

מבחן בגרות 2011

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 : מחסנית, תור [מעקב]

שאלה 2 : יעילות [מעקב, מי יעיל יותר?]

שאלה 3 : מחסנית, תור [ניתוח...מימוש מחלקה ומימוש פעולות]

שאלה 4 : עצים, מחסנית [ניתוח...מימוש מחלקה ומימוש פעולות]

פרק שני - מודלים חישוביים

שאלה 13 : חופשיות הקשר – אוטומט מחסנית [בנה אוטומט מחסנית]

שאלה 14 : רגולריות פעולות על מילים ושפות [שייכות מילים לשפה, זיהוי שפה, השלמת אוטומט]

שאלה 15 : רגולריות [הוכחת אי רגולריות של שפה, חיתוך שפות]

שאלה 16 : מכונת טיורינג

פתרון פרק ראשון - עיצוב תוכנה 2011

פתרון שאלה 2

נושא מרכזי: טיפוסים, יעילות
סוג השאלה: מעקב, מי יעיל יותר?

נתונים:

פעולות sod ו-what המקבלות מערך a שאיבריו מטיפוס שלם, ממוין בסדר עולה ומספר שלם k.

מערך a הבא:

| | | | | | |
|---|---|---|---|----|----|
| a | 2 | 4 | 7 | 12 | 18 |
|---|---|---|---|----|----|

(1) טבלת מעקב אחר ביצוע הפעולה sod בעבור המערך הנתון a והמספר k=11

| k | i | j | a[i] | a[j] | if | ערך מוחזר |
|----|---|---|------|------|----|-----------|
| 11 | 0 | 1 | 2 | 4 | X | |
| | | 2 | | 7 | X | |
| | | 3 | | 12 | X | |
| | | 4 | | 18 | X | |
| | | 4 | | 2 | ✓ | true |

(2) טבלת מעקב אחר ביצוע הפעולה sod בעבור המערך הנתון a והמספר k=10

| k | i | j | a[i] | a[j] | if | ערך מוחזר |
|----|---|---|------|------|----|-----------|
| 10 | 0 | 1 | 2 | 4 | ✓ | |
| | | 2 | | 7 | X | |
| | | 3 | | 12 | X | |
| | | 4 | | 18 | X | |
| | 1 | 2 | 4 | 7 | X | |
| | | 3 | | 12 | X | |
| | | 4 | | 18 | X | |
| | 2 | 3 | 7 | 12 | X | |
| | | 4 | | 18 | X | |
| | 3 | 4 | 12 | 10 | X | false |

3) טענות כניסה ויציאה של הפעולה sod :

טענת כניסה (לא נדרש בשאלה) : הפעולה sod מקבלת מערך a ומספר שלם k.
טענת יציאה : הפעולה sod מחזירה true אם קיים במערך זוג איברים שסכומם שווה ל- k או false אחרת.

4) סיבוכיות זמן הריצה של הפעולה sod :

$O(N^2)$, כאשר N – מייצג את מספר האיברים במערך.
נימוק : עבור כל איבר במערך a נסרקים כל האיברים שאחריו במערך. $O(N)*O(N)=O(N^2)$.

5) טבלת מעקב אחר ביצוע הפעולה what בעבור המערך הנתון a והמספר k=11

| k | left | right | a[left] | a[right] | if1 | if2 | ערך מוחזר |
|----|------|-------|---------|----------|-----|-----|-----------|
| 11 | 0 | 4 | 2 | 18 | X | X | |
| | | 3 | | 12 | X | X | |
| | | 2 | | 7 | X | ✓ | |
| 1 | | | 4 | | ✓ | | |
| | | | | | | | true |

6) סיבוכיות זמן הריצה של הפעולה what :

$O(N)$, כאשר N מייצג את מספר האיברים במערך.
נימוק : הסריקה על אברי המערך מתבצעת פעם אחת. ההתקדמות על המערך היא משני צידי אך במקרה הגרוע ביותר יהיה מעבר על כל אברי המערך.

7) מי מבין הפעולות sod או what יעילה יותר?

הפעולה what יעילה יותר מהפעולה sod $O(N) < O(N^2)$, השיפור הוא שיפור בסדר גודל.

8) בהנחה ש : טענת הכניסה של הפעולות sod ו-what שונתה כך שאפשר להעביר אליהן מערך לא ממורן.

(1) טענת היציאה של הפעולה sod לא תשתנה מאחר ומתבצעת סריקה על כל הזוגות האפשריים של אברי המערך.

(2) טענת היציאה של הפעולה what תשתנה מאחר וסריקת המערך וההתקדמות של מצייני המערך מתייחסים לסדר המיון במערך, וערך הסכום של הזוג הנוכחי משפיע על עדכון המציינים.

פתרון שאלה 3

נושא מרכזי: מחסנית, תור
סוג השאלה: ניתוח... מימוש מחלקה ומימוש פעולות

א.

| | |
|--------------------------------|-------------|
| UndoQueue q = new UndoQueue(); | תור ריק |
| q.insert(1) | → 1 → |
| q.insert(2) | 2 → 1 → |
| q.insert(3) | 3 → 2 → 1 → |
| q.remove() | 3 → 2 → |
| q.insert(4) | 4 → 3 → 2 → |
| q.undo() | 3 → 2 → |
| q.undo() | 3 → 2 → 1 → |

ב.ג. גרסת פתרון א:

```
public class UndoQueue
{
    private Queue<Integer> q; // התור הנוכחי
    private Stack<Queue<Integer>> s; // מחסנית השומרת את כל התורים הקודמים
    public QueueUndo() // פעולה בונה
    {
        this.q = new Queue<Integer>();
        this.s = new Stack<Queue<Integer>>(); // מחסנית של תורים
    }
    public void insert(int x) // פעולה המכניסה העתק של התור למחסנית ומוסיפה איבר לתור הנוכחי
    {
        this.s.push(newQ());
        this.q.insert(x);
    }
    public int remove() // פעולה המכניסה העתק של התור למחסנית ומוציאה איבר מהתור הנוכחי
    {
        if (!this.q.isEmpty()) // רק אם התור אינו ריק ניתן לבצע שליפה
        {
            this.s.push(newQ());
            return this.q.remove();
        }
    }
    public void undo() // פעולה המעדכנת את התור לאחר ביטול הפעולה האחרונה
    {
        if (!this.s.isEmpty())
            this.q = this.s.pop();
    }
    private Queue<Integer> newQ() // פעולה המעתיקה את התור הנוכחי לתור חדש
    {
        Queue<Integer> nQ = new Queue<Integer>();
        Queue<Integer> tQ = new Queue<Integer>();
        while (!this.q.isEmpty())
        {
            int num = this.q.remove();
            nQ.insert(num);
            tQ.insert(num);
        }
        while (!tQ.isEmpty())
            this.q.insert(tQ.remove());
    }
}
```

```

    return nQ;
}
}

```

גרסת פתרון ב:

```

public class UndoQueue
{
    private Queue<Integer> q; // התור הנוכחי
    private Stack<Integer> s; // מחסנית השומרת את כל המספרים הקודמים
    public QueueUndo() // פעולה בונה
    {
        this.q = new Queue<Integer>();
        this.s = new Stack<Integer>(); // מחסנית של מספרים
    }
    public void insert(int x)
    {
        this.s.push(x);
        this.q.insert(x);
    }
    public int remove()
    {
        if (!this.q.isEmpty())
        {
            int num = this.q.remove();
            this.s.push(num*(-1));
        }
        return num;
    }
    public void undo()
    {
        if (!this.s.isEmpty())
        {
            int num = this.s.pop();
            if (num < 0) // שיחזור לפעולת remove
            {
                Queue<Integer> tQ = new Queue<Integer>();
                while (!this.q.isEmpty())
                    tQ.insert(this.q.remove());
                this.q.insert(num*(-1));
                while (!tQ.isEmpty())
                    this.q.insert(tQ.remove());
            }
            else // שיחזור לפעולת insert
            {
                Stack<Integer> tS1 = new Stack<Integer>();
                Stack<Integer> tS2 = new Stack<Integer>();
                while (!this.q.isEmpty()) // העברת אברי התור למחסנית
                    tS1.push(this.q.remove());
                tS1.pop(); // הוצאת האיבר שהוכנס אחרון
                while (!tS1.isEmpty()) // העברת האיברים למחסנית נוספת כדי להפוך בחזרה את סדר האיברים
                    tS2.push(tS1.pop());
                while (!tS2.isEmpty()) // החזרת כל שאר אברי המחסנית לתור
                    this.q.insert(tS2.pop());
            }
        }
    }
}
}

```

פתרון שאלה 4

נושא מרכזי: עצים (עץ בינארי), מחסנית
סוג השאלה: ניתוח... מימוש מחלקה ומימוש פעולות

נתוני השאלה:

כותרת של פעולה: `public static void leaves (BinTreeNode<Integer> t, Stack<Integer> s)`
הפעולה מקבלת עץ בינארי לא ריק `t` של מספרים שלמים, ומחסנית ריקה `s` של מספרים שלמים.
הפעולה מכניסה למחסנית את ערכי כל העלים של העץ `t`, על-פי סדר סריקה מימין לשמאל.

א. מימוש הפעולה

```
public static void leaves (BinTreeNode<Integer> t, Stack<Integer> s)
{
    if (t!=null)
    {
        if (t.getLeft() == null && t.getRight() == null)
            s.push(t.getInfo());
        leaves(t.getRight(),s);
        leaves(t.getLeft(),s);
    }
}
```

ב. פעולה בוליאנית

טענת כניסה: הפעולה מקבלת 2 עצים בינאריים לא ריקים של מספרים שלמים.
טענת יציאה: הפעולה מחזירה `true` אם מתקיימים שני התנאים הבאים:
- יש להם אותו מספר עלים
- ערכי העלים שווים, על-פי סדר הסריקה מימין לשמאל.
אחרת, הפעולה מחזירה `false`.

```
public static boolean sameLeaves(BinTreeNode<Integer> t1, BinTreeNode<Integer> t2)
{
    Stack<Integer>s1 = new Stack<Integer>();
    Stack<Integer>s2 = new Stack<Integer>();
    leaves(t1,s1);
    leaves(t2,s2);
    while (!s1.isEmpty() && !s2.isEmpty())
    {
        if (s1.pop()!=s2.pop())
            return false;
    }
    if (!s1.isEmpty() || !s2.isEmpty())
        return false;
    return true;
}
```

פתרון סרק לני - מ/א/מ'ס הי"לוב"מ 2011

פתרון שאלה 13

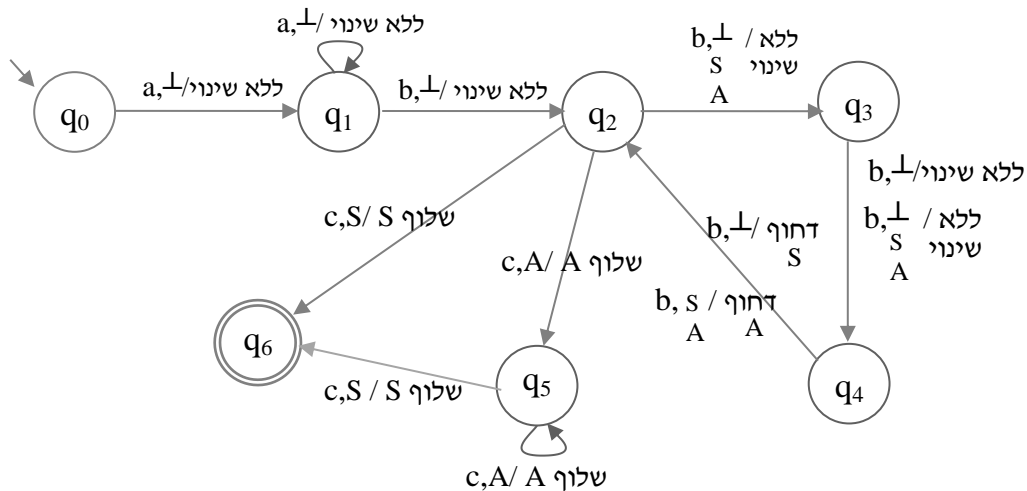
נושא מרכזי: חופשיות הקשר – אוטומט מחסנית
סוג השאלה: בנה אוטומט מחסנית...

נתונים:

השפה L מעל הא"ב $\{a, b, c\}$: $L = \{a^n b^{3k+1} \mid n > 0, k > 0\}$

א. המילה הקצרה ביותר בשפה L : $a^n b^{3k+1} c^k \mid n > 0, k > 0$

ב. אוטומט מחסנית שמקבל את השפה L :



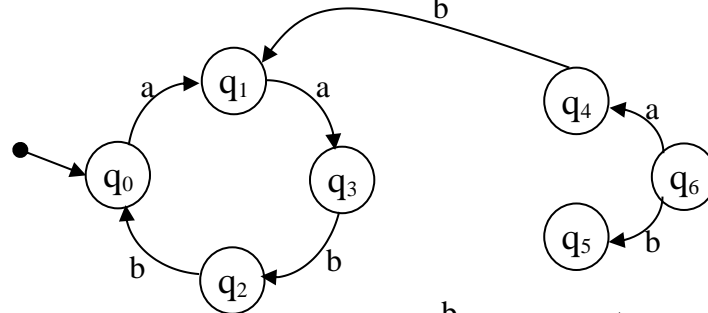
פתרון שאלה 14

נושא מרכזי: פעולות על מילים ושפות
סוג השאלה: שייכות מילים לשפה, זיהוי שפה, השלמת אוטומט

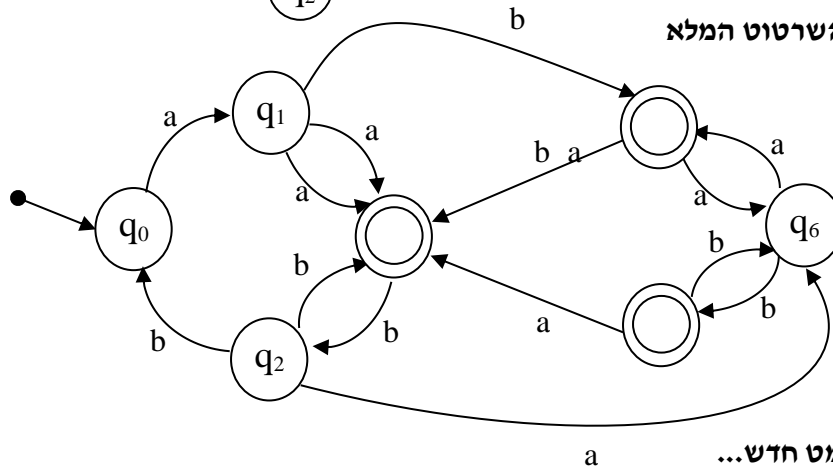
א. נתונה השפה L מעל הא"ב $\{a, b\}$:

$L = \{ w \mid |w| > 0, w \text{ מופיע בכל המילה מספר זוגי של פעמים, } \}$
 לדוגמה: המילה $abaabbab$ שייכת לשפה. המילה $abbabaa$ שייכת לשפה.
 המילה $babbaba$ לא שייכת לשפה.

נתון שרטוט חלקי (חסרים מעברים וסימני קלט) של אוטומט סופי דטרמיניסטי המקבל את השפה L .



פתרון: השרטוט המלא

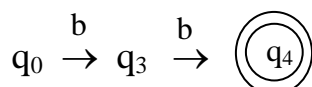
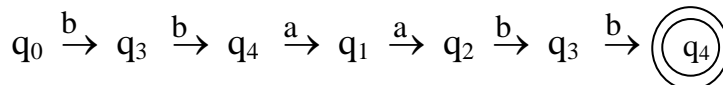


ב. נתון אוטומט חדש...

1) קביעה האם המילה מתקבלת מהאוטומט:

i. המילה $aaba$ – לא מתקבלת מהאוטומט

ii. המילה $bbaabb$ – מתקבלת מהאוטומט



2) השפה L המוגדרת על-ידי האוטומט: שפת כל המילים מעל $\{a, b\}$ המסתיימות ברצף אותיות זהות שאורכו גדול מ-1.



א. נתונה השפה $L = \{ 0^n 1^k 2 \mid n > k \geq 0 \}$ $\{0, 1, 2\}$ מעל הא"ב

הוכחה כי השפה $L = \{ 0^n 1^k 2 \mid n > k \geq 0 \}$ היא אי-רגולרית.

גרסת פתרון 1:

ההוכחה נעשית בדרך השלילה.

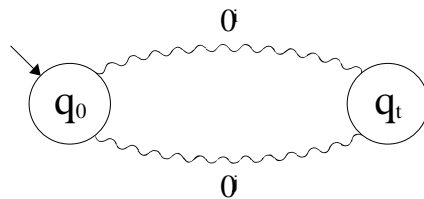
(1) נניח שהשפה L רגולרית ולכן קיים אוטומט סופי A המקבל אותה. נחלק את המילים השייכות לשפה לשני חלקים.

(2) נגדיר את קבוצת תת-המילים שהן התחלות של מילים בשפה: $W = \{ 0, 0^2, 0^3, \dots, 0^m \}$. קבוצה זו היא קבוצה אינסופית עבור כל m וניתנת לרישום כ: $W = \{ 0^m \mid m > 0 \}$.

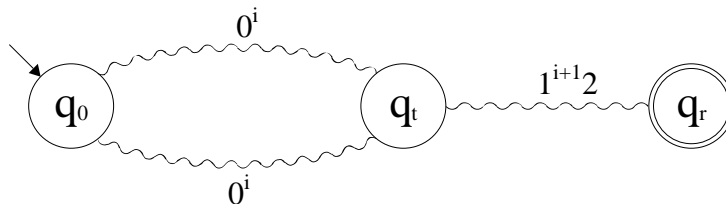
(3) נבחר שתי מילים כלליות כלשהן מתוך W שהן התחלות של מילים מהשפה L , למשל: $w_i = 0^i, w_j = 0^j$ כך ש- $i > j$.

(4) מאחר והאוטומט הוא אוטומט סופי, חייבות להיות מילים שהתחלות שלהן מגיעות לאותו מצב, אחרת, כל תת מילה תיצור מסלול חדש באוטומט ומאחר והשפה אינסופית, גם האוטומט יהיה אינסופי. מכיוון שהמילים שבחרנו הן כלליות, ניתן להניח בלי הגבלת הכלליות כי הן מגיעות למצב משותף q_t באוטומט A .

(5) קיבלנו מסלול מ- q_0 עבור תת-המילה w_i המגיע למצב q_t , ומסלול המגיע מ- q_0 עבור תת-המילה w_j גם כן למצב q_t , כמתואר בשרטוט:



מאחר והמילה $w = a^i b^j$ בשפה, אזי יש מעבר ממצב q_t למצב מקבל q_f היוצר למילה $0^i 1^j 2$ מסלול מקבל כמתואר בגרף האוטומט A :

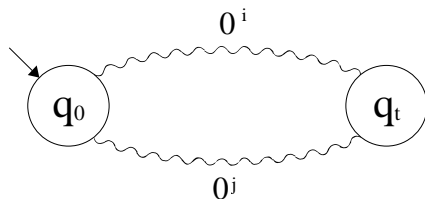


(6) מכאן מתקבל מסלול מקבל מ- q_0 ל- q_t גם עבור $w = 0^i 1^j 2$ כאשר $i > j$, וזה בניגוד לכללי השפה.

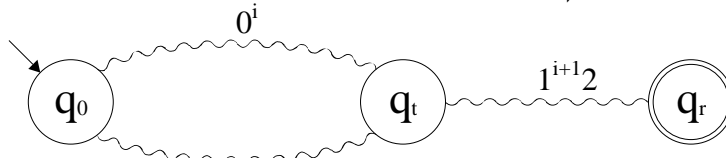
(7) קיבלנו תוצאה שגויה מאחר ו- $0^i 1^j 2 \notin L$. כלומר, ההנחה שיש בשפה שתי מילים המגיעות לאותו מצב שגויה ולכן למעשה, כל מילה מגיעה למצב אחר. מאחר ובשפה יש אינסוף מילים, נדרשים אינסוף מצבים וזוהי סתירה להגדרת אוטומט סופי. סתרנו את ההנחה ש- L שפה רגולרית ולכן לא קיים אוטומט A המקבל אותה, ועל-כן השפה אינה רגולרית.

גרסת פתרון 2: ההוכחה נעשית בדרך השלילה.

- (1) נניח שהשפה L רגולרית ולכן קיים אוטומט סופי A המקבל אותה. נחלק את המילים השייכות לשפה לשני חלקים.
- (2) בלי הגבלת הכלליות השפה $0^{i+1}1^j$ מייצגת תת-קבוצה של מילים בשפה L , ומספיק להראות שהאוטומט לא יכול לזהות מילים של השפה הזו.
- (3) (בדוגמה זו $W = \{0, 0^2, 0^3, \dots, 0^m\}$). מאחר ו- W היא קבוצה אינסופית, די בה כדי לסתור את ההנחה בהמשך.
- (4) נבחר שתי מילים כלליות כלשהן מתוך W שהן התחלות של מילים מהשפה L , למשל: $w_i = 0^i, w_j = 0^j, i \neq j$.
- (5) מאחר והאוטומט הוא אוטומט סופי, חייבות להיות מילים שההתחלות שלהן מגיעות לאותו מצב, אחרת, כל תת מילה תיצור מסלול חדש באוטומט ומאחר והשפה אינסופית, גם האוטומט יהיה אינסופי. מכיוון שהמילים שבחרנו הן כלליות, ניתן להניח בלי הגבלת הכלליות כי הן מגיעות למצב משותף q_t באוטומט A .
- (6) קיבלנו מסלול מ- q_0 עבור תת-המילה w_i המגיע למצב q_t , ומסלול המגיע מ- q_0 עבור תת-המילה w_j גם כן למצב q_t , כמתואר בשרטוט:



- (7) מאחר והמילה $w = 0^i1^{i+1}2$ בשפה, אזי יש מעבר ממצב q_t למצב מקבל q_f היוצר למילה $0^i1^{i+1}2$ מסלול מקבל כמתואר בגרף האוטומט A :



- (7) מכאן מתקבל מסלול מקבל מ- q_0 ל- q_t גם עבור $w = 0^j1^{i+1}2$ כאשר $i \neq j$, וזה בניגוד לכללי השפה.
- (8) קיבלנו תוצאה שגויה מאחר ו- $0^j1^{i+1}2 \notin L$. כלומר, ההנחה שיש בשפה שתי מילים המגיעות לאותו מצב שגויה ולכן למעשה, כל מילה מגיעה למצב אחר. מאחר ובשפה יש אינסוף מילים, נדרשים אינסוף מצבים וזוהי סתירה להגדרת אוטומט סופי. סתרנו את ההנחה ש- L שפה רגולרית ולכן לא קיים אוטומט A המקבל אותה, ועל-כן השפה אינה רגולרית.

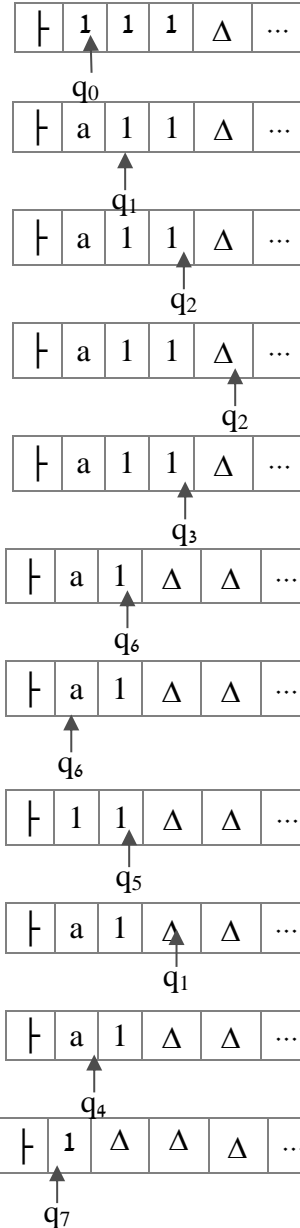
ב. נתונה השפה L_1 מעל הא"ב $\{0, 1\}$ $L_1 = \{1^n \mid n \geq 0\}$

השפה $L_1 \cap L$ היא: $L_1 \cap L = \emptyset$ מכיוון שמילה בשפת L חייבת להתחיל ב-0 ואילו מילה בשפה L_1 לא יכולה להתחיל ב-0.

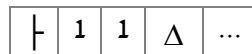
פתרון שאלה 16

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

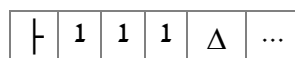
א. לאחר חישוב $f(3)$ הסרט יכיל:



ב. לאחר חישוב $f(5)$ הסרט יכיל:



ג. לאחר חישוב $f(6)$ הסרט יכיל:



ד. הפונקציה $f(x)$ שהמכונה מחשבת היא: $f(x) = (\text{int})x/2$

ה. מעבר מ q_0 ל- q_7 : ימין, Δ/Δ