סיבוכיות מקום

קודם עסקנו בסיבוכיות זמן. היום נעסוק בסיבוכיות מקום, כלומר **המקום** המינימאלי הנדרש לחישוב.

סיבוכיות מקום עוסקת במקום הנדרש לביצוע החישוב בלי לכלול את אורך הקלט ואורך הפלט.

המודל שנעסוק בו הוא המודל הבא:

מודל של מ"ט בעלת שלושה סרטים: קלט, פלט, עבודה.

סרט קלט: סרט לקריאה בלבד

סרט פלט: סרט לכתיבה בלבד

סרט עבודה: סרט לכתיבה וקריאה. שטח הזיכרון בסרט זה הוא השטח שנמדוד כשנעסוק ב**סיבוכיות מקום**.

א"ב בינארי(למרות שניתן להשתמש בכל א"ב, כל עוד הוא לא אונארי)

הגדרה - $\mathrm{DSPACE}\left(s\left(n\right)\right)$

נאמר שבעיה מסויימת שייכת למחלקה $\mathrm{DSPACE}\left(s\left(n\right)\right)$, עבור פונקציה $s\left(\cdot\right)$ כלשהי, אם קיימת מכונת טיורינג כנ"ל שבהנתן קלט באורך n מוציאה פלט מתאים ומשתמשת בלא יותר מאשר $O\left(s\left(n\right)\right)$ תאים מסרט כדי להכריע את הבעיה.

אבחנה

עבור פונקציה $t\left(n\right)$ כלשהי,

$$\mathrm{DTIME}\left(t\left(n\right)\right)\subseteq\mathrm{DSPACE}\left(t\left(n\right)\right)$$

הסבר

ב$t\left(n\right)$ צעדים ניתן לגעת/לגשת לכל היותר $t\left(n\right)$ תאי זיכרון.

משפט

$$\mathrm{DSPACE}\left(s\left(n\right)\right)\subseteq\mathrm{DTIME}\left(n\cdot2^{O\left(s\left(n\right)\right)}\right)$$

בפרט עבור $s\left(n\right)\geq\Omega\left(\log\left(n\right)\right)$ נקבל

$$\mathrm{DSPACE}\left(s\left(n\right)\right)\subseteq\mathrm{DTIME}\left(2^{O\left(s\left(n\right)\right)}\right)$$

הוכחה

בהינתן מ"ט M וקלט x, תיאור מלא של המצב החישובי של המכונה אחרי t צעדים ניתן לתאר ע"י הדברים הבאים:

- תאור תוכן סרט העבודה
- מיקום הראש על סרט העבודה
- מיקום הראש על סרט הקלט
- המצב הפנימי של המכונה

(תוכן סרט הקלט נתון וקבוע ולכן הוא לא חלק מהמצב החישובי, וסרט הפלט הוא לכתיבה בלבד ולא יכול להשפיע על הפעולה הבאה ולכן גם הוא לא חלק מהמצב החישובי)

נשים לב כי אם קונפיגורציות חישוביות מסויימות חוזרות יותר מאשר פעם אחת בזמן ריצת המכונה אזי המכונה נכנסת ללולאה אינסופית, ולכן כמות הקונפיגורציות האפשרית היא חסם עליון על זמן ריצת המכונה. לכן זמן הריצה חסום ע"י

$$\underbrace{2^{s(n)}}_{\substack{\text{Possible}\\\text{contents}\\\text{for}\\\text{work-tape}}} \cdot \underbrace{s(n)}_{\substack{\text{Place of}\\\text{head in}\\\text{work-tape}}} \cdot \underbrace{n}_{\substack{\text{Place of}\\\text{head in}\\\text{input-tape}}} \cdot \underbrace{c}_{\substack{\text{Internal}\\\text{state of}\\\text{machine}}} = n \cdot 2^{O(s(n))}$$

המחלקה L

$$L = \bigcup_{\substack{c\\ l_c = c\log n}} \mathrm{DSPACE}\left(l_c\right)$$

מסקנה(משפט קודם)

$$L \subseteq P$$

השערה

$$L \neq P$$

סיבוכיות מקום של חישוב לא דטרמיניסטי

ניזכר שכאשר עסקנו בסיבוכיות זמן לא דטרמיניסטית, עסקנו בשני מודלים שקולים של חישוב:

1. מודל ה*on-line* - למכונה יש מעברים שלא מוגדרים באופן יחיד, והבחירה אינה מעבר לבצע, נעשית on-line ע"י המכונה(הקלט מתקבל $\Longleftrightarrow$ יש מסלול מקבל)
2. מודל ה*off-line* - במקרה זה המכונה היא דטרמיניסטית, אך בנוסף לקלט היא מקבלת עד(שזהו למעשה קלט נוסף). (הקלט מתקבל $\Longleftrightarrow$ יש עד שגורם למכונה להתקבל)

בהקשר של סיבוכיות מקום שני המודלים **אינם שקולים!** למודל הoff-line יש יותר כוח(יוכח בתרגיל), ואנו נעסוק לפיכך במודל הon-line בהקשר של סיבוכיות מקום לא דטרמיניסטית.

הגדרה - $\text{NSPACE}\left(s\left(n\right)\right)$

בעיה מסויימת שייכת למחלקה $\text{NSPACE}\left(s\left(n\right)\right)$ אם קיימת מ"ט ל"ד(במודל הon-line) המכריעה את הבעיה ומשתמשת בלכל היותר $O\left(s\left(n\right)\right)$ תאים מסרט העבודה.

אבחנות

$$\text{DSPACE}\left(s\left(n\right)\right)\subseteq\text{NSPACE}\left(s\left(n\right)\right)$$

$$\text{DTIME}\left(t\left(n\right)\right)\subseteq\text{NSPACE}\left(t\left(n\right)\right)$$

$$\text{NSPACE}\left(s\left(n\right)\right)\subseteq\text{DSPACE}\left(?\right)$$

המחלקה הלא-דטרמיניסטית המקבילה לL היא NL

$$NL=\bigcup_{\substack{c\\ l_c=c\log n}}\text{NSPACE}\left(l_c\right)$$

טענה

$$L\subseteq NL\subseteq P$$

הוכחה

עבור $s(n) \geq \log n$:

$$\text{DSPACE}(s(n)) \subseteq \text{DTIME}\left(2^{O(s(n))}\right)$$

ראינו קודם ש

$$\text{NSPACE}(s(n)) \subseteq \text{DTIME}\left(2^{O(s(n))}\right)$$

נובע משיקול זהה לקודם, שמספר הקונפיגורציות הכולל הוא חסם עליון על זמן ריצת המכונה גם במקרה הלא דטרמיניסטי.

נחזור לשאלה:

$$\text{NSPACE}(s(n)) \subseteq \text{DSPACE}(?)$$

ניקח לדוגמה $s(n) = \log n$:

$$\text{NSPACE}(\log n) \subseteq \text{DSPACE}(?)$$

הגדרה - רדוקציית log-space

רדוקציית log-space מL_1 לL_2 היא פונקציה $f(x)$ חשיבה בסיבוכיות זיכרון $\log(n)$ כך שלכל x מתקיים

$$x \in L_1 \iff f(x) \in L_2$$

רדוקציית log-space היא מקרה פרטי של רדוקציית Karp, שכן $f(x)$ החשיבה בסיבוכיות מקום $\log(|x|)$ חשיבה בסיבוכיות זמן פולינומית ב$|x|$.

הגדרה

A היא שלמה בNL אם $A \in NL$, וכן לכל $A' \in NL$ קיימת רדוקציית log-space מA' לA.

הגדרת בעייה שהיא שלמה בNL

$$st - Conn = \left\{ (G,s,t) \middle| \begin{array}{c} G \text{ is directed graph} \\ \text{where there is a} \\ \text{route from } s \text{ to } t \end{array} \right\}$$

טענה

$$st - Conn \in NL$$

הוכחה

להלן תיאור של אלגוריתם ל$st - Conn$ הרץ בסיבוכיות מקום לא דטרמיניסטית חסומה $\log n$:

- אתחל מונה ב1
- בצע כל עוד המונה$\leq n$
  - בחר שכן אקראי של קודקוד נוכחי
  - אם השכן האקראי הינו t, קבל
  - שכן אקראי $\leftarrow$ קודקוד נוכחי
  - קדם מונה ב1
- החזר לא

סיבוכיות המקום של החישוב היא לוגריתמית, כי אנו נדרשים לשמור **קדקד נוכחי** ו**מונה**, כל אחד מהם דורש $\log n$ ביטים.

טענה

$st - Conn$ היא NL שלמה.

עבור $A \in NL$ קיימת רדוקציית log-space מA ל$st - Conn$.

הוכחה

כיוון ש$A \in NL$, קיימת מ"ט M ל"ד המכריעה את A בסיבוכיות זיכרון$\leq c \log n$ עבור כל קלט $x \in \{0,1\}^*$ וc קבוע כלשהו. לכן ע"פ חישוב דומה לנעשה בשעה הקודמת, מספר הקונפיגורציות של המכונה עבור קלט בגודל n הוא לכל היותר

$$\underbrace{2^{c \log n}}_{\substack{\#\text{ of possible}\\ \text{contents for}\\ \text{work-tape}}} \cdot \underbrace{c \log n}_{\substack{\#\text{ of possible}\\ \text{places for head}\\ \text{in work-tape}}} \cdot \underbrace{n}_{\substack{\#\text{ of possible}\\ \text{places for head}\\ \text{in input-tape}}} \cdot \underbrace{c'}_{\substack{\#\text{ of internal}\\ \text{machine states}}} = O\left(n^{c+2}\right)$$

רדוקציית הlog-space תקבל באופן הבא: בהינתן קלט x לבעיה A, ניצור גרף $G_A = (V_x, E_x)$ באופן הבא:

V_x יהי קודקוד עבור כל קונפיגורציה של M על הקלט x, וכן קודקוד נוסף מיוחד v_{accept}. נסמן בv_0 את הקודקוד המתאים לקונפיגורציה ההתחלתית.

E_x תהיה קשת בין כל שתי קונפיגורציות שאפשר לעבור ביניהן ע"י צעד בודד של המכונה, וכן תהיה קשת בין כל קונפיגורציה מקבלת לקודקוד v_{accept}.

$$x \in A \iff M \text{ מקבלת את } x \iff (G_x v_0, v_{\text{accept}}) \in st-Conn$$

$$f(x) = (G_x, v_0, v_{\text{accept}})$$

כדי לסיים יש להראות כי $f(x)$ חשיבה בסיבוכיות מקום לוגריתמית. נשים לב כי גודל כל קונפיגורציה הוא $O(\log |x|)$, ובדיקה האם שתי קונפיגורציות הן עוקבות ניתנת להעשות לפיכך בסיבוכיות מקום התלויה בגודל שתי הקונפיגורציות, כלומר סיבוכיות מקום $O(\log |x|)$.

משפט Savitch

$$\text{NSPACE}(\log n) \subseteq \text{DSPACE}\left(\log^2 n\right)$$

הוכחה

למעשה, די להראות אלגוריתם דטרמיניסטי המכריע את $st-Conn$ תוך שימוש במקום $O\left(\log^2 n\right)$ והמשפט ינבע משלמות $st-Conn$.

האלגוריתם הבא מחשב האם יש בגרף G מסלול v באורך$\leq k$ $\Phi(G, v, u, k)$:

1. אם $k = 1$, אם ישנה בG קשת בין u לv החזר 1, אחרת החזר לא
2. עבור כל קודקוד $w \in V(G)$:
   () חשב את $y = \Phi\left(G, u, w, \left\lfloor \frac{k}{2} \right\rfloor\right)$
   () חשב את $z = \Phi\left(T, w, v, \left\lfloor \frac{k}{2} \right\rfloor\right)$
   () החזר כן אם $y \wedge z = 1$
3. החזר לא

קל להיווכח כי האלגוריתם מחזיר כן אם"ם קיים מסלול באורך$\leq k$ מu לv.

ניתוח סיבוכיות המקום: לכל הפעלה של הרקורסיה נדרשת סיבוכיות מקום $O(\log n)$ לשמירת קודקוד נוכחי. עומד הרקורסיה $O(\log n)$ ולכן סה"כ סיבוכיות המקום של האלגוריתם היא $O(\log n \cdot \log k)$, ומכיוון שאפשר לפתור את $st-Conn$ ע"י קריאה לΦ עם $k = n$, נקבל כי סיבוכיות המקום של $st-Conn$ היא $O\left(\log^2 n\right)$.

■