10. רדוקציות

לכל ההוכחות האחרונות שהצגנו לאי-כריעות של בעיות, היה מבנה משותף. הן הניחו בדרך השלילה את כריעות הבעיות שבהן עסקנו והגיעו לסתירה ע"י בנית מכונת טיורינג המכריעה בעיה אחרת, אשר כבר היתה ידועה לנו כאי-כריעה. מבנה רעיוני זה ניתן למצות בטענה, שהבעיה הידועה לנו כלא ניתנת להכרעה **ניתנת לרדוקציה** לבעיה שאנו מנסים להוכיח שאינה כריעה.

<u>הגדרה 10.1:</u> בעיה חישובית Π_1 **ניתנת לרדוקציה** לבעיה חישובית Π_2, אם קיום מכונת טיורינג הפותרת את Π_2, גורר קיום מכונת טיורינג הפותרת את Π_1.

בהגדרה זו Π_1 ו-Π_2 יכולות להיות בעיות של חישוב פונקציה (כלומר, "בהנתן x חשב את $f(x)$"), או בעיות הכרעה ("בהנתן x האם x מקיים תכונה מסוימת?", ובפרט "האם x שייך לשפה מסוימת L?"). לעובדה שהבעיה Π_1, ניתנת לרדוקציה לבעיה Π_2, יכולות להיות השלכות משני סוגים מנוגדים:

<u>השלכה חיובית:</u> אם קיימת מכונה הפותרת את Π_2 אז קיימת מכונה הפותרת את Π_1. מסקנה זו היא המקובלת יותר בשימוש בהנדסה, במדע ובחשיבה בכלל. זהו שימוש קונסטרוקטיבי ברדוקציה והאינטואיציה שלו היא, מעבר מבעיה חדשה (Π_1) שיש לפתור, לבעיה ישנה (Π_2) שכבר ידוע דרך לפתרונה.

<u>השלכה שלילית:</u> אם לא קיימת מכונה הפותרת את Π_1, אז בהכרח לא קיימת מכונה הפותרת את Π_2. שימוש זה ברדוקציה הינו פחות מקובל והרבה יותר מבלבל. שימוש זה אינו על מנת לפתור בעיות, אלא על מנת להוכיח כי לא קיים להן פתרון. אנו מראים שקיום פתרון לבעיה החדשה (Π_2), יגרור קיום פתרון לבעיה ישנה (Π_1) שידוע כי אין לה פתרון. שים לב לחילוף התפקידים בין Π_1 ל-Π_2. הרדוקציה היא מבעיה ישנה (Π_1) שאינה ניתנת לפתרון, לבעיה חדשה (Π_2).

שימוש חיובי במושג הרדוקציה הוא ענין של שגרה ואין כמעט צורך להדגימו. למעשה, גם אנחנו עשינו שימוש חיובי שכזה, בהוכחות בפרק 7. אמנם הרדוקציות לא צוינו במפורש, אך קל לראות כי הוכחות של משפטים דוגמת 7.1, 7.2 ו-7.3, הן למעשה שימוש חיובי ברדוקציה בשני כיווני ההוכחה. "שימוש שלילי" (סמוי) במושג הרדוקציה, עשינו לראשונה בסעיף 9.2. השימוש הראשון נעשה כאשר הסקנו כי $\overline{U}$ אינה ניתנת לחישוב, מתוך הטענה ש-$\tilde{U}_1$ אינה ניתנת לחישוב. זכור כי $\tilde{U}_1(<M>) = \overline{U}(<M>,<M>)+1$. ברור כי הבעיה של חישוב $\tilde{U}_1$ ניתנת לרדוקציה לבעיה של חישוב $\overline{U}$. "שימוש שלילי" בעובדה זו, גוזר מן הטענה האומרת כי "$\tilde{U}_1$ אינה ניתנת לחישוב", את הטענה האומרת כי "$\overline{U}$ אינה ניתנת לחישוב". דוגמא נוספת לשימוש סמוי במושג הרדוקציה הוא בהוכחת משפט רייס. בהוכחה זו הראנו למעשה, כי בעית העצירה ניתנת לרדוקציה לבעית הזיהוי של קידודי מכונות שהשפה שלהן נמצאת במחלקה C (C היא מחלקה לא טריביאלית של **RE**). מכך הסקנו (בהתבסס על אי-כריעותה של בעית העצירה) כי בעית ההכרעה שעניינה אותנו, אינה כריעה.

שים לב שההגדרה של רדוקציה שניתנה כאן היא כללית מאד ולכן ערטילאית למדי. בפרט, לא דרשנו שתהיה דרך אפקטיבית לעבור ממכונה הפותרת את Π_2 למכונה הפותרת את Π_1. מושגים מצומצמים יותר של רדוקציה יספיקו

לצרכנו בעתיד. יתרונם בכך שהם קונקרטיים יותר ולפיכך פשוטים יותר להבנה ולשימוש.

10.1 רדוקציה ע"י סברוטינה (Turing reduction)

רוב הרדוקציות (מ-Π_1 אל Π_2) שראינו, לא רק הסיקו מקיום מכונה הפותרת את Π_2 את קיומה של מכונה הפותרת את Π_1, אלא גם עשו שימוש במכונה הפותרת את Π_2 על מנת לפתור את Π_1. ברוב המקרים, השימוש היה בצורה של סברוטינה. המכונה הפותרת את Π_1 השתמשה במכונה כלשהי, הפותרת את Π_2, כבסברוטינה. שים לב כי שימוש במונח "סברוטינה", זהה להגדרה המדויקת של סברוטינה כפי שמופיעה בספרי לימוד לתכנות, אך לא זהה עם הניסיון התכנותי שלנו, בו אנו לפעמים מניחים הנחות סמויות על הסברוטינה, ביודענו שאנו עומדים לתכנת גם אותה. כאן עלינו להזהר שלא להניח במובלע הנחות כאלו.

על מנת להגדיר את מושג הרדוקציה ע"י סברוטינה, נזדקק לפיקציה נוספת ששמה "מכונת אוב" (Oracle-Machine). אינטואיטיבית, "מכונת אוב" הינה מכונת טיורינג בעלת סרט מיוחד (הנקרא סרט האוב) עליו המכונה כותבת וקוראת. מכונת האוב כותבת שאלות על הסרט המיוחד ועוברת למצב מיוחד, הנקרא מצב קריאה לאוב. ברגע זה מתחלף תוכן סרט האוב באופן "פלאי". התוכן החדש נקרא "תשובת האוב". שים לב כי מכונת אוב בפני עצמה אינה מהווה מודל סביר של חישוב, אלא אם כן האוב עצמו ניתן למימוש ע"י מודל סביר של חישוב.

הגדרה 10.2: **מכונת אוב** הינה מכונת טיורינג בעלת מספר סרטים אשר אחד מהם נקרא **סרט האוב**, ובעלת מספר מצבים אשר אחד מהם נקרא **מצב זימון האוב** ואחר נקרא **מצב הופעת האוב**.

הגדרה 10.3: תהי M מכונת אוב ותהי f פונקציה מלאה כלשהי. **החישוב של M עם אוב לפונקציה f** מוגדר כדלקמן:

(1) אם בקונפיגורציה הנוכחית מצב המכונה אינו **מצב זימון האוב**, אז הקונפיגורציה העוקבת מוגדרת כפי שהוגדרה עבור מכונת טיורינג רגילה. שים לב, לצורך זה (ובכלל) מצב הופעת האוב הינו מצב רגיל.

(2) אם בקונפיגורציה הנוכחית מצב המכונה הינו **מצב זימון האוב** ותוכן סרט האוב הינו w, אז בקונפיגורציה העוקבת תוכן סרט האוב הינו $f(w)$, מצב המכונה הינו **מצב הופעת האוב** ותוכן שאר סרטי המכונה ומיקום ראש המכונה על גביהם, זהה לזה שבקונפיגורציה הנוכחית.

תרגיל 10.1: הוכח כי כל פונקציה הניתנת לחישוב, ניתנת לחישוב ע"י מכונת אוב עם אוב לפונקציה כלשהי (רמז: התעלם מהאוב).

תרגיל 10.2: תהי M מכונת אוב. הוכח כי אם M, עם אוב לפונקציה f, מחשבת את הפונקציה g, והפונקציה f

ניתנת לחישוב (ע"י מכונה רגילה), אז הפונקציה g ניתנת לחישוב (ע"י מכונה רגילה) גם כן (רמז: ממש את האוב).

<u>תרגיל 10.3:</u> הוכח כי לכל פונקציה g קיימת מכונת אוב M ופונקציה f כך ש-M עם אוב לפונקציה f מחשבת את g.

<u>הגדרה 10.4:</u> חישוב הפונקציה g **ניתן לרדוקצית טיורינג** לחישוב הפונקציה f אם קיימת מכונת אוב M, כך שהמכונה M עם אוב לפונקציה f מחשבת את הפונקציה g.

מתרגיל 10.2 נובע כי **רדוקצית טיורינג** היא מקרה פרטי של **רדוקציה** כפי שהוגדרה בתחילת הפרק.

<u>הערה למתקדמים:</u> ניתן להגדיר את מושג הרדוקציה גם עבור פונקציות חלקיות. לשם כך תדרש הגדרת חישוב של מכונת אוב עם אוב לפונקציה חלקית. יש להגדיר חישוב זה, כך שבמקרה והמכונה שואלת את האוב לגבי ערך עבורו פונקצית האוב אינה מוגדרת, החישוב "יתקע" (כלומר, לא יסתיים).

10.2 רדוקציה ע"י העתקה (many-to-one reduction)

מושג פשוט יותר של רדוקציה הוא המושג של רדוקציה ע"י העתקה. זהו מקרה פרטי של רדוקצית טיורינג בו מכונת האוב קוראת לאוב פעם אחת ויחידה ומוציאה כפלט בדיוק את תשובת האוב (זוהי למעשה "קריאת זנב" לסברוטינה). במקרה כזה החלק המהותי בפעולת מכונת האוב הוא החישוב הקודם לקריאה לאוב. חישוב זה הינו העתקה של הקלט לשאלה לאוב (ניתן לראות בהעתקה זו "תרגום" במובן שיתברר בהמשך).

<u>הגדרה 10.5:</u> פונקציה f_1 **ניתנת לרדוקציה ע"י העתקה** לפונקציה f_2 אם קיימת פונקציה מלאה וניתנת לחישוב h, כך שלכל x מתקיים $f_1(x)=f_2(h(x))$.
מקרה פרטי של הגדרה זו, הוא כאשר f_1 ו-f_2 הן פונקציות בוליאניות (כלומר בעלות טווח של שני ערכים). בעיות חישוב פונקציות כאלו נקראות בעיות הכרעה והן למעשה בעיות קבלת שפות.

<u>הגדרה 10.6:</u> שפה L_1 **ניתנת לרדוקציה ע"י העתקה** לשפה L_2 אם קיימת פונקציה ניתנת לחישוב h כך ש-$x \in L_1$ אם ורק אם $h(x) \in L_2$.

שים לב שבעת הצגת רדוקציה ע"י העתקה, יש לודא שני דברים:

(1) ההעתקה (h) אמנם ניתנת לחישוב.

(2) ההעתקה (h) הינה תקפה (כלומר משמרת את הפתרוניות של הבעיות). כאשר מדובר בפונקציות, פירוש תנאי זה הינו שלכל x מתקיים $f_1(x)=f_2(h(x))$, ואילו במקרה של שפות, פירוש תנאי זה שלכל x מתקיים $x \in L_1$ אמם $h(x) \in L_2$.

<u>תרגיל 10.4:</u> הוכח כי רדוקציה ע"י העתקה הינה מקרה פרטי של רדוקצית טיורינג. כלומר, הראה כי אם פונקציה f_1 ניתנת לרדוקציה ע"י העתקה לפונקציה f_2, אז הפונקציה f_1 ניתנת לרדוקצית טיורינג לפונקציה f_2.

<u>תרגיל 10.5:</u> הוכח ישירות (מבלי להעזר בתרגילים 10.2 ו-10.4), כי אם שפה L_1 ניתנת לרדוקציה ע"י העתקה לשפה L_2 והשפה L_2 הינה ב-**R**, אז $L_1 \in \mathbf{R}$.

שים לב כי כל הרדוקציות שהשתמשנו בהן להוכחת המשפטים בפרק 9, הן למעשה רדוקציות ע"י העתקה. הרדוקציה היתה מבעיה ישנה (שהיתה כבר ידועה לנו כלא כריעה), לבעיה חדשה (שאת אי כריעותה נתבקשנו להוכיח). כמו כן, שים לב כי בסיום כל הוכחה, נדרשת לודא שלושה דברים:

(1) ההעתקה אמנם ניתנת לחישוב.

(2) ההעתקה הינה תקפה.

(3) קיום הרדוקציה וכריעות הבעיה החדשה, גוררים את כריעות הבעיה הישנה (ובאופן שקול קיום הרדוקציה ואי-כריעות הבעיה הישנה, גורר את אי-כריעות הבעיה החדשה).

בשלב זה אנו מתורגלים מספיק על מנת להמנע מלודא את קיום התנאי השלישי בכל פעם. מה עוד שכבר הוכחנו תנאי זה בצורה כללית (ראה תרגילים 10.2, 10.4 ו-10.5). די יהיה בקיום רדוקציה (אשר מקיימת כמובן את (1) ו-(2)) ובהיות "הבעיה הישנה" אי-כריעה, כדי להסיק את אי-כריעות "הבעיה החדשה".

<u>תרגיל 10.6:</u> הוכח כי השפות הבאות אינן ב-**R**.

א. $\{(<M>,q) : $ המכונה M אינה נכנסת למצב q על אף קלט $\}$.

<u>רמז:</u> העתק כל מכונה, למכונה שקולה בעלת מצב סופי יחיד.

ב. $\{(<M>,\alpha,\beta) : \exists k \ s.t. \ \alpha \overset{*}{\vdash}_M \beta \not\vdash^k\}$.

<u>הערה:</u> שים לב כי השפה $\{(<M>,\alpha,\beta) : \alpha \overset{*}{\vdash}_M \beta\}$, הינה ב-**R**, מכיוון שכפי שהגדרנו מעבר בין קונפיגורציות, של מכונת טיורינג, קונפיגורציות אינן קטנות במהלך החישוב. כלומר, קיים מספר סופי (ניתן לחישוב) של קונפיגורציות באורך $|\beta|$, לכל היותר ולכן מספיק להריץ את המכונה M, מקונפיגורציה α, מספר זה של צעדים ולקבל את הקלט אמם המכונה M מגיעה לקונפיגורציה β במהלך חישוב זה (שים לב כי אם המכונה M, לא עצרה ולא הגיעה לקונפיגורציה ארוכה מ-$|\beta|$, קיימת קונפיגורציה שאליה הגיעה M, לפחות פעמיים

במהלך החישוב ולכן היא בלולאה אינסופית).

ג. $\{$קיימת קונפיגורציה התחלתית, α, וקיים מספר טבעי, k, כך ש– $\alpha \overset{*}{\underset{M}{\vdash}} \beta \not\downarrow^{k} : (<M>,\beta)\}$.

ד. $\{\forall \beta \exists q \exists k \quad \alpha \overset{*}{\underset{M}{\vdash}} \beta q \not\downarrow^{k} : (<M>,\alpha)\}$.

ה. $\{$ לכל קלט x, אם M עוצרת על x אז מספר הצעדים שהיא עושה הוא זוגי $: <M>\}$.