

מבחן בגרות 2013

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

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

תוכן עניינים של פתרון המבחן

פרק ראשון - עיצוב תוכנה

- שאלה 1 : רשימות [מעקב]
 שאלה 2 : עצים, מחסנית [פיתוח פעולה]
 שאלה 3 : טיפוסים, יעילות [פיתוח פעולות]
 שאלה 4 : עצים [ניתוח, נכונות של טענות]

פרק שני - מודלים חישוביים

- שאלה 13 : רגולריות [השלמת מודל, זיהוי שפה]
 שאלה 14 : חופשיות הקשר [השלמת מודל, בדיקת שייכות מילים לשפה, בדיקת נכונות של טענות]
 שאלה 15 : אוטומט מחסנית [בנה אוטומט]
 שאלה 16 : פעולות על מילים ושפות [בדיקת שייכות מילים לשפה, בדיקת נכונות של טענות]

פתרון שאלה 1

נושאים מרכזיים: רשימות
סוג השאלה: כתוב פעולה

list1: 2 → 4 → 5 → 1 → 1 → 9 → ||

list2: 2 → 4 → 5 → 1 → 4 → ||

א. מעקב אחת כל אחת מהפעולות Sod1, Sod2, Sod3 עם הרשימות:

הפעולה Sod1

node1	node2	i	return
↑2	↑2	1	
↑4		2	
↑5		3	
↑1		4	false

הפעולה Sod2

node1	node2	return
↑2	↑2	
↑4	↑4	
↑5	↑5	
↑1	↑1	
↑1	↑4	false

הפעולה Sod3

node1	found	node2	return
↑2	false true	↑2 ↑4	
↑4	false true	↑2 ↑4 ↑5	
↑5	false true	↑2 ↑4 ↑5 ↑1	
↑1	false true	↑2 ↑4 ↑5 ↑1 ↑4	
↑1	false true	↑2 ↑4 ↑5 ↑1 ↑4	
↑9	false	↑2 ↑4	

		↑5 ↑1 ↑4 null	
			false

ב. סיבוכיות זמן הריצה כל אחת מהפעולות Sod1, Sod2, Sod3

סיבוכיות זמן הריצה של הפעולה Sod1: $O(1)$.
 נימוק: מספר האיטרציות של הלולאה הוא לכל היותר 4, ועל כן משך הביצוע אינו תלוי באורך הקלט.

סיבוכיות זמן הריצה של הפעולה Sod2: $O(n)$.
 n מייצג את מספר האיברים ברשימה הקצרה יותר.
 נימוק: מתבצעת סריקה במקביל על שתי הרשימות.

סיבוכיות זמן הריצה של הפעולה Sod3: $O(n*m) \cong O(n^2)$.
 ■ n מייצג את מספר האיברים ברשימה list1.
 ■ m מייצג את מספר האיברים ברשימה list2.

נימוק: במקרה הגרוע ביותר, עבור כל איבר ב list1 מתבצעת סריקה של הרשימה list2.
 הערה: בשלוש הפעולות Sod1, Sod2, Sod3, כל הפעולות מהמחלקה Node<T> הן בסיבוכיות $O(1)$ ולכן אינן משפיעות על סיבוכיות זמן הריצה.

פתרון שאלה 2

נושאים מרכזיים: עצים
 סוג השאלה: כתוב פעולה

```
public static int Big(BinTreeNode<int> t)
{
    if (t == null)
        return -1;
    int big = t.GetInfo();
    int bigLeft = Big(t.GetLeft());
    int bigMiddle = Big(t.GetMiddle());
    int bigRight = Big(t.GetRight());
    return Math.Max(Math.Max(big, bigLeft), Math.Max(bigMiddle, bigRight));
}
public static bool NoThree(BinTreeNode<int> tr)
{
    if (tr == null)
        return true;
    if (tr.GetLeft() != null && tr.GetMiddle() != null && tr.GetRight() != null)
        return false;
    return ( NoThree(tr.GetLeft()) && NoThree(tr.GetMiddle()) &&
    NoThree(tr.GetRight()) );
}
```

פתרון שאלה 3

נושאים מרכזיים: טיפוסים, רשימות, יעילות
סוג השאלה: כתוב פעולה, בנה מחלקה, סיבוכיות זמן ריצה

סיפור המסגרת: המחלקה Stones היא אוסף של כל אבני Bistone האפשריות. במחלקה אין שתי אבנים שעליהן אותו צירוף של מספרים. בסה"כ יש במחלקה 49 stones אבני Bistone.

א. $(1)+(2)+(3)$ המחלקה Bistone, פעולה בונה ופעולה פנימית

```
public class BiStone
{
    private int num1;
    private int num2;
    public BiStone (int num1, int num2)
    {
        this.num1 = num1;
        this.num2 = num2;
    }
    public bool EqualsDig()
    {
        return (num1 == num2%10);
    }
}
public class Stones
{
    private BiStone[] stones = new BiStone[49];

    public Stones()
    {
        int p = 0;
        for (int i=0 ; i<=6 ; i++) // לולאה מקוננת לכל האפשרויות של האבנים
            for (int j=10 ; j<=16 ; j++)
            {
                this.stones[p] = new BiStone(i,j); // יצירת אבן והכנסתה למערך
                p++;
            }
    }
}
```

ב. (1) פעולה שמקבלת מספר שלם num, גדול מ-1 ורשימה lst המכילה מספרים שלמים גדולים מ-0, שכולם קטנים מ num.

```
// פעולת עזר
public static bool Exist(List<int> list, int num)
// הפעולה מקבלת רשימה ומספר שלם
// הפעולה מחזירה אמת אם המספר נמצא ברשימה או שקר אחרת
{
    Node<int>p = list.GetFirst();
    while (p!=null)
    {
```

```

    if (p.GetInfo()==num)
        return true;
    p = p.GetNext();
}
return false;
}
public static List<int> NewList(List <Integer> list1, int num)
// הפעולה מקבלת רשימה ומספר שלם num
// הפעולה מחזירה רשימה חדשה של כל המספרים בטווח num-1 שלא נמצאים ברשימה שהתקבלה
{
    for (int i=1 ; i<num ; i++)
        if (!Exist(list1,i)) // אם המספר לא נמצא ברשימה
            list2.Insert(null,i); // הכנסת המספר שלא נמצא לרשימה החדשה
    return list2;
}

```

ב (2) סיבוכיות זמן הריצה של הפעולה הנ"ל

סיבוכיות זמן הריצה של הפעולה היא $O(n^2)$ כאשר n מייצג את num .
 נימוק: מתבצעת לולאה על כל אחד מהערכים $1 - (num-1)$ ועבור כל ערך מתבצעת בפעולה Exist סריקה של הרשימה.

פתרון שאלה 4

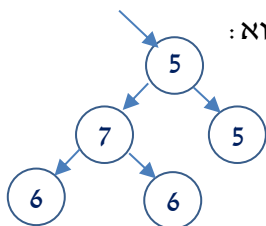
נושאים מרכזיים: עצים
 סוג השאלה: ניתוח, נכונות של טענות

נתוני השאלה

הפעולה amir:	העץ
<pre> public static void Amir(BinTreeNode<int> t, int x) { if (t != null) { Amir(t.GetLeft () , x + 1) ; if ((t.GetLeft() == null) && (t.GetRight() == null)) t.SetInfo (x) ; Amir(t.GetRight () , x + 1) ; } } </pre>	<pre> graph TD 5((5)) --> 7((7)) 5((5)) --> 3((3)) 7((7)) --> 4((4)) 7((7)) --> 6((6)) </pre>

א.

(i) העץ tree שיתקבל לאחר הפעלת הזימון Amir(tree, 4) הוא :



(ii) הפעולה Amir מבצעת בעבור עץ כלשהו ו $x=0$:

א. הפעולה מעדכנת את ערכי העלים להיות סכום ערך הרמה שלהם X .

ב. עבור $x=0$ הפעולה תעדכן את ערך העלים להיות הרמה שלהם.

ב.

(1) הטענה נכונה בחלק מהמקרים. הטענה נכונה במקרה בו $x=y=z$. בכל מקרה אחר הטענה אינה נכונה.

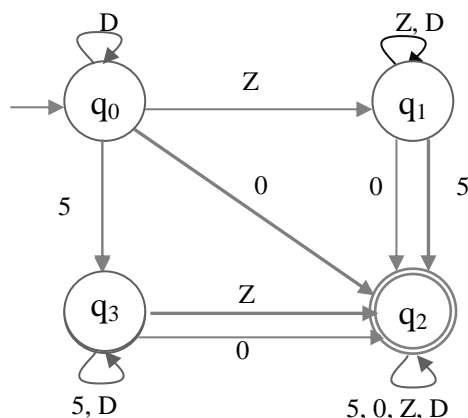
(2) הטענה תמיד אינה נכונה. סריקה תוכנית משמאל לימין של עץ חיפוש מחזירה את ערכי העץ ממוינים בסדר לא יורד.

פתרון סרק לני - מוליס היילוב"ס 2013

פתרון שאלה 13

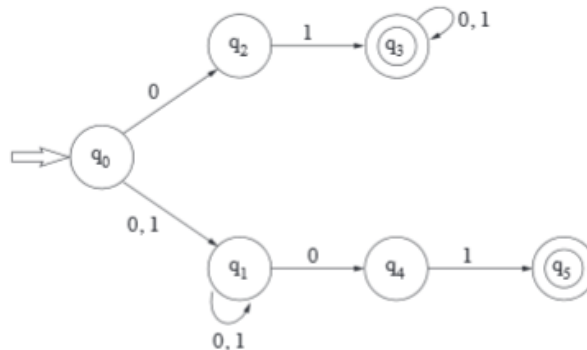
נושא מרכזי: אוטומט סופי דטרמיניסטי
סוג השאלה: השלמת מודל, נכונות של טענות

א. השלמת אוטומט



ב. מתקבל/לא מתקבל על-ידי האוטומט

לפניך אוטומט סופי לא-דטרמיניסטי המקבל את השפה L מעל הא"ב $\{0, 1\}$.

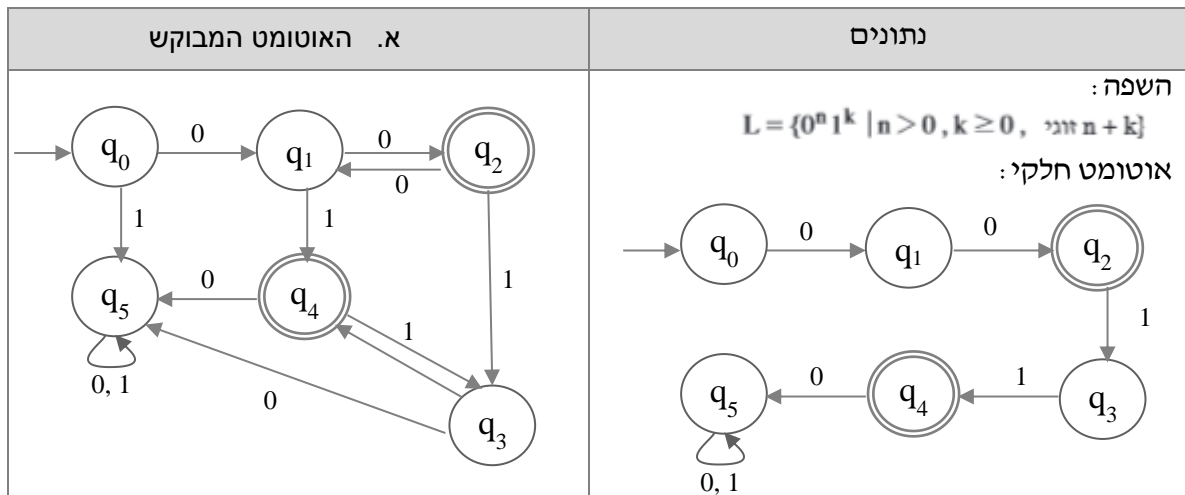


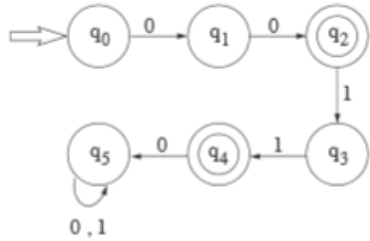
	מתקבלת על-ידי האוטומט	המילה	
	אינה מתקבלת.	000	i.
$q_0(0)q_2(1)q_3(0)q_3(0)q_3(0)q_3$	מתקבלת	01000	ii.
	אינה מתקבלת	1100	iii.
$q_0(0)q_1(0)q_1(1)q_1(0)q_4(1)q_5$	מתקבלת	00101	iv.
	אינה מתקבלת	1011	v.

פתרון שאלה 14

נושא מרכזי: אוטומט סופי דטרמיניסטי
סוג השאלה: השלמת אוטומט...

א. השלמת אוטומט





ב. נכונות/אי נכונות של טענות נתון האוטומט

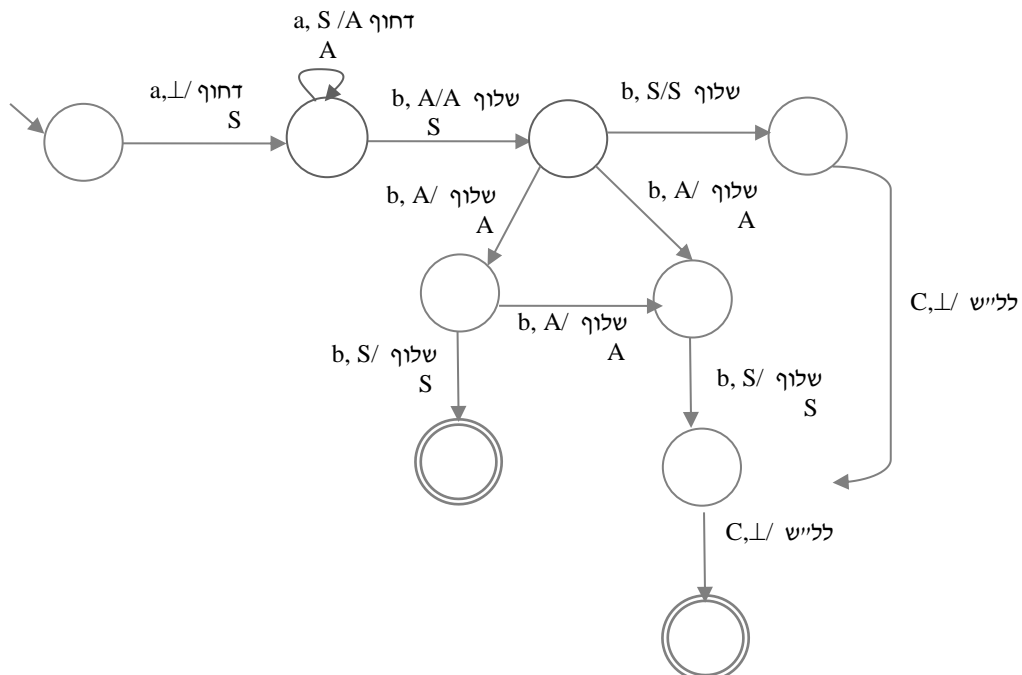
הטענה	נכון/לא נכון + נימוק
i. L היא אוסף כל המילים מעל הא"ב {a, b} שמכילות את התת-מילה ab ואינן מסתיימות ב-b.	הטענה אינה נכונה. למשל: ϵ , b, מתקבלות ואינן מכילות את התת-מילה ab.
ii. L היא אוסף כל המילים מעל הא"ב {a, b} אשר מסתיימות ב b ואינן מכילות את התת-מילה ab.	הטענה אינה נכונה. למשל: המילה ab מתקבלת ומכילה את התת-מילה ab.
iii. L היא אוסף כל המילים מעל הא"ב {a, b} אשר מסתיימות ב b או אינן מכילות את התת-מילה ab.	הטענה נכונה.

פתרון שאלה 15

נושא מרכזי: חופשיות הקשר – אוטומט מחסנית סוג השאלה: בנה אוטומט מחסנית...

אוטומט מחסנית לשפה L

L היא שפה מעל הא"ב {a, b, c}.
 $L = \{a^n b^n c^k \mid n \geq 1, 3 \leq k\}$



פתרון שאלה 16

נושא מרכזי: שפות
סוג השאלה: פעולות על מילים ושפות

א. דוגמה למילה w שעבורה מתקיים $W = ba^i$: $w_1 \cdot w \in L_1$
 $w_2 \cdot w \in L_1$

השפות L_1, L_2 מעל הא"ב $\{a, b\}$:

$$L_1 = \{v \cdot v^R \mid v \in \{a, b\}^*\}$$

$$L_2 = \{a^m b \mid m \geq 1\}$$

$\{a, b\}^*$ הוא אוסף כל המילים מעל הא"ב $\{a, b\}$ (כולל המילה הריקה),
 v^R היא ההיפוך של המילה v .

ב. דוגמה למילה w שעבורה מתקיים $W = b^i c, b^i c^i, b^{j-1} c^{j-1}$: $w_1 \cdot w \in L_1$
 $w_2 \cdot w \in L_1$

השפה L_1 מעל הא"ב $\{a, b, c\}$, והשפה L_2 מעל הא"ב $\{a\}$:

$$L_1 = \{a^n b^k c^m \mid k = \max(n, m), n \geq 1, m \geq 1\}$$

$$L_2 = \{a^s \mid s \geq 1\}$$

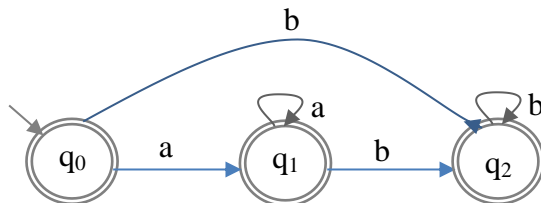
נתונות שתי מילים ב- L_2 : $w_1 = a^i, w_2 = a^j$, כך ש: $j > i$.

לפניך השפה L_1 מעל הא"ב $\{a, b\}$, והשפה L_2 מעל הא"ב $\{a, b, c\}$:

$$L_1 = \{a^n b^k \mid n \geq 0, k \geq 0\}$$

$$L_2 = \{a^m c b^r \mid m + r \geq 2\} \cup \{a^n b^n \mid n \geq 0\}$$

(1) אוטומט סופי דטרמיניסטי המקבל את השפה L_1



(2) השפה $L_1 \cap L_2$ היא: השפה $\{a^n b^n \mid n \geq 0\}$ ואינה שפה רגולרית כי יש תלות בין מספר ה-b-ים ובין מספר ה-a-ים.

(3) הסבר מדוע השפה L_2 אינה רגולרית: L_1 היא שפה רגולרית. אם L_2 הייתה רגולרית, החיתוך שלהן היה שפה רגולרית. ראינו בסעיף (2) שהחיתוך אינו רגולרי, לכן L_2 אינה רגולרית.