

## מדעי המחשב ב'

### פתרון בחינת הקלרות

#### פרק א - עיצוב תכנה

שאלה 1 - תור-לפי-שכיחות (Q)

פתרון בשפת C#:

```
//--- פעולה המקבלת תור של מספרים ומחזירה תור מצומצם ---  
//--- שבו עבור כל איבר הוא יופיע פעם אחת ואח"כ מספר מופעיו ---  
  
//--- יעילות הפעולה  $f(n) = n + 2 + n \cdot (n + 3) \Rightarrow n^2 + 4n + 2$  כי  $O(n^2)$  ---  
//--- מייצג את מספר האיברים בתור n כאשר ---  
public static Queue<int> QueCompress(Queue<int> que)  
{  
    Queue<int> qCopy = Clone(que); // O(n)  
    Queue<int> q1 = new Queue<int>(); // O(1)  
  
    while (!qCopy.IsEmpty()) // n *  
    {  
        int num = qCopy.Head(); // O(1)  
        int count = CountX(qCopy, num); // O(n)  
  
        q1.Insert(num); // O(1)  
        q1.Insert(count); // O(1)  
    }  
    return q1; // O(1)  
}  
  
//--- פעולה המקבלת תור של תווים ותו נוסף ---  
//--- ומחזירה כמה פעמים הופיע התו בתור ---  
//--- בסיום התו לא יהיה יותר בתור ---  
public static int CountX(Queue<int> que, int num)  
{  
    int count = 0, x;  
    Queue<int> qTemp = new Queue<int>();  
  
    //--- נספור אותו num-אם האיבר שווה ל ---  
    //--- אחרת - נשמור אותו בתור העזר ---  
    while (!que.IsEmpty())  
    {  
        x = que.Remove();  
        if (x == num)  
            count++;  
        else  
            qTemp.Insert(x);  
    }  
  
    //--- החזרת האיברים לתור ---  
    while (!qTemp.IsEmpty())  
        que.Insert(qTemp.Remove());  
  
    return count;  
}
```

```
//--- פעולה המשכפלת תור של מספרים תווים ---  
public static Queue<int> Clone(Queue<int> que)  
{  
    Queue<int> qTemp = new Queue<int>();  
    Queue<int> qCopy = new Queue<int>();  
  
    while (!que.IsEmpty())  
    {  
        qTemp.Insert(que.Head());  
        qCopy.Insert(que.Remove());  
    }  
  
    while (!qTemp.IsEmpty())  
        que.Insert(qTemp.Remove());  
  
    return qCopy;  
}
```

**סיבוכיות הפעולה תור-לפי-שכיחות:**  $O(n^2)$

יש סה"כ  $n$  איברים בתור. עבור כל איבר יש לספור את מספר מופעיו -  $n$  צעדים.  
כל פעולות התור בסיבוכיות  $O(1)$ , ולכן הסיבוכיות ריבועית.  
(במקרה הגרוע כל איבר מאיברי התור מופיע בדיוק פעם אחת).

## פתרון בשפת java

נכתב ע"י: יוסף מלכה, אורט סינגלובסקי, ת"א

```
public class B_2006_1
{
    public static Queue<Integer> queFrequency (Queue<Integer> q)
    {
        Queue<Integer> qFrequency = new Queue<Integer> ();
        Queue<Integer> tmpQ = copyQueue ( q );
        while (! tmpQ.isEmpty())
        {
            int x = tmpQ.remove();
            int countX = 1 + countAndRemove (tmpQ, x);
            qFrequency.insert (x);
            qFrequency.insert(countx);
        }
        return qFrequency;
    }

    public static int countAndRemove (Queue<Integer> q , int x)
    {
        int k, count=0;
        Queue<Integer> tmp_q = new Queue<Integer> ();
        while (! q.isEmpty())
        {
            k = q.remove();
            if (k == x)
                count++;
            else
                tmpQ.insert(k);
        }
        // return q as before just not with x inside q.
        while (! tmpQ.isEmpty())
            q.insert (tmpQ.remove());
        return (count);
    }
}
```

```

public static Queue<Integer> copyQueue (Queue<Integer> q )
{
    Queue<Integer> tmpQ1 = new Queue<Integer>();
    Queue<Integer> tmpQ2 = new Queue<Integer>();

    while (! q.isEmpty())
    {
        int k = q.remove();
        tmpQ1.insert (k);
        tmpQ2.insert (k);
    }

    // return q as it was
    while (! tmpQ2.isEmpty())
        q.insert (tmpQ2.remove());

    return tmpQ1;
}

```

הערה :

דוגמא להרצה כאשר ממלאים 10 ערכים לתוך התור.

לפני	
3	
0	
18	
18	
11	
18	
11	
11	
13	
12	
	אחרי
	3 1
	0 1
	18 3
	11 3
	13 1
	12 1

עלף 2:

	מחסנית-ריקה?	a	זימון / ביצוע	ערך מוחזר	א. סוד 2 (S1)
7	F	7	סוד 2 + (a) 7	1 + 7 = 8	
4	F	4	סוד 2 + (a) 4	-3 + 4 = 1	ערך מוחזר: 8
-5	F	-5	סוד 2 + (a) (-5)	-5 + 2 = -3	
2	T	2	החזר a	2	

S1

ב. האלגוריתם סוד 2 מחזיר את סכום איברי המחסנית.

	n = 1	a	d	n	זימון	ג. סוד 1 (S2, 2, 6, 4)
20	F	2	6	4	סוד 1 (S, 2, 6, 4)	
14	F	8	6	3	סוד 1 (S, 8, 6, 3)	
8	F	14	6	2	סוד 1 (S, 14, 6, 2)	
2	T	20	6	1	סוד 1 (S, 20, 6, 1)	

S

ערך מוחזר: 44

ד. סוד 1 מחזיר את סכום n האיברים הראשונים בסדרה החשבונית שהאיבר הראשון שלה הוא a והפרשה הוא d. הערה: אם המחסנית מועברת לסוד 1 (בלבד!) לפי כתובת, יהיו במחסנית איברי הסדרה החשבונית.

שאלה 3: - ריבוע (s, color, x, y) - פתרון C# תוך שימוש בצב:

```
//--- הגדרת מערך הצבעים ---
static Color[] colors = { Color.Blue, Color.Red, Color.Green, Color.Orange,
                          Color.Coral, Color.Cyan, Color.Azure, Color.DarkSeaGreen };
static int k = colors.Length;

//--- פעולה רקורסיבית המקבלת כפרמטר: ---
//--- size - אורך צלע ריבוע ---
//--- code - קוד של צבע ריבוע ---
//--- x, y - מיקום התחלתי של הצב ---
//--- ומציירת סידרה של ריבועים זה בתוך זה ---
public static void Squares (Turtle t, int size, int colorCode, int x, int y)
{
    if (size > 0)
    {
        t.SetTailColor(colors[colorCode % k]);    // החלפת צבע
        DrawSquare(t, size, x, y);

        Squares(t, size - 20, colorCode + 1, x+10, y+10);
    }
}

//--- הפעולה: צייר-ריבוע הנתונה ---
public static void DrawSquare(Turtle t, int size, int x, int y)
{
    t.MoveTo(x, y);
    t.TurnRight(t.GetHeading());

    t.TailDown();
    for (int i = 0; i < 4; i++)
    {
        t.MoveForward(size);
        t.TurnRight(90);
    }
    t.TailUp();
}

static void Main(string[] args)
{
    Turtle t1 = new Turtle();
    t1.SetVisible(false);
    t1.TailUp();

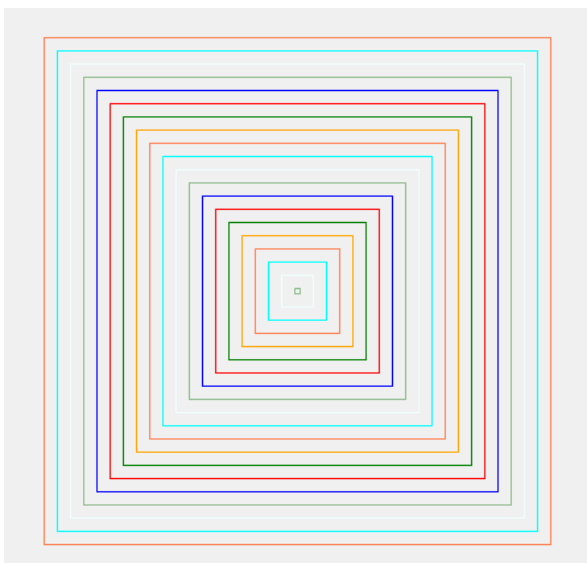
    Random rnd = new Random(0);

    //--- קביעת מיקום התחלתי לצב ---
    int x = rnd.Next(-400, 0);
    int y = rnd.Next(-300, 0);
    t1.MoveTo(x, y);

    //--- הבא את הצב לפנות בכיוון מעלה ---
    t1.TurnRight(t1.GetHeading());
    Console.WriteLine(t1.GetHeading());

    int size = rnd.Next(500);
    int colorCode = rnd.Next(k);

    Squares(t1, size, colorCode, x, y);
}
```



## אלף 4 - סידור-רשימות (L1, L2)

פתרון בשפת C# :

```

//--- פעולה המקבלת שתי רשימות של מספרים שלמים lst1 ו- lst2 ---
//--- עבור כל איבר ב lst2: אם הוא זוגי, יימחק האיבר במקום זה ב- lst1 ---
//--- ואם אי-זוגי, יתווסף האיבר במקום זה לרשימה חדשה lst3 ---
//--- אם אין k איברים ברשימה, לא יעשה כלום ---
//--- הפעולה מחזירה את הרשימה החדשה lst3 ---
public static Node<int> ArrangeLst (Node<int>lst1, Node<int> lst2)
{
    Node<int> lst3 = null;
    Node<int> pos1 = lst1, p;
    int k;

    while (pos1 != null)
    {
        k = pos1.GetValue();
        p = GetPosition(lst2, k);
        if (p != null)
        {
            if (k % 2 == 0) // איבר זוגי - יש לבצע מחיקה
                lst2 = Delete(lst2, p);
            else
                lst3 = new Node<int>(p.GetValue(), lst3);
        }
        pos1 = pos1.GetNext();
    }

    return lst3;
}

//--- פעולה המקבלת רשימה ומספר שלם וחיובי k ---
//--- ומחזירה את החוליה ה- k ברשימה ---
//--- אם אין k חוליות ברשימה יוחזר null ---
public static Node<int> GetPosition(Node<int> lst, int k)
{
    Node<int> pos = lst;
    while (pos != null && k > 1)
    {
        pos = pos.GetNext();
        k--;
    }
    return pos;
}

```

```
//--- פעולה המקבלת מקום ברשימה ומוחקת את החוליה במקום זה ---  
//--- הנחה: המקום אינו null ---  
public static Node<int> Delete (Node<int> lst, Node<int> pos)  
{  
    Node<int> prev = GetPrev(lst, pos);  
  
    if (prev == null)  
        return lst.getNext();  
    prev.SetNext(pos.GetNext());  
  
    return lst;  
}  
  
//--- פעולה המקבלת רשימה ומיקום ---  
//--- ומחזירה הפניה לחוליה הקודמת למיקום זה ---  
//--- אם המיקום הוא החוליה הראשונה ברשימה, יוחזר ערך null ---  
public static Node<int>GetPrev(Node<int> lst, Node<int> pos)  
{  
    if (pos == lst)  
        return null;  
    Node<int> prev = lst;  
    while (prev.GetNext() != pos)  
        prev = prev.GetNext();  
    return prev;  
}
```



פתרון בשפת java

נכתב ע"י: יוסף מלכה, אורט סינגלובסקי, ת"א

```
public class B_2006_4
{
    public static List<Integer> getL3 ( Node<Integer> L1 , Node<Integer> L2)
    {
        int p, k = 0;
        Node<Integer> L3 = null;
        Node<Integer> n1 = L1;
        Node<Integer> n2 = null;

        while (n1 != null)
        {
            k = n1.getValue();
            n2 = L2;
            p = 1;

            while (n2 != null && p < k)
            {
                n2 = n2.getNext();
                p++;
            }

            if (p == k && n2 != null)
            {
                if (k%2 == 0)
                    L2 = remove(L2, n2);
                else
                    L3 = new ( Node<Integer> (n2.getValue(), L3));
            }

            n1 = n1.getNext();
        }

        return L3;
    }
}
```

```
//--- פעולה המקבלת מקום ברשימה ומוחקת את החוליה במקום זה ---
//--- הנחה: המקום אינו null ---

public static Node<Integer> delete (Node<Integer> L2, Node<Integer> n2)
{
    Node<Integer> prev = getPrev(L2, n2);

    if (prev == null)
        return L2.getNext();
    prev.setNext(n2.getNext());

    return L2;
}

//--- פעולה המקבלת רשימה ומיקום ---
//--- ומחזירה הפניה לחוליה הקודמת למיקום זה ---
//--- אם המיקום הוא החוליה הראשונה ברשימה, יוחזר ערך null ---
public static Node<Integer> getPrev(Node<Integer> L2, Node<Integer> n2)
{
    if (n2 == L2) // האם למחוק חוליה ראשונה?
        return null;
    Node<Integer> prev = L2;
    while (prev.getNext() != n2)
        prev = prev.getNext();
    return prev;
}
```

### פרק ב'

#### מערכות מחשב ואסמבלר

הפתרון לפרק זה נכתב ע"י טובי סטפ

#### תרגיל 5:

- א. לא נכון. כדי להוכיח נהפוך את המספר ל-Hexadecimal:  $2D4D_{16} = 10110101001101_2$ .  $643_{16}$  אינו גדול מ- $2D4D_{16}$ . התקבל מספר בינארי באורך 14 ולכן ה-msb הוא 0 (יש שני 0 מקדמים) ולכן אם נתרגם נקבל שהמספר השני גדול יותר.
- ב. נכון. ניתן להגדיר אותם כחופפים. אם נסתכל – יש לנו תוכניות שאנחנו מגדירים את ה-CS וה-DS באותו מקום.
- ג. נכון. אם כתובת המקטע תהיה שונה, יתכן ותוצאת החישוב תהיה אותה כתובת אבסולוטית. (לדוגמא: כתובת המקטע היא 4500 והכתובת היחסית 0 – הכתובת האבסולוטית היא 45000H. ואם המקטע יתחיל ב-4400H והכתובת היחסית היא 1000H אזי הכתובת המוחלטת שלו תהיה:  $44000 + 1000 = 45000H$ ).
- ד. נכון. הכוונה ל-IR - instruction register, הוא אינו משתנה במהלך ביצוע הפקודה.
- ה. נכון. בהנחה שגודל האוגר הכתובות הוא כגודל מילה, ובהנחה שהקריאה היא קריאה קרובה (Near) גודל המחסנית יספיק. – תחילה תישמר במחסנית הכתובת IP המקורית (מילה – 2 בתים) ואחריה נדחוף למחסנית את הכתובת שבפרוצדורה, ולכן זה יתפוס עוד 2 בתים – סה"כ 4 בתים.
- ו. לא נכון. זה תלוי כמה סיבובים יתבצעו (ROL AL, CL). אבל גם אם יהיה רק סיבוב אחד, זה לא בהכרח יעבוד. לדוגמא: 10000001 (129) ונבצע ROL al, 1 יתקבל 00000011 הערך הוא 3, ולכן הטענה לא נכונה.

#### תרגיל 6:

- א. שעת היציאה: 10:15
- ב. אם היינו מגדירים את השעה כבית אזי כל nibble היה מכיל שעה ודקות. המספר המקסימלי שאפשר לשמור בכל nibble הוא 1111 דהיינו 15 ואז השעה והדקות המקסימליות היו 15 וזה אינו מתאים לדרישות ולכן זה אינו מתאים.
- ג. השגיאה: בדקות: זמן היציאה בדקות הוא 15 ומשך הנסיעה בדקות הוא: 50 (32 בהקסה זה 50 בעשרוני) אם נחבר אותם נקבל 65 שזה מעבר למה שניתן לשמור בדקות כדי לתקן נבצע את הדבר הבא:

```
ADD    BX, AX
CMP    BL, 03CH
JL     CONT
INC    BH
SUB    BL, 03CH
```

```
CONT:  MOV    ARRIVALTIME, BX
```

תרגיל 7:

instr	[BP + 4]	[BP + 2]	DX	CX	AX
mov AX,					0FFF3H
PUSH AX					
PUSH Line					
CALL exa					
mov BP, SP	0FFF3H	0EACFH			
mov AX,0					0
MOV cx				15 (0FH)	
MOV DX,			0EACFH		
OR DX,			0FFFFH		
CMP DX,...					
JZ FOUND					
ROL WO	0FFE7H				
Loop Again				14 (0EH)	
MOV DX,			0EACFH		
OR DX,			0FFE7H		
CMP DX,					
JZ					
ROL	0FFCFH				
Loop Again				13 (0DH)	
mov DX,			0EACFH		
OR DX,			0FFCFH		
CMP DX,					
JZ Found					
Mov AX, 1					1
Ret 4					
NOP					

הערך שיהיה ב – ANSWER יהיה 1

הפרוצדורה EXAMPLE בודקת האם בשורה LINE יש מקומות פנויים כפי שהוצג באוגר AX

עכ"ל:

```

DATA SEGMENT
    FIRST DB ?
    SECOND DB ?
    ANSWER DW ?
DATA ENDS

CODE SEGMENT
    ASSUME CS: CODE, DS: DATA

START:  MOVE AX, DATA
        MOV DS, AX
        MOV AL, SECOND
        MOVE BL, FIRST
        XOR DX, DX
        XOR BH, BH

NEXT:   TEST AL, 1
        JZ CONT
        ADD DX, BX
        DEC AL
        JZ SOF

CONT:   SHL BX, 1
        SHR AL, 1
        JMP NEXT

SOF:    MOV ANSWER, DX
        MOV AX, 04C00H
        INT 021H

CODE ENDS

END START
    
```

פרק ב'  
תורת המחשב

עלף 9:

10.00	-4.00	2.00		16.00		$g = 0.50$
-5.00	4.00	-3.00		14.00	$R2 \leftarrow R2 + 0.5 * R1$	$i = 1 \quad j = 1$
-5.00	-8.00	11.00		88.00		$c = 1.4$
-----						
10.00	-4.00	2.00		16.00		$g = 0.50$
0.00	2.00	-2.00		22.00		$i = 1 \quad j = 3$
-5.00	-8.00	11.00		88.00	$R3 \leftarrow R3 + 0.5 * R1$	$c = 1.4$
-----						
10.00	-4.00	2.00		16.00	$R1 \leftarrow R1 + 2 * R2$	$g = 2.00$
0.00	2.00	-2.00		22.00		$i = 2 \quad j = 1$
0.00	-10.00	12.00		96.00		$c = 2.4$
-----						
10.00	0.00	-2.00		60.00		$g = 5.00$
0.00	2.00	-2.00		22.00		$i = 2 \quad j = 3$
0.00	-10.00	12.00		96.00	$R3 \leftarrow R3 + 5 * R2$	$c = 2.4$
-----						
10.00	0.00	-2.00		60.00	$R1 \leftarrow R1 + R3$	$g = 1.00$
0.00	2.00	-2.00		22.00		$i = 3 \quad j = 1$
0.00	0.00	2.00		206.00		$c = 3.4$
-----						
10.00	0.00	0.00		266.00		$g = 1.00$
0.00	2.00	-2.00		22.00	$R2 \leftarrow R2 + R3$	$i = 3 \quad j = 2$
0.00	0.00	2.00		206.00		$c = 3.4$
-----						
10.00	0.00	0.00		266.00		
0.00	2.00	0.00		228.00		
0.00	0.00	2.00		206.00		

solutions :  $10 * x_1 = 266.00$   $2 * x_2 = 228.00$   $2 * x_3 = 206.00$

**סוד**  $(a, n)$  פותר מערכת משוואות ליניאריות בעלת  $n$  מעלמים בסגנון גאוס (סגנון כמעט דומה לשיטה).  
הפתרון מוצג בצורת:  $m * x_i = b$

הפתרון לשאלה זו נכתב ע"י: שירלי רוזנברג כהן

אלף 10:

א. (i) הקבוצות: (א) 1, 5, 4 (ב) 2, 3, 6 (ii) הוספת קשתות: רק בין צמתים שאינם באותה קבוצה:



ב. כאשר צומת מחובר ל-2 צמתים אחרים, שני צמתים אלה אינם בקבוצתו. לכן ייווצר מעגל שבו כל צומת חבר לסירוגין בקבוצה אחרת. מספר הצמתים זהה בשתי הקבוצות (ולכן זוגי) כמו גרף Ga.

ג. עץ הוא "עץ פיצול", השורש הוא בקבוצה אחת והבנים בקבוצה אחרת (משום שביניהם אין קשר). בעץ ברמה  $1 <$  כל רמה היא קבוצה.

ד. כאשר דרגת צומת היא  $n-1$  הרי שהוא מחובר לכל הצמתים האחרים. לכן, הפיצול בגרף כזה מחייב ששאר הצמתים יהיו מחוברים רק אליו, ודרגתם 1. (כמו בגרף הראשון בשאלה עצמה).

שאלה 11:

		א. עץ גזירה למילה: aabbcc	ב. עץ גזירה למילה: caac
G1:	$S \rightarrow CAC$ $C \rightarrow cC \mid \varepsilon$ $A \rightarrow aAb \mid \varepsilon$		לא ניתן
G2:	$S \rightarrow aSa \mid bSb \mid cSc \mid \varepsilon$	לא ניתן	
G3:	$S \rightarrow CABC$ $A \rightarrow aA \mid \varepsilon$ $B \rightarrow bB \mid \varepsilon$ $C \rightarrow cC \mid \varepsilon$		
G4:	$S \rightarrow AB$ $A \rightarrow aA \mid \varepsilon$ $B \rightarrow bBc \mid \varepsilon$		לא ניתן

ג. השפות הנוצרות על ידי הדקדוקים:

א"ב השפות:  $\Sigma = \{\varepsilon, a, b, c\}$

G1:  $c^n a^i b^j c^m \mid i, n, m \geq 0$

G2: פאלינדרום באורך זוגי מעל  $\{a, b, c\}$  הכולל את המילה הריקה

G3:  $c^n a^i b^j c^m \mid i, j, n, m \geq 0$

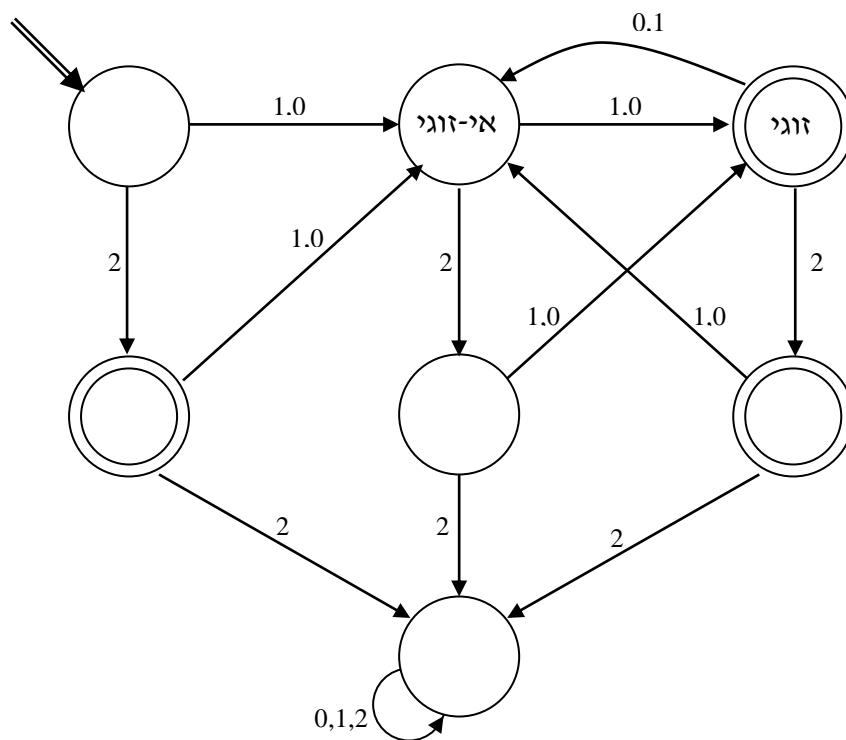
G4:  $a^i b^m c^m \mid i, m \geq 0$



שאלה 12:

$\Sigma = \{0, 1, 2\}$  א"ב:

- תנאים: יש לפחות אות אחת בשפה.
- מספר ה-0 + מספר ה-1 = זוגי.  $(\#0 + \#1) \bmod 2 = 0$
- אין רצפים של '2' בשפה.



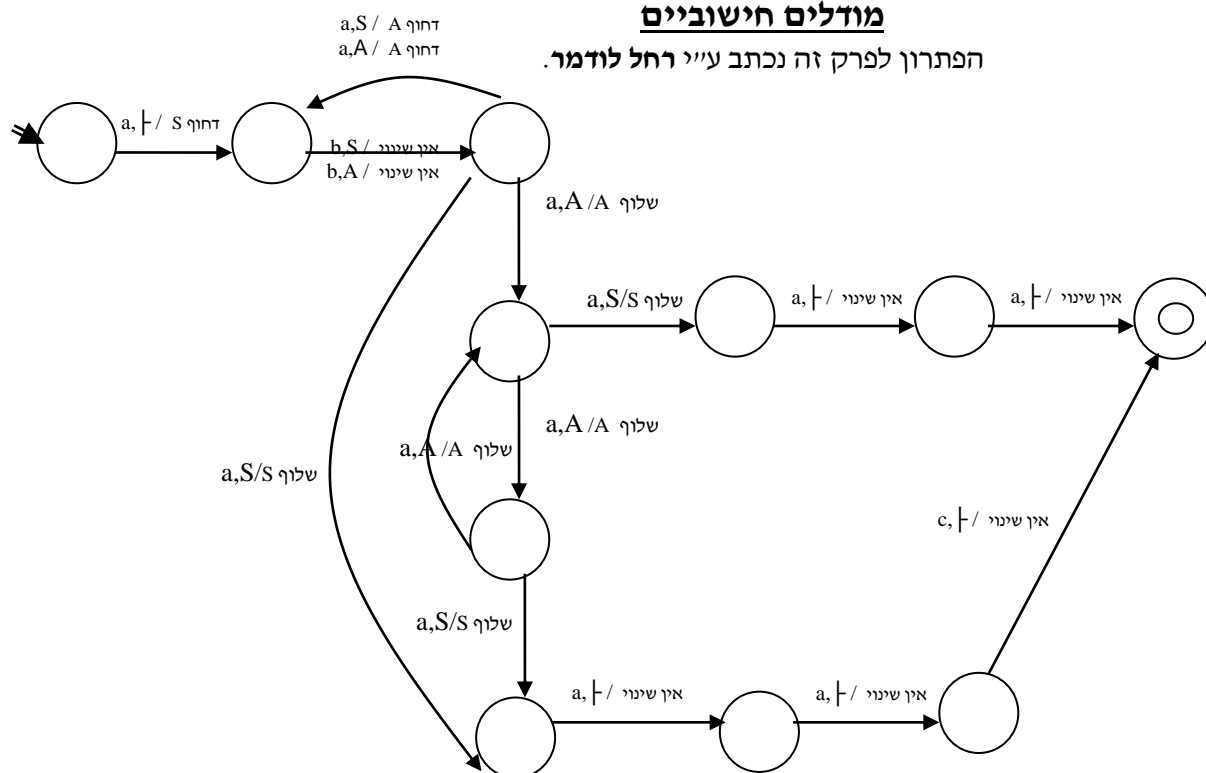
הערה: ניתן לבנות אוטומט בפחות מצבים כפי שמוצג בפתרון של רחל לודמור  
במודלים חישוביים - שאלה 16.

פרק ב'

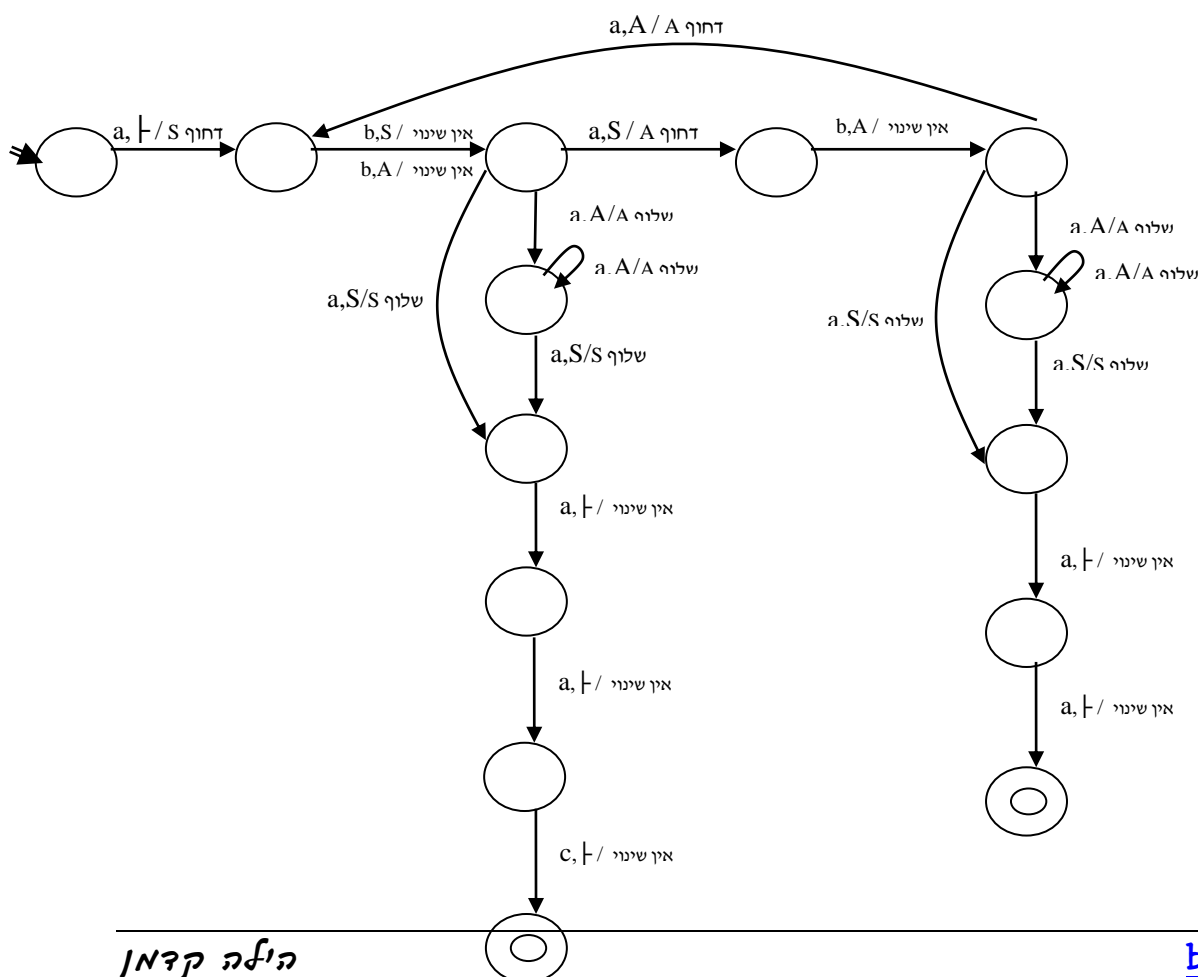
מודלים חישוביים

הפתרון לפרק זה נכתב ע"י רחל לודמר.

תרגיל 13:



פתרון אחר: (שמירת הזוגיות על הרצפים של (ab))



## תרגיל 14:

א.  $w_1 = aaa$  שייכת לשפות  $L_1, L_3, L_4$ .

המילה שייכת לשפה  $L_1$  עבור  $i=3$  ו- $k=0$ , ולכן  $i+k=3$  מתחלק ב-3 ללא שארית.

המילה שייכת לשפה  $L_3$  עבור  $i=3$  ו- $k=0$ , שניהם מתחלק ב-3 ללא שארית.

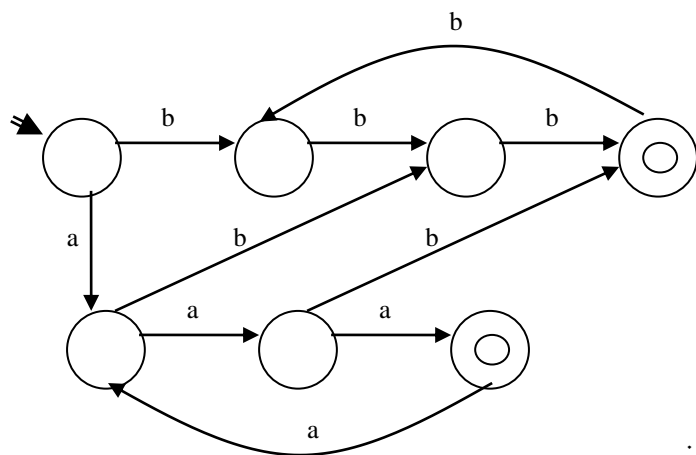
המילה שייכת לשפה  $L_4$  עבור  $i=0$  ו- $k=3$ , ולכן  $k$  מתחלק ב-3 ללא שארית.

$w_2 = ab^3a^3$  שייכת לשפה  $L_4$ , עבור  $i=1, k=3$ , מספר ה- $b$  כפולה של 3 ממספר ה- $a$ , ומספר ה- $a$  ברצף האחרון מתחלק ב-3 ללא שארית.

ב.  $L_1$  רגולרית. ניתן לבנות אס"ד שיאפיין את המצבים הבאים:

- ❖ המילה מכילה 0 אותיות  $a$ , ורצף ה- $b$  מתחלק ב-3.
- ❖ המילה מכילה רצף- $a$  המתחלק ב-3, ו 0 אותיות  $b$ .
- ❖ המילה מכילה רצף- $a$  המתחלק ב-3 עם שארית 1, ורצף אותיות  $b$  המתחלק ב-3 עם שארית 2.
- ❖ המילה מכילה רצף- $a$  המתחלק ב-3 עם שארית 2, ורצף אותיות  $b$  המתחלק ב-3 עם שארית 1.

האוטומט: (לא חובה לבנות)



$L_2$  רגולרית.

עבור  $i=0$ , השפה מכילה מילים מהצורה:  $b^{3n}$ .

עבור  $i=1$ , השפה מכילה מילים מהצורה:  $ab^{3n+1}a$ .

עבור  $i=2$ , השפה מכילה מילים מהצורה:  $aab^{3n+2}aa$ .

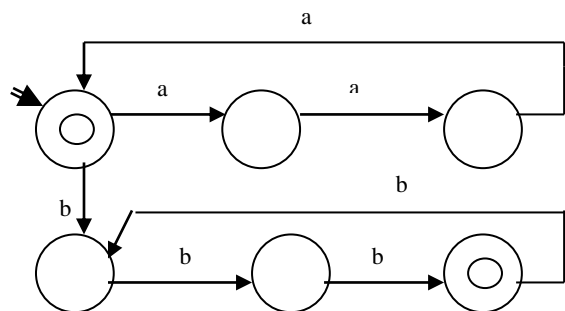
כל אחד מהם ניתן לבניה ע"י אס"ד. והשפה היא איחוד של השפות הנוצרות ע"י האוטומטים הנ"ל.

$$L_2 = \{b^{3n} \mid n \geq 0\} \cup \{ab^{3n+1}a \mid n \geq 0\} \cup \{aab^{3n+2}aa \mid n \geq 0\}, \text{ לכן,}$$

לפי תכונות הסגירות, השפה  $L_2$  רגולרית.

$L_3$  שפה רגולרית.

נבנה את האוטומט (לא מלא).



האוטומט יוצר רצף של אותיות  $a$  שאורכו מתחלק ב-3 ללא שארית, ולאחריו רצף של  $b$  עם אותה תכונה.

$L_4$  היא לא רגולרית. יש מניה בין הרצף הראשון של אותיות  $a$  לבין מספר אותיות ה- $b$  ברצף.

תרגיל 15:

$$L_1 \cap L_2 = \{c^i b^k a^k \mid i, k \geq 0\} \cap \{a^j b^t c^t \mid j, t \geq 0\} \quad (i) \quad \text{א.}$$

$$= \{c^n \mid n \geq 0\}$$

החיתוך יהיה קיים רק כאשר מספר הרצפים של  $a$ ,  $b$  יהיו 0, בשתי השפות. לכן המילה הקצרה (הלא ריקה) היא  $c$ .

$$L_2 \cap L_3 = \{wR(w) \mid w \in \{a, b, c\}^*\} \cap \{a^j b^t c^t \mid j, t \geq 0\} \quad (ii)$$

$$= \{c^n \mid n \geq 0\}$$

המילים ב-  $L_2$  הן פלינדרום באורך זוגי (כל מילה  $w$  מעל  $\{a, b, c\}$  משורשרת למילת ההיפוך שלה, ולא משנה מהו אורכה של  $w$ , המילה  $wR(w)$  תמיד תהיה באורך זוגי). בגלל המבנה של המילים בשפה  $L_3$ , החיתוך יתקיים רק כאשר לא יהיו רצפים של  $a$  ו-  $b$  ( $i=0$ ), ולכן נקבל רק אותיות  $c$  באורך זוגי, שהן ממלא פלינדרום. לכן המילה הקצרה (הלא ריקה) היא  $cc$ .

$$(L_1 \cdot L_3) \cap L_2 = (\{c^i b^k a^k \mid i, k \geq 0\} \cdot \{a^j b^t c^t \mid j, t \geq 0\}) \cap \{wR(w) \mid w \in \{a, b, c\}^*\} \quad \text{ב.}$$

$$= \{c^i b^k a^k a^k b^k c^i \mid i, k \geq 0\}$$

השפה בנויה משרשר כל המילים ב-  $L_1$  עם כל מילה ב-  $L_3$ , ושרשר זה צריך להיות פלינדרום (החיתוך עם  $L_2$ ), לכן מחייב שוויון בין הרצפים המתאימים.

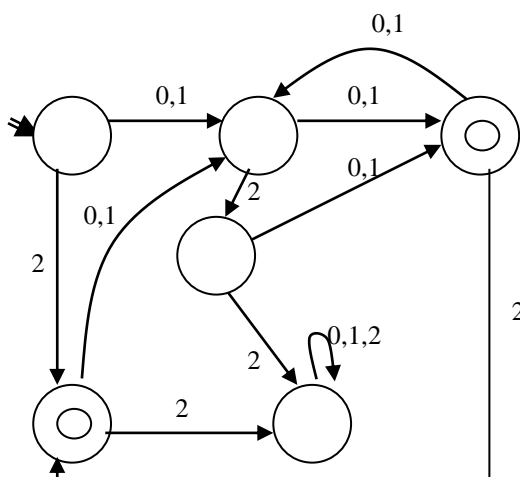
$$R(L_2) = L_2 \quad \text{הטענה נכונה.} \quad (i) \quad \text{ג.}$$

שפה הבנויה ממילים שהן פלינדרום (באורך זוגי), ההיפוך שלה היא עצמה.

$$R(L_2) = \{R(w \cdot R(w)) \mid w \in \{a, b, c\}^*\} = \{R(R(w)) \cdot R(w)\} = \{w \cdot R(w) \mid w \in \{a, b, c\}^*\} = L_2$$

$$R(L_1) = L_3 \quad \text{הטענה נכונה.} \quad (ii)$$

$$R(L_1) = \{R(c^i b^k a^k) \mid i, k \geq 0\} = \{a^k b^k c^i \mid i, k \geq 0\} = L_3$$



תרגיל 16:

- באוטומט צריכים להיות התנאים הבאים:
- ❖ אורך המילה באורך 1 לפחות, כלומר אין מילה ריקה.
  - ❖ מספר אותיות 0 + מספר אותיות 1 הוא זוגי.
  - ❖ הרצף '22' לא קיים.

פרק ב'

תכנות מונחה עצמים Java

הפתרון לפרק זה נכתב ע"י טובי סטפ

תרגיל 17:

תוצאת ההדפסה:

⇒ 11 35 47 22 8 17 53 40 21 13 39 62

פתרון של ממי גוטביר לשאלה 17:

12	11	10	9	8	צעד 7	6	צעד 5	4	צעד 3	2	צעד 1	התכונה:
					3		2		1		0	container: count
					address of SingleOne object, with num1=11 num2=35		address of SingleOne object, with num1=11 num2=35		address of SingleOne object, with num1=11 num2=35		null	container: arr[0]
					address of SingleOne object, with num1=47 num2=22		address of SingleOne object, with num1=47 num2=22		null		null	container: arr[1]
					address of SingleOne object, with num1=8 num2=17		null		null		null	container: arr[2]
					null		null		null		null	container: arr[3]
					null		null		null		null	container: arr[4]
											0	container: num1
						8		47		11		s1: num1
						17		22		35		s2: num2
												SubContainer: count
												subContainer: arr[0]
												subContainer: arr[1]
												subContainer: arr[2]
												subContainer: arr[3]
												subContainer: arr[4]
												subContainer: num1

המשך טבלת המעקב:

15 צעד	14 צעד	13	12 צעד	11	10 צעד	9	8	7	6	5	4	3	2	1	התכונה:
4															container: count
															container: arr[0]
															container: arr[1]
															container: arr[2]
address of MultiOne object: subContainer															container: arr[3]
															container: arr[4]
														0	container: num1
		39		21		53			8		47		11		s1: num1
		62		13		40			17		22		35		s2: num2
	3		2		1		0								SubContainer: count
	address of SingleOne object, with num1=53 num2=40		address of SingleOne object, with num1=53 num2=40		address of SingleOne object, with num1=53 num2=40		null								subContainer: arr[0]
	address of SingleOne object, with num1=21 num2=13		address of SingleOne object, with num1=21 num2=13		null		null								subContainer: arr[1]
	address of SingleOne object, with num1=39 num2=62		null		null		null								subContainer: arr[2]
	null		null		null		null								subContainer: arr[3]
	null		null		null		null								subContainer: arr[4]
							0								subContainer: num1

## תרגיל 18:

א. פעולות למימוש ב-AAA

```
public boolean opA(Object stam)
public void opB (int num)
```

פעולות למימוש ב-BBB

```
public boolean opA(Object stam)
public void opB (int num)
public int opC ( )
```

פעולות למימוש ב-CCC

```
public boolean opA(Object stam)
public void opB (int num)
public int opC ( )
public int opD ( )
```

- ב. (i) אינו חוקי. מנסים לייצור אובייקט מסוג ממשק, וזה אסור.
- (ii) חוקי. מזמנים את השיטה הבונה ברירת מחדל של BBB ויוצרים אובייקט BBB.
- (iii) חוקי. יוצרים אובייקט AAA ושומרים בתוך משתנה מסוג AAA. אח"כ מעבירים הפניה לאובייקט מסוג ממשק (חוקי כי AAA מממש את הממשק של IFirst).
- (iv) חוקי. מאחר ו-CCC מממש גם את ISecond ו-IThird ו-ISecond יורש את IFirst אז יש כאן upcasting וזה חוקי.

```
ג. public static void main (String []arg){
    BBB b = new BBB( );
    b.opB( 3);
    Object c = new CCC ( );
    AAA a = (AAA) c;
}
```

(i) }  
(ii) }

אומנם נעשתה פה המרה אבל כל ניסיון שלנו להפעיל פעולות יגרמו לשגיאת זמן אמת

תרגיל 19:

```
class Drawing {  
    private Line [] lines;  
    private Point [] points;  
    private int countPoints=0;  
    private int countLines = 0;  
  
    public Drawing (int np, int nl){  
        lines = new Line[nl];  
        points = new Point [np];  
    }  
  
    public Point getPoint (int num){ return points[n]; }  
  
    public void addPoint (Point p){ points[countPoints++] = p; }  
  
    public int findPoints (Point p){  
        int sum = 0;  
        for (int i=0; i < countLines; i++){  
            Point q = lines[i].getPoint1( );  
            if (p.getX( ) == q.getX( ) && p.getY( ) == q.getY( ))  
                sum++;  
            q = lines[i].getPoint2( );  
            if (p.getX( ) == q.getX( ) && p.getY( ) == q.getY( ))  
                sum++;  
        }  
        return sum;  
    }  
  
    public boolean noLineWithPoint (Point p) { return findPoint (p) == 0; }  
}
```

:20 fke



פרק ב'

תכנות מונחה עצמים C#

מי האורה שיפח פתרון ?

תרגיל 21:

ראה פתרון לשאלה 17

תרגיל 22:

תרגיל 23:

שאלה 24: