תורת הסיבוכיות — תרגול 1
ריפוד ומשפט לדנר

הקדמה

פונקציות שעון וזכרון

בחלק מההוכחות בקורס אנו מניחים כי עבור חסם זמן $t(n)$ ניתן לחשב את הערך $t(n)$ ביעילות בהינתן n; דבר דומה אנו מניחים עבור חסם זכרון $s(n)$. אף שזוהי הנחה טבעית שמתקיימת עבור כל הפונקציות שיעניינו אותנו, היא אינה נכונה לכל פונקציה. מכאן ההגדרה הבאה:

1. פונקציה $t : \mathbb{N} \to \mathbb{N}$ נקראת **פונקצית שעון** (Time Constructible Function) אם קיימת מכונת טיורינג דטרמיניסטית M בעלת <u>חסם זמן</u> $t(n)$ אשר על קלט 1^n מוציאה כפלט את $t(n)$.
2. פונקציה $s : \mathbb{N} \to \mathbb{N}$ נקראת **פונקצית זכרון** (Space Constructible Function) אם קיימת מכונת טיורינג דטרמיניסטית M בעלת <u>חסם זכרון</u> $s(n)$ אשר על קלט 1^n מוציאה כפלט את $s(n)$.

קיימות גרסאות שונות להגדרה זו. לעיתים יהיה נח להניח שחישוב $t(n)$ נעשה בסיבוכיות זמן $\mathcal{O}(t(n))$.

ריפוד

ריפוד (Padding) של שפה L משמעותו הוספת ג'יבריש למילות L כדי להאריך באופן מלאכותי את אורכן. השפה המרופדת המתקבלת L' אינה שונה רעיונית מ־ L פרט לכך שהמילים בה ארוכות יותר מהנדרש, וכתוצאה מכך אלגוריתמים שאולי אינם יעילים עבור L הם כן יעילים עבור L', שכן כמות המשאבים שהם צורכים עבור מילה ב־L' היא פונקציה של חלק קטן מהמילה (החלק שאינו שייך לריפוד). לדוגמה: תהא M מכונה המכריעה את השפה L בסיבוכיות זיכרון $\mathcal{O}(n^2)$. נגדיר שפה חדשה $L' = \left\{ x\$^{|x|^2-|x|} \,\middle|\, x \in L \right\}$. נגדיר מכונה M' המכריעה את L' על ידי כך שמפעילה את M על x ועונה כמוה. אם ל־ M סיבוכיות זמן $\mathcal{O}(n^2)$ הרי שסיבוכיות הזמן של M' היא $\mathcal{O}(\sqrt{n}^2) = \mathcal{O}(n)$.

במבט ראשון ריפוד נראה כמו פעולה חסרת טעם. אף על פי כן, בתרגול זה נראה דוגמאות לאופן שבו ניתן להשתמש בו כדי להוכיח טענות מעניינות.

שיטת הניפוח כלפי מעלה

נסמן:

$$\mathsf{DL} = \mathsf{DSPACE}(\log n)$$

$$\mathsf{NL} = \mathsf{NSPACE}(\log n)$$

<u>טענה</u>: אם $\mathsf{DL} = \mathsf{NL}$ אזי $\mathsf{DSPACE}(n) = \mathsf{NSPACE}(n)$.

<u>הוכחה</u>: נניח כי $\mathsf{DL} = \mathsf{NL}$.

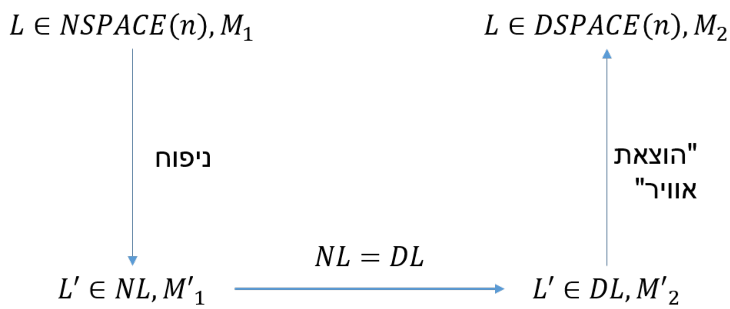
כיוון ראשון: $\mathsf{DSPACE}(n) \subseteq \mathsf{NSPACE}(n)$ היות שכל מכונה דטרמיניסטית ניתנת לתיאור כמכונה אי דטרמיניסטית.

כיוון שני: נראה ש־ $\mathsf{NSPACE}(n) \subseteq \mathsf{DSPACE}(n)$.

נפעל באופן הבא:

1. ניקח $L \in \mathsf{NSPACE}(n)$ כלשהי, ומכונת טיורינג אי דטרמיניסטית M_1 שמקבלת אותה בזכרון $\mathcal{O}(n)$.
2. נבצע "ניפוח": נבנה מ־ L את השפה המנופחת $L' = \left\{ x\$2^{|x|} \,\middle|\, x \in L \right\}$.
3. נשתמש ב־ M_1 כדי להראות קיום של מכונה M_1' עבור L' שמראה כי $L' \in \mathsf{NL}$.
4. מההנחה ש־$\mathsf{NL} \subseteq \mathsf{DL}$ נקבל כי $L' \in \mathsf{DL}$, ולכן קיימת עבורה מכונה דטרמיניסטית M_2' עם חסם זכרון $\mathcal{O}(\log n)$.
5. "נוציא אוויר": נשתמש ב־ M_2' כדי להראות קיום של מכונה M_2 עבור L שמראה כי $L \in \mathsf{DSPACE}(n)$.

סכימת ההוכחה



תהא $L \in \mathsf{NSPACE}(n)$ ותהא M_1 מכונה אי דטרמיניסטית שמכריעה את L בסיבוכיות זיכרון $\mathcal{O}(n)$. נבנה $L' = \left\{ x\$2^{|x|} \,\middle|\, x \in L \right\}$. כעת, נגדיר מכונה M_1' שעל קלט y מאורך n פועלת כך:

1. מוודאת שהקלט הוא מהצורה $y = x\$^t$ וש־ $t = 2^{|x|}$. ($\mathcal{O}(\log n)$ זיכרון.)
2. מריצה את M_1 על x ועונה כמוה. ($\mathcal{O}(\log n)$ זיכרון.)

נסיק כי $L' \in \mathsf{NL}$, ולכן $L' \in \mathsf{DL}$. כלומר קיימת מכונה M_2' דטרמיניסטית בעלת סיבוכיות זיכרון $\mathcal{O}(\log n)$.

נגדיר M_2 על קלט x באופן הבא:

1. כתבי $x\$2^{|x|}$.
2. הריצי את M_2'ועני כמוה.

נשים לב שכדי לכתוב $2^{|x|}$ דולרים, יש צורך ב־$\mathcal{O}(2^n)$ תאי זיכרון, דבר שגורם לחריגה מסיבוכיות הזיכרון הלינארית הרצויה.

הפתרון: הזכרון ייכתב רק בצורה וירטואלית. כלומר, M_2 תסמלץ את ריצת M_2' על x ותעקוב אחרי מיקום הראש הקורא של סרט הקלט של M_2'. אם הראש עבר את $|x|$ התווים הראשונים, M_2 תגרום ל־M_2' לחשוב שהיא רואה $; ואחרי $|x| + 2^{|x|}$ התווים הראשונים, היא תגרום לה לחשוב שהיא רואה את קצה הקלט.

המונה דורש רק $\mathcal{O}\left(\log\left(|x| + 2^{|x|}\right)\right) = \mathcal{O}(|x|)$ זיכרון, כנדרש.

הראינו כי $\mathsf{NSPACE}(n) \subseteq \mathsf{DSPACE}(n)$.

שיטת הניפוח כלפי מעלה: ניסוח עיקרון כללי

בהתבסס על הדוגמה נוכל כעת לתת ניסוח לעיקרון כללי:

יהיו $\mathsf{CLASS}_1, \mathsf{CLASS}_2 \in \{\mathsf{DSPACE}, \mathsf{NSPACE}\}$ ותהיינה $f_1(n), f_2(n), g(n)$ פונקציות זכרון כך ש־ $g(n) \geq n$, $f_2(n) \geq n$ (לחלופין, $f_2(n) \geq \log n$ ו־ $f_2(g(n))$ פונקצית זכרון). אזי, אם מתקיים $\mathsf{CLASS}_1(f_1(n)) \subseteq \mathsf{CLASS}_2(f_2(n))$, מתקיים גם $\mathsf{CLASS}_1(f_1(g(n))) \subseteq \mathsf{CLASS}_2(f_2(g(n)))$.

משפט דומה קיים גם עבור סיבוכיות זמן אך עם נתונים שונים. (נותיר את הניסוח ואת ההוכחה של שני המשפטים כתרגיל.)

משפט לדנר

בקורס בתורת החישוביות ראינו מספר שפות השייכות ל־ P (שפות סופיות, שפות רגולריות, 2־צביעה של גרף, מסלול אוילרי בגרף, וכדומה) ושפות רבות שהן NP־שלמות (ספיקות, מעגל המילטוני, 3־צביעה של גרף...), ולא עסקנו בקורס בשפות שאינן ב־ P וגם אינן NP־שלמות. סיבה אחת לכך היא שאיננו יודעים אם $\mathsf{P} \neq \mathsf{NP}$, ואם $\mathsf{P} = \mathsf{NP}$ הרי שכל שפה לא טריוויאלית ב־ NP היא בו זמנית גם ב־ P וגם NP־שלמה. עם זאת, נשאלת השאלה מה קורה אם $\mathsf{P} \neq \mathsf{NP}$; האם עדיין כל שפה היא או ב־ P או NP־שלמה? התשובה שלילית: קיימות שפות NP־**ביניים** שאינן ב־ P ואינן NP־שלמות. משפט לדנר מוכיח את קיומה של שפה כזו. הבניה תהיה מלאכותית משהו והשפה שנקבל תהיה SAT עם ריפוד מתאים. כיום לא ידועות דוגמאות לשפות לא מלאכותיות שהן שפות NP־ביניים. בין המועמדות הבולטות ביותר לתפקיד ניתן למצוא את השפה שמתאימה לבעיית הפירוק לגורמים ואת השפה שמתאימה לבעיית האיזומורפיזם של גרפים.

בהערת אגב נציין שתוצאה דומה קיימת גם עבור המחלקות R ו־ RE, וגם במקרה זה בניית שפות הביניים איננה טריוויאלית (אך גם לא קשה במיוחד).

השפה

ראשית נתאר ריפוד באופן כללי. אם $h \colon \mathbb{N} \to \mathbb{N}$ היא פונקציה, אז נגדיר שפה $\mathrm{SAT}_h \triangleq \left\{\varphi\$^{h(|\varphi|)} \,\middle|\, \varphi \in \mathrm{SAT}\right\}$. כלומר, זוהי פשוט השפה SAT כאשר כל φ בשפה מרופד עם $h\left(|\varphi|\right)$ תווי "זבל" (אנו מניחים שהתו \$ אינו חלק מאלפבית השפה SAT). במילים אחרות, כל פסוק מאורך n הוגדל באופן מלאכותי לגודל $n + h\left(n\right)$.

בהמשך נגדיר פונקציה $H\left(n\right)$ הניתנת לחישוב בזמן פולינומי, ובאמצעותה נגדיר פונקציה $h\left(n\right) = n^{H(n)}$ (שהיא פונקציית שעון בהינתן זחישבנו את החזקה), ואז שפת הביניים שלנו תהיה SAT_h.

הבנייה של $H\left(n\right)$ תבטיח שתתקיים התכונה הבאה: $\mathrm{SAT}_h \in \mathsf{P}$ אם ורק אם H חסומה (שימו לב להבדל שבין h ו־H!).

ראשית נשתמש ב־ SAT_h כדי להוכיח את הטענה של משפט לדנר, ולאחר מכן נסביר כיצד לבנות את H.

השימוש בתכונה

הבה ונבין מדוע התכונה המתוארת לעיל עוזרת לנו לפסול הן את $\mathrm{SAT}_h \in \mathsf{P}$ והן את $\mathrm{SAT}_h \in \mathsf{NPC}$.

ראשית, אם $\mathrm{SAT}_h \in \mathsf{P}$ אז כאמור H חסומה, כלומר $H\left(n\right) < c$ עבור קבוע c כלשהו ולכן $h\left(n\right) \le n^c$ — פולינומית. לכן קיימת רדוקציה $\mathrm{SAT} \le_p \mathrm{SAT}_h$ הנתונה על ידי $\varphi \mapsto \varphi\$^{h(|\varphi|)}$ (כאן הסתמכנו על כך ש־h היא פונקציית שעון) ומכאן ש־ $\mathrm{SAT} \in \mathsf{P}$, בסתירה להנחה ש־ $\mathsf{P} \neq \mathsf{NP}$.

אם $\mathrm{SAT}_h \in \mathsf{NPC}$ אז בפרט קיימת רדוקציה $\mathrm{SAT} \le_p \mathrm{SAT}_h$. נניח ש־ $c \cdot n^d$ הוא חסם זמן עבור הרדוקציה. הרעיון כעת יהיה להשתמש ברדוקציה הזו כדי ליצור רדוקציה מ־SAT <u>לעצמה</u> שמקטינה את אורך הפסוקים שעליהם היא מופעלת, ובכך לפתור את SAT בזמן פולינומי.

ראינו כבר כי $\mathrm{SAT}_h \notin \mathsf{P}$, ולכן מהתכונה של $H\left(n\right)$ נקבל כי היא אינה חסומה, ובפרט נקבל שקיים n_0 כך שלכל $n > n_0$ מתקיים $h\left(n\right) > c \cdot n^{2d}$.

עבור נוסחה $\varphi \in \mathrm{SAT}$ המקיימת $|\varphi| = n > n_0$ הנוסחה תתמפה למחרוזת מהצורה $\psi\$^{h(|\psi|)}$ כאשר $\psi \in \mathrm{SAT}$ ו־$\left|\psi\$^{h(|\psi|)}\right| \le c \cdot n^d$.

נסמן $k = |\psi|$, אז קיבלנו כי $k + h\left(k\right) \le c \cdot n^d$.

מצד שני, $h\left(k\right) > c \cdot k^{2d}$ ולכן $c \cdot k^{2d} \le c \cdot n^d$, כלומר $k \le \sqrt{n}$.

כלומר, הרדוקציה העבירה את φ לפסוק ψ קטן בהרבה ממנה, ועוד הרבה ג'יבריש שאפשר להתעלם ממנו.

לסיום, הראינו שאם יש רדוקציה $\mathrm{SAT} \le_p \mathrm{SAT}_h$ אז ניתן לפתור את SAT באופן הבא: עבור קלט φ חשב את הרדוקציה עליו לקבלת פלט $\psi\$^{h(|\psi|)}$. בדוק שהפלט הוא אכן מהצורה המבוקשת ואחרת דחה מייד (כלומר, ψ ואז המספר המתאים של דולרים; לשם כך יש לחשב את h ולכן גם כאן משתמשים בהנחה ש־ h היא פונקציית שעון). אם $|\psi|$ קטן מ־ n_0 הכרע באופן ישיר אם $\psi \in \mathrm{SAT}$ או לא וענה בהתאם; אחרת פעל בצורה רקורסיבית על ψ. מכיוון שבכל איטרציה גודל הפסוק הנבדק קטן מגודלו הקודם, יידרשו לכל היותר (ולמעשה, הרבה פחות) מספר פולינומי ב־ $|\varphi|$ של איטרציות (שכל אחת פולינומית ב־ $|\varphi|$).

הגדרת h

כעת נגדיר את h כך שתקיים את התכונה שתיארנו לעיל: $\mathrm{SAT}_h \in \mathsf{P}$ אם ורק אם H חסומה.

ראשית, נמספר את כל מכונות הטיורינג: M_k תהיה המכונה ה־k־ית בסדר כלשהו על כל המכונות. כעת נגדיר את $H(n)=k$ להיות ה־ k הקטן ביותר שמקיים $k<\log\log n$ וגם ש־ M_k מכריעה את SAT_h עבור x־ים המקיימים $|x|\le\log n$, ובזמן ריצה $k\cdot|x|^k$ לכל היותר. אם אין מכונה כזו, אז $k=\log\log n$.

ההגדרה נראית מעגלית, אך מכיוון שכדי להגדיר את $H(n)$ עלינו להכיר רק ערכים של H הקטנים או שווים ל־ $\log n$, אין כאן מעגליות (זוהי הגדרה <u>רקורסיבית</u>, לא מעגלית).

חישוב $H(n)$ הוא עניין ישיר למדי: פשוט עוברים בהרצה מבוקרת על כל המכונות $M_1,\ldots,M_{\log\log n}$ ולכל מכונה, מריצים אותה על כל הקלטים x המקיימים $|x|\le\log n$. בנוסף צריך לבדוק באופן ישיר האם $x\in\mathrm{SAT}_h$, זאת עושים באמצעות פתרון בכוח גס של SAT וחישוב רקורסיבי של $H(n)$ לגדלים קטנים יותר. הבחירה של $\log\log n$ כחסם על המכונות הונדסה בדיוק כדי להבטיח זמן ריצה סביר לחישוב $H(n)$ (בדקו זאת!).

נותר כעת להבין מדוע $H(n)$ אכן מקיימת את תכונת החסימות. ראשית נניח כי $\mathrm{SAT}_h\in\mathsf{P}$ ונוכיח כי H חסומה. אם $\mathrm{SAT}_h\in\mathsf{P}$ אז יש אינסוף מכונות פולינומיות המכריעות את SAT_h בזמן cn^c עבור $c>0$ קבוע גדול דיו; תהא M_k מכונה שכזו כך ש־ $k>c$. אז בפרט M_k מכריעה את SAT_h עבור קלטים המקיימים $|x|\le\log n$ בזמן ריצה $k\cdot|x|^k$ לכל n; מכאן ש־ $H(n)\le k$ לכל n שעבורו $k\le\log\log n$, כלומר לכל $n>2^{2^k}$, כלומר H חסומה.

בכיוון השני, אם H חסומה נוכיח ש־ $\mathrm{SAT}_h\in\mathsf{P}$ באופן הבא: מכיוון ש־ $H(n)$ חסומה, מעקרון שובך היונים יש ערך k שאותו היא מקבלת אינסוף פעמים, והדבר גורר ש־ M_k מכריעה את SAT_h בזמן $k\cdot n^k$. כדי לראות זאת, נניח שעל קלט x כלשהו, M_k לא עוצרת עם התשובה הנכונה בזמן $k\cdot|x|^k$, אז על פי הגדרת H בהכרח $H(n)\neq k$ לכל $n>2^{|x|}$, סתירה.