ממן 11

שאלה 1 :

0-> R

1->R

0

0

0

0

**נתאר מכונת טיורינג המכריעה את PAL**

רעיון האלגוריתם:

הסבר:

המכונה בודקת האם התו הראשון והתו האחרון שווים. בכל סיבוב היא שומרת מה היה התו הראשון (0 או 1) ורושמת במקומו רווח. אחר כך היא רצה עד סוף הסרט ובודקת מה היה התו האחרון. אם התו האחרון שונה מהתו הראשון, המכונה עובדת למצב דוחה. אחרת, היא רושמת רווח וחוזרת להתחלה לסיבוב נוסף. אם בסיבוב כלשהו אין יותר תווים ונשארו רק רווחים. המכונה עובדת למצב מקבל. המכונה מכריעה את PAL כי בהכרח לא יכול לקרות מצב של לולאה אינסופית. כל מילת קלט מתקבלת או נדחית.

**State Diagram**

1. תחילה בודקת האם ריק אם ריק תילך שמאלה
2. בעבור קלט 0 תילך ימינה או שמאלה מהצד אשר התחלנו
3. באותה צורה אם התקבל קודם 1 תילך ימינה או שמאלה
4. תרשום רווח בעבור השמירה
5. המשך ריצה עד סוף הקלט
6. אם התו האחרון שונה מהתו הראשון עבור למצב דחייה
7. אחרת רשום רווח
8. חזור לסיבוב נוסף
9. כאשר נותרו רק רווחים
10. עבור למצב מקבל

שאלה 2

א. עבור מילים מהצורה w#w כאשר |w| = n, מספר הפעמים שהמכונה תקרא את הסימן # הוא 2n + 1. הסיבה היא שהמכונה בודקת כל סיבית במילה w לפני ואחרי ה-#. כלומר לכל 1 ≤ i ≤ n המכונה משווה את הסיבית ה-i של המילה w שלפני ה-#, לסיבית ה-i של המילה w אחרי ה-#, ואז מוחקת את הסיבית הזאת משתי המילים (רושמת x במקום הסיבית). לכן המכונה עוברת 2n פעמים מעל התו #, ועוד פעם נוספת כאשר שתי המילים כבר מחוקות.

ב. המכונה החדשה תקרא כל פעם סיבית ממילה אחרת. כלומר, הסיבית הראשונה תיקרא מהמילה השמאלית, ואז תושווה לסיבית הראשונה במילה הימנית. הסיבית השניה תיקרא מהמילה הימנית ואז תושווה לסיבית השניה במילה השמאלית. הסיבית השלישית תיקרא מהמילה הראשונה ותושווה וסיבית השלישית במילה הימנית, וכן האלה. כלומר, לכל סיבית המכונה עוברת פעם אחת על סרט הקלט. לכן מספר הפעמים שהמכונה תעבור מעל הסימן # הוא n + 1.

שאלה 3:

לפי נתוני השאלה נראה כי בעבור המכונה המתוארת אין בה יותר כוח מאשר למכונה רגילה.

נעמוד את ההבדלים: מכונה רגילה יכולה לחקות את פעולתה של המכונה החדשה כך:

בעבור כל מעבר מהצורה

(q, a) = (r, b, Rk )

יהיו במכונה הרגילה k-1 מצבים נוספים, , p1, ..., pk1.

ונשים לב כי המעברים המתאימים לכך יהיו:

(q, a) = (p1, b, R)

לכל ולכל (pi, ) = (pi+1, , R) ,1 i < k1

לכל(pk1,) = (r,,R) ,

באופן דומה אפשר לחקות מעברים מן הצורה )(q, a) = (r, b, Lk.

# שאלה 4

1. נרוץ כך:

אם הקלט ריק אז נעבור למצב דוחה (1).

אם הקלט , נעבור למצב מקבל (2).

אחרת נבדוק שהקלט הוא מהצורה ואם לא אז נעבור למצב דוחה (3).

בשלב הבא נעתיק חלק כלשהו באורך בין שני תווים לבין כל רצף ה- (באופן לא דטרמניסטי נבחר מתי לסיים להעתיק את הקטע) אל לאחר מילת הקלט כאשר נכתוב אותיות במקום .

בשלב הבא נבצע אותה פעולה רק שנכתוב אותיות במקום ונכתוב אחרי אותיות ה- (שוב, מתי לסיים להעתיק נבחר באופן לא דטרמיניסטי).

כעת בכל פעם נסמן אות אחת בתור ונעתיק את אותיות ה-c בתור אותיות בסוף הסרט עד שנסיים לסמן את כל אותיות ה- בתגים.

כעת מספר אותיות ה- הוא המכפלה .

נבדוק אם מספר האותיות שווה למספר האותיות , אם כן אז נעבור למצב מקבל שכן הוא מכפלת שני מספרים שונים מ-1 ולכן פריק.

אחרת (4) נעביר למצב דוחה (אין זה אומר ש- ראשוני אלא שזוג המספרים שנבחרו, מכפלתם אינה ).

מאחר והמכונה מקבלת מילה אם"ם יש מסלול מקבל עבורה, נובע שהמכונה מקבלת מילה אם"ם היא מהצורה , וגם או שישנם זוג מספרים בין 2 ל- שמכפלתם היא , כלומר אם"ם הוא פריק (הרי לא יתכן שאחד הגורמים שמכפלתם היא יהיה גדול מ-).

ב.

השפה שהמכונה לאחר השינוי מכריעה היא

בגלל (1),   
**איחוד**  
בגלל (3), {כל המילים מעל א"ב הקלט שלא מהצורה }

**איחוד**

בגלל (4), {כל המילים עבור טבעי כך שקיימים טבעיים המקיימים }

נשים לב שהשפה האחרונה באיחוד היא השפה כי תמיד אפשר לקחת ואז עבור מתקיים ולכן .

לכן השפה שהמכונה החדשה מכריעה היא המשלימה ל-, כלומר

# שאלה 5

בתור א"ב לסרט השני נבחר את הקבוצה כאשר הוא מספר הבחירות הלא דטרמיניסטיות הגדול ביותר שאפשרי ממצב כלשהי בהנתן אות כלשהי בסרט.

בנוסף לשינויים לא"ב הסרט שמתוארים בהוכחת המשפט בספר, נוסיף תו חדש, '' ל-. נשתמש בו בתור מפריד בסרט הראשון כך שנוכל לשמור מילים רבות מופרדות בו בסרט הראשון.

נוסיף גם תוים (כאשר מספר המצבים באוטומט ) שבדיוק אחד מהם יופיע לפני כל תו במצבי הסרט שיקודדו בסרט הראשון; מסמל הראש הקורא כותב אינו על התו העוקב, ו- מסמל שהראש הקורא כותב כן על התו העוקב והמצב הנוכחי של הוא הערך אשר צרוף ל- ( מספר המצבים של ).

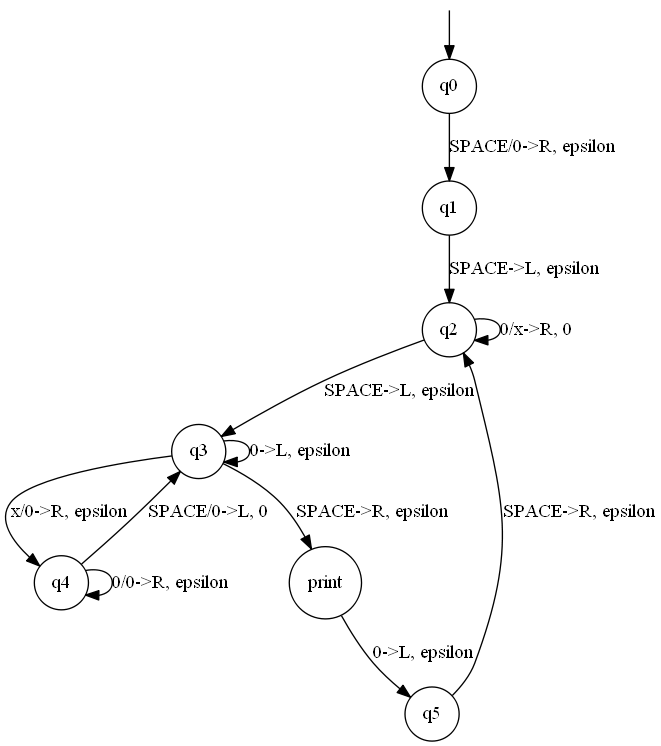
נשתמש בסרט הראשון כדי לדמות מחסנית של מצב הזכרון, נשתמש בסרט השני כדי לדמות מחסנית של מספרי הבחירות ב-.

נשים לב שכל מעבר שאנו מדמים ב- אינו על מצב הסרט הנוכחי אלא על עותק חדש שלו – דבר אשר מאפשר לנו לבצע back track ל-DFS מבלי לחשב את המצב מהתחלה ע"י חזרה לשורש.

1. תחילה סרט 1 מכיל את מילת הקלט וסרט 2 ריק.
2. נכתוב לאחר סוף את התו ואז נעתיק את שוב אחרי הסימן (באופן כמו שתואר בשאלה 2) כאשר ה- אחרי בלבד (לפני התו הראשון של ) – לפני שאר התוים של נכתוב ; נכתוב גם 1 לסרט השני.
3. נבצע באופן הבא:  
   - אם המצב הנוכחי ב- הוא מקבל הרי שיש מסלול מקבל ונעבור למצב מקבל.  
   - אם המצב דוחה ב- או שהערך הנוכחי בסרט השני הוא , אז נבצע backtrack ע"י חזרה מסגרת אחת בסרט 1 תוך מחיקת המסגרת הקודמת ולאחר מכן חזרת הראש הקורא כותב בהתאם למיקום התו כאשר ונחזור גם תו אחד שמאלה בסרט 2.  
   - אחרת, אנו עומדים בסרט 2 על תו בתחום 1..b, נצעד (ראו 4) באפשרות המתאימה בהתאם למצב במסגרת הנוכחית.  
   - וחוזר חלילה
4. הצעידה באפשרות מתבצעת באופן הבא:  
   כותבים בסוף מסגרת הזכרון הנוכחית . משכפלים את מסגרת הזכרון מימין לתו כאשר משנים את תווי ה- לפני כל תו ל- למעט התו שהראש הקורא כותב של מצביע עליו כעת שיהיה בהתאם למצב שעברנו אליו.

# שאלה 6

רוצים לבנות מונה לשפה . (ההנחה שהסרט ריק בתור התחלה)



**הסבר:**

המצבים נועדו רק כדי לאתחל את הסרט לתוכן "1" כאשר הראש הקורא-כותב מצביע על התו הראשון.

אל המצב עוברים תמיד כאשר הראש הקורא-כותב מצביע לתחילת רצף ה-0ים בסרט.

התפקיד של מצב זה הוא להחליף את כל ה-0 ב-x תוך הדפסת 0ים לסרט ההדפסה.

עם הגיענו לסוף המילה, עובדים למצב ואז אנו בכל פעם מחליפים את ה-x הימני ביותר ב-0 ואז מוסיפים 0 בסוף המילה הנוכחית ע"י מעבר ל-, לאחר הוספת 0 זה חוזרים למצב .

כעת אם לא נותרו ים אז סיימנו להכפיל את המילה שעל הסרט, וכל התוים בה הם 0, אז מדפיסים, חוזרים לתחילת המילה הנוכחית בסרט וחוזר חלילה.

נשים לב כי המילה שתודפס כל פעם היא לפי מספר ה-, והמילה הבאה בסרט תכיל 0ים במספר כפול ממספר ה-ים, ששוב יהפכו ל-ים וכן הלאה.

לכן תחילה נדפיס “0”, לאחר מכן “00”, לאחר מכן “0000” וכן הלאה, בכל פעם אורך המילה יוכפל פי 2 – וזו אכן השפה הנדרשת; השפה אינסופית וגם אנו לא עוברים למצב .