

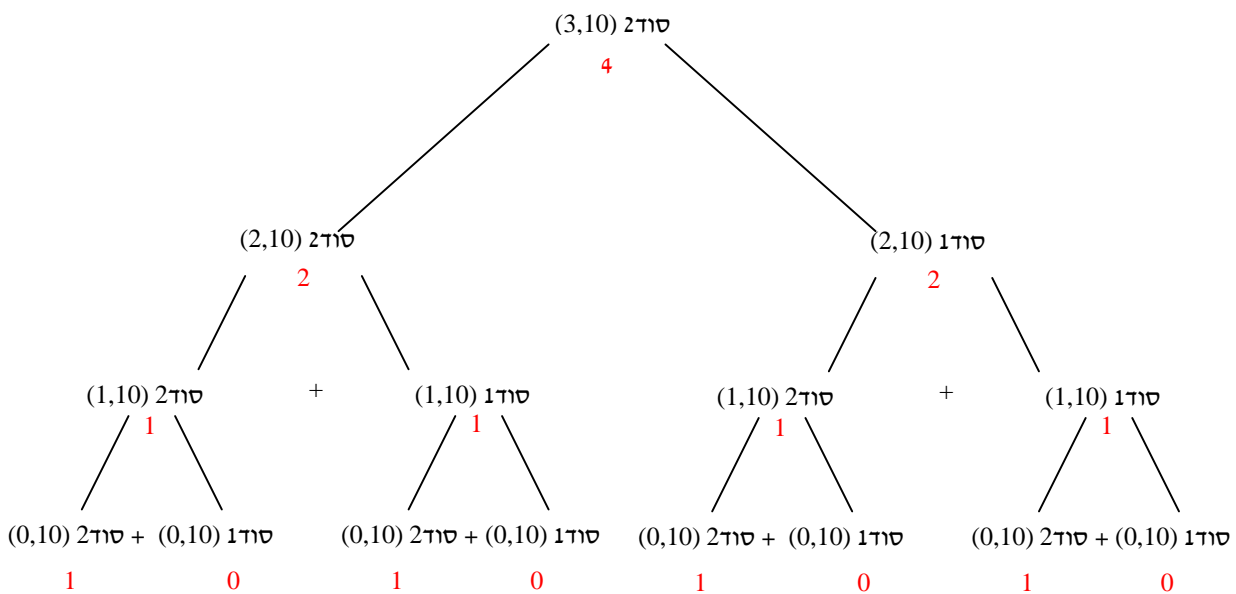
מדעי המחשב ב'
פתרון בחינת הפארות
פרק א - עיצוב תכנה

שאלה 1:

- א. מוחזר 4
- ב. מוחזר 8
- ג. כן. בשני המקרים מתבצע צעד 2.1.1 ומוחזר סוד1 (n,n) + סוד2 (n,n)

הערה: האלגוריתם מחזיר את הערך 2^{x-1} (לא נדרש לכתוב זאת בבחינה).

דוגמא למעקב אחר האלגוריתם בסעיף א':



שאלה 2:

האם-פרו-ורבו? (T)

} פעולה המחזירה "אמת" אם קיים צומת ב-T אשר יש לו לפחות שני נכדים,
{ וכל נכד מבין אחר. הנחה: T מאותחל.

(1) אם העץ T הוא null, החזר "שקר"

(2) אם ל-T יש שני בנים, אזי

(2.1) $T1 \leftarrow T$ בן-שמאלי של T

(2.2) $T2 \leftarrow T$ בן-ימני של T

(2.3) אם ל-T1 יש בן-שמאלי או שיש לו בן-ימני וגם ל-T2 יש בן-שמאלי או שיש לו בן-ימני

החזר "אמת"

(3) החזר האם-פרו-ורבו (בן-שמאלי של T) או האם-פרו-ורבו (בן-ימני של T)

אפשר להמיר את צעד 2 בהוראה:

(2) אם מספר-בנים (בן-שמאלי של T) < 0 וגם מספר-בנים (בן-ימני של T) < 0

אזי - החזר "אמת"

ואז יש לממש את הפעולה: מספר-בנים (T)

פתרון בשפת Java:

--- פעולה המחזירה "אמת" אם קיים צומת ב-T אשר יש לו לפחות שני נכדים, וכל נכד מבין אחר. ---

```
public static boolean pruRvuTree (BinTreeNode<Integer> t)
{
    if (t == null)
        return false;
    if (t.getLeft() != null && t.getRight() != null)
    {
        BinTreeNode<Integer> t1 = t.getLeft();
        BinTreeNode<Integer> t2 = t.getRight();
        if ( (t1.getLeft() != null || t1.getRight() != null) &&
            (t2.getLeft() != null || t2.getRight() != null) )
            return true;
        return pruRvuTree (t.GetLeft()) || pruRvuTree (t.GetRight());
    }
}
```

נכתב ע"י ראמי ג'באלי

פתרון בשפת C#:

--- פעולה מחזירה אמת אם הצומת עלה אחרת תחזיר הפעולה שקר ---

```
public static bool IsLeaf(BinTreeNode<int> bt)
{
    return (bt.GetLeft() == null) && (bt.GetRight() == null);
}
```

--- פעולה מקבלת עץ ומחזירה אמת אם העץ פרו ורבו אחרת יוחזר שקר ---

```
public static bool ProRbo(BinTreeNode<int> t)
{
    if (t == null)
        return false;
    if (t.GetLeft() != null && !IsLeaf(t.GetLeft()) &&
        t.GetRight() != null && !IsLeaf(t.GetRight()))
        return true;
    return ProRbo(t.GetLeft()) || ProRbo(t.GetRight());
}
```

ההוראה בודקת ששי בן וגם שהבן אינו עלה.
שימו ♥:
אם אין בן, אסור לבדוק אם הוא עלה.
אם החלק הראשון של הבדיקה יחזיר "שקר" לא
יתבצע החלק האחר הבודק עלה.

שאלה 3:

א. פתרון התרגיל בשפות Java ו-C# מחייב מידול של הבעיה למחלקות הבאות:
מחלקת שירות קוביית-משחק - Dice המכילה פעולה שמחזירה את סכום 5 ההטלות.
המחלקה Game המממשת את שאר פעולות המשחק כפי שהופיעו בממשק.
פעולות הממשק נתונות ואין צורך לממש אותן במבחן.
לצרכי הצגת פתרון מלא לשאלה מומשו שתי מחלקות אלו.

ייצוג: ניקוד השחקנים יישמר במערך בגודל 17 של שלמים (מערך צוברים).
מערך זה יהיה תכונה של המחלקה Game.

שים ♥: מספרי השחקנים מתחיל ב-1. מקומו של השחקן הראשון במערך הוא בתא מספר 0.
הפעולה winner מבצעת היסט של מספר התא המכיל מינימום למספרו של המנצח.

ב. משחק (g) { פעולה המנהלת משחק עד להכרזת המנצח. הנחה: המשחק g מאותחל }
בצע 15 פעמים:

סיבוב-משחק (g)

כל עוד אין מנצח() במשחק, בצע:

סיבוב-משחק (g)

החזר את מספרו של המנצח במשחק

סיבוב-משחק (g) { פעולה המנהלת משחק של כל אחד מ-17 השחקנים }

עבור i משחקן ראשון ועד שחקן אחרון, בצע:

א. בחר מספר ← announce

ב. זרוק-קובייה-5-פעמים() ← sum

ג. עדכן-ניקוד (i, announce, sum)

פתרון בשפת Java : (הפתרון בשפת C# זהה לפתרון ב-Java למעט הוראת הפלט שבפעולה הראשית)

```

/*
 *          Dice משחק קוביית-משחק
 * בגרות 2004 - שאלה 3 - משחק קובייה "הטל וצבור"
 */
public class Dice
{
    //--- פעולה פנימית המחזירה את תוצאת ההטלה של הקובייה ---
    private static int throwDice ()
    {
        return (int) (Math.random()*6)+1;
    }

    //--- פעולה המחזירה את סכום 5 ההטלות של הקובייה ---
    public static int throw5times()
    {
        int sum = 0;
        for (int i = 0 ; i < 5 ; i++)
            sum = sum + throwDice ();
        return sum;
    }
}

/*
 *          Game המחלקה
 * המממשת את פעולות המשחק שהוצגו במבחן
 */
public class Game
{
    public static int N = 5;          // מייצג את מספר השחקנים במשחק
                                     // לפי השאלה צריכים להיות 17 שחקנים

    private int [] game;

    //--- פעולה המחזירה משחק חדש ---
    public Game()
    {
        this.game = new int [N];
        this.init();
    }

    //--- פעולה פנימית המאתחלת את המשחק ---
    private void init()
    {
        for (int i = 0; i < N ; i++)
            this.game[i] = 0;
    }

    //--- פעולה המעדכנת את הניקוד של שחקן ---
    public void updatePlayerScore (int num, int announce, int sum)
    {
        int value = Math.abs(sum - announce);
        if (value == 0)

```

```

        this.game[num] = 0;
    else
        this.game[num] += value;
    }

    //--- פעולה המחזירה את מספרו של השחקן המנצח ---
    //--- אם אין מנצח יוחזר 0 ---
    public int winner()
    {
        int min = 0;
        boolean oneWinner = true; //--- C# בשפת bool ---
        for (int i = 1 ; i < N ; i++)
        {
            if (this.game[i] < this.game[min])
            {
                min = i;
                oneWinner = true;
            }
            else
                if (this.game[i] == this.game[min])
                    oneWinner = false;
        }
        if (oneWinner)
            return min + 1; //--- מספרי השחקנים הוא בין 1 ו-N ---
        else //--- ואילו המערך מתחיל במקום 0 ---
            return 0;
    }

    //--- פעולה המחזירה מחרוזת המתארת את מצב השחקנים המשחק ---
    public String toString()
    {
        String str = "[";
        for (int i = 0 ; i < N-1 ; i++)
            str += this.game[i] + ", ";
        str += this.game[N-1] + "]";
        return str;
    }
}

```

הפתרון הנדרש בתרגיל:

```

public class T3_2004_Dice_Game
{
    /**
     *          בגרות 2004 - שאלה 4
     *          משחק הקובייה - הטל וצבור
     *
     *          לצרכי נוחות שונה מספר השחקנים במשחק ל- 5
     *          וכן מספר הסיבובים הראשוני שונה ל- 5
     */
}

```

```
//--- פעולה המנהלת משחק של כל אחד מהשחקנים ---
static void roundOfGame (Game g)
{
    int i, announce, sum ;
    for (i = 0 ; i < 5 ; i++)
    {
        announce = chooseAnnounce();
        sum = Dice.throw5times();
        g.updatePlayerScore(i, announce, sum);
    }
}

//--- פעולה המדמה את בחירת המספר ---
//--- סכום הקוביות ב- 5 יהיה מספר בין 5 - 30 ---
static int chooseAnnounce()
{
    return (int) (Math.random()*26 + 5);
}

//--- תכנית המנהלת משחק עד להכרזת המנצח ---
static int playGame (Game g)
{
    for (int j = 0 ; j < 5 ; j++)
        roundOfGame(g) ;
    while (g.winner() == 0)
        roundOfGame(g);
    return g.winner();
}

public static void main(String[] args)
{
    Game g = new Game();
    System.out.println("start game: " + g.toString());

    int win = playGame(g);
    System.out.println("winner is: " + win);
}

/*~~~~~ פלט התכנית ~~~~~
start game: [0, 0, 0, 0, 0]

round #0 : [10, 1, 9, 18, 4]           לאחר 5 סיבובים ראשונים:
round #1 : [20, 10, 11, 25, 15]
round #2 : [39, 12, 14, 44, 21]
round #3 : [47, 15, 15, 48, 28]
round #4 : [57, 18, 18, 53, 38]

[65, 27, 27, 54, 46]           כל עוד אין מנצח:
[71, 40, 33, 67, 49]
winner is: 3

*/
```

שאלה 4:

הערה: בבחינת הבגרות נדרש "כתוב תת-תכנית בסביבת העבודה". מטעמי נוחות נכתב תחילה האלגוריתם והתכנית בהמשך. אלגוריתם בלבד לא יתקבל במקום תכנית.

מכפלת-בלוקים (M)

```

}
פעולה המקבלת מחסנית M ומחזירה מחסנית חדשה שכל איבר בה הוא מכפלת האיבר
שב"בלוק" במספר מופעיו. "בלוק" הינו רצף של לפחות שני איברים זהים.
הנחה: המחסנית מאותחלת.
{
(1) אתחל-מחסנית S ←
(2) אם המחסנית M לא ריקה, אזי:
(2.1) שלוף ממחסנית M ← num
(2.2) counter ← 1
(3) כל עוד המחסנית M לא ריקה, בצע:
(3.1) שלוף ממחסנית M ← x
(3.2) אם x = num
    אז - counter ++
(3.3) אחרת -
(3.3.1) אם counter > 1
    אז - דחוף למחסנית S את counter*num
(3.3.2) num ← x
(3.3.3) counter ← 1
(4) אם counter > 1
    אז - דחוף-למחסנית S את counter*num
(5) חזור את S

```

ב. סיבוכיותה של כל אחת מפעולות ממשק המחסנית - $O(1)$.
 הפעולה עוברת פעם אחת על כל איברי המחסנית, ובמהלך האלגוריתם מבצעת פעולות חישוביות שסיבוכיותן קבועה. ומכאן שסיבוכיות הפעולה כולה $O(n)$.

פתרון בשפת Java :

```
// פעולה המקבלת מחסנית M ומחזירה מחסנית חדשה שכל איבר בה הוא מכפלת האיבר
// שב"בלוק" במספר מופעיו. "בלוק" הינו רצף של לפחות שני איברים זהים.
// הנחה: המחסנית מאותחלת.
public static Stack<Integer> multiplyBlocks (Stack<Integer> M)
{
    Stack<Integer> S = new Stack<Integer>();
    int num = 0, x, counter = 1 ;
    if (! M.isEmpty())
        num = M.pop();
    while (! M.isEmpty())
    {
        x = M.pop();
        if (x == num)
            counter ++;
        else
        {
            if (counter > 1)
                S.push(num * counter);
            num = x;
            counter =1;
        }
    }
    if (counter > 1)
        S.push (num*counter);
    return S ;
}
```

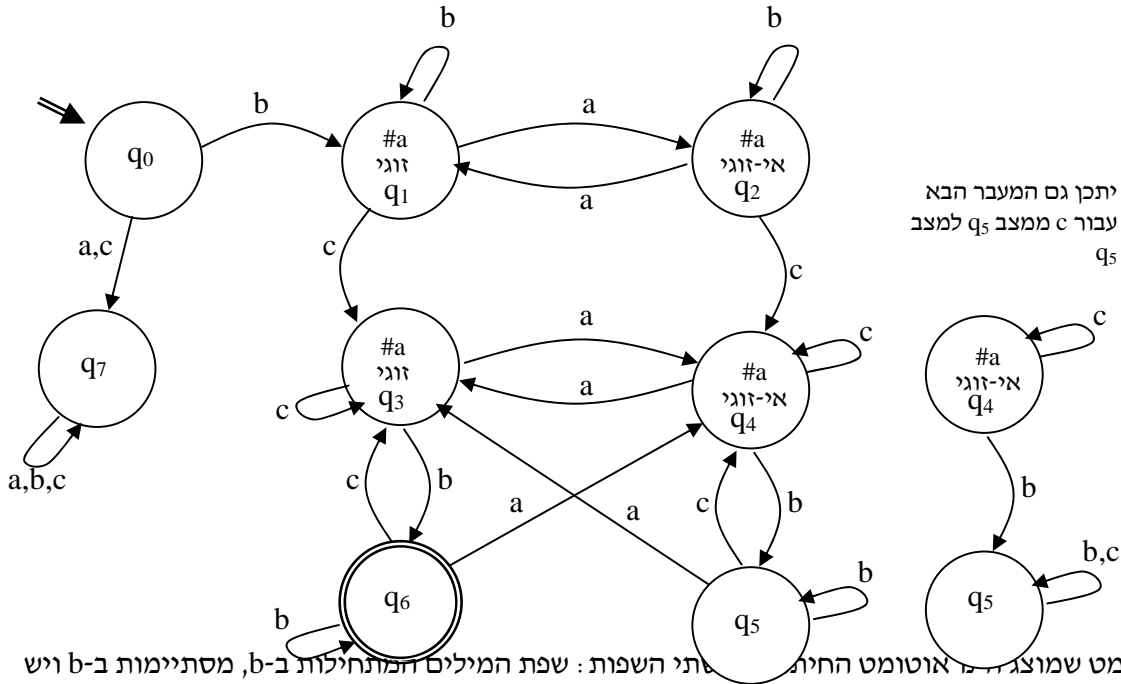
נכתב ע"י ראמי ג'באלי

פתרון בשפת C# :

```
// פעולה המקבלת מחסנית s ומחזירה מחסנית חדשה שכל איבר בה הוא מכפלת האיבר
// שב"בלוק" במספר מופעיו. "בלוק" הינו רצף של לפחות שני איברים זהים.
// הנחה: המחסנית מאותחלת.
public static Stack<int> Block(Stack<int> s)
{
    int c = 1;
    int x = 0;
    Stack<int> s2 = new Stack<int>();
    while (!s.IsEmpty())
    {
        x = s.Pop();
        if (!s.IsEmpty() && (s.Top() == x))
            c++;
        else
        {
            if (c > 1)
                s2.Push(c * x);
            c = 1;
        }
    }
    return s2;
}
```

פרק ב'
תורת המחשב

שאלה 11:



יתכן גם המעבר הבא
עבור c ממצב q_5 למצב q_5

הערה: האוטומט שמוצג הוא אוטומט החיטה שתי השפות: שפת המילים המתחילות ב-b, מסתיימות ב-b ויש לפחות c אחת, ושפת כל המילים שבהן מספר ה-a-ים זוגי. ניתן לבנות אוטומט בעל מספר קטן יותר של מצבים.

שאלה 12:

- א. $S \rightarrow ABD$
 $A \rightarrow bB \mid \epsilon$
 $B \rightarrow aA \mid \epsilon$
 $D \rightarrow cDd \mid \epsilon$

- ב. המילים: (i), (ii), (iv) שייכות לשפה. המילה (iii) אינה שייכת לשפה. לא יתכן בשפה שתי אותיות b רציפות.

ג. $L = \{ (ba)^* b^j c^i d^i (ab)^* a^k \mid j, k = \{0 \text{ or } 1\}, i \geq 0 \}$

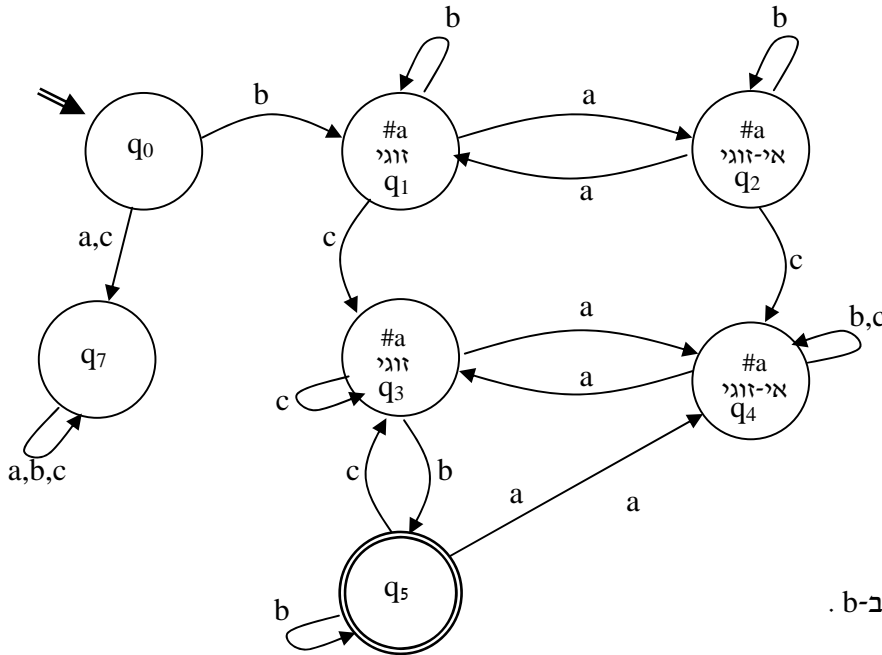
- אוסף כל המילים שלהן 3 חלקים:
 חלק I: רצף של a ו-b לסירוגין המתחיל ב-b. (אין שתי אותיות זהות סמוכות זו לזו).
 חלק II: רצף של אותיות c ואחריו רצף באורך זהה של אותיות d. אורך הרצף גדול או שווה ל-0. (אין אות c אחרי האות d ואין אף אות d לפני האות c).
 חלק III: רצף של a ו-b לסירוגין המתחיל ב-a. (אין שתי אותיות זהות סמוכות זו לזו).

הערה: אם חלק I מסתיים ב-a וחלק II ריק, יהיה מקרה יחיד של שתי אותיות a הסמוכות זו לזו.

מודלים חישוביים

ע"י: רחל לודמר - תיכון כצנלסון כפר-סבא

שאלה 13:



- האוטומט מקבל את השפה:
- המתחילות ב-b, מסתיימות ב-b.
 - יש לפחות c אחת.
 - מספר ה-a ים זוגי.

שאלה 13:

א. (i) $a^4 b^8 a^4$

המילה שייכת ל- L_1 .

המילה לא שייכת ל- L_2 (מספר אותיות ה-a (8) שווה למספר אותיות ה-b (8)).

המילה לא שייכת ל- L_3 (מספר אותיות ה-b הוא זוגי).

(ii) aabbabb

המילה אינה שייכת ל- L_1 (אינה בתבנית של $a^n b^m b^n$).

המילה שייכת ל- L_2 (מספר אותיות a שונה ממספר אותיות b).

המילה שייכת ל- L_3 (מספר אותיות b הוא אי-זוגי).

(iii) $L_1 \# (L_2 \cdot L_3) = \{a^n b^{2n} a^n \mid n > 0\}$

(מספר ה-b ים זוגי ושווה בסה"כ למספר ה-a).

ב. L_C, L_B, L_A - הן שפות רגולריות.

$$L_A \# (L_B, L_C) = L_A \cap (\overline{L_B} \cup \overline{L_C}) = L_A \cap \overline{L_B} \cap \overline{L_C}$$

L_C, L_B הן רגולריות. ע"י תכונות הסגירות של שפות רגולריות, משלים של שפה רגולרית הוא

רגולרי, ולכן $\overline{L_B}$ ו- $\overline{L_C}$ רגולרי. כמו כן, חיתוך של שפות רגולריות הוא רגולרי, לכן הוא רגולרי, ומכאן שהשפה רגולרית.

שאלה 15:

(i) הטענה $aaaabbbb \in L_1^2$ לא נכונה. נימוק: $L_1^2 = \{a^n b^n \cdot a^k b^k \mid n, k > 0\}$ והמילה $aaaabbbb$ אינה שייכת ל- L_1^2 .

(ii) הטענה $R(L_1) \cdot d \cdot R(L_1) \subset R(L_2)$ נכונה.

נימוק: $R(L_1) = \{b^n a^n \mid n > 0\}$

$R(L_2) = \{b^k a^k d b^n a^n \mid n, k \geq 0\}$

$R(L_1) \cdot d \cdot R(L_1) = \{b^n a^n \mid n > 0\} \cdot d \cdot \{b^k a^k \mid k > 0\}$ לכן:

$R(L_2) = \{b^n a^n d b^k a^k \mid n \geq 0, k \geq 0\}$ וזו קבוצה חלקית לשפה:

(iii) הטענה $L_1 \cap L_2 = (L_2 \cup L_3) \cap \{d\}$ לא נכונה.

נימוק: $L_1 \cap L_2 = \{a^n b^n \mid n > 0\} \cap \{a^n b^n d a^k b^k \mid n, k \geq 0\} = \emptyset$

החיתוך הוא קבוצה ריקה. כי, L_1 אינה מכילה את האות d.

הקבוצה: $(L_2 \cup L_3) \cap \{d\} = (L_2 \cap \{d\}) \cup (L_3 \cap \{d\}) = \{d\} \cup \emptyset = \{d\}$

מאחר והקבוצות זרות, הן אינן שוות. לכן, הטענה אינה נכונה.

(iv) הטענה $L_1 = (R(L_3) \cap L_1)$ נכונה.

נימוק: $R(L_3) \cap L_1 = \{ab \text{ שמכילות את הרצף } ab \text{ מעל } \{a,b\} \text{ כל המילים מעל } \{a,b\} \} \cap \{a^n b^n \mid n > 0\} = \{a^n b^n \mid n > 0\} = L_1$

החיתוך הוא הקבוצה הקטנה L_1 - כי כל אחת מהמילים ב- L_1 מכילה את הרצף ab (שתי האותיות האמצעיות במילה).

(v) הטענה $\overline{L_3} \subset L_1$ לא נכונה.

נימוק: $\overline{L_3} = \{ba \text{ מעל } \{a,b\} \text{ שאינן מכילות את הרצף } ba\}$

L_1 היא אוסף כל המילים מעל $\{a,b\}$ שאינן מכילות את הרצף ba, אבל היא יותר קטנה

מ- $\overline{L_3}$ (L_1 קבוצה עם "סדר מסוים" של אותיות a ו-b).

$\overline{L_3}$ זהו אוסף כללי של אותיות a, b שאינן מכילות את הרצף ba, לכן $L_1 \subset \overline{L_3}$ (ולא מתקיים

$\overline{L_3} \subset L_1$).

שאלה 16:

