7. קבלת שפות ע"י מכונות טיורינג ונושאים הקשורים בכך

7.1 קבלת שפות והקשר לחישוב פונקציות

תכליתם של מחשבים היא לחשב פונקציות מן הקלטים האפשריים, לפלטים הרצויים. לפיכך הגדרנו מכונות טיורינג כמחשבות פונקציות. אולם, תכליתנו בקורס זה אינה חישוב ותכנות באופן ממשי, אלא חקירת מהות הבעיות הניתנות לפתרון ע"י תכנית מחשב (כלומר, הפונקציות הניתנות לחישוב). כפי שנראה בעתיד, הדיון בפונקציות הניתנות לחישוב מסורבל יחסית. יש להבחין בין פונקציות מלאות לחלקיות, ויש לקחת בחשבון את כל הערכים שהפונקציה יכולה לקבל, שהרי הפונקציה היא העתקה ממחרוזות אל מחרוזות ($f:\{0,1\}^*\to\{0,1\}^*$). מטרתנו בסעיף זה להראות כי הדיון המסורבל הזה ממוצה ע"י דיון מסורבל פחות אך טבעי פחות (לכאורה) הדן ב"קבלת שפות ע"י מכונת טיורינג".

כמובן פשוט יותר לדון בפונקציות בולאניות (מלאות או חלקיות) כלומר פונקציות $f:\{0,1\}^*\to\{0,1\}$. נצטרך לראות שהדיון בחישוב פונקציות בולאניות (ז.א. המקרה הפרטי) אמנם ממצה את הדיון בחישוב פונקציות כלשהן (המקרה הכללי).

<u>בניה 7.1:</u> בהנתן פונקציה חלקית או מלאה $f:\{0,1\}^*\to\{0,1\}^*$, נגדיר את הפונקציה הבוליאנית הבאה המוגדרת רק על קלטים ש- f מוגדרת עליהם:

$$F_f(x,y)\triangleq\begin{cases}1 & y\in\mathbf{pref}(f(x))\\ 0 & y\notin\mathbf{pref}(f(x))\end{cases}$$

כאשר **pref(x)** היא קבוצת כל הפרפיקסים של x. לכל x,y הפונקציה F_f לא מוגדרת על (x,y) אם f אינה מוגדרת על x. זכור כי y הוא פרפיקס (רישא) של ζ אם קיים $\alpha\in\{0,1\}^*$ (יתכן ריק!) כך ש- $\zeta=y\alpha$.

<u>משפט 7.1:</u> לכל פונקציה f, קיימת מכונת טיורינג המחשבת את f אם ורק אם קיימת מכונת טיורינג המחשבת את F_f (המוגדרת בבניה 7.1).

<u>הוכחה:</u> נניח תחילה כי קיימת מכונת טיורינג (בעלת סרט יחיד) M המחשבת את f, ונבנה מכונה רבת-סרטים M' המחשבת את F_f. המכונה M' פועלת כדלקמן (על קלט $x\#y$): M' מעתיקה את x לסרט השני ומריצה את M עליו. כאשר (ובמידה ו) M עוצרת על הסרט השני, המכונה M' משווה את הפלט של M למחרוזת y וקובעת את הפלט שלה בהתאם. שים לב כי אם M לא עצרה על x (ז.א. f לא מוגדרת על x) המכונה M' לא תעצור על (x,y). מסקנה – M' מחשבת את F_f כהלכה.

נניח כעת כי קיימת מכונה (בעלת סרט יחיד) M המחשבת את F_f, ונבנה מכונה רבת-סרטים M' המחשבת את f. על קלט x , המכונה M' פועלת כדלקמן: M' מעתיקה את x אל הסרט השני וכותבת מימין לו סימן # יחיד. M' מבצעת איטרציות עד לעצירה על-פי תנאי שיפורט להלן. בכל איטרציה M' מבצעת שני "נסיונות", האחד נקרא

ניסוי-0 והשני ניסוי-1. בניסוי-σ ($\sigma \in \{0,1\}$) M' מעתיקה את תוכן הסרט השני (באיטרציה הראשונה זהו $x\#$ אך באיטרציות הבאות יהיו אלו $x\#y'$ כלשהם) אל הסרט השלישי, וכותבת מימין לו את הביט σ. המכונה M' מריצה את המכונה M על הסרט השלישי וזוכרת את התוצאה (שים לב כי כאשר M עוצרת, תוצאת החישוב היא $F_f(x,y'\sigma)$). כמו כן, שים לב כי אם f מוגדר על x, המכונה M תעצור על (x,y) עבור כל y, וכן כי במקרה זה מתקיים $F_f(x,y'0)=0$ או $F_f(x,y'1)=0$. אם באחד הניסויים, נאמר בניסוי-σ, תוצאת החישוב של M היתה 1 (משמע כי $y'\sigma$ הוא פרפיקס של $f(x)$) המכונה M' מוסיפה את σ אל תוכן הסרט השני ופונה לאיטרציה הבאה. אם בשני הניסויים, תוצאת החישוב של M היתה 0 (משמע כי $y'=f(x)$) המכונה M' מעתיקה את y' הנוכחי מן הסרט השני אל סרט הפלט ועוצרת. ניתן לראות כי M' מחשבת את f כהלכה. □

<u>הערה:</u> בהוכחת המשפט בנינו מכונה רבת-סרטים M', המריצה מכונה אחרת M על אחד מסרטיה. הסתפקנו בכך שאמרנו "נבנה מכונה כזאת" ולא תארנו את פעולתה במפורט. ודא שאתה יודע איך פועלת M', ובפרט שהינך מסוגל, בהנתן מכונה M, לבנות את המכונה M' המתאימה לה (בהרבה עמל אך מעט מחשבה).

במידה והפונקציה f מלאה, הרי גם הפונקציה F_f שבנינו מלאה. במקרה כזה, ה"מעמד" של הקלט $\zeta = x\#y$ ביחס לפונקציה F_f נקבע ע"י ביט יחיד (כלומר 0 או 1). מתבקש ממש להגדיר את קבוצת הקלטים שעליהם ערך הפונקציה הוא 1. קבוצה זו היא שפה (מעל הא"ב $\{0,1\}$) המוגדרת כדלקמן: $L \triangleq \{\zeta : F_f(\zeta)=1\}$. השפה המשלימה $\overline{L} \triangleq \{0,1\}^* \setminus L$, היא קבוצת כל המילים שערך הפונקציה עליהם הוא 0. במקרה הכללי בו f אינה בהכרח מלאה (וכך גם F_f) עדיין ניתן להגדיר את השפה L (קבוצת כל המילים בהן ערך הפונקציה F_f מוגדר ושווה 1) אולם השפה המשלימה $\overline{L}$ תכיל מילים משני סוגים:

(1) מילים שעליהם ערך הפונקציה F_f מוגדר ושווה ל-0.

(2) מילים שעליהם ערך הפונקציה F_f אינו מוגדר.

כל הנאמר בפסקה הנ"ל תקף לגבי כל פונקציה בולאנית $g : \{0,1\}^* \to \{0,1\}$ אשר הינה חלקית או מלאה. מטרתנו כעת היא לפתח הגדרה של "זיהוי שפות ע"י מכונת טיורינג" אשר תאפשר טיפול נוח בשאלת החישוב של פונקציות בולאניות מלאות או חלקיות.

<u>הגדרה 7.1:</u> **מכונת טיורינג לזיהוי שפות**, היא מכונת טיורינג בעלת שני מצבים סופיים $F \triangleq \{q_{ACC}, q_{REJ}\}$.

<u>הערה:</u> שים לב שבשלב זה לשמות של המצבים הסופיים אין שום משמעות! אך בעתיד יקבלו מצבים אילו משמעות כאשר תוגדר קבלה ודחיה ע"י מכונת טיורינג.

<u>תרגיל 7.1:</u> הראה כי לכל מכונת טיורינג קיימת מכונת טיורינג שקולה (ז.א. מחשבת אותה פונקציה) בעלת שני מצבים סופיים (כלומר שתוכל לקיים את הגדרה 7.1).

מנקודה זו בקורס ועד לסיומו נניח כי כל מכונות הטיורינג בהן מדובר הן מכונות לזיהוי שפות (ז.א. יש להן בדיוק שני מצבים סופיים).

הגדרה 7.2: נאמר כי מכונת טיורינג M **מקבלת** את הקלט x, אם קיימים $\beta,\gamma\in\Gamma^*$ כך ש- $q_0x \overset{*}{\underset{M}{\vdash}} \beta q_{ACC}\gamma$.

הערה: המכונה M מקבלת את הקלט x אם החישוב שלה המתחיל עם קלט x מסתיים במצב q_{ACC} (שים לב ACC נבחר כקיצור ל Accept – קבלה בלעז). מכונה M יכולה לא לקבל את הקלט x ע"י עצירה במצב q_{REJ} (שים לב כי REJ הוא קיצור ל Reject) או ע"י כך שהיא לא עוצרת כלל על x.

הגדרה 7.3: נאמר כי מכונת טיורינג M **דוחה** את הקלט x, אם קיימים $\beta,\gamma\in\Gamma^*$ כך ש- $q_0x \overset{*}{\underset{M}{\vdash}} \beta q_{REJ}\gamma$.

הערה: שוב נדגיש כי אם M לא מקבלת את x אין משמעו שהיא בהכרח דוחה את x. יתכן שאינה עוצרת על x (כלומר ישנן שלוש אפשרויות!). כאן במיוחד יש להזהר כנגד טרמינולוגיות של מחברים אחרים!

הגדרה 7.4: **השפה המתקבלת** ע"י מכונת טיורינג M היא קבוצת המילים אשר M מקבלת.

ובמפורש: $L_M \triangleq \{x : \exists\beta,\gamma\in\Gamma^* \ s.t. \ q_0x \overset{*}{\underset{M}{\vdash}} \beta q_{ACC}\gamma\}$.

הערה: בהגדרה 7.4 אין דרישה ש- M תעצור על כל קלט וגם אין דרישה מנוגדת ש- M תעצור רק על הקלטים ב- L_M. כל מה שנדרש (במובלע) ביחס לקבוצת הקלטים עליהם M עוצרת, הוא שקבוצה זו מכילה את L_M.

תרגיל 7.2: הראה כי לכל מכונת טיורינג, M, קיימת מכונת טיורינג, M', המקבלת את L_M ועוצרת רק על קלטים ב- L_M.

הערה: לפי תרגיל 7.2 יכולנו לשם פשטות ההגדרה, להגדיר שהמכונה מקבלת אם ורק אם היא עוצרת. אך, ההתעקשות שהמכונה תעצור אך ורק על מילים שהיא מקבלת, נראית לנו מלאכותית. בייחוד מכיוון שיש ערך מיוחד למכונות העוצרות תמיד ולא רק על קלטים השייכים לשפה.

דוגמאות:

[1] השפה Σ^* (כלומר השפה המכילה את כל המילים מעל הא"ב $\Sigma = \{0,1\}$) מתקבלת ע"י המכונה בעלת פונקצית המעברים: $\delta(q_0,0) = (q_{ACC},0,S)$ ו- $\delta(q_0,1) = (q_{ACC},1,S)$.

[2] השפה הריקה $\varnothing$ מתקבלת ע"י המכונה בעלת פונקצית המעברים: $\delta(q_0,0) = (q_{REJ},0,S)$ ו- $\delta(q_0,1) = (q_{REJ},1,S)$.

[3] השפה המקבלת את כל המחרוזות המתחילות ב-111 מתקבלת ע"י המכונה בעלת פונקצית המעברים:

$\delta(q_0,0) = (q_{REJ},0,S)$ - דחיית מילים שהאות הראשונה בהם היא 0.

$\delta(q_0,1) = (q_1,1,R)$ - המשך החישוב על מילים שהאות הראשונה בהם היא 1.

$\delta(q_1,0) = (q_{REJ},0,S)$ - דחיית מילים שהאות השניה בהם היא 0.

$\delta(q_1,1) = (q_2,1,R)$ - המשך החישוב על מילים שהאות השניה בהם היא 1.

$\delta(q_2,0) = (q_{REJ},0,S)$ - דחיית מילים שהאות השלישית בהם היא 0.

$\delta(q_2,1) = (q_{ACC},1,S)$ - קבלת מילים המתחילות ב- 111.

כעת נראה כי הדיון בקבלת שפות ממצה את הדיון בחישוב של פונקציות בולאניות בין אם הן מלאות או חלקיות. תוצאה זו מצדיקה את הדיון בנושא המוזר לכאורה של קבלת שפות ע"י מכונות טיורינג, שיחליף לעיתים את הדיון המעניין בפונקציות המחושבות ע"י מכונות אלו. לצורך זה נגדיר לכל פונקציה בולאנית $g : \{0,1\}^* \to \{0,1\}$ (חלקית או מלאה) שפה כדלקמן:

<u>בניה 7.2:</u> $L_g \triangleq \{(x,\sigma) : \sigma$ וערכו g מוגדר על $x\}$

<u>משפט 7.2:</u> תהי g פונקציה בולאנית מלאה, אז g ניתנת לחישוב אם ורק אם L_g מתקבלת ע"י מכונת טיורינג העוצרת על כל קלט.

<u>משפט 7.3:</u> תהי g פונקציה בולאנית כלשהי (חלקית או מלאה) אז g ניתנת לחישוב אם ורק אם L_g מתקבלת ע"י מכונת טיורינג.

<u>הוכחת משפט 7.2:</u>

<u>כיוון I:</u> נניח כי g מלאה וניתנת לחישוב ע"י מכונת טיורינג M בעלת סרט יחיד. נבנה מכונה רבת-סרטים M' המקבלת את L_g. המכונה M' תפעל כדלקמן (על קלט $x\#\sigma$): M' תעתיק את x לסרט השני, תריץ את M על הסרט השני ותשווה את הפלט של M לביט σ. אם מתקבל שיוויון, M' תעבור **למצב קבלה**. אחרת, M' תעבור **למצב דחיה** (ותעצור בשני המקרים). מכיוון ש-g מוגדר על כל x מובטח כי M תעצור תמיד, וקל לראות כי M' אמנם מקבלת את L_g ועוצרת תמיד.

<u>כיוון II:</u> נניח כעת כי g מלאה וקיימת מכונת טיורינג M (בעלת סרט יחיד) המקבלת את L_g ועוצרת תמיד. נבנה מכונה (רבת סרטים) M' המחשבת את g. על קלט x, המכונה M' תעתיק את x לסרט השני ותכתוב מימינו את המחרוזת $1\#$. המכונה M' תריץ את המכונה M על הסרט השני שתכנו $x\#1$ (שים לב כי המכונה M עוצרת

תמיד). אם M מקבלת את $x\#1$ משמע כי $g(x)=1$. אחרת (M דוחה את $x\#1$) משמע כי $g(x)=0$. המכונה M' תוציא את הפלט המתאים ותעצור. □

<u>הוכחת משפט 7.3:</u>

<u>כיוון I:</u> נניח כי g (חלקית או מלאה) ניתנת לחישוב ע"י מכונת טיורינג M בעלת סרט יחיד. נבנה מכונת טיורינג רבת-סרטים M' המקבלת את L_g. כיוון זה דומה לכיוון המתאים בהוכחת משפט 7.2. הבניה של M' זהה, אולם כעת יתכן ש-M לא תעצור על הקלט x ובמקרה זה גם M' לא תעצור על $x\#\sigma$ (זה בסדר משום שבמקרה בו M לא עוצרת על x, משמע כי g לא מוגדרת על x ולכן $(x,\sigma)\notin L_g$ לכל ערך $\sigma\in\{0,1\}$). במידה ו-M תעצור על הקלט x, תוכל M' לעצור על $x\#\sigma$ ולקבל או לדחות לפי הצורך.

<u>כיוון II:</u> בכיוון השני יש להזהר! כעת נתון כי g חלקית או מלאה, וקיימת מכונת טיורינג M (בעלת סרט יחיד) המקבלת את L_g ואינה בהכרח עוצרת תמיד. אם נחזור על הבניה מהוכחת משפט 7.2 אנו עלולים להתקל בבעיה. זאת משום שאם $g(x)=0$ הרי $x\#1\notin L_g$ ויתכן כי M לא תעצור על $x\#1$ ולכן M' (כפי שהוצגה בהוכחת משפט 7.2) לא תעצור על x. המוצא הוא באבחנה כי במקרה ש g מוגדרת על x הרי $x\#0$ או $x\#1$ שייך ל- L_g והמכונה M חייבת לעצור על אותה מילה שבשפה L_g. לפיכך נתקן את הבניה של M' אשר תפעל כדלקמן: על קלט x, המכונה M' תעתיק את x לסרט השני, תרשום מימינו $0\#$ וכן תעתיק את x לסרט השלישי ותרשום מימינו $1\#$. המכונה M' תריץ שני עותקים של המכונה M "במקביל": עותק אחד על הסרט השני ועותק שני על הסרט השלישי (הכוונה בהרצה במקביל היא ש-M' תבצע לסרוגין צעד חישוב אחד של M על הסרט השני, וצעד חישוב אחד של M על הסרט השלישי). כאשר אחד העותקים של M יעצור ויקבל את המחרוזת שלו, כלומר כאשר העותק העובד על הסרט ה $(\sigma+2)$-י יקבל את המחרוזת $x\#\sigma$ ($\sigma\in\{0,1\}$) המכונה M' תכתוב את σ כפלט שלה ותעצור. שים לב כי אם g מוגדר על x הרי שהמאורע המדובר יתרחש לאחר מספר סופי של צעדים (כי $x\#g(x)\in L_g$) ואילו אם g אינו מוגדר על x הרי שהמאורע לא יתרחש והמכונה M' לא תעצור על x. □

<u>הערה למתקדמים:</u> אין כמובן חשיבות לכך שבהרצה במקביל של שני עותקים של M מבצעים צעד אחד לסירוגין. מה שחשוב הוא שמבצעים לסרוגין מספר סופי של צעדים על כל עותק, כך שאם אחד העותקים של M, עוצר לאחר מספר סופי של צעדים, M' תעצור אחרי מספר סופי של צעדים.

<u>הערה נוספת:</u> מטעמים פדגוגיים הופרדה הטרנספורמציה של דיון בחישוב פונקציות לדיון בקבלת שפות לשני שלבים המוצגים בבניות 7.1 ו 7.2. ניתן כמובן להרכיב את שתי הבניות לבניה אחת המתאימה לכל פונקציה (חלקית או מלאה) $f:\{0,1\}^*\to\{0,1\}^*$ שפה L_f כדלקמן:

$$L_f \triangleq \{(x,y,\sigma) : \sigma=\mathbf{pref}(y,f(x)) \text{ ומתקיים } f \text{ מוגדרת על } x\},$$

כאשר $\mathbf{pref}(y,\zeta)$ הוא 1 אם y הוא פרפיקס של ζ (ובפרט אם $y=\zeta$) ו- 0 אחרת. התפקיד היחידי של המשתנה השלישי הוא לאפשר לזהות את המקרה ש-f מוגדר על x אך y אינו פרפיקס של $f(x)$ אלא הארכה שלו (ז.א.

$y = f(x)y'$). שים לב לתפקיד של מקרה זה בהוכחה של משפט 7.1. כדי לזהות שאמנם קיבלנו את כל הביטים של $f(x)$, בדקנו כי גם $f(x)0$ וגם $f(x)1$ אינם פרפיקסים של $f(x)$. בניה שונה במקצת מוצגת להלן:

<u>בניה 7.3:</u> לכל פונקציה (חלקית או מלאה) $f : \{0,1\}^* \to \{0,1\}^*$ נתאים את השפה הבאה:

$L_f \triangleq \{(x,y) : f(x)$ מוגדרת על x והמילה y היא פרפיקס ממש של $f(x)\}$

$\cup \{(x,y\#) : y-$ ל זהה וערכה x על מוגדרת $f\}$

באמרנו כי y היא פרפיקס ממש של ζ הכוונה כי $y \neq \zeta$ וקיים y' כך ש- $yy' = \zeta$.

<u>הגדרה 7.5:</u> מחלקת השפות **הניתנות-למניה-רקורסיבית**, המסומנת **RE** (Recursive Enumerable), היא קבוצת השפות המתקבלות ע"י מכונות טיורינג.

$$\mathbf{RE} \triangleq \{L : L \text{ קיימת מכונת טיורינג המקבלת את } \}$$

מחלקת השפות **הרקורסיביות** המסומנת **R** (Recursive sets), היא קבוצת השפות המתקבלות ע"י מכונות טיורינג העוצרות תמיד.

$$\mathbf{R} \triangleq \{L : L \text{ קיימת מכונת טיורינג העוצרת על כל קלט ומקבלת את } \}$$

<u>הערה:</u> מקור השם "ניתן למניה רקורסיבית" יובהר בסעיף 7.5.

<u>תרגיל 7.3:</u> ביחס לבניה 7.3 הוכח:

א. לכל פונקציה מלאה f מתקיים: f ניתנת לחישוב אסם $L_f \in \mathbf{R}$.

ב. לכל פונקציה חלקית f מתקיים: f ניתנת לחישוב אסם $L_f \in \mathbf{RE}$.

<u>מסקנה מן המשפטים:</u> יש התאמה בין הפונקציות הרקורסיביות ובין השפות במחלקה **RE**, וכן יש התאמה בין הפונקציות הפרימיטיביות-רקורסיביות ובין השפות במחלקה **R**. שים לב כי ההתאמות מבלבלות במקצת (מכיוון שפונקציות רקורסיביות מותאמות לשפות ניתנות-למניה-רקורסיבית ולא לשפות רקורסיביות סתם ...). מן ההתאמות הללו ומן ההבטחה שלנו (בפרק הקודם) להראות שלא כל הפונקציות הן רקורסיביות ולא כל הפונקציות הרקורסיביות הן פרימיטיביות-רקורסיביות, נובע שלא כל השפות הן ב-**RE**, ולא כל השפות ב-**RE** הן גם ב-**R**. ננסח שוב טענות אלו במפורש (טענות אלו יוכחו בשלב מאוחר יותר):

(1) קיימות שפות אשר אין מכונות טיורינג המקבלות אותן. בניסוח פורמלי: $\mathbf{RE} \neq \{L : L \subseteq \{0,1\}^*\} \setminus$.

(2) קיימות שפות אשר ניתנות לקבלה ע"י מכונות טיורינג אך אין מכונות טיורינג <u>העוצרות תמיד</u> ומקבלות אותן. בניסוח פורמלי: $\mathbf{RE} \setminus \mathbf{R} \neq \emptyset$.

הערה: לכל שפה L ניתן להגדיר את **הפונקציה האופינית** שלה, שתסומן χ_L, ותוגדר כדלקמן: $\chi_L(x)=1$ אם $x\in L$ ואילו $\chi_L(x)=0$ אם $x\notin L$. קל לראות (הראה!) כי $L\in\mathbf{R}$ אמם הפונקציה (המלאה) χ_L ניתנת לחישוב. כמו-כן ניתן להגדיר לשפה L **פונקצית עצירה**, שתסומן h_L, כך ש- h_L מוגדר על x (וערכו נאמר 1) אמם $x\in L$ (h_L לא מוגדר על $x\notin L$). הראה כי הפונקציה החלקית (או מלאה) h_L ניתנת לחישוב אמם $L\in\mathbf{RE}$.

מכך אנו למדים במפורש את שהנחנו במובלע והוא, שהדיון בזיהוי שפות אינו מקיף יותר מאשר הדיון בחישוב פונקציות.

התזה של צ'רץ' (ניסוח במונחים של קבלת שפות): כל "מודל כללי וסביר של חישוב" (כלומר, אוסף של "מכונות סבירות") מאפשר לקבל בדיוק את השפות **RE**. שוב נציין כי במונח "מודל כללי" הכוונה היא שמכונות במודל יכולות לבצע חישובים כלליים, ובפרט חישובים הניתנים לביצוע בעזרת מכונת טיורינג. במונח "מודל סביר" הכוונה היא למגבלות של מושג החישוב הנובעות מן הדרישה שכל צעד חישוב יוגדר ע"י מספר סופי של "כללים פשוטים".

תרגיל 7.4: הוכח כי אם $L\in\mathbf{R}$ אז גם $\overline{L}\in\mathbf{R}$ ($\overline{L}$ הינה השפה המשלימה של L).

תרגיל 7.5: הוכח כי אם $L\in\mathbf{RE}$ וגם $\overline{L}\in\mathbf{RE}$ אז $L\in\mathbf{R}$.

הערה למתקדמים: מכיוון שכבר הובטח שנראה כי $\mathbf{RE}\setminus\mathbf{R}\neq\varnothing$, נובע מתרגיל 7.5 כי יש שפות ב-**RE** אשר השפות המשלימות שלהן אינן ב-**RE**.

7.2 בעיות חיפוש והקשר שלהן לקבלת שפות

לפני שנעביר את הדיון במושג "חישוב פונקציות" לדיון ב"קבלת שפות", נפנה את תשומת לב הקורא לעובדה שדיון זה ממצה גם דיון רחב יותר, והוא הדיון ב"פתרון בעיות". באופן כללי ניתן לנסח בעית חישוב כדלקמן:

נתון: קלט x.

משימה: מצא "פתרון" y המקיים יחס מסוים עם הקלט x.

באופן פורמלי, נגדיר את **בעית החיפוש** המתאימה ליחס הדו-מקומי $R\subseteq\{0,1\}^*\times\{0,1\}^*$ כדלקמן:

נתון: קלט x.

משימה: מצא y כך ש- $(x,y)\in R$.

שים לב כי חישוב פונקציה הוא מקרה פרטי של בעית חיפוש. את חישוב הפונקציה f ניתן להציג כבעית חיפוש המתאימה ליחס: $R_f\triangleq\{(x,y):y=f(x)\text{ מוגדר על }x\text{ וערכו }y\}$.

שים לב כי יחס יכול להיות מלא או חלקי. היחס מלא אם לכל x קיים פתרון y (כלומר $\forall x\exists y\ (x,y)\in R$).

הגדרה 7.6: תהא R רלציה דו-מקומית חלקית או מלאה. נאמר כי בעית החיפוש של R **ניתנת לפתרון** (R ניתנת לחישוב) אם קיימת מכונת טיורינג M כך שאם קיים y כך ש- $(x,y)\in R$ אז M עוצרת על x עם פלט $M(x)$ כך שמתקיים, $(x,M(x))\in R$.

כדי לראות שהדיון בבעיות חיפוש מתמצה גם הוא בדיון בקבלת שפות, אך במובן חלש יותר, נרחיב את בניה 7.3 באופן פשוט ביותר.

בניה 7.3': תהא R רלציה דו-מקומית חלקית או מלאה,

$$L_R \triangleq \{(x,y,1^n) : \exists y' \in \{0,1\}^n \quad s.t. \quad (x,yy')\in R\}$$

משפט 7.4: תהא R רלציה דו-מקומית מלאה או חלקית. אם $L_R \in \mathbf{RE}$ אז בעית החיפוש של R ניתנת לפתרון. אם $L_R \in \mathbf{R}$ אז קיימת מכונת טיורינג הפותרת את בעית החיפוש של R באופן שהיא דוחה קלטים שאין להם פתרון. ההוכחה מושארת כתרגיל. היא דומה לכיוון אחד של תרגיל 7.3 ומשתמשת בשיטה הנקראת **הרצה מבוקרת** (שתובא באופן מפורט בהוכחת משפט 7.5) על מנת לגלות את אורכו של הפתרון.

הערה למתקדמים: שים לב כי לא מתקיים כאן בהכרח, כי אם בעית החיפוש של R ניתנת לפתרון אז $L_R \in \mathbf{RE}$ ובפרט ההוכחה שניתנה למקרה של חישוב פונקציות אינה תופסת. זאת משום שלקלט x יתכנו מספר פתרונות ואילו קיום מכונה הפותרת את בעית החיפוש, רק מבטיחה זיהוי של אחד מהם. למרות החד-כיווניות של משפט 7.4 די בו כדי להבטיח, שקיום בעיות חיפוש שאינן ניתנות לפתרון יגרור קיום שפות שאינן ב **RE**. מצד שני, ברור שקיום שפות שאינן ב **RE**, יגרום קיום בעיות חיפוש שאינן ניתנות לפתרון (בפרט יהיו אלו בעיות של חישוב פונקציות).

כהצדקה אלטרנטיבית, ניתן להראות כי הדיון בחישוב פונקציות (שכבר ראינו כי הוא מתמצה בדיון בקבלת שפות) ממצה את הדיון בפתרון בעיות חיפוש.

משפט 7.4': בעית החיפוש של היחס R ניתנת לפתרון אם ורק אם קיימת פונקציה f כך שמתקיים:

(1) אם $(x,y)\in R$ אז f מוגדרת על x וכן $(x,f(x))\in R$.

(2) הפונקציה f ניתנת לחישוב.

7.3 בעית חיפוש ובעית זיהוי הפתרון

בסעיף הקודם הגדרנו לכל יחס דו-מקומי R את בעית החיפוש המתאימה לו. אך הרי ניתן להסתכל על היחס R כעל קבוצת זוגות (ולפיכך שפה) ולדון בשאלת הקבלה של אברים בו. בעית החיפוש של יחס R היא: בהנתן x למצוא

y כך ש- $(x,y)\in R$). בעית הזיהוי/קבלה של היחס R היא: בהנתן x,y להכריע האם $(x,y)\in R$. דרך מהותית יותר להציג את בעית הקבלה של R, היא לאמר שלפני שאנו דנים בבעיה "בהנתן x מצא פתרון y כך ש- $(x,y)\in R$" (ז.א. בעית החיפוש של R) עלינו לדון בשאלה "בהנתן x וכביכול-פתרון y, האם y הוא אמנם פתרון ל- x?" (כלומר האם $(x,y)\in R$?). איזה טעם יש לדון במציאת פתרונות, אם לא דנים קודם בשאלה "האם ניתן בכלל לזהות פתרונות חוקיים?". לפני שנעבור לדיון, נגדיר במפורש את "בעית זיהוי הפתרונות".

<u>הגדרה 7.7:</u> יהי R יחס דו-מקומי. נאמר כי בעית הזיהוי (של פתרונות) של היחס R **ניתנת לפתרון** אם R שייכת ל-**RE**, כלומר, אם קיימת מכונת טיורינג M אשר מקבלת את השפה $\{x\#y : (x,y)\in R\}$. (שהיא למעשה השפה R).

<u>הערה למתקדמים:</u> למרות שנראה ממבט ראשון שהיכולת לזיהוי פתרונות של היחס R היא תנאי הכרחי ליכולת לפתור את בעית החיפוש של R, הדבר אינו בהכרח כך. תתכן מכונת טיורינג אשר לכל x מוצאת y כך ש- $(x,y)\in R$ ואמנם זוגות אילו ניתנים לזיהוי, אך יתכנו גם זוגות אחרים שאינם ניתנים לזיהוי. במקרה הפרטי בו R הוא פונקציה (כלומר לכל x קיים y, אחד ויחיד, כך ש- $(x,y)\in R$) האינטואיציה הראשונית נכונה, כפי שנראה בתרגיל הבא (השווה להערה המסיימת את הסעיף הקודם).

<u>תרגיל 7.6:</u> יהי R יחס דו-מקומי, כך שלכל x קיים y יחיד המקיים $(x,y)\in R$. הוכח כי אם בעית החיפוש של R ניתנת לפתרון אז גם בעית זיהוי הפתרונות של R ניתנת לפתרון. (רמז: בהנתן זוג (x,y), מצא פתרון ל- x).

בהקשר שלנו, שעד כה אינו מתייחס כלל ליעילות החישוב, מתקיים תמיד דוקא הכיוון ההפוך, כלומר, אם בעית זיהוי הפתרונות של R ניתנת לפתרון אז גם בעית החיפוש של R ניתנת לפתרון.

<u>תרגיל 7.7:</u> יהי R יחס דו-מקומי, ונניח כי בעית הזיהוי של R ניתנת לפתרון ע"י מכונת טיורינג <u>העוצרת תמיד</u>. הוכח כי בעית החיפוש של R ניתנת לפתרון.

<u>מבנה הפתרון:</u> תהי M מכונה העוצרת תמיד, אשר מקבלת את $x\#y$ אמם $(x,y)\in R$ (מכונה זו מובטחת ע"י תנאי התרגיל). בנה מכונה M' אשר בהנתן x מוצאת y כך ש- $(x,y)\in R$ כדלקמן: המכונה M' תנסה את כל המחרוזות בסדר לקסיקוגרפי. בנסיון עם מחרוזת y' היא תשתמש ב- M על מנת לבדוק האם $(x,y')\in R$. אם התשובה חיובית, M' תחזיר את y' כפלט. אחרת, M' תעבור למחרוזת הבאה.

<u>הערה:</u> שים לב לשימוש המהותי בהנחה ש- M עוצרת על כל קלט $x\#y'$.

<u>משפט 7.5:</u> יהי R יחס דו-מקומי, ונניח כי בעית הזיהוי של R ניתנת לפתרון (כלומר, השפה L_R המקיימת $x\#y\in L_R$ אם ורק אם $(x,y)\in R$, הינה במחלקה **RE**). אז בעית החיפוש של R ניתנת לפתרון.

<u>הוכחה:</u> תהא M מכונה המקבלת את השפה $\{x\#y : (x,y)\in R\}$. הבניה מן התרגיל הקודם לא נכונה במקרה שלנו כי לא מובטח ש-M תעצור על y' שאינו פתרון ל-x (ז.א. $(x,y')\notin R$). במקום זאת נצטרך להריץ את M על כל הפתרונות (ה-y-ים) האפשריים. אך זהו אבסורד, כי מספרם אינו חסום (ולא נוכל לבצע "סימולציה" של הרצתם במקביל). ננסה להתבונן בבעיה מזוית קצת שונה. אם אמנם קיים y כנדרש, הרי שקיים מספר טבעי k כך ש M מקבלת את הקלט $x\#y$ בדיוק ב-k צעדים. כל מה שעלינו לעשות הוא, לדאוג לכך שלכל מחרוזת y ולכל מספר טבעי k, לאחר זמן סופי, ננסה להריץ את M על $x\#y$ לפחות k צעדים. בדרך זו, אם קיימים y ו-k כנדרש הרי שנוכל להגיע אליהם. השיטה בה ננקוט נקראת **הרצה מבוקרת** (או חסומה) על כל הקלטים. המכונה M' (הפותרת את בעית החיפוש של R) תפעל כדלקמן על קלט x: על הסרט השני תתחזק M' מונה אשר יאותחל ל-1. המכונה M' תפעל באיטרציות עד שתעצור (אם אכן קיים פתרון). באיטרציה ה-i-ת המכונה M' תריץ את המכונה M למשך i צעדים על x עם כל אחת מ-i המחרוזות הראשונות בסדר הלקסיקוגרפי. אם עבור אחת המחרוזות הללו, y, תקבל M את הזוג $x\#y$, אז M' תעתיק את y לסרט הפלט שלה ותעצור. אחרת, תקדם M' את המונה ותעבור לאיטרציה הבאה. היווכח כי המכונה M' אמנם פותרת את בעית החיפוש של היחס R. □

<u>הערה:</u> שים לב לכך שהנקודה הקריטית בבניה הנ"ל היא, שהמכונה M' מריצה את המכונה M, על הקלט, עם חסם על מספר הצעדים. על מנת להתרכז ברעיון חדש זה, חשוב על הפונקציה $f_M^{(k)}:\{0,1\}^*\to\Gamma^*$ המוגדרת כקונפיגורציה אליה מגיעה M על קלט x לאחר k צעדים (או פחות, במקרה ש-M עוצרת קודם לכן) ובצורה פורמלית: $f_M^{(k)}(x)=\beta$ אם קיימים: $\gamma\in\Gamma^*$ (Γ היא א"ב העבודה של M) ו-$q\in Q$ (Q הם מצבי M) כך שמתקיים: $q_0x\overset{k}{\underset{M}{\vdash}}\beta q\gamma$ או $q_0x\overset{j}{\underset{M}{\vdash}}\beta p\gamma$ כאשר $j<k$ וגם $p\in F$ (F היא קבוצת המצבים המקבלים של M). בדוק כי אמנם הפונקציה $f_M^{(k)}$ אכן ניתנת לחישוב, ושים לב שגם במקרה בו k הוא חלק מהקלט, $f_M^{(k)}$ עדיין ניתנת לחישוב.

דרך אלטרנטיבית להסתכל על הבעיה, היא כעל מטריצה אינסופית (דו-מימדית) ששורותיה מתאימות למחרוזות האפשריות (y-ים) ועמודותיה מתאימות למספר הצעדים (k). אנו מחפשים כניסה (k,y) כזו המתאימה לחישוב מקבל של M. סריקת המטריצה עמודה אחר עמודה מקבילה רעיונית להרצה "במקביל" של M על כל ה-y-ים, וסריקה שורה אחר שורה מקבילה להרצה ("לא חסומה") של M על כל ה-y-ים בסדר לקסיקוגרפי זה אחר זה. הפתרון מהוכחת משפט 7.5, מקביל לסריקת עמודה אחר עמודה של המשולש התחתון של המטריצה, וע"י כך בדיקת מספר סופי בלבד של כניסות בכל עמודה. גם שיטות אחרות של סריקת המטריצה, כגון סריקה לפי אלכסונים או ריבועים, תוביל לפתרון נכון. מה שמסתתר מאחורי האופן בו נבחרו הזוגות y ו-k מהוכחת משפט 7.5 (ומסריקת המטריצה) הוא הטענה כי הקבוצה $N\times\{0,1\}^*$ (N היא קבוצת הטבעיים) היא קבוצה בת-מניה. בגלל נכונות טענה זו, אנו מסוגלים לעבור סדרתית על זוגות אלו בצורה שתבטיח לנו שנגיע בזמן סופי לכל זוג כזה.

<u>הערה למתקדמים:</u> שים לב שהוכחת משפט 7.5 יוצרת מכונה לבעית החיפוש, אשר מספר הצעדים שהיא עושה אקספוננציאלי באורך הפתרון שהיא מוצאת. בשלב זה עובדה זו אינה מעניינת אותנו, אך פתרון כזה יהיה בלתי

קביל בחלק השני של הקורס משום שאינו "יעיל".

7.4 מכונות לא דטרמיניסטיות

הקשר בין בעית החיפוש של יחס R לבין בעית זיהוי הפתרונות של יחס זה הוא (והוא בלבד) הסיבה להצגת ה"פיקציה" של "מכונה לא-דטרמיניסטית" (פיקציה היא הסכם פורמלי לדון במשהו שאיש אינו מאמין שיש לו קיום ממשי וזאת מכיוון שהדיון הנ"ל משמש לטיפול בענין אחר לגמרי). לאחר שנגדיר חישובים של "מכונות" כאלה, נראה כי בעית זיהוי הפתרונות של יחס דו-מקומי, זהה לפתרון בעית החיפוש של היחס הזה ע"י "מכונות לא-דטרמיניסטיות". זהות זו תהיה קלה להוכחה והמשפט שהוכח בפרק הקודם (משפט 7.5) ינבע מן הטענה שבעית חיפוש של יחס דו-מקומי הניתנת לפתרון ע"י "מכונה לא-דטרמיניסטית", ניתנת לפתרון ע"י מכונות של ממש (ז.א. מכונות דטרמיניסטיות שעסקנו בהן עד עתה). בכל מקרה, כאשר נאמר מכונת טיורינג נתכוון למכונה דטרמיניסטית (ז.א. דבר המייצג חישוב במובן סביר) ואם נרצה להתייחס לפיקציה של "מכונות לא-דטרמיניסטיות" נציין זאת במפורש.

הגדרה 7.8: **מכונת טיורינג לא דטרמיניסטית**, היא שביעיה כמו בהגדרת מכונת טיורינג (רגילה) מלבד שפונקצית המעברים δ היא פונקציה מזוגות אל תת-קבוצה של שלישיות מתוך $Q \times \Gamma \times \{L,R\}$, ז.א.

$$\delta : (Q \setminus F) \times \Gamma \to 2^{(Q \times \Gamma \times \{L,R\})}.$$

הגדרה 7.9: עבור מכונה לא-דטרמיניסטית M, היחס $\vdash_M$ (והסגור שלו $\vdash_M^*$) מוגדר בצורה דומה לצורה שהוגדר עבור מכונה דטרמיניסטית, אלא שכאן ההגדרה אינה מתאימה בהכרח לכל קונפיגורציה, קונפיגורציה עוקבת יחידה, ויתכן (וזהו המקרה המהותי) כי לקונפיגורציה $\beta_0 q \gamma_0$ תהיינה מספר קונפיגורציות עוקבות (כלומר, קונפיגורציות, $\beta_1 q_1 \gamma_1, \beta_2 q_2 \gamma_2, \cdots, \beta_d q_d \gamma_d$ שונות, כך ש- $\beta_0 q_0 \gamma_0 \vdash_M \beta_j q_j \gamma_j$ לכל $1 \le j \le d$). במילים אחרות (בניגוד למצב במכונה דטרמיניסטית) היחס $\vdash_M$ אינו בהכרח פונקציה.

הגדרה 7.10: יהי R יחס דו-מקומי (חלקי או מלא) המכונה הלא-דטרמיניסטית M **פותרת** את בעית החיפוש של R אם לכל x מתקיים:

(1) אם יש פתרון עבור x, אז קיים חישוב של המכונה העוצר במצב מקבל. כלומר, אם קיים y כך ש- $(x,y) \in R$ אז קיימים $\beta,\gamma \in \Gamma^*$ כך ש- $q_0 x \vdash_M^* \beta q_{ACC} \gamma$ (זכור כי q_{ACC} הוא המצב המקבל של M).

(2) אם המכונה עוצרת במצב מקבל, אז הפלט שלה הוא פתרון לקלט. כלומר, אם קיימים $\beta,\gamma \in \Gamma^*$ כך ש- $q_0 x \vdash_M^* \beta q_{ACC} \gamma$ אז $(x,\beta) \in R$.

<u>הערה:</u> מתנאי (2) בהגדרה נובע שאם קיים חישוב של המכונה (על קלט x) העוצר במצב מקבל (q_{ACC}) אז הפלט של חישוב זה (y) הינו פתרון לקלט x (כלומר, $(x,y)\in R$). תנאי (1) אומר, שאם ל-x מסויים קיים פתרון, אז קיים חישוב של M בו המכונה עוצרת במצב מקבל (ומשמע, בשילוב עם תנאי (2), שהיא פולטת פתרון). חישובים של מכונה לא-דטרמיניסטית מלבד היותם פיקציה, הם גם מבלבלים. אין דרישה שהמכונה תמצא בכל חישוב פתרון לקלט x (גם אם קיים כזה) אך יש דרישה שכל מה שהמכונה תפלוט כפתרון (לקלט x) הוא אכן פתרון (לקלט x) וכן שקיים חישוב בו המכונה תמצא פתרון (ל-x) אם הוא קיים. יש להזהר מן הטעות הנפוצה, לנסות להריץ את המכונה הלא-דטרמיניסטית שנית, בתקוה שבפעם זו אולי היא תקבל. נסיון זה חסר ערך, ובעיקר אין להסיק מהעובדה שהמכונה לא קיבלה ב"הרבה" חישובים, שאין חישוב בו היא אכן מקבלת. מכונה לא דטרמיניסטית, "מתאימה" לכל קלט מספר חישובים אפשריים. חלקם טובים (מסתיימים במציאת פתרון) חלקם אינם טובים (אינם עוצרים, או עוצרים במצב דחיה למרות שקיים פתרון) וחלקם אינו עוצר כלל.

<u>תרגיל 7.8:</u> הראה כי לכל מכונה לא-דטרמיניסטית M, קיים יחס דו-מקומי R_M המוגדר (באופן הטבעי) להלן, כך ש-M פותרת את בעית החיפוש של R_M.

$$R_M \triangleq \{(x,y) : \exists \gamma\in\Gamma^* \ s.t. \ q_0x \overset{*}{\underset{M}{\vdash}} yq_{ACC}\gamma\}$$

<u>הערה:</u> היחס R_M הוא המאפיין את יחסי הקלט-פלט של מכונה אי-דטרמיניסטית M. עבור מכונות דטרמיניסטיות, R_M מזוהה עם הפונקציה f_M המחושבת ע"י M (שהיא המאפיין הטבעי של יחסי הקלט-פלט של מכונה דטרמיניסטית M). מכונות לא-דטרמיניסטיות, לעומת זאת, אינן מחשבות פונקציות, אלא אם מגבילים אותן כך שיקימו, לכל x,

$$|\{(x,y) : \exists\gamma \ s.t. \ q_0x \overset{*}{\underset{M}{\vdash}} yq_{ACC}\gamma\}| = 1.$$

<u>הערה:</u> הדיון בבעיות חיפוש הנפתרות ע"י מכונות לא-דטרמיניסטיות (כמו אילו הנפתרות ע"י מכונות דטרמיניסטיות) ממצה גם את הדיון בשפות המתקבלות ע"י מכונות אילו. שים לב כי השפה המתקבלת ע"י מכונה M (בין אם היא דטרמיניסטית או לא) היא קבוצת כל הקלטים שיש להם פתרון. כלומר, $L_M \triangleq \{x : \exists y \ s.t. \ (x,y)\in R_M\}$ כאשר R_M הוא כמוגדר בתרגיל 7.8.

<u>תרגיל 7.9:</u> יהי R יחס דו-מקומי, ונניח כי בעית זיהוי הפתרונות של R ניתנת לפתרון (ע"י מכונת טיורינג דטרמיניסטית). הוכח כי בעית החיפוש של R ניתנת לפתרון ע"י מכונה לא-דטרמיניסטית.

<u>הערה:</u> כמובן שאין כאן כוונה להשתמש במשפט 7.5 (אשר מבטיח כי בעית החיפוש של R ניתנת לפתרון, אפילו ע"י מכונה של ממש). הכוונה היא לפתרון פשוט הרבה יותר, אשר פתרונו יתואר להלן.

<u>פתרון:</u> תהא M מכונת טיורינג (דטרמיניסטית!) אשר מזהה את הפתרונות של היחס R. נבנה מכונה לא-דטרמיניסטית M' אשר פותרת את בעית החיפוש של R כדלקמן. על קלט x, המכונה M' מעתיקה את x לסרט השני שלה ומוסיפה לו סימן # מימינו. המכונה M' "מנחשת" (כלומר, כותבת באופן לא דטרמיניסטי, סדרת ביטים כלשהי) מחרוזת y וכותבת אותה בהמשך הסרט השני. [תקוותנו היא ש $(x,y)\in R$. אם <u>קיים</u> y כך ש- $(x,y)\in R$, אז קיים חישוב בו y זה ינוחש]. המכונה M' מריצה כעת את M על הסרט השני שלה. אם M קיבלה את $x\#y$, אז M' תעתיק את y אל סרט הפלט שלה ותעצור במצב מקבל. אם M לא קיבלה את $x\#y$, אז M' תעצור במצב דוחה. מי שמעדיף, יכול לכתוב ש- M' נכנסת ברגע זה ללולאה אינסופית בה היא אינה עושה דבר. מיותר לחלוטין לנסות "לנחש" y חדש ולחזור על הסיפור, וטעות לחשוב כי אחרי מספר מסויים של ניסיונות, יובטח משהו.

<u>תרגיל 7.10:</u> יהי R יחס דו-מקומי, וניח כי בעית החיפוש של R ניתנת לפתרון ע"י מכונה לא-דטרמיניסטית. הראה כי בעית החיפוש של R ניתנת לפתרון (ע"י מכונה דטרמיניסטית כמובן).

<u>הדרכה:</u> ראשית הזהר לא להניח כי המכונה M בתרגיל זה, זהה עם מכונה M' כלשהי, המתקבלת ע"י התרגיל הקודם. לעצם פתרון התרגיל, כדאי להעזר בשלושת התרגילים הבאים.

<u>תרגיל 7.11:</u> תהא M מכונה לא-דטרמיניסטית. הוכח כי קיימת מכונה לא-דטרמיניסטית שקולה (כלומר הפותרת אותה בעית חיפוש, לאותו יחס R) M', בה דרגת האי-דטרמיניזם (שתוגדר להלן) חסומה ע"י 2. (רמז: ממש בחירה לא-דטרמיניסטית מדרגה 2^k ע"י סדרה של k בחירות לא-דטרמיניסטיות).

<u>הגדרה 7.11:</u> **דרגת האי–דטרמיניזם** של מכונה לא-דטרמיניסטית M, היא חסם על גודל הקבוצה $\delta(q,\sigma)$ אשר תקף לכל $\sigma\in\Gamma$ ו $q\in Q\setminus F$. נאמר כי דרגת האי-דטרמיניזם של M חסומה ע"י 2 אם, $\forall q\in(Q\setminus F)\,\forall\sigma\in\Gamma:\ |\delta(q,\sigma)|\le 2$.

שים לב כי ניתן לחשוב על מכונה דטרמיניסטית, כעל מכונה לא-דטרמיניסטית בעלת דרגת אי-דטרמיניזם 1.

נוח יהיה לצמצם את הדיון במכונות לא-דטרמיניסטיות, למכונות בעלות מבנה מיוחד הנקראות **מכונות ניחוש תחילה** ופועלות בשני שלבים. בשלב הראשון, הנקרא השלב הלא-דטרמיניסטי, המכונה "מנחשת" מחרוזת (ביט אחר ביט) אותה היא רושמת על הסרט השני (כלומר, מבצעת שורת צעדים, שבכל צעד היא כותבת באופן לא-דטרמיניסטי, אות $\sigma\in\{0,1\}$ על הסרט השני, ונעה עליו תא אחד ימינה). ניחוש זה מתבצע ללא תזוזה על סרט הקלט, ולכן אינו תלוי בקלט. השלב הראשון יכול להסתיים (באופן לא-דטרמיניסטי) כאשר המכונה עוברת למצב "דטרמיניסטי". בשלב השני (הנקרא השלב הדטרמיניסטי) המכונה מבצעת צעדים דטרמיניסטיים בלבד. בכל צעד שלה המכונה זזה צעד אחד שמאלה על הסרט השני ולפיכך מספר הצעדים בשלב השני אינו עולה על מספר הצעדים בשלב הראשון. שים לב כי המכונה מסוגלת שלא לעצור, רק ע"י כך שהיא לא תצא לעולם מהשלב הראשון.

<u>תרגיל 7.12:</u> תהא M מכונה לא-דטרמיניסטית. אז קיימת מכונה לא-דטרמיניסטית מטיפוס ניחוש תחילה, השקולה למכונה M (כלומר, פותרת את אותה בעית חיפוש).

<u>פתרון:</u> ללא הגבלת הכלליות (תוך הסתמכות על תרגיל 7.11) נניח כי דרגת האי-דטרמיניזם של M היא 2. נבנה מכונה לא-דטרמיניסטית M' מן הטיפוס של ניחוש תחילה, אשר בשלב הדטרמיניסטי שלה תבצע סימולציה של חישוב של M עם צעדים לא-דטרמיניסטיים המתוארים ע"י המחרוזת שנוחשה תחילה. במילים פשוטות יותר, על קלט x המכונה הלא-דטרמיניסטית M' מנחשת תחילה מחרוזת $\alpha = \sigma_1 \cdots \sigma_n$. כעת מבצעת M' חישוב דטרמיניסטי הזהה לחישוב אפשרי של M על הקלט, כאשר בצעד ה i-י אם למכונה M שתי אפשרויות לביצוע הצעד, המכונה M' תבחר בצורה דטרמיניסטית את האפשרות (הראשונה או השניה) לפי הערך של σ_i. במהלך n צעדי הסימולציה, יתכן כי המכונה M עצרה במצב סופי. במקרה כזה, M' עוצרת אף היא באותו מצב. אחרת (M לא עצרה במהלך הסימולציה) M' עוצרת במצב דחיה (שים לב שסרט הפלט של M' צריך להיות זהה לזה של M). ודא תחילה שאתה מבין את פעולת M', והראה כי M' אמנם שקולה ל-M (כלומר, קיים חישוב של M על x המקבל עם קלט y אם קיים חישוב כזה של M').

<u>תרגיל 7.13:</u> תהא M מכונה לא-דטרמיניסטית מטיפוס ניחוש תחילה. הראה כי קיימת מכונה דטרמיניסטית השקולה לה (כלומר פותרת אותה בעיית חיפוש).

<u>פתרון:</u> המכונה הדטרמיניסטית M' תעבור בסדר לקסיקוגרפי על כל המחרוזות האפשריות של ניחוש תחילה במכונה M, ולכל אפשרות תריץ את M מהשלב הדטרמיניסטי שלה.

נבהיר: את המכונה הלא-דטרמיניסטית M של ניחוש תחילה, יש לראות כהרכבה של שתי מכונות:

M_N - מכונה לא-דטרמיניסטית המנחשת מחרוזת באופן בלתי תלוי בקלט,

M_D - מכונה דטרמיניסטית המבצעת חישוב על תוכן הסרטים שהושארו בידי M_N.

המכונה M' תסרוק את כל המחרוזות α בסדר לקסיקוגרפי, ותריץ את M_D על קלט x ותוכן סרט שני α , עד שבאחת ההרצות M_D תעצור במצב מקבל. אם זה אכן קורה, M' תעתיק את הפלט של M_D לסרט הפלט שלה ותעצור במצב מקבל. אחרת, M' עוברת למחרוזת העוקבת ל-α בסדר הלקסיקוגרפי. חשוב לציין כי על קלט x ותוכן סרט שני α, המכונה M_D עושה לכל היותר $|\alpha|$ צעדים, ולכן כל הרצה של M_D היא סופית.

<u>הערה למתקדמים:</u> שים לב כי בתרגיל 7.13 נפתרה הבעיה של הרצת M_D על כל קלט α (בתוספת לקלט המקורי x) עם כל חסם אפשרי על מספר הצעדים ע"י הרצתו לכל היותר $|\alpha|$ צעדים. בעיה זו דומה לבעיה שהעסיקה אותנו בהוכחת משפט 7.5. שים לב שבניגוד להוכחת משפט 7.5 שילבנו כאן את האי-דטרמיניזם (y) עם חסם הריצה (k) במחרוזת אחת α.

משילוב תוצאות תרגילים 7.9 ו- 7.13, נובעת הוכחה אלטרנטיבית של משפט 7.5 (בעית החיפוש של יחס דו-מקומי

ניתנת לפתרון אם בעית זיהוי הפתרונות של היחס ניתנת לפתרון). למעשה ניתן להוכיח את משפט 7.5 אפילו אם נגביל את טענת תרגיל 7.13 למכונות ניחוש תחילה (כי כאלו, כמעט, הן המכונות המתקבלות מתרגיל 7.7). אם נפרש, הרי שההוכחה האלטרנטיבית של משפט 7.5 נובעת משתי טענות:

(1) אם בעית זיהוי הפתרונות של R ניתנת לפתרון (ע"י מכונה דטרמיניסטית) אז בעית החיפוש של R ניתנת לפתרון ע"י מכונה לא-דטרמיניסטית מטיפוס ניחוש תחילה (זה מוכח כמעט לגמרי בתרגיל 7.9).

(2) אם בעית החיפוש של R ניתנת לפתרון ע"י מכונת ניחוש תחילה, אז היא ניתנת לפתרון ע"י מכונה דטרמיניסטית (זהו בדיוק תרגיל 7.13).

כאן מתברר תפקיד הפיקציה של מכונה לא-דטרמיניסטית כשלב מעבר בהוכחה האלטרנטיבית הנ"ל. בתור שלב מעבר, ניתן לוותר על מושג המכונה הלא-דטרמיניסטית, ואמנם ההוכחה הראשונה למשפט 7.5 אינה משתמשת במושג זה.

<u>הערה למתקדמים:</u> המעבר שנעשה בתרגיל 7.13 ממכונה לא-דטרמיניסטית למכונה דטרמיניסטית, אקספוננציאלי במספר הצעדים, ופסול בהקשר בו ישנה חשיבות לסיבוכיות החישוב. בהקשר כזה (חלק ב' של הקורס) תשמש לכן הפיקציה של מכונה לא-דטרמיניסטית מושג של קבע (המשמש לצורך התייחסות לבעיות חיפוש שניתן לזהות את הפתרונות בהן) ולא שלב מעבר בלבד.

7.5 אנומרטורים (מניה רקורסיבית)

בניגוד לסעיף הקודם שהיה פיקציה בעלת ערך (בעיקר לעתיד) הרי שתוכן סעיף זה הוא תרגיל כמעט חסר ערך, שמירב התועלת החינוכית שלו, היא בהסבר לשם המיסתורי **RE** ("מחלקת השפות הניתנת למניה רקורסיבית"). בסעיף זה נעסוק במכונות טיורינג מיוחדות הנקראות "אנומרטורים" אשר (בדרך כלל) אינן עוצרות לעולם, אלא מדפיסות על סרט מיוחד סדרה (בדרך כלל אינסופית) של מילים מעל א"ב $\{0,1\}$. שים לב שקשה לקרוא לסדרה זו פלט, משום שפלט הוגדר כתוכן סרט (מסוים) בעת העצירה של המכונה. ה"פלט" כאן גדל ללא הרף ויתכן שכתיבתו אינה נפסקת לעולם.

<u>הגדרה 7.12:</u> **אנומרטור** M הוא מכונת טיורינג בעלת סרט יחיד וא"ב עבודה המכיל (בנוסף ל-0, 1 ו-␣) גם את הסימן #, אשר פונקצית המעברים שלה מקיימת לכל $q \in Q \setminus F$ קיים $p \in Q$ כך ש $\delta(q,\#) = (p,\#,R)$.

המילה ה-i-ית שאנומרטור M **מייצר** היא w_i אם קיימים $w_1, \ldots, w_{i-1}, \beta \in \{0,1\}^*$ כך שמתקיים:

$$q_0 \overset{*}{\underset{M}{\vdash}} w_1 \# w_2 \# \cdots \# w_{i-1} \# w_i \# \beta$$

הסבר: הפעולה של אנומרטור נבחנת רק ביחס לקלט הריק (ואין ענין בשאלה מה הוא עושה על קלט אחר). האנומרטור אף פעם אינו חוזר אל קטע סרט שכתב, לאחר שכתב סימן # מימין לו. לפיכך, מרגע שהאנומרטור כתב סימן # מימין ל-w_i כל הקטע $\#w_i\# \cdots \#w_i\#w_2\#w_1$ לא ישתנה עוד במהלך החישוב. במידה ומספר המילים שמייצר האנומרטור הוא סופי ושווה ל-n, פרוש הדבר, שהאנומרטור אף פעם לא כותב סימן # בנוסף ל-n הסימנים שכתב בשלב מסוים. המקרה של אנומרטורים המייצרים מספר סופי של מילים אינו מעניין וסתם מסרבל את הדיון, כל ההתיחסויות לאנומרטורים אלו מוקפות בסוגריים מרובעים ([ ]) וכדאי להתעלם מהן ולהניח בהמשך כי מדובר רק באנומרטורים המייצרים מספר אינסופי של מילים (שייצג שפה אינסופית).

הגדרה 7.13: אנומרטור M נקרא **בלתי חוזר**, אם לכל $i < j$ מתקיים [אם M מייצר לפחות j מילים אז] המילה ה-j-ית שהוא מייצר שונה מן המילה ה-i-ית שהוא מייצר.

מרגע זה והלאה, נתעניין רק באנומרטורים בלתי חוזרים, ובכל מקום שנכתוב אנומרטור יש לקרוא אנומרטור בלתי חוזר.

הגדרה 7.14: אנומרטור M **מייצר את השפה** L, אם הוא מייצר בדיוק את כל המילים ב-L (כלומר, $w \in L$ אם ורק אם קיים i כך שהמילה ה-i-ית ש-M מייצר היא w).

הערה: היה טבעי לקרוא לשפה L "ניתנת למניה רקורסיבית", אם קיים אנומרטור המייצר את השפה L, אולם השם "ניתן למניה רקורסיבית" כבר ניתן לשפות **RE**. כפי שנראה בקרוב שם זה מוצדק כי אמנם $L \in \mathbf{RE}$ אמם קיים אנומרטור המייצר את השפה L, וזהו למעשה מקור השם "שפות הניתנות למניה רקורסיבית".

תרגיל 7.14: הוכח כי $L \in \mathbf{RE}$ אמם קיים אנומרטור המייצר את השפה L.

כיוון I: נניח תחילה כי קיים אנומרטור M אשר מייצר את L. נבנה מכונה M' המקבלת את L. על קלט x המכונה M' מפעילה את M על הסרט השני, ובכל פעם ש-M כותבת #, המכונה M' תשווה את המילה האחרונה שיוצרה עם x. אם יש שיוויון, המכונה M' עוצרת במצב מקבל. אחרת, M' ממשיכה את הרצת M. שים לב שאם $x \notin L$ יתכן כי M' לא תעצור לעולם, אך זה בסדר. לעומת זאת, אם $x \in L$, אז קיים i כך ש-x היא המילה ה-i-ית ש-M מייצר ומובטח לכן ש-M' תעצור במצב מקבל.

כיוון II: נניח כי קיימת מכונה M אשר מקבלת את השפה L ונבנה אנומרטור M' ע"י "הרצה במקביל" של M על כל המחרוזות עם כל חסמי מספר הצעדים, תוך שימוש ברעיון שהוצג בהוכחת משפט 7.5. המכונה M' תעבוד באיטרציות. באיטרציה ה-i-ית היא תריץ את M על כל אחת מ $i-1$ המילים הראשונות בסדר הלקסיקוגרפי למשך i צעדים בדיוק ואת המילה ה-i-ית תריץ i צעדים לכל היותר. אם M מקבלת אחת או יותר מן המילים האלו (במספר הצעדים האמור) אז M' תרשום אותם על סרט ה"פלט" (מופרדים ע"י #). וודא כי כל $x \in L$ יווצר ע"י האנומרטור

בדיוק פעם אחת, וכן כי כל $x \notin L$ לא יווצר ע"י M' לעולם. שים לב כי האנומרטור M' מייצר את השפה L אך לא בהכרח בסדר לקסיקוגרפי.

הגדרה 7.15: אנומרטור M נקרא **קנוני**, אם לכל $i > 1$ מתקיים [אם M מייצר לפחות i מילים אז] המילה ה-i-ית שהוא מייצר, גדולה (בסדר לקסיקוגרפי של מחרוזות) מן המילה ה $i-1$ שהוא מייצר. שפה L ניתנת לאנומרציה קנונית, אם קיים אנומרטור קנוני המייצר את השפה L.

נחזור כעת על ההערה שקדמה להגדרה: האנומרטור שנבנה בהוכחת תרגיל 7.14 אינו בהכרח אנומרטור קנוני, ובפרט שפות **RE** אינן ניתנות בהכרח לאנומרציה קנונית. נראה כי השפות הניתנות לאנומרציה קנונית זהות למחלקה **R** ולפי האמור בסעיף 7.1 נובע שהן תת-קבוצה ממש של השפות שיש להן אנומרטור שמייצר אותן.

תרגיל 7.15: הוכח כי $L \in \mathbf{R}$ אסם קיים אנומרטור קנוני המייצר את L. (יש לדון בנפרד בשפות סופיות ומקרה זה פשוט. נניח כעת L אינסופית).

ביוון I: נניח כי קיים אנומרטור קנוני, M, המייצר את L. נבנה מכונה M' העוצרת תמיד ומקבלת את L. M' פועלת באופן דומה ל M' מתרגיל 7.14, אלא שהיא מסוגלת לזהות בנוסף למקרה ש $x \in L$, גם את המקרה ש $x \notin L$. המקרה $x \notin L$ מזוהה כאשר מכונה M כבר ייצרה מילה y כך ש $x < y$ (בסדר לקסיקוגרפי) ועדיין לא ייצרה את x. מן הקנוניות של M נובע שאם x לא יוצר עד עתה, הוא לא ייוצר בעתיד. כאשר M' מזהה כי $x \in L$ או $x \notin L$, היא עוצרת במצב מקבל או דוחה בהתאמה. שים לב כי מאינסופיות L נובע כי תמיד קיים $y \in L$ הגדול מ-$x \notin L$.

ביוון II: נניח כי קיימת מכונה M העוצרת תמיד ומקבלת את השפה L. נבנה אנומרטור קנוני M' לשפה L, בניה זו תהיה פשוטה יותר מאשר הבניה בתרגיל הקודם. המכונה M' "מייצרת" על סרט עזר את כל המחרוזות זו אחר זו (בסדר לקסיקוגרפי קנוני). על כל מחרוזת היא מריצה את M. M' מוסיפה מחרוזת לסרט הפלט שלה (בתוספת #) אסם M מקבלת את המחרוזת. שים לב כי אנו משתמשים בעובדה ש M עוצרת על כל קלט. □

צורה אלטרנטיבית לדון באנומרטורים, עבור שפות אינסופיות, היא להגדיר "מכונות עוקב". מכונות אלו יהיו מכונות טיורינג העוצרות תמיד, אשר מייצרות פונקציה "חסרת מחזורים" (ראה הגדרות).

הגדרה 7.16: **מכונת עוקב**, M, היא מכונה אשר הפונקציה שהיא מחשבת, f_M, מקיימת את הדרישה הבאה. לכל מחרוזת x עליה M מוגדרת ולכל מספר $n > 0$ מתקיים, $f_M^{(n)}(x) \neq x$ כאשר $f^{(n)}$ מוגדר אינדוקטיבית ע"י, $f^{(0)}(x) = x$ וכן $f^{(i+1)}(x) = f(f^{(i)}(x))$.

הגדרה 7.17: מכונת עוקב נקראת **מונוטונית**, אם הפונקציה שהיא מחשבת (כלומר, f_M) מקיימת לכל קלט x, עליו M מוגדרת, $x < f_M(x)$.

<u>הגדרה 7.18:</u> **השפה המושרה ע"י מכונת עוקב** M על גרעין s מוגדרת כקבוצה $\{x : \exists n \geq 0 \;\; x = f_M^{(n)}(s)\}$.

<u>תרגיל 7.16:</u> הוכח כי לכל שפה L יש אנומרטור המייצר אותה אסם קיים $s \in L$ ומכונת עוקב M, כך ש L מושרה ע"י M על הגרעין s.

<u>תרגיל 7.17:</u> תהא s_0 מילה ו-M_0 מכונת עוקב אשר משרה את השפה L על הגרעין s_0. הראה כי לכל $s \in L$ קיימת מכונת עוקב M_s אשר משרה את השפה L על הגרעין s. (רמז: העזר באנומרטור המובטח מתרגיל 7.16).

<u>תרגיל 7.18:</u> האם יתכן שאותה מכונת עוקב תשרה את אותה שפה על שני גרעינים שונים?

<u>תרגיל 7.19:</u> הוכח כי לכל שפה L יש אנומרטור קנוני המייצר אותה אסם קיים $s \in L$ ומכונת עוקב מונוטונית המשרה את L על הגרעין s (מהו אותו s?).

<u>תרגיל 7.20:</u> הוכח כי לכל שפה אינסופית $L \in \mathbf{RE}$, קיימת שפה אינסופית $L' \in \mathbf{R}$ המוכלת ב-L (רמז: חשוב על האנומרטור המייצר את L, ועל העובדה כי לכל מילה, x, הוא מייצר מילה, y, הגדולה מ-x בסדר הלקסיקוגרפי).