תורת הסיבוכיות – תרגול 3
משפטי היררכיה ו־NL־שלמות

משפטי היררכייה

משפטי ההיררכייה הם ניסוח פורמלי לתחושה האינטואיטיבית של "יותר משאבים $\Longleftarrow$ יותר כוח חישוב". הנה ניסוחם המדויק עבור סיבוכיות זמן/זכרון דטרמיניסטי:

- אם $f(n), g(n)$ פונקציות כך ש־ g פונקציית זיכרון, $f(n) \geq \log n$ ו־ $f(n) = o(g(n))$, אז $\mathsf{DSPACE}(f(n)) \subsetneqq \mathsf{DSPACE}(g(n))$.
- אם $f(n), g(n)$ פונקציות כך ש־ $g \geq n$ פונקציית שעון ו־ $f(n)\log f(n) = o(g(n))$, אז $\mathsf{DTIME}(f(n)) \subsetneqq \mathsf{DTIME}(g(n))$.

הוכחת שני המשפטים זהה: בונים מכונה U אשר מפרשת את הקלט שלה בתור מכונה M ומריצה אותה "מספיק צעדים" (גם מספר זה תלוי בקלט) ועונה הפוך ממנה אם M עצרה. ההבדל שבין תנאי שני המשפטים נובע מהתקורה של ביצוע סימולציה – סימולציה של מכונה דורשת תקורה קבועה של זכרון, אך תקורה לוגריתמית של זמן ריצה.

הקושי המהותי בזמן ריצה מגיעה מכך שעלינו לסמלץ מכונה רב־סרטית באמצעות המכונה U, שמספר סרטיה קבוע. הכרחי ש־ U תתקוף גם מכונות רב־סרטיות, שכן מכונה חד־סרטית שקולה מתקבלת על ידי הגדלה ריבועית של זמן הריצה, מה שמספיק טוב לצורך דיון במחלקה כמו P אך לא מספיק טוב לצורך התנאים של משפטי ההיררכייה. באופן נאיבי ניתן לחשוב שסימולציה של מכונה רב־סרטית אכן תדרוש הגדלה ריבועית של זמן הריצה, אך ניתן להתחכם (לא נפרט כאן איך) ולהסתפק בתקורה לוגריתמית כאמור.

קיימים משפטי היררכייה גם עבור מחלקות זמן/זכרון אי דטרמיניסטיות.

שימוש חכם במשפט היררכיית הזמן ובעיקרון הניפוח

נניח כי אנו רוצים להוכיח כי $\mathsf{DTIME}(2^n) \subsetneq \mathsf{DTIME}(n2^n)$.

לרוע המזל לא ניתן להשתמש ישירות במשפט ההיררכייה שכן $2^n \cdot \log(2^n) = n2^n$. עם זאת, שימוש חכם בעיקרון הניפוח כלפי מעלה יפתור את הבעיה.

נניח בשלילה כי $\mathsf{DTIME}(2^n) \subseteq \mathsf{DTIME}(n2^n)$. "ננפח" על ידי החלפת n ב־$n + \log n$ (קל לבדוק כי תנאי המשפט הכלל עובדים כאן, או ממש לשחזר את ההוכחה). נקבל כי

$$\mathsf{DTIME}\left((n+\log n)\, 2^{n+\log n}\right) \subseteq \mathsf{DTIME}\left(2^{n+\log n}\right)$$

כלומר:

$$\mathsf{DTIME}\left(n^2 2^n\right) \subseteq \mathsf{DTIME}\left(n2^n\right)$$

ומהנחת השלילה שלנו עולה ש־

$$\mathsf{DTIME}\left(n^2 2^n\right) \subseteq \mathsf{DTIME}\left(2^n\right)$$

אבל $2^n \cdot \log(2^n) = n2^n = o\left(n^2 2^n\right)$, וקיבלנו סתירה למשפט היררכיית הזמן. לכן, $\mathsf{DTIME}(2^n) \subsetneq \mathsf{DTIME}(n2^n)$.

הרכבת פונקציות ושמירה על סיבוכיות זכרון

כאשר עוסקים במכונות טיורינג לחישוב פונקציות ורוצים להגדיר עבורן סיבוכיות זכרון, הכרחי להגביל גם את אמצעי הפלט. אנו חושבים על סרט הפלט כסרט לכתיבה בלבד שניתן להוסיף לו ביטים אך לא ניתן להזיז את הראש לאורכו או לקרוא ממנו דבר מה. כך ניתן להמשיך למדוד את סיבוכיות הזיכרון של מכונה באמצעות השימוש בסרט העבודה בלבד.

בהרצאה ראינו כי $\mathsf{FSPACE}\left(\log n\right)$ (מחלקת כל הפונקציות שניתנות לחישוב בזיכרון לוגריתמי) סגורה להרכבה. הבעיה שיכולה לצוץ היא שהפלט של המכונה הראשונה יהיה גדול מכדי שניתן יהיה לרשום אותו על סרט העבודה של המכונה שמחשבת את ההרכבה. הפתרון היה שימוש במכונה הראשונה כ"קופסה שחורה" – בכל פעם שבה המכונה השניה צריכה לקרוא ביט מהפלט של המכונה הראשונה, המכונה הראשונה מופעלת מחדש ומורצת עד שערכו של אותו ביט נקבע. כך, במחיר התעללות בסיבוכיות זמן הריצה, נשמרת סיבוכיות הזכרון.

האם אותו תעלול עובד גם עבור $\mathsf{FSPACE}\left(n\right)$? התשובה שלילית. הבעיה היא שעכשיו הפלט הסופי עשוי להיות גדול מדי ולא יהיה ניתן לייצר אותו ללא חריגה ממגבלות הזכרון. כמובן שאין זו הוכחה, ועל כן נציג כעת פונקציה קונקרטית שאינה ניתנת לחישוב בזכרון לינארי אך ניתן להציגה כהרכבה של שתי פונקציות שכן ניתנות לחישוב שכזה.

נגדיר $f\left(x\right)=1^{2^{|x|}}$. במילים אחרות, לכל קלט x נכתבים $2^{|x|}$ סימני 1 על הסרט. ברור כי $f\in\mathsf{FSPACE}\left(n\right)$, שכן כל שיש לעשותו בסרט העבודה הוא להחזיק מונה שערכו המקסימלי הוא $2^{|x|}$, מה שדורש $\log\left(2^{|x|}\right)=|x|$ ביטים.

נגדיר $h\left(x\right)=f\left(f\left(x\right)\right)$. כלומר, $h\left(x\right)=1^{2^{2^{|x|}}}$. הפונקציה הזו מוציאה פלטים גדולים מדי מכדי שניתן יהיה לחשבם במקום לינארי. נניח בשלילה שקיימת מכונה M שמחשבת את הפונקציה ופועלת בזיכרון $O\left(n\right)$. אזי, מספר הקונפיגורציות המקסימלי שלה הוא $2^{O(n)}$, מה שאומר שאם היא מבצעת יותר מ־ $2^{O(n)}$ צעדים היא נכנסת ללולאה אינסופית ולא תעצור לעולם. אבל בכדי לכתוב $1^{2^{2^{|x|}}}$ תווים על סרט הפלט, עליה לבצע מספר צעדים גדול יותר מ־ $2^{O(n)}$ (עבור x גדול דיו) – סתירה.

השפה $2\mathrm{SAT}$ והקשר שלה ל־NL

כזכור, הגדרנו:

- $\mathsf{DL}=\mathsf{DSPACE}\left(\log n\right)$
- $\mathsf{NL}=\mathsf{NSPACE}\left(\log n\right)$

מכיוון שמחלקות אלו הן מקבילות ל־ P ו־ NP, היינו רוצים לדבר על מושג של NL־שלמות בדומה ל־ NP־שלמות. אלא שלצורך כך יש לשנות את סוג הרדוקציה. כזכור, אנו בונים את הרדוקציה בהתאם לכוחה של המחלקה החלשה יותר, DL במקרה זה. על כן, אנו דורשים שפונקציית הרדוקציה תהיה בעלת סיבוכיות זכרון לוגריתמית (הפלט שלה עדיין עשוי להיות פולינומי בגודלו; זכרו שאנו משתמשים בסרט פלט לכתיבה בלבד).

נעבור כעת לדון בשפה $2\mathrm{SAT}$ של פסוקי $2\mathrm{CNF}$ ספיקים. כזכור, $3\mathrm{SAT}$ היא NP־שלמה. לעומת זאת, נראה כעת כי $2\mathrm{SAT}$ שייכת ל־ NL, ועל כן בפרט היא ב־ P. נרצה להראות ש־ $2\mathrm{SAT}$ היא NL־שלמה ביחס לרדוקציות בעלות זיכרון לוגריתמי.

הרעיון הבסיסי הוא שניתן לחשוב על פסוקית מהצורה $\left(\overline{x}\vee y\right)$ כפסוקית מהצורה $\left(x\to y\right)$, וגם כפסוקית מהצורה $\left(\overline{y}\to\overline{x}\right)$. מכאן שניתן לבנות עבור פסוק $2\mathrm{CNF}$ φ את "גרף הגרירות" שלו שצמתיו הם הליטרלים של הפסוק (כלומר, x ו־ $\overline{x}$ לכל משתנה x שמופיע בפסוק), והפסוקית $\left(\alpha\vee\beta\right)$ מייצרת את הקשתות $\overline{\alpha}\to\beta$ ו־ $\overline{\beta}\to\alpha$. הרעיון הוא שאם $\alpha\to\beta$, אז מתן ערך True ל־ α מאלץ אותנו לתת ערך True גם ל־ β.

שימו לב לסימטריה של הגרף: אם יש קשת $\alpha\to\beta$, אז יש גם קשת $\overline{\beta}\to\overline{\alpha}$. מכאן שאם יש מסלול $\alpha\leadsto\beta$, אז יש גם מסלול $\overline{\beta}\leadsto\overline{\alpha}$.

כעת ניתן לנסח קריטריון לספיקות φ: φ <u>אינו ספיק</u> אם ורק אם קיים משתנה x כך שיש מסלול מ־ x אל $\overline{x}$ ומסלול מ־ $\overline{x}$ אל x (כלומר, x ו־ $\overline{x}$ נמצאים על מעגל מכוון). ברור כי קיום זוג מסלולים שכזה מבטיח שהפסוק אינו ספיק.

בכיוון השני, נרצה להראות שהיעדר זוג מסלולים כנ"ל מבטיח את היות הפסוק ספיק. המטרה היא לבצע השמה של ערכי אמת לצמתי הגרף כך שאם $u\to v$ אז לא ייתכן ש־ v קיבל True ואילו u קיבל False. נניח שאכן לא קיים זוג מסלולים כנ"ל, ונראה אלגוריתם שמוצא השמה מספקת. האלגוריתם מתחיל מכך שאף משתנה לא מקבל ערך, ובהדרגה נותן ערכים למשתנים עד שמקבלים פסוק ספיק. האלגוריתם הולך כך (זכרו שהמשתנים הם $x_{1},x_{2},\ldots$ והליטרים הם המשתנים ושלילתם):

- כל עוד קיים משתנה שלא קיבל ערך:
  - בחר את אחד מהמשתנים הללו, x_{i}.
  - אם לא קיים בגרף מסלול מ־ x_{i} ל־ $\overline{x_{i}}$:
    * תן ל־ x_{i} את הערך True.
    * לכל ליטרל α כך שקיים מסלול מ־ x_{i} ל־ α בגרף:
      · אם המשתנה של α כבר קיבל ערך, אז המשך בלולאה לליטרל הבא (פקודת continue).
      · אם $\alpha=x_{j}$ עבור משתנה x_{j} כלשהו, אז תן ל־ x_{j} את הערך True; ואם $\alpha=\overline{x_{j}}$, אז תן ל־ x_{j} את הערך False.
  - אם קיים מסלול מ־ x_{i} ל־ $\overline{x_{i}}$, אז לא קיים מסלול מ־ $\overline{x_{i}}$ ל־ x_{i}; תן ל־ x_{i} את הערך False, ובצע תהליך הדומה למה שמופיע עבור המקרה הראשון.

נראה שההשמה שמוצא האלגוריתם מספקת את הפסוק. נניח בשלילה שלא, ולכן קיימת פסוקית שלא מסתפקת. נניח בלי הגבלת הכלליות שהפסוקית הזאת מכילה שני ליטרלים שמופיעים בחיוב, כלומר, הפסוקית שאינה מסתפקת היא מהצורה $x_l \vee x_k$. נניח ש־ x_l מקבל ערך לפני x_k, וש־ x_l מקבל ערך באיטרציה שתחילתה הצבנו ערך True ב־ x_i כאשר לא קיים מסלול מ־x_i ל־ $\overline{x_i}$, עבור x_i כלשהו. מפני שהפסוקית $x_l \vee x_k$ לא מסתפקת, בפרט הצבנו במשתנה x_l את הערך False. זה יכול לקרות רק כאשר קיים בגרף מסלול מ־ x_i ל־ $\overline{x_l}$, ובגלל שהגרף מכיל את הקשת $\overline{x_l} \to x_k$, נובע כי קיים מסלול מ־ x_i ל־ x_k. נותר להראות כי אין בגרף מסלול מ־ x_i ל־ $\overline{x_k}$, כי אם נראה זאת אז ינבע כי באיטרציה הזאת x_k לא יקבל את הערך False, וינבע מכך שהוא יקבל את הערך True עד סוף האיטרציה, והפסוקית $x_l \vee x_k$ תסתפק בסתירה להנחת השלילה. אם כן, נניח בשלילה שקיים מסלול מ־ x_i ל־ $\overline{x_k}$. נובע כי קיים מסלול מ־ x_k ל־ $\overline{x_i}$, ומכך נובע כי קיים מסלול מ־ x_i ל־ $\overline{x_i}$ דרך x_k, בסתירה לכך שלא קיים מסלול כנ"ל.

הקריטריון לעיל מניב מייד אלגוריתם שמראה כי $2\mathrm{SAT}$ היא ב־coNL: מנחשים משתנה x, ומנחשים מסלולים $x \rightsquigarrow \overline{x}$ ו־$\overline{x} \rightsquigarrow x$ בסיבוכיות זכרון לוגריתמית (בדומה ל־ CON). כעת נסיק ממשפט אימרמן כי $2\mathrm{SAT} \in \mathsf{NL}$.

ניתן להשתמש ברעיון דומה כדי להוכיח כי $2\mathrm{SAT}$ היא NL־שלמה על ידי רדוקציה מ־ $\overline{\mathrm{CON}}$ (שפת כל השלשות (G,s,t) כך ש־ G גרף מכוון שאין בו מסלול מ־ s ל־ t) שאף היא NL־שלמה. הרעיון הוא לבנות פסוק שבו יש משתנה לכל צומת של הגרף, ולכל קשת $u \to v$ קיימת הפסוקית $(\bar{v} \vee u)$, וכמו כן קיימות שתי הפסוקיות $(\bar{t})$, (s).

אם קיים מסלול ב־ G מ־ s ל־ t, אז השמה של ערך True ל־ s "תפעפע" דרך המסלול ותגרור שגם t יקבל את הערך True, וכך הפסוקית $(\bar{t})$ אינה ספיקה.

בכיוון השני, נניח שאין מסלול ב־ G מ־ s ל־ t. נביט בהשמה שנותנת ערך True לכל המשתנים שמייצגים צמתים הישיגים מ־ s בגרף, וערך False לכל הצמתים הנותרים. נראה שהשמה זאת מספקת את הפסוק כנדרש. ראשית, היא מספקת את הפסוקיות s ו־ $\bar{t}$. שנית, לכל פסוקית מהצורה $\bar{v} \vee u$, הדרך היחידה שלה לא להסתפק היא כאשר v מקבל ערך True ו־ u מקבל ערך False. זה יכול לקרות רק כאשר v ישיג מ־ s בגרף ו־ u לא ישיג ממנו, וזה לא ייתכן מפני שבגרף קיימת הקשת $u \to v$.