18. מחלקות סיבוכיות נוספות

פרק זה הינו מעבר לחומר הקורס "תורת החישוביות", וחלק גדול מהמשפטים בו ינתן ללא הוכחה. הסטודנט המעוניין להתעמק בנושאים אלו מופנה לקורס "תורת הסיבוכיות".

18.1 חישובים אקראיים יעילים (RP,BPP)

פעולה אלמנטרית שלא התרנו עד כה בדיונים שלנו היא **הטלת מטבע**. השאלה "האם סביר להניח כי מחשב יכול לבצע הטלת מטבע?" הינה לגיטימית. תשובתנו חיובית ומתבססת על האינטואיציה שלנו בדבר פעולות הנראות לנו אקראיות ועל האפשרות לגרום לפעולות כאלו (למשל: הטלת מטבע ממשית, קריאה במונה גייגר, מוליכות של דיודה במצב מעורער ועוד...). שים לב כי תשובה חיובית אינה מניחה כי העולם מכיל במהותו אקראיות, אלא שהוא נראה לנו ככזה. בפרק זה נניח תשובה חיובית (כלומר, נניח כי מחשב מסוגל לבצע הטלת מטבע).

מכונת טיורינג מטילת מטבעות, תוגדר באופן זהה להגדרת מכונה לא-דטרמיניסטית, אך היחס שלנו אל החישובים האפשריים שלה יהיה שונה. ללא הגבלת הכלליות, נניח כי למכונה ישנה דרגת אי-דטרמיניזם שווה ל-2 (כלומר, לכל זוג (q,σ) מתקיים $|\delta(q,\sigma)|=2$). זכור כי במקרה של מכונה לא-דטרמיניסטית, שאלנו "האם קיים חישוב של המכונה המוביל מקונפיגורציה אחת לקונפיגורציה שניה?". במקרה של מכונה אקראית, השאלה תהא כמותית ולא איכותית: "מהי ההסתברות שהמכונה עוברת מקונפיגורציה α לקונפיגורציה β?". ההסתברות שהמכונה עוברת מקונפיגורציה α לקונפיגורציה β ב-t צעדים, הינה היחס בין מספר החישובים בהם המכונה עוברת מקונפיגורציה α לקונפיגורציה β ב-t צעדים, לבין מספר החישובים האפשריים באורך t (כלומר, במקרה שלנו של דרגת אי-דטרמיניזם שווה ל-2, מספר החישובים הוא 2^t לכל היותר).

<u>תרגיל 18.1:</u> תהי M מכונת טיורינג בעלת מצב סופי יחיד $F=\{p\}$, אשר פונקצית המעברים שלה היא: $\delta(q_0,\sigma)=\{(q_0,\sigma,S),(p,\sigma,R)\}$ $\forall\sigma\in\{0,1,\not b\}$. מה ההסתברות שהמכונה עוצרת?

<u>תרגיל 18.2:</u> תהי M מכונת טיורינג שפונקצית המעברים שלה היא: $\forall\sigma\in\{0,1\}\ \ \delta(q_0,\sigma)=\{(q_0,\sigma,R),(q_0,\overline{\sigma},R)\}$ כאשר $\overline{\sigma}$ הינו הביט המשלים של σ, וכן $\delta(q_0,\not b)=\{(p,\not b,S)\}$, כאשר $p\in F$. תהי f_2 הפונקציה המחושבת ע"י המכונה הנ"ל. מהי ההסתברות ש-$f_2(x)$ שונה מהקלט x בדיוק ב-$\frac{|x|}{2}$ מקומות?

לגבי מכונות אקראיות המנסות לפתור בעיות חישוביות נבחין בין שתי קטגוריות. מכונות אשר אינן טועות (ידועות בשם Las Vegas) ומכונות שקיימת הסתברות (חסומה בדרך כלל) שהן טועות (ידועות בשם Monte Carlo). המכונה שתוארה בתרגיל 18.1, מחשבת את הפונקציה $f(\sigma\alpha)=\sigma$ באופן אקראי ואינה טועה לעולם (כלומר, היא

Las Vegas). המכונה שתוארה בתרגיל 18.2, מוצאת בהסתברות $\frac{1}{\sqrt{|x|}}$ בערך, על קלט באורך זוגי x, מחרוזת זוגית, y, באורך $|x|$ כך שהמחרוזות x ו-y מתאימות בדיוק ב-$\frac{|x|}{2}$ מקומות (כמובן שדוגמאות אלו "מגוחכות" ואין צורך באלגוריתמים אקראיים בכדי לפתרן. דוגמאות מעניינות יותר, כגון אלגוריתמים לבדיקת ראשוניות, ינתנו בקורסים מתקדמים). נסיים את הדיון בהצגת מספר מחלקות של שפות, הניתנות לזיהוי ע"י מכונות אקראיות בזמן פולינומיאלי. ההבדל בין המחלקות הוא בשאלה "האם מותרת טעות בעת קבלת קלט או דחיתו?" (הגדרה מדויקת של מחלקות של בעיות חישוב, מסורבלת יותר אך זהה ברוחה).

<u>הגדרה 18.1</u> (חישובים אקראיים ללא טעות): המחלקה **ZPP** (Zero-Probability Polynomial-time) הינה מחלקת שפות, כך שעבור כל $L \in$ **ZPP** קיימת מכונה אקראית M המקיימת:

(1) לכל $x \in L$, ההסתברות המותנית, שמכונה M מקבלת את הקלט x בתנאי שהיא עוצרת, הינה 1.

(2) לכל $x \notin L$, ההסתברות המותנית, שמכונה M דוחה את הקלט x בתנאי שהיא עוצרת, הינה 1.

(3) קיים פולינום Q, כל שלכל קלט באורך n, בהסתברות $\frac{1}{2}$ לפחות, המכונה M עוצרת תוך $Q(n)$ צעדים.

<u>הגדרה 18.2</u> (חישובים אקראיים עם טעות דו-כיוונית): המחלקה **BPP** (Bounded-Probability Polynomial-time) הינה מחלקת שפות, כך שעבור כל $L \in$ **BPP** קיימת מכונה אקראית פולינומיאלית, M, אשר מקיימת:

(1) לכל $x \in L$, ההסתברות שהמכונה M מקבלת את x הינה $\frac{2}{3}$ לכל הפחות.

(2) לכל $x \notin L$, ההסתברות שהמכונה M דוחה את x הינה $\frac{2}{3}$ לכל הפחות.

<u>הערה:</u> שים לב כי המכונה M בהגדרה 18.2 היא פולינומיאלית. כלומר, המכונה M <u>עוצרת תמיד</u> תוך מספר פולינומיאלי (באורך הקלט) של צעדי חישוב.

<u>תרגיל 18.3:</u> הוכח כי:

1. $L \in$ **BPP** אמם $\overline{L} \in$ **BPP**.
2. $L \in$ **ZPP** אמם $\overline{L} \in$ **ZPP**.

<u>תרגיל 18.4:</u> הוכח כי **ZPP** $\subseteq$ **BPP**.

<u>תרגיל 18.5:</u> תן דעתך על החשיבות של תנאי (3) בהגדרת **ZPP**. מדוע חשוב לדרוש הסתברות גדולה מ-0? מדוע לא נדרשה הסתברות 1?

<u>תרגיל 18.6:</u> מדוע בהגדרת **BPP** יכלנו להתעקש על כך שהמכונה עוצרת על כל קלט בזמן פולינומיאלי?

<u>הדרכה:</u> הראה טרנספורמציה של מכונות אשר בהסתברות $\frac{2}{3}$ עוצרות בזמן פולינומיאלי עם פלט נכון, למכונות העוצרות תמיד בזמן פולינומיאלי, והפלט שלהן נכון בהסתברות $\frac{2}{3}$ לכל הפחות.

<u>תרגיל 18.7:</u> הראה כי ההגדרה של **BPP** אינה משתנה כאשר מחליפים את $\frac{2}{3}$ ב- $\frac{1}{2}+\frac{1}{P(|x|)}$ (כאשר P הינו פולינום כלשהו). כמו כן, הראה כי ההגדרה אינה משתנה לגבי החלפה של $\frac{2}{3}$ ב- $1-2^{-|x|}$. תן דעתך על משמעות הדבר.

<u>הדרכה:</u> הרץ את המכונה הרבה פעמים והחלט לפי רוב.

<u>תרגיל 18.8:</u> מדוע לא ניתן, בהגדרת **BPP**, להחליף את $\frac{2}{3}$ ב-$\frac{1}{2}$?

<u>הדרכה:</u> הראה כי במקרה זה כל שפה היתה ב-**BPP**.

<u>הגדרה 18.3</u> (חישובים אקראיים עם טעות חד-כיוונית): המחלקה **RP** (Random Polynomial-time) הינה מחלקת שפות, כל שלכל $L \in$ **RP**, קיימת מכונה אקראית פולינומיאלית המקיימת:

(1) לכל $x \in L$, ההסתברות ש-M מקבלת את x הינה $\frac{1}{2}$ לכל הפחות.

(2) לכל $x \notin L$, ההסתברות ש-M דוחה את x הינה 1.

<u>הערה:</u> פרוש הדבר הוא שאם M מקבלת את x, אז בהכרח $x \in L$. לעומת זאת, יתכן כי M דוחה את x ובכל זאת $x \in L$.

<u>תרגיל 18.9:</u> הראה כי **ZPP** $\subseteq$ **RP** $\subseteq$ **BPP**.

<u>הדרכה:</u> עבור **ZPP** $\subseteq$ **RP**, קטע חישובים ארוכים והחלט שרירותית לדחות. עבור **RP** $\subseteq$ **BPP**, הרץ את המכונה פעמיים וקבל אם אחת הריצות לפחות היא ריצה מקבלת (ודחה אחרת).

<u>תרגיל 18.10:</u> הראה כי **RP** $\subseteq$ **NP**.

<u>הערה:</u> לא ידוע האם $\mathbf{BPP} \subseteq \mathbf{NP}$. שים לב כי הטיעון הקל של תרגיל 18.10 נכשל כאן.

<u>תרגיל 18.11:</u> שים לב כי $L \in \mathbf{RP}$ אינו גורר בהכרח $\bar{L} \in \mathbf{RP}$. הוכח כי אם $L \in \mathbf{RP}$ וכן $\bar{L} \in \mathbf{RP}$ אז $L \in \mathbf{ZPP}$.

<u>תרגיל 18.12:</u> הראה כי ההגדרה של **RP** אינה משתנה כאשר מחליפים את $\frac{1}{2}$ ב-$\frac{1}{P(|x|)}$ (כאשר P הוא פולינום כלשהו). כמו כן, הראה כי ההגדרה אינה משתנה לגבי החלפה של $\frac{1}{2}$ ב- $1-2^{-|x|}$. חזור על התרגיל לגבי **ZPP**.

18.2 בין P ל-NP

תמונת העולם תחת ההנחה $\mathbf{P} \neq \mathbf{NP}$ הינה:

* $\mathbf{NP}\text{–}complete \subsetneq \mathbf{NP}$
* $\mathbf{P} \subsetneq \mathbf{NP}$
* $(\mathbf{P} \cap \mathbf{NP}\text{–}complete) = \varnothing$

השאלה הנשאלת היא "האם קיימות ב-**NP** בעיות אשר אינן ב-**P** ואינן **NP**-שלמות?". התשובה תלויה כמובן בהגדרה המדויקת של מושג השלמות. לגבי הגדרת שלמות המתאימה לרדוקצית קוק (תחת ההנחה $\mathbf{P} \neq \mathbf{NP}$) ידוע כי התשובה חיובית (ולפיכך היא בהכרח חיובית עבור רדוקצית קארפ).

<u>משפט 18.1:</u> תחת ההנחה $\mathbf{P} \neq \mathbf{NP}$, קיימת שפה $L_0 \in \mathbf{NP} \setminus \mathbf{P}$ כך שלא כל שפה $L \in \mathbf{NP}$ ניתנת לרדוקצית קוק לשפה L_0.

<u>הערה:</u> שים לב להכרחיות התנאי במשפט. הרי אם $\mathbf{P} = \mathbf{NP}$, אז כל שתי שפות ב-**NP** ניתנות לרדוקצית קוק זו לזו (הדבר נכון אפילו ביחס לרדוקצית קארפ, אם השפות אינן טריביאליות).

כבר ציינו כי המחלקה **NP** כנראה אינה "סימטרית". כלומר, $L \in \mathbf{NP}$ אינו גורר בהכרח $\bar{L} \in \mathbf{NP}$ (בניגוד למחלקה **P** שהיא סימטרית). לפיכך, יש טעם להגדיר את המחלקה המשלימה ל-**NP**, הנקראת $co\text{–}\mathbf{NP}$, אשר מכילה את כל השפות שמשלימותיהן שייכות למחלקה **NP**.

<u>הגדרה 18.4:</u> $co\text{–}\mathbf{NP} \triangleq \{L : \bar{L} \in \mathbf{NP}\}$.

כמובן שאין ביכולתנו כיום להוכיח כי $\mathbf{NP} \neq co\text{–}\mathbf{NP}$, הוכחה כזו תגרור כי $\mathbf{P} \neq \mathbf{NP}$.

<u>תרגיל 18.13:</u> הראה כי אם $\mathbf{NP} \neq co\text{–}\mathbf{NP}$ אז $\mathbf{P} \neq \mathbf{NP}$.

הכיוון ההפוך לתרגיל אינו ידוע ונחשב לבעיה קשה בפני עצמה.

<u>בעית מחקר קשה מאד:</u> הוכח את הטענה, "אם $\mathbf{P} \neq \mathbf{NP}$ אז $\mathbf{NP} \neq co\text{–}\mathbf{NP}$", או הסבר את הקושי בהוכחה כזאת.

<u>תרגיל 18.14:</u> הראה כי אם $\mathbf{NP} \neq co\text{–}\mathbf{NP}$ אז $\mathbf{NP}\text{–}complete \cap co\text{–}\mathbf{NP} = \varnothing$.

תחת ההנחות $\mathbf{NP} \neq co\text{–}\mathbf{NP}$ ו- $\mathbf{P} \subsetneq \mathbf{NP} \cap co\text{–}\mathbf{NP}$ מתקבלת תמונת העולם הבאה:

18.3 מעבר ל-NP

כבר ראינו כי ניתן להתבונן על המחלקה **NP**, כעל מחלקת הטענות שיש להן הוכחות קצרות, אשר ניתנות לאישוש ביעילות. נזכור כי הוכחה נחשבה קצרה, אם ארכה אינו עולה על פולינום בגודל הטענה וכי יעילות זוהתה עם חישובים פולינומיאלים (דטרמיניסטים). לנוכח האמור בתת-הפרק 18.1, טבעי להרחיב כעת את המחלקה **NP** ע"י זיהוי אישוש יעיל עם חישובים אקראיים פולינומיאלים. נקודה זו חשובה לצורך ההרחבה הבאה שנבצע.

נתבונן שוב בשפות אשר נמצאות במחלקה **NP**. תהי $L \in \mathbf{NP}$ ויהי R_L הפרדיקט המתאים לשפה L. משמע שקיימת "דרך" לשכנע מוודא יעיל כי $x \in L$. דרך זו היא מתן מחרוזת y המקיימת $(x,y) \in R_L$. שים לב כי מחרוזת כזו קיימת תמיד (כי $x \in L$) וכי "מוכיח" אשר אינו מוגבל במשאבי חישוב, מסוגל תמיד למצוא אותה. לעומת זאת, אם $x \notin L$ אין y המקיים $(x,y) \in R_L$. שים לב כי הניסוח הנ"ל מציג באופן סמוי שני צדדים: "מוכיח" ו"מוודא". תפקיד המוכיח, שאינו מוגבל בכוח חישוב, הוא למצוא, בהנתן קלט $x \in L$, עדות y המקיימת $(x,y) \in R_L$ ולשלוח עדות זו למוודא. המוודא, בקבלו מחרוזת y מהמוכיח, בודק האם $(x,y) \in R_L$ ומשתכנע "בנכונות הטענה $x \in L$" אמם התנאי מתקיים. שים לב כי המוודא עובד בזמן פולינומיאלי באורך של x. חשוב להדגיש כי המוודא אינו סומך על המוכיח, אלא בודק את טענותיו. תהליך ההוכחה מופרד כאן לשני חלקים: החלק היצירתי והקשה של מציאת הוכחות והחלק הקל יחסית של בדיקת נכונות ההוכחות. מערכת ההוכחה מאפשרת להוכיח טענות נכונות (תנאי השלמות) אך אינה מאפשרת "להוכיח" טענות שקריות (תנאי הנאותות).

ההוכחות שנדונות בפסקה הקודמת, הן הוכחות שבהן אין אינטראקציה של ממש בין המוכיח למוודא. הודעה אחת ויחידה מוחלפת בין הצדדים והיא המחרוזת ("עדות") שהמוכיח שולח למוודא. ניסוח זה מתאים למושג ההוכחה, עבור הוכחות המופיעות בספר לימוד, אך לא להוכחות כפי שהן נמסרות בכיתת לימוד (אידאלית). בכיתת לימוד, משכנע המורה (המוכיח) את התלמידים (המוודא) באמיתות טענותיו, ע"י מתן תשובות לשאלות שהם מציגים לו. אינטראקציה כזאת מורכבת מסדרה של הודעות, הלוך וחזור, המוחלפות בין הצדדים.

<u>הגדרה 18.5:</u> **מכונה אינטראקטיבית**, היא מכונת טיורינג מטילת מטבעות, אשר יש לה זוג סרטים לתקשורת: סרט תקשורת לכתיבה וסרט תקשורת לקריאה. **זוג מכונות משוחח**, הן שתי מכונות אינטראקטיביות, אשר משתפות את סרט הקלט שלהן (המשמש לקריאה בלבד) ואת סרטי התקשורת שלהן, באופן שסרט התקשורת לקריאה של אחת הוא סרט התקשורת לכתיבה של השניה ולהיפך. **החישוב של זוג מכונות** כזה, מורכב משלבים. בכל שלב, מכונה אחת בלבד פעילה, בעוד שהמכונה האחרת במצב המתנה מיוחד. בסיום שלב, המכונה הפעילה עוברת למצב המתנה והמכונה שהיתה במצב המתנה הופכת פעילה. החישוב מסתים כאשר אחת המכונות עוצרת. סיבוכיות המכונות נמדדת כפונקציה של הקלט (תוכן סרט הקלט המשותף).

<u>הגדרה 18.6:</u> **הוכחה אינטראקטיבית לשפה** L, היא זוג מכונות משוחחות (P,V) [V הוא המוודא (Verifier) ו-P הוא המוכיח (Prover)], כך שקיים פולינום Q, כך שלכל קלט x, מספר הצעדים שהמכונה V מבצעת חסום ע"י $Q(|x|)$ ומתקיימים התנאים הבאים:

(1) <u>תנאי השלמות:</u> אם $x \in L$ אז המכונה V, לאחר אינטראקציה עם המכונה P על קלט x, עוברת למצב מקבל בהסתברות $\frac{2}{3}$ לכל הפחות.

(2) <u>תנאי הנאותות:</u> אם $x \notin L$ אז <u>לכל</u> מכונה P^*, המכונה V, לאחר אינטראקציה עם המכונה P^* על קלט x, עוברת למצב דוחה בהסתברות $\frac{2}{3}$ לכל הפחות.

<u>הערה:</u> שים לב כי הכרחי לדרוש, בתנאי הנאותות, קיום עבור כל מכונה P^* ולא רק עבור המכונה P. לו דרשנו קיום רק עבור המכונה P, פרוש הדבר היה שאנו משתמשים במכונה P כבאוב (כלומר, סומכים על קביעתה ש-$x \notin L$). תנאי השלמות והנאותות של הוכחה אינטראקטיבית, מקבילים לתנאי השלמות והנאותות של תורה לוגית קלסית. תנאי השלמות דורש שישנה אפשרות להוכיח טענות נכונות, בעוד שתנאי הנאותות דורש שאין אפשרות להוכיח טענות שיקריות.

<u>סימון:</u> נסמן ב-**IP** (Interactive Proofs) את מחלקת השפות שיש להן הוכחות אינטראקטיביות.

<u>דוגמא 18.1:</u> תהי GNI (Graph Non-Isomorphism) קבוצת הזוגות של גרפים <u>לא איזומורפים</u>. שני גרפים, $G_1(V_1,E_1)$ ו-$G_2(V_2,E_2)$, הם **איזומורפים**, אם קיימת העתקה חד-חד ערכית ועל, $\Pi : V_1 \to V_2$, כך ש-$(\Pi(u),\Pi(v)) \in E_2$ אמם $(u,v) \in E_1$. ודא כי $GNI \in co\text{–}\mathbf{NP}$. <u>לא ידוע</u> האם $GNI \in \mathbf{NP}$. נראה כי $GNI \in \mathbf{IP}$. להלן הרעיון להוכחה אינטראקטיבית: על קלט, גרפים G_1 ו-G_2, המוודא בוחר באקראי אינדקס $i \in \{1,2\}$ ועותק איזומורפי אקראי H של G_i. המוודא שולח את H למוכיח שאמור לנחש את i (כלומר, לנחש איזה משני הגרפים איזומורפי ל-H). הראה כי אם $(G_1,G_2) \in GNI$ המוכיח (שאינו מוגבל במשאבי חישוב) מסוגל לענות תמיד נכון, בעוד שאם $(G_1,G_2) \notin GNI$ (כלומר, G_1 איזומורפי ל-G_2) המוכיח מסוגל לענות נכון בהסתברות $\frac{1}{2}$ בדיוק (רמז: במקרה שהגרפים איזומורפים, אין בגרף H שום אינפורמציה על i. אזהרה: זוהי אינטואיציה בלבד ונחוץ טיעון מדויק). שים לב כי האינטראקציה הנ"ל אינה מראה כי $GNI \in \mathbf{IP}$ (תנאי הנאותות אינו מתקיים) לשם כך יש לבצע את האינטראקציה פעמיים (כאשר בכל פעם יש לבחור באקראי H שונה).

<u>תרגיל 18.15:</u> הראה כי $\mathbf{BPP} \subseteq \mathbf{IP}$ וכן כי $\mathbf{NP} \subseteq \mathbf{IP}$.

<u>תרגיל 18.16:</u> הראה כי הגדרת **IP** אינה משתנה אם מחליפים את $\frac{2}{3}$ ב- $\frac{1}{2}+\frac{1}{P(|x|)}$ (עבור P פולינום כלשהו) או ב- $1-2^{-|x|}$.

<u>תרגיל 18.17:</u> הוכח כי אם לשפה L ישנה הוכחה אינטראקטיבית, בה אין המוודא מטיל מטבעות, אז $L \in \mathbf{NP}$ (רמז: במקרה כזה, ה"עד" יהיה כל הטקסט של האינטראקציה המקורית).

לאחרונה הוכח כי המחלקה **IP** זהה למחלקת הסיבוכיות **PSPACE**, המוגדרת כמחלקת השפות אשר ניתנות לקבלה ע"י מכונות טיורינג אשר החישוב שלהן מוגבל <u>לשטח</u> (space) פולינומיאלי באורך הקלט. כלומר, $L \in \mathbf{PSPACE}$ אם קיים פולינום Q ומכונת טיורינג M, כך שלכל קלט x, המכונה M סורקת, במהלך החישוב על x, לכל היותר $Q(|x|)$ תאים מסרט העבודה שלה. בפרט, המחלקה **IP** מכילה את $co\text{–}\mathbf{NP}$. משמעות הדבר (לדוגמא) היא שניתן להוכיח ביעילות כי בגרף נתון לא קיים מעגל המילטוני.

תרגיל 18.18: הוכח כי המחלקה **PSPACE** סגורה תחת משלים. כלומר, אם $L \in$ **PSPACE** אז גם $\bar{L} \in$ **PSPACE**.

תרגיל 18.19: הוכח כי **BPP** $\subseteq$ **PSPACE** (רמז: סרוק את כל הטלות המטבע האפשריות).

תרגיל 18.20 (קשה): הוכח **IP** $\subseteq$ **PSPACE**.

הדרכה: יש לסרוק את כל השיחות האפשריות של ההוכחה האינטראקטיבית, תוך תשומת לב לכך שההודעה הנוכחית שהמוכיח שולח, יכולה להיות פונקציה של כל ההודעות שקיבל בעבר, אך אינה יכולה להיות פונקציה של הודעות שיקבל בעתיד.

משפט 18.2 (קשה מאד): **PSPACE** $\subseteq$ **IP**.

מסקנה: **P** $\subseteq$ **NP** $\subseteq$ **IP** = **PSPACE**.

מהמסקנה שלעי"ל נובע כי אם **NP** $\neq co-$**NP** אז **NP** $\neq$ **IP**. ודא כי אתה מבין מדוע.

הערה: לא ידוע אפילו האם **P** $\neq$ **PSPACE** ולפיכך יהיה קשה להוכיח אי-שיוויונים כגון **NP** $\neq$ **IP** או **BPP** $\neq$ **IP**.

מחלקות סיבוכיות נוספות בין **NP** לבין **PSPACE**, הן המחלקה **PH** (Polynomial-time Hierarchy) והמחלקה **#P** (Sharp Polynomial-time).

דרך אחת להגדיר את **PH** היא באמצעות נוסחאות מתחשיב פרדיקטים בעלות מספר סופי של כמתים. נוסחא בתחשיב היא מהצורה:

$$\exists y_1 \forall y_2 \exists y_3 \; \phi(x, y_1, y_2, y_3)$$

כאשר ה-y_i-ים אורכם פולינומיאלי ב-$|x|$ ואילו ϕ הינו פרדיקט בולאני הניתן לחישוב בזמן פולינומיאלי. שים לב כי כל שפה ב-**NP** ניתנת לאיפיון באמצעות נוסחא כזאת (כלומר, **NP** $\subseteq$ **PH**). בפרט $L \in$ **NP**, בעלת הפרדיקט הבוליאני המתאים לה R_L, מאופיינת ע"י:

$$L = \{x : \exists y \; R_L(x, y) = 1\}$$

תרגיל 18.21: הראה כי **PH** $\subseteq$ **PSPACE**.

דרך נוספת להגדיר את **PH** היא באמצאות מכונות אוב לא-דטרמיניסטיות בעלות גישה לאוב לשפה ב-**PH**. זוהי הגדרה רקורסיבית בעלת תנאי התחלה **NP** $\subseteq$ **PH**.

מחלקה נוספת, המוכלת במחלקה **PSPACE**, היא מחלקת השפות ה"מתאימות" לבעיות חישוב פונקציות הסופרות את מספר הפתרונות של יחסים דו-מקומיים הניתנים לזיהוי בזמן פולינומיאלי. לכל יחס דו-מקומי R, נגדיר את הפונקציה $\#_R : \{0,1\}^* \to N$, כדלקמן:

$$\#_R(x) \triangleq |\{y : (x,y) \in R\}|$$

מחלקת הפונקציות **P**#, היא קבוצת הפונקציות $\#_R$ אשר מתאימות ליחסים דו-מקומיים R, שיש להם מכונת טיורינג M המכריעה על קלט (x,y) האם $(x,y) \in R$, בזמן פולינומיאלי ב-$|x|$.

<u>תרגיל 18.22:</u> הראה כי כל פונקציה ב-**P**#, ניתנת לחישוב ע"י מכונת טיורינג (דטרמיניסטית) בעלת שטח עבודה חסום פולינומיאלית (כלומר, **P**#$\subseteq$**PSPACE**).

<u>תזכורת:</u> **POLY** היא קבוצת כל הפונקציות הניתנות לחישוב בזמן פולינומיאלי.

<u>תרגיל 18.23:</u> הראה כי אם **P**#$\subseteq$**POLY** אז **P** = **NP** (רמז: אם $L \in$ **NP** אז מתקיים $\#_{R_L}(x) > 0$ אמ"מ $x \in L$).

<u>תרגיל 18.24:</u> הראה כי **P**#$\subseteq$**POLY** (רמז: התאם לפונקציה f יחס R, כך ש- $(x,y) \in R$ אמ"מ $y \le f(x)$).

18.4 מחלקות סיבוכיות מפורשות

בניגוד למחלקות הסיבוכיות שעסקנו בהן עד כה, נעסוק בפרק זה במחלקות סיבוכיות בעלות חסם סיבוכיות (זמן או שטח) מפורש. למחלקות אילו יש חסרון שהן תלויות במודל החישוב הספציפי ושהן אינן סגורות תחת הרכבה אלגוריתמית. לשם פשטות, נציג הגדרות עבור מכונות טיורינג בעלות שלושה סרטים: סרט קלט, סרט פלט וסרט עבודה. נניח כי ניתן ללכת רק משמאל לימין על סרטי הקלט והפלט (הגדרות דומות אפשר להציג לגבי מודלים אחרים של חישוב, אולם המחלקות שתתקבלנה תהיינה שונות).

<u>הגדרה 18.7:</u> תהי $t : N \to N$ פונקציה כלשהי, $\text{Dtime}(t(n))$ היא מחלקת השפות הניתנות לקבלה ע"י מכונת טיורינג דטרמיניסטית, אשר עוצרת לאחר $t(|x|)$ על קלט x. באופן דומה, $L \in \text{Ntime}(t(n))$ אם קיימת מכונת טיורינג לא-דטרמיניסטית, העוצרת לאחר $t(x)$ צעדים על קלט x ומקבלת את L.

<u>הגדרה 18.8:</u> תהי $s : N \to N$ פונקציה כלשהי, $\text{Dspace}(s(n))$ (בהתאמה $\text{Nspace}(s(n))$) היא מחלקת השפות הניתנות לקבלה ע"י מכונת טיורינג דטרמיניסטית (בהתאמה לא-דטרמיניסטית) אשר משתמשת בלכל היותר $s(|x|)$ תאים של סרט העבודה שלה על קלט x.

<u>הערה:</u> שים לב להכרח להקצות סרט עבודה, על מנת לדבר על המחלקות $\text{Dspace}(s(n))$ ו-$\text{Nspace}(s(n))$ עבור פונקציות תת-לינאריות ($s(n)<<n$) בכדי שניתן יהיה לקרוא את כל הקלט מבלי לחרוג מסיבוכיות המקום.

<u>תרגיל 18.25:</u> הוכח:

א. $\text{Dtime}(t(n))\subseteq\text{Ntime}(t(n))$ (וכנ"ל לגבי space).

ב. $\text{Ntime}(t(n))\subseteq\text{Nspace}(t(n))$ (וכנ"ל לגבי D).

<u>תרגיל 18.26:</u> הוכח כי קיים קבוע k כך שלכל פונקציה $s(n)$ (אשר ניתנת לחישוב בשטח $s(n)$) מתקיים: $\text{Dspace}(s(n))\subseteq\text{Dtime}(n^k\cdot s(n)^k\cdot 2^{s(n)})$ (וכנ"ל לגבי N).

<u>הדרכה:</u> חשב את מספר הקונפיגורציות האפשריות בחישוב של מכונה המוגבלת בשטח $s(n)$, ותן דעתך לסיבוכיות הזמן של מימוש מונה באורך l.

<u>משפטים קשים</u> (הוכחות ינתנו בקורס "תורת הסיבוכיות"):

(1) לכל קבוע $c>0$,

$$\text{Dtime}(n^c)\subsetneq\text{Dtime}(n^{c+1})$$

$$\text{Dspace}(n^c)\subsetneq\text{Dspace}(n^{c+1})$$

(וכנ"ל לגבי N).

(2) $\text{Nspace}(s(n))\subseteq\text{Dspace}(s(n)^2+\log(n))$.

(3) לכל $s(n)\geq\log(n)$ המחלקה $\text{Nspace}(s(n))$ סגורה תחת משלים.