**חישוביות ממ"ן 11**

**שם המגיש:** רועי דימינטשטיין

**ת.ז:** 302792361

**שאלה 1**

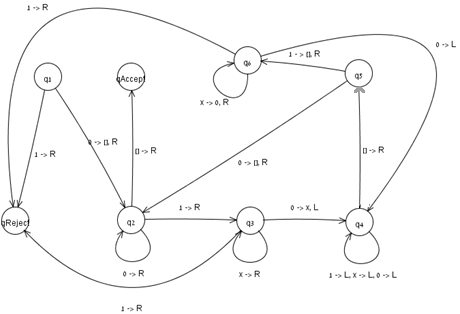
**נתאר מכונת טיורינג המכריעה את D**

רעיון האלגוריתם

הקלט שעל המכונה לזהות מורכב ממקטעים של 0-ים אשר 1-ים מפרידים ביניהם והחוקיות שנשמרת בכל מילה היא שכל מקטע ארוך מקודמו.

1. נעבור על כל המקטעים ועבור כל מקטע נדרוס ב x-ים אפסים מהמקטע שאחריו כאורך ה0-ים במקטע הראשון. נשים לב שכל חלק בקלט שעובר עיבוד נדרס עם . לכן ה הימני ביותר ברצף מתחילת הסרט מסמל את המקום אליו הגענו בעיבוד הקלט
2. אם אחרי הדריסה נשארו עוד 0-ים במקטע השני, ז"א שיחס אורכי המקטעים תקין – השני ארוך מהראשון. אם לא נשארו עוד 0-ים (נתקלנו ב1 לפני שעברנו על כל ה 0-ים במקטע הנוכחי), נעבור ל reject
3. נשחזר את ה0-ים שדרסנו בx ונמשיך לבדוק את המקטעים הבאים
4. אם הגענו לסוף הקלט ולא דחינו ניתן להסיק שווידאנו את תקינות כלל המקטעים ולכן נעבור ל accept

**State Diagram**

Q1 is start state

מהו המצב ההתחלתי? q1?

מדוע המכונה מכריעה את D

1. עבור  -

בכל שלב יסומן [] (blank) במקום 0, נרוץ ימינה עד שניתקל ב1, נמשיך ימינה ונחלוף כלפי x-ים שסימנו בשלבים קודמים ונסמן ב x אחד 0 אחד מהמקטע שאחרי. לאחר מכן נחזור כל הדרך שמאלה עד שנגיע ל [] הראשון שנפגוש ומשם נמשיך אותו הדבר עד שנסיים את כל ה 0-ים במקטע הנוכחי. באיטרציה הראשונה q1 יחליף את ה0 הראשון ב[] ויעביר ל q2 שתפקידו לחלוף על פני שאר ה 0-ים במקטע הנוכחי. במצב בו q2 יתקל כבר בהתחלה ב[], זהו סוף הקלט ולכן לפנינו מילה המכילה רק 0-ים ולכן נקבל. אם q2 נתקל ב-1 אז יש לחלוף על פניו ולאחריו על x-ים כפי שתיארנו, מצב q3 דואג שנחלוף על פניהם וכאשר הוא נתקל ב0 הבא הוא מחליף אותו ב x. משם נצטרך לרוץ חזרה שמאלה ל [] שסימנו במצב q1 או q5 במידה וזה לא המקטע הראשון שאנחנו בוחנים. Q4 מחזיר אותנו שמאלה. Q5 מסמן [] במקום ה0 הבא כאמור אם הוא נתקל ב0 ומעביר שוב ל q2 להמשך התהליך. במידה ו q5 נתקל ב1 סימן שהגענו לסוף מקטע ויש לבחון את תקינות הקלט במובן של האם באמת נשארו עוד 0-ים שלא דרסנו עם x במקטע השני. Q6 חולף על פני כל ה x-ים שסימנו ובנוסף מחזיר אותם ל0-ים לצורך הבדיקה של המקטע הבא, אם הוא רואה שקיים במקטע השני עוד 0-ים שלא סימנו אז המקטע הנוכחי תקין והוא מעביר למצב q4 שיחזיר אותנו ל[] האחרון שסימנו שהוא המקום בקלט אלינו הגענו להמשך בדיקת שאר הקלט. כך נמשיך על כל הקלט, ברגע שנגיע למקטע התקין האחרון מצב q2 יחלוף על פני האפסים האחרונים בקלט, יתקל ב [] ויעבור ל accept

1. עבור 

או שהקלט מתחיל ב1 ואז q1 ישר יעבור ל reject או שאחד המקטעים של ה0-ים קצר מקודמו או שיש מספר שווה של 0-ים. במקרה של מספר שווה כאשר q5 יראה את ה1 האחרון במקטע הנבדק הוא יעביר ל q6 שיחלוף על פני ה x-ים שסימנו, יתקל ב1 במקום ב0 ויעבור ל reject. במקרה של מקטע קצר יותר מקודמו, q3 יחלוף על ה x-ים שסימנו, יתקל ב1 "מוקדם מידי" ויעבור ל reject

הערה: מעברים "לא אפשריים" לא מתוארים בדיאגרמה והכוונה היא כמובן שהם מעבירים ל reject

**שאלה 2**

1. אפשר כי ב q4\q5 עברנו על פני כל ה x-ים לכיוון ימין, לאחר מכן במעבר ל q6 ראינו 0 או 1, החלפנו אותם ל x וחזרנו שמאלה אז או שאנחנו על אחד ה x-ים שחלפנו על פניהם ולפניהם יש # או שלא היו x-ים וזה אומר שאנחנו באיטרציה הראשונה ואנחנו על # כרגע. בכל מקרה 0 או 1 לא יהיו שם
2. אם נסמן את המקום אליו הגענו מתחילת הקלט ב במקום ב x כמו במכונה הנוכחית, נוכל לאחד את q6 ו q7 למצב q8 בעל לולאה שחולפת שמאלה על פני הקלט עד שהוא נתקל ב ואז להעביר ל q1. ה"בעיה" בסימון המיקום עם x היא שאנחנו משתמשים בסימן הזה כדי למחוק גם את התווים שאחרי ה # ולכן אם נשתמש בתו אחר נוכל פשוט "לרוץ" שמאלה עד אליו כדי להמשיך ל"איטרציה" הבאה

**שאלה 3**

כן, למכונה כזאת יש יותר כוח מאשר למכונה רגילה.

הערה מחוץ לגבולות התרגיל:

ראינו בשיעור את פתרון התרגיל הזה באמצעות מ"ט בעלת "מצב לכל מילה" בכל שפה נתונה. את ההוכחה שלי כתבתי לפני השיעור הזה. אני בכל זאת משאיר את ההוכחה שלי משום שאני רוצה לבחון את יכולת הפתרון שלי.. אשמח להערות בונות.

טענה:

באמצעות מ"ט M בעלת אינסוף מצבים ניתן לזהות כל שפה L מעל 

הוכחה:

תהי L שפה כלשהי מעל .

L היא קבוצה המכילה מחרוזות סופיות מעל הא"ב לכן עבור כל מחרוזת w ניתן לייצר מספר סופי של מצבים המכילים מצב qStart, מצב qAccept, מצב qReject ומצבים נוספים שבהינתן והם חלק ממ"ט, ניתן "לתכנת" אותם - כאן כבר איבדתי אותך..

(ממש לייצר מצב עבור כל אות המילה..) כך שבהינתן ואנחנו ב qStart אם המחרוזת בקלט היא w – נעבור למצב qAccept ואם לא, נעבור ל qReject.

נשים לב שאם L היא שפה סופית אין בעיה לזהות אותה בצורה הזאת גם באמצעות מ"ט בעלת מספר סופי של מצבים.

רעיון הזיהוי של L באמצעות M

במידה ו L אינסופית עומדת לטובתנו ההנחה שM בעלת אינסוף מצבים. עבור L אינסופית קיימת M שבה עבור כל מילה בשפה L קיימת קבוצת מצבים סופית כפי שתיארנו מעלה המזהה את המילה.

במכונה M קיימים מספר מצבים נוספים, בין היתר מצב התחלתי שממנו קיים מעבר לכל אחד מ qStart של קבוצות המצבים וקיים מעבר מכל qReject של כל אחת מקבוצות המצבים חזרה אל המצב ההתחלתי.

במכונה קיים סרט נוסף אליו נעתיק את הקלט בהתחלה וממנו נעתיק את הקלט אל סרט העבודה בתחילת כל איטרציה – נקרא לו סרט הגיבוי

במכונה קיים סרט ייעודי נוסף שתפקידו לעזור לנו לבצע מעבר מהמצב ההתחלתי ל qStart של כל אחד מקבוצות המצבים ללא חזרות לקבוצות מצבים שכבר היינו בהם. נקרא לסרט זה סרט הקבוצות

אם w נמצאת בשפה אזי קיימת עבורה קבוצת מצבים ב M אשר מתחילה באחד ממצבי qStart ומזהה אותה. המכונה M תעבור על קבוצות המצבים עד שתמצא אותה

תיאור פעולת המכונה

המכונה M פועלת באופן הבא על קלט w:

1. העתק את w לסרט הגיבוי
2. העתק את תכולת סרט הגיבוי לסרט העבודה
3. בדוק את סרט הקבוצות, לפי תכולתו בחר את qStart של קבוצת המצבים הבאה ועדכן את סרט הקבוצות
4. עבור למצב qStart הבא
5. אם המכונה הגיעה ל qAccept קבל, אם המכונה הגיעה ל qReject עבור לשלב 2

לא הבנתי הרבה ממש שכתבת

(3-)

הסבר

אם  אזי קיימת קבוצת מצבים כמתואר בתחילת הפתרון עבורה. המכונה M מבצעת מעבר על כל קבוצות המצבים ולכן באחת האיטרציות היא תגיע לקבוצת המצבים המזהה את w ותקבל

סיכום

ידוע כי קיימות שפות אשר אינן ניתנות לזיהוי באמצעות מ"ט בעלת מספר סופי של מצבים. לדוגמא השפה  המתוארת בספר בעמוד 210 (corollary 4.23).

לפי הטענה שהוכחה כאן, גם שפה זאת מזוהה טיורינג בהינתן אינסוף מצבים ולכן המודל בעל אינסוף המצבים חזק מהמודל הרגיל.

אין בקיומה של מכונה כזו סתירה לתזה של צ'רץ' טיורינג משום שהתזה מדברת על כך שכל מודל סביר וכללי של מכונת חישוב שקול למודל מ"ט כאשר הכוונה ב"סביר" היא שהמכונה ניתנת לתיאור סופי, כלומר ניתן לרשום תיאור חד-חד ערכי שלה כמחרוזת כלשהי ואין בה יכולות מופלאות כמו אינסוף המצבים המוצגים בשאלה

**שאלה 4**



**תיאור מ"ט M א"ד המזהה את F:**

M על קלט w:

1. סרוק את w וספור כמה # הוא מכיל. נקרא למספר הזה k.
2. אם k=1 דחה
3. נחש i ו j כאשר  וגם 
4. בחן את  מול  באופן הבא:
   1. סמן את ה# שלפני 
   2. סמן את ה# שלפני 
   3. באמצעות "זיגזוג" שמאלה וימינה בין שני ה #-ים המסומנים, השווה את  ל. אם הם שווים, קבל

**הסבר כללי:**

כל ענף אי דטרמיניסטי של המכונה M מבצע השוואה בין ל עבורם M "ניחשה" i ו j.

אם עבור אחד הניחושים האלה, נמצא ש , אותו ענף יקבל ולכן לפי הגדרת מ"ט א"ד, המכונה M תקבל.

**מדוע M מזהה את F**

1. M מבצעת בהכרח מספר סופי של ניחושים מאחר וכל ניחוש של i או j מוגבל ב k ולכן מספר הניחושים השונים מוגבל ב k^2. בנוסף, k בהכרח סופי מאחר וכל מילה בקלט היא סופית ולכן גם קיימים מספר סופי של # בתוכה
2. אם קיימים i, j עבורם , המכונה M תנחש אותם פשוט מאחר ו M מנחשת i, j מתוך כל האפשרויות הקיימות

מכאן שעבור קלט  אחד הניחושים של M יהיה "נכון" כי היא מנחשת את כל האפשרויות עבור הקלט והתהליך סופי ולכן לפחות אחד הענפים יקבל וM תקבל

**שאלה 5**

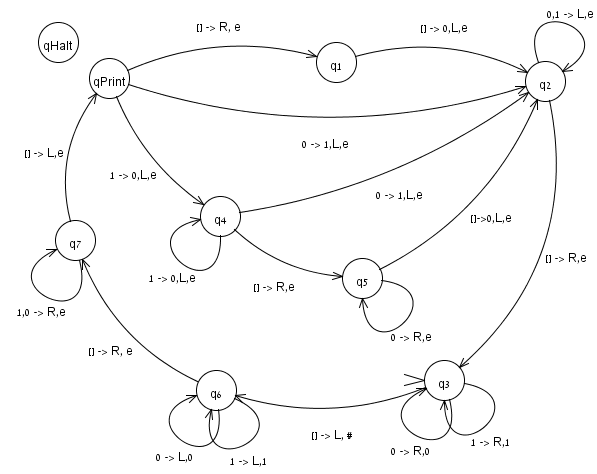


נציג מונה לשפה D

**הרעיון הכללי**

יהיו לנו מספר מצבים שתפקידם לייצר מילה w, הם ידעו לייצר את כל המילים מ {0,1}\*. מנגד יהיו לנו מספר מצבים שתפקידם יהיה לרוץ על המילה ולהדפיסה משמאל לימין, להדפיס # ולאחר מכן להדפיס את המילה מימין לשמאל. בסוף ההדפסה השליטה תחזור למצבים המייצרים בכדי לייצר את המילה הבאה

**State Diagram**

****

* מהו המצב ההתחלתי? qprint?

מעט הסברים סינטקטיים (שנובעים מהתוכנה לשרטוט בה השתמשתי..):

1. בכל מקום שמסומן [] הכוונה היא ל
2. בכל מקום שמסומן e הכוונה היא ל 
3. במצבים q3 ו q6 נאלצתי לשרטט 2 קשתות עצמיות למרות שניתן לתאר אותן באמצעות אחת אבל לא יכולתי לרדת שורה בתיאור המעבר בתוכנה
4. q3 הוא המצב ההתחלתי

**הסבר פעולת המונה**

כפי שתיארתי ברעיון הכללי קיימים מצבים המייצרים מילה וקיימים מצבים שתפקידם להדפיס אותה בפורמט של מילה הנמצאת ב D.

המצבים המדפיסים הם q2, q3, q6, q7 ודרך פעולתם היא כך:

1. q3 הינו מצב ה"מצפה" שיעברו אליו כאשר על הסרט כתובה המילה שיש להדפיס (אותה ואז # ואז את היפוכה..) והראש של סרט העבודה נמצא בתחילת המילה שיש להדפיס (בקצה השמאלי שלה). q3 יודע לרוץ מתחילתה של המילה ועד הקצה השמאלי שלה ולשלוח אותה לסרט ההדפסה. כאשר הוא מגיע לקצה הימני (פוגש ב ) הוא כותב # לסרט ההדפסה, לוקח את הראש של סרט העבודה אחד שמאלה לקצה הימני של המילה הנוכחית ומעביר למצב q6. נשים לב שהוא המצב ההתחלתי משום שיש להדפיס את המילה '#' עוד לפני שמייצרים מילים מעל {0,1}\*
2. q2 – זהו מצב שלוקח את הראש הקורא שמאלה עד הקצה של המילה הנוכחית כדי ש q3 יקבל אותה כפי שהוא מצפה. כל "תהליך הדפסה" מתחיל ממנו
3. q6 כותב לסרט ההדפסה את המילה הנוכחית בסדר הפוך, מהקצה הימני של המילה שם השאיר q3 את הראש הקורא ועד הקצה השמאלי שלה. כאשר הוא מגיע לתחילת הסרט (שוב, רואה ) הוא מעביר למצב q7
4. q7 אחראי על להעביר את השליטה למצבים המייצרים את המילים במצב בו הם מצפים לו, הוא בסהכ מביא את הראש הקורא לקצה הימני של המילה הנוכחית כדי שיהיה יותר "קל" למצבים המייצרים לייצר את המילה הבאה. לכאורה יכולתי לוותר עליו ולשנות את המצבים האחרים כך שידעו להתמודד עם זה שהראש מתחיל בקצה השמאלי אבל לא הייתי בדחק של מצבים. בנוסף q7 מעביר ל qPrint כדי שידפיס את מה ש q3, q6 כתבו על סרט ההדפסה

המצבים המייצרים את המילה הם qPrint, q1 ,q4, q5 ודרך פעולתם היא כך:

1. qPrint כשמו, מדפיס את המילים לאחר ש q3, q6 הדפיסו אותם לפי החוקיות על סרט ההדפסה. מצב זה גם משמש כ switch להחלטת הדרך לייצר את המילה הבאה:
   1. כאשר הוא רואה שהאות האחרונה היא זאת אומרת שאנחנו בהתחלה והוא מעביר ל q1 שייצר את המילה הראשונה – '0' ומשם q1 יעביר ל q2 להדפסה
   2. כאשר האות האחרונה במילה הנוכחית היא 0 הוא פשוט יחליף אותה ל 1 כי זה יביא אותנו למילה הבאה ב {0,1}\* ומעביר ל q2 להדפסה
   3. כאשר האות האחרונה במילה הנוכחית היא 1, הוא יחליף אותו ב0, יעביר למצב q4 שיחפש את ה 0 הבא כדי לקדם אותו. כל עוד q4 רואה 1-ים הוא הופך אותם ל0. בסוף או ש q4 נתקל ב0 ,הופך אותו ל 1 ושולח ל q2 להדפסה או שהוא לא נתקל ב 0 בשום מקום במילה ואז כשהוא מגיע לסוף הוא מעביר ל q5 שמוסיף למילה (שעכשיו כולה 0-ים) עוד 0 אחד בסוף ומעביר ל q2 להדפסה

**שאלה 6**

1. כיוון ראשון, קיים מונה המפיק את A וכל מילה מודפסת פעם אחת  A מזוהה טיורינג.

נציג מ"ט M המזהה את A. M על קלט w:

* + 1. הרץ את המונה של A. בכל פעם שהוא מדפיס מילה השווה אותה עם w
    2. אם המונה הדפיס את w, קבל

קל לראות ש M מקבלת את המילים שהמונה מייצר והכיוון הזה גם זהה לכיוון הראשון בהוכחת 3.21

1. כיוון שני,

נציג מונה E לשפה A המדפיס כל מילה פעם אחת בלבד בהינתן מ"ט M המזהה את A.

**הרעיון של המונה:**

למונה יהיו 2 "איזורים" על סרט העבודה – איזור כתיבת המילים שכבר הדפסנו נקרא לו איזור המודפסים, ואיזור בו נשתמש למנייה לקסיקוגרפית של המילים בשפה בו בכל שלב תיהיה המילה הנוכחית ונקרא לו איזור המנייה

נעבור על כל המילים בא"ב בסדר לקסיקוגרפי כמו במשפט 3.21 ונכתוב אותן על איזור המנייה אבל בנוסף באיזור המודפסים נכתוב את כל ה -ים שכבר הדפסנו. בכל שלב כאשר M תקבל מילה  מסויימת, E ירוץ על איזור המודפסים ורק אם יראה שהוא לא רשם שם את  בעבר הוא ידפיס ויוסיף אותה לאיזור המודפסים

איזור המנייה יתחיל מתחילת הסרט, לאחר מכן יהיה $ ולאחר מכן יכתב איזור המודפסים כאשר כל מילה שהודפסה בעבר מופרדת באמצעות #.

נשתמש בסימון שבמשפט 3.21 עבור רשימת כל המחרוזות ב ∑ באמצעות s1, s2, s3,…

לשם המחשה ככה נראה סרט העבודה אם הדפסנו כבר את מילים 00, 01 והמילה הנוכחית היא 11:



**פעולת המונה E:**

1. חזור על הצעדים הבאים עבור i = 1,2,3,…:
   1. הרץ את M למשך i צעדים על כל אחד מהקלטים s1, s2, s3, …, si
   2. אם M עצרה על si במצב מקבל, בדוק האם נמצא באיזור המודפסים. אם לא, בצע את השלבים הבאים:
      1. הדפס את si
      2. הוסף את  באיזור המודפסים

המונה המתואר פועל בדומה למונה ב 3.21, מריץ את M עם i-ים הולכים וגדלים כל פעם ל i צעדים על s1, s2, s3,…,si ולכן נעבור על כל המילים הקיימות ותתאפשר ל M האפשרות לנסות לקבל את כולן.

בכל פעם ש M מקבלת מילה si, המונה בודק אם הוא כבר הדפיס את המילה הזאת באמצעות בדיקה מול איזור המודפסים עליו הוא מנהל רשימה של המילים שהוא כבר הדפיס ולכן אין אפשרות שמילה תודפס יותר מפעם אחת

התהליך של הבדיקה "האם כבר הודפסה המילה הזאת?" הוא סופי משום שבכל איטרציה i הודפסו לכל היותר i מילים ולכן המונה יצטרך לבצע לכל היותר סדר גודל של i השוואות ועדכונים מול סרט העבודה וההדפסה כאשר בכל איטרציה i הוא מספר קבוע וסופי

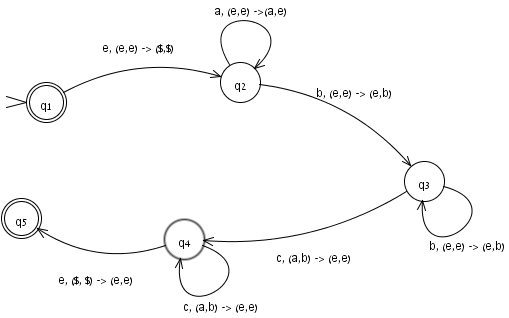
לסיכום המונה מדפיס את כל המילים אותן M מקבל, אך ורק פעם אחת וכל איטרציה שלו היא סופית וחסומה בסדר גודל של i צעדים

**שאלה 7**

**סעיף א**



State Diagram

****

הסבר

פעולת האוטומט:

1. מתחיל ממצב q1 שהוא מצב מקבל משום שהמילה הריקה מקיימת את החוקיות
2. מכניס $ לכל אחת מהמחסניות כדי לסמן את סוף המחסניות (במעבר מ q1 ל q2 )
3. עבור כל a שהוא מקבל בקלט הוא דוחף a למחסנית הראשונה (לולאה בq2)
4. עבור כל b שהוא מקבל בקלט הוא דוחף b למחסנית השניה (מעבר מq2 ל q3 ובלולאה ב q3)
5. עבור כל c שהוא מקבל בקלט הוא מוציא a ממחסנית 1 ו b ממחסנית 2 (מעבר מ q3 ל q4 ובלולאה ב q4)
6. אם מצב q4 רואה את סוף 2 המחסניות, עוברים למצב q5 – המצב המקבל

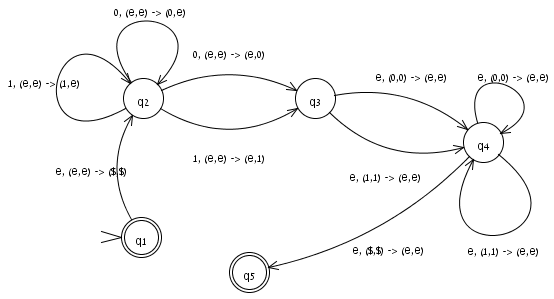
נכונות האוטומט

האוטומט מקבל ב2 מצבים, הראשון זאת המילה הריקה אשר מקיימת את החוקיות של השפה והשני הוא כאשר מגיעים למצב q5 אשר אליו האוטומט יגיע רק אם מצב q4 רואה את סוף המחסנית ב2 המחסניות ביחד.המצבים הקודמים ל q4 דואגים לדחוף למחסנית 1 את כל ה a-ים שנכנסים מהקלט ומחסנית 2 את כל ה b-ים שנכנסים מהקלט. מאחר ו q4 מוציא זוגות של a וb בכל צעד, אם הגענו לסוף המחסניות ביחד זאת אומרת שמספר ה a-ים בקלט שווה למספר ה b-ים בקלט. מאחר ומוציאים את הזוגות האלה בכל צעד בכל פעם שמקבלים c – מספר ה a-ים ו b-ים שהוצאו שווה למספר ה c-ים בקלט ברגע בו 2 המחסניות ריקות ולכן יש לעבור למצב מקבל כפי שהמכונה מתארת

**סעיף ב**



State Diagram

****

הסבר

פעולת האוטומט:

1. מתחיל ממצב q1 שהוא מצב מקבל משום שהמילה הריקה מקיימת את החוקיות
2. מכניס $ לכל אחת מהמחסניות כדי לסמן את סוף המחסניות (במעבר מ q1 ל q2 )
3. דחוף את הסמלים הנקראים מהקלט באופן לא דטרמיניסטי למחסנית 1 או למחסנית 2 (מצבים q2 וq3)
4. בכל שלב, "נחש" שהכנסנו את החצי הראשון של הקלט למחסנית 1 ואת החצי השני למחסנית 2 ונסה להוציא את הסמלים מ2 המחסניות באמצעות הוצאת סמל זהה מ2 המחסניות ביחד בכל צעד (מצב q4)
5. אם הצלחנו לרוקן את שתי המחסניות ביחד (הגענו ל $ בשתיהן) עבור למצב q5 – המצב המקבל

נכונות האוטומט

האוטומט מנחש חלוקה כלשהי של הקלט על מנת לנסות לדחוף את החצי הראשון של הקלט עד כה למחסנית 1 ואת החצי השני למחסנית 2. הוא עושה זאת ע"י מעברים לא דטרמיניסטיים בקבלת הקלט במצבים q2 ו q3. לאחר הניחוש, אם הקלט נמצא בשפה, בכל מחסנית יהיה חצי מהקלט מסודר בדיוק באותו הסדר ולכן מה שנשאר זה להוציא זוגות של סמלים זהים מ2 המחסניות בהתאמה ולקבל אם 2 המחסניות התרוקנו ביחד

**הערה** – ההסברים הסינטקטיים שרשמתי עבור שאלה 5 נכונים גם עבור השרטוטים האלה (מוגבלויות של תוכנת העריכה לאוטומטים שהשתמשתי בה..)

(1-)ב-q3 צריכה להיות לולאה עצמית שמכניסה עוד תווים מהקלט למחסנית