

מדעי המחשב ה'

פתרון בחינת הבגרות

פרק א - עיצוב תכנה

שאלה 1:

הערה: בין אם התור ממומש באמצעות מערך (בשפת C) ובין אם במימוש דינאמי של שרשרת חוליות (בשתי השפות), הצאת איבר מהתור מוציאה איבר גם מהתור המקורי, ולכן, כדי שנוכל לבנות את התור החדש מבלי לשנות את התור המקורי, נעתיק את התור המקורי לתור עזר ונעבוד עליו.

נכתוב פעולת עזר: **ספור-ומחק** (Q, x) המחזירה את מספר המופעים של x בתור Q ותוך כדי כך גם מוחקת מופעים אלו מהתור. (כמובן שניתן לפרק פעולה זו לשתי פעולות במחיר של עוד n צעדים).

תור-לפי-שכיחות (Q)

טענת כניסה: תור Q המכיל שלמים.
טענת יציאה: תור חדש המכיל עבור כל ערך מהתור המקורי, את הערך ואת מספר מופעיו.
הנחה: התור Q מאותחל.

(1) $Q_{temp} \leftarrow (Q)$ העתק-תור

(2) $Q1 \leftarrow$ אתחל-תור

(3) כל עוד לא תור-ריק (Q_{temp}) בצע:

(3.1) $x \leftarrow$ הוצא-מתור (Q_{temp})

(3.2) **ספור-ומחק** (Q_{temp}, x) $count \leftarrow 1 +$

(3.3) הכנס-לתור ($Q1, x$)

(3.4) הכנס-לתור ($Q1, count$)

(4) החזר את $Q1$

ספור-ומחק (Q, x)

טענת כניסה: תור Q , הערך x .

טענת יציאה: מוחזר מספר המופעים של x בתור. **שים לב:** אם x לא בתור יוחזר 0

בתהליך הספירה נמחקים כל המופעים של x מהתור.

הנחה: התור Q מאותחל.

(1) $count \leftarrow 0$

(2) $Q1 \leftarrow$ אתחל-תור

(3) כל עוד לא תור-ריק? (Q) בצע:

(3.1) $y \leftarrow$ הוצא-מתור (Q)

(3.2) אם $x = y$ אזי

$count \leftarrow count + 1$

(3.3) אחרת - הכנס-לתור ($Q1, y$)

(4) העתק-תור ($Q1$) $\leftarrow (Q)$

(5) החזר את $count$

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

יש סה"כ n איברים בתור.

עבור כל איבר יש לספור את מספר מופעיו - עוברים על כל התור ולכן n צעדים.

כל פעולות התור בסיבוכיות $O(1)$, ולכן הסיבוכיות ריבועית.

(במקרה הגרוע כל איבר מאיברי התור מופיע בדיוק פעם אחת).

שאלה 2:

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

S1

ב. האלגוריתם סוד 2 מחזיר את סכום איברי המחסנית.

ג. סוד 1 (S2, 2, 6, 4)

	n = 1	a	d	n	זימון
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)

} טענת כניסה: s - אורך צלע הריבוע. x ו-y - שיעורי פינה שמאלית עליונה של הריבוע.
color - צבע הריבוע.

טענת יציאה: ציור של ריבוע שפינתו השמאלית עליונה בנקודה (y,x), אורך צלעו s וצבעו color
הנחות: $0 \leq s \leq \min(N,M)$, $1 \leq x \leq N$, $1 \leq y \leq M$, $1 \leq \text{color} \leq 10$

הערות: 1. אורך צלע הריבוע יהיה הערך הקטן מבין אורך המסך N ורוחבו M - $\min(N,M)$.
2. בניגוד למקובל בפסקל ו-C (בהוראה gotoxy) בשאלה זו x מייצג מספר שורה, ו-y מספר עמודה.

(1) אם $s > 0$ אזי

(1.1) צייר-ריבוע (s, color, x, y)

(1.2) ריבוע (s-2, color, x+1, y+1)

procedure **squares** (s, color, x, y : integer);

פסקל:

begin

if s > 0 then

begin

drawSquare (s, color, x, y);

square (s-2, color, x+1, y+1);

end;

end;

void **squares** (int s, int color, int x, int y)

:C

{

if (s > 0)

{

drawSquare (s, color, x, y);

square (s-2, color, x+1, y+1);

}

}

הערה לתיעוד: אם רשמנו אלגוריתם מילולי וצירפנו לו תיעוד, אין חובה להעתיק את התיעוד שוב במימוש בסביבת העבודה. מספיק שהתיעוד יופיע פעם אחת בפתרון.

מיקום התחלתי לשומר המסך:

אורך הריבוע s הינו ערך שלם בתחום: $0 \leq s \leq \min(N,M)$ (עצירת הלולאה) (

מספר שורה x לא יעלה על ההפרש שבין מספר השורות במסך N ואורך צלע הריבוע s.

מספר עמודה y לא יעלה על ההפרש שבין מספר העמודות במסך M ואורך צלע הריבוע s.

צבע תקין הינו ערך מספרי בין 1 ו-10 (לפי נתוני השאלה).

הנחות: בתכנית הופעלה הפעולה randomize(); / randomize();
 M ו-N קבועים המוגדרים בתכנית.

פסקל: { פעולה המציירת ריבועים בגודל s ובצבע color במיקום אקראי על המסך. }

```

procedure screenSaver;
var
    s, color, x, y : integer;
    size : integer;
begin
    if M < N then
        size := M
    else size := N ;
    s := random (size + 1) ;           { מוחזר ערך בין 0 ל-size (כולל) }
    while s > 0 do
        begin
            color := random (10) + 1;
            x := random (N-size+1) + 1;
            y := random (M-size+1) + 1;

            squares (s, color, x, y);
            s := random (size + 1);
        end;
    end;

```

C: // פעולה המציירת ריבועים בגודל s ובצבע color במיקום אקראי על המסך.

```

void screenSaver ()
{
    int s, color, x, y ;
    int size;

    if (M < N)
        size = M ;
    else size = N ;

    s = random (size + 1) ;           // מוחזר ערך בין 0 ל-size (כולל)
    while (s > 0)
    {
        color = random (10) + 1;
        x = random (N-size+1) + 1;
        y = random (M-size+1) + 1;

        squares (s, color, x, y);
        s = random (size + 1);
    }
}

```

שאלה 4:

סידור-רשימות (L1, L2)

} טענת כניסה: L1 ו-L2 רשימות מאותחלות של מספרים שלמים חיוביים.
 { טענת יציאה: רשימה של ערכים מ-L2 הנמצאים במקומות של האיברים האי-זוגיים ב-L1.

- (1) $L3 \leftarrow$ אתחל-רשימה
- (2) $p3 \leftarrow$ עוגן-רשימה (L3)
- (3) $p1 \leftarrow$ עוקב-ברשימה (עוגן-רשימה (L1, (L1
- (4) כל עוד $p1 \neq$ סוף-רשימה (L1) בצע:
 - (4.1) $k \leftarrow$ אחזר-מרשימה (L1, p1)
 - (4.2) אם k זוגי אזי
 - (4.2.1) $(L2, k)$ מחק-איבר-במקום-k
 - (4.3) אחרת -
 - (4.3.1) $p2 \leftarrow$ מצא-איבר-במקום-k (L2, k)
 - (4.3.2) אם $p2 \neq$ סוף-רשימה (L2) אזי
 - (4.3.2.1) $x \leftarrow$ אחזר-מרשימה (L2, p2)
 - (4.3.2.2) הכנס-לרשימה (L3, p3, x)
 - (4.3.2.3) $p3 \leftarrow$ עוקב-ברשימה (L3, p3) (*)
 - (4.4) $p1 \leftarrow$ עוקב-ברשימה (L1, p1)
 - (5) החזר את L3

procedure List_Arrange (L1, L2 : list_type; var L3 : list_type);

var

p1, p2, p3 : pos_type;
 k, x : list_info_type ;

begin

list_init (L3);

p3 := list_anchor (L3);

p1 := list_next (L1, list_anchor (L1));

while (p1 <> list_end (L1)) do

begin

list_retrieve (L1, p1, k);

if k mod 2 = 0 then delete_element_place_k (L2, k)

else

begin

p2 := find_elemnt_place_k (L2, k);

if p2 <> list_end (L2) then

begin

list_retrieve (L2, p2, x);

list_insert (L3, p3, x);

p3 := list_next (L3, p3);

end;

end;

end;

end;

פסקל:

(*) אין חובת קידום p3 כי נכתב שאין חשיבות לסדר הכנסת האיברים ל-L3

```
{
    טענת כניסה : רשימה מאותחלת L ומספר שלם k (טיפוס איברי הרשימה).
    טענת יציאה : נמוק האיבר במקום ה- k ברשימה L.
    אם אין k איברים ב- L, לא יבוצע כלום.
}
```

```
procedure delete_element_place_k (var L : list_type; k : list_info_type);
var
    p : pos_type;
begin
    p := list_anchor (L);
    while (p <> list_end (L)) and (k > 0) do
        begin
            p := list_next (L, p);
            k := k - 1;
        end;
    if p <> list_end (L) then
        list_delete (L, p);
end;
```

```
{
    טענת כניסה : רשימה מאותחלת L ומספר שלם וחיובי k (טיפוס איברי הרשימה).
    טענת יציאה : מוחזר מקומו של האיבר במקום ה- k ברשימה L.
    אם אין k איברים ב- L, יוחזר המקום סוף-רשימה.
}
```

```
function find_elemnt_place_k (L: list_type; k: list_info_type): pos_type;
var
    p: pos_type;
    y: list_info_type;
begin
    p := list_anchor (L);
    while (p <> list_end (L)) and (k > 0) do
        begin
            p := list_next (L, p);
            k := k - 1;
        end;
    find_element_place_k := p;
end;
```

list_type List_Arrange (list_type L1, list_type L2)

:C

```

{
    list_info_type k, x ;
    pos_type p2;

    list_type L3 = list_init ();
    pos_type p3 = list_anchor (L3);
    pos_type p1 = list_next (L1, list_anchor (L1));
    while (p1 != list_end (L1))
    {
        k = list_retrieve (L1, p1);
        if ( k % 2 == 0 )
            delete_element_place_k (&L2, k);
        else
        {
            p2 = find_element_place_k (L2, k);
            if ( p2 <> list_end (L2) )
            {
                x = list_retrieve (L2, p2);
                list_insert (&L3, p3, x);
                p3 = list_next (L3, p3);
            }
        }
    }
}

```

(*) אין חובת קידום p3 כי נכתב שאין חשיבות לסדר הכנסת האיברים ל-L3

// טענת כניסה: רשימה מאותחלת L ומספר שלם וחיובי k (טיפוס איברי הרשימה).
// טענת יציאה: נמחק האיבר במקום ה-k ברשימה L.
// אם אין k איברים ב-L, לא יבוצע כלום.

void delete_element_place_k (list_type *L, list_info_type k)

```

{
    pos_type p = list_anchor (*L);
    while (p != list_end (*L) && k > 0)
    {
        p = list_next (*L, p);
        k -- ;
    }
    if (p != list_end (*L))
        list_delete (L, &p);
}

```

```
// טענת כניסה: רשימה מאותחלת L ומספר שלם k (טיפוס איברי הרשימה).  
// טענת יציאה: מוחזר מקומו של האיבר במקום ה-k ברשימה L.  
// אם אין k איברים ב-L, יוחזר המקום סוף-רשימה.
```

```
pos_type find_element_place_k(list_type L, list_info_type k)  
{  
    pos_type p = list_next(L, list_anchor(L));  
    while (p != list_end(L) && k > 0)  
    {  
        k -- ;  
        p = list_next(L, p);  
    }  
    return p;  
}
```


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: $45000H = 44000 + 1000$)
- 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,			0FFEFH		
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

שאלה 8:

```

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

שאלה 9:

$\begin{array}{ccc c} 10.00 & -4.00 & 2.00 & 16.00 \\ -5.00 & 4.00 & -3.00 & 14.00 \\ -5.00 & -8.00 & 11.00 & 88.00 \end{array}$	$R2 \leftarrow R2 + 0.5 * R1$	$g = 0.50$ $i = 1 \quad j = 1$ $c = 1..4$

$\begin{array}{ccc c} 10.00 & -4.00 & 2.00 & 16.00 \\ 0.00 & 2.00 & -2.00 & 22.00 \\ -5.00 & -8.00 & 11.00 & 88.00 \end{array}$	$R3 \leftarrow R3 + 0.5 * R1$	$g = 0.50$ $i = 1 \quad j = 3$ $c = 1..4$

$\begin{array}{ccc c} 10.00 & -4.00 & 2.00 & 16.00 \\ 0.00 & 2.00 & -2.00 & 22.00 \\ 0.00 & -10.00 & 12.00 & 96.00 \end{array}$	$R1 \leftarrow R1 + 2 * R2$	$g = 2.00$ $i = 2 \quad j = 1$ $c = 2..4$

$\begin{array}{ccc c} 10.00 & 0.00 & -2.00 & 60.00 \\ 0.00 & 2.00 & -2.00 & 22.00 \\ 0.00 & -10.00 & 12.00 & 96.00 \end{array}$	$R3 \leftarrow R3 + 5 * R2$	$g = 5.00$ $i = 2 \quad j = 3$ $c = 2..4$

$\begin{array}{ccc c} 10.00 & 0.00 & -2.00 & 60.00 \\ 0.00 & 2.00 & -2.00 & 22.00 \\ 0.00 & 0.00 & 2.00 & 206.00 \end{array}$	$R1 \leftarrow R1 + R3$	$g = 1.00$ $i = 3 \quad j = 1$ $c = 3..4$

$\begin{array}{ccc c} 10.00 & 0.00 & 0.00 & 266.00 \\ 0.00 & 2.00 & -2.00 & 22.00 \\ 0.00 & 0.00 & 2.00 & 206.00 \end{array}$	$R2 \leftarrow R2 + R3$	$g = 1.00$ $i = 3 \quad j = 2$ $c = 3..4$

$\begin{array}{ccc c} 10.00 & 0.00 & 0.00 & 266.00 \\ 0.00 & 2.00 & 0.00 & 228.00 \\ 0.00 & 0.00 & 2.00 & 206.00 \end{array}$		

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

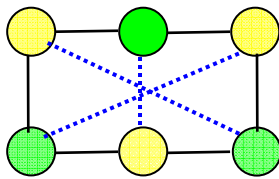
סוד (a,n) פותר מערכת משוואות ליניאריות בעלת n מעלמים בסגנון גאוס (סגנון כמעט דומה לשיטה).
 הפתרון מוצג בצורת: $m * x_i = b$

הפתרון לשאלה זו נכתב ע"י: שירלי רוזנברג כהן

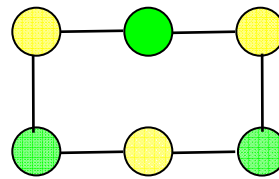
שאלה 10:

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

גרף Gb



גרף Ga



- 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:

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

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

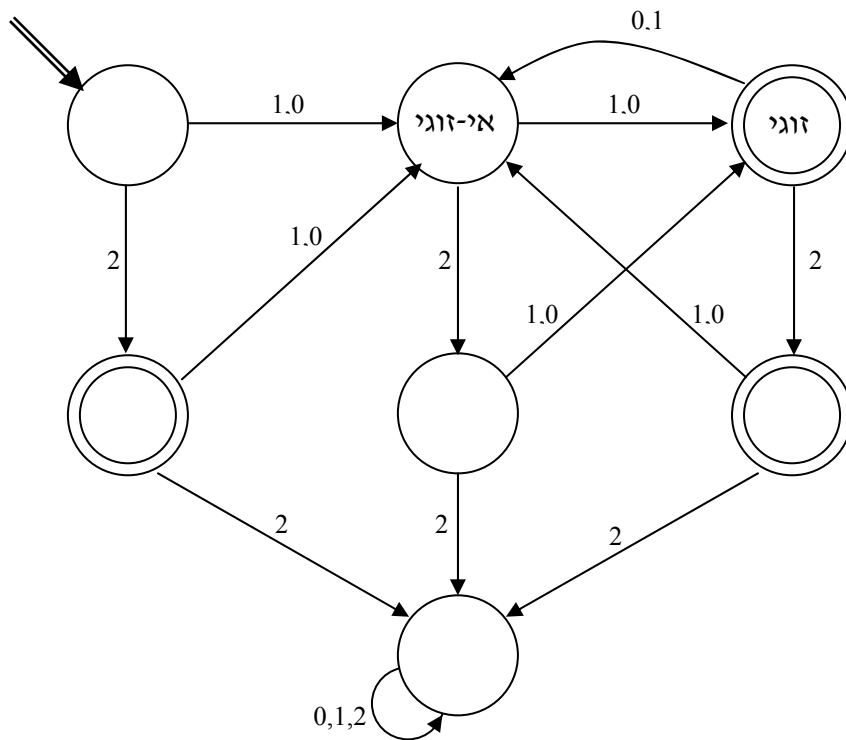
G3: $c^n a^i b^j c^m \mid i, j, n, m \geq 0$

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.



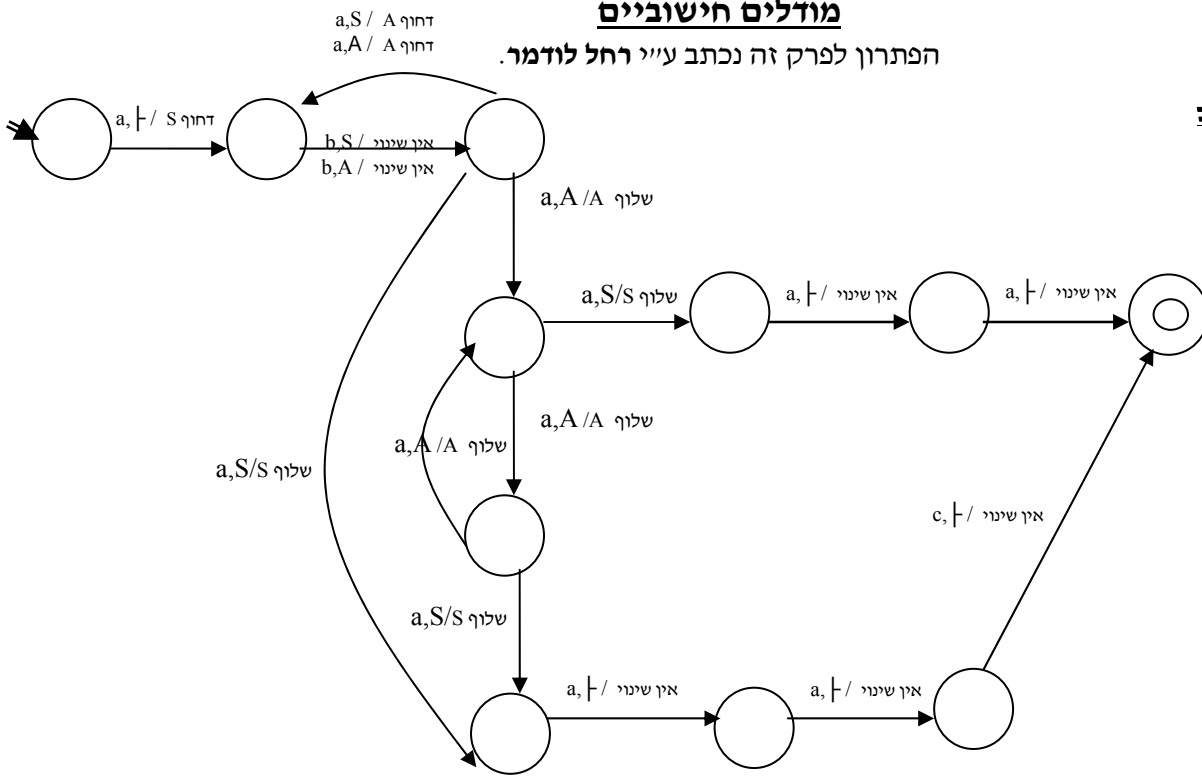
הערה: ניתן לבנות אוטומט בפחות מצבים כפי שמוצג בפתרון של רחל לודמר במודלים חישוביים - שאלה 16.

פרק ב'

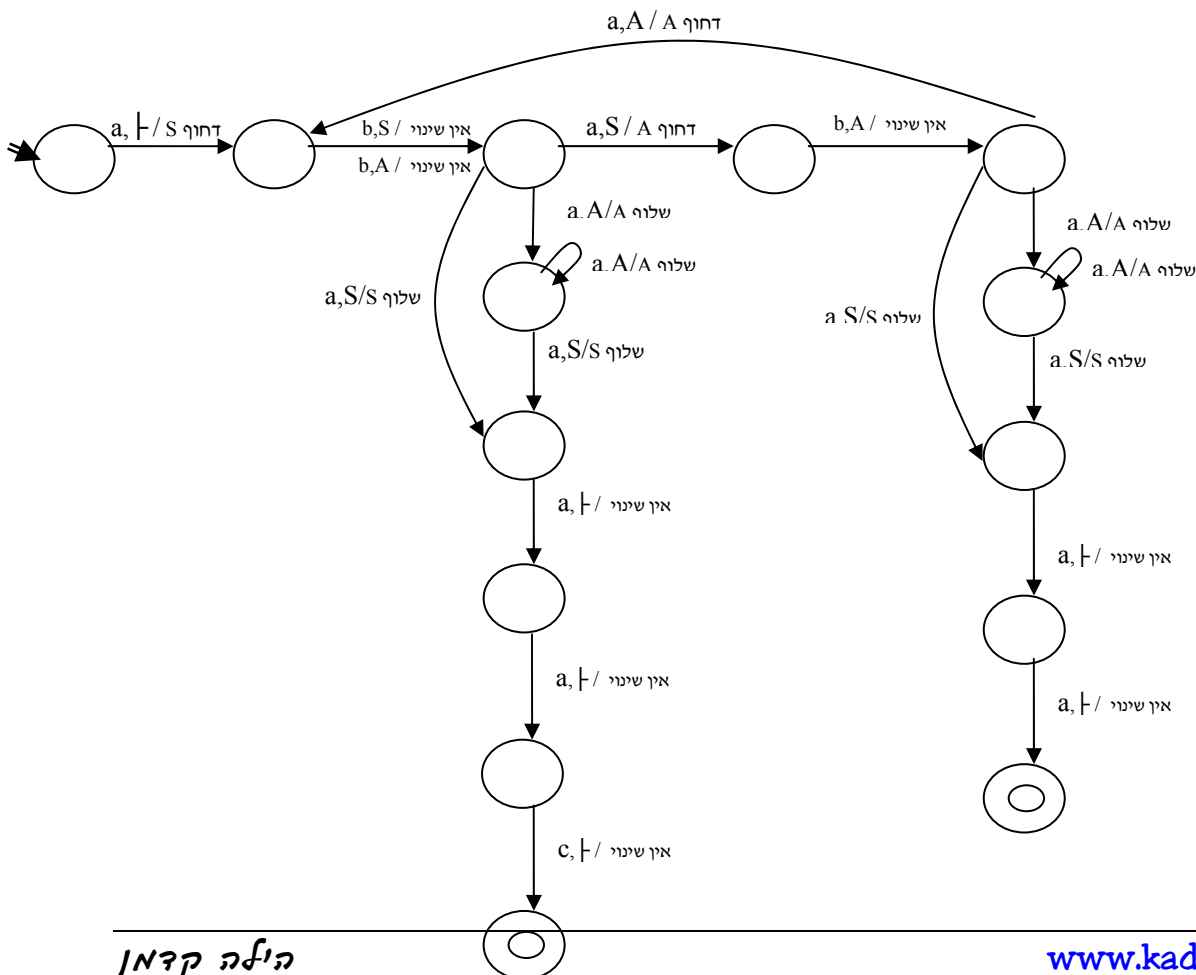
מודלים חישוביים

הפתרון לפרק זה נכתב ע"י רחל לודמר.

תרגיל 13:



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



תרגיל 14:

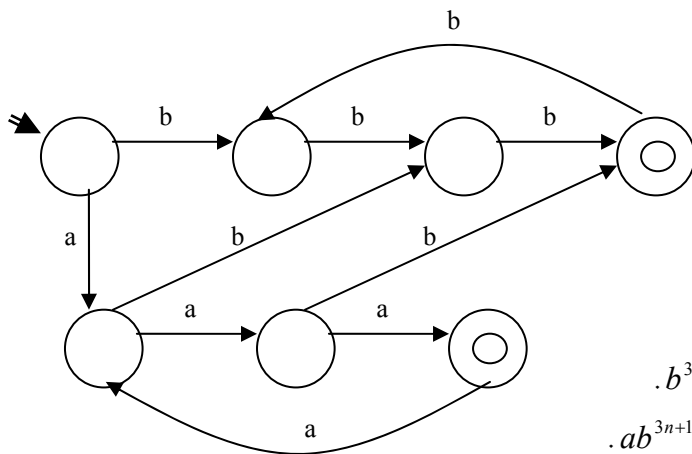
א. $w_1 = aaa$ שייכת לשפות L_1, L_3, L_4 .

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} .

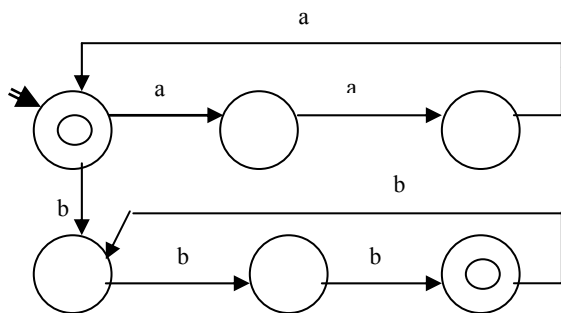
עבור $i=1$, השפה מכילה מילים מהצורה: $ab^{3n+1}a$.

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:

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

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

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

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.

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

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.

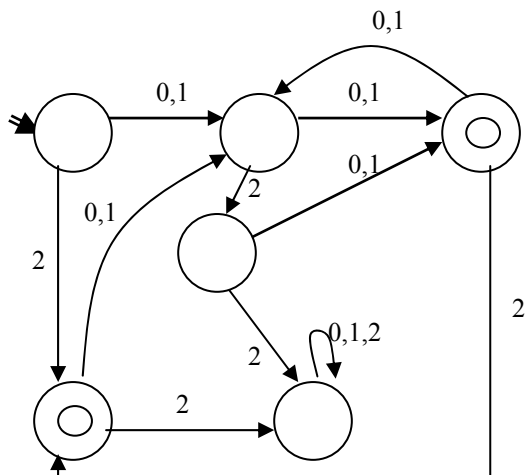
ג. (i) הטענה $R(L_2) = L_2$ נכונה.

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

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

(ii) הטענה $R(L_1) = L_3$ נכונה.

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



שאלה 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

תראו 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;
}
```

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; }
}
    
```

שאלה 20:

פרק ב'

תכנות מונחה עצמים C#

מי האורח שיעח פתרון ?

תרגיל 21:

ראה פתרון לשאלה 17

תרגיל 22:

תרגיל 23:

שאלה 24: