9. פונקציות שאינן ניתנות לחישוב

למן הצעד הראשון בקורס נאמר, שלא כל הפונקציות ניתנות לחישוב. הצהרה זו, הנשמעת מפתיעה בשמיעה ראשונה, היתה בין הגורמים שהפנו את תשומת ליבנו לזהירות שיש לנקוט בנסיון לבירור אילו פונקציות ניתנות לחישוב. בינתיים, צברנו יותר ניסיון ואפילו ראינו רמזים כמעט מפורשים לקיום פונקציות שאינן ניתנות לחישוב. בפרק זה נעסוק בנושא זה במפורש ובפירוט.

9.1 נימוק ספירה

לא רק שישנן פונקציות שלא ניתנות לחישוב, אלא שרוב הפונקציות הן כאלו. עובדה זו נובעת מתוך השוני בעוצמות של קבוצת כל הפונקציות ושל קבוצת כל הפונקציות הניתנות לחישוב.

<u>טענה 9.1:</u> ישנה התאמה חח"ע ועל בין המספרים הממשיים בקטע [0,1] לבין קבוצת כל הפונקציות הבוליאניות (ממחרוזות לביט בודד) [הטענה כבר הופיעה בצורה דומה בפרקים 6.1, 6.2 ומופיעה כתרגיל בפרק הרקע הטכני].

<u>תזכורת:</u> נשתמש בסדר הלקסיקוגרפי (שהוגדר בפרק הרקע הטכני) על קבוצת המחרוזות, ונתאים לכל מספר ממשי (ביצוגו האינסופי) $r=0.\sigma_1\sigma_2\sigma_3\cdots$ את הפונקציה f_r המוגדרת $f_r(x_i)=\sigma_i$ כאשר x_i הוא המחרוזת ה i-ית בסדר הלקסיקוגרפי. ודא כי התאמה זו הינה חח"ע.

<u>טענה 9.2:</u> ישנה התאמה חח"ע ועל מקבוצת כל מכונות טיורינג אל קבוצת כל המחרוזות (שהיא קבוצה הניתנת למניה).

טענה דומה הושמעה כבר ביחס לפונקציות רקורסיביות (פרימיטיביות) וביחס לכל המכונות במודל חישוב שמכונותיו ניתנות לתיאור סופי. לפיכך, ישנה התאמה חח"ע בין קבוצת הפונקציות הניתנות לחישוב ע"י מכונת טיורינג, לבין קבוצת הטבעיים.

<u>פתרון:</u> נתאים לכל מכונת טיורינג מחרוזת המתאימה לקידוד סטנדרטי של המכונה (ראה פרק 8.1).

משתי הטענות נובע כי עוצמת קבוצת כל הפונקציות ממחרוזות לביט בודד היא כעצמת הממשיים, בעוד שעצמת קבוצת מכונות טיורינג (ולכן גם קבוצת הפונקציות הניתנות לחישוב ע"י מכונת טיורינג) היא כעצמת הטבעיים ולכן רוב הפונקציות אינן ניתנות לחישוב. שים לב שמסקנה זו לא מתבססת על תכונות מסוימות של המודל של מכונת טיורינג אלא רק על עצם העובדה שכל המכונות ניתנות לתיאור סופי. באופן דומה ניתן היה להוכיח כי רוב השפות אינן ב-**RE** (וודאי אינן ב-**R** אשר מוכלת ב-**RE**).

9.2 הרחבה של הפונקציה האוניברסלית ובעית העצירה

מהנאמר בסעיף הקודם, ברור שרוב הפונקציות אינן ניתנות לחישוב, אך לא ניתנה דוגמא ספציפית של פונקציה שאינה ניתנת לחישוב (מי שעשה את התרגיל למתקדמים בפרק 8.2, כבר יודע על קיום פונקציה מסוימת שאינה ניתנת לחישוב, הרי היא סיבוכיות קולמגורוב). למעשה ניתן "לשים את היד" על פונקציה כזאת, ע"י תרגום של ההוכחה (שניתנה בפרק הרקע) שעוצמת הרצף (עוצמת הממשיים) גדולה ממש מעוצמת הטבעיים. למעשה אותו רעיון של הוכחה טמון בהוכחה שנציג בקרוב.

בפרק 8 הצגנו את הפונקציה האוניברסלית U והודגש שזו פונקציה חלקית המוגדרת על זוגות $(<M>,x)$, כאשר $<M>$ היא קידוד של מכונת טיורינג, x הינו קלט חוקי עבור M (כלומר, x הוא מחרוזת הנמצאת בא"ב הקלט) והמכונה M עוצרת על קלט x. באותם מקרים שאחד התנאים לא מתקיים, U לא מוגדרת על $(<M>,x)$. שים לב לכך ששני התנאים הראשונים קלים לבדיקה:

(1) קל לבדוק האם מחרוזת מהווה קידוד של מכונת טיורינג. למעשה מרגע זה ואילך נניח כי כל מחרוזת מהווה קידוד של מכונת טיורינג (מחרוזת שלא מהווה קידוד של מכונת טיורינג, תחשב בעינינו כקידוד של מכונה העוצרת מיד על כל קלט).

(2) קל גם לבדוק האם מחרוזת x הינה מחרוזת מעל א"ב הקלט של מכונה M. שוב ניתן יהיה להתייחס (באופן דומה) לכל מחרוזת כאילו היא מחרוזת מעל א"ב הקלט של M.

התנאי המהותי להיות U מוגדרת על $(<M>,x)$ הינו לפיכך שהמכונה M עוצרת על קלט x. העובדה שהפונקציה החלקית U ניתנת לחישוב, נבעה מכך שאותה מכונת טיורינג המחשבת את $U(<M>,x)$ לא נדרשה לזהות (בזמן סופי) את המקרים ש-M אינה עוצרת על x. באותם מקרים (ז.א. כאשר M לא עוצרת על x) המכונה האוניברסלית שבנינו אינה עוצרת. כפי שעוד נראה, תכונה זו של המכונה האוניברסלית היא מהותית, במובן שלא קיימת מכונת טיורינג העוצרת על כל קלט ומחשבת את הערך של U, באותם המקומות בהם היא מוגדרת.

נחזור על הרעיון שנית. למכונה אוניברסלית M_U (כלשהי) ישנן התכונות הבאות (הנובעות מהגדרתה כמחשבת את הפונקציה החלקית U):

(1) אם U מוגדר על $(<M>,x)$, אז M_U על קלט $(<M>,x)$ מחזירה את $f_M(x)$ (הפלט של M על קלט x).

(2) אם U אינו מוגדר על $(<M>,x)$ (כלומר, M אינה עוצרת על x) אז M_U אינה עוצרת על הקלט $(<M>,x)$.

השאלה שאנו שואלים את עצמנו כאן, היא האם קיימת מכונה M_U' בעלת התכונות הבאות:

(1) אם U מוגדר על $(<M>,x)$, אז M_U' מתנהגת על קלט $(<M>,x)$ בדיוק כמו M_U (כלומר, מחזירה את $f_M(x)$).

(2) M_U' עוצרת על כל קלט, ובפרט גם על קלטים שעליהם M_U אינה עוצרת (כלומר קלט $\langle M,x\rangle$ כך ש-M אינה עוצרת על x). במקרה זה אין כל דרישה לגבי הפלט.

התשובה לשאלה זו הינה שלילית. ניסוח פורמלי של תשובה זו ניתן במשפט הבא.

<u>משפט 9.1:</u> תהי $\overline{U}$ פונקציה מלאה כלשהי המרחיבה את הפונקציה U (ראה להלן הגדרה 9.1). הפונקציה $\overline{U}$ אינה ניתנת לחישוב.

<u>הגדרה 9.1:</u> פונקציה (מלאה) f, נקראת **הרחבה של פונקציה** (חלקית) g, אם עבור כל x עליו g מוגדרת, גם f מוגדרת ומתקיים: $f(x)=g(x)$.

שים לב כי $\overline{U}$ המוזכרת במשפט מקיימת $\overline{U}(\langle M,x\rangle)=U(\langle M,x\rangle)$ עבור כל $\langle M,x\rangle$ עליו U מוגדרת. כמו כן, שים לב כי המשפט גורר תשובה שלילית לשאלה "האם קיימת מכונה, M_T, העוצרת תמיד ומחשבת נכון את U באותן נקודות בהן U מוגדרת?" (פשוט משום שהפונקציה ש-M_T מחשבת, היא פונקציה מלאה המרחיבה את U). המשפט יוכח ע"י הבניה הבאה:

<u>בניה 9.1:</u> בהנתן פונקציה מלאה $\overline{U}$, המרחיבה את U, נגדיר פונקציה מלאה $\tilde{U}_1$ כדלקמן:

$$\tilde{U}_1(\langle M\rangle) = \overline{U}(\langle M\rangle,\langle M\rangle)+1$$

שים לב כי $\overline{U}$ היא פונקציה מלאה ובפרט מוגדרת על כל קלט מהצורה $\langle M\rangle,\langle M\rangle)$. הערך של $\tilde{U}_1(\langle M\rangle)$ יעניין אותנו בעיקר במקרה שהמכונה M עוצרת על הקלט $\langle M\rangle$. שים לב כי למרות שהשאלה, "מה הפלט של מכונה M כאשר נותנים לה כקלט את הקידוד של עצמה?", נראית מוזרה, היא לגיטימית. רעיון ההוכחה הוא שאם $\tilde{U}_1$ ניתנת לחישוב אז אנו ניקלע לסתירה בשאלינו את עצמנו, מה המכונה המחשבת את $\tilde{U}_1$ עושה בקבלה כקלט את הקידוד של עצמה. הוכחת במשפט תנבע משתי הטענות הבאות:

<u>טענה 9.3:</u> אם $\overline{U}$ ניתנת לחישוב, אז $\tilde{U}_1$ ניתנת לחישוב (טענה מסוג זה תקרא בשלב מאוחר יותר **רדוקציה**). ההוכחה לטענה זו מיידית מהגדרת $\tilde{U}_1$.

<u>טענה 9.4:</u> $\tilde{U}_1$ אינה ניתנת לחישוב.

<u>הוכחה:</u> נניח בשלילה כי $\tilde{U}_1$ ניתנת לחישוב ונסמן ב-M_1 את המכונה המחשבת אותה. כלומר, M_1, על קלט $\langle M\rangle$, מחזירה כפלט את $\tilde{U}_1(\langle M\rangle)$. כמו שנרמז, אנו גם נתעניין במה M_1 עושה על הקלט $\langle M_1\rangle$. שים לב כי M_1 חייבת לעצור על הקלט $\langle M_1\rangle$, משום ש-$\tilde{U}_1$ הינה פונקציה מלאה (אולם כפי שנראה מיד הערך ש-M_1 תיתן, במקרה זה,

יהיה מוטעה). נסמן ב-a את הפלט של M_1 על הקלט $<M_1>$. לעומת זאת, לפי הגדרת $\tilde{U}_1$ נקבל:

$$\tilde{U}_1(<M_1>) = \bar{U}(<M_1>,<M_1>)+1 = U(<M_1>,<M_1>)+1 = a+1.$$

השיוויון השני הוא לפי הגדרת $\bar{U}$ הזהה ל-U באותם מקומות בהם U מוגדרת. U מוגדרת על $(<M_1>,<M_1>)$ משום ש-M_1 עוצרת על $<M_1>$. השיוויון השלישי (משמאל) מתקיים לפי הגדרת U. קיבלנו כי הפלט של M_1 על קלט $<M_1>$ (שקראנו לו a) שונה מהערך של $\tilde{U}_1(<M_1>)$ (שהינו $a+1$) וכך קיבלנו סתירה להנחתנו המוטעית כי M_1 מחשבת את $\tilde{U}_1$. □

<u>דיון בהוכחה:</u> שים לב, שאין בהוכחה כל התייחסות לערך של $\bar{U}$ באותם מקומות בהם U אינה מוגדרת. לפיכך, אין חשיבות לאופן בו מרחיבים את U לפונקציה מלאה. בכל מקרה, פונקציה מלאה זו אינה ניתנת לחישוב. נעשה שימוש רק בעובדה שמכונה המחשבת את $\bar{U}$, חייבת לעצור על כל קלט. כמו כן, שים לב כי בהוכחה נעשה שימוש רק ב"אלכסון" של $\bar{U}$ (כלומר הערכים של $\bar{U}$ על קלטים מהסוג (x,x)). הוגדרה בכוונה תחילה כשונה מערכי האלכסון בכל נקודה. הדבר נעשה על מנת למנוע מכל מכונה לחשב את $\tilde{U}_1$, משום שמכונה M העוצרת על הקלט $<M>$ חייבת לתת עליו את הערך $\bar{U}(<M>,<M>)$ ולכן לא יכולה לתת את הערך $\tilde{U}_1(<M>)$ שהוגדר בכוונה תחילה כשונה. ניתן היה מלכתחילה להציג את הבניה של $\tilde{U}_1$ כך: מטרת הבניה היא להבטיח שכל מכונה M העוצרת על הקלט $<M>$, הפלט שלה שונה מ-$\tilde{U}_1(<M>)$. לשם כך נשתמש במניה של כל מכונות טיורינג (כאן נעשה שימוש רק בעובדה שכל מכונת טיורינג ניתנת לתיאור סופי). תהי M_i המכונה ה-i-ית במניה, נגדיר את $\tilde{U}(<M_i>)$ כערך כלשהו השונה מהפלט של M_i על הקלט $<M_i>$. די בכך על מנת להבטיח ש-M_i לא מחשבת את $\tilde{U}$ (כדי לחשב פונקציה, יש לחשב אותה נכון בכל נקודה). שים לב לדימיון בין צורת הצגה זו של ההוכחה, לבין ההוכחה שנתנה בפרק הרקע הטכני לטענה שעוצמת הממשיים גדולה מעוצמת הטבעיים. שים לב שהוכחת המשפט אינה מסתמכת על תכונות ספציפיות למכונות טיורינג ולפיכך תשאר תקפה עבור כל מודל סביר של חישוב (אשר בו לכל מכונה יש קידוד סופי).

קל לראות שאי-קיום מכונה המחשבת הרחבה כלשהי של U לפונקציה מלאה, גוררת אי-קיום מכונה אשר על קלט $(<M>,x)$ עונה האם M עוצרת על x או לא. בעיה זו נקראת **בעית העצירה** וניתן להציגה כחישוב הפונקציה המלאה הבאה:

$$h(<M>,x) = \begin{cases} 1 & M \text{ עוצרת על } x \\ 0 & M \text{ אינה עוצרת על } x \end{cases}.$$

<u>משפט 9.2:</u> הפונקציה h (שהוגדרה לעי"ל) אינה ניתנת לחישוב.

ניסוח פחות מסורבל שניתן להשתמש בו כאן ובעתיד, הוא ניסוח במונחים של "בעית הכרעה". במונחים אלו בעית העצירה היא, "בהנתן קלט $(<M>,x)$, האם M עוצרת על x?". המשפט אומר כי בעית העצירה אינה ניתנת להכרעה

(כלומר, אין מכונת טיורינג שעוצרת תמיד, אשר פותרת נכון את הבעיה על כל קלט).

<u>הוכחה:</u> נניח בשלילה כי הפונקציה h ניתנת לחישוב ונראה כי ניתן לחשב גם את ההרחבה הבאה של U לפונקציה מלאה, בסתירה למשפט 9.1. נגדיר את ההרחבה הבאה של U לפונקציה מלאה שתסומן $\overline{U}_0$:

$$\overline{U}_0(\langle M\rangle,x) = \begin{cases} U(\langle M\rangle,x) & M \text{ עוצרת על } x \\ 0 & \text{אחרת} \end{cases}.$$

תהי M_h מכונה המחשבת את הפונקציה h (כמובטח מהנחתנו) ותהי M_U מכונה אוניברסלית כלשהי. נבנה מכונה M_0 המחשבת את $\overline{U}_0$ כדלקמן: על קלט $\langle M\rangle,x)$, המכונה M_0 תריץ תחילה את M_h על אותו קלט, על מנת למצוא את הערך $h(\langle M\rangle,x)$. מובטח לנו כי M_h תעצור עם התשובה הנכונה. במקרה והתשובה היא 0 (M לא עוצרת על x) המכונה M_0 תוציא 0 כפלט ותעצור. במקרה והתשובה היא 1 (M עוצרת על x) המכונה M_0 תריץ את המכונה M_U על $\langle M\rangle,x)$ ותוציא את הפלט ש-M_U תתן (שים לב כי במקרה זה מובטח כי M_U תעצור ותוציא את הפלט הנכון $U(\langle M\rangle,x)$). □

שים לב, כי בהוכחת המשפט נעשה שימוש במה שאנו רוצים להוכיח את אי-קיומו, על מנת לבנות דבר שאנו כבר יודעים כי אינו קיים. הצלחת הבניה משמשת עדות לאי-קיום הדבר הראשון.

הוכחה <u>אלטרנטיבית</u> של משפט 9.2, מעתיקה תוך שינויים קלים את ההוכחה של משפט 9.1. מי שאינו זקוק ל"תרגול נוסף" מוזמן לדלג על שתי הטענות הבאות.

<u>טענה 9.5:</u> אם קיימת מכונה המחשבת את הפונקציה h, אז קיימת מכונה, $\overline{M}_h$, אשר לכל מכונה M, המכונה $\overline{M}_h$ עוצרת על קלט $\langle M\rangle$ אם ורק אם M לא עוצרת על $\langle M\rangle$.

<u>הוכחה:</u> בהנתן מכונה M_h המחשבת את h, נבנה את המכונה $\overline{M}_h$ המבוקשת כדלקמן: על קלט $\langle M\rangle$ המכונה $\overline{M}_h$ מריצה את M_h על הקלט $(\langle M\rangle,\langle M\rangle)$ ועוצרת אמם M_h מחזירה פלט 0 (כלומר, אם M לא עוצרת על $\langle M\rangle$). □

<u>טענה 9.6:</u> אין מכונה $\overline{M}_h$ אשר, עבור כל מכונה M, המכונה $\overline{M}_h$ עוצרת על קלט $\langle M\rangle$ אם ורק אם M אינה עוצרת על $\langle M\rangle$.

<u>הוכחה:</u> אם נניח קיום $\overline{M}_h$ כזו, נקלע לסתירה בבוחננו את השאלה "האם $\overline{M}_h$ תעצור על הקלט $\langle\overline{M}_h\rangle$ (הקידוד של עצמה)?". הדרישה הבלתי אפשרית שבטענה, מתורגמת במקרה זה ל-"$\overline{M}_h$ עוצרת על $\langle\overline{M}_h\rangle$ אמם $\overline{M}_h$ אינה עוצרת על $\langle\overline{M}_h\rangle$". □

<u>תרגיל 9.1:</u> הוכח כי השפה $L_U \triangleq \{\langle M\rangle,x) : x \text{ עוצרת על } M\}$ שייכת ל-**RE** \ **R** (רמז: הראה כי $L_U \in \mathbf{RE}$

נובע מקיום מכונות אוניברסליות, בעוד ש- $L_U \in \mathbf{R}$ יגרור קיום מכונה המחשבת את הפונקציה h ממשפט 9.2).

<u>תרגיל 9.2:</u> הראה כי ישנה שפה $L \in \mathbf{RE}$ כך ש- $\overline{L} \notin \mathbf{RE}$ (רמז: L_U).

<u>תרגיל 9.3:</u> הראה כי השפה L המוגדרת: $L \triangleq \{0y : y \notin L_U\} \bigcup \{1y : y \in L_U\}$, מקיימת, $L \notin \mathbf{RE}$ וגם $\overline{L} \notin \mathbf{RE}$ (רמז: הראה כי $L \in \mathbf{RE}$ גורר כי L_U (מתרגיל 9.1) שייכת ל-**R** ובאופן סימטרי עבור $\overline{L} \in \mathbf{RE}$).

<u>תרגיל 9.4:</u> הוכח כי קיימת מכונה M כך שהשפה $L_M \triangleq \{y : y$ עוצרת על $M\}$ אינה ב-**R**. האם היא ב-**RE**?

9.3 הוכחת נכונות של תכניות

מה שנאמר עד עתה בנושא בעית העצירה, דיו כדי להצביע על מגבלות מהותיות של הוכחת נכונות של תכניות. בפרק זה ננסח מגבלות אילו בצורה מפורשת יותר.

<u>משפט 9.3:</u> הבעיה הבאה אינה להכרעה: "בהנתן תיאור של מכונת טיורינג Π, האם Π עוצרת על כל קלט?".

ברור מכך שלא ניתן לבנות תכנית מחשב אשר תבדוק האם תכנית נתונה עוצרת על כל קלט.

<u>משפט 9.4:</u> הבעיה הבאה אינה להכרעה: "בהנתן תיאור של שתי מכונות טיורינג Π_1 ו-Π_2, האם מכונות אילו שקולות (כלומר, מחשבות את אותה פונקציה)?".

ברור מכך שלא ניתן לבנות תכנית מחשב אשר בהנתן כקלט ספסיפיקציה פורמלית של תכנית מחשב ותכנית מחשב, תקבע האם התכנית מתאימה לספסיפיקציה.

הוכחת שני המשפטים תעשה בדרך השלילה, תוך הסתמכות על כך שכבר ידוע לנו כי בעית העצירה אינה ניתנת להכרעה. מבנה ההוכחות יהיה כדלקמן: נניח בשלילה כי מכונה M, הסותרת את המשפט, אכן קיימת ונראה איך נוכל להשתמש במכונה זו על מנת לבנות מכונה M_h, המחשבת את הפונקציה h (בעית העצירה). המכונה M_h שנבנה תשתמש במכונה M כסברוטינה, תוך תרגום הקלט שלה לקלט עבור M. ריבוי המכונות במשפטים והשוני במעמדן, עלול לגרום בלבול לקורא. לא רק שישנן M_h ו-M, אלא ישנו גם הקלט של כל אחת מהן שאף הוא קידוד של מכונה (לשם הנוחיות נסמן מכונות הניתנות כקידוד ב-Π). אולם מעבר לבלבול הנגרם ע"י ריבוי המכונות, אין בהוכחות אלו קושי מהותי.

<u>הוכחת משפט 9.3:</u> נניח <u>בשלילה</u> כי קיימת מכונה M_1 הפותרת את בעית ההכרעה שבמשפט. נראה כי ניתן לבנות מכונה M_h המחשבת את הפונקציה h. על קלט $(x,\langle\Pi\rangle)$, המכונה M_h תבנה קידוד של מכונה Π_x העוצרת על כל קלט אסם Π עוצרת על x. M_h תתן את הקידוד של Π_x ל-M_1 ותתנהג כמותה (כלומר תחליט בחיוב אסם M_1 החליטה בחיוב). בציור 9.1 ישנה הצגה סכמטית של המכונה M_h. תת המכונה T מתרגמת את הזוג $(x,\langle\Pi\rangle)$ לקידוד של מכונה $\langle\Pi_x\rangle$.

ציור 9.1 המכונה M_h

מהי אותה מכונה Π_x שאת קידודה M_h מייצרת? האם קיימת בכלל מכונה כזו? זכור, ברצוננו ש-Π_x תעצור על כל קלט אסם Π עוצרת על x. להלן תיאור אפשרי של Π_x: המכונה Π_x מתעלמת מן הקלט שלה ומבצעת סימולציה של ריצת Π על הקלט x. במפורט, על קלט y, המכונה Π_x מוחקת את הקלט y וכותבת במקומו את x. לבסוף, Π_x מבצעת סימולציה של ריצת Π על x. בציור 9.2 ישנה הצגה סכמטית של המכונה Π_x.

ציור 9.2 המכונה Π_x

ראינו כי קיימת מכונה Π_x כזו, אך האם M_h יכולה ליצור את הקידוד שלה (כלומר את $\langle\Pi_x\rangle$) מתוך $(x,\langle\Pi\rangle)$? גם לכך התשובה חיובית. הקידוד של Π_x יורכב מפקודות למחיקת קלט (מספיקים לשם כך שני מצבים וקידוד קבוע) פקודות לצורך הדפסת $(x,\langle\Pi\rangle)$ על גבי הסרט (ניתן לבצע זאת בעזרת $O(|x,\langle\Pi\rangle|)$ מצבים) ולבסוף פעולות של מכונה אוניברסלית. יצירת קידוד פעולות אלו, יכולה להעשות בצורה אוטומטית מתוך $\langle\Pi\rangle$ ו-x. שים לב כי M_h אינה מריצה (בשום אופן!) את Π על x, אלא מבצעת מעין "קומפילציה" של $\langle\Pi\rangle$ ו-x לתכנית $\langle\Pi_x\rangle$. לאחר שהובנו כל פרטי ההוכחה, וודא את נכונות הטענות הבאות:

(1) ישנה מכונת טיורינג אשר על קלט $(x,\langle\Pi\rangle)$ יוצרת קידוד $\langle\Pi_x\rangle$ כמתואר לעי"ל.

(2) Π_x אמנם עוצרת על כל קלט אשם Π עוצרת על x.

(3) בהנחה שקיימת מכונה M_1 (המכריעה את הבעיה שבמשפט 9.3) אז שילובה עם המכונה מסעיף (1) יוצרת מכונת טיורינג M_h אשר (בהתבסס על סעיף (2)) פותרת את בעית העצירה (המכונה M_h, על קלט $\langle x,\langle\Pi\rangle\rangle$, יוצרת קידוד, $\langle\Pi_x\rangle$ ומחזירה את הפלט של M_1 על קלט $\langle\Pi_x\rangle$) וזאת בסתירה לכך שבעיה זו אינה ניתנת להכרעה.

הבעיה בהבנת ההוכחה נובעת מהמעמד השונה של המכונות המוזכרות. המכונות (ההיפוטטיות) M_1 ו-M_h, הן מכונות המנסות להכריע שאלות לגבי מכונות אחרות (Π_x ו-Π בהתאמה). לשם כך הן אינן מריצות אותן, אלא "בוחנות" אותן בלבד. ה"בחינה" של M_h את $\langle\Pi\rangle$, כוללת קומפילציה (ולא הרצה) של $\langle\Pi\rangle$ למכונה חדשה, שקידודה ($\langle\Pi_x\rangle$) נמסר כקלט למכונה M_1. □

<u>הערה למתקדמים:</u> שים לב כי ממבנה ההוכחה נובע כי אי אפשר אפילו להכריע האם מכונה עוצרת על כל קלט או אינה עוצרת על אף קלט.

<u>הוכחת משפט 9.4:</u> ההוכחה מסתמכת על הוכחת משפט 9.3. נראה שאם ניתן להכריע האם שתי מכונות טיורינג הינן שקולות, אז ניתן להכריע האם מכונת טיורינג נתונה עוצרת על כל קלט (בעיה שידוע לנו, ממשפט 9.3, כי אינה ניתנת להכרעה). נניח בשלילה קיום מכונה M_2 בניגוד לטענת משפט 9.4 ונבנה מכונה M_1 בניגוד לטענת משפט 9.3 (שכבר הוכחנו את נכונותו). המכונה M_1, בהנתן כקלט קידוד של מכונה $\langle\Pi\rangle$ (שלגביו יש להכריע האם Π עוצרת על כל קלט) בונה זוג מכונות Π_1 ו-Π_2 השקולות זו לזו אשם Π עוצרת על כל קלט. בניה אפשרית היא ש-Π_1 היא מכונה קבועה (ופשוטה) המחשבת למשל, את פונקצית הזהות ($\forall x\ f(x)=x$) ואילו Π_2 על קלט x מריצה תחילה את Π על x (על סרט אחר) <u>ואחר-כך</u> ללא קשר לתוצאת החישוב של Π על x, מחשבת את פונקצית הזהות. בציור 9.3 ישנה הצגה סכמטית של המכונה M_1 (T הינה תת מכונה המתרגמת את הקלט $\langle\Pi\rangle$ לזוג קידודים $(\langle id\rangle,\langle id_\Pi\rangle)$.

ציור 9.3 המכונה M_1

בציור 9.4 ישנה הצגה סכמטית של המכונה id_Π.

ציור 9.4 המכונה id_{Π} (המכונה id מופעלת לאחר שהמכונה Π עוצרת)

ודע את הטענות הבאות:

(1) ישנה מכונת טיורינג אשר על קלט <Π> מייצרת זוג <Π_1>,<Π_2> כמתואר לעי"ל.

(2) הזוג Π_1 ו-Π_2 מחשב את אותה פונקציה אם Π עוצרת על כל קלט.

(3) אם אמנם קיימת מכונה M_2, אז שילובה עם המכונה מסעיף (1), יוצרת מכונה, M_1, הסותרת את משפט 9.3 (המכונה M_1, על קלט <Π>, יוצרת קידודים <Π_1>,<Π_2> ומחזירה את הפלט של M_2 על הקלט (<Π_1>,<Π_2>)).

9.4 משפט רייס (Rice)

במשפט 9.3 ראינו דוגמא לבעיה שאינה ניתנת להכרעה הנוגעת לפלט של מכונת טיורינג שקידודה נתון. זהו מקרה פרטי של העובדה הכללית, האומרת שלא ניתן להכריע אף שאלה <u>לא טריביאלית</u> הנוגעת לפלט של מכונת טיורינג נתונה. על מנת לנסח את העובדה הכללית הזאת, נבהיר שני מושגים:

(1) מהן שאלות הנוגעות לפלט של מכונת טיורינג?

שאלות הנוגעות לפלט של מכונת טיורינג הן שאלות המתייחסות לפונקציה שהמכונה מחשבת ולה בלבד. דוגמאות לשאלות כאלה הינן: "האם הפונקציה שהמכונה מחשבת הינה מלאה?" (כלומר, האם המכונה עוצרת תמיד) "האם הפונקציה קבועה?", "האם הערך שלה על המחרוזת 01 הינו 111?" וכו'...

דוגמאות לשאלות שאינן מתייחסות (רק) לפונקציה שהמכונה מחשבת הינן: "מה מספר המצבים של המכונה הנתונה?", "האם המכונה עושה מספר זוגי של צעדים על כל קלט?" וכו'...

(2) מהן שאלות לא טריביאליות מסוג (1)?

שאלות הנוגעות לפונקציה שהמכונה מחשבת, יכולות להיות טריביאליות במובן שהתשובה עליהן אחידה לגבי כל פונקציה הניתנת לחישוב. למשל, "האם הפונקציה ניתנת לחישוב?", "האם ערכה בנקודה 01 שונה

מערכה בנקודה 01?" וכו'... שאלות לא טריביאליות, מחלקות את הפונקציות הניתנות לחישוב, לשתי תתי קבוצות (לא ריקות): פונקציות שלגביהן התנאי מתקיים ופונקציות שלגביהן התנאי אינו מתקיים.

<u>משפט רייס (ניסוח פונקציונלי):</u> תהא F_1 ו-F_2 חלוקה כלשהי, של קבוצת הפונקציות הניתנות לחישוב, לשתי תתי-קבוצות (זרות) לא ריקות. אז, בהנתן קידוד של מכונה (<Π>) לא ניתן להכריע האם הפונקציה שהמכונה מחשבת (כלומר, f_{Π}) הינה ב-F_1 או ב-F_2.

<u>משפט רייס (ניסוח במונחי שפות):</u> תהא C תת-קבוצה לא טריביאלית של **RE** (כלומר, $C \subseteq \mathbf{RE}$ וגם $\varnothing \neq C \neq \mathbf{RE}$). אז, השפה הבאה אינה ניתנת להכרעה (אינה ב-**R**),

$$L_C \triangleq \{<\Pi> : L(\Pi) \in C\}$$

כאשר $L(\Pi)$ הינה השפה שהמכונה Π מקבלת.

שים לב כי כאשר $C = \mathbf{RE}$ או $C = \varnothing$, המכונה המקבלת כל קלט והמכונה הדוחה כל קלט מכריעות בהתאמה את השפה L_C.

הוכחת המשפט תסתמך על כך שכבר הוכחנו כי השפה $L_U \triangleq \{(<\Pi>,x) : x$ **עוצרת על הקלט** $\Pi\}$ אינה ב-R.

<u>הוכחת משפט רייס (בניסוח במונחי שפות):</u> נניח ללא הגבלת הכלליות כי $\varnothing \notin C$ (השפה הריקה אינה נמצאת במחלקה C). אחרת נסתכל ב-$\mathbf{RE} \setminus C$, מה שיוכיח את הטענה הזהה כי השפה $\{<\Pi> : L(\Pi) \notin C\}$ (שהיא המשלימה של L_C) אינה ב-**R**. תהא $L_0 \neq \varnothing$ שפה כלשהי ב-C (קיום שפה כזו מובטח מהיות C תת-קבוצה לא טריביאלית של **RE**) ונסמן ב-Π_0 את המכונה המקבלת אותה (שים לב לכך שהשפה L_0 והמכונה Π_0 קבועות). נניח בשלילה כי קיימת מכונה M_C העוצרת תמיד ומקבלת את הקלט <Π> אם $L(\Pi) \in C$. נראה כי ניתן יהיה לבנות מכונה M_U, אשר עוצרת תמיד ומקבלת את השפה L_U (מתרגיל 9.1). המכונה M_U תקבל כקלט זוג $(<\Pi>,x)$ לגביו היא צריכה להכריע האם Π עוצרת על x. המכונה M_U, תבנה קידוד של מכונה $<\Pi_x>$ אשר תקבל את L_0, אם Π עוצרת על x ואת השפה הריקה, אם Π אינה עוצרת על x. שים לב כי $L(\Pi_x) \in C$ אם Π עוצרת על x.

מהי אותה Π_x? Π_x תהיה דומה ל-Π_x מהוכחת משפט 9.3, רק שהיא לא תמחק את הקלט המקורי שלה (y) אלא תשמור אותו. במקרה שהסימולציה של Π על x תסתיים, המכונה Π_x תריץ את Π_0 על y ותנהג כמותה. אחרת (הסימולציה של Π על x לא מסתיימת) Π_x תשאר תקועה בלולאה אינסופית על כל y (ולכן לא תקבל אף y). ודא כי הינך מבין את הבניה וכן ודא:

(1) ישנה מכונה שעל קלט $(<\Pi>,x)$ מוציאה פלט $<\Pi_x>$ כנדרש.

(2) $L(\Pi_x) \in C$ אם Π עוצרת על x.

(3) שילוב של המכונה המתוארת בסעיף (1) עם המכונה M_C, יוצר מכונה M_U המכריעה את בעית העצירה וזוהי סתירה להנחת קיומה של M_C (המכונה M_U, על קלט, $(<\Pi>,x)$, יוצרת קידוד, $<\Pi_x>$ ומחזירה את

הפלט של M_C על $<\Pi_x>$).

<u>תרגיל 9.5:</u> הוכח באמצעות שימוש במשפט רייס, כי השפות הבאות אינן ב-**R**.

א. $\{<M> : L(M)=\emptyset\}$

ב. $\{<M> : \lambda \in L(M)\}$

ג. $\{<M> : |L(M)|<100\}$

ד. $\{<M> : L(M)$ מכילה רק מילים מאורך זוגי$\}$

אם הינך זקוק לתרגול נוסף, נסה להוכיח טענות אלו "ישירות", תוך שימוש ברעיונות שבהוכחת משפט רייס.

<u>תרגיל 9.6:</u> האם השפות הבאות הינן ב-**R**?

א. $\{<M> : L(M) \in \mathbf{RE} \setminus \mathbf{R}\}$

ב. $\{<M> :$ ישנה העתקה חח"ע ועל בין $L(M)$ לבין הממשיים$\}$

ג. $\{<M> :$ ישנה העתקה חח"ע ועל בין $L(M)$ לבין הרציונליים$\}$

ד. $\{<M> : L(M)=L(M) \circ L(M)\}$

ה. $\{<M> : L(M)=L(M)^*\}$

ו. $\{<M> : L(M)^*=\Sigma^*\}$

ז. $\{<M> :$ קיים קבוע c כך ש-M עוצרת (דוחה או מקבלת) על כל קלט תוך c צעדים$\}$

ח. $S_c = \{<M> : M$ עוצרת (דוחה או מקבלת) על כל קלט תוך c צעדים$\}$

<u>תרגיל 9.7:</u> הוכח את משפט רייס בניסוחו הפונקציונלי.

<u>תרגיל 9.8:</u> אלו מהשפות הבאות הינן ב-**RE**, ואלו הן בעלות משלים ב-**RE**?

א. $\{<M> : L(M)=\Sigma^*\}$

ב. $\{<M> : L(M)=\emptyset\}$

ג. $\{<M> : |\Sigma^* \setminus L(M)|<100\}$

ד. $\{<M> : |L(M)|<100\}$

ה. L_C לגבי C לא טריביאלי כלשהו (אבחן מספר מקרים, אך אל תנסה לתת אפיון מלא).