

מבחן 2008

מבחן בגרות 2008

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 : רקורסיה [מעקב]

שאלה 2 : טיפוס מורכב, רשימות [הגדרת טיפוסים, פיתוח פעולה]

שאלה 3 : טיפוסים [הגדרת טיפוסים, פיתוח פעולה]

שאלה 4 : מחסנית, יעילות (סיבוכיות) [פיתוח פעולה]

פרק שני - מודלים חישוביים

שאלה 13 : רגולריות - אוטומט סופי דטרמיניסטי [בניית מודל]

שאלה 14 : פעולות על מילים ושפות [שיוך מילים והגדרת שפות, קביעת נכונות של טענות]

שאלה 15 : אוטומט מחסנית [בניית מודל]

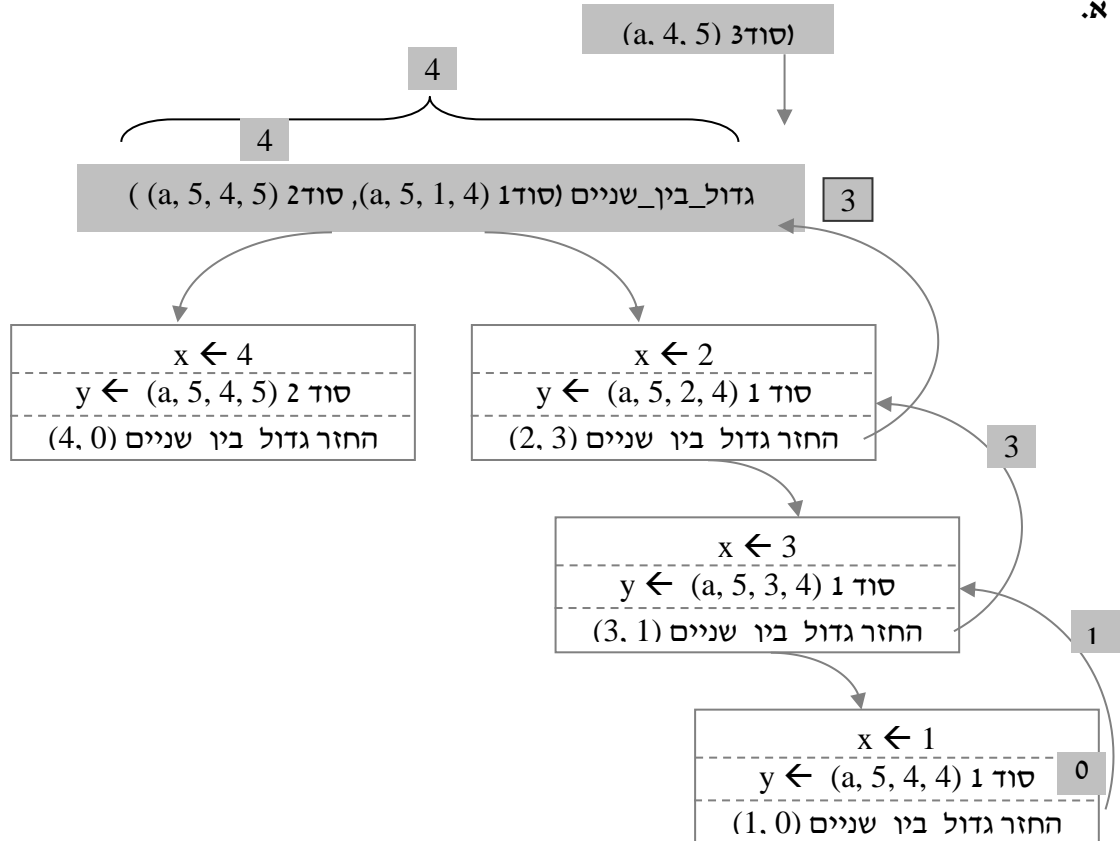
שאלה 16 : שפות רגולריות - אוטומט סופי דטרמיניסטי [בניית מודל]

פתרון סרק ראשון - עיצוב תוכנה 2008

פתרון שאלה 1

נושא מרכזי: רקורסיה
סוג השאלה: מעקב

א.



ב. האלגוריתם סוד3 מחזיר את ההפרש הגדול ביותר בערכו המוחלט בין האיבר במקום ה-p ובין שאר אברי המערך a.

ג. טבלת מעקב לזימון סוד4(a,5):

i	n	i ≤ n	t	t > k	k	return
					sod3(a,1,5)=6	
2	5	true	sod3(a,2,5)=7	true	7	
3	5	true	sod3(a,3,5)=4	false	7	
4	5	true	sod3(a,2,5)=4	false	7	
5	5	true	sod3(a,5,5)=7	false	7	7

הזימון סוד4(a,5) יחזיר 7.

ד. האלגוריתם סוד4 מחזיר את ההפרש הגדול ביותר בערכו המוחלט במערך a בין כל זוגות האיברים.

פתרון שאלה 2

נושאים מרכזיים: טיפוס מורכב, רשימות
סוג השאלה: הגדרת טיפוסים, פיתוח פעולה

כותרת המחלקה מיון בתחומים - SortByRange והתכונות שלה:

```
public class SortByRange // מיון בתחומים
{
    private int[] mins; // מערך הערכים המינימאליים של כל תחום
    private List<Integer>[] values; // מערך רשימות המספרים בכל תחום
    private int current; // מספר התחומים הנוכחי
```

פעולה פנימית insert בתוך המחלקה SortByRange

```
public void insert(int num)
{ // הפעולה מקבלת מספר שלם ומכניסה אותו למקומו המתאים לפי סדר המיון
    int listIndex = 0;
    while (listIndex < current - 1 && num > this.mins[listIndex + 1])
    {
        listIndex++; // לולאה למציאת אינדקס התחום
    }
    List<Integer> lst = this.values[listIndex]; // הרשימה אליה צריך num להיכנס
    insertIntoSortedList(lst, num); // פעולה המכניסה מספר לרשימה ממוינת
}
```

פעולה פנימית addRange בתוך המחלקה SortByRange

```
public static void insertIntoSortedList(List<Integer> lst, int num)
```

```
{ // הפעולה מקבלת רשימה ממוינת של מספרים שלמים ומספר נוסף
// הפעולה מוסיפה את המספר למקום המתאים ברשימה כך שתישאר ממוינת
    Node<Integer> prev = null; // שומר על המקום הקודם ל- pos
    Node<Integer> pos = lst.getFirst();
    while (pos != null && pos.getInfo() < num)
```



מאחר והפעולה insert מכניסה את האיבר החדש אחרי מיקומו של pos, יש לשמור את המקום הקודם ל-pos. הלולאה תתבצע כל עוד המספר גדול מערך הרשימה הנוכחי וגם הרשימה לא הסתיימה.

```
{
    prev = pos;
    pos = pos.getNext();
}
lst.insert(prev, num);
}
```

```
public void addRange(int r)
```

```
{ // הפעולה מקבלת מספר שלם הגדול מכל המספרים במערך values
// הפעולה תוסיף תחום חדש שהערך המינימאלי שלו הוא r
    this.mins[this.current] = r; // הוספת התחום למקום הראשון הפנוי במערך
    this.values[this.current] = new List<Integer>(); // בניית רשימה חדשה השייכת לתחום
    this.current++; // הגדלת מספר התחומים ב-1
}
```

סיבוכיות הפעולה insert היא $O(n)$ כאשר n מייצג את מספר האיברים ברשימה אליה מוכנס הערך החדש. נימוק: סיבוכיות הלולאה היא $O(1)$ מכיוון שהיא סורקת את המערך שמספר האיברים שלו בפועל הוא לכל היותר 100. בפעולה insert יש זימון לפעולה insertIntoSortedList. סיבוכיות

הפעולה insertIntoSortedList היא $O(n)$ מכיוון שהיא סורקת את הרשימה הממוינת כדי לאתר את המקום בו צריך להכניס את הערך החדש. סיבוכיות הפעולה addRange היא $O(1)$ מכיוון שזמן הביצוע שלה קבוע והוא אינו תלוי באורך הקלט.

פתרון שאלה 3

נושא מרכזי: טיפוסים
סוג השאלה: הגדרת טיפוס, פיתוח פעולה

ייצוג למשחק: א

```
public class Game // משחק
{
    private int code; // קוד המשחק
    private String name; // שם המשחק
    private int minAge; // גיל מינימלי
    private int soldCopies; // מספר העותקים שנמכרו
    private int numPlayed; // מספר הפעמים ששיחקו במשחק
    private int stopError; // מספר הפעמים שהייתה תקלה
```

ב

```
public class GamesData // מאגר מידע על משחקים
{
    private Game[] games; // מערך משחקים במאגר
    public void updateStop(int code)
    public int soldMoreThanK(int k)
```

ייצוג למאגר מידע על משחקים: ב

```
i. public int numOfStars()
{
    double rate = (double)this.stopError / this.numPlayed;
    if (rate < 0.05) return 4;
    if (rate < 0.25) return 3;
    if (rate < 0.5) return 2;
    return 1;
}
```



הפעולה מוגדרת במחלקה GamesData. הפעולה מקבלת קוד של משחק במאגר המידע ומוסיפה 1 למספר הפעמים שהמשחק נפסק בגלל תקלה. ii. כותרת הפעולה:

ג

```
ii. public List<int> gameCodes(int minAge)
{
    List<int> lst = new List<int>();
    Game game;
    int i = 0;
    while (i < this.games.length && this.games[i] != null)
    {
        game = this.games[i];
        if (game.getMinAge() >= minAge && game.numOfStars() == 4)
            lst.insert (null, game.getCode());
        i++;
    }
    return lst;
}
```



הפעולה מוגדרת במחלקה GamesData. הפעולה מקבלת מספר טבעי k ומחזירה את מספר המשחקים במאגר שנמכרו ביותר מ-k עותקים.

פתרון שאלה 4

נושא מרכזי: מחסנית, יעילות (סיבוכיות)
סוג השאלה: פיתוח פעולה

פעולה חיצונית שמקבלת מחסנית לא ריקה של מספרים שלמים...ומכניסה אחרי כל איבר מחליף כיוון, איבר זהה לו:

```
public static void changeDirection(Stack< Integer > s)
{ // הפעולה מקבלת מחסנית עם מספרים שלמים ללא מספרים זהים צמודים.
  // הפעולה מכניסה אחרי כל איבר מחליף כיוון, איבר זהה לו
  Stack< Integer > temp = new Stack< Integer >(); // מחסנית עזר
  boolean up = true; // direction: up - true, down - false
  int one = 0, two = 0;
  if (!s.isEmpty())
  {
    one = s.pop();
    temp.push(one);
  }
  if (!s.isEmpty())
  {
    two = s.pop();
    temp.push(two);
    up = (two > one); // קבע את כיוון המחסנית
  }
  while (!s.isEmpty())
  {
    one = two;
    two = s.pop();
    if (up != (two > one)) // one שינוי כיוון באיבר
    {
      temp.push(one); // דחוף איבר מחליף כיוון נוסף
      up = !up; // שנה את הכיוון מלמעלה למטה או ההיפך
    }
    temp.push(two);
  }
  while (!temp.isEmpty())
  {
    s.push(temp.pop());
  }
}
```

סיבוכיות זמן הריצה היא $O(n)$ כאשר n מייצג את מספר האיברים במחסנית. מתבצעת העברה של כל אברי המחסנית למחסנית עזר ולאחר מכן החזרתם למחסנית המקורית. סיבוכיות כל פעולות המחסנית היא $O(1)$.

פתרון סדרק שני - מוליטים היילובים 2008

פתרון שאלה 13

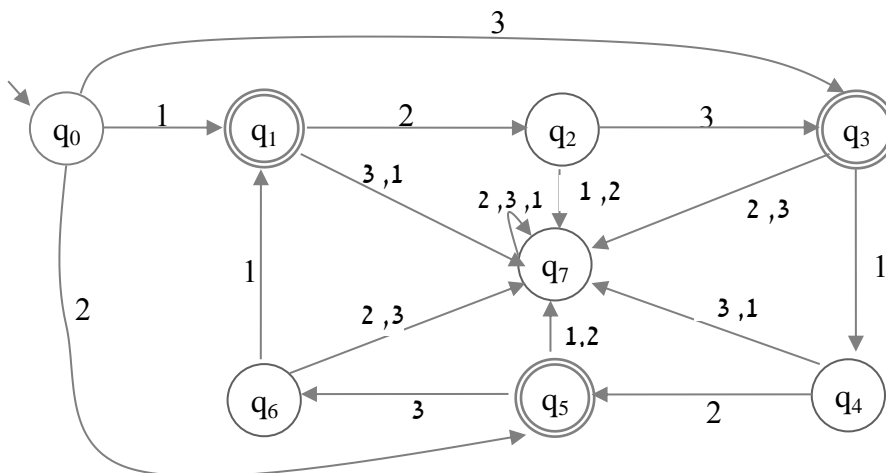
מספר פתרונות



נושא מרכזי: רגולריות - אוטומט סופי דטרמיניסטי
סוג השאלה: בניית מודל

נתונה מחרוזת אינסופית: ...3123123123

אוטומט דטרמיניסטי סופי מעל הא"ב $\{1, 2, 3\}$, המקבל את כל המילים באורך אי-זוגי שכל אחת מהן היא תת-מחרוזת של המחרוזת הנתונה.



מספר מצב	תיאור המצב
0	מצב התחלתי
1	התו האחרון היה 1, רצף באורך אי זוגי
2	התו האחרון היה 2, רצף באורך זוגי
3	התו האחרון היה 3, רצף באורך אי זוגי
4	התו האחרון היה 1, רצף באורך זוגי
5	התו האחרון היה 2, רצף באורך אי זוגי
6	התו האחרון היה 3, רצף באורך זוגי
7	מצב מלכודת

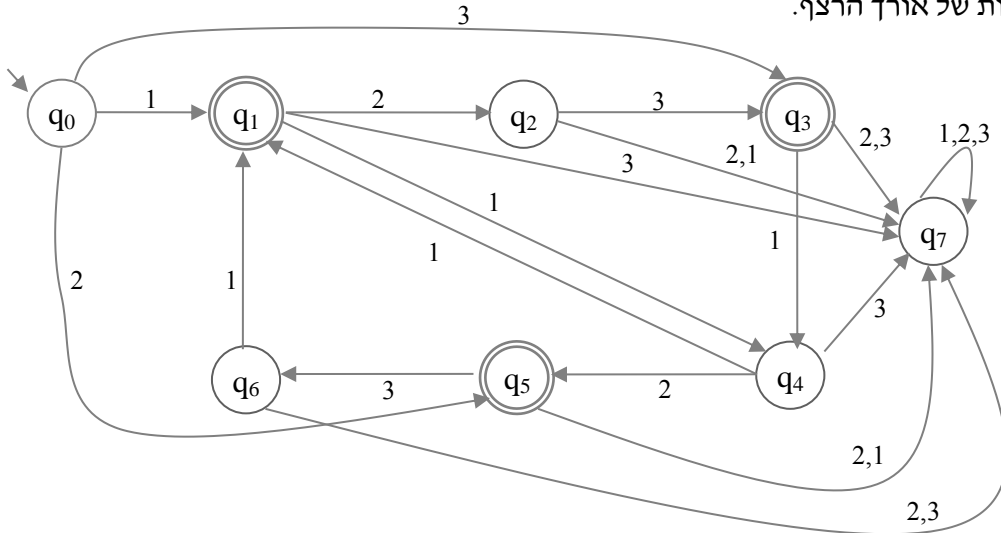
הערה: ההנחיה בשאלה היא לבנות אוטומט סופי דטרמיניסטי.

הסבר הפתרון: כל תת מחרוזת של המחרוזת הנתונה צריכה לשמר על סדר הספרות המבוקש ועל-כן יכולה להתחיל ב-1, 2 או 3 ובלבד שתשמור על סדר הספרות. בנוסף, יש לוודא שאורך המילה אי-זוגי ולכן יש צורך להפריד עבור כל רצף שמקיים את הסדר, אורך ברצף אי-זוגי או זוגי. מאחר ויש 3 ספרות שונות, ועבור כל רצף שמסתיים באחת האותיות צריך לזכור אורך זוגי או אי-זוגי יש 6 מצבים (3×2) למעט המצב ההתחלתי שבמקרה זה זוכר שהרצף עדיין לא התחיל.



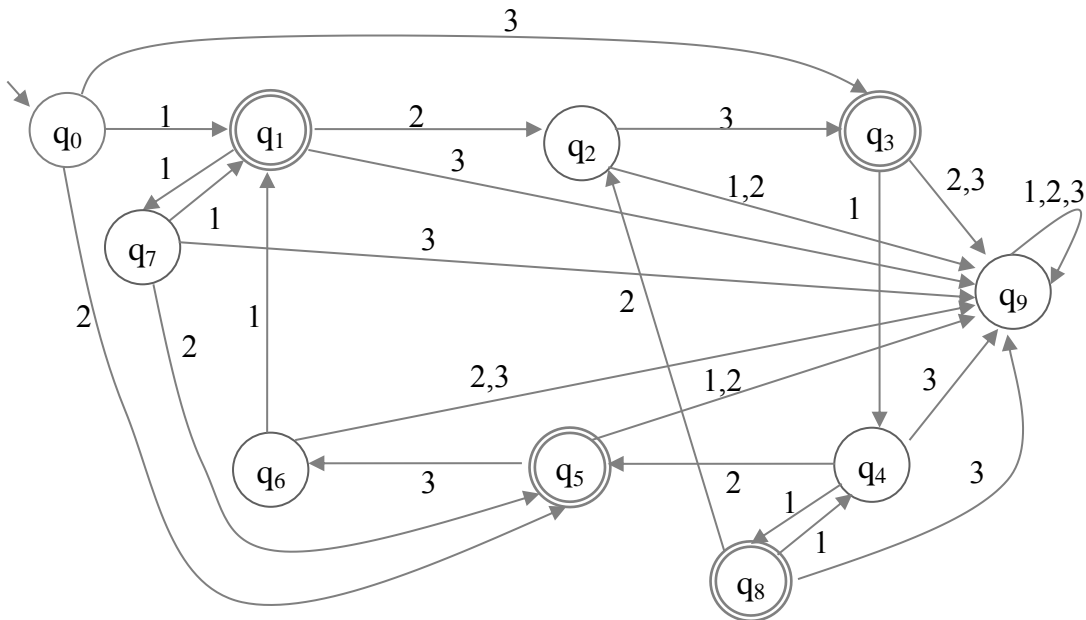
אוטומט דטרמיניסטי סופי מעל הא"ב $\{1, 2, 3\}$, המקבל את כל המילים שהן באורך אי-זוגי וסדר הספרות בהן הוא כמו הסדר במחרוזת הנתונה, אך הספרה 1 יכולה לחזור כמה פעמים ברצף.

פתרון א': האוטומט זהה בבסיסו לאוטומט של סעיף א' בתוספת "נדנדה" של רצף 1 על-מנת לבדוק את הזוגיות של אורך הרצף.



טבלת המצבים זהה לאוטומט בסעיף א. לא התווספו מצבים מכיוון שניצלנו ב"נדנדה" את שני המצבים q_1, q_4 הזוכרים רצף או-זוגי וזוגי של 1.

פתרון ב': פתרון זה מבצע את בקרת הזוגיות של רצף 1 על-ידי הוספה של שני מצבים.



על-פי טבלת המצבים של האוטומט ניתן לראות שלזוגות המצבים q_1, q_7 ו- q_4, q_8 אותו תפקיד ולכן ניתן לאחד ביניהם. בבחינת הבגרות אין צורך לבנות את האוטומט המצומצם ביותר ולכן זו בהחלט תשובה נכונה.

טבלת מצבים



מספר מצב	תיאור המצב
0	מצב התחלתי
1	התו האחרון היה 1, רצף באורך אי זוגי
2	התו האחרון היה 2, רצף באורך זוגי
3	התו האחרון היה 3, רצף באורך אי זוגי
4	התו האחרון היה 1, רצף באורך זוגי
5	התו האחרון היה 2, רצף באורך אי זוגי
6	התו האחרון היה 3, רצף באורך זוגי
7	התו האחרון היה 1, רצף באורך זוגי
8	התו האחרון היה 1, רצף באורך אי זוגי
9	מצב מלכודת

פתרון שאלה 14

נושא מרכזי: פעולות על מילים ושפות
סוג השאלה שיוך מילים והגדרת שפות, קביעת נכונות של טענות

א

i. מילה השייכת לשפה L_4 ולא שייכת לשפה L_3
 $w=1100$ שייכת ל- L_4 כי מספר האפסים שווה למספר האחדים. לא שייכת ל- L_3 כי מספר זה שונה מ-5. כל מילה בעלת מספר שווה של אפסים ואחדים תתאים כל עוד מספר זה יהיה שונה מ-5.

ii. מילה השייכת לשפה L_1 ולא שייכת לשפה L_2
 $w=1111100$ שייכת ל- L_1 כי אורכה גדול מ-5. לא שייכת ל- L_2 כי מספר האחדים גדול מ-5. כאן תתאים כל מילה בעלת אורך הגדול מ-5 אך עם מספר אחדים הגדול או שווה ל-5.

iii. מילה השייכת לשפה L_5 ולא שייכת לשפה L_2
 $X=1101$
 $Y=0$
 משרשור xyx מתקבלת המילה $W=101110110$ השייכת ל- L_5 ולא שייכת ל- L_2 מכיון שהמספר האחדים גדול מ-5. כל מילה במבנה xyx עם מספר אחדים גדול או שווה ל-5 תתאים.

ב

i. הגדרת השפה $\overline{L_2}$
 המשלים ל- L_2 תהיה שפה בה לכל המילים יהיה לפחות 5 אחדים.
 $\overline{L_2} = \{w \mid \#_1(w) \geq 5\}$

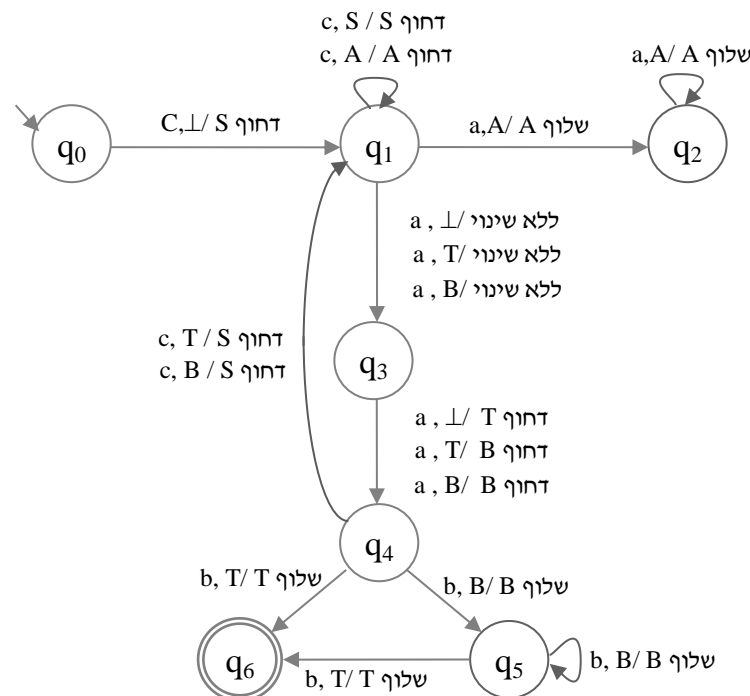
ii. הגדרת השפה $\overline{L_3}$
 המשלים ל- L_3 תהיה שפה בה לכל המילה יהיה מספר השונה מ-5 של אחדים או אפסים. מספיק ש אחת התכונות תתקיים כדי שהשפה תשלים את L_3 .

$$\overline{L_3} = \{w \mid \#_1(w) \neq 5 \text{ or } \#_0(w) \neq 5\}$$

- i. **הטענה אינה נכונה** כי יש אינסוף מילים שאורכן גדול מ-5 ומספר האחדים שלהם קטן מ-5. לדוגמה: 111100. מכיוון שיש מילה המתאימה לשתי השפות החיתוך בנייהן אינו שפה ריקה.
- ii. **הטענה אינה נכונה**. הטענה הייתה נכונה לו היה מדובר על L_3 ו- L_4 . מכיוון שמדובר על $\overline{L_3}$ ו- $\overline{L_4}$ נוכיח בכך שנראה שישנן מילים השייכות ל- $\overline{L_3}$ אך לא שייכות ל- $\overline{L_4}$ (כלומר, שייכות ל- L_4). המילים המתאימות הן מילים שמספר האחדים או מספר האפסים שלהן שונה מ-5 ($\overline{L_3}$) ומספר האחדים והאפסים שלהם שווה (על מנת שיהיו ב- L_4 ולא ב- $\overline{L_4}$). לדוגמה: 00110011. מכיוון שיש מילה השייכות ל- $\overline{L_3}$ אך לא שייכות ל- $\overline{L_4}$ הטענה אינה נכונה.
- iii. **הטענה אינה נכונה**, כאשר נרשר מילה w_1 הנמצאת ב- L_4 המקיימת $\#_0(w_1)=\#_1(w_1)$ עם מילה w_2 הנמצאת ב- L_4 המקיימת $\#_0(w_2)=\#_1(w_2)$ תתקבל מילה w_3 אשר בהכרח תקיים: $\#_0(w_3)=\#_0(w_1)+\#_0(w_2)$ ו- $\#_1(w_3)=\#_1(w_1)+\#_1(w_2)$. בגלל ההגבלות על w_1 ו- w_2 בהכרח מתקיים $\#_0(w_3)=\#_1(w_3)$ לכן w_3 שייכת ל- L_4 , לכן מתקיים $L_4 \cdot L_4 = L_4$.
- iv. **הטענה אינה נכונה**, על מנת להפריך אותה נראה שיש מילה הנמצאת גם ב- L_3 וגם ב- L_5 .
 $Y=01$ $X=0011$
 מרשור xy מתקבלת $w=001100101$ אשר מקיימת את מבנה L_5 ואת התכונה ב- L_3 בה מספר האחדים והאפסים חייב להיות 5. כל X בו שני אפסים ושני אחדים ו- Y עם אפס ואחד יתאימו, או X עם אפס ואחד ו- Y עם שלושה אפסים ושלושה אחדים.

פתרון שאלה 15

נושא מרכזי: אוטומט מחסנית
 סוג השאלה: בניית מודל

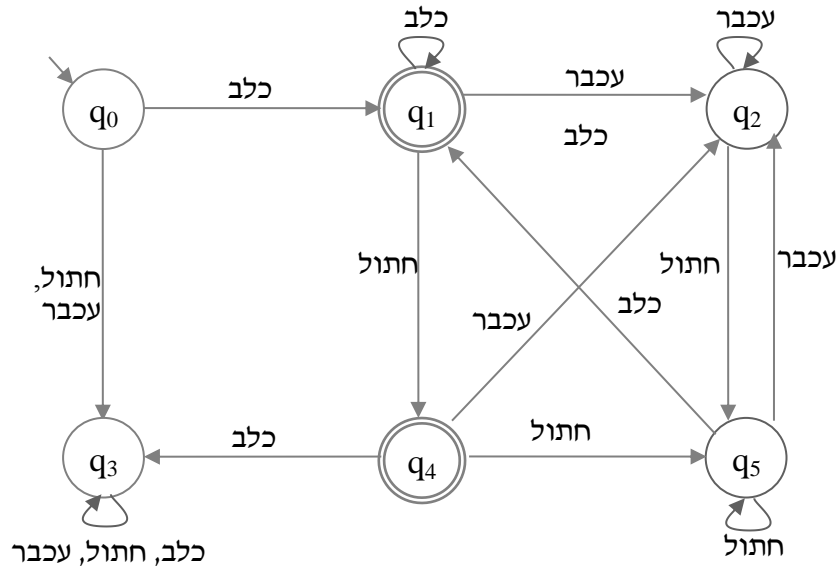


פתרון שאלה 16

נושא מרכזי: שפות רגולריות - אוטומט סופי דטרמיניסטי
סוג השאלה: בניית מודל

אוטומט סופי דטרמיניסטי שמקבל את כל התהלוכות החוקיות:

- חתול אף פעם לא יעמוד לבדו בין שני כלבים.
- בראש התהלוכה תמיד יהיה כלב.
- עכבר לא יכול להיות אחרון בתהלוכה.



מספר מצב	תיאור המצב
0	מצב התחלתי
1	החיה האחרונה הייתה כלב
2	החיה האחרונה הייתה עכבר
3	מצב מלכודת. מגיעים אליו כאשר התהלוכה התחילה בחיה שאינה כלב או שחתול עמד בין שני כלבים.
4	רצף של כלב וחתול
5	החיה האחרונה הייתה חתול

נתון אוטומט סופי דטרמיניסטי המתאר את הכללים שקבע האריה.

הכללים שהאוטומט מתאר:

- תהלוכה חוקית היא תהלוכה אשר לא יתקיים בה רצף של חתול, עכבר או עכבר, חתול.
- התהלוכה יכולה להיות ריקה.