סיבוכיות

© ארזים

5 באפריל 2017

בשיעור שעבר נתנו הגדרה עבור פונקציות שהן time constructible - נרצה לעדכן אותה:

הגדרה 0.1 פונקציה $t(n) \geq n$ היא time constructible אם קיימת מכונת טיורינג שעל קלט 1^n מוציאה את $t(n)$ כפלט תוך $O(t(n))$ צעדים.

1 סיבוכיות זיכרון

הגדרה 1.1 מכונת טיורינג מוגבלת זיכרון היא מכונת טיורינג רגילה עם שלושה סרטים:

1. סרט הקלט, שהוא לקריאה בלבד.
2. סרט הפלט, שהוא לכתיבה בלבד (אחרי כל כתיבה הראש זז ימינה, ואי אפשר לזוז שמאלה.
3. סרט עבודה, כמו במכונת טיורינג רגילה.

בהינתן מכונת טיורינג M מוגבלת זיכרון נאמר כי יש לה סיבוכיות זכרון $\mathrm{DSPACE}(S(n))$, עבור פונקציה $S : \mathbb{N} \to \mathbb{R}$, אם לכל קלט x המכונה עוצרת, ומשתמשת בלכל היותר $O(S|x|)$ תאים בסרט העבודה. קבוצה זו היא קבוצת כל השפות שיש מכונת טיורינג מוגבלת זיכרון המכריעה אותן בזכרון $O(S(n))$.

דוגמא כפל מטריצות בוליאני: יהיו $A, B \in \{0,1\}^{n^2}$ מטריצות $n \times n$. נגדיר

$$(A \cdot_b B)_{i,j} = \bigvee_{k=1}^{n} a_{i,k} \wedge b_{k,j}$$

אפשר לחשוב על השאלה בעיית קשירות. ניקח גרף דו צדדי, וכל אחת מבין A, B תהיה מטריצת שכנויות של אחד הצדדים. המכפלה הבוליאנית מציינת האם יש מסלול בין שני קודקודים.
כפל בוליאני שכזה ניתן לחשב בסיבוכיות מקום $\mathrm{DSPACE}(\log n)$. בהינתן i, j, נרוץ על $1 \leq k \leq n$ ונכתוב אותו יחד עם i, j על הסרט ($3 \log n$ ביטים בינתיים). עם עוד מספר קבוע של תאי זיכרון נוכל לחשב כל $\wedge$. ברגע שנמצא אחד שערכו 1, ניתן לתא את הערך 1, ונמשיך הלאה. אם נעבור על כולם וכולם 0, ניתן לתא 0. כעת נקדם את i, j בהתאם, וחוזר חלילה.

הגדרה 1.2 מסמנים

$$\mathcal{L} := \mathrm{DSPACE}\left(\log n\right) =: \mathrm{Log-space}$$

בנוסף,

$$\mathrm{PSPACE} = \bigcup_{c\in\mathbb{N}} \mathrm{PSPACE}\left(n^c\right)$$

הערה 1.3

$$\mathcal{L} \subseteq \mathcal{P} \subseteq \mathrm{PSPACE}$$
$$\mathcal{L} \subsetneq \mathrm{PSPACE}$$

כאשר

$$\mathcal{P} = \bigcup_{c\in\mathbb{N}} \mathrm{DTIME}\left(n^c\right)$$

הגדרה 1.4 קונפיגורציה של מכונת טיורינג נתונה על ידי המידע הבא: מיקום הראשים (חוץ משל סרט הפלט), מצב הזכרון (מה שכתוב על סרט העבודה), והמצב הפנימי $q \in Q$. למכונת טיורינג עם סיבוכיות זיכרון $\log n \leq S\left(n\right)$ יש $2^{O(S(n))}$ קונפיגורציות אפשריות.

משפט 1.5 אם $\log n \leq S\left(n\right)$ אזי

$$\mathrm{DSPACE}\left(S\left(n\right)\right) \subseteq \mathrm{DTIME}\left(2^{O(S(n))}\right) = \bigcup_{c\in\mathbb{N}} \mathrm{DTIME}\left(2^{c\cdot S(n)}\right)$$

בפרט נובע

$$\mathcal{L} \subseteq \mathcal{P}$$

וכן

$$\mathrm{PSPACE} \subseteq \mathrm{EXP}$$

כאשר

$$\mathrm{EXP} = \bigcup_{c\in\mathbb{N}} \mathrm{DTIME}\left(2^{n^c}\right)$$

הוכחה: אם מכונה דטרמיניסטית מגיעה לאותה קונפיגורציה פעמיים, היא תגיעה אליה אינסוף פעמים. לכן זמן הריצה של מכונה שעוצרת על כל קלט חסום על יד מספר הקונפיגורציות. ■

הגדרה 1.6 גרף הקונפגורציות של מכונת טירוינג $M \in \mathrm{DSPACE}\left(S\left(n\right)\right)$ על קלט x הוא גרף מכוון שקודקודיו הם אוסף הקונפיגורציות של ריצת M על x (יש $2^{O(S(n))}$). קשת מכוונת בין שני קודקודים משמעה שאפשר לעבור מקונפיגורציה אחת לשנייה. נשים לב שדרגת היציאה של כל קודקוד היא 1 (המכונה דטרמיניסטית).

הגדרה 1.7 פונקציה $S:\mathbb{N}\rightarrow\mathbb{N}$ המקיימת $S\left(n\right)\geq\log n$ נקראת space-constructible אם יש מכונת טיורינג מוגבלת בזיכרון שעל קלט 1^{n} מסמנת את התא $S\left(n\right)$ בסרט העבודה, תוך שימוש בזיכרון $O\left(S\left(n\right)\right)$.

משפט 1.8 (משפט היררכיית הזיכרון) אם $S_{1}:\mathbb{N}\rightarrow\mathbb{N}$ היא space-constructible, $S_{2}:\mathbb{N}\rightarrow\mathbb{N}$ היא כזו שמקימת $S_{1}=o\left(S_{2}\right)$, אזי

$$\mathrm{DSPACE}\left(S_{1}\left(n\right)\right)\subsetneq\mathrm{DSPACE}\left(S_{2}\left(n\right)\right)$$

הוכחה: נגדיר מכונה המקבלת שפה שאינה בתוך $\mathrm{DSPACE}\left(S_{1}\left(n\right)\right)$, שמשתמשת בזיכרון $O\left(S_{2}\left(n\right)\right)$.

בהינתן קלט α, נחשוב על α כקידוד של מכונת טיורינג מוגבלת בזיכרון, ונסמן $S_{1}\left(n\right)$ תאי זיכרון בסרט העבודה. נסמלץ את ריצת M_{α} על הקלט α, כאשר, נאמר, $S_{1}\left(n\right)$ התאים הראשונים על סרט העבודה שלנו יסמלצו את סרט העבודה של M_{α}. אם M_{α} צריכה יותר זכרון מזה, נעצור ונקבל את α. כמו כן, נחזיק מונה שיספור $2^{O(s_{1}(n))}$ צעדים, כשהקבוע תלוי בקידוד של M_{α}. אם הריצה נמשכת יותר צעדים מזה, נעצור ונקבל את α. אחרת, M_{α} תקבל או תדחה את α - ואנחנו נתנהג הפוך.

כמות הזיכרון הדרושה היא $O\left(S_{1}\left(n\right)\right)$. נשים לב שהיינו לא זהירים - אפשרנו רק בדיוק $S_{1}\left(n\right)$ זיכרון. נוכל לתקן זאת עם אותו טריק שעשינו במשפט היררכיית הזמן - כופלים בפונקציה שגדלה לאינסוף אבל קטנה מספיק, וכן הלאה. אז מקבלים $\omega\left(S_{1}\left(n\right)\right)$ זיכרון, אבל עדיין $O\left(S_{2}\left(n\right)\right)$. כמו כן, לא קשה לראות שהשפה שלנו שונה מכל שפה מתוך $\mathrm{DSPACE}\left(S_{1}\left(n\right)\right)$. לכן סיימנו. ■

1.1 הרכבה של מכונות מוגבלות זיכרון

נניח $f_{1},f_{2}:\left\{ 0,1\right\} ^{*}\rightarrow\left\{ 0,1\right\} ^{*}$. נרצה לחשב את $f_{2}\left(f_{1}\left(x\right)\right)$. נניח כי $f_{1}\in\mathrm{DSPACE}\left(S_{1}\left(n\right)\right)$, $f_{2}\in\mathrm{DSPACE}\left(S_{2}\left(n\right)\right)$.

פתרון נאיבי: נחשב את $f_{1}\left(x\right)$, נכתוב זאת על סרט העבודה, ואז נחשב את $f_{2}\left(f_{1}\left(x\right)\right)$. זה דורש זיכרון

$$O\left(S_{1}\left(\left|x\right|\right)+\left|f_{1}\left(x\right)\right|+S_{2}\left(\left|f_{1}\left(x\right)\right|\right)\right)$$

זה לא כל כך טוב - למשל, עבור מכפלה של כמה מטריצות בוליאניות, כל כפל לוקח $\log n$ זיכרון, אבל הכתיבה לוקחת n^{2} זיכרון.

פתרון טוב יותר: נחשב את f_2. אם תוך כדי החישוב נצטרך סימבול קלט מסויים מתוך $f_1(x)$, נחשב את הסימבול הזה מתוך f_1. זה דורש זיכרון

$$O\left(S_1(|x|)+S_2(|f_1(x)|)\right)$$

הערה 1.9 מתקיים

$$|f_1(x)|\leq 2^{O(S_1(|x|))}$$

לפי משפט שהוכחנו. לכן, לומר איזה ביט מהפלט $f_1(x)$ אנו רוצים דורש $\log|f_1(x)|=O\left(S_1(|x|)\right)$ זכרון.

נשווה כעת בין זמני הריצה. נניח כי $f_1\in\mathrm{DTIME}\left(t_1(n)\right)$, $f_2\in\mathrm{DTIME}\left(t_2(n)\right)$. הפתרון הנאיבי רץ בזמן

$$O\left(t_1(|x|)+t_2(|f_1(x)|)\right)$$

לעומת זאת, הפתרון השני רץ בזמן

$$O\left(t_1(|x|)\,t_2(|f_1(x)|)\right)$$

מסקנה 1.10 ניתן לחשב חזקת k של מטריצה בוליאנית (כפל רגיל או כפל בוליאני) בזכרון $O(\log k\log n)$ וזמן $n^{O(\log k)}$.

בהנתן גרף G עם n קודקודים, תהי A מטריצת השכנויות שלו. נוסיף לולאה עצמית לכל הקודקודים ונסמן A' את מטריצת השכנויות החדשה (שמנו 1 באלכסון הראשי). אזי $(A')^k_{i,j}=1$ אם ורק אם יש מסלול בין קודקודים i,j באורך לכל היותר k. לכן $(A')^n_{i,j}=1$ אם ורק אם יש מסלול כלשהו בין קודקודים i,j.

מסקנה 1.11 ניתן לבדוק קשירות בגרף (מכוון או לא) בעזרת $O\left(\log^2|V|\right)$ זכרון.

כמובן, ידועים גם אלגוריתמים פולינומיאליים בזכרון פולינומיאלי (למשל BFS). שאלה סבירה היא האם יש אלגוריתם עם זכרון לוגריתם וזמן ריצה פולינומיאלי? ידוע שכן עבור גרפים לא מכוונים (תוצאה של ריינגולד, המאמר באתר הקורס), אבל השאלה הזו עבור גרפים מכוונים היא פתוחה.

הגדרה 1.12 STCON היא הבעיה הבאה: בהינתן גרף מכוון $G=(V,E)$ ושני קודקודים $s,t\in V$, נרצה להכריע האם יש מסלול מכוון בין s ובין t. אנחנו יודעים שמתקיים

$$\mathrm{STCON}\in\mathcal{P}\cap\mathrm{DSPACE}\left(\log^2 n\right)$$

הגדרה 1.13 (רדוקציית קארפ) נאמר כי $\varphi : \{0,1\}^* \to \{0,1\}^*$ הוא רדוקציה בין שפות A, B אם $x \in A$ אם ורק אם $\varphi(x) \in B$. אם φ ניתנת לחישוב בזמן פולינומיאלי נסמן $A \leq_p B$. אם φ ניתנת לחישוב בזכרון לוגריתמי נסמן $A \leq_L B$.

הערה 1.14 הרדוקציות שראינו בקורס במודלים חישוביים הן כולן רדוקציות L.

טענה 1.15
1. אם $A \leq_p B \leq_p C$ אזי $A \leq_p C$.
2. אם $A \leq_L B \leq_L C$ אזי $A \leq_L C$.

הוכחה: נובע מהניתוח שביצענו לגבי הרכבת פונקציות. ■

הגדרה 1.16 CVAL היא הבעיה הבאה: בהינתן מעגל בוליאני עם n קלטים ופלט אחד, וקלט $a \in \{0,1\}^n$, נשאל האם המעגל מקבל את a - מחזיר 1.

ראינו כבר כי

$$\mathrm{CVAL} \in \mathcal{P}$$

ובתרגול נראה כי אם המעגל הוא נוסחה אז ניתן לבצע את החישוב בזכרון לוגריתמי.

הגדרה 1.17 שפה A היא קשה למחלקה $\mathcal{C}$ תחת רדוקציות p (או באופן סימטרי L) אם לכל $B \in \mathcal{C}$ מתקיים $B \leq_p A$ ($B \leq_L A$). אם בנוסף $A \in \mathcal{C}$, אזי A שלמה למחלקה $\mathcal{C}$ תחת רדוקציות p (L).

משפט 1.18 CVAL היא שלמה למחלקה $\mathcal{P}$ תחת רדוקציות L.

הוכחה: נראה כעת את הרעיון, ובשיעור הבא את ההוכחה הפורמלית.
תהי $A \in \mathcal{P}$ ויהי x קלט עבורה. נתבונן בטבלת החישוב של מכונת טיורינג המכריעה את A בזמן פולינומי. אם יש מכונת טיורינג M שמכריעה את A בזמן n^c לכל היותר, אז הטבלה תהיה מטריצה $n^c \times n^c$, בה השורה i מכילה את תוכן הסרט של M בזמן הריצה על הקלט x בזמן i. כמו כן מסומן התא שעליו מצביע הראש הקורא באופן מיוחד, ובאותו תא נכתוב גם את המצב שבו נמצאת המכונה.
בלי הגבלת הכלליות נניח כי אחרי הכרעת x, M מוחקת את כל הסרט וכותבת את התוצאה בתא הראשון.
נשים לב שבמעבר בין שורה לשורה, רק התא שעליו הראש מצביע ושני התאים שמצדדיו יכולים להשתנות.
נמשיך בשיעור הבא. ■