

סיבוכיות

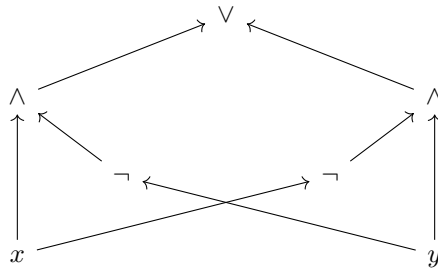
© ארזים

4 באפריל 2017

1 חישוביות ומודלים

1.1 מעגלים בוליאניים

נראה דוגמה למעגל בוליאני שמחשב את $x \oplus y$.



הגדרה 1.1 נקרא למספר הקשתות הנכנסות לשער fan-in, ולכמות הקשתות היוצאות fan-out.

לעתים נביט רק במעגלים עם fan-in לכל היותר 2. כיוון שלמעגל יכולים להיות הרבה שערי פלט, מעגלים בוליאניים יכולים גם לפתור בעיות חיפוש.

הגדרה 1.2 נוסחה בוליאנית היא מעגל בוליאני בו fan-out של כל שער הוא לכל היותר 1. במעגלים כאלה, הגרף הוא עץ.

1.1.1 מידות סיבוכיות של מעגלים בוליאניים

הגדרה 1.3 גודל המעגל הוא מספר הקשתות במעגל.

במעגלים על fan-in לכל היותר 2, גודל המעגל שקול (עד כיד פקטור 2) לכמות הקודקודים.

הגדרה 1.4 עומק של מעגל הוא אורך המסלול המכוון הארוך ביותר בין קלט לפלט.

בדוגמה של $x \oplus y$, הגודל הוא 8, כמות הקודקודים היא 7, והעומק הוא 3. למעשה, גודל הוא מדד לזמן חישוב, ועומק הוא מדד לזמן מקבילי.

הגדרה 1.5 הגדרנו מתי מעגל מחשב פונקציה $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$. הסיבוכיות של f , שנסמנה $S(f)$, היא

$$S(f) = \min_{\substack{C \text{ is a} \\ \text{boolean circuit} \\ \text{that computes } f}} S(C)$$

כאשר $S(C)$ הוא גודל המעגל.

הגדרה 1.6 הגדרנו גם מתי סדרת מעגלים $\{C_n\}$ מחשבת שפה $L \subseteq \{0, 1\}^*$. סיבוכיות המעגלים של שפה L מוגדרת כך: נאמר שניתן לחשב שפה L על ידי מעגלים בגודל $S(n)$ אם יש סדרת מעגלים $\{C_n\}$ עם

$$S(C_n) \leq S(n)$$

כאשר C_n מחשב את

$$L_n = L \cap \{0, 1\}^n$$

משפט 1.7 לכל שפה L יש סדרת מעגלים המחשבת אותה בגודל $O(n2^n)$.

הוכחה: תהי $L_n \subseteq \{0, 1\}^n$. נגדיר

$$f_n : \{0, 1\}^n \rightarrow \{0, 1\}$$

$$f_n(x) = \begin{cases} 1 & x \in L_n \\ 0 & x \notin L_n \end{cases}$$

עבור f , קיימת נוסחת DNF שמחשבת אותה. זו תהיה נוסחה שהיא \vee על לכל היותר 2^n דברים (כל המילים האפשריות), כשכל דבר שכזה הוא \wedge על לכל היותר n קלטים (ביטי פלט או שלילתם). בסך הכל יש לנו לכל היותר $2n2^n$ שערים - n קלטים, אולי עוד n שלילות, וכל זה כפול 2^n מילים לכל היותר. לכן סיימנו. ■

עובדה למעשה, אפשר לחשב כל שפה L בגודל $O\left(\frac{2^n}{n}\right)$.

משפט 1.8 לרוב הפונקציות $f : \{0, 1\}^n \rightarrow \{0, 1\}$ הסיבוכיות היא $S(f) = \Omega\left(\frac{2^n}{n}\right)$.

כיוון אפשרי להוכחה הוא להציג פונקציה והלראות שצריך מעגל גדול בשביל לחשב אותה. האמת היא שאין לנו מושג איך עושים את זה. לא סתם:

בעיה פתוחה מצאו פונקציה $f : \{0, 1\}^n \rightarrow \{0, 1\}$ עבורה $S(f) > 5n$. **הוכחה:** (של המשפט האחרון) נספור. כמות הפונקציות $f : \{0, 1\}^n \rightarrow \{0, 1\}$ היא 2^{2^n} . נראה שאין מספיק שערים מגודל לכל היותר $\frac{2^n}{10n}$, כשאנחנו מניחים כי fan-in הוא לכל היותר 2 (אפשר להניח ולשלם לכל היותר בפקטור כפלי) ונספור קודקודים (כי זה שקול לקשתות). כמה מעגלים יש בגודל S ? ראשית יש לתאר את גרף החישוב. לכל שער יש לכל היותר 2 בנים, ולכן יש $\binom{S}{2}$ אפשרויות לבנים לכל שער. בסך הכל כמות הגרפים האפשריים היא לכל היותר

$$\binom{S}{2}^S \leq S^{2S}$$

כעת יש לקבוע את הפעולות בקודקודים פנימיים. כל קודקוד שאינו קלט הוא \neg, \vee, \wedge , ולכן יש לכל היותר 3^S אפשרויות. בסך הכל, כמות העגלים היא לכל היותר:

$$S^{2S} \cdot 3^S = (3S^2)^S \leq S^{3S}$$

נשים לב שאם $S < \frac{2^n}{4n}$, אזי

$$S^{3S} \leq \left(\frac{2^n}{4n}\right)^{3 \frac{2^n}{4n}} < (2^n)^{\frac{3}{4} \frac{2^n}{n}} = (2^{2^n})^{\frac{3}{4}} \ll 2^{2^n}$$

- לכן מספר המעגלים בגודל לכל היותר $\frac{2^n}{4n}$ קטן בהרבה ממספר הפונקציות הללו. למעשה ראינו שכל שפה אפשר לחשב בסדרת מעגלים.

1.2 חזרה למכונות טיורינג

עד כה דיברנו על מכונות עם סרט אחד. אפשר להגדיר אותן עם k סרטים - כל מה שישתנה הוא שפונקציית המעברים תיראה כך:

$$\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{\leftarrow, \rightarrow\}^k$$

משפט 1.9 כל מכונת טיורינג עם k סרטים אפשר לסמלץ על ידי מכונות טיורינג עם סרט אחד, ואם המכונה המקורית רצה בזמן T אז המכונה המסמלצת תרוץ בזמן $O(T^2)$.

עובדה $\Omega(T^2)$ הוא הכרחי - יש מכונות שזה הזמן הנדרש. שפת הפלינדרומים ניתנת להכרעה על ידי מכונה עם שני סרטים בזמן לינארי, אבל בסרט אחד אפשר להוכיח שנדרש זמן ריבועי.

1.2.1 מכונת טיורינג אוניברסלית

נאמר כי U היא מכונת טיורינג אוניברסלית אם על קלט $\langle M, w \rangle$, כאשר M תיאור של מכונת טיורינג, w קלט שלה, U מסמלצת את החישוב של U על w .

משפט 1.10 קיימת מכונת טיורינג אוניברסלית. יתרה מכך, יש U כזו שאם M מכריעה את w תוך T צעדים, U תרוץ $O(T \log T)$ צעדים על $\langle M, w \rangle$.

למכונת טיורינג אוניברסלית נקרא גם מחשב.

1.2.2 זמן הריצה של מכונת טיורינג

נאמר כי מכונת טיורינג מכריעה קלט x בזמן t אם M רצה t צעדים (כלומר t הפעלות של δ) ומכריעה את x . נאמר כי שפה L ניתנת להכרעה בזמן $t(n)$, אם $L \in DTIME(t(n))$, אם יש מכונת טיורינג M שעל קלט x רצה בזמן $O(t(|x|))$ ומכריעה האם $x \in L$.

1.2.3 מכונות טיורינג עם אורקל (סב־פרוצדורות)

תזכורת אם $L, L' \subseteq \{0, 1\}^*$ שפות, נאמר כי פונקציה $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ היא רדוקציה מהשפה L לשפה L' אם מתקיים

$$x \in L \iff f(x) \in L'$$

אם f חשיבה (על ידי מכונת טיורינג) מסמנים

$$L \leq_{TM} L'$$

אם f חשיבה בזמן פולינומיאלי, מסמנים גם

$$L \leq_p L'$$

רדוקציות כאלה נקראות רדוקציות קארפ.

דוגמא כדי לחשב כפל מטריצות אפשר או צריך להשתמש בהרבה תת חישובים של מכפלות סקלריות.

רדוקציות בהן כדי להכריע את L קוראים הרבה פעמים לחישוב L' נקראות רדוקציות טיורינג.

הגדרה 1.11 (מכונת טיורינג עם אורקל) מכונת טיורינג עם אורקל לשפה A היא מכונת טיורינג רגילה עם התוספות הבאות:

1. יש סרט ייעודי שנקרא סרט האורקל או סרט השאילתות.

2. יש שלושה מצבים חדשים: $q_{query}, q_{yes}, q_{no}$.

תיאור הריצה: כאשר המכונה נכנסת למצב q_{query} , מתבצעת שאילתה לשפה A לגבי הקלט הרשום בסרט השאילתה. המכונה עוברת למצב q_{yes} אם המחרוזת בסרט השאילתה שייכת לשפה A , ואחרת למצב q_{no} .

הגדרה 1.12 תהי A שפה ותהי $t : \mathbb{N} \rightarrow \mathbb{R}$ פונקציה. נסמן $DTIME(t(n))^A$ את אוסף השפות L עבורן יש מכונת טיורינג עם אורקל לשפה A שמכריעה את השפה בזמן $O(t(n))$.

סימון M^A מסמן כי M מכונת טיורינג עם אורקל לשפה A .

1.2.4 משפט היררכיית הזמן

נרצה להוכיח שיותר זמן נותן יותר כח.

הגדרה 1.13 פונקציה $t : \mathbb{N} \rightarrow \mathbb{N}$ היא time constructible אם יש מכונת טירינג, שעל קלט n בבינארי רצה בזמן $O(t(n))$, וכותבת כפלט את $t(n)$ בבינארי.

משפט 1.14 (משפט היררכיית הזמן) לכל פונקציה $t : \mathbb{N} \rightarrow \mathbb{N}$ כך שמתקיים $n \leq t(n)$, כאשר t היא time constructible, לכל שפה A ולכל פונקציה $T : \mathbb{N} \rightarrow \mathbb{R}$ כך שמתקיים

$$t \log t = o(T)$$

כלומר

$$\lim_{n \rightarrow \infty} \frac{t(n) \log t(n)}{T(n)} = 0$$

מתקיים

$$\begin{aligned} DTIME(t(n)) &\subsetneq DTIME(T(n)) \\ DTIME(t(n))^A &\subsetneq DTIME(T(n))^A \end{aligned}$$

הוכחה: נוכיח את הטענה ללא אורקל (ההוכחה עם אנלוגית). הרעיון הוא בדיוק כמו ברעיון הלכסון, כשמראים אי כריעות של בעיית העצירה.

בקצרה: על קלט α , נריץ את המכונה M_α שאותה α מייצג על הקלט α במשך $t(|\alpha|)$ צעדים, ונענה תשובה הפוכה.

תהי U מכונת טירינג אוניברסלית שמסמלצת מכונות שרצות בזמן $O(r(n))$ בזמן $O(r(n) \log r(n))$. נגדיר מכונת טירינג חדשה, L תהיה השפה שאותה מכונה תקבל. המכונה שלנו תפעל כך:

בהינתן קלט α , נריץ את U על $\langle M_\alpha, \alpha \rangle$, כאשר M_α היא מכונת הטיורינג שאותה α מתארת. אם α לא מתארת מכונת טירינג, נקבל את α , ואם היא כן, נסמלץ את ריצת M_α על α במשך $t(|\alpha|)$ שלבים. אם M_α עצרה בזמן זה, המכונה שלנו תוציא תוצאה הפוכה (נקבל אם M_α דחתה ולהיפך). אם M_α לא עצרה, נקבל את α .

תהי L השפה שהדרנו. לפי הסמלוך של U , $L \in DTIME(t \log t)$. כמו כן, $L \notin DTIME(t(n))$, כי אחרת יש מכונת טירינג שתיאורה הוא α , עבור איזשהו α המכריעה את L בזמן $O(t(n))$ - אבל דאגנו שהמחרוזת תהיה בתוך L אם ורק אם היא לא בשפה של M_α . ■

הערה 1.15 קריטי שהפונקציה t היא time constructible, כדי שנדע כמה זמן לסמלץ. אחרת הטענה לא נכונה.

הערה 1.16 אנחנו תמיד מניחים כי לכל מכונת טירינג, יש אינסוף מחרוזות המתארות את M (רק כדי שלא נתעסק/נתקע עם בעיה באורכי קלט ממש קטנים, בהם ייתכן $t(|\alpha|) \log t(|\alpha|) > T(|\alpha|)$).

הערה 1.17 הגדרנו $DTIME(t(n))$ בעזרת $O(t(n))$, ולכן אולי צריך לסמלץ יותר מאשר עם $f : \mathbb{N} \rightarrow \mathbb{R}$ פונקציה לקחת פונקציה $t(|\alpha|)$

$$t \log t f^2 = o(T)$$

וכן

$$\lim_{n \rightarrow \infty} f(n) = \infty$$

ובהוכחה נסמלץ את ריצת M_α במשך $t(|\alpha|) f(|\alpha|)$ צעדים.

1.3 סיבוכיות זיכרון

נרצה להעריך כמה זיכרון צריך כדי לחשב כל ביט בפלט. בכפל מטריצות למשל, אפשר לחשב את התא i, j במטריצה בעזרת $\log n$ זיכרון כאשר אורך הקלט הוא $2n^2$ ואורך הפלט הוא n^2 .