14. שלמות ב-NP

14.1 דיון בהגדרת המחלקות P ו-NP.

נזכיר כי **NP** הוגדרה כמחלקת השפות שיש לאבריהן (כלומר, לאברי השפות) עדויות "קצרות" הניתנות לאישוש ביעילות. כלומר, לכל שפה $L \in \mathbf{NP}$, קיים יחס דו-מקומי חסום פולינומיאלית, R_L, כך שבעית הזיהוי של היחס היא פולינומיאלית וכן מתקיים: $x \in L$ אמם קיים y כך ש- $(x,y) \in R_L$. y שכזה נקרא **עדות** לשייכות x לשפה L, ו- R_L נקרא **היחס המאפיין** את L. שים לב כי בהנתן זוג (x,y), קל לבדוק האם אמנם $(x,y) \in R_L$ מכיוון ש- R_L ניתן לזיהוי בזמן פולינומיאלי. כמו כן מובטח כי לכל $x \in L$ יש עדות לשייכות x לשפה L, ורק ל- $x \in L$ יש עדות כזו. שים לב כי לא מובטח שבהנתן $x \in L$, ניתן למצוא ביעילות (כלומר, בזמן פולינומיאלי) עדות כזו. כפי שנראה בתרגיל 14.1, אם ניתן למצוא עדויות כאלו ביעילות הרי ש- $L \in P$.

<u>תרגיל 14.1:</u>

תהי $L \in \mathbf{NP}$ ויהי R_L היחס הדו-מקומי המאפיין את L. הראה כי אם קיימת מכונת טיורינג פולינומיאלית, M, כך שלכל $x \in L$, המכונה M עוצרת עם פלט $M(x)$, המקיים $(x,M(x)) \in R_L$, אז השפה L הינה במחלקה **P**.

ניתן להוכיח באופן מלאכותי גם את הכיוון ההפוך כלומר:

<u>תרגיל 14.2:</u> תהי $L \in \mathbf{P}$. הוכח כי קיים יחס דו-מקומי ב- $PolyVerify$ (שנסמנו ב- R_L) אשר מאפיין את L (כלומר, $x \in L$ אמם קיים y כך ש- $(x,y) \in R_L$).

<u>הדרכה:</u> חשוב לדוגמא על היחס, $R_L \triangleq \{(x,1^{|x|}) : x \in L\}$. הראה כי R_L הינו ב- $PolyVerify$, מאפיין את L וניתן למצוא עדויות ביחס אליו ביעילות.

דרך סבירה להסתכל על העדות להיות $x \in L$, היא כעל "הוכחה להיות $x \in L$". המשמעות הנובעת מכך, היא שהמחלקה **NP** היא מחלקת (קבוצות) משפטים אשר יש להם הוכחות קצרות הניתנות לאישוש במהרה. קבוצות המשפטים בהם מדובר, הם מהצורה "$x \in L$" (לדוגמא, x הינו גרף המילטוני) וההוכחות הן העדויות y (לדוגמא, y הינו מסלול המילטוני ב- x). מה שהופך מחרוזות אלו ל"הוכחות", הוא קיום אלגוריתם יעיל אשר מוודא את תקפותן של ההוכחות (כלומר קיום מכונת טיורינג פולינומיאלית לזיהוי R_L). מכאן מתקבל כי **P** היא מחלקת קבוצות משפטים, אשר לא רק שיש להם הוכחות קצרות (הניתנות לאישוש ביעילות), אלא שגם ניתן למצוא ביעילות את ההוכחות הללו. השאלה בדבר היחס בין המחלקות **P** ו-**NP**, היא לפיכך השאלה הבסיסית "האם לגבי כל משפט שיש לו הוכחה קצרה הניתנת לאישוש ביעילות, ניתן גם למצוא הוכחה כזאת ביעילות?".

בניגוד להגדרה הנ"ל של המחלקה **NP**, אשר מבהירה את החשיבות של המחלקה **NP** ושל שאלת היחס בינה לבין המחלקה **P**, מקובלת יותר ההגדרה הבאה של **NP** (שהיא גם מקור הסימון **NP**). לצורך ההגדרה הבאה נאזכר גם את המושג הפיקטיבי של קבלה ע"י מכונת טיורינג לא-דטרמיניסטית (ראה סעיף 7.4).

<u>הגדרה 14.1:</u> **מכונה לא–דטרמיניסטית**, הינה שביעיה, כמו בהגדרת מכונת טיורינג (דטרמיניסטית) מלבד שפונקצית המעברים שלה היא התאמה מזוגות אל תתי-קבוצות של שלישיות מתוך $Q \times \Gamma \times \{L,R\}$.
מכונה לא-דטרמיניסטית, M, נקראת **פולינומיאלית**, אם קיים פולינום, P, כך שלכל קלט x, <u>כל</u> חישוב של M על קלט x, עוצר תוך לכל היותר $P(|x|)$ צעדים. זכור כי למכונה M שני מצבים סופיים: q_{ACC} (מצב קבלה) ו-q_{REJ} (מצב דחיה).
השפה המתקבלת ע"י מכונה לא–דטרמיניסטית M, L_M (תסומן M) היא קבוצת הקלטים <u>שקיים</u> חישוב של M אשר מקבל אותם. כלומר:

$$L_M \triangleq \{x : \exists \beta,\gamma \in \Gamma^* \; q_0 x \overset{*}{\underset{M}{\vdash}} \beta q_{ACC} \gamma\}.$$

<u>הגדרה 14.2:</u> המחלקה **NP** (Non-deterministic Polynomial-time) הינה קבוצת השפות המתקבלות באמצעות מכונת טיורינג לא-דטרמיניסטית פולינומיאלית. כלומר,

$$\mathbf{NP} \triangleq \{L : L \text{ קיימת מכונה לא–דטרמיניסטית פולינומיאלית המקבלת את } L\}$$

<u>אזהרה חוזרת:</u> שים לב כי העובדה שקיים חישוב של המכונה הלא-דטרמיניסטית, M, אשר מקבל את הקלט x, אינו מבטיח כי בהרצה ספציפית של המכונה אמנם נקבל את הקלט x. כמו כן, לא מובטח שאם נריץ את M פעמים רבות, יגדל "הסיכוי" שלנו לסיים במצב מקבל. אין פה מקום לדיון על סיכוי, משום שלא מוגדר מרחב הסתברות על הבחירות האי-דטרמיניסטיות של M. כל מה שמובטח הוא שקיימת סדרת בחירות (של צעדים אי-דטרמיניסטים) המובילה לקבלה של $x \in L$. לפיכך, המושג הפיקטיבי של מכונה לא-דטרמיניסטית פולינומיאלית, אינו מייצג חישוב יעיל.

<u>תרגיל מבהיר:</u> הראה כי המכונה הלא-דטרמיניסטית הבאה, מקבלת את השפה Σ^*. להלן פונקצית המעברים של המכונה:

$$\forall \sigma \in \Sigma \qquad \delta(q_0,\sigma) = \{(q_0,\sigma,R),(q_{REJ},\sigma,R)\}$$
$$\delta(q_0,\not{b}) = \{(q_{ACC},\not{b},S)\}$$

<u>הערה:</u> שים לב כי לכל $x \in \Sigma^*$, קיים חישוב של המכונה אשר מקבל את x (כלומר, עוצר במצב q_{ACC}). לעובדה

שרוב החישובים של המכונה M על קלט x מסתיימים במצב q_{REJ} (דחיה) אין כלל חשיבות. אם נריץ את המכונה M על x מספר פעמים, לא ישתנה מצב ענינים זה.

<u>תרגיל מבהיר (נוסף):</u> הראה כי המכונה הלא-דטרמיניסטית הבאה, מקבלת את השפה Σ^*. להלן פונקצית המעברים של המכונה:

$$\forall\sigma\in\Sigma \quad \delta(q_0,\sigma)=\{(q_0,\sigma,R)\}$$
$$\delta(q_0,\text{ב})=\{(q_{ACC},\text{ב},S),(q_{REJ},\text{ב},L)\}$$

<u>הערה:</u> שוב נדגיש כי מה שחשוב הוא קיום חישוב מקבל עבור כל x בשפה. לעובדה שמחצית מהחישובים הינם חישובים מקבלים, אין כלל חשיבות. אם נריץ את המכונה הנ"ל פעמים רבות על כל קלט רוב החישובים יהיו מקבלים, אך אין בכך הבטחה כי אמנם נקבל במהלך חישוב מסוים. שוב נדגיש כי הגדרת הקבלה של מכונה לא-דטרמיניסטית, אינה מתחשבת בכמות החישובים המקבלים קלט מסוים, אלא רק בשאלה "האם קיים חישוב המקבל את הקלט?". כל ההבהרות נחוצות בעיקר מכיוון שהמושג של מכונות לא-דטרמיניסטיות איננו טבעי ולפיכך הוא לעיתים מבלבל. החשיבות של המושג היא בכך שהוא מאפשר לדון במושגים אחרים. למשל, נוח טכנית להגדיר באמצעותו את המחלקה **NP**, כפי שעשינו בהגדרה 14.2. הטבעיות של הגדרה זו, <u>נובעת מן השקילות</u> שלה להגדרה 13.7.

<u>תרגיל (לא מטופש לשם שינוי):</u> הוכח כי קבוצת המספרים הפריקים (כלומר, שאינם ראשוניים) הינה שפה ב-**NP**.
<u>הדרכה:</u> בנה מכונה אי-דטרמיניסטית פולינומיאלית, אשר על קלט מספר N מנחשת פרוק שלו לשני מספרים N_1,N_2 כך ש $N_1\cdot N_2=N$ (וכמובן $N_1\neq 1$ וכן $N_2\neq 1$).

<u>תרגיל למתקדמים מאד:</u> הוכח כי קבוצת הראשוניים הינה שפה ב-**NP**.
<u>רמז:</u> אם P ראשוני וכן $P-1=p_1^{e_1}\cdot\ldots\cdot p_t^{e_t}$ (כאשר ה p_i-ים ראשוניים) אז קיים g כך שלכל i מתקיים ש $g^{\frac{P-1}{p_i}}$ אינו שווה ל-1 (מודולו P).

<u>משפט 14.1:</u> המחלקה **NP** כפי שהוגדרה בהגדרה 13.7, זהה למחלקה **NP** כפי שהוגדרה בהגדרה 14.2.

<u>הערה:</u> על סמך משפט 14.1 נוכל בעתיד להתייחס למחלקה **NP** על פי כל אחת מן ההגדרות הנ"ל.

<u>הוכחת המשפט:</u>

<u>כיוון I:</u> תהי L שפה במחלקה **NP**, על פי הגדרה 13.7. מובטח לנו קיום יחס דו-מקומי, R_L, המאפיין את L ומכונת טיורינג פולינומיאלית, M, אשר מזהה את R_L. נבנה מכונה לא-דטרמיניסטית פולינומיאלית, M', המזהה את L

(כלומר, נראה כי $L \in \mathbf{NP}$ על פי הגדרה 14.2). המכונה M' פותחת בניחוש של מחרוזת, y, שארכה חסום ע"י פולינום ב-x (הפולינום כאן הוא זה המובטח בחסם האורך של R_L). הניחוש מתבצע ע"י סדרה של צעדים לא-דטרמיניסטים, שבכל אחד מהם נרשמת אות על גבי סרט העבודה. בשלב שני, המכונה M' מפעילה את המכונה M על הקלט (x,y). שים לב כי M' היא פולינומיאלית, ומקבלת את x אם קיימת מחרוזת y כך שהמכונה M מקבלת את הזוג (x,y) (כלומר, אם $x \in L$). קיבלנו כי $L \in \mathbf{NP}$ על פי הגדרה 14.2.

<u>כיוון II:</u> תהי L שפה ב-**NP**, על פי הגדרה 14.2. מובטח לנו קיום מכונת טיורינג לא-דטרמיניסטית פולינומיאלית, M, אשר מקבלת את L. הרעיון הוא שניתן לקודד את הבחירות הלא-דטרמיניסטיות של המכונה M ולראות בהם עדות פוטנציאלית לשייכות $x \in L$. במילים אחרות, ניתן להתאים לכל סדרת צעדים אי-דטרמיניסטית, מחרוזת (כלומר, קידוד) כך שניתן "לשחזר" באופן חד-ערכי את סדרת הבחירות האי-דטרמיניסטיות מתוך מחרוזת זו. נניח תחילה, ללא הגבלת הכלליות, כי המכונה הלא-דטרמיניסטית המקבלת את L, היא לכל היותר בעלת דרגת אי-דטרמיניזם 2. היחס R_L יוגדר:

$$R_L \triangleq \{(x,y) : x \text{ מקבלת את } M \text{ שבהם } \text{הוא קידוד של צעדים לא–דטרמיניסטים } y\}$$

ברור כי לכל $x \in L$ קיים y מתאים, וכן עבור $x \notin L$ לא קיים y שכזה. המכונה הדטרמיניסטית הפולינומיאלית, המזהה את R_L, מפעילה את המכונה M על הקלט x תוך בחירת הצעדים הלא-דטרמיניסטים לפי הקידוד y ומקבלת את הזוג (x,y) אם M מקבלת את x (עם בחירת הצעדים הלא-דטרמיניסטים לפי הקידוד y). קיבלנו כי $L \in \mathbf{NP}$ לפי הגדרה 13.7. □

לפני שנסיים סעיף זה, נדגיש כי $\mathbf{NP} \subseteq \mathbf{R}$. למעשה ניתן לאמר יותר מכך.

<u>תרגיל 14.3:</u> הוכח כי לכל שפה, $L \in \mathbf{NP}$, קיימים פולינום P_L ומכונה (דטרמיניסטית) M, אשר מקבלת את L, כך שזמן הריצה של M על קלט x חסום ע"י $2^{P_L(|x|)}$.

<u>הדרכה:</u> עבור על כל העדויות האפשריות לטענה $x \in L$ (מהו הפולינום P_L?).

<u>תרגיל 14.4:</u> הסק מתרגיל 14.3 כי $\mathbf{NP} \subseteq \mathbf{R}$.

14.2 השאלה $\mathbf{NP} \stackrel{?}{=} \mathbf{P}$ ומה ניתן לאמר, על שפה $L \in \mathbf{NP}$, מבלי לדעת את התשובה

בנקודה זו, החשיבות של שאלת היחס בין המחלקות **P** ו-**NP** צריכה להיות ברורה. ברור כי $\mathbf{P} \subseteq \mathbf{NP}$, אך לא ידוע האם ההכלה חדה (כלומר, האם $\mathbf{P} \neq \mathbf{NP}$). ההשערה המקובלת היא כי $\mathbf{P} \neq \mathbf{NP}$, אולם לצערינו לא הוביל המחקר הנרחב בעשרים השנים האחרונות לתשובה לשאלה. חמור מכך, נראה כי הידע המדעי שבידנו רחוק מאד מלהיות

מסוגל לטפל בשאלה כזו. מכאן, שלא יהיה מנוס בשנים הקרובות אלא לחיות בעולם מחשבתי שבו התשובה לשאלה בסיסית כזו אינה ידועה והינה במעמד של השערה (או "אמונה") בלבד.

ההשערה היא ש- $\mathbf{P} \neq \mathbf{NP}$. העולם הממשי אינו מתחשב בקשיי "העולם התאורטי" ובפרט ממשיך לדרוש פתרונות יעילים לבעיות חיפוש אשר מתאימות למחלקה **NP**. כאשר אנו עומדים מול דרישה כזו (למציאת אלגוריתם יעיל לבעיה) חשוב שנדע תחילה האם הבעיה מתאימה למחלקה **P** (כלומר, האם קיים אלגוריתם יעיל לפתרונה). ברור (עקרונית) איך ניתן להראות כי הבעיה מתאימה למחלקה **P**. יש להציג מכונת טיורינג פולינומיאלית אשר מזהה את השפה המתאימה. אך איך נוכל להראות כי הבעיה (אשר מתאימה לשפה ב-**NP**) איננה מתאימה למחלקה **P**? הרי אם נוכל להראות דבר זה, נקבל הוכחה לכך ש- $\mathbf{P} \neq \mathbf{NP}$ (והוכחה לכך היא כנראה מעבר לכוחותנו). אולם, ישנה חשיבות מעשית לדעת שבעיה אינה ניתנת לפתרון באופן יעיל (כלומר, אינה מתאימה למחלקה **P**) משום שאז יש לנטוש את הנסיון למצוא לבעיה אלגוריתם יעיל ויש להתמקד בתחליפים (שעוד נדון בהם בפרק 17) כגון פתרון בעיות מצומצמות יותר (כלומר, מקרים פרטיים) מציאת פתרונות "מקורבים" ועוד...

ננסח כעת בבהירות את המשימה הניצבת לפנינו (במונחים של שפות). השערתנו היא כי $\mathbf{P} \neq \mathbf{NP}$ ונתונה לנו שפה $L \in \mathbf{NP}$, אשר אנו "חושדים" בה שאינה ב-**P**. איך נוכל להראות כי $L \notin \mathbf{P}$ (הרי זה יגרור $\mathbf{P} \neq \mathbf{NP}$)? במקום להוכיח כי $L \notin \mathbf{P}$, נסתפק בהוכחה מותנית. כלומר, נסתפק בהוכחה כי "תחת ההנחה $\mathbf{P} \neq \mathbf{NP}$ נובע כי $L \notin \mathbf{P}$". הוכחה זו היא לגבינו "כמעט הוכחה" כי $L \notin \mathbf{P}$, משום שהאמון שלנו בהשערה $\mathbf{P} \neq \mathbf{NP}$ הינו רב (למרות שאיננו מאמינים בכוחינו להוכיח את ההשערה).

שפה $L \in \mathbf{NP}$, אשר לגביה מוכח כי "$\mathbf{P} \neq \mathbf{NP}$ גורר $L \notin \mathbf{P}$" הינה במעמד מיוחד. גורלה כשייכת למחלקה **P**, כרוך באופן הדוק בגורל כל המחלקה **NP** ($L \in \mathbf{P}$ אם $\mathbf{P} = \mathbf{NP}$). לפיכך, טבעי לקרוא לשפה כזאת **שפה שלמה ב–NP** (מושג השלמות המוגדר כאן, רחב בהרבה מהמושגים שיוגדרו בהמשך ולמעשה לא נשתמש בו לעולם).

14.3 רדוקציות פולינומיאליות

איך אפשר להוכיח טענה מהסוג "אם $\mathbf{P} \neq \mathbf{NP}$ אז $L \notin \mathbf{P}$"?. נוכיח טענות כאלו בעזרת הטענה השקולה לוגית, "אם $L \in \mathbf{P}$ אז $\mathbf{P} = \mathbf{NP}$", ובעזרת המושג של רדוקציות פולינומיאליות (אשר יוגדר בהמשך).

דרך שכבר מוכרת לנו להוכיח שבעיה הינה קשה (במקרה שלנו, $L \notin \mathbf{P}$) היא להראות שניתן לעשות רדוקציה מבעיה קשה אחרת לבעיה זו. אולם במקרה הנוכחי, אין לנו אף בעיה קשה ממנה ניתן להתחיל. לכן נצטרך לעשות רדוקציה מכל בעיה במחלקה הנחשבת קשה (כלומר, מכל בעיה במחלקה **NP**) לבעיה שאת הקושי שלה אנו רוצים להוכיח.

מושג הרדוקציה צריך לשמר את "קלות החישוב" של הפתרון, כדי שאם קיימת רדוקציה מבעיה א' לבעיה ב' ובעיה ב' קלה לפתרון, ינבע מכך כי גם בעיה א' קלה לפתרון.

הגדרה 14.3 (רדוקצית Cook): נאמר שקיימת **רדוקצית טיורינג פולינומיאלית** (רדוקצית קוק) משפה L_1 לשפה L_2, אם קיימת מכונת אוב פולינומיאלית, M, כך שלכל x, המכונה M מקבלת את x עם גישה לאוב L_2 אם ורק אם $x \in L_1$.

הערה: שים לב כי "רדוקצית טיורינג פולינומיאלית" (רדוקצית קוק) הינה מקרה פרטי של רדוקצית טיורינג. התנאי הנוסף הוא שמכונת האוב רצה בזמן פולינומיאלי (ולא רק שהיא עוצרת תמיד). חשוב להדגיש שזמן החישוב של מכונת אוב, הינו מספר הצעדים שהמכונה עושה. הצעדים שעושה האוב על מנת לענות על שאלות המכונה, אינם ידועים וכמובן שאינם נלקחים בחשבון. שים לב כי בכיוון השני של הוכחת משפט 13.1, השתמשנו באופן סמוי ברדוקצית קוק.

תרגיל 14.5: הוכח כי אם L_1 ניתנת לרדוקצית קוק לשפה L_2 ואם $L_2 \in \mathbf{P}$, אז נובע $L_1 \in \mathbf{P}$.

הדרכה: התאם את הוכחת תרגיל 10.2. כל מה שנחוץ, הוא להוכיח כי המכונה M (מהוכחת תרגיל 10.2) אמנם פולינומיאלית. שים לב לכך שמחלקת הפולינומים סגורה תחת ארבע פעולות החשבון ותחת הרכבת פונקציות.

כמו בחלק הראשון של הקורס, יספיק לנו גם כאן מושג מצומצם יותר של רדוקציה פולינומיאלית. מושג זה הינו מושג הרדוקציה ע"י תרגום פולינומיאלי.

הגדרה 14.4 (רדוקצית Karp): נאמר שישנה **רדוקציה על–ידי תרגום פולינומיאלי** (רדוקצית קארפ) משפה L_1 לשפה L_2, אם קיימת פונקציה $f : \{0,1\}^* \to \{0,1\}^*$, הניתנת לחישוב בזמן פולינומיאלי, כך ש- $x \in L_1$ אם ורק אם $f(x) \in L_2$.

הערה: שים לב כי רדוקצית קארפ היא מקרה פרטי של רדוקציה ע"י תרגום. התנאי הנוסף הוא שפונקצית התרגום, f, ניתנת לחישוב בזמן פולינומיאלי (ולא רק שהיא ניתנת לחישוב).

תרגיל 14.6: הראה שרדוקצית קארפ הינה מקרה פרטי של רדוקצית קוק. כלומר, אם L_1 ניתנת לרדוקצית קארפ לשפה L_2, אז L_1 ניתנת לרדוקצית קוק לשפה L_2.

תרגיל 14.7: הוכח (ללא הסתמכות על תרגיל 14.6) כי אם L_1 ניתנת לרדוקצית קארפ לשפה L_2 ואם $L_2 \in \mathbf{P}$, אז $L_1 \in \mathbf{P}$.

<u>תרגיל 14.8:</u> הוכח כי כל שתי שפות ב-P ניתנות לרדוקצית קוק זו לזו.

<u>תרגיל 14.9:</u> הוכח כי כל שתי שפות לא טריביאליות ב-P ניתנות לרדוקצית קארפ זו לזו. שפות לא טריביאליות הינן כל השפות פרט ל-$\varnothing$ ו-Σ^* (רמז: הרדוקציה תשלח כל $x \in L_1$ למילה קבועה $w_Y \in L_2$, וכל $x \notin L_1$ למילה קבועה $w_N \notin L_2$).

מרגע זה ואילך, בכל פעם שנדבר על רדוקציה פולינומיאלית נתכוון לרדוקצית קארפ. בפרט, לא נעשה שימוש נוסף במושג הכללי יותר של רדוקצית קוק. מרגע שהגדרנו את מושג הרדוקציה הפולינומיאלית, נפתח מוצא לבעיה שהטרידה אותנו בתחילת הפרק (איך להוכיח טענות מהסוג: "אם **NP** $\neq$ **P** אז $L \notin$ **P**").

<u>הגדרה 14.5</u> (**NP**-קושי): שפה L הינה **NP**–**קשה** (**NP**–*hard*) אם לכל שפה $L' \in$ **NP**, קיימת רדוקציה פולינומיאלית מהשפה L' לשפה L.

<u>הערות:</u>

1. שים לב כי אם שפה **NP**-קשה הינה ב-**P**, אז **P** = **NP** (הוכח זאת!).
2. שים לב כי הגדרה 14.5 אינה דורשת כלל שהשפה L תהא במחלקה **NP**. לפיכך, **P** = **NP** לא גורר כי כל שפה **NP**-קשה הינה ב-**P**. בפרט, קיימות שפות **NP**-קשות שאינן ב-**P** (ואף אינן ב-**NP**) באופן בלתי תלוי בשאלה "האם **P** = **NP**?".

<u>תרגיל למתקדמים:</u> הוכח כי בעיית העצירה, $H \triangleq \{(<M>,x) : x$ **המכונה** M **מקבלת את** $x\}$, הינה **NP**-קשה (זכור כי $H \notin$ **R** ובפרט $H \notin$ **NP**).

<u>הדרכה:</u> בהנתן שפה $L' \in$ **NP**, נבנה רדוקציה פולינומיאלית מהשפה L' לשפה H. הרדוקציה שתסומן $f_{L'}$, תעתיק את המחרוזת x (שאנו מנסים לברר האם היא ב-L') אל הזוג $(<M_{L'}>,x)$, כאשר $M_{L'}$ היא מכונה קבועה (לא תלויה ב-x) אשר מקבלת את השפה L' ולא עוצרת על קלטים שאינם ב-L'. הראה כי מכונה כזו קיימת. ודא כי $f_{L'}$ ניתנת לחישוב בזמן פולינומיאלי וכי מתקיים $x \in L'$ אמם $f_{L'}(x) \in H$.

כפי שנאמר, הגדרת **NP**-קושי לא התייחסה כלל לשאלה "האם השפה הינה ב-**NP**?". כזכור, אותנו עניין "קיטלוג" של בעיות במחלקה **NP**.

<u>הגדרה 14.6</u> (**NP**-שלמות): שפה L הינה **NP**–**שלמה** (**NP**–*complete*) אם היא מקיימת:

(1) השפה L הינה במחלקה **NP**.

(2) השפה L הינה **NP**-קשה (כלומר, לכל שפה L' ב-**NP** קיימת רדוקציה פולינומיאלית מהשפה L' לשפה L).

השפות ה **NP**-שלמות, הן שפות אשר "גורלן" קשור בגורל המחלקה **NP** כולה (שים לב שזהו מושג מצומצם של שלמות בהשוואה למושג השלמות מסעיף 14.2).

<u>תרגיל 14.10:</u> הוכח כי אם השפה L הינה **NP**-שלמה, אז מתקיים: $L \in \mathbf{P}$ אם ורק אם $\mathbf{NP} = \mathbf{P}$.

<u>הדרכה:</u> אם $L \in \mathbf{P}$, אז מתכונת ה **NP**-קושי נובע כי $\mathbf{NP} \subseteq \mathbf{P}$ (לכל שפה $L' \in \mathbf{NP}$, קיימת רדוקציה מ-L' לשפה L ולכן $L' \in \mathbf{P}$) ומכיוון ש-$\mathbf{P} \subseteq \mathbf{NP}$ נובע $\mathbf{P} = \mathbf{NP}$. מצד שני, אם $L \notin \mathbf{P}$, הרי מהיות L במחלקה **NP** נובע כי $\mathbf{P} \neq \mathbf{NP}$.

14.4 קיום בעיה NP-שלמה

האם קיימות בכלל שפות **NP**-שלמות (האם קיימות שפות, אשר גורלן כרוך בצורה כה הדוקה בגורל כל המחלקה **NP**)? התשובה, כפי שנראה בקרוב, היא חיובית (שים לב כי לא ברור מראש כי התשובה חיובית. עצם העובדה שהגדרנו מחלקת שפות, אינה מבטיחה כי המחלקה אינה ריקה).

<u>משפט 14.2:</u> קיימת שפה **NP**-שלמה.

<u>הוכחה:</u> נגדיר את הגרסא הבאה של בעית העצירה. גירסא זו, הנקראת Bounded Halting, תתיחס למכונות טיורינג לא-דטרמיניסטיות ולקבלה תוך מספר <u>נתון</u> של צעדי חישוב.

$$BH \triangleq \{(\langle M\rangle, x, 1^n) : M \text{ הינה מכונה לא-דטרמיניסטית, המקבלת את } x \text{ תוך } n \text{ צעדים לכל היותר}\}$$

כלומר, $(\langle M\rangle, x, 1^n) \in BH$ אמם M הינה מכונת טיורינג לא-דטרמיניסטית וקיים חישוב של M המקבל את הקלט x תוך לכל היותר n צעדים. נראה כעת כי BH הינה **NP**-שלמה. יש להראות את קיום שני התנאים (שייכות BH ל-**NP** וקיום רדוקציה פולינומיאלית מכל שפה ב-**NP** ל-BH).

<u>טענה 14.1:</u> $BH \in \mathbf{NP}$.

<u>הוכחה:</u> נציג יחס דו-מקומי, R_{BH}, המאפיין את BH. $((\langle M\rangle, x, 1^n), (\alpha_0, \alpha_1, \ldots, \alpha_k)) \in R_{BH}$ אם α_0 היא הקונפיגורציה ההתחלתית של M על קלט x, $\alpha_i \vdash_M \alpha_{i+1}$ לכל $0 \le i < k$, α_k הינה קונפיגורציה סופית מקבלת, וכן

$k \le n$. ודא כי היחס R_{BH}, הינו חסום פולינומיאלית וכן כי ניתן לזהות את R_{BH} בזמן פולינומיאלי. אינטואיטיבית, העדות לכך ש-M מקבלת את x תוך n צעדים, הינה סדרת הקונפיגורציות של חישוב (של M על x) העוצר תוך n צעדים.

שים לב כי עבור קלט $(<M>,x,1^n) \in BH$, קיימת עדות באורך $n \cdot \max\{|x|,n\}$ לכל היותר, אשר מאששת את שייכות הקלט ל-BH. לפיכך קריטי כי השלם, n, ניתן בייצוג אונרי. ייצוג זה מבטיח כי העדות תהיה אכן פולינומיאלית בקלט. כלומר:

$$n \cdot \max\{|x|,n\} \le (n+|x|)^2 < |<M>,x,1^n|^2$$

<u>טענה 14.2:</u> לכל שפה $L \in \mathbf{NP}$, קיימת רדוקציה פולינומיאלית מהשפה L לשפה BH.

<u>הוכחה:</u> תהי $L \in \mathbf{NP}$ ויהיו M_L מכונה לא-דטרמיניסטית ו-P_L פולינום, כך ש-M_L מקבלת את L, ולכל קלט x, מספר הצעדים של המכונה M_L על הקלט x חסום ע"י $P_L(x)$. נציג את הרדוקציה הבאה, שתסומן f_L, מהשפה L לשפה BH: $f_L(x) = (<M_L>,x,1^{P_L(|x|)})$. ודא כי:

1. f_L ניתנת לחישוב בזמן פולינומיאלי (שים לב כי כתיבת $<M_L>$ אינה תלויה כלל בקלט x ולפיכך נעשית בזמן <u>קבוע</u> ע"י המכונה הפולינומיאלית המחשבת את f_L, וכתיבת $1^{P_L(|x|)}$ נעשית בזמן פולינומיאלי ב-$|x|$).
2. מתקיים: $x \in L$ אם $f_L(x) \in BH$ (עבור $x \in L$ קיים חישוב של M_L העוצר תוך $P_L(|x|)$ צעדים לכל היותר, ועבור $x \notin L$ לא קיים חישוב כזה).

<u>הערה:</u> שים לב כי לכל שפה L, מוצגת רדוקציה f_L אחרת. כמובן שכך צריך להיות. כל אחת מאותן רדוקציות ניתנת לחישוב בזמן פולינומיאלי.

<u>הערה למתקדמים:</u> בדרך כלל, מעדיפים ספרי הלימוד לפתוח את רשימת השפות ה-**NP**-שלמות בהוכחה כי SAT (ראה פרק 15) הינה **NP**-שלמה. הוכחה זו, שהינה מסובכת בהרבה מהוכחת משפט 14.2, תוצג בפרק 15. היתרון של SAT על BH, הינו ש-SAT היא בעיה "טבעית יותר". לו כל מה שאכפת לנו היה עצם הקיום של בעיות **NP**-שלמות, די היה לנו בשפה BH ובמשפט 14.2.

<u>הערה למתקדמים:</u> הבעיה BH נראית מאד לא טבעית (כמו כל המחלקה **NP** כאשר היא מוגדרת על-סמך מכונות לא-דטרמיניסטיות) אם שוכחים את ההתאמה בין מכונות לא-דטרמיניסטיות פולינומיאליות ובין בעית מציאת פתרונות שנכונותם ניתנת לאישוש ביעילות. את BH ניתן לפרש כבעיה הטבעית הבאה: בהנתן תיאור קומפקטי של בעיה (ע"י קידוד של מכונת טיורינג דטרמיניסטית המאוששת נכונות פתרונות) וקלט לבעיה, x, מצא פתרון לבעיה

(או הכרע האם קיים פתרון כזה) אשר ניתן לאישוש ב-n צעדים לכל היותר.

<u>תרגיל 14.11:</u> הוכח כי BH' (המוגדרת למטה) הינה **NP**-שלמה.

$\{M$ הינה מכונה דטרמיניסטית

וקיים y כך ש-M מקבלת את $x\#y$ ב-n צעדים לכל היותר : $(<M>,x,1^n)\}$ $\triangleq$ BH'

<u>הדרכה:</u> ברור כי $BH' \in$ **NP** (y הינו העדות עבור $(<M>,x,1^n) \in BH'$). הרדוקציה, מ-$L \in$ **NP** אל BH', תעתיק את x לשלישיה, $(<M_{R_L}>,x,1^n)$, כאשר M_{R_L} הינה המכונה המזהה את היחס R_L, המאפיין את השפה L ואילו n הוא חסם על זמן הריצה של M_{R_L} על זוגות (x,y). שים לב (עבור ההוכחה כי ההעתקה הינה פולינומיאלית) כי ניתן לחסום את n ע"י פולינום ב-$|x|$.