**מבוא לחישוביות וסיבוכיות  
ממ"ן 12 – ערן ברודר**

**שאלה 1**

*הערה: במהלך ההוכחה אני משתמש בצורה חופשית במשפטים מתוך הקורס "אוטומטים ושפות פורמליות". אין בידי את הספרים, ולכן אני לא מביא סימוכין מדוייקים. כל שימוש כזה אציין ב(\*)*

נתאר כיצד ניתן לבנות מכונת טיורינג המכריעה את השפה.

קיים אלגוריתם המעביר כל ביטוי רגולרי לאוטומט סופי לא דטרמיניסטי. (\*)  
קיים אלגוריתם המעביר כל אוטומט סופי לא דטרמיניסטי לאוטומט סופי דטרמיניסטי. (\*)  
לכן, קיים אלגוריתם המעביר כל ביטוי רגולרי לאוטומט סופי דטרמיניסטי.

קיים אלגוריתם המעביר כל אוטומט סופי דטרמיניסטי לאוטומט אחר המייצג את המשלים של השפה. (\*)

קיים אלגוריתם לבניית אוטומט סופי דטרמיניסטי שמהווה חיתוך בין שתי השפות של שני אוטומטים אחרים. (\*)

נגדיר את הביטוי הרגולרי הבא . הוא מייצג את שפת כל המילים בשפה שמכילות את המחרוזת 111. המשלים של השפה הוא כל המילים שאינן מכילות את המחרוזת 111.

על סמך האמור לעיל, אם הוא ביטוי רגולרי כלשהו, נוכל לבנות אוטומט המייצג את החיתוך בין השפה של לבין השפה המשלימה של S.

אם יש לפחות מילה אחת ב , שמכילה את המחרוזת 111, השפה המיוצגת ע"י האוטומט של לא תהייה ריקה, והאוטומט שלה יקבל לפחות מילה אחת.

ולהפך : אם כל המילים ב מכילות את המחרוזת 111, השפה המיוצגת ע"י האוטומט של היא השפה הריקה, והאוטומט לא יקבל דבר.

הראנו למעשה כיצד ניתן לבנות מביטוי רגולרי כלשהו אוטומט אחר שידחה את כל המילים במידה ו הוא ביטוי רגולרי כך ש .

משפט 4.4 קובע כי כריעה, ומציע גם מכונה מתאימה.

כל שעלינו לעשות הוא להזין את המכונה שתוארה בספר באוטומט התואם ך כפי שתואר למעלה.  
אם המכונה מקבלת את הקלט – אז נסיק כי , ונקבל גם כן.  
אחרת, נדחה.

ולכן כריעה.

**שאלה 2 - סעיף א'**

תיאור ההוכחה : נראה כי ניתן להציג את הקבוצה כאוסף בן מנייה של קבוצות בנות מנייה.  
כפי שלמדנו בקורס "מתמטיקה דיסקרטית", אוסף כזה גם הוא בן מנייה בעצמו.

מכיוון שהאלפבית שלנו הוא בן מנייה, נוכל לתת שמות לכל אות, לפי סדר המספרים הטבעיים.  
התו הראשון יקרא '1', השני יקרא '2' וכו'.  
הקבוצות שאותן אנו נאחד הן:

הקבוצה הריקה:   
הקבוצה **הראשונה:** כל המחרוזות בנות תו **אחד** לכל היותר, המורכבות מהתווים **{1}**  
הקבוצה **השנייה** : כל המחרוזות בנות **שני** תווים לכל היותר, המורכבות מהתווים **{1,2}**  
הקבוצה **השלישית**: כל המחרוזות בנות **שלושה** תווים לכל היותר, המורכבות מהתווים **{1,2,3}  
.  
.  
.**הקבוצה ה n: כל המחרוזות בנות **n** תווים לכל היותר, המורכבות מהתווים **{1,2,3…,n}  
.  
.**

עצמת כל קבוצה היא סופית, שהרי הגודל סופי, והאלפבית בה סופי.

כל שנותר להראות הוא, שלכל מחרוזת סופית יש קבוצה שמכילה אותה:

תהי מחרוזת סופית כלשהי.  
יהי "ערך" התו הגבוהה ביותר במחרוזת.  
יהי אורך המחרוזת  
יהי

הקבוצה וודאי מכילה את , שהרי כל התווים הערך שלהם קטן או שווה ל , והרי לפי ההגדרה הם חלק מהאלפבית של הקבוצה . כמו כן, אורך המחרוזת הוא לכל היותר , וזה האורך המקסימלי בקבוצה .  
ולכן, ובהסתמך על הגדרת הקבוצה , בהכרח מתקיים .

הקבוצה אם כן היא בן מנייה והיא מכילה את קבוצת כל המחרוזות הסופיות מעל .

**ומכאן שגם קבוצת כל המחרוזות הסופיות מעל היא בת מנייה גם כן.**

מ.ש.ל

**שאלה 2 - סעיף ב'**

נראה דוגמא ספציפית שעבורה הקבוצה הנתונה איננה בת-מנייה.

נגדיר את אלפבית שיהיה . בוודאי ש

נשתמש בהוכחה של קנטור שמראה כי

תהי קבוצת כל המחרוזות האינסופיות מעל .

כמובן, שלכל מספר בין 0 ל 1 מהצורה נוכל להתאים את המחרוזת .  
נבחר לייצג את כל המספרים בצורה שיהיו להם אינסוף ספרות.   
ז"א את 5, למשל, ניתן להציג גם כ או כ .

לכן קיימת פונקציה חח"ע ועל בין קבוצת המספרים לבין G.

ומכיוון ש נובע גם ש .

זו הייתה דוגמא ספציפית כאשר , עבורה קל היה לנו להציג את המספרים שמיוצגים בבסיס עשרוני.

עם זאת, אין כל בעיה להכליל את ההוכחה לכל אלפבית **סופי** אחר, אם נחליט על ייצוג מתאים של שיטת ספירה.

לדוגמא, עבור האלף בית נציג פונקצייה חח"ע מהקבוצה לקבוצת המחרוזות האינסופיות שמייצגות מספרים בבסיס בינארי, ובכך נראה למעשה כי גם קבוצה זו אינה בת מנייה.

חשוב להדגיש, כי אין אנחנו מוגבלים לשפות שהאלפבית שלהם מייצג "ספרות", שהרי די לנו להציג פונקציה חח"ע מכל אלפבית סופי אל האלף-בית הספרתי הסופי המתאים, והרכבת הפונקציות תהייה גם היא פונקצייה חח"ע מהקבוצה אל השפה הנדונה.

מ.ש.ל

**שאלה 3 - סעיף א'**

נעבור על כל שלבי המכונה.  
*הערה: על סמך תשובתו של ד"ר אלעזר בירנבוים, הקלט הוא תיאור של מכונת טיורינג*

1. ***מצא את כך ש- היא המחרוזת ה לפי הסדר הסטנדרטי.***וודאי שצעד זה לא יימשך לעד, שהרי בשלב מסויים נגיע ל המבוקש אם נקפיד על הסדר.
2. ***הרץ את המונה עד שהוא מדפיס את המחרוזת ה - .***

כנ"ל. אנחנו חסומים במספר האיטרציות.

1. **הרץ את על .**

מכיוון שהמכונה A "הופקה" ע"י המונה , בהכרח זו מכונה שעוצרת עבור כל קלט, בפרט עבור הקלט .

ולכן, המכונה עוצרת תמיד.

**שאלה 3 - סעיף ב'**

מכיוון שהמכונה עוצרת תמיד (כפי שהראינו), היא מזוהה ע"י המכונה של , ולכן היא גם מופקת אחרי איטרציות (כאשר ) ע"י המונה שלה.

**שאלה 3 - סעיף ג'**

נעבור שוב על שלבי האלגוריתם ונגיע לסתירה.  
  
0. הרץ את על (נתייחס להוראה זו כאל ההוראה הראשונה שמתחילה למעשה את האלגוריתם).

1. ***מצא את כך ש- היא המחרוזת ה לפי הסדר הסטנדרטי.***כפי שתואר בשאלה, במקרה זה .
2. ***הרץ את המונה עד שהוא מדפיס את המחרוזת ה - .***

המחרוזת שקיבלנו מתאימה למכונה . כלומר .

1. **הרץ את על .**  
   למעשה השורה הזו שקולה להוראה : "הרץ את על ".  
   בעצם חזרנו לשורה 0 בדיוק באותו מצב, ונתחיל להריץ שוב את מכונה M על אותו הקלט.  
   כלומר המכונה M לעולם לא עוצרת. וזו סתירה!

כמו כן נשים לב, שאם בשלב 3 אנחנו מקבלים ש מקבלת את , אז אנחנו דוחים, ואז למעשה דוחה את . אם בשלב 3 דוחה את אז אנחנו מקבלים, ולמעשה מקבלת את .   
וגם זו סתירה!.

מ.ש.ל

**שאלה 4**

נפתח את התשובה באבחנה עליה תתבסס ההוכחה בהמשך.  
באיזה שלב אנחנו אומרים כי מכונת טיורינג "מקבלת" או "דוחה"? שלא כמו באוטומט רגיל, שם אנחנו בוחנים את המצב שאנחנו נמצאים בו (אם הוא מקבל או לא) **כשנגמר הקלט**, במכונת טיורינג אנחנו בודקים את המצב כשהמכונה החליטה שהיא סיימה, והיא לא זזה יותר. בשלב זה נסתכל על המצב: אם הוא מקבל, אז אנחנו אומרים שהמכונה קיבלה את הקלט. אם הוא לא מקבל, אנחנו אומרים שהמכונה דחתה את הקלט.  
תחת הבחנה זו נציג את ההוכחה.

אנחנו צריכים דרך להבחין בין מילה שלא התקבלה במכונה בגלל שהמילה נדחתה, לבין מילה שלא התקבלה במכונה בגלל שהמכונה רצה עד אינסוף. מבחינת אין הבחנה כזו. כדי לפתור את הבעיה, עבור כל מכונה שנרצה לבדוק, נייצר מכונה שכמעט זהה לה למעט הבדל אחד : כל המצבים שלה יהיו מקבלים (או בצורה פורמלית יותר – נוסיף צעד נוסף לכל המצבים שמטפל במעבר למצב המקבל היחיד עם סיום העבודה של המכונה). בצורה כזו, אם המכונה של תכריז כי המילה לא התקבלה נדע בוודאות שהסיבה לכך היא שהמכונה עם השינוי רצה ללא הפסקה. אחרת נדע שהיא הפסיקה, ואז נוכל ללא חשש להריץ את המכונה המקורית M על הקלט, כי אנחנו יודעים שהיא תעצור.

כעת נציג את האלגוריתם בצורה רשמית יותר:

Let’s assume for the purpose of obtaining a contradiction that TM R decides . we construct TM S to decide , with operating as follows:

On input an encoding of a TM M and a string

1. Create a new machine which is the same as , only all it’s states are accepting.
2. Run TM R on input .
3. If R rejects, .
4. If R accepts, simulate on until it halts.
5. if M has accepted, , if M has rejected, ”

**שאלה 5 – הקדמה**

עפ"י משפט רייס:

לכן, אם נראה כי התכונה המתוארת בכל סעיף היא תכונה לא-טריוויאלית, יהיה בזה די על מנת להראות כי השפה איננה כריעה. במידה והתכונה טריוויאלית, משפט רייס לא אומר דבר על השפה, ולא ניתן להוכיח בעזרתו דבר לגבי הכריעות.

נזכיר כי תכונה טריוויאלית מקיימת:

**שאלה 5 – סעיף א'**

נראה כי (למעשה A) היא תכונה טריוויאלית, ולכן ניתן להשתמש במשפט רייס.

נראה כי איננה ריקה (וברור שאיננה מכילה את **כל** המכונות).

המכונה הפשוטה שעבורה שמכילה מצב אחד בלבד, שהוא המצב ההתחלתי והיא גם מצב שאינו מקבל. בנוסף, מכולה זו איננה עושה דבר, אלא נשארת במצב הראשוני שלה. זו למכונה שעבורה .

מאידך, וודאי שאם מכונות עבורן מתקיים אז הן באותו הגודל, או שהגודל של שתיהן קטן מחמישים או שהגודל של שתיהן גדול\שווה מחמישים ולכן .

הראנו למעשה כי היא תכונה טריוויאלית של , ולכן עפ"י משפט רייס  **לא כריעה.**

**שאלה 5 – סעיף ב'**

נראה כי התכונה בשאלה איננה תכונה טריוויאלית, ולכן לא ניתן להשתמש במשפט רייס.

לשם כך נתבונון במכונה , שבנויה באופן הבא: המכונה מכילה מצב מקבל אחד, שהוא המצב ההתחלתי שלה, ואין לה עוד מצבים. המכולה לא עושה דבר, אלא נשארת במצב מקבל. במצב כזה ברור ששפת המכונה היא , וכי היא מקבלת כל מילה בפחות מ-1000 צעדים (למעשה היא מקבלת אחרי 0 צעדים), ולכן .

כעת נתבונן במכונה . מכונה זו תכיל 1,001 מצבים, שיסומנו באותיות כאשר .  
המצב המקבל היחיד הוא .   
המכונה מתחילה ב .  
לכל אחד מ-1,000 המצבים הראשונים פונקצית המעבר היא זהה: הזז את הראש שמאלה, אל תשנה את הקלט, ועבור למצב שמספרו הסידורי גדול ב-1.

כשמגיעים למצב עוצרים במצב מקבל.

לכן מתקיים .

אבל בגלל שהמכונה תמיד מבצעת מעל 1000 צעדים, מתקיים .

הראנו מצב בו אבל (כאשר )

לכן התכונה הנ"ל איננה טריוויאלית, **ולא ניתן** להשתמש במשפט רייס כדי להוכיח שהשפה איננה כריעה.

**שאלה 5 – סעיף ג'**

נראה כי היא תכונה טריוויאלית.

וודאי ש איננה ריקה (כבר הצגנו מספיק דוגמאות בשלב זה) שהרי יש מכונות שמכריעות שפה (לפי ההגדרה).  
וודאי ש איננה מכילה את כל המכונות, שהרי יש מכונות שהשפה שלהן ניתנת לזיהוי, אבל איננה כריעה (דוגמאת בעיית העצירה – המכונה שפשוט מריצה מכונות נתונה על מילה וקובעת אם הן עוצרות)

ובקיצור – יש שפות לא כריעות שקיימת להן מכונה שמזהה אותן.

אם יש שתי מכונות כך ש אז בהכרח או שהשפה של שתי המכונות כריעה או שלא, ואז או ששתיהן ב או ששתיהן לא.

לכן P טריוויאלית, וניתן להשתמש במשפט רייס.

**שאלה 6**

נתאר תחילה כיצד לייצג את הקלט

* כל מצב והמעברים שלו ייוצגו ע"י מחרוזת שתתחיל בשם המצב, אחריה אות מהאלפבית ואחריה כל המצבים האפשריים שאות זו מובילה אליהם (מסעות אפסילון יסומנו ע"י שימוש בתו המתאים). אחרי רשימת כל המצבים האפשריים, תופיע האות הבאה בתור, ואחריה כל המצבים האפשריים, עד שנכנס את כל המעברים האפשריים למצב.
* כל מחרוזת שתוארה מעלה תופרד זו מזו ע"י סימן מיוחד .
* לשם נוחות, נגדיר כי המצב ההתחלתי הוא זה שמופיע ראשון.
* כל מצב מקבל יסומן ע"י תו מיוחד. אם, למשל, המצב הוא מצב מקבל, בתיאור הוא יופיע כ .
* אחרי תיאור המכונה תופיע המחרוזת עצמה.
* לשם פישוט השאלה, נניח כי למחרוזת המתארת את המכונה והקלט יש קריאה יחידה. אם לשם כן אנחנו צריכים להניח או לכפות מצב שלא תהייה חפיפה בין סימונים של המצבים, האלפבית, וסימן ההפרדה – אין בעיה בכך.

לדוגמא קלט תקין יראה כך:

*התנהגות המכונה מזכירה במידת מה את שיטת העבודה של האלגוריתם להעברת אוטומט סופי לא דטרמיניסטי, לאוטומס סופי דטרמיניסטי.המכונה שלנו תבצע סימולציה של האטומט על מילת הקלט. בכל שלב אנחנו מסמנים מהם המצבים שכרגע הגענו אליהם (בגלל שזהו אוטומט לא דטרמיניסטי אחרי קריאת כל תו אפשר להגיע ליותר ממצב אחד). אנחנו קוראים את האות הבאה, ואז מסמנים את כל המצבים האפשריים שאליהם אפשר להגיע עכשיו.*

*המכונה תעבוד בצורה הבאה:*

1. *סמן את המצב ההתלתי בסימון מסוג #1, שמציין שזה המצב הנוכחי.*
2. *לכל כאשר היא מילת הקלט המופיעה בסוף המחרוזת:*
   1. *עבור כל מצב מסומן מסוג #1*
      1. *בצע את כל מסעות האפסילון שהוא מאפשר, והוסף לכל מצב מצב חדש בדרך שלא היה מסומן עדיין, את הסימן #1.*
   2. *אם בתהליך של היו סימונים חדשים, בצע שוב את (עד שמיצינו את מסעות האפסילון)*
   3. *עבור כל מצב מסומן מסוג #1*
      1. *הוסף סימון מסוג #2 לכל המצבים שאליהם ניתן להגיע מהמצב הנ"ל,  
          ע"י קריאת האות*
   4. *הורד את כל הסימנים המיוחדים מסוג #1*
   5. *החלף את כל הסימונים המיוחדים מסוג #2 לסוג #1*
3. *בסיום קריאת מילת הקלט , עבור על כל המצבים המסומנים ב #1*
   1. *אם יש מצב מסומן שהוא גם מצב מקבל –* ***קבל****.*
4. *אם בצעד 3 לא קיבלנו,* ***דחה.***

*נשים לב, שבשום שלב לא השתמשנו בחלק נוסף של הטייפ, והשתמשנו במספר סימנים חדשים סופי.  
לכן המכונה שהצגנו היא LBA.*

**שאלה 7 – סעיף א'**

*נניח כי כריעה, והמכונה המכריעה אותה היא .*

*נבנה מכונה חדשה באופן הבא שתהווה את פונקציית המיפוי שלנו:*

***= "עבור קלט :***

1. ***בנה מכונה חדשה כך:  
     
    = "עבור קלט :***
   1. ***הרץ את על .***
   2. ***אם קיבלה את , קבל את , אחרת דחה את .”***
2. ***החזר את P***

נשים לב שאם מקבלת את זה אומר שהרצת על התקבלה.  
אם הרצת על התקבלה, זה אומר ש קיבלה.  
אם קיבלה, זה אומר ש אינסופית, וזה קורה רק אם קיבלה את , כי אז .

אם לא קיבלה את אז , ואז גם לא מקבלת את , וגם לא מקבלת את

ולמעשה הראנו פונקציית מיפוי, שהרי

**שאלה 7 – סעיף ב'**

*נניח כי כריעה, והמכונה המכריעה אותה היא .*

*נבנה מכונה חדשה באופן הבא שתהווה את פונקציית המיפוי שלנו:*

***= "עבור קלט :***

1. ***בנה מכונה חדשה כך:  
     
    = "עבור קלט :***
   1. ***הרץ את על לכל היותר צעדים, עד לקבלת המילה.***
   2. ***אם קיבלה את , דחה את , אחרת קבל את .”***
2. ***החזר את P***

הסבר: הפונקציה שקיבלנו מחזירה מכונה שפועלת כך עבור :  
אם מקבלת את , אז היא בהכרח מקבלת את במספר סופי של צעדים, נאמר .  
אם היא אכן קיבלה את , אז לכל שהמכונה תקבל, שמקיים , המכונה לא תקבל את , אז וודאי שהשפה של סופית, ולכן . ההיפך גם נכון : אם לא מקבלת את , אז על סמך אותם שיקולים, המכונה תקבל כל מילה ולכן היא אינסופית.

ולמעשה הראנו פונקציית מיפוי, שהרי

**שאלה 7 – סעיף ג'**

כפי שמוסבר בע"מ 238 בספר

(1) .

הראנו כי

(2)

ומשילוב השניים נקבל

(3)

עפי משפט 4.23 אינה מזוהה טיורינג, ובשילוב עם משפט 5.29 ו (3) נקבל כי גם **אינה מזוהה טיורינג**.

ובאותה שיטה:

ולכן גם **אינה מזוהה טיורינג**.