גירסה 1.01 – 14.7.2010

תורת החישוביות – חלק 2

ניר אדר

הבהרה: מסמך זה מסתמך בין היתר על הקורס "תורת החישוביות" בטכניון, אך איננו חומר רשמי של הקורס, אלא סיכום אישי בלבד. רשימת המקורות נמצאת בסוף המסמך. אתר הקורס "תורת החישוביות": http://webcourse.cs.technion.ac.il/236343

תוכן עניינים

1. מבוא

השאלה המרכזית בה יעסוק מסמך זה: מה ניתן לחשב ביעילות? על מנת לענות על שאלה זו נגדיר מדדים ליעילות של חישוב. המדד העיקרי והטבעי ביותר הינו **זמן החישוב** הנמדד על ידי מספר צעדי הריצה של מכונת טיורינג. זהו המדד בו נשתמש לרוב במסמך זה.
מדדים נוספים ליעילות: סיבוכיות זמן/מקום/סיבוכיות זמן-מקום מקבילי (קיימות לעתים בעיות הלוקחות זמן סדרתי דומה אך נבדלות משמעותית בזמן הדרוש לפתרון הבעיה בצורה מקבילית), אקראיות בחישובים אקראיים, מספר שינויי כיוון הראש של מ"ט ועוד.
כרגיל בנושאי חישוביות לא נדבר על זמן החישוב של קלט בודד או קבוצה סופית של קלטים. לכל קבוצה כזו ניתן להתאים מ"ט שתבצע את החישוב בזמן ליניארי $O(n)$. נמדוד את זמן החישוב על כל הקלטים בתור פונקציה מתמטית של אורך הקלט. תמיד ניקח סיבוכיות עבור המקרה הגרוע ביותר.

במסמך זה כל השפות אותן נציג ניתנות להכרעה. השאלה בה נעסוק כעת הינה– אילו שפות ניתן להכריע ביעילות.

1.1. הגדרת סיבוכיות של מכונת טיורינג

<u>הגדרה</u>: **סיבוכיות הזמן** של מ"ט M היא פונקציה חלקית $t_M(x)$ המוגדרת כך:

- אם M עוצרת על x, אז הפונקציה מחזירה את מספר צעדי החישוב של M על x.
- אם M אינה עוצרת על x, $t_M(x)$ אינה מוגדרת.

נשים לב כי $t_M(x)$ ניתנת לחישוב: על ידי מכונת טיורינג אוניברסלית שגם מחזיקה מונה עבור צעדי החישוב של M על x.

<u>הגדרה</u>: תהא $T:\mathbb{N}\to\mathbb{N}$ פונקציה מלאה. נאמר כי **זמן הריצה** של M **חסום** על ידי $T(n)$ אם $t_M(x)\le T(|x|)$ לכל $x\in\Sigma^*$. נאמר גם: M **עובדת/רצה** בזמן $T(n)$. כלומר: לכל קלט x באורך $|x|=n$ זמן הריצה של M על x חסום על ידי $T(n)$ וזהו חסם סופי $\Leftarrow$ <u>M עוצרת על כל קלט</u>.

<u>הגדרה</u>: נאמר כי מכונת טיורינג M היא **בעלת סיבוכיות פולינומית**, או נאמר כי M **יעילה**, אם קיים פולינום $P(n)=O(n^c)$ כך ש-M עובדת בזמן $P(n)$.

2.1. זיהוי חישוב יעיל עם זמן ריצה פולינומי

נשאלת השאלה – מדוע אנחנו מזהים חישוב יעיל עם זמן ריצה פולינומלי? להגדרה זו יתרונות וחסרונות.

יתרונות:

- פולינומים הן פונקציות בעלות עליה מתונה יחסית (ראו n^2 לעומת 2^n).
- מסתבר כי כמעט כל אלגוריתם שנחשב באופן מעשי יעיל– מקיים הגדרה זו.
- המושג אינו רגיש למודל: לכל זוג מודלים שראינו– אוסף הפונקציות שניתנות לחישוב יעיל בשני המודלים הוא זהה.

  התזה של אדמונדס/קרפ (גירסה חזקה ומורחבת של התזה של צ'רץ):
  - אוסף הפונקציות הניתנות לחישוב יעיל (=בזמן פולינומי) הוא זהה לכל מודל סביר וכללי של חישוב.
- פולינומים סגורים לחיבור, כפל, הרכבה וזה נוח לצורך הרכבה של אלגוריתמים/מכונות טיורינג.

חסרונות:

- **הזיהוי מכליל אלגוריתמים שאינם באמת יעילים**: האם אלגוריתם שרץ בזמן n^{200} הוא יעיל? לפי ההגדרה – כן. במציאות – וודאי שלא. עם זאת הנסיון מלמד שעבור בעיות "טבעיות" הניתנות לפתרון בזמן פולינומי, הפולינומים החוסם הוא לרוב קטן מאוד - $O\left(n^2\right), O\left(n^3\right)$.
- **הזיהוי אינו כולל את כל האלגוריתמים היעילים**: קיימים אלגוריתמים שעבור כל הקלטים השכיחים הינם יעילים, אולם במקרה הגרוע אינם יעילים, ולכן למרות השימושיות שלהם הם אינם כלולים בהגדרה.

<u>הגדרה</u>: $POLY$ היא <u>קבוצת כל הפונקציות</u> עבורן קיימת מכונת טיורינג יעילה המחשבת אותן.

<u>הגדרה</u>: P היא <u>קבוצת כל השפות</u> עבורן קיימת מכונת טיורינג יעילה המחשבת אותן.

<u>תכונות בסיסיות</u>:

1. $f \Leftarrow f \in POLY$ היא פונקציה מלאה (כי M פולינומית המחשבת אותה עוצרת על כל קלט).
2. $P \subseteq R$ (כי M פולינומית המקבלת את L בהכרח עוצרת על כל קלט, ולכן מכריעה את L).
3. $POLY$ סגורה תחת הרכבה. אם $f, g \in POLY$ אז גם $h = g \circ f \in POLY$.

3.1. קשר בין שפות לפונקציות

בחלק הראשון בסדרת המסמכים על חישוביות הגדרנו לכל פונקציה f שפה באופן הבא:

$$L_f \triangleq \{(x,y) \mid y = f(x)\}$$

בפרט ראינו כי מתקיים: f ניתנת לחישוב $\Leftrightarrow L_f \in \mathrm{RE}$.

נתעניין בשאלה – האם קיימת גם גירסה יעילה של המשפט? כלומר: $L_f \in P \Leftrightarrow f \in \mathrm{POLY}$

והתשובה: קיימת גירסה כזו, אולם נצטרך להגדיר את השפה המתאימה לפונקציה בצורה חדשה.

<u>הגדרה</u>: תנאי הכרחי לכך שפונקציה f תהיה שייכת ל-POLY הוא קיום פולינום $P(n)$ כך שלכל x יתקיים $|f(x)| \le P(x)$. פונקציה f עבורה קיים פולינום כזה תכונה **פונקציה חסומה פולינומית**.

<u>דגש</u>: אין קשר בין העובדה שפונקציה חסומה פולינומית לשאלה האם קיימת מכונת טיורינג המכריעה אותה. לדוגמא פונקציה העונה על בעית העצירה היא פונקציה חסומה פולינומית (הפלט שלה הוא0 או 1 – פלט באורך חסום).

האם $L_f \in P \Leftrightarrow f \in \mathrm{POLY}$? הכיוון הבא אכן מתקיים: $L_f \in P \Leftarrow f \in \mathrm{POLY}$. ניתן לראות זאת על ידי הוכחה דומה להוכחה המקורית עם פירוט נוסף עבור היעילות.

הכיוון $L_f \in P \Rightarrow f \in \mathrm{POLY}$ אינו מתקיים באופן כללי. ניקח לדוגמא פונקציה שאינה חסומה פולינומית: $f(x) = 1^{\#x}$ כאשר x זוהי מחרוזת בינארית ו-$\#x$ זהו המספר הטבעי המיוצג על ידי x. בהינתן (x,y) קל לזהות בזמן פולינומי באורך הקלט האם $y = f(x)$, אולם $f \notin \mathrm{POLY}$ כיוון שאינה חסומה פולינומית (אורך הפלט חסום על ידי מספר צעדי הריצה).

בנוסף – גם אם f היא חסומה פולינומית, לא בהכרח ניתן להוכיח כי $L_f \in P \Rightarrow f \in \mathrm{POLY}$. (זו הבעיה הפתוחה העיקרית בה נעסוק בהמשך).

לפיכך, נבנה הגדרה חדשה ל-L_f שאיתה יהיה נוח לעבוד.

<u>הגדרה</u>: $L_f' \triangleq \{(x,y) \mid y \text{ is a prefix of } f(x)\}$

<u>משפט</u>: בהינתן פונקציה f, מתקיים כי $L_f' \in P \Leftrightarrow f \in POLY$ וגם f חסומה פולינומית.

2. מכונת טיורינג אי-דטרמיניסטית

2.1. מכונת טיורינג אי-דטרמיניסטית

מ"ט אי דטרמיניסטית דומה למכונת טיורינג רגילה, אך פונקצית המעברים מוגדרת כך:

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

כלומר, לכל זוג $\left(q,a\right)$ מתקיים כי $\delta\left(q,a\right)=\left\{\left(p_0,b_0,d_0\right),\left(p_1,b_1,d_1\right)\right\}$.

הערות:

- ייתכן כי $\left(p_0,b_0,d_0\right)=\left(p_1,b_1,d_1\right)$ ולכן מ"ט דטרמיניסטית היא מקרה פרטי של מ"ט א"ד.
- **קונפיגורציה רגעית** מוגדרת בצורה דומה למעט העובדה שלכל קונפיגורציה יש 2 קונפיגורציות עוקבות, ולכן במקום מסלול חישוב יש לנו עץ חישוב– עץ בינארי של קונפיגורציות.

<u>הגדרה</u>: **עץ חישוב** הוא עץ מכוון אשר צמתיו מייצגים קונפיגורציות. השורש הוא הקונפיגורציה ההתחלתית ובניו של כל צומת מייצגים את הקונפיגורציה העוקבת של האב. עלים בעץ הם קונפיגורציות סופיות (עם מצב מקבל/דוחה).

<u>הגדרה</u>: אומרים שקלט x **מתקבל** על יד מ"ט א"ד M אם קיימת בעץ החישוב שנוצר על ידי הקונפיגורציה ההתחלתית $\left(q_0,x\right)$ קונפיגורציה מקבלת (קונפיגורציה עם מצב q_A). כמו כן, נגדיר את **השפה המתאימה למ"ט א"ד** באופן הבא: $L\left(M\right)\triangleq\left\{x\,|\,M\ \text{accepts}\ x\right\}$.

<u>הגדרה</u>: אומרים שקלט x **נדחה** על יד מ"ט א"ד M אם קיימת בעץ החישוב שנוצר על ידי הקונפיגורציה ההתחלתית $\left(q_0,x\right)$ קונפיגורציה לא מקבלת (קונפיגורציה עם מצב q_R) וכן לא קיים מסלול אל מצב מקבל.

הערות:

- לא הגדרנו את הפונקציה שמחשבת מ"ט א"ד. אנחנו בוחרים להתייחס רק לשאלות שהמכונה עונה כן/לא (האם מילה שייכת לשפה או לא), ולא להתעסק בפונקציה שהמכונה מגדירה.
- הגדרות שקולות להגדרת מ"ט א"ד:
  - מ"ט א"ד עם k סרטים.
  - אי דטרמיניזם לא מוגבל: $\delta:\left(\left(Q\setminus F\right)\times\Gamma\right)\to 2^{\left(Q\times\Gamma\times\{L,R,S\}\right)}$.
- בניגוד למכונת טיורינג הדטרמיניסטית, נוח להסתכל על מכונת טיורינג א"ד לא כעל מכונה אמיתית, אלא כעל דרך להגדרת שפה.

2.2. שקילות למודל הדטרמיניסטי

טענה: מודל מ"ט א"ד הינו שקול למודל הרגיל הדטרמיניסטי. כלומר מודל מ"ט א"ד מקבל בדיוק את כל השפות ב-RE.

כיוון אחד של הוכחת הטענה הוא קל. ראינו שכל מ"ט דטרמיניסטית היא מקרה פרטי של מ"ט א"ד, ולכן ברור שקבוצת מכונות הטיורינג הדרטמיניסטיות מוכלות בקבוצת מכונות הטיורינג הא"ד.

הוכחת הכיוון השני נעשית באופן הבא: תהא שפה L המתקבלת על ידי מ"ט אי דטרמיניסטית M.

נבנה מ"ט דטרמיניסטית M' המקבלת את אותה L.

הרעיון: נבצע חיפוש בעץ החישוב של M על הקלט x הנתון. החיפוש יהיה חיפוש BFS (על מנת שלא נמשיך על מסלול אינסופי וניתקע).

חיפוש BFS: $\varepsilon,0,1,00,01,10,11,...$ מחרוזות בינאריות שמייצגות לאן פונים בכל צעד בטיול על העץ. M' על קלט x הינה M' עם 3 סרטים:

1. בכל איטרציה נייצר על סרט 2 את המחרוזת הבאה על פי סדר לקסיקוגרפי, w.
2. נבצע סימולציה של M על x במסלול המתואר על ידי w.
3. אם הסימולציה מגיעה ל-q_A - עצור וקבל. אחרת עוברים לאיטרציה הבאה.

נכונות: צ"ל כי $L\left(M'\right)=L\left(M\right)$.

ניקח $x\in L\left(M\right)$: לפי הגדרת $L\left(M\right)$ קיים מסלול שבו M מקבלת את x $\Leftarrow$ קיימת w המתארת את המסלול. לפי הבניה, וכיוון שכל איטרציה היא סופית, נגיע בשלב כלשהו ל-w כזה.

מכאן: w' מקבלת ומתקיים $x\in L\left(M'\right)$.

ניקח $x \notin L(M)$: לא קיים מסלול שבו M מקבלת את x $M' \Leftarrow$ אינה עוצרת, ובפרט איננה מקבלת ומתקיים $x \notin L(M')$.

מימוש הסימולציה דורש 3 סרטים כאשר בסרט 1 נמצא הקלט x, בסרט 2 נמצאת בכל שלב המילה w וסרט 3 משמש לצורך ביצוע הסימולציה.

2.3. המחלקה NP

הגדרה: מ"ט א"ד תיקרא **פולינומית** אם קיים פולינום $P(n)$ כך שלכל x, M עוצרת על x בכל מסלולי החישוב תוך $P(|x|)$ צעדים.

במילים אחרות: מכונה א"ד תיקרא פולינומית אם גובה עץ החישוב שלה חסום על ידי פולינום.

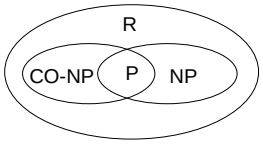
הגדרה: **המחלקה NP** (הגדרה 1) היא אוסף כל השפות שקיימות עבורן מ"ט א"ד פולינומית.

באופן דומה נגדיר את **המחלקה CO-NP** כך: $\text{CO-NP} = \{L \mid \overline{L} \in \text{NP}\}$.

אבחנה: $\text{P} \subseteq \text{NP} \subseteq \text{R}$ וכן $\text{P} \subseteq \text{CO-NP} \subseteq \text{R}$.

הבעיה הפתוחה המרכזית (נוסח 1): האם $P = NP$?

תמונת העולם המשוערת:



נשים לב לאבחנה לגבי השאלה הפתוחה: ראינו שקבוצת השפות הניתנת לחישוב במודל הדטרמיניסטי זהה לקבוצת השפות הניתנת לחישוב במודל האי דטרמיניסטי.
כלומר: כל שפה שב-NP נמצאת גם ב-RE.

השאלה הפתוחה היא בעצם האם קבוצת השפות הניתנות לחישוב יעיל במודל הדטרמיניסטי זהה לקבוצת השפות הניתנות לחישוב יעיל במודל האי דטרמיניסטי.

<u>דוגמא</u>: נגדיר את הקבוצה הבאה:

Composites = { m | m הוא ייצוג בינארי של מספר טבעי לא ראשוני }

הקבוצה המשלימה לה היא קבוצת המספרים הראשוניים: $\text{Primes} = \overline{\text{Composites}}$

<u>טענה</u>: $\text{Composites} \in \text{NP}$.

<u>הוכחה</u>: נבנה מ"ט א"ד המקבלת את Composites. המכונה M מקבלת קלט m בן n ביטים.

פעולת המכונה:

1. כותבת על הסרט השני מספר g בן $\left\lceil \frac{n}{2} \right\rceil$ ביטים (מספר היכול להתחיל באפסים) על ידי שימוש באי דטרמיניזם. המימוש הוא על ידי בחירת אחת משתי אפשרויות הכותבות0 או 1 בהתאמה ומזיזות את הראש ימינה. הבחירה נעשית על ידי ניחוש.

for i = 1 to $\left\lceil \frac{n}{2} \right\rceil$ do: Guess bit b. Write b on second tape and move right.

לאחר השלב הראשון, עוברים לפאזה דטרמיניסטית:

2. אם $g=0$ או $g=1$ עוצרים ודוחים.
3. בודקים אם $g \mid m$ (על ידי אלגוריתם פולינומי לחלוקה). אם כן עוצרים ב-q_A ואחרת עוצרים ב-q_R.

<u>נכונות</u>:

1. אם m אינו ראשוני, אז קיים $1 < g < \sqrt{m}$ המחלק אותו, ולכן קיים מסלול ב-M המקבל את m.
2. אם m ראשוני, אז אין $1 < g < \sqrt{m}$ המחלק אותו, ולכן כל מסלולי החישוב דוחים.
3. הזמן לביצוע החלוקה הוא פולינומיאלי (אפשר לבצע ב-$O\left(n^2\right)$) ולכן הזמן הכולל של החישוב הוא פולינומיאלי.

מכאן: $\text{Composites} \in \text{NP}$ ולכן מתקיים גם $\text{Primes} \in \text{CO-NP}$.

דוגמא:

בעיית המעגל ההמילטוני: מעגל המילטוני בגרף הוא מעגל העובר בדיוק פעם אחת בכל קודקוד וחוזר למקור.

HC= { G | G גרף לא מכוון המכיל מעגל המילטוני }

טענה: $HC \in NP$

הוכחה: נבנה מ"ט א"ד הרצה בזמן פולינומי ומקבלת את השפה HC.

1. המכונה מנחשת פרמוטציה π של הצמתים $\{1,...,n\}$ (בכל שלב המכונה כותבת מספר נוסף כלשהו מבין אלו שלא נכתבו). אחר כך המכונה בודקת האם קיימות בגרף קשתות מ-$\pi(1)$ אל $\pi(2)$, מ-$\pi(2)$ אל $\pi(3)$ וכך הלאה, עד בדיקה האם קיימות קשתות מ-$\pi(n)$ אל $\pi(1)$.
2. אם כל הקשתות קיימות, המכונה עוצרת ב-q_A, ואחרת המכונה עוצרת ב-q_R.

נכונות:

1. אם קיים מעגל המילטוני בקלט G, אז קיים מסלול חישוב של המכונה המנחש את המעגל הזה ומקבל. אם לא קיים מעגל המילטוני ב-G, אז כל מסלולי החישוב דוחים.
2. זמן החישוב של מסלול כולל ניחוש $\pi(1),...,\pi(n)$ ובדיקת הקשתות המתאימות – כל הצעדים הם בזמן פולינומיאלי ולכן גם הזמן הכולל.

אינטואיציה

P היא מחלקת השפות עבורן קל למצוא הוכחת שיוך לשפה בזמן פולינומיאלי. ניתן לעבור על מסלול החישוב היחיד של מ"ט דטרמיניסטית ולבדוק אם הוא מסיים בקבלה– זו בדיקה פולינומיאלית האם הקלט בשפה.

NP היא מחלקת השפות עבורן ניתן לוודא ביעילות הוכחת שיוך לשפה (אבל לא בהכרח למצוא ביעילות הוכחה כזו). ההוכחה היא הניחוש הנכון שגורם למכונה לקבל. מעבר על כל הניחושים יכול לקחת זמן אקספוננציאלי. דוגמא לניחוש הנכון: מספר שמחלק את הקלט עבור Composites, המעגל ההמילטוני בגרף עבור HC.

טענה: שפה L ניתנת לקבלה בזמן פולינומי אם ורק אם היא ניתנת לדחייה בזמן פולינומי.

הוכחה: נוכיח את הטענה על ידי בנייה.
תהי L הניתנת לקבלה בזמן פולינומי. תהי M המכונה המתאימה הפותרת אותה, ויהי P(n) הפולינום החוסם.

נבנה מכונה M' שתריץ ריצה מבוקרת את M למשך P(n) צעדים.
במהלך צעדים אלו:

- אם M דוחה, אז M' מקבלת.
- אם M מקבלת, אז M' דוחה.

אם M לא מקבלת או דוחה בזמן פולינומי, אז M' מקבלת.

נכונות: אם M דוחה אם מקבלת ב-P(n) הצעדים הראשונים, M' עושה ההפך, כנדרש.
מכיוון שנתון ש-M ניתנת לקבלה בזמן פולינומי, אחרי P(n) צעדים אנחנו יודעים בוודאות שאם M עדיין לא קיבלה היא גם לא תקבל. לפיכך M' יכולה לקבל את המילה.

זמן פולינומי: המכונה M' פועלת בזמן פולינומי: היא מבצעת את הפעולה של M, ואנחנו מניחים שההרצה המבוקרת לוקחת זמן פולינומי (קריאת הקידוד <M> ופעולה לפי מה שכתוב שם). הרכבת פולינומים נותנת אף היא פולינום.

3. בעיות חיפוש

הגדרה: בהינתן יחס דו מקומי $S \subseteq \Sigma^* \times \Sigma^*$ אומרים שמכונת טיורינג M **פותרת את בעיית החיפוש** של S אם לכל x: אם קיים y כך ש- $(x,y) \in S$, אז M עוצרת ב- q_A עם y כזה כפלט. אם לא קיים y כנ"ל, אז M לא עוצרת.

חיפוש יעיל יוגדר להיות חיפוש המבוצע על ידי מכונת טיורינג פולינומית.

הגדרה: נאמר שיחס S **ניתן לחיפוש יעיל** אם קיימת מ"ט פולינומית שלכל קלט x: אם קיים y כך ש- $(x,y) \in S$ אז M עוצרת ב- q_A עם y כזה כפלט. אם לא קיים y כנ"ל, אז M עוצרת ב- q_R.

תזכורת: בעיית הזיהוי של יחס $\equiv$ קבלת L_S. זיהוי יעיל $L_S \in P \equiv$.

מה הקשר בין זיהוי יעיל לחיפוש יעיל? חיפוש יעיל $\not\Leftarrow$ זיהוי יעיל. לדוגמא: עבור השפה $L = \left\{ \left(\langle M_1 \rangle, \langle M_2 \rangle \right) \middle| L(M_1) = L(M_2) \right\}$ ניתנת לחיפוש יעיל – לוקחים את אותה מכונה, אולם כמו שהוכח במסמך הראשון בסדרה זו שפה זו אינה שייכת כלל ל-RE.

האם מתקיים כי זיהוי יעיל $\Leftarrow$ חיפוש יעיל? קיימים יחסים עבורם זיהוי יעיל אכן גורר חיפוש יעיל.

הגדרה: יחס S נקרא **יחס חסום פולינומית** אם קיים פולינום P כך שלכל x, y המקיימים ש- $(x,y) \in S$, מתקיים כי $|y| \le P(|x|)$.

נשים לב כי בדומה לפונקציה חסומה פולינומית – הגדרנו כאן מושג כללי לגבי כל יחס.

הבעייה הפתוחה המרכזית (נוסח 2): האם לכל יחס חסום פולינומית S, מתקיים כי זיהוי יעיל גורר חיפוש יעיל?

דוגמא

$$Factor = \left\{ (n,m) \,|\, 1 < m < n, n \bmod m = 0 \right\}$$

בהינתן m, n קל לבדוק שייכות ליחס בזמן פולינומי (בעיית הזיהוי). נתון n, צריך למצוא m כנידרש – בעית Factoring – לא ידוע לה אלגוריתם פולינומי. (בעית חיפוש).

הגדרה: **המחלקה NP (הגדרה 2)**: שפה L שייכת ל-NP אם קיים יחס דו מקומי R_L כך ש:

1. R_L הוא יחס חסום פולינומית.
2. R_L ניתן לזיהוי פולינומי.
3. $L=\{x\mid \exists y,(x,y)\in R_L\}$

משפט: שתי ההגדרות של NP שקולות.

הוכחה: כיוון 1:

תהי שפה L ונניח ש-$L\in NP$ לפי הגדרה 2. מכאן קיים R_L המקיים את התנאים הנ"ל.

ניצור M שהיא מ"ט א"ד. המכונה M על קלט x תתנהג באופן הבא:

1. תנחש y שאורכו לכל היותר $q(|x|)$ (מובטח שקיים מכיוון ש-R_L חסום פולינומית).
2. תבדוק האם $(x,y)\in R_L$ ע"י המכונה הפולינומית המובטחת לזיהוי R_L.
3. אם כן, נקבל. אחרת נדחה.

M היא פולינומית: ניחוש דורש $O(q|x|)$ צעדים.

$\exists y\ x\in L \Leftarrow$ כך ש-$(x,y)\in R_L$ ומכאן קיים מסלול שבו M מנחשת את y ומקבלת את x ומכאן מתקיים $x\in L(M)$.

$x\notin L \Leftarrow$ לא קיים y כזה $\Leftarrow$ בכל מסלול M דוחה ומכאן $x\notin L(M)$.

כיוון 2: $L\in NP \Leftarrow$ קיימת מ"ט M המקבלת את L בזמן פולינומי ויהי P הפולינום. נגדיר את היחס הבא:

$R_L=\{(x,y)|$ y מזהה מסלול המתואר על ידי המחרוזת x במסלול את M מקבלת }

- R_L חסום פולינומילאלית כי אורך המסלול חסום מכיוון ש-M פולינומית.
- ראינו שסימולציה של M במסלול y לוקחת $O(|y|)$ צעדים, ומכאן R_L ניתן לזיהוי פולינומי.
- $x\in L \Leftrightarrow$ יש ל-M מסלול מקבל על x $\Leftrightarrow \exists y,(x,y)\in R_L$.

משפט: שני הנוסחים של הבעיה הפתוחה המרכזית הינם שקולים.

4. רדוקציות פולינומיאליות

4.1. הגדרה

הגדרה: אומרים שפונקציה f היא **רדוקציה פולינומית** מ-L_1 ל-L_2 אם $f \in POLY$ וכן f היא רדוקציה תקפה: $x \in L_1 \Leftrightarrow f(x) \in L_2$. אם קיימת f כזו עבור L_1, L_2 נסמן $L_1 \leq_P L_2$

טענה: $L_1 \in P \Leftarrow \begin{cases} L_1 \leq_P L_2 \\ L_2 \in P \end{cases}$.

טענה: $L_1 \leq_P L_3 \Leftarrow L_1 \leq_P L_2, L_2 \leq_P L_3$.

4.2. שפות NP-שלמות

הגדרה: שפה L נקראת **NP-שלמה** אם מתקיים:

1. $L \in NP$.
2. לכל $L' \in NP$, $L' \leq_P L$.

טענה: תהי L שפה NP-שלמה. מתקיים כי $P = NP \Leftrightarrow L \in P$.

הוכחה:

$\Rightarrow$: L היא שפה NP-שלמה. בפרט $L \in NP \Leftarrow L \in P$ מההנחה $P = NP$.

$\Leftarrow$: מספיק להוכיח כי $NP \subseteq P$. תהי L' ב-NP. נראה כי $L' \in P$. משלמות L קיימת $L' \leq_P L$.

מההנחה $L \in P \Leftarrow L' \in P$.

4.3. דוגמא לשפה NP-שלמה: שפת העצירה החסומה

תהי M מכונה אי דטרמיניסטית. נגדיר את השפה הבאה:

$$BH = \left\{ \left(\left\langle M \right\rangle, x, 1^t \right) \mid M \text{ accepts } x \text{ in at most } t \text{ steps} \right\}$$

נטען כי $BH \in NPC$ (NP-Complete = NPC).

כשאנחנו צריכים להוכיח ששפה היא NP-שלמה אנחנו צריכים להוכיח כי: 1. השפה שייכת ל-NP. 2. קיימת רדוקציה פולינומית מכל שפה ב-NP אליה.

נוכיח כי BH ב-NP. נוכיח זאת על ידי מכונה שמכריעה האם מילים בשפה.

1. ננחש מחרוזת בינארית באורך t.
2. נריץ את $\left\langle M \right\rangle$ על הקלט x. בכל פעם שיהיה לנו פיצול אי דטרמיניסטי לבחור, נעשה זאת לפי התו הבא במחרוזת הבינארית.
3. נעצור אחרי t צעדים, ונדחה אם $\left\langle M \right\rangle$ לא הגיעה למצב מקבל.

הראנו מכונה אי דטרמיניסטית המכריעה את השפה ולכן היא ב-NP.

סקיצת הוכחה שהשפה BH היא NPC:

1. תהא L שפה ב-NP.
2. נגדיר את הפונקציה הבאה: $f(x) = \left\langle M_L \right\rangle, x, 1^{P_L(|x|)}$. ($L$ היא שפה ב-NP ולכן קיימת מכונת טיורינג אי דטרמיניסטית M_L שמכריעה אותה בזמן P_L).
3. טענה 1 (בלא הוכחה): $x \in L \Leftrightarrow f(x) \in BH$.
4. טענה 2 (בלא הוכחה): $f(x)$ היא רדוקציה פולינומית מכל שפה ב-NP אל BH.

4.4. דוגמא: בעיית כיסוי בצמתים: VC – vertex cover

בהינתן גרף $G=(V,E)$, קבוצת צמתים $B\subseteq V$ נקראת **כיסוי בצמתים** אם לכל קשת בגרף $e=(a,b)\in E$ מתקיים ש-$a\in B$ או $b\in B$.

דוגמאות:

- גרף בצורה של כוכב (כל הצמתים מחוברים לצומת אחד מרכזי)– מספיק צומת אחד – הצומת המרכזי.
- גרף מלא (קיימת קשת בין כל זוג צמתים) בעל t צמתים – גודל B יהיה $t-1$ צמתים.

נגדיר את השפה VC באופן הבא:

$VC=\{G,k\mid$ יש כיסוי של G על ידי k צמתים$\}$

<u>טענה:</u> $VC\in NPC$

א. $VC\in NP$:

1. תנחש קבוצה $B\subseteq V$ (למשל ע"י ניחוש של $|V|$ ביטים, והגדרה כי צומת הוא ב-B אם ניחשנו עבורו "1").
2. נבדוק ש-$|B|\le k$ ושלכל קשת $e\in E$ לפחות אחד מצמתיה ב-B. אם כן נקבל ואחרת נדחה.

נכונות - מהגדרת הכיסוי. סיבוכיות: $p(|E|,|V|)$.

<u>טענה</u>: תהי $L\in NP$ וגם $L_1\in NPC$. אם מתקיים $L_1\le_P L$ אז: $L\in NPC$

הוכחה: נראה ש-L היא NP-שלמה:

א. נתון: $L\in NP$.

ב. לכל $L'\in NP$ מתקיים כי $L'\le_P L$. ידוע כי $L'\le_P L_1$ כי L_1 שלמה, וכן $L_1\le_P L$ מהנתון. מתכונת הטרנזיטיביות אנחנו מקבלים: $L'\le_P L$.

4.5. דוגמא: כיסוי קבוצות SC - Set-Cover

נתון:

- קבוצה W
- תתי קבוצות $C_1, C_2, ...C_m \subseteq W$ ומספר k.

השאלה עליה נרצה לענות היא האם קיימות k תתי קבוצות $C_{i1},...,C_{ik}$ כך ש $\bigcup_{y=1}^{k} C_{iy} = W$.

טענה: $SC \in NPC$

1. $SC \in NP$ (ננחש $i_1,...,i_k$ ונבדוק – ובהתאם נקבל או נדחה).
2. $VC \in NPC$ (ראינו בדוגמא הקודמת).
3. נגדיר את הפונקציה הבאה: $f(G,k) = \left(E, C_1, C_2, ..., C_{|V|}, k\right)$, כאשר נגדיר כי לכל צומת $a \in V$ מתקיים $C_a = \{e \mid e$ אחד הצמתים של $a\}$.

נראה כי f רדוקציה פולינומית.

$(G,k) \in VC \Leftrightarrow$ קיים כיסוי V בגודל k לגרף אמ"מ קיימות k קבוצות $C_{i1},...,C_{ik}$

כך ש- $f(G,k) \in SC \Leftrightarrow \bigcup_{y=1}^{k} C_{iy} = E$.

בנוסף, f היא פונקציה הפועלת בסיבוכיות פולינומיאלית.

6.4. SAT, 3-SAT

הגדרות:

- **משתנה בוליאני**: משתנה שיכול לקבל 0 או 1.
- **ליטרל**: משתנה בוליאני או שלילה שלו (יסומנו x , $\bar{x}$) נקרא בשם ליטרל.
- **פסוקית**: רשימה של ליטרלים עם סימן $\vee$ ביניהם. (לעיתים במקום $\vee$ מסמנים +).
  לדוגמא: $(x_1 \vee x_2 \vee \bar{x}_3)$.
- **פסוק CNF**: רשימה של פסוקיות עם $\wedge$ ביניהן.
  לדוגמא: $(x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_2 \vee x_3 \vee x_4 \vee \bar{x}_5) \wedge (\ ) \ldots$
- **פסוק 3-CNF**: פסוק CNF שבו כל פסוקית מכילה שלושה ליטרלים בדיוק.
- נאמר כי השמה v **מספקת פסוק CNF** אמ"מ היא נותנת ערך T לפחות לליטרל אחד בכל פסוקית שבו.

נגדיר את שתי השפות הבאות:

$SAT = \{\varphi \mid \text{ספיק } CNF \text{ הוא פסוק } \varphi\}$

$3SAT = \{\varphi \mid \text{ספיק } 3-CNF \text{ הוא פסוק } \varphi\}$

<u>טענה</u>: SAT שייכת ל NP :

נראה מכונת טיורינג א"ד פולינומית המקבלת אותה:

המכונה M_{SAT} פועלת על קלט φ באופן הבא:

1. אם φ איננו פסוק CNF נדחה.
2. המכונה תנחש השמה ותבדוק האם ההשמה מספקת את φ. אם כן המכונה תקבל ואחרת היא תדחה.

- **פולינומיות**: ניחוש המכונה פולינומי באורך הפסוק. הבדיקה האם ההשמה מספקת פולינומיאלית אף היא, ולכן המכונה פולינומית..
- **תקפות**: אם $\varphi \in SAT$ אז הפסוק ספיק ולכן קיימת השמה שמספקת אותו. במסלול בו מנחשים השמה זו M_{SAT} תקבל, ומכאן $\varphi \in L(M_{SAT})$.

אם $\varphi \notin SAT$ אז הפסוק אינו ספיק ולכן לא קיימת השמה שמספקת אותו. בכל המסלולים M_{SAT} תדחה ונקבל כי $\varphi \notin L(M_{SAT})$.

<u>טענה</u>: $3-SAT \leq_P SAT$

הוכחה: נגדיר פונקציה $f(\varphi)=\varphi'$ כך שיתקיים כי φ' הוא פסוק CNF ספיק אמ"מ φ הוא פסוק 3-CNF ספיק.

נגדיר את f באופן הבא:

- אם φ הוא $3-CNF$ אז $f(\varphi)=\varphi$.
- אחרת $f(\varphi)=(x_1 \wedge \overline{x}_1)$ עבור x_1 משתנה.

פולינומיות: הרדוקציה פולינומית - בדיקה האם φ הוא פסוק $3-CNF$ והעתקת הפסוק או כתיבת מחרוזת קבועה היא פולינומית באורך הקלט.

תקפות: אם $\varphi \in 3-SAT$ אז φ הוא פסוק $3-CNF$ ספיק ולכן φ' פסוק CNF ספיק ומכאן $\varphi' \in SAT$.

אם $\varphi \notin 3-SAT$ אז קיימות שתי אפשרויות:

1. φ הוא לא פסוק $3-CNF$. במקרה זה $\varphi' = x_1 \wedge \overline{x}_1$ ולכן לא ספיק ולכן $\varphi' \notin SAT$.
2. φ הוא פסוק $3-CNF$ לא ספיק. לכן $\varphi' = \varphi$ ולכן הוא לא ספיק ומתקיים $\varphi' \notin SAT$.

<u>טענה</u>: $SAT \leq_P 3SAT$.

<u>טענה</u>: אם P=NP אז כל שפה ב-NP היא NPC למעט ϕ,Σ^*.

סקיצת הוכחה:

תהי L שפה כנ"ל.

- $L\neq\Sigma^*$ ולכן קיימת $w_1\notin L$.
- $L\neq\phi$ ולכן קיימת $w_2\in L$.

נתון כי L ב-NP. נראה כי L ב-NPC על ידי רדוקציה מכל שפה L' ב-NP אל L.

תהי L' שפה ב-NP. נגדיר את הפונקציה הבאה:

$$f(x)=\begin{cases}w_2 & x\in L'\\ w_1 & w\notin L'\end{cases}$$

הרדוקציה מלאה, תקפה וניתנת לחישוב בזמן פולינומי.

(הרדוקציה ניתנת לחישוב בזמן פולינומי לפי ההנחה כי P=NP).

5. רשימת מקורות

- הרצאות "תורת החישוביות" בטכניון. http://webcourse.cs.technion.ac.il/236343
- תרגולים בקורס "תורת החישוביות" בטכניון.

- Sipser, Michael', Introduction to the Theory of Computation 2nd edition (2004). Course Technology