיה:



טענת כניסה: הפעולה חיצונית מקבלת רשימה לא ריקה, המכילה מספרים שלמים שונים זה מזה ממוינים בסדר עולה ומחזירה את רשימת הטווחים שלה.

```
/* הפעולה מחזירה את רשימת הטווחים של list*/
public static List<RangeNode> createRangeList(List<int> sourceList)
{
    List<RangeNode> lst = new List<RangeNode>();
    Node<RangeNode> pLst = null;
    Node<int> p = sourceList.getFirst();
    int from, to;
    boolean  in;
    while ( p !=null )
    {
        from = p.getInfo();
        to = from;
        p = p.getNext();
```



```
            in = true;
            while (p != null  &&  in)
            {
                if (p.getInfo() == to + 1)
                {
                    to = p.getInfo();
                    p = p.getNext();
                }
                else
                    in = false;
            }
            pLst = lst.insert (pLst, new RangeNode(from, to));
        }
        return lst;
}
```

פתרון שאלה 3

נושא מרכזי: טיפוסים
סוג השאלה: פיתוח מחלקה ופיתוח פעולות

א. כותרת המחלקה Message המייצגת הודעה ב- MessageBox ואת התכונות שלה:

```
class  Message
{
        private int counter;              // מספור להודעות החדשות
        private String sender;            // שם השולח
        private String content;           // תוכן ההודעה
        private int id;                   // מספר סידורי
        private int size;                 // גודל ההודעה
}
```

ב. כותרת המחלקה MessageBox והתכונות שלה:

```
class  MessageBox
{
        private String name;                    // שם בעל התיבה
        private List<Message> inbox;            // רשימת ההודעות הפעילות
        private List<Message> bin;              // רשימת ההודעות הסל האשפה
}
```

ג. מימוש הפעולה addMessage המוצגת בממשק המחלקה MessageBox

```
public boolean addMessage (Message m)
{
        if (getActiveSize() + getBinSize() + m.getSize() < =100)
        {
                inbox.insert (null, m);
                return true;
        }
        if (getBinSize()>=m.getSize())
        {
                int sum=-0;
                while (sum<=m.getSize())
                        sum+=removeFromBin();
                inbox.insert (null, m);
```

```
            return true;
        }
        return false;
    }
```

ד. לאחר שגולשים פטפטנים רבים ביקשו להגדיל את התיבה, הוחלט שפעם ביום תעבור המערכת על כל התיבות של הגולשים ותקטין או תגדיל את התיבות שלהם לפי הפעילות שלהם: תיבה שהתפוסה שלה יותר מ-80% תוגדל פי 2, ותיבה שהתפוסה שלה פחות מ-30% תוקטן בחצי.

(1) לאור החלטה זו צריך לשנות את הפעולה שנכתבה בסעיף ב.

השינויים: יש להוסיף תכונה שתייצג את גודל התיבה, ולאתחל את התכונה ל-100 בפעולה הבונה.

(2) לאור החלטה זו צריך לשנות את הפעולה שנכתבה בסעיף ג.

הסיבה: מאחר וכעת MaxSize שומר לכל תיבת דואר את הגודל שלה יש להשתמש בפעולה המאחזרת את הגודל.

פתרון שאלה 4

נושא מרכזי: תור
סוג השאלה: מעקב וניתוח פעולות

נתוני השאלה: נתון התור myQueue מטיפוס Queue<Integer>:

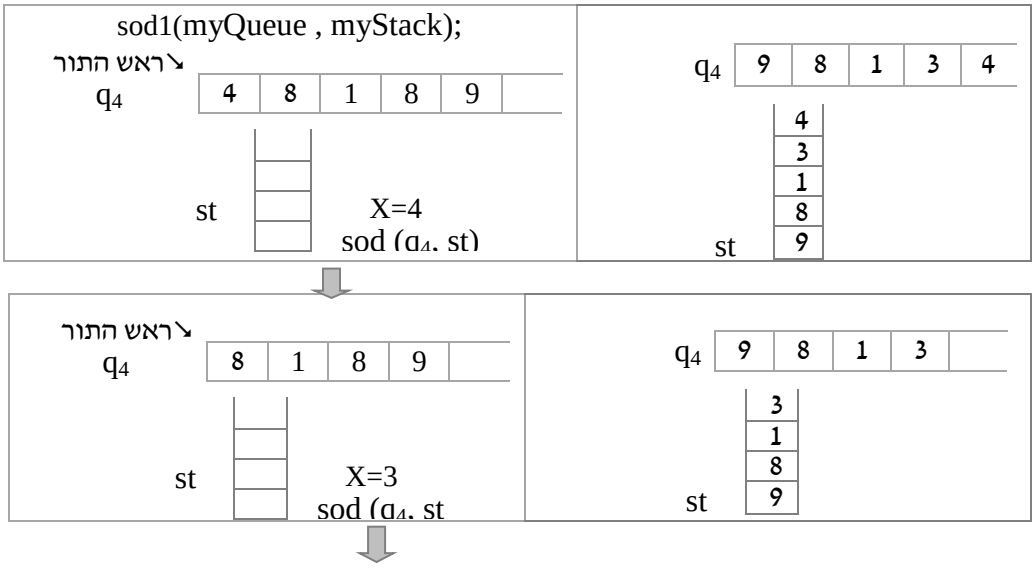
הוצאת ערכים → | 9 | 8 | 1 | 3 | 4 | ← הכנסת ערכים

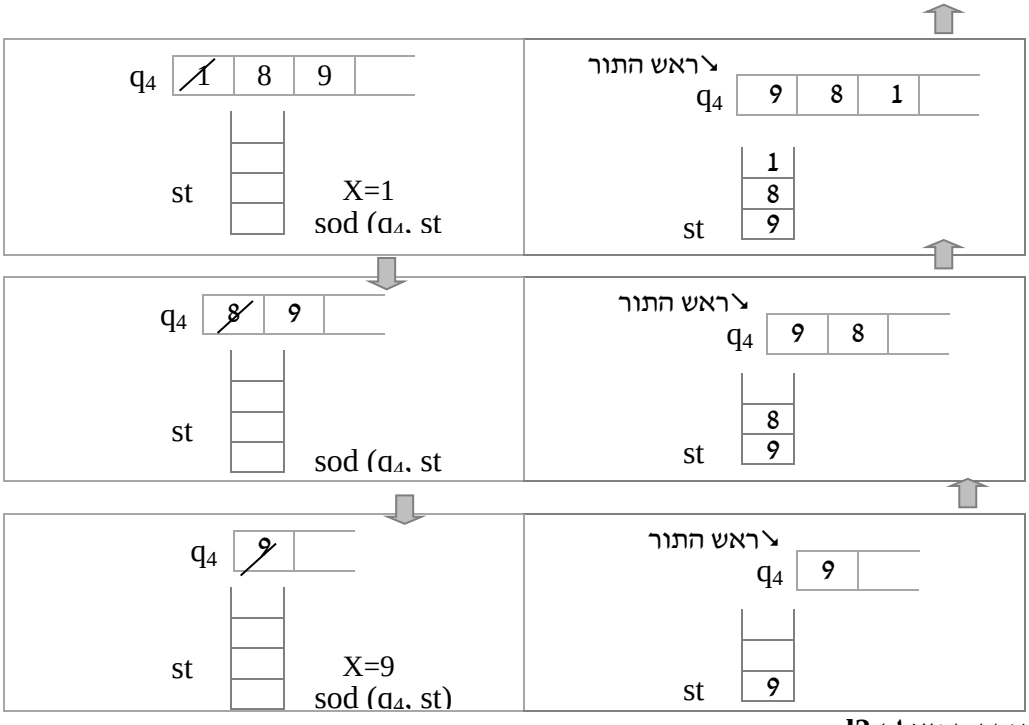
ונתון קטע התכנית

```
Stack<Integer> myStack = newStack<Integer>();
sod1(myQueue , myStack);
sod2(myQueue , myStack);
```

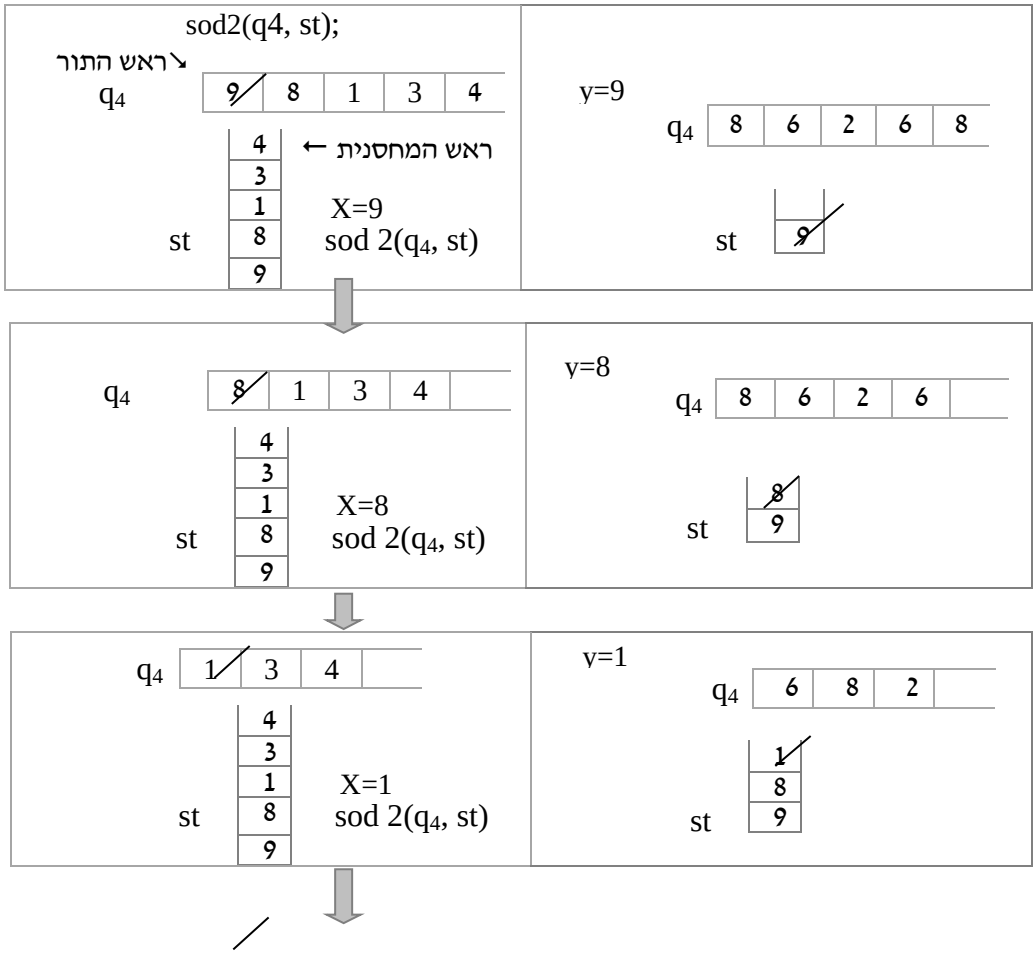
א. מעקב אחר הפעולה sod1

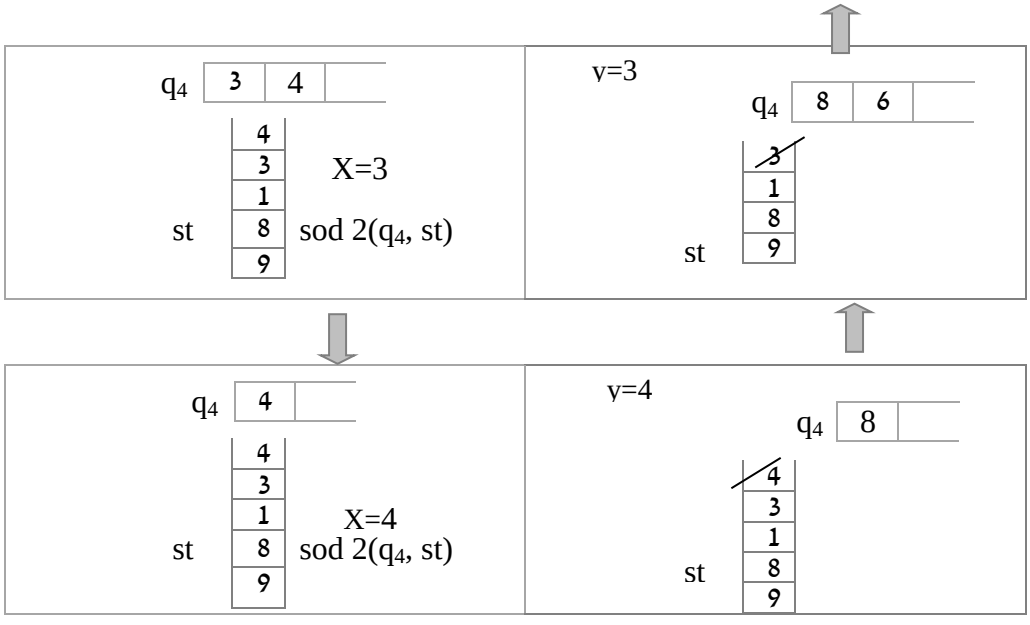
(1) התור myQueue והמחסנית myStack לאחר הזימון: sod1(myQueue , myStack) מכילים:





מעקב אחר הפעולה sod2





(2) התור myQueue והמחסנית myStack בסוף קטע התכנית מכילים

(i) myQueue מההתחלה לסוף, מימין לשמאל: 4,3,1,8,9

myStack מבפנים החוצה, מימין לשמאל: 4,3,1,8,9

(ii) myQueue מההתחלה לסוף, מימין לשמאל: 18,16,2,6,8

myStack: ריק

ב. בעבור תור כלשהו לא ריק myQueue מטיפוס Queue<Integer> ומחסנית myStack ריקה מטיפוס Stack<Integer> הפעולה sod1 מבצעת:

מחזירה במחסנית את אברי התור בסדר זהה כך שהאיבר שהיה בראש התור הוא בראש המחסנית, והופכת את סדר האיברים בתור.

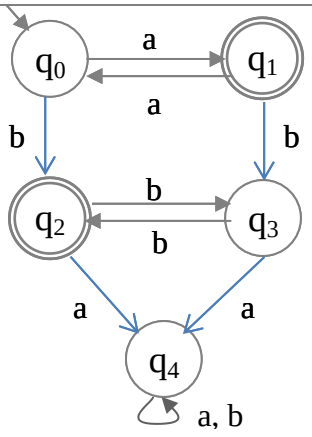
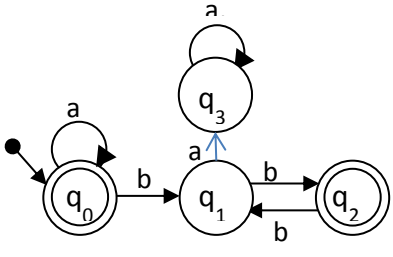
ג. בעבור תור כלשהו לא ריק myQueue מטיפוס Queue<Integer> ומחסנית myStack ריקה מטיפוס Stack<Integer> קטע התכנית מבצע:

הכפלה של ערכי כל האיברים בתור תוך שמירה על סדר האיברים.

פתרון שאלה 13

נושא מרכזי: אוטומט סופי, רגולריות של שפה
סוג השאלה: ניתוח, בניית מודל

א. קביעת רגולריות של שפות ובניית מודל לשפות הרגולריות

השפה	קביעת רגולריות + הסבר	אוטומט דטרמניסטי סופי
$L_1 = \{a^n b^{n-1} \mid n \geq 1\}$	השפה L_1 אינה רגולרית. יש לזכור כמה a נקלטו כדי לבדוק את מספר ה-b.מספר ה-a-ים תלוי במספר ה-b-ים כתלות אינסופית של מניה.	
$L_2 = \{ a^n b^k \mid$ $n\geq 0, k\geq 0$, שארית החלוקה של n ב-2 שונה משארית החלוקה של k ב-2. $\}$	השפה L_2 רגולרית	
$L_3 = \{a^k b^{2m} \mid k \geq 0,$ $m\geq 0\}$	השפה L_3 רגולרית	

הסבר הפתרון לאוטומט עבור השפה L_2:

יש לשים לב שהשפה היא שפה של רצפים, בתחילתה ניתן לקלוט רק a ואז לעבור לקליטת b.

המצב q_0 – רצף a-ים באורך זוגי (0 – מספר זוגי), עוד אין b – לכן רצף זוגי לכאורה, לכן אינו מצב מקבל.

המצב q_1 – רצף a-ים באורך אי-זוגי –אין תווי b, כלומר מספר b-ים זוגי – מצב מקבל

המצב q_2 – רצף a-ים היה באורך זוגי, רצף b-ים באורך אי זוגי – מצב מקבל

המצב q_3 – רצף a-ים באורך אי-זוגי, רצף b-ים באורך אי-זוגי

הסבר הפתרון לאוטומט עבור השפה L_3:

השפה היא שפה של רצפים. המשמעות של רצף b באורך 2m – הוא שהרצף חייב להיות זוגי, כי עבור כל מספר שלם m, המספר 2m הוא זוגי.

ב. השפה L_5 היא:

נתונות השפות L_4 -L_5 מעל האו"ב {a, b}:

$L_4 = \{a^n b^{n-1} \mid n, n \geq 1 \text{ הוא אי-זוגי} \}$

$L_5 = L_1 \cdot R(L_4)$

$L_5 = L_1 \cdot R(L_4) = \{a^n b^{n-1} b^{m-1} a^m \mid n \geq 1, \text{ אי-זוגי } m, m \geq 1\} = \{a^n b^{n+m-2} a^m \mid n \geq 1, \text{ אי-זוגי } m, m \geq 1\}$

הסבר הפתרון לשפה L_5:

השפה היא שרשור של שתי שפות שאינן תלויות זו בזו, לכן הדבר החשוב הוא לא לדרוש את אותו משתנה n בשתי השפות, אלא להשתמש במשתנה <u>אחד</u> המתאר את התלות בשפה הראשונה, ובמשתנה <u>אחר</u> לתיאור התלות בשפה השנייה.

פתרון שאלה 14

נושא מרכזי: שפות רגולריות- אוטומט סופי
סוג השאלה: בניית מודל

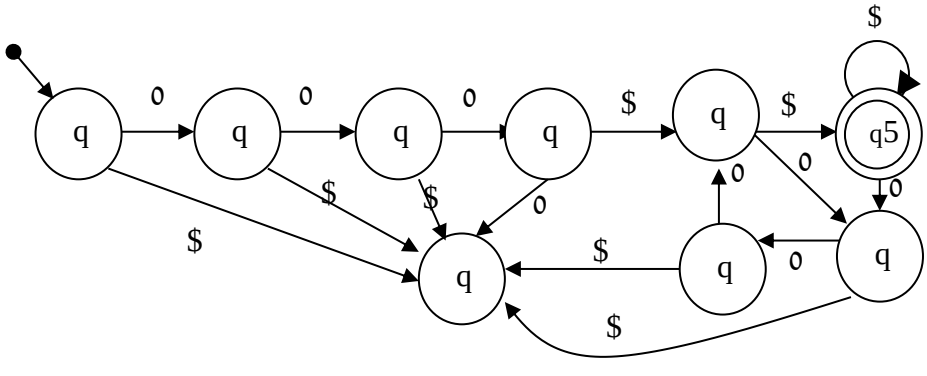
נתוני השאלה: נתונה השפה l מעל האו"ב {0, $}:

$$L = \left\{ 0^3\$0^{i_1}\$0^{i_2}\$...0^{i_k}\$ \;\middle|\; \begin{array}{l} k \geq 1 \text{ ,} \\ \text{לכל } m \text{ בין 1 ל- } k: \; i_m \geq 0 \\ \text{ו- } i_m \text{ מתחלק ב- 3 ללא שארית} \end{array} \right\}$$

א. המילה הקצרה ביותר בשפה L היא:

המילה תתקבל עבור k=1, זאת אומרת המילה היא מהצורה $0^3\$0^{i_1}\$$, ולכן נבחר $i_1=0$, ונקבל את המילה 000$$, וזאת המילה הכי קצרה.

ב. אוטומט סופי דטרמניסטי שמקבל את השפה L:



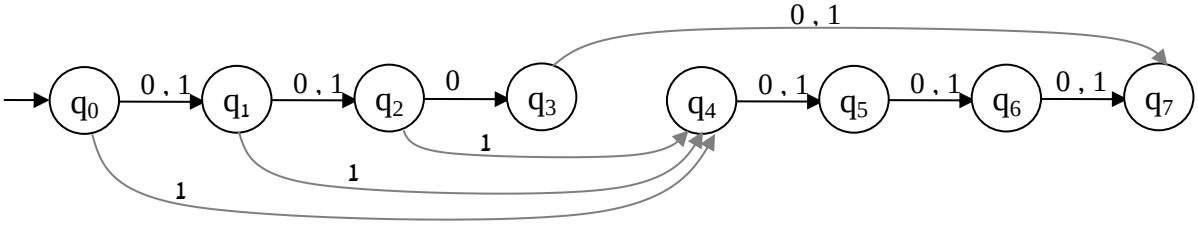
הסבר לתיאור האוטומט:

- אחרי ה- $ הראשון, יש להקפיד בכל מקרה שיתקבלו רצפים של 0 באורך 3.
- רצף יכול להיות גם באורך 0, ולכן יש לאפשר קלטה חוזרת של $.

פתרון שאלה 15

נושא מרכזי: פעולות על מילים ושפות
סוג השאלה: זיהוי מילים, זיהוי שפה רגולרית/לא רגולרית

נתוני השאלה: נתון אוטומט סופי לא דטרמניסטי הבא:



א

(1) **קביעה האם המילה מתקבלת ע"י האוטומט:**

i. המילה 001001 מתקבלת ע"י האוטומט. דוגמה למסלול: $q_0 \xrightarrow{0} q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_4 \xrightarrow{0} q_5 \xrightarrow{0} q_6 \xrightarrow{1} q_7$

ii. המילה 01010 מתקבלת ע"י האוטומט. דוגמה למסלול: $q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_4 \xrightarrow{0} q_5 \xrightarrow{1} q_6 \xrightarrow{0} q_7$

iii. המילה 0101 לא מתקבלת ע"י האוטומט.

(2) **האורך המינימלי** של מילה שמתקבלת ע"י האוטומט הוא 4, למשל המילה 1111.

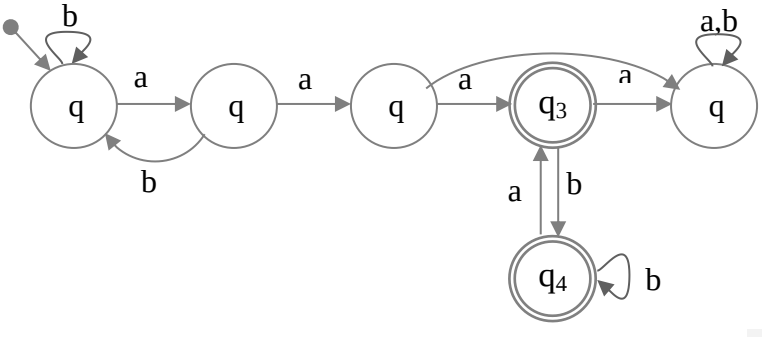
(3) **האורך המקסימלי** של מילה שמתקבלת ע"י האוטומט הוא 6, למשל המילה 001001.

(4) **השפה המוגדרת** על-ידי האוטומט היא $L = \{ \{0,1\}^n \cdot 1 \cdot \{0,1\}^3 \mid 0 \le n \le 2 \}$

ניתן להגדיר את השפה גם באופן מילולי: השפה מקבלת מילים מעל האי"ב {0,1} בהן התו הרביעי מן הסוף הוא 1, ולפניו יכולים להיות עד שני תווים.

ב

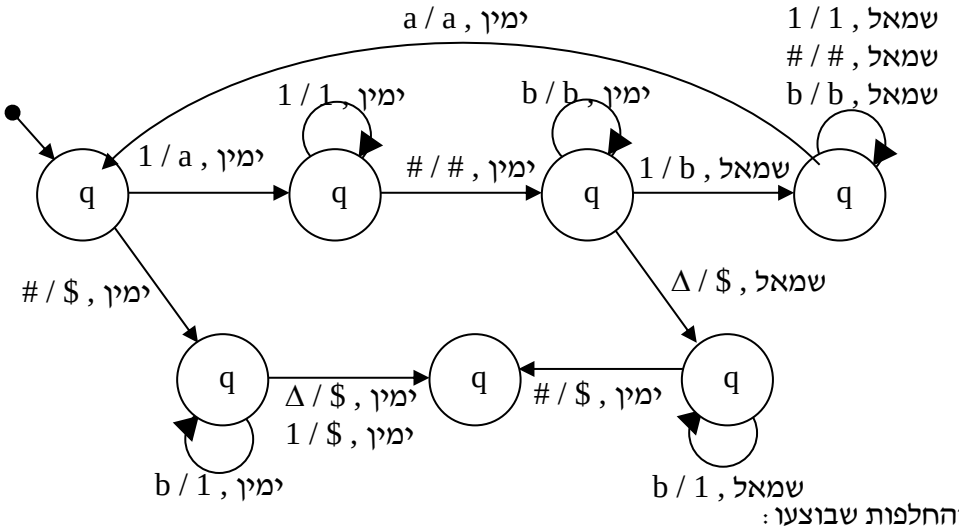
השלמת אוטומט



פתרון שאלה 16

נושא מרכזי: מכונת טיורינג
סוג השאלה: זיהוי מילים, זיהוי שפה רגולרית/לא רגולרית

א. השלמת סרטוט של מכונה



ההחלפות שבוצעו:

א. ימין , # / #

ב. ימין , $ / #

ג. שמאל, $ / Δ

ב. תהליך החישוב של המכונה בעבור הקלט: n=1, m=1.

q_0

1	#	1	Δ	Δ	⋯

q_1

a	#	1	Δ	Δ	⋯

q_2

a	#	1	Δ	Δ	⋯

q_3

a	#	b	Δ	Δ	⋯

q_3

a	#	b	Δ	Δ	⋯

q_0

a	#	b	Δ	Δ	⋯

q_4

a	$	b	Δ	Δ	⋯

q_4

a	$	1	Δ	Δ	⋯

q_5

a	$	1	$	Δ	⋯

הסבר של פעולת המכונה להעשרה: המכונה מסמנת בהתאמה, 1 –ים מן המספר הראשון ב- a, ו- 1 –ים מהמספר השני ב- b. מסמנת תו אחד מן המספר הראשון, עוברת למספר השני ומסמנת תו אחד ממנו. אם במצב q_2, אין עוד ספרה 1 לסימון, סימן שהמספר השני קטן יותר, אך בכל מקרה מספר התווים שהוחלפו בשני המספרים הוא באותו גודל – כגודל המספר הקטן. בשלב זה עובר האוטומט ל "סידור" הפלט. כלומר לרשום שני $ משני הצדדים ולהפוך חזרה את תווי b לתווי 1. אם במצב q_0 אין עוד ספרה 1 לסימון, סימן שהמספר הראשון קטן יותר, אך עדיין מספר תווי b זהה למספר תווי a, שהוא כגודל המספר הקטן. גם כאן יש מעבר ל "סידור" הפלט.