סיבוכיות

מאיה פרבר ברודסקי

סיכומי ההרצאות של פרופ' אמנון תא־שמע, סמסטר א' תשפ"ב.

תוכן עניינים

1 מעגלים בוליאניים והיררכיית מעגלים

הגדרה 1.1 מעגל בוליאני גרף מכוון חסר מעגלים עם קודקודים מבין n קלטים $x_1,\ldots,x_n \in \{0,1\}$, שערים $\wedge,\vee,\neg$ ויציאה אחת, וקשתות ביניהם.

גודל המעגל הוא מספר הקודקודים + מספר הקשתות, ו**עומק המעגל** הוא אורך המסלול הארוך ביותר מקלט לפלט.

טענה 2.1 לכל פונקציה $f\colon \{0,1\}^n \to \{0,1\}$ יש מעגל בוליאני על n ביטים שמחשב אותה.

הוכחה: נסתכל על טבלת האמת של f, ונבנה מעגל עם שער $\vee$ שהקלטים עבורו הם שערי $\wedge$, אחד לכל קלט $x \in \{0,1\}^n$ כך ש־$f(x) = 1$. ■

הגדרה 3.1 תהי $L \subseteq \{0,1\}^*$. נאמר ש$L \in \text{Size}(s(n))$ אם לכל $n \in \mathbb{N}$ יש מעגל C_n בגודל $s(n) \geq$ שמקבל קלט באורך n ומחזיר 1 אם ורק אם הקלט ב$L \cap \{0,1\}^n$. כמו כן, נגדיר $\text{Size}(poly) = \bigcup_c \text{Size}(n^c)$.

טענה 4.1 כל שפה L מקיימת $L \in \text{Size}\left(\frac{2^n}{n}\right)$.

טענה 5.1 יש פונקציה $f\colon \{0,1\}^n \to \{0,1\}$ שאין לה מעגל בגודל s, עבור s כך ש־$n > \log s + \log\log s + O(1)$.

הוכחה: נספור כמה מעגלים בגודל s יש. נייצג מעגל על ידי לכל קשת את זוג הקודקודים שהיא מחוברת אליהם, ולכל קודקוד את ה־label שלו, קלט/פלט/שער. זה דורש לכל היותר $O(s\log s) + O(s\log n) = O(s\log s)$ ביטים, ולכן יש לכל היותר $2^{O(s\log s)} = s^{O(s)}$ מעגלים כנ"ל.

כעת, יש 2^{2^n} פונקציות על n ביטים אבל $s^{c\cdot s}$ מעגלים בגודל s, ולכן אם $2^{2^n} > s^{c\cdot s}$ אז משיקולי ספירה יש פונקציה על n ביטים שאין לה מעגל בגודל s. מתקיים

$$2^{2^n} > s^{c\cdot s} \iff 2^{2^n} > 2^{c\cdot s\log s} \iff 2^n > c\cdot s\log s \iff n > \log s + \log\log s + O(1)$$

■

דוגמה 6.1 לדוגמה עבור $s(n) = \frac{2^n}{n^2}$ יש פונקציה על n ביטים עבורה אין מעגל בגודל s.

משפט 7.1 היררכיה של מעגלים לכל $n < s(n) < \frac{2^n}{n^2}$ יש שפה $L \in \text{Size}(s(n) + 10n)$ כך ש־$L \notin \text{Size}(s(n))$.

הוכחה: נשתמש בטיעון היברידי. יודעים שיש פונקציה קשה g על n ביטים שדורשת מעגלים בגודל $\frac{2^n}{n^2}$, ופונקציה קלה f על n ביטים שדורשת מעגלים בגודל n.

נגדיר $h_i\colon \{0,1\}^n \to \{0,1\}$ על ידי $h_i(x) = \begin{cases} g(x) & x \leq i \\ f(x) & x > i \end{cases}$, אז $h_0 = f, h_{2^n} = g$ ומתקיים $\text{Size}(h_0) = n, \text{Size}(h_{2^n}) \geq \frac{2^n}{n^2}$.

נטען כי $\text{Size}(h_{i+1}) - \text{Size}(h_i) \leq 10n$. אכן, h_i, h_{i+1} זהים מלבד אולי בקלט $i+1$, ולכן בהינתן מעגל לh_i נוכל לבנות מעגל לh_{i+1} שמחזיר את $g(i+1)$ אם $x = i+1$ (השוואת מחרוזות, אפשר במעגל בגודל $10n$), ואחרת את התוצאה של h_i.

לסיום ההוכחה, נמצא את הi הראשון כך ש$\text{Size}(h_i) > s(n)$, אז המקום הקודם מקיים $\text{Size}(h_{i-1}) \leq s(n)$ ולכן

$$s(n) < \text{Size}(h_i) \leq \text{Size}(h_{i-1}) + 10n = s(n) + 10n$$

■

2 חישוב אוניפורמי והיררכיה

2.1 זמן ריצה והיררכיית זמן

מעכשיו נרצה להתעסק באלגוריתמים בעלי יצוג סופי שטובים לכל קלט באורך סופי (לא חסום). נתאר מספר מודלים כאלו:

הגדרה 1.2 מכונת **random access machine** היא מכונה עם סרט עבודה חצי־אינסופי שהקלט בהתחלה שלו, ופקודות LOAD ו־STORE מכל מקום על הסרט. כל פקודה בסיסית לוקחת יחידת זמן אחת.

הגדרה 2.2 מכונת טיורינג היא מודל עם סרט עבודה חצי־אינסופי כאשר הקלט בהתחלה שלו וראש קורא שמתחיל בתחילת הסרט, קבוצת מצבים פנימיים Q סופית, ופונקציית מעבר $\delta\colon Q\times\Sigma\to Q\times\Sigma\times\{\text{left},\text{stay},\text{right}\}$.

הגדרה 3.2 שפה L שייכת ל־$\text{Time}\left(t\left(n\right)\right)$ אם יש מכונת טיורינג M שעבור כל קלט x עוצרת תוך $\geq t\left(n\right)$ זמן ו־$x\in L$ אם ורק אם M מקבלת.

משפט 4.2 יש שפה L שלא ניתנת לפתרון על ידי שום מכונת טיורינג.

הוכחה: מטיעון ספירה, יש $2^{\aleph_0}$ שפות אבל $\aleph_0$ מכונות טיורינג (ניתן לייצג מכונת טיורינג כמחרוזת סופית). ■

טענה 5.2 השפה $\text{HALT}=\left\{x\in\{0,1\}^{*}\mid\text{the TM represented by }x\text{ halts on }x\right\}$ אינה ניתנת לפתרון ע"י מכונת טיורינג.

הוכחה: נניח בשלילה שיש מכונת טיורינג B שפותרת את HALT, נבנה מכונה חדשה A שמקבלת מחרוזת x, מסמלצת את B על x ואם B מחזירה כן נכנסת ללולאה אינסופית, ואחרת עוצרת. כעת נריץ את A על הייצוג של A כמחרוזת. אם היא עוצרת, אז מהגדרה B תחזיר עליה כן, ולכן היא תכנס ללולאה אינסופית, סתירה. אם היא לא עוצרת, אז מהגדרה B תחזיר עליה לא, ולכן היא תעצור, שוב סתירה. לכן אין מכונה B כזו. ■

משפט 6.2 ההיררכיה לזמן אם $T\left(n\right)\gg t\left(n\right)$ (כלומר לסמלץ $t\left(n\right)$ צעדים של מכונה לוקח לכל היותר $T\left(n\right)$ זמן, במכונת טיורינג עם סרט אחד מספיק $T\left(n\right)=\Omega\left(t\left(n\right)^{2}\right)$) אז יש שפה $L\in\text{Time}\left(T\left(n\right)\right)$ כך ש־$L\notin\text{Time}\left(t\left(n\right)\right)$.

הוכחה: נבנה מכונה M עם קלט $x\in\{0,1\}^{n}$ שתסמלץ את M_x (המכונה המיוצגת ע"י x) על x למשך $t\left(n\right)$ צעדים, אם היא עצרה תענה הפוך מהתוצאה שלה, ואחרת תעצור ותענה משהו (לא משנה). נסמן ב־L את השפה של M.

מתקיים $L\in\text{Time}\left(T\left(n\right)\right)$, כי $T\left(n\right)\gg t\left(n\right)$ ולכן אכן ניתן לסמלץ בזמן $T\left(n\right)$.

אבל $L\notin\text{Time}\left(t\left(n\right)\right)$, כי נניח בשלילה שכן אז נריץ את M על הייצוג של M שנסמן m, M_m בהכרח תעצור ולכן M על m תחזיר את ההפך מ־$M_m=M$ על m, סתירה. ■

הגדרה 7.2 מחלקות סיבוכיות $P=\bigcup_{c}\text{Time}\left(n^{c}\right),E=\bigcup_{c}\text{Time}\left(2^{c\cdot n}\right),EXP=\bigcup_{c}\text{Time}\left(2^{\left(n^{c}\right)}\right)$.

טענה 8.2 $P\subseteq\text{Size}\left(poly\right)$.

הוכחה: תהי $L\in P$, אז יש קבוע c ומכונה M שפותרת את L בזמן n^{c}. בהינתן $n\in\mathbb{N}$, נבנה מעגל C_n עם n קלטים וגודל $n^{O(c)}$ שמחשב את $L\cap\{0,1\}^{n}$, ומכאן נסיק $L\in\text{Size}\left(poly\right)$. נסתכל על טבלת החישוב של המכונה M על קלט x באורך n:

זו טבלה ברוחב n^{c} ובאורך n^{c}, כאשר כל שורה מייצגת את מצב הזכרון של מכונה בזמן נתון, וכאשר מתקדמים בשורות מתקדמים בזמן.

בתא ה(i,t) בטבלה מופיע ביט אחד כדי לציין האם הראש הקורא על התא, $\log|\Sigma|$ ביטים עבור מה שכתוב בתא, וכן $\log|Q|$ ביטים עבור המצב של המכונה באותו התא, כאשר דורשים נכונות רק אם הראש הקורא נמצא על התא.

בין השורות נרצה לסמלץ את הפונקציה δ על ידי חלקי מעגלים. נשים לב שניתן לחשב את התא $(i,t+1)$ על ידי התאים $(i-1,t),(i,t),(i+1,t)$ בלבד, כלומר הוא פונקציה שלהם. כל פונקציה כזו אפשר לייצג על ידי מעגל, ואפילו בגודל קבוע כי זו פונקציה עם תחום וטווח בגודל קבוע. נשים את המעגל הזה בין התאים הנ"ל.

אז המעגל C_n הוא בגודל פולינומי (יש $n^c\cdot n^c=n^{2c}$ תאים ולפני כל תא מעגל בגודל קבוע), ואכן מכריע את L עבור גודל קלט n כי הוא מסמלץ את L, ולכן סיימנו. ∎

2.2 זכרון והיררכיית זכרון

הגדרה 9.2 נאמר כי מכונת טיורינג M עם שלושה סרטים (סרט קלט לקריאה בלבד, סרט עבודה לקריאה וכתיבה, סרט פלט לכתיבה בלבד) משתמשת בזכרון $s(n)$ אם בכל שלב משתמשים רק ב$s(n)$ תאים בסרט העבודה, כאשר n הוא אורך הקלט.

דוגמה 10.2 כפל מטריצות בוליאניות סיבוכיות זמן $O\left(n^3\right)$, אפשר לשפר עם אלגוריתמים כמו Strassen.

סיבוכיות זכרון $O(\log n)$ כי צריך לשמור רק אינדקסים ואת האיבר הנוכחי שיכול לקחת לכל היותר $\log n$ ביטים.

דוגמה 11.2 העלאת מטריצה בחזקת k סיבוכיות זכרון $O(\log n\cdot\log k)$, משתמשים בiterated squaring אבל במקום לשמור את $A^2,A^4,\ldots$ כל פעם שצריך ערך במטריצה קודמת "קוראים לפרוצדורה" ומחשבים אותו מחדש. בסך הכל צריך $\log n$ ביטים כדי לחשב ערך מסוים ועומק המחסנית הוא לכל היותר $\log k$, ולכן $\log n\cdot\log k$ זכרון - הרכבה של רדוקציות space היא סכום הspace שלוקחת כל רמה.

הערה 12.2 המספרים גדלים ככל שעוברים רמות אבל נשארים $O(\log n)$, כי המספרים בA^2 הם לכל היותר n, המספרים בA^4 הם לכל היותר n^3 וכן הלאה, ו$\log\left(n^k\right)=k\log n=O(\log n)$.

טענה 13.2 קשר בין זכרון לזמן אם M מכונת טיורינג שרצה על קלט x בזכרון $s(n)\geq\log n$ אז M רצה על x בזמן $2^{O(s(n))}$.

הוכחה: נסתכל על גרף הקונפיגורציות של המכונה M על הקלט x (בקונפיגורציה יש מצב, סרט עבודה, מיקום ראשים קוראים), מספר הקונפיגורציות האפשריות הוא

$$|V|=|Q|\cdot 2^{s(n)}\cdot n\cdot s(n)=2^{O(s(n))}$$

נשים קשת מc_1 לc_2 אם פונקציית הδ של M מעבירה מc_1 לc_2. דרגת היציאה בגרף היא 1 כי המכונה דטרמיניסטית.

ריצה של המכונה M על x היא מסלול מc_{init} לקונפיגורציה מסיימת, וזמן הריצה הוא אורך המסלול, שהוא לכל היותר $|V|=2^{O(s(n))}$ (אחרת יש לולאה והמכונה לא עוצרת). לכן גם זמן הריצה של M על x הוא לכל היותר $2^{O(s(n))}$. ∎

משפט 14.2 היררכיית זכרון אם $s(n)\geq\log n$, $S(n)$ היא space-constructible ו$s(n)=o(S(n))$ אז $\mathrm{Space}(s(n))\subsetneq\mathrm{Space}(S(n))$.

הגדרה 15.2 מחלקות זכרון

$$L=\bigcup_c\mathrm{Space}(c\log n)$$

$$PSPACE=\bigcup_c\mathrm{Space}\left(n^c\right)$$

מתקיים $L\subseteq P\subseteq PSPACE\subseteq EXP$.

3 בעיות שלמות

3.1 P־שלמות ביחס לרדוקציות L

הגדרה 1.3 רדוקציה $L_2 \leq_\varphi L_1$ אם $\varphi\colon \{0,1\}^* \to \{0,1\}^*$ וכן $x\in L_1 \iff \varphi(x)\in L_2$.

הגדרה 2.3 נגיד ששפה $A\in P$ היא P־שלמה (ביחס לרדוקציות LogSpace) אם לכל $B\in P$ יש $\varphi\in L$ כך ש$B\leq_\varphi A$.

טענה 3.3 נגדיר $CVAL$, הבעיה של חישוב $C(x)$ עבור מעגל C וקלט x. אזי $CVAL$ היא P־שלמה.

הוכחה: $CVAL\in P$, כי אפשר להריץ מעגל בזמן פולינומי על ידי BFS.

כעת תהי $B\in P$, ראינו $P\subseteq \mathrm{Size}(\mathrm{poly})$ ולכן יש מעגל C_B בגודל פולינומי שפותר את B, ואפשר לבנות אותו במקום לוגריתמי (רצים בלולאה על המקומות בטבלה, מוסיפים את החלקים ביניהם, יש n^{2c} מקומות בטבלה). כלומר הרדוקציה תהיה $\varphi(x)=(C_B,x)$, ואכן מתקיים $\varphi(x)\in CVAL \iff C_B(x)=1 \iff x\in B$, וכן $\varphi\in L$, ולכן $B\leq_L A$. ■

טענה 4.3 אם A היא P שלמה, $A\in L$, אז $P=L$.

הוכחה: $L\subseteq P$ ראינו, כעת תהי $B\in P$ אז יש רדוקציה $B\leq_\varphi A$, ולכן על מנת לפתור את B נוכל להרכיב את φ ואת האלגוריתם בL שפותר את A, זו הרכבה של מכונות space ולכן זה לוקח לכל היותר כמו הסכום (אותו טריק של לחשב את הביט המתאים כשצריך), שהוא עדיין לוגריתמי. לכן $B\in L$. ■

3.2 NP־שלמות ביחס לרדוקציות L

הגדרה 5.3 שפה $L\in NP$ אם קיימת מ"ט דטרמיניסטית $M(x,y)$ כך ש$|y|=\mathrm{poly}(|x|)$ (עבור פולינום קבוע כלשהו) ולכל $x\in\{0,1\}^*$ מתקיים $x\in L \iff \exists y.M(x,y)=1$.

טענה 6.3 $P\subseteq NP\subseteq PSPACE$ ($P\subseteq NP$ ברור, $NP\subseteq PSPACE$ כי אפשר לעבור על כל הy באורך $p(|x|)$ ולבדוק).

טענה 7.3 נגדיר $CSAT$, הבעיה של בדיקה אם יש קלט שמספק מעגל נתון. אזי $CSAT$ היא NP־שלמה.

הוכחה: $CSAT\in NP$ כי אפשר לבנות מוודא $M(C,y)$, שבודק אם $C(y)=1$ (העד הוא השמה שמספקת את המעגל).

תהי $B\in NP$, אז יש מוודא M עבורה. נראה $B\leq_L CSAT$, כאשר הרדוקציה היא $\varphi(x)=C_{M(x,\cdot)}$, כלומר φ מחזירה מעגל ששקול למכונה שמקבלת קלט y ומחשבת את $M(x,y)$ (כאשר x קבוע hardwired). אז אכן

$$x\in B \iff \exists y.M(x,y)=1 \iff \exists y.C_{M(x,\cdot)}(y)=1 \iff \varphi(x)=C_{M(x,\cdot)}\in CSAT$$

וכן הרדוקציה לוגריתמית, כדרוש. ■

טענה 8.3 SAT (הקלט הוא נוסחת CNF) היא NP־שלמה.

הוכחה: נראה רדוקציה $CSAT\leq_L SAT$. בהינתן מעגל, בונים נוסחה שהמשתנים שלה הם הקלטים והשערים, ולדוגמה שער AND ממירים לפסוקית $x\wedge y \iff out$, ובדומה ליתר השערים, והנוסחה היא הAND של כל הפסוקיות יחד עם פסוקית נוספת שבודקת ששער הפלט הוא T. ■

3.3 EXP-שלמות ביחס לרדוקציות L

טענה 9.3 נגדיר $H(w)$, גודל המעגל הכי קטן שמייצג את w. אז כמעט לכל $w\in\{0,1\}^n$, $H(w)\geq\frac{2^n}{10n}$.

הגדרה 10.3 הבעיה $succCVAL$ הקלט הוא $x\in\{0,1\}^n, w\in\{0,1\}^n$, כאשר w מעגל עם m קלטים, ומחזירה $\langle tt(w)\rangle(x)$.

הערה: $tt(w)$ זו מחרוזת באורך 2^m שמכילה את כל הערכים של C על כל הקלטים, ואם המחרוזת הזו מייצגת מעגל, $\langle tt(w)\rangle$ זה מעגל חוקי, והייצוג הרבה יותר קצר.

טענה 11.3 $succCVAL\in EXP$.

הוכחה: בהינתן w,x, נחשב את $C=\langle tt(w)\rangle$ ונבדוק האם $(C,x)\in CVAL$.

חישוב $tt(w)$ דורש חישוב המעגל w על 2^n קלטים, $m\leq n$ וכן $|w|$ הוא אורך הקלט ולכן החישוב לוקח זמן אקספוננציאלי, ו$CVAL$ פולינומי בגודל אקספוננציאלי ולכן גם אקספוננציאלי. ∎

טענה 12.3 $succCVAL$ היא EXP-שלמה.

הוכחה: נקח $L\in EXP$, נראה רדוקציה $L\leq_{\varphi} succCVAL$ כך ש$\varphi\in LOG$.

$L\in EXP$, יש מכונה M שרצה בזמן $2^{(n^c)}$ שפותרת אותה. בהינתן מילה x, נסתכל על טבלת החישוב של M על x, היא בגודל $2^{(n^c)}\times 2^{(n^c)}$.

אפשר לייצג את המעגל של טבלת החישוב בעזרת מעגל קטן, כי צריך לשמור את המעגל הקבוע של הפונקציית δ רק פעם אחת, .. הרדוקציה תפלוט את המעגל הקטן, ואת x.

הרדוקציה היא לוגריתמית, כי $|x|=n, |s|, |t|=n^c$ והדבר הכי קשה שφ צריכה לעשות זה לכתוב מעגל שמוסיף או מוריד 1 ממחרוזת באורך n^c, ואת זה אפשר לעשות במקום לוגריתמי. ∎

הערה 13.3 $succCSAT$ היא $NEXP$-שלמה.

4 מכונות עם אורקל וההיררכיה פולינומית

עבור SAT, בהינתן אלגוריתם הכרעה אפשר לכתוב אלגוריתם למציאת השמה מספקת, כך:

בודקים אם $\varphi\in SAT$. אם לא - לא ספיק. אם כן, מציבים $x_1=0$, ובודקים אם הנוסחה שמתקבלת (עם $n-1$ משתנים) היא ספיקה. אם לא, נמשיך עם $x_1=1$, נמצא אינדוקטיבית השמה מספקת לנוסחה החדשה עם $n-1$ משתנים, ונקבל השמה מספקת לנוסחה כולה.

טענה 1.4 האלגוריתם פותר את בעיית החיפוש של SAT.

טענה 2.4 $A\in P^{SAT}$ - רץ בזמן פולינומי עם SAT כאורקל.

הגדרה 3.4 מכונת טיורינג עם אורקל היא מכונת טיורינג רגילה שבנוסף יש לה סרט שאילתות ושלושה מצבים מיוחדים $query, query_0, query_1$, כך שממצב $query$ היא עוברת ל$query_0$ או $query_1$ לפי תשובת האורקל על תוכן סרט השאילתות.

טענה 4.4 $P^O=P^{\bar{O}}$, $P^{SAT}=P^{NP}$.

משפט 5.4 אם T,t פונקציות time-constructible כך ש$T\gg t$, אז לכל אורקל A, $\mathrm{Time}^A(t(n))\subsetneq\mathrm{Time}^A(T(n))$.

הוכחה: אותה הוכחה כמו משפט ההיררכיה לזמן. ■

טענה 6.4 אם $A \subseteq B$ אז $coA \subseteq coB$. כמו כן, $\overline{SAT}$ היא $coNP$ שלמה.

מסקנה 7.4 $Clique = \{(G,k) \mid G \text{ has a clique of size } k\}$ היא NP־שלמה, ו$\overline{Clique}$ היא $coNP$־שלמה.

הגדרה 8.4 נגדיר $ExactClique = \{(G,k) \mid G\text{'s largest clique is of size } k\}$. אז $ExactClique \in P^{Clique}$ (2 קריאות אורקל, עם k ועם $k+1$).

הגדרה 9.4 שפה A היא C־קשה ביחס לרדוקציות טיורינג אם לכל $L \in C$, $L \in P^A$. כלומר, מאפשרים מספר פולינומי של שאילתות אדפטיביות לA, בניגוד לרדוקציה רגילה שמאפשרים רק אחת.

הגדרה 10.4 $L \in DP$ אם יש $L_1 \in NP, L_2 \in coNP$ כך ש$L = L_1 \cap L_2$.

דוגמה 11.4 $SAT, \overline{SAT}, ExactClique \in DP$.

משפט 12.4 $ExactClique$ היא DP־שלמה.

הגדרה 13.4 הבעיה CorrectSATSolver: מעגל C על m קלטים הוא בשפה אם $\varphi \in SAT \iff C(\varphi) = T$.

באיזו מחלקה היא? ב$PSPACE$, אפשר לעבור על כל הנוסחאות, לבדוק אם φ ספיקה ($SAT \in NP \subseteq PSPACE$), להריץ את C, ולהשוות. היא גם ב$\exists P$.

הגדרה 14.4 המחלקה $\Sigma_2 = \exists\forall P$: כל השפות L כך שיש $M \in P$ כך ש$x \in L \iff \exists y \forall z. M(x,y,z) = 1$ וכן $|y|, |z| = poly(|x|)$.

המחלקה $\Pi_2 = \forall\exists P$: כל השפות L כך שיש $M \in P$ כך ש$x \in L \iff \forall y \exists z. M(x,y,z) = 1$ וכן $|y|, |z| = poly(|x|)$.

באופן דומה מסמנים $NP = \exists P, coNP = \forall P$ ($coNP$ כל השפות L כך שיש $M \in P$ כך ש$x \in L \iff \forall y. M(x,y) = 1$).

כמו כן, נגדיר $PH = \bigcup_k \Sigma_k = \bigcup_k \Pi_k$.

משפט 15.4 $\text{CorrectSATSolver} \in \forall P$.

הוכחה: נסמן $C^*(\varphi)$ את המכונה שמשתמשת בC כפותר SAT כדי למצוא השמה מספקת לφ, נטען כי

$$\begin{aligned} C \in \text{CorrectSATSolver} &\iff \forall \varphi \in \{0,1\}^m . C(\varphi) = T \implies \varphi(C^*(\varphi)) = T, C(\varphi) = F \implies \forall b. \varphi(b) = F \\ &\iff \forall \varphi \in \{0,1\}^m \, \forall b. \left(C(\varphi) = F \vee \varphi(C^*(\varphi)) = T\right) \wedge \left(C(\varphi) = T \vee \varphi(b) = F\right) \end{aligned}$$

ואז נקבל $\text{CorrectSATSolver} \in coNP$, כי את הטענה בתוך הלכל אפשר לבדוק בזמן פולינומי.

אם $C \in \text{CorrectSATSolver}$ כלומר לכל φ מתקיים $C(\varphi) = SAT(\varphi)$, אז לכל φ, אם $C(\varphi) = T$ אז יש לφ השמה מספקת ומנכונות $C^*(\varphi)$ תהיה כזו השמה, כלומר $\varphi(C^*(\varphi)) = T$. כמו כן אם $C(\varphi) = F$ אז מנכונות אין השמה מספקת, כלומר לכל b $\varphi(b) = F$.

אחרת, אם $C \notin \text{CorrectSATSolver}$ אז קיימת נוסחה φ לא ספיקה עם $C(\varphi) = T$, או φ ספיקה עם $C(\varphi) = F$.

נניח φ לא ספיקה כך ש$C(\varphi) = T$, אז $C(\varphi) = T$ אבל $\varphi(C^*(\varphi)) = F$ כי אחרת זו השמה מספקת לφ והיא ספיקה, ולכן אגף ימין לא מתקיים.

נניח φ ספיקה כך ש$C(\varphi) = F$, אז מכך שהיא ספיקה יש b כך ש$\varphi(b) = T$, ואגף ימין שוב לא מתקיים. ■

טענה 16.4 $PH \subseteq PSPACE$.

הוכחה: תהי $L \in PH$, אז יש k כך ש$L \in \Sigma_k$, ויש מכונה $M \in P$ כך ש$x \in L \iff \exists x_1 \forall x_2 \cdots \exists x_k M\left(x, x_1, \ldots, x_k\right) =$ T. עושים סוג של minmax רקורסיבי (יש מספר קבוע של קריאות, k) כדי לפתור ב$PSPACE$. ■

הגדרה 17.4 הבעיה OptimalCircuit: הקלט הוא מעגל C על m קלטים, C בשפה אם ורק אם הוא המעגל הכי קטן עם הפונקציונליות הזו. כלומר,

$$\forall B, |B| < |C| \, \exists a . C\left(a\right) \neq B\left(a\right)$$

אז $\text{OptimalCircuit} \in \Pi_2$.

משפט 18.4 OptimalCircuit היא Π_2 שלמה תחת רדוקציות טיורינג (לא בדיוק, אבל בערך).

משפט 19.4 $P^{NP} \subseteq \Sigma_2 = \exists\forall P$. וגם גרסה מוכללת: $NP^{SAT} \subseteq \Sigma_2$ (מופיעה בספר).

הוכחה: תהי $L \in P^{NP}$, נניח $L \in P^{SAT}$. אז יש מכונת אורקל M שרצה בזמן $|x|^c$ כך ש$M^{SAT}\left(x\right) = T \iff x \in L$.
בהינתן קלט x, נבקש מהמוכיח להגיד לנו איזה שאילתות $M^{SAT}\left(x\right)$ הולכת לשאול את האורקל, $\varphi_1, \ldots, \varphi_t \in \{0,1\}^{n^c}$, φ_i השאילתה לשאול בזמן i, וכן איזה תשובות $a_1, \ldots, a_t \in \{0,1\}$ נקבל על השאילתות, וכן לכל i אם $a_i = 1$ נבקש עד w_i שמראה $\varphi\left(w_i\right) = 1$. לכל i כך ש$a_i = 0$, נרצה שלכל b_i יתקיים $\varphi\left(b_i\right) = 0$. אז

$$x \in L \iff M^{SAT}\left(x\right) = T \iff \exists \left\{\varphi_i\right\}_{i=1}^{t}, \left\{a_i\right\}_{i=1}^{t}, \left\{w_i\right\}_{i=1}^{t} \forall \left\{z_i\right\}_{i=1}^{t} . \varphi_i \text{ is the i'th query given } \varphi_{<i}, a_{<i}$$
$$, \text{if } a_i = 1 \text{ then } \varphi_i\left(w_i\right) = T, \text{ if } a_i = 0 \text{ then } \varphi_i\left(z_i\right) = F, M\left(x\right) = T$$

■

מסקנה 20.4 $\Sigma_2 \subseteq NP^{SAT[1]} \subseteq NP^{SAT} \subseteq \Sigma_2$, ולכן הם כולם שווים. באופן דומה עבור $coNP^{SAT}, \Pi_2$.

הגדרה 21.4 השפה $TQBF_2$ כל הנוסחאות מהצורה $\exists y \forall z . \varphi\left(y, z\right)$, כאשר φ נוסחה ב־$\wedge, \vee, \neg$ על המשתנים, שיש להן ערך T.

טענה 22.4 $TQBF_2$ היא Σ_2 שלמה (באופן דומה, $TQBF_{\forall 2}$ היא Π_2 שלמה).

הוכחה: ניקח $L \in \Pi_2$, אז מהגדרה L נפתרת על ידי $\forall y \exists z . M\left(x, y, z\right) \iff x \in L$ כאשר $M \in P, |y|, |z| = poly\left(|x|\right)$.
נסתכל על טבלת החישוב של M, אפשר להמיר אותה למעגל ואז לנוסחה, אבל בנוסחה יהיו הרבה משתנים חדשים (משתנה חדש לכל שער בטבלת החישוב). אבל החישוב דטרמיניסטי בהינתן הקלט, ולכן יש רק בחירה אחת לערכי המשתנים שהיא קונסיסטנטית עם החישוב. לכן, הרדוקציה תחזיר את הנוסחה

$$\forall y \exists z, \text{values for new variables} . \varphi\left(\cdots\right)$$

זה אפשרי גם עבור Σ_2, כי $\overline{TQBF_2} = TQBF_{\forall,2}$ היא $\Pi_2 = co\Sigma_2$ שלמה ולכן $TQBF_2$ היא Σ_2 שלמה. ■

טענה 23.4 אם $NP = coNP$ אז $PH = NP = coNP = NP \cap coNP$.

הוכחה: נראה $\Sigma_2 = NP$. צריך להראות $\Sigma_2 \subseteq NP$, ניקח $L \in \Sigma_2$ אז מהגדרה $\exists y \forall z . M\left(x, y, z\right) \iff x \in L$.
נגדיר שפה L' כך ש$\forall z . M\left(x, y, z\right) \iff \left(x, y\right) \in L'$, אז $L' \in \Pi_1 = coNP$, מהנחה $L' \in NP$ ולכן ניתן לכתוב $\left(x, y\right) \in L' \iff \exists w . M'\left(x, y, w\right)$.
אז $\exists y, w . M'\left(x, y, w\right) \iff \exists y . \left(x, y\right) \in L' \iff \exists y \forall z . M\left(x, y, z\right) \iff x \in L$, כלומר $L \in NP$. ■

טענה 24.4 אם לPH יש שפה שלמה אז ההירכיה קורסת.

הוכחה: נניח L היא PH שלמה, בפרט $L \in PH = \bigcup_{k=1}^{\infty} \Sigma_k$ ולכן יש k כך ש$L \in \Sigma_k$. כעת תהי $L' \in PH$, אז $L' \leq L$, ולכן גם $L' \in \Sigma_k$. כלומר קיבלנו $PH = \Sigma_k$. ■

טענה 25.4 $coP^{NP} = P^{NP}$.

הגדרה 26.4 השפה $TQBF$ כל הנוסחאות מהצורה

$$\exists x_1 \in \{0,1\} \forall x_2 \in \{0,1\} \cdots \exists x_m \in \{0,1\} . \varphi (x_1, \ldots, x_m)$$

שהערך שלהן הוא T.

טענה 27.4 $TQBF$ היא $PSPACE$־שלמה.

הוכחה: ראשית נראה $TQBF \in PSPACE$. נניח שקיבלנו קלט $\varphi = \exists x_1 \forall x_2 \cdots \exists x_m \varphi (x_1, \ldots, x_m)$.

נבנה בראש עץ בעומק m, כאשר ברמה הראשונה מחליטים את הערך של x_1 וממנו מתפצלים, ואז x_2, ואז x_3 וכן הלאה עד x_m. רוצים לראות אם יש מסלול שמקיים את התנאים (זה סוג של min-max), וזה יהיה הvalue של העץ (האם יש אסטרטגיה מנצחת ־ קיים מהלך, כך שלכל מהלך של היריב, יש מהלך טוב).

אפשר לחשב את ערך העץ (בלי לבנות אותו, כי הוא בגודל אקספוננציאלי) רקורסיבית במקום פולינומי, על ידי סריקה שלו לעומק. זה מראה ש$TQBF \in PSPACE$.

כעת נראה $TQBF$ היא $PSPACE$ שלמה. תהי $L \in PSPACE$, אז היא נפתרת על ידי מכונת טיורינג M עם זכרון פולינומי, נניח n^c.

צריך לבנות רדוקציה φ שמקבלת $x \in \{0,1\}^n$, וצריכה להוציא $\varphi(x)$ פסוק $TQBF$ כך ש$x \in L \iff \varphi(x) \in TQBF$.

בנייה: נסתכל על גרף הקונפיגורציות של המכונה M (שפותרת את L) ־ יש $2^{n^c} \cdot n^c \cdot n \cdot O(1) \leq 2^{2n^c}$ קודקודים (קונפיגורציות), ויש קשת מכוונת בין שני קודקודים אם קונפיגורציה אחת מובילה לשנייה. דרגת היציאה של כל קודקוד חוקי היא 1, כי M דטרמיניסטית. מתקיים $x \in L$ אם ורק אם יש מסלול בG מc_{init} לc_{accept}, וננסה לנסח את זה כ$TQBF$.

נגדיר $Reach(c, c', t)$ שמקבל ערך T אם ורק אם יש מסלול מc לc' בG באורך $\geq 2^t$. מתקיים $Reach(c, c', 0)$ בP (בודקים אם אפשר לעבור בין הקונפיגורציות בצעד אחד, ישירות), אנחנו מתעניינים ב$Reach(c_{init}, c_{accept}, 2n^c)$ (כי יש 2^{2n^c} קודקודים). מתקיים

$$\begin{aligned} Reach(c_1, c_2, t) &= \exists c_{mid} . Reach(c, c_{mid}, t-1) \wedge Reach(c_{mid}, c_2, t-1) \\ &= \exists c_{mid} \forall b \in \{0,1\} \exists w_1, w_2 \in V. (b = 0 \implies w_1 = c_1, w_2 = c_{mid}) \\ &, (b = 1 \implies w_1 = c_{mid}, w_2 = c_2), Reach(w_1, w_2, t-1) \end{aligned}$$

נמשיך רקורסיבית n^c רמות ונקבל $TQBF$ ־ זה יהיה הפלט של הרדוקציה, φ_x. היא אכן לוגריתמית, הוכחת נכונות הייתה בכיתה, באינדוקציה. ■

משפט 28.4 קרפ־ליפטון ראינו כבר שאם $NP = P$ אז $PH = P$, וכמו כן אם $NP \subseteq \text{Size}(poly)$ אז $PH \subseteq \text{Size}(poly)$.

משפט קרפ־ליפטון טוען שאם $NP \subseteq \text{Size}(poly)$ אז $PH = \Sigma_2$.

הוכחה: נניח $NP \subseteq \text{Size}(poly)$, ונראה ש$\Pi_2 \subseteq \Sigma_2$. זה יסיים, כי אז גם $\Sigma_2 = co\Pi_2 \subseteq co\Sigma_2 = \Pi_2$ ולכן $\Sigma_2 = \Pi_2$ וההיררכיה קורסת.

תהי $L \in \Pi_2$, אז $x \in L \iff \forall y \exists z M(x, y, z)$.

נראה רדוקציה מ$TQBF_{\forall,2}$ לΣ_2. הקלט הוא $\forall y\exists z\varphi(x,y,z)$, נסתכל על הנוסחה

$$\exists\text{circuit } C.\underbrace{C\in CorrectSATSolver}_{\in\forall P}\wedge\forall y\left[C\left(\underbrace{\varphi(x,y,z)}_{z\text{ is the variable}}\right)=T\right]\in\Sigma_2$$

■

5 זכרון לא דטרמיניסטי

הגדרה 1.5 זכרון דטרמיניסטי $L\in DSPACE(s(n))$ אם היא נפתרת על ידי מכונת טיורינג דטרמיניסטית עם סרט קלט, עבודה ופלט ומשתמשת בלכל היותר $s(n)$ זכרון בסרט העבודה.

יש שתי הגדרות אפשריות לחישוב לא דטרמיניסטי:

1. פונקציית מעברים $\delta\colon\Sigma\times Q\to P(\Sigma\times Q\times\{L,R\})$.
2. דוחפים את כל כמתי הקיים להתחלה. לדוגמה, בזמן

$$NTIME(t)=\{L\in\Sigma^*\mid\exists M\in P.x\in L\iff\exists y.M(x,y),M(x,y)\text{ runs in }t(|x|)\text{ time}\}$$

נרצה להגדיר גם זכרון לא דטרמיניסטי.

הגדרה 2.5 זכרון לא דטרמיניסטי $L\in NSPACE(s(n))$ אם יש מכונה עם פונקציית מעברים $\delta\colon\Sigma\times Q\to P(\Sigma\times Q\times\{L,R\})$ עם סרט קלט, עבודה ופלט שמשתמשת בלכל היותר $s(n)$ זכרון בסרט העבודה.
באופן שקול, $L\in NPSPACE(s(n))$ אם היא נפתרת על ידי $M(x,y)$ עם סרט קלט, פלט, עבודה וסרט ניחושים, כאשר כל הסרטים כרגיל וסרט הניחושים הוא read-once ־ לקריאה בלבד, ובכל קריאה זזים ימינה. ואז אפשר לדחוף את כמתי הקיים על סרט הניחושים להתחלה.

הערה 3.5 המכונה תמיד חייבת לעצור (לכל אפשרות של ניחושים), וצריך להיות סדרת ניחושים אחת שתקבל כדי שמילה תתקבל.

הגדרה 4.5 השפה STCON הקלט הוא גרף מכוון $G=(V,E)$ ושני קודקודים $s,t\in V$, והוא בשפה אם יש מסלול בין s לt בG.

אלגוריתם לא דטרמיניסטי בזכרון לוגריתמי לSTCON אנחנו יודעים $STCON\in P$, על ידי לדוגמה BFS. נציג אלגוריתם לא דטרמיניסטי:

1. נגדיר $curr\leftarrow s$.
2. לכל $i=1,2,\ldots,|V|$, ננחש שכן v' של $curr$, אם $curr=t$ נסיים ונקבל, אחרת נמשיך.
3. אם סיימנו את הלולאה בלי להגיע לt, נדחה.

סיבוכיות הזכרון היא לוגריתמית, כי צריך לשמור את $curr,s,v'$. לכן, $STCON\in NSPACE(O(\log(n)))=NL$.

משפט 5.5 $STCON$ היא NL שלמה.

הוכחה: ראינו $STCON\in NL$. תהי $L\in NL$, ותהי M מכונה לא דטרמיניסטית שפותרת אותה. נבנה רדוקציה $L\leq_\varphi STCON$.
על קלט x, הרדוקציה תוציא את $G=(V,E)$ גרף הקונפיגורציות של M, את s הקונפיגורציה ההתחלתית על x וt הקונפיגורציה המקבלת. ■

אלגוריתם דטרמיניסטי ב$\log^2 n$ זכרון לSTCON נייצג את הגרף כמטריצת שכנויות A בגודל $|V| \times |V|$. נגדיר כפל מטריצות בוליאני עם $\vee$ כחיבור ו$\wedge$ ככפל, אז מתקיים

$$A^k[i,j] = 1 \iff \exists \text{path of length } k \text{ from } i \text{ to } j \text{ in } G$$

רוצים לדעת האם $A^n[s,t] = 1$. האלגוריתם יקבל את A, s, t, יחשב את $A^2, A^4, \ldots, A^{\left(2^{\log n}\right)}$ ויבדוק את $A^n[s,t]$. אבל לאחסן עוד מטריצות שכנויות לוקח יותר מדי זכרון, אז נעשה הרכבה של רדוקציות space. מפעילים $\log n$ רדוקציות וכל אחת לוקחת $\log n$ זכרון, לכן בסך הכל נצטרך $\log^2 n$ זכרון (כמו stack).

משפט 6.5 סאביץ' $STCON \in DSPACE\left(O\left(\log^2 n\right)\right)$.

הוכחה: נתון גרף G, $s,t \in V$. נבנה אלגוריתם $Reach(a,b,\ell)$, שיהיה T אם ורק אם יש מסלול מa לb באורך לכל היותר 2^ℓ.

הבסיס יהיה $\ell = 0$: עבור $Reach(a,b,0)$ נעבור על כל הקשתות ונבדוק אם יש קשת בין a לb, או אם $a = b$.

עבור $\ell > 0$, נפתור רקורסיבית $Reach(a,b,\ell) = \vee_{k=1}^{n} Reach(a,k,\ell-1) \wedge Reach(k,b,\ell-1)$.

סיבוכיות: נסמן ב$s(\ell)$ את סיבוכיות הזכרון של $Reach(a,b,\ell)$. אז $s(0) = O(\log n)$, ומתקיים $s(\ell+1) = s(\ell) + O(\log n)$ כי עוברים על כל k ($\log n$ זכרון), ובתוכו מחשבים כל פעם את $Reach$ עם $\ell - 1$, שזה לוקח $s(\ell)$ זכרון (רק פעם אחת ולא פעמיים, כי בין ה$\wedge$ אפשר לשחרר את הזכרון). נפתור את הנוסחה ונקבל $s(\ell) = \ell \log n$, ולכן $s(\log n) = \log n \log n = \log^2 n$. ∎

הערה 7.5 האלגוריתם של סאביץ' לוקח $n^{O(\log n)}$ זמן - הרבה. לא ידוע אם יש אלגוריתם בזמן פולינומי וזכרון $\sqrt{n}$ לSTCON, אבל בגרפים לא מכוונים ידוע שיש אלגוריתם ל$USTCON$ בזכרון $O(\log n)$, ולכן גם זמן פולינומי (ריינגולד).

טענה 8.5 כל אלגוריתם $NSPACE(s(n))$ רץ בזמן לכל היותר $2^{O(s)}$.

הוכחה: זמן הריצה הוא לכל היותר גודל גרף הקונפיגורציות (כי אחרת יש לולאה אינסופית), וגודל גרף הקונפיגורציות הוא לכל היותר $2^{O(s)}$ (כי כל קונפיגורציה בגודל $s \cdot \log s \cdot \log n$). ∎

מסקנה 9.5 קשר בין מחלקות $L \subseteq NL$ וכן $NL \subseteq P$, $NL \subseteq DSPACE\left(O\left(\log^2 n\right)\right)$.

הגדרה 10.5 המחלקות ZNP, ZNL מתקיים $L \in ZNP$ אם יש מכונה לא דטרמיניסטית עם זמן פולינומי M שיכולה לענות $accept, reject, quit$, אם $x \in L$ אז יש מסלול שעונה $accept$ ואין מסלול שעונה $reject$, ואם $x \notin L$ אז יש מסלול שעונה $reject$ ואין מסלול שעונה $accept$. באופן דומה מגדירים ZNL (M לא דטרמיניסטית עם זכרון לוגריתמי).

טענה 11.5 $ZNP = NP \cap coNP, ZNL = NL \cap coNL$.

הוכחה: נניח $L \in ZNP$. אז $L \in NP$, כי נבנה מכונה M' שמסמלצת את M, אם מחזירה $accept$ מקבלת, ואם מחזירה $reject$ או $quit$ דוחה. באופן דומה $\overline{L} \in NP$, בונים מכונה M' שמסמלצת את M, אם מחזירה $reject$ מקבלת, ואם מחזירה $accept$ או $quit$ דוחה.

נניח $L \in NP \cap coNP$, יהיו M_1, M_2 מכונות לא דטרמיניסטיות שפותרות את $L, \overline{L}$ בהתאמה. נגדיר מכונה M, שמסמלצת את M_1, M_2, אם רק M_1 מקבלת היא מחזירה $accept$, אם רק M_2 מקבלת היא מחזירה $reject$, אם שתיהן דוחות היא מחזירה $quit$ (לא יכול להיות ששתיהן מקבלות). ∎

משפט 12.5 $STCON \in ZNL$. מכאן נובע גם $ZNL = NL = coNL$.

הוכחה: נגדיר פרוצדורה לא דטרמיניסטית $Enumerate\left(k,C_k\right)$, שבהינתן שמובטח לנו שיש בדיוק C_k קודקודים שנגישים מs בלכל היותר k צעדים, היא תפלוט את C_k הקודקודים לסרט הפלט בסדר לקסיקוגרפי עולה, והיא תעשה זאת במובן ZNL (יש לפחות ניחוש אחד שמוביל לתוצאה נכונה וaccept, אין ניחושים שמובילים לתוצאה לא נכונה וaccept, ואפשר להחזיר quit).

הפרוצדורה תפעל באופן הבא: נגדיר $v_{curr}=-\infty$, ולכל $i=1,2,\ldots,C_k$ ננחש $v>v_{curr}$, ננחש מסלול באורך אורך $\geq k$ בין s לv, אם המסלול לא חוקי נחזיר quit, אחרת נגדיר $v_{curr}=v$ ונפלוט את v. בסוף נחזיר accept.

נכונות הפרוצדורה: תמיד יש מסלול שאינו quit, כי C_k באמת מספר השכנים במרחק $\geq k$, אז יש $v_1,\ldots,v_{C_k}$ חוקיים, נסדר אותם בסדר לקסיקוגרפי אז סדרת הניחושים הזו עם המסלולים המתאימים תחזיר accept. אם מסלול מחזיר accept אז התוצאה נכונה, כי פולטים בדיוק C_k קודקודים שונים, ויש מסלול מכולם לv, כי בדקנו את נכונות המסלול שניחשנו.

נגדיר פרוצדורה לא דטרמיניסטית $Cnext\left(k,C_k\right)$, שבהינתן שC_k נכון, היא מוציאה את C_{k+1}.

הפרוצדורה תפעל באופן הבא: נגדיר count=0, נעבור על כל $w\in V$, נעבור על כל צומת v ב$Enumerate\left(k,C_k\right)$ (אם מחזיר quit, גם אנחנו נחזיר quit), אם w שכן של v נגדיל את count ב1 ונמשיך לw הבא, ובסוף נוציא את count.

לכן, כדי לפתור את $STCON$, נתחיל ב$C_0=1$, נריץ את $Cnext\left(k,C_k\right)$ על $k=0,\ldots,|V|-1$ ונקבל בסוף את C_n. נריץ $Enumerate\left(n,C_n\right)$, ונבדוק אם t ברשימה. ∎

6 אלגוריתמים הסתברותיים

הגדרה 1.6 שפה $L\in BPP_{s,c}$ אם קיימת מכונת טיורינג $M\in P$ שמקבלת x,y כאשר $|y|=poly\left(|x|\right)$ (y המטבעות שהאלגוריתם מטיל), ומקיימת שאם $x\in L$ אז $Pr_y\left[M\left(x,y\right)=1\right]\geq c$ ואם $x\notin L$ אז $Pr_y\left[M\left(x,y\right)=1\right]\leq s$.

שפה $L\in RP_{\varepsilon}$ אם קיימת מכונת טיורינג $M\in P$ שמקבלת x,y כאשר $|y|=poly\left(|x|\right)$ (y המטבעות שהאלגוריתם מטיל), ומקיימת שאם $x\in L$ אז $Pr_y\left[M\left(x,y\right)=1\right]\geq 1-\varepsilon$ ואם $x\notin L$ אז $Pr_y\left[M\left(x,y\right)=1\right]=0$.

כמו כן, $coRP=BPP_{\frac{1}{2},1}, RP=BPP_{0,\frac{1}{2}}, BPP=BPP_{\frac{1}{3},\frac{2}{3}}$, ונשים לב ש$P=BPP_{0,1}, NP=BPP_{0,>0}$. כלומר RP היא כמו NP אבל כשיש המון עדים, ולכן במקום לבקש ממוכיח אפשר להגריל ולבדוק, באופן פיזיבילי.

טענה 2.6 $RP_{\frac{1}{2}}=RP_{2^{-t}}$ (t יכול להיות פולינומי בגודל הקלט).

הוכחה: נניח $L\in RP_{\frac{1}{2}}$, אז יש $M\left(x,y\right)$ כך ש$m=|y|=poly\left(|x|\right)$ כך שאם $x\in L$ אז $Pr_y\left[M\left(x,y\right)=1\right]\geq\frac{1}{2}$, ואם $x\notin L$ אז $Pr_y\left[M\left(x,y\right)=1\right]=0$.

נבנה מכונה חדשה $M'\left(x,y'\right)$ שתטיל $y_1,\ldots,y_t$ כאשר $y_i\in\{0,1\}^m$, תריץ $M\left(x,y_1\right),\ldots,M\left(x,y_t\right)$ ותענה כן אם אחד הנסיונות ענה כן.

נכונות: אם $x\notin L$ אז לכל $y'=y_1,\ldots,y_t$ לכל i מתקיים $M\left(x,y_i\right)=0$ ולכן תמיד נענה לא. אם $x\in L$ אז יש n נסיונות בלתי תלויים, בכל נסיון מצליחים בהסתברות $\leq\frac{1}{2}$ אז ההסתברות שנכשל בכולם היא $\frac{1}{2^t}$, כלומר ההסתברות שנצליח היא $\leq 1-\frac{1}{2^t}$. ∎

טענה 3.6 $P\subseteq RP\subseteq NP$, וכן $RP\subseteq BPP$. לא יודעים מה הקשר בין BPP,NP, כן יודעים $BPP\subseteq NEXP$ אבל לא יודעים אם ההכלה היא שוויון.

הוכחה: $RP=BPP_{0,\frac{1}{2}}=BPP_{0,\frac{3}{4}}\subseteq BPP_{\frac{1}{3},\frac{2}{3}}$. ∎

משפט 4.6 $BPP_{\frac{1}{3},\frac{2}{3}}=BPP_{2^{-n},1-2^{-n}}$ (בהמשך).

אלגוריתם לבדיקת שוויון במינימום תקשורת לאליס ולבוב יש שני מספרים בני n ביטים, הם רוצים לבדוק אם הם שווים במינימום תקשורת ביניהם. באופן נאיבי צריך n ביטים, אבל אם אליס בוחרת ראשוני אקראי p בין 1 ל־n^4, שולחת את $p, x \mod p$ ובוב משווה אז זו תקשורת של $O(\log n)$ והסתברות 1 לצדוק אם יש שוויון, ו־$\frac{1}{n^2}$ לטעות אם אין שוויון.

ראינו גם דוגמה לקארפ רבין, השוואה של תת מחרוזת.

אלגוריתם לזיווג מושלם בגרף דו צדדי אפשר לפתור עם אלגוריתם זרימה ב־P. נרצה למצוא אלגוריתם אחר. נבנה מטריצה $n \times n$ שמייצגת את הגרף, אם $(i,j) \in E$ אז $M[i,j] = 0$, ואם $(i,j) \notin E$ אז $M[i,j] = x_{ij}$. לכל x_{ij} נבחר ערך a_{ij} באקראי בין $1,\ldots,T = 2n$, מקבלים מטריצה של מספרים A ומחשבים $\det A$. אם $\det A = 0$ נאמר שאין זיווג, אחרת יש.

נטען שאם אין זיווג מושלם תמיד נגיד לא. אכן, מתקיים $\det M = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \cdot M_{1,\sigma(1)} \cdots M_{n,\sigma(n)}$, ואם אין זיווג מושלם אז לכל σ אחת הקשתות $(1,\pi(1)), \ldots, (n,\pi(n))$ חסרה ולכן אחד הערכים $M_{i,\sigma(i)}$ יהיה 0, והמכפלה תהיה 0, ולכן גם הסכום.

אם יש זיווג מושלם בגרף, נניח π, אז $M_{i,\pi(i)} = a_{i,\pi(i)}$ עבור $1 \le i \le n$ כולם לא 0 ולכן המכפלה שלהם לא 0, אבל בעיה פוטנציאלית היא שאולי יש כמה זיווגים והם יתבטלו עם הסימן. נטען שנחזיר כן בהסתברות $\le \frac{1}{2}$. אכן, הדטרמיננטה פולינום מדרגה n ב־n^2 משתנים, וממשפט שוורץ־זיפל (אם $0 \ne p \in F[x_1,\ldots,x_m]$ אז לכל $A \subseteq F$, מתקיים $Pr_{a_1,\ldots,a_m \in A}[p(a_1,\ldots,a_m) = 0] \le \frac{\deg p}{|A|}$) יש הסתברות של לכל היותר $\frac{n}{T} = \frac{n}{2n} = \frac{1}{2}$ שהדטרמיננטה תצא 0, כלומר הסתברות לכל היותר חצי שנטעה, כדרוש.

היתרון של האלגוריתם הוא שאפשר למקבל אותו: יודעים לחשב $\det M_{n \times n}$ עם $poly(n)$ מעבדים בזמן $O(\log^2 n)$.

הגדרה 5.6 התפלגות נורמלית עם תוחלת μ וסטיית תקן σ היא $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$.

דוגמה 6.6 אם מטילים 1000 מטבעות, ההסתברות שיצאו 500 פעמים 1 היא $\binom{1000}{500} 2^{-1000}$, אפשר לקרב עם סטירלינג.

משפט 7.6 צ'רנוף אם $X_1,\ldots,X_n \sim \operatorname{Ber}(p)$ בלתי תלויים, אז

$$Pr\left[\left|\sum_{i=1}^n X_i - np\right| > \varepsilon np\right] \le e^{-\Omega\left(\varepsilon^2 np\right)}$$

כלומר,

$$Pr\left[\left|\sum_{i=1}^n X_i - np\right| > t \underbrace{\sqrt{np(1-p)}}_{=\operatorname{std}\left(\sum_{i=1}^n X_i\right)}\right] \le e^{-\Omega\left(t^2\right)}$$

הוכחה: נחשב

$$Pr(X \ge a) = Pr\left(t^X \ge t^A\right) \le \frac{\mathbb{E}\left[t^X\right]}{t^A}$$

$$\mathbb{E}\left[t^X\right] = \mathbb{E}\left[t^{\sum X_i}\right] = \mathbb{E}\left[\prod t^{X_i}\right] \overset{X_i \text{ are independent}}{=} \prod_{i=1}^n \mathbb{E}\left[t^{X_i}\right] = \prod_{i=1}^n \left((1-p)t^0 + p \cdot t\right) = (1 + p(t-1))^n$$

$$\Longrightarrow Pr(X \ge a) \le \frac{\mathbb{E}\left[t^X\right]}{t^A} = \frac{(1+p(t-1))^n}{t^A} \overset{1+x \le e^x}{\le} \frac{e^{pn(t-1)}}{t^A}$$

נבחר $t = \frac{A}{\mu}$, נציב ונקבל ... ■

משפט 8.6 מרקוב לכל משתנה מקרי $X \geq 0$ מתקיים $P\left(X \geq A\right) \leq \frac{\mathbb{E}[X]}{A}$.

משפט 9.6 צ'בישב אם $X_1, \ldots, X_n$ ב"ת בזוגות נסמן $\mu = \mathbb{E}\left[\sum_{i=1}^{n} X_i\right]$ אז $Pr\left(\left|\sum_{i=1}^{n} X_i - \mu\right| \geq A\right) \leq \frac{\mathrm{Var}\left(\sum_{i=1}^{n} X_i\right)}{A^2}$.

טענה 10.6 אמפליפיקציה לכל $a > 0$, לכל $b \in \mathbb{N}$ ולכל $p \in (0,1)$ מתקיים (כאשר n אורך הקלט)

$$BPP_{\left(p, p+\frac{1}{n^a}\right)} = BPP_{\left(2^{-n^b}, 1-2^{-n^b}\right)}$$

הוכחה: נניח $L \in BPP_{\left(p, p+\frac{1}{n^a}\right)}$, אז היא נפתרת ע"י $M\left(x,y\right)$.

בנייה: נבנה מכונה חדשה M', שבהינתן $x \in \{0,1\}^n$ נטיל $y_1, \ldots, y_T \in \{0,1\}^m$, נריץ $M\left(x, y_i\right)$ עבור $i = 1, \ldots, T$ ואם $\left(p + \frac{1}{2n^a}\right) T$ ענו כן נענה כן, אחרת נענה לא. נקבע את T בהמשך.

נכונות: אם $x \in L$, יש T נסיונות בלתי תלויים, נסמן X_i הסיכוי שהניסוי הi יהיה 1 אז $X_1, \ldots, X_T$ ב"ת ומתקיים $\mathbb{E}\left[X_i\right] \geq p + \frac{1}{n^a}$ אז

$$Pr\left(M' \text{ is wrong}\right) = Pr\left(\sum_{I=1}^{T} X_i < \left(p + \frac{1}{2n^a}\right) T\right) \leq Pr\left(\left|X - pT\right| > \frac{1}{2n^a} T\right)$$
$$\overset{\text{Chernoff}}{\leq} e^{-\Omega\left(\frac{T}{n^{2a}}\right)} \leq 2^{-n^b}$$

אם בוחרים $T = \Omega\left(n^{2a+b}\right)$. ■

משפט 11.6 אדלמן $BPP \subseteq \mathrm{Size}\left(poly\right)$. מכאן גם אם $NP \subseteq BPP$ אז ההיררכיה קורסת (כי $SAT \in \mathrm{Size}\left(poly\right)$).

הוכחה: תהי $L \in BPP = BPP_{\left[2^{-2n}, 1-2^{-2n}\right]}$, שנפתרת על ידי $M\left(x,y\right)$ פולינומית עם $|y| = poly\left(|x|\right)$ כך שלכל $x \in \{0,1\}^n$,

$$Pr_y\left[M\left(x,y\right) \text{ is wrong}\right] \leq 2^{-2n}$$

נטען שיש $y_0 \in \{0,1\}^m$ כך שלכל $x \in \{0,1\}^n$ מתקיים $M\left(x, y_0\right) = L\left(x\right)$ (כלומר $M\left(x, y_0\right)$ צודקת).
אכן, לכל x מתקיים $Pr_y\left[M\left(x,y\right) \text{ is wrong}\right] \leq 2^{-2n}$ ולכן

$$Pr_y\left[\exists x . M\left(x,y\right) \text{ is wrong}\right] \overset{\text{union bound}}{\leq} \sum_{x \in \{0,1\}^n} Pr_y\left[M\left(x,y\right) \text{ is wrong}\right] \leq 2^n \cdot 2^{-2n} = 2^{-n}$$

ולכן $Pr_y\left[\forall x . M\left(x,y\right) = L\left(x\right)\right] \geq 1 - 2^{-n}$, בפרט קיים y כזה, ומשפחת המעגלים תהיה $M\left(x,y\right)$ עם y הנ"ל hardwired. ■

משפט 12.6 סיפסר $BPP \subseteq \Sigma_2$ (ומסגירות למשלים של BPP גם $BPP \subseteq \Pi_2$).

הוכחה: תהי $L \in BPP = BPP_{\left[\frac{1}{4m}, \frac{1}{2}\right]}$, כלומר L נפתרת ע"י מ"ט $M\left(x,y\right)$ עם $|x| = n, |y| = m$ כך ש

$$x \in L \implies Pr_y\left[M\left(x,y\right) = 1\right] \geq \frac{1}{2}$$
$$x \notin L \implies Pr_y\left[M\left(x,y\right) = 1\right] \leq \frac{1}{4m}$$

נקבע $x \in \{0,1\}^m$. נסתכל על עולם המטבעות $\{0,1\}^m$, ונסמן $Y_x = \{y \in \{0,1\}^m \mid M\left(x,y\right) = 1\}$. אזי אם $x \notin L$ אז $|Y_x| \leq \frac{2^m}{4m}$, ואם $x \in L$ אז $|Y_x| \geq 2^{m-1}$.

בנייה: הרודקציה φ מקבלת x ובונה פסוק Σ_2 (כאשר חיבור הוא מודולו 2, כלומר $\oplus$)

$$\exists \{s_i\}_{i=1}^{2m} \subseteq \{0,1\}^m .\forall w \in \{0,1\}^m .M(x, w+s_1) = 1 \vee \cdots \vee M(x, w+s_{2m}) = 1$$

נכונות: נשים לב שמתקיים $w \in Y_x + s_i \iff w + s_i \in Y_x \iff M(x, w+s_i) = 1$, ולכן הביטוי בתוך הפסוק שקול ל$w \in \bigcup_{i=1}^{2m} Y_x + s_i$

אם $x \notin L$, אז לכל $s \in \{0,1\}^m$ מתקיים $|Y_x + s| = |Y_x| \le \frac{2^m}{4m}$ לכן לכל $\{s_i\}_{i=1}^{2m} \subseteq \{0,1\}^m$ מתקיים $\left|\bigcup_{i=1}^{2m} Y_x + s_i\right| \le 2m \cdot \frac{2^m}{4m} < 2^m$, כלומר יש $w \in \{0,1\}^m$ כך שהביטוי בתוך הפסוק הוא F.

אם $x \in L$, נוכיח באמצעות השיטה ההסתברותית. לכל $w \in \{0,1\}^m$ מתקיים

$$\begin{aligned} Pr_{\{s_i\}_{i=1}^{2m} \leftarrow \{0,1\}^m}\left[w \notin \bigcup_{i=1}^{2m} Y_x + s_i\right] &= Pr_{\{s_i\}_{i=1}^{2m} \leftarrow \{0,1\}^m}\left[\forall 1 \le i \le 2m . w \notin Y_x + s_i\right] \\ &= \prod_{i=1}^{2m} Pr_{s \leftarrow \{0,1\}^m}\left[w \notin Y_x + s\right] \le \left(\frac{1}{2}\right)^{2m} = \frac{1}{2^{2m}} \end{aligned}$$

ולכן

$$Pr_{\{s_i\}_{i=1}^{2m} \leftarrow \{0,1\}^m}\left[\exists w . w \notin \bigcup_{i=1}^{2m} Y_x + s_i\right] \le \sum_{w \in \{0,1\}^m} 2^{-2m} = 2^m 2^{-2m} = 2^{-m} < 1$$

ובפרט קיים $\{s_i\}_{i=1}^{2m} \subseteq \{0,1\}^m$ כך שלכל $w \in \{0,1\}^m$ מתקיים $w \in \bigcup_{i=1}^{2m} Y_x + s_i$, כלומר הפסוק הוא T. ■

7 הוכחות אינטראקטיביות

נרצה לדון במה קורה כשמשלבים מוכיח כל יכול עם בודק הסתברותי.

הגדרה 1.7 תהי C מחלקה, נגדיר את $\$C$ להיות מחלקת השפות L כך שיש $M(x,y)$ בC עם $|y| = poly(|x|)$ כך ש

$$x \in L \implies Pr_y[M(x,y)=1] \geq \frac{2}{3}$$
$$x \notin L \implies Pr_y[M(x,y)=1] \leq \frac{1}{3}$$

בפרט, $BPP = \$P$.

הגדרה 2.7 איזומורפיזם בין גרפים $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ איזומורפיים אם $|V_1| = |V_2|$ ויש תמורה $\pi : V_1 \to V_2$ כך ש$(a,b) \in E_1 \iff (\pi(a), \pi(b)) \in E_2$.

השפה $GISO$ היא שפת כל הזוגות (G_1, G_2) כך ש$G_1 \sim G_2$. אז $GISO \in NP$.

פרוטוקול ל$\overline{GISO}$ נציג את הפרוטוקול הבא: בהינתן קלט משותף למוכיח ולבודק G_1, G_2 גרפים על n קודקודים,

1. הבודק בוחר $b \leftarrow \{1,2\}, \pi \leftarrow S_n$ אקראיים.
2. הבודק שולח למוכיח את הגרף $G_{new} = \pi(G_b)$, כלומר גרף על n קודקודים כך שיש קשת בין (i,j) אם $(\pi(i), \pi(j)) \in E_b$.
3. המוכיח שולח לבודק $c \in \{1,2\}$, לפי האם הוא חושב ש$G_1 \sim G_{new}$ או $G_{new} \sim G_2$.
4. הבודק יבדוק האם $b = c$. אם כן יחזיר כן, אחרת לא.

נכונות: אם $G_1 \nsim G_2$, כלומר הטענה נכונה, אז יש מוכיח הוגן שבהינתן G_{new} הוא יבדוק אם $G_1 \sim G_{new}$ או $G_2 \sim G_{new}$ ויחזיר את הביט המתאים. יודעים ש$G_b \sim G_{new}$ ולכן הוא איזומורפי לאחד מהם, והוא לא איזומורפי לשני כי אז מטרנזיטיביות $G_1 \sim G_2$ סתירה. כלומר המוכיח ישלח לבודק את התשובה הנכונה b, והוא יחזיר כן.

להיפך, אם $G_1 \sim G_2$ כלומר הטענה שגויה, G_{new} יהיה גרף אקראי שאיזומורפי גם לG_1 וגם לG_2 ללא תלות בb, ולכן התשובה שהמוכיח ישלח תהיה בלתי תלויה בb, כלומר ההסתברות שהיא תהיה שווה לb היא בדיוק $\frac{1}{2}$.

כלומר, קיבלנו שאם $(G_1, G_2) \notin \overline{GISO}$ אז לכל מוכיח הבודק ידחה בהסתברות $\frac{1}{2}$ (soundness), ואם $(G_1, G_2) \in \overline{GISO}$ אז יש מוכיח הוגן עבורו הבודק תמיד יקבל (completeness).

הערה 3.7 השתמשנו פה בכך שהבודק יכול לשמור סודות (בפרט המטבעות שלו), נגדיר מחלקה בלי סודות (private coins) אבל נראה בהמשך שהיא שקולה למחלקה עם סודות (public coins).

הגדרה 4.7 ארתור מרלין המחלקה AM היא מחלקת כל השפות L עבורן יש מכונה $M(x,y,w)$ (x הקלט, y מטבעות, w הוכחה) כך ש$|y|, |w| = poly(|x|)$, M פולינומית וכן

$$x \in L \implies Pr_y[\exists w. M(x,y,w)=1] \geq \frac{2}{3} \text{ (completeness)}$$
$$x \notin L \implies Pr_y[\exists w. M(x,y,w)=1] \leq \frac{1}{3} \text{ (soundness)}$$

חשוב לציין שהמטבעות פומביים, כלומר אין סודות ובפרט w יכול להיות פונקציה של y.

מרלין ארתור המחלקה MA היא מחלקת כל השפות L עבורן יש מכונה $M\left(x,y,w\right)$ כך ש$\left|y\right|,\left|w\right|=poly\left(\left|x\right|\right)$, M פולינומית וכן

$$x\in L\implies\exists w.Pr_{y}\left[M\left(x,y,w\right)=1\right]\geq\frac{2}{3}\text{ (completeness)}$$
$$x\notin L\implies\forall w.Pr_{y}\left[M\left(x,y,w\right)=1\right]\leq\frac{1}{3}\text{ (soundness)}$$

הערה 5.7 בהגדרת \$ שראינו למעלה, מתקיים $MA=\exists\$P$ וכן $AM=\$\exists P$. מרלין המוכיח וארתור הבודק, כלומר AM זו המחלקה בה ארתור נותן אתגר ואז מרלין נותן הוכחה, וMA המחלקה בה מרלין נותן הוכחה ובודק הסתברותי בודק אותה.

טענה 6.7 אמפליפיקציה $MA_{\left[\frac{1}{3},\frac{2}{3}\right]}=MA_{\left[2^{-2n},1-2^{-2n}\right]}$.

הוכחה: תהי $L\in MA_{\left[\frac{1}{3},\frac{2}{3}\right]}$ שנפתרת ע"י $M\left(x,y,w\right)$, נבנה M' ע"י $M'\left(x,\left(y_{1},\ldots,y_{T}\right),w\right)=Maj_{1\leq i\leq T}M\left(x,y_{i},w\right)$.

אם $x\in L$ אז יש w כך ש$Pr_{y}\left[M\left(x,y,w\right)=1\right]\geq\frac{2}{3}$, ועבור ה$w$ הזה ההסתברות שנטעה היא לכל היותר 2^{-2n} מצ'רנוף.

אם $x\notin L$ אז לכל w מתקיים $Pr_{y}\left[M\left(x,y,w\right)=1\right]\leq\frac{1}{3}$, ולכן לכל w ההסתברות שנטעה ששולחים w הוא ההסתברות שהרוב טעו שהיא גם לכל היותר 2^{-2n} מצ'רנוף. ■

טענה 7.7 אמפליפיקציה $AM_{\left[\frac{1}{3},\frac{2}{3}\right]}=AM_{\left[2^{-2n},1-2^{-2n}\right]}$.

הוכחה: תהי $L\in AM_{\left[\frac{1}{3},\frac{2}{3}\right]}$ שנפתרת ע"י $M\left(x,y,w\right)$, נבנה M' ע"י $M'\left(x,\left(y_{1},\ldots,y_{T}\right),\left(w_{1},\ldots,w_{T}\right)\right)=Maj_{1\leq i\leq T}M\left(x,y_{i},w_{i}\right)$.
נסמן $Y=\left\{ y\mid\exists w.M\left(x,y,w\right)=1\right\}$.

אם $x\notin L$ ההסתברות שנטעה היא ההסתברות שחצי מההטלות נפלו לY, אבל לY יש הסתברות לכל היותר $\frac{1}{3}$ ולכן זה $\leq2^{-\Omega(T)}$ לפי צ'רנוף.

אם $x\in L$ אז ההסתברות שנטעה היא ההסתברות שפחות מחצי מההטלות נפלו לY, אבל שוב מצ'רנוף זה קטן כי ההסתברות של Y היא לפחות $\frac{2}{3}$. ■

משפט 8.7 $MA\subseteq AM$.

הוכחה: תהי $L\in MA=MA_{\left[\frac{1}{3},\frac{2}{3}\right]}=MA_{\left[2^{-2m},1-2^{-2m}\right]}$ כאשר m עורך העד, אז יש מוודא $M\left(x,w,y\right)$ כך ש

$$x\in L\implies\exists w.Pr_{y}\left[M\left(x,w,y\right)=1\right]\geq1-2^{-2n}$$
$$x\notin L\implies\forall w.Pr_{y}\left[M\left(x,w,y\right)=1\right]\leq2^{-2n}$$

נבנה פרוטוקול AM לשפה L. המוודא שולח y למוכיח, המוכיח שולח בחזרה w והמוודא יקבל אם ורק אם $M\left(x,w,y\right)=1$.

אם $x\in L$, אז יש w_{0} כך ש$Pr_{y}\left[M\left(x,w_{0},y\right)=1\right]\geq1-2^{-2m}$. אם המוכיח תמיד ישלח את w_{0} הזה, אז בהסתברות לפחות $1-2^{-2m}$ יתקיים $M\left(x,w_{0},y\right)=1$, כלומר

$$Pr_{y}\left[\exists w.M\left(x,y,w\right)=1\right]\geq Pr_{y}\left[M\left(x,y,w_{0}\right)=1\right]\geq1-2^{-2m}$$

להיפך, אם $x\notin L$ אז לכל w מתקיים $Pr_{y}\left[M\left(x,w,y\right)=1\right]\leq2^{-2m}$. לכן,

$$Pr_{y}\left[\exists w.M\left(x,y,w\right)=1\right]\leq\sum_{w\in\left\{ 0,1\right\} ^{m}}Pr_{y}\left[M\left(x,y,w\right)=1\right]\leq2^{m}\cdot2^{-2m}=2^{-m}$$

■

משפט 9.7 יש perfect completeness לMA. כלומר, $MA_{\left[\frac{1}{3},\frac{2}{3}\right]} = MA_{\left[\frac{1}{2},1\right]}$. משם אפשר לעשות אמפליפיקציה ולקבל $MA = MA_{[2^{-n},1]}$.

הוכחה: תהי $L \in MA$, אז היא נפתרת על ידי $M(x,w,y)$ כך ש

$$x \in L \implies \exists w.Pr_y\left[M(x,w,y)=1\right] \geq \frac{1}{2}$$
$$x \notin L \implies \forall w.Pr_y\left[M(x,w,y)=1\right] \leq \frac{1}{4m}$$

כאשר m עורך העד w והמטבעות y.

נבנה פרוטוקול MA לשפה L. יש קלט x, נבקש מהמוכיח את w, וגם $s_1,\ldots,s_{2m} \in \{0,1\}^m$ מטבעות. המוודא יטיל $y \in \{0,1\}^m$, ויבדוק ש

$$M(x,w,y\oplus s_1) = 1 \vee \cdots \vee M(x,w,y\oplus s_{2m}) = 1$$

אם $x \in L$, אז יש w כך ש$|Y| \geq \frac{2^m}{2}$, ולכן כמו במשפט סיפסר (השיטה ההסתברותית) יש $2m$ הזזות כך ש$\bigcup_{i=1}^{2m}(Y\oplus s_i) = \{0,1\}^m$, ולכן לכל y יש i כך ש$y \in Y \oplus s_i$ כלומר $y \oplus s_i \in Y$ ולכן הביטוי למעלה יתקיים והמוודא יחזיר כן תמיד.

להיפך, אם $x \notin L$ אז לכל w מתקיים $|Y| \leq \frac{2^m}{4m}$, ולכן לכל $s_1,\ldots,s_{2m}$ יתקיים $\left|\bigcup_{i=1}^{2m}(Y\oplus s_i)\right| \leq \frac{2^m}{2}$, כלומר בהסתברות לכל היותר $\frac{1}{2}$ נקבל $y \in \bigcup_{i=1}^{2m}(Y\oplus s_i)$, כלומר בהסתברות לפחות $\frac{1}{2}$ המוודא ידחה. ■

הערה 10.7 גם לAM אפשר להבטיח perfect completeness, כלומר $AM_{\left[\frac{1}{3},\frac{2}{3}\right]} = AM_{[2^{-n},1]}$.

טענה 11.7 קשר בין מחלקות $BPP, NP \subseteq MA \subseteq AM \subseteq \Pi_2$, כאשר המעבר האחרון מperfect completeness. כמו כן $MA \subseteq \Sigma_2$.

משפט 12.7 כל פרוטוקול אינטראקטיבי עם מספר קבוע של סיבובים מוכל בAM.

הגדרה 13.7 המחלקה IP מחלקת כל השפות L עבורן מוכיח כל יכול להוכיח שייכות בשפה לבודק הסתברותי פולינומי באמצעות פרוטוקול אינטראקטיבי עם מספר פולינומי של סיבובים ומטבעות פומביים.

כלומר, מספר פולינומי של סיבובים שבכל סיבוב V בוחר $y_i \in \{0,1\}^m$ באקראי ושולח לP, P מחשב את $a_i = P(x,y_1,y_2,\ldots,y_i)$ ושולח לV. בסיבוב האחרון $T = poly(|x|)$ (מספר הסיבובים ידוע מראש כתלות בפרוטוקול), הבודק מחשב $V(x,y_1,a_1,\ldots,y_T,a_T)$ ומחליט אם לקבל או לדחות. נגיד ש$L \in IP_{[s,c]}$ כך ש

$$x \in L \implies \exists P.Pr_{y_1,\ldots,y_T}\left[V \text{ accepts on interaction with } P\right] \geq c$$
$$x \notin L \implies \forall P^*.Pr_{y_1,\ldots,y_T}\left[V \text{ accepts on interaction with } P^*\right] \leq s$$

משפט 14.7 $IP \subseteq PSPACE$.

הוכחה: $L \in IP$ לכן יש פרוטוקול אינטראקטיבי עם בודק V כך שהתנאי מעלה מתקיים. זה כמו ההוכחה ש$TQBF \in PSPACE$, מסתכלים על העץ של כל האפשרויות לאינטראקציה ובודקים מה הערך שלו על ידי טיול על העץ והתפצלות על כל האפשרויות.

פורמלית, נגדיר $Val\colon \text{partial history} \to [0,1]$ על ידי $Val(x,y_1,a_1,\ldots,y_m,a_m)$, באופן הבא: נגדיר עץ בעומק $2Y$ שיש בו התפצלויות מתחלפות y_i, a_i, ואז Val של עלה הוא $M(x,y_1,a_1,\ldots,y_T,a_T)$, Val של קודקוד שמתפצל על y_i הוא ממוצע הVal של בניו, וa_i המקסימום של הVal של בניו.

נטען שVal של קודקוד $x, y_1, a_1, \dots, y_i, a_i$ זו בדיוק ההסתברות שהבודק יקבל במשחק מול מוכיח אופטימלי שממשיך את ההיסטוריה עד עכשיו (הוכחה באינדוקציה), ולכן בהינתן $L \in IP$ שנפתרת על ידי M אפשר לחשב ב$PSPACE$ את $Val(\emptyset)$, כמו את $TQBF$. ■

משפט 15.7 ניסן $coNP \subseteq IP$. כלומר, $\overline{3SAT} \in IP$.

הוכחה: האינטואיציה שלנו תהיה קודים מתקני שגיאות, ונרצה לבצע אריתמטיזציה של הבעיה.

אריתמטיזציה: הקלט הוא פסוק $3SAT$, $\bigcap_{i=1}^m C_i$ כאשר $C_i = \ell_{i1} \vee \ell_{i2} \vee \ell_{i3}$ וכל ליטרל ℓ_i הוא משתנה או שלילתו. ϕ תמיר משתנה x לא, $\neg x$ ל $1-x$ ו $\ell_1 \wedge \ell_2$ ל$\phi(\ell_1) \cdot \phi(\ell_2)$, עם דה מורגן. לדוגמה, $\phi(x_1 \vee \neg x_2 \vee x_7) = \neg(\neg x_1 \wedge x_2 \wedge \neg x_7) = 1 - (1 - x_1) x_2 x_7$. כלומר, אם ניקח פסוק $\psi(x_1, \dots, x_n)$ ונמיר אותו לפולינום $p(x_1, \dots, x_n)$ (מכפלת הפולינומים של הפסוקיות), נקבל שלכל השמה $(a_1, \dots, a_n) \in \{0,1\}^n$ מתקיים $p(a_1, \dots, a_n) = \psi(a_1, \dots, a_n)$ וביתר הערכים הפולינום מרחיב את הנוסחה (כמו בקידוד). כלומר, לכל $b_1, \dots, b_n \in \mathbb{F}$ (שדה סופי שמרחיב את $\mathbb{F}_2$), אנחנו יכולים לחשב את $p(b_1, \dots, b_n)$.

הפרוטוקול: הקלט הוא פסוק $3SAT$, $\psi(x_1, \dots, x_n) = \bigcap_{i=1}^m C_i$, והמוכיח טוען שהפסוק לא ספיק.

נתרגם כל פסוקית C_i ל$q_i(x_1, \dots, x_n)$ ונגדיר

$$q(x_1, \dots, x_n) = \prod_{i=1}^m q_i(x_1, \dots, x_n)$$

רוצים לבדוק שאכן $\sum_{x_1, \dots, x_n \in \{0,1\}^n} q(x_1, \dots, x_n) = 0$ בשלמים (כלומר מספר ההשמות המספקות הוא 0), ומספיק גם לבדוק בשדה גדול מספיק $\mathbb{F}$ (עם מציין גדול מ2^n).

בסיבוב הראשון, המוכיח שולח פולינום במשתנה אחד מדרגה לכל היותר $3m$, $p_1 \in \mathbb{F}[x]$ (המוכיח ההוגן שולח את $p_1(x) = \sum_{a_2, \dots, a_n \in \{0,1\}^{n-1}} q(x, a_2, \dots, a_n)$), זה אפשרי כי לפולינום כזה יש ייצוג קצר. המוודא יבדוק שמתקיים $p_1(0) + p_1(1) = 0$, יטיל $y_1 \in \mathbb{F}$ וישלח אותו למוכיח.

עכשיו יש טענה חדשה על השולחן, והיא ש$\sum_{a_2, \dots, a_n} q(y_1, a_2, \dots, a_n) = p_1(y_1)$.

בסיבוב השני, המוכיח שולח פולינום $p_2 \in \mathbb{F}[x]$ מדרגה לכל היותר $3m$ (המוכיח ההוגן שולח את $p_2(x) = \sum_{a_3, \dots, a_n} q(y_1, x, a_3, \dots, a_n)$). המוודא יבדוק שמתקיים $p_2(0) + p_2(1) = p_1(y_1)$, יטיל $y_2 \in \mathbb{F}$ וישלח אותו למוכיח.

וכן הלאה, במשך n סיבובים. בסוף נבדוק שבאמת מתקיים $p_n(y_n) = q(y_1, \dots, y_n)$.

נכונות: נראה perfect completeness, וsoundness $\frac{3mn}{|F|}$.

אם $\psi \in 3SAT$, אז מהתיאור שלנו למוכיח ההוגן נקבל נכונות: לכל הטלה של y_i,

$$p_{i+1}(0) + p_{i+1}(1) = \sum_{a_{i+1}, a_{i+2}, \dots, a_n \in \{0,1\}^{n-i-2}} q(y_1, \dots, y_i, a_{i+1}, a_{i+2}, \dots, a_n) = p_i(y_i)$$

אם $\psi \notin 3SAT$, ניקח מוכיח P^* שממקסם את ההסתברות של V לקבל. נסמן בp_i את הפולינומים האמיתיים

$$p_i(x) = \sum_{a_{i+1}, \dots, a_n} q(y_1, \dots, y_{i-1}, x, a_{i+1}, \dots, a_n)$$

ונסמן ב$\tilde{p_i}$ את הפולינומים שP^* שולח. כמו כן, נשים לב שמכך ש$\psi \notin 3SAT$, בהכרח מתקיים $\sum_{a_1, \dots, a_n \in \{0,1\}} q(a_1, \dots, a_n) > 0$.

ראשית נטען ש$\tilde{p_1} \neq p_1$. אכן, אחרת מתקיים

$$\tilde{p_1}(0) + \tilde{p_1}(1) = p_1(0) + p_1(1) = \sum_{a_1, \dots, a_n \in \{0,1\}} q(a_1, \dots, a_n) > 0$$

כלומר הבודק בוודאות ידחה, וזו סתירה לכך שP^* ממקסם את ההסתברות שלו לקבל.

כעת, נניח ששיחקנו i שלבים והפולינום שנשלח בשלב הi $\tilde{p_i}$ מקיים $\tilde{p_i} \neq p_i$. אז בהסתברות לפחות $1-\frac{3m}{|F|}$ הבודק בוחר y_i כך ש$p_i\left(y_i\right) \neq \tilde{p_i}\left(y_i\right)$, ובמקרה זה בהכרח $p_{i+1} \neq \tilde{p_{i+1}}$, כי אחרת

$$\tilde{p_{i+1}}\left(0\right)+\tilde{p_{i+1}}\left(1\right)=p_{i+1}\left(0\right)+p_{i+1}\left(1\right)=p_i\left(y_i\right)\neq\tilde{p_i}\left(y_i\right)$$

בסך הכל, אם $\tilde{p_1}\left(y_1\right) \neq p_1\left(y_1\right),\ldots,\tilde{p_{n-1}}\left(y_n\right) \neq p_{n-1}\left(y_n\right)$ אז בהסתברות $1-\frac{3m}{|F|}$ הבודק יבחר y_n כך ש$\neq p_n\left(y_n\right)$ $q\left(y_1,\ldots,y_n\right)$ וידחה.

בסך הכל, ההסתברות שV יקבל היא

$$P\left(\tilde{p_1}\left(y_1\right)=p_1\left(y_1\right)\vee\cdots\vee\tilde{p}_n\left(y_n\right)=p_n\left(y_n\right)\right)\leq\sum_{i=1}^{n}P\left(\tilde{p_i}\left(y_i\right)=p_i\left(y_i\right)\right)=\frac{n\cdot 3m}{|F|}$$

■

מסקנה 16.7 זה גורר $PH \subseteq IP$ (לא נראה), ועדי שמיר הראה שאפשר להכליל ולהראות $TQBF \in IP$, כלומר $IP=PSPACE$.

8 משפט הPCP וקושי של קירוב

8.1 משפט הPCP ובעיות הבטחה

הגדרה 1.8 המחלקה $PCP\left(r\left(n\right),q\left(n\right)\right)$ מכילה את כל השפות L עבורת קיים בודק $V\left(x,w,y\right)$ פולינומי כאשר x הקלט באורך n, w ההוכחה באורך $m\left(n\right)$ וy המטבעות של הבודק באורך $r\left(n\right)$,

כך שבהינתן $y\in\{0,1\}^{r(n)}$ הבודק V מחשב $q=q\left(n\right)$ אינדקסים $\bar{i}=i_1,\ldots,i_q\in\{1,\ldots,m\}$ [תלויים רק בy, הבודק לא רואה את w] ומחשב פרדיקט $V\left(x,y,w_{i_1},\ldots,w_{i_q}\right)$, ומתקיים

$$x\in L\implies\exists w.Pr_y\left[V\left(x,w,y\right)=1\right]\geq c$$
$$x\notin L\implies\forall w.Pr_y\left[V\left(x,w,y\right)=1\right]\leq s$$

דוגמה 2.8 $3SAT\in PCP_{\left[1-\frac{1}{m},1\right]}\left(O\left(\log m\right),3\right)$ כאשר m מספר הפסוקיות בנוסחה.

הוכחה: הקלט הוא $\varphi\left(\bar{x}\right)=\bigcap_{i=1}^{m}C_i\left(\bar{x}\right)$, המוכיח שולח השמה $w=\left(a_1,\ldots,a_n\right)\in\{0,1\}^n$ והבודק יגריל $1\leq i\leq m$ (לשם כך צריך $\log m$ ביטים של אקראיות) ויבדוק ש$C_i\left(\overline{a}\right)=T$ - לשם כך הוא צריך לקרוא רק 3 אינדקסים מa, על פי שלושת הליטרלים שמופיעים בC_i.

אם $\varphi\in 3SAT$ אז מוכיח שישלח w השמה מספקת (יש כזו) יקיים שלכל i הפסוקית C_i תסתפק, ולכן הבודק יקבל בהסתברות 1.

אם $\varphi\notin 3SAT$ אז לכל w יש i כך שC_i לא מסתפקת (אחרת φ ספיקה), ולכן בהסתברות לפחות $\frac{1}{m}$ הבודק יבחר אותה וידחה. ∎

משפט 3.8 הPCP $NP=PCP_{[0.9,1]}\left(O\left(\log n\right),3\right)$.

דוגמה 4.8 $GNI\in PCP_{\left[\frac{1}{2},1\right]}\left(O\left(n\log n\right),1\right)$.

הוכחה: נזכר שראינו פרוטוקול אינטראקטיבי: הקלט היה $\left(G_1,G_2\right)$, הבודק בחר $\pi\in S_n,b\in\{0,1\}$ ואז שלח למוכיח את $\pi\left(G_b\right)$, המוכיח שלח $c\in\{0,1\}$ והבודק בדק אם $b=c$. נרצה לתרגם אותו לפרוטוקול לא אינטראקטיבי.

הפרוטוקול: הבודק מטיל $\pi\in S_n,b\in\{0,1\}$ ($\left|\pi\right|=\log\left(n!\right)=O\left(n\log n\right)$) כלומר $y=\left(\pi,b\right)$, ולפי y מסתכל על הביט בהוכחה w שמייצג את התשובה של המוכיח על השאילתה $\pi\left(G_b\right)$, ובודק אם הביט שם שווה לb. ∎

הגדרה 5.8 בעיית הבטחה נגיד ש$\left(Y,N\right)$ היא בעיית הבטחה אם $Y,N\subseteq\{0,1\}^*$ וכן $Y\cap N=\emptyset$ (אבל לא בהכרח $Y\cup N=\{0,1\}^*$).

מכונה פותרת בעיית הבטחה אם לכל $x\in Y$ היא מחזירה כן, לכל $x\in N$ היא מחזירה לא, ועבור x שלא בY ולא בN, לא משנה מה היא מחזירה.

לדוגמה, שפה L היא בעיית הבטחה שבה $Y=L,N=\{0,1\}^*\setminus L$.

הגדרה 6.8 בעיית gap נגדיר בעיית הבטחה $Gap_{[a,b]}Clique$ עבור $a<b$ על ידי קלט גרף $G=\left(V,E\right)$, Y אם יש קליקה בגודל $\leq b$, וN אם כל קליקה בגודל $\geq a$.

באופן דומה, ניתן להגדיר גם דברים כמו $Gap_{[a,b]}col$ עם קלט G וY אם הוא צביע על ידי $\geq a$ צבעים, N אם כל צביעה דורשת $\leq b$ צבעים, וגם $Gap_{[\alpha,\beta]}\max 3SAT$ עם קלט $\varphi=\bigcap_{i=1}^{m}C_i$ וY אם יש השמה שמספקת לפחות βm פסוקיות, N אם כל השמה מספקת לכל היותר αm פסוקיות.

הערה 7.8 הבעיה $3COL = Gap_{[3,4]}col$ היא NP־קשה, אבל $Gap_{[3,3\sqrt{n}]}col$ היא בP. כמו כן, $Gap_{[\frac{9}{10},1]}\max 3SAT$ היא NP־קשה (נובע ממשפט הPCP, נראה) אבל $Gap_{[<\frac{1}{2},1]}\max 3SAT$ היא בP, כי במקרה זה $N=\emptyset$ ולכן אפשר לומר תמיד כן ־ אכן, אם כל השמה מספקת $<\frac{1}{2}$ מהפסוקיות אז ניקח השמה כלשהי כזו אז גם הnot שלה יספק $<\frac{1}{2}$ פסוקיות, וזו סתירה כי הnot מספק בדיוק את הפסוקיות שההשמה המקורית לא מספקת.

נשים לב גם שאם $\varphi = \bigcap_{i=1}^{m} C_i$ כאשר כל C_i זה $\vee$ על 3 ליטרלים שונים (כלומר הבעיה היא $E3SAT$) אז השמה אקראית מספקת בתוחלת $\frac{7}{8}m$ מהפסוקיות (לכל פסוקית יש סיכוי $\frac{7}{8}$ להסתפק) ולכן תמיד יש השמה שמספקת $\le \frac{7}{8}m$ מהפסוקיות, כלומר $Gap_{[\frac{7}{8},1]} \in P$ כי שוב $N=\emptyset$.

הגדרה 8.8 רדוקציות בין בעיות הבטחה נניח $\Pi_1 = (Y_1,N_1), \Pi_2 = (Y_2,N_2)$ בעיות הבטחה, נאמר ש$\varphi\colon \{0,1\}^* \to \{0,1\}^*$ היא רדוקציה מΠ_1 לΠ_2 ונסמן $\Pi_1 \le_\varphi \Pi_2$ אם לכל $x\in Y_1$ מתקיים $\varphi(x)\in Y_2$, ולכל $x \in N_1$ מתקיים $\varphi(x) \in N_2$.

נאמר שבעיית הבטחה Π היא NP־קשה אם לכל $L\in NP$ יש רדוקציה $L \le_{Log} \Pi$.

טענה 9.8 $Gap_{[\alpha,\beta]}\max 3SAT \in PCP_{[\alpha,\beta]}(O(\log n),3)$.

הוכחה: בהינתן קלט $\varphi = \bigcap_{i=1}^{m} C_i$ הבודק מגריל פסוקית C_i עבור $1\le i\le m$, מסתכל על 3 הביטים בw (שמייצגת השמה) המתאימים לליטרלים בC_i, ובודקת אם הם מספקים את הפסוקית C_i.

אם $\varphi \in Y$ אז יש w שמספקת לפחות βm מהפסוקיות ולכן הבודק יקבל בהסתברות לפחות β (אם הוא בוחר אחת מהפסוקיות הללו)1, ואם $\varphi \in N$ אז לכל w לכל היותר αm מהפסוקיות יסתפקו, ולכן בהסתברות לכל היותר α הבודק יקבל. ■

נרצה להוכיח באמצעות משפט הPCP שלכל $\varepsilon > 0$, הבעיה $Gap_{[\frac{7}{8}+\varepsilon,1]}\max 3SAT$ היא NP קשה.

ממשפט הPCP נובע $NP = PCP_{[\frac{1}{2},1]}(O(\log n), O(1))$, כלומר לכל $L\in NP$ יש פרוטוקול שבו המוכיח שולח הוכחה $\bar{a} = a_1,\ldots,a_m$ כאשר $m = poly(n)$, והבודק מטיל $1\le i\le N$ ($N = poly(n)$) ולפי הi מחליט איזה $q = O(1)$ ביטים מההוכחה לקרוא, ובודק האם $V\left(x, a_{j_1(i)},\ldots,a_{j_q(i)}\right) = 1$. הבעיה היא שפה V יכולה להיות כל דבר, עם אילוצים כלליים ולא בהכרח נוסחה. לכן, נגדיר:

הגדרה 10.8 בעיית הCSP הקלט הוא עולם Λ ואוסף של אילוצים (constraints) $C_1,\ldots,C_m$, כאשר כל C_i מגדיר q אינדקסים מΛ ומגדיר פרדיקט V_i שהוא דורש על q הביטים הנ"ל.

נגדיר את Y להיות כל הקלטים כך שיש השמה לΛ שמספקת $\le \beta m$ מהאילוצים, וN להיות כל הקלטים כך שכל השמה לΛ מספקת $\ge \alpha m$ מהאילוצים.

טענה 11.8 אם $(Y,N)\in PCP_{[\alpha,\beta]}(r(n),q(n))$ כאשר $q = O(1)$ אז יש רדוקציה $(Y,N) \le_P CSP_{[\alpha,\beta]}$.

הוכחה: $(Y,N)\in PCP_{[\alpha,\beta]}(r(n),q(n))$ לכן יש מערכת הוכחה שבה המוכיח שולח m ביטים כאשר $m = poly(n)$, הבודק מטיל $r(n)$ מטבעות שנסמן i ולפיהם מחליט איזה $q(n)$ ביטים לקרוא, ואז בודק $V_i\left(x, a_{j_1},\ldots,a_{j_q}\right)$ (כלומר $V\left(x, i, a_{j_1},\ldots,a_{j_q}\right)$).

הרדוקציה φ תקבל $x\in\{0,1\}^*$ ותבנה מופע של CSP: מגדירה $\Lambda = \{1,\ldots,m\}$, לכל $0\le i\le 2^{r(n)}$ נגדיר C_i טבלת האמת על המשתנים $j_1(i),\ldots,j_q(i)$ שמקבלת את הערך שלה לפי V_i.

אם $x\in Y$ אז מהגדרת מערכת הPCP יש מוכיח $\bar{a}$ שעבורו הבודק יקבל בהסתברות $\le \beta$, ולכן אם נבחר את ההשמה לΛ על ידי $a(i) = \bar{a}_i$ אז $Pr_i[V \text{ accepts}] \ge \beta$, ובאותו האופן אם $x\in N$. ■

טענה 12.8 יש $\alpha' < 1$ כך ש$CSP_{[\alpha,1]}(r,q) \le Gap_{[\alpha',1]}\max 3SAT$ כאשר $q = O(1)$.

הוכחה: נניח $C_1,\ldots,C_m$ קלט לבעיית הCSP עם $m = 2^r$. נעבור על $i = 1,\ldots,m$, כל C_i מסתכל על $j_1,\ldots,j_q$, נתרגם אותו לפסוק $3SAT$ באורך ℓ (קבוע, כי q קבוע). כלומר, כל constraint בודד הופך לℓ פסוקיות בנוסחה. הנוסחה הסופית תהיה ה$\wedge$ של כל הפסוקיות הללו.

ואז, אם $x \in Y_{CSP}$ אז יש השמה שמספקת את כל V_i ולכן יש השמה שמספקת את כל הפסוקיות בנוסחת ה$3SAT$.

להיפך, אם $x \in N_{CSP}$ אז לכל השמה

$$Pr_i\left[V_i \text{ is satisfied}\right] \le \alpha \implies Pr_i\left[\psi_i \text{ is satisfied}\right] \le \alpha \implies Pr_i\left[\exists 1 \le j \le \ell . \psi_{i,j} \text{ is not satisfied}\right] \ge 1-\alpha$$
$$\implies Pr_{i,j}\left[\psi_{i,j} \text{ is not satisfied}\right] \ge (1-\alpha)\cdot\frac{1}{\ell}$$

כלומר $(1-\alpha)\cdot\frac{1}{\ell}$ מהפסוקיות לא מסתפקות, וזה מספיק כי ℓ קבוע. ■

מסקנה 13.8 יש $\alpha' < 1$ כך שהבעיה $Gap_{[\alpha',1]}\max 3SAT$ היא NP־קשה. (לא נוכיח, אבל אפשר להראות, שלכל $\varepsilon > 0$ קבוע $Gap_{\left[\frac{7}{8}+\varepsilon,1\right]}\max 3SAT$ היא NP־קשה).

הוכחה: תהי $L \in NP$, ממשפט הPCP נובע $L \in PCP_{\left[\frac{1}{2},1\right]}\left(O\left(\log n\right), O\left(1\right)\right)$ ולכן מהטענה הראשונה שהראינו $L \le CSP_{\left[\frac{1}{2},1\right]}$, וכעת מהטענה האחרונה נובע $CSP_{\left[\frac{1}{2},1\right]} \le Gap_{[\alpha',1]}\max 3SAT$ ולכן $L \le Gap_{[\alpha',1]}\max 3SAT$. ■

8.2 רדוקציות משמרות gap וקושי של קירוב

כעת, נזכר שראינו במודלים $3SAT \le Clique \le IS \le VC$. נבדוק האם הרדוקציות שראינו הן משמרות פער, כלומר עובדות גם בגרסאות הgap ־ ואם כן (ואכן כן), נקבל שגם גרסאות הgap של יתר הבעיות הללו הן NP־קשות.

דוגמה 14.8 הרדוקציה $3SAT \le Clique$ לוקחת נוסחה עם m פסוקיות והופכת אותו לגרף עם m רמות (אחת עבור כל פסוקית), כאשר בכל רמה יש 3 קודקודים (אחד עבור כל ליטרל), ומחברים בין קודקודים ברמות שונות אם הם עקביים (אותו משתנה עם אותו סימן או משתנים שונים).

נבצע את אותה הרדוקציה מ$Gap_{[\alpha,1]}\max 3SAT$ ל$Gap_{[s,c]}\max Clique$ ־ מה הפער בין s,c שנקבל? אם $\varphi \in Y_{GAP\max 3SAT}$ אז יש השמה שמספקת את כל הפסוקיות ולכן יש בG קליקה בגודל m. להיפך, אם $x \in N_{GAP\max 3SAT}$ אז כל השמה מספקת לכל היותר αm מהפסוקיות ולכן בG הקליקה המקסימלית היא בגודל $\ge \alpha m$. אכן, זאת כי אם יש בG קליקה בגודל t היא בהכרח הולכת בין שכבות שונות, בכל שכבה היא בחרה קודקוד והם עקביים ולכן ההשמה שמספקת אותם מספקת t פסוקיות בφ. כלומר, $s = \alpha\frac{|V|}{3}, c = \frac{|V|}{3}$ (כי בגרף יש $3m$ קודקודים) ויש פער.

באופן דומה, הרדוקציה מ$Clique$ לIS (לקחת משלים) מראה ש$Gap_{[a,b]}\max Clique \le Gap_{[a,b]}\max IS$.

לגבי הרדוקציה $IS \le VC$, נזכר שA היא IS בG $\iff$ $V \setminus A$ היא VC בG, ולכן הרדוקציה מקבלת G,k ומוציאה את $G, n-k$. לכן, מקבלים $Gap_{[a,b]}\max IS \le Gap_{[n-b,n-a]}\max VC$.

כלומר, קיבלנו

$$Gap_{[0.9,1]}\max 3SAT \le GAP_{\left[0.9\frac{|V|}{3},\frac{|V|}{3}\right]}\max IS \le Gap_{\left[\frac{2}{3}|V|,0.7|V|\right]}\max VC$$

ולכן $Gap_{\left[\frac{2}{3}|V|,0.7|V|\right]}\max VC$ היא בעיה NP קשה.

טענה 15.8 אם מצליחים לקרב את גודל הVC המינימלי בגרף באופן יעיל, כלומר למצוא אלג' פולינומי שבהינתן גרף G מציא מספר num כך ש

$$\text{size of } \min VC \le num \le 1.01 \cdot \text{size of } \min VC$$

אז $P = NP$.

הוכחה: תהי $L \in NP$, בהינתן קלט x נפעיל את הרדוקציה $L \le Gap_{[\frac{2}{3}|V|,0.7|V|]} \min VC$, נקבל גרף G ונפעיל את האלג' לקירוב על G. נקבל מספר num, אם $num \le 1.01 \cdot \frac{2}{3}|V|$ אז נגיד כן, אחרת נגיד לא.

אז, אם $x \in L$ אז $G = \varphi(x) \in Y_{Gap \min VC}$ ולכן בG יש VC בגודל $\le \frac{2}{3}|V|$, ומכאן $num \le 1.01 \cdot \frac{2}{3}|V|$ כלומר נגיד כן.

באופן דומה, אם $x \notin L$ אז בG הVC המינימלי הוא בגודל $\le 0.7|V|$, ומכאן $num \ge 0.7|V| > 1.01 \cdot \frac{2}{3}|V|$ כלומר נגיד לא.

כלומר, הצלחנו לפתור את L בזמן פולינומי, ולכן $P = NP$. ∎

הגדרה 16.8 קירוב נגיד שאלגוריתם A c־מקרב (כאשר $c \ge 1$) את הפתרון של בעיית מקסימיזציה אם לכל $x \in \{0,1\}^*$, מתקיים

$$\frac{opt(x)}{c} \le A(x) \le opt(x)$$

כאשר $opt(x) = \max_{w \text{ s.t. } M(x,w)=1} Val(w)$, כאשר Val פונקציה שנותנת ערך מספרי לפתרונות.

טענה 17.8 אם יודעים עבור יחס $M(x,w)$ וval שהבעיה $Gap_{[a,b]} \max Val\, M$ היא NP קשה, אז NP־קשה לקרב את $\max Val\, M$ עד כדי פקטור $\frac{b}{a}$ (ובאותו האופן עבור min).

הוכחה: באותו האופן כמו הטענה הקודמת (יש קצת עדינות אם האם דורשים קטן ממש או קטן שווה מa בN, הטענה נכונה עם קטן ממש ואחרת צריך פקטור $< \frac{b}{a}$). ∎

הערה 18.8 ראינו בתרגיל ש$Gap_{[0.68,0.69]} \max 2SAT$ היא NP־קשה, ולכן קשה לקרב את $\max 2SAT$ עד כדי פקטור $\frac{0.69}{0.68}$. אבל זה לא אומר ש$Gap_{[0.9,1]} \max 2SAT$ היא NP־קשה (היא אפילו בP) ־ טענה על gap היא יותר ספציפית, ולא נגררת מקושי של קירוב.

הגדרה 19.8 נגדיר $3SAT(b)$, הבעיה של להכריע ספיקות של נוסחה, כאשר הנוסחה בצורת $3SAT$ כאשר בנוסף כל משתנה מופיע לכל היותר b (קבוע) פעמים.

טענה 20.8 $3SAT(3)$ היא NP־קשה.

הוכחה: נראה רדוקציה $3SAT \le 3SAT(3)$. בהינתן פסוק $\varphi = \bigcap_{i=1}^{m} C_i$ ניצור $3m$ משתנים חדשים x_{i1}, x_{i2}, x_{i3} כאשר $1 \le i \le m$, ונמיר את הפסוקיות למשתנים החדשים ־ לדוגמה, הפסוקית C_i שמוגדרת $x_{10} \vee x_7 \vee \neg x_5$ עוברת ל$x_{i1} \vee x_{i2} \vee \neg x_{i3}$. כמו כן, נוסיף פסוקיות חדשות של קונסיסטנטיות ־ לכל $1 \le j \le n$, נסתכל על כל המופעים של x_j בφ, נניח הוא הופיע k_j פעמים והוחלף על ידי המשתנים $w_1, \ldots, w_{k_j}$, נוסיף פסוקיות שמתאימות לטענה $w_1 \implies w_2 \implies \cdots \implies w_{k_j} \implies w_1$, על ידי $(\neg w_1 \vee w_2), (\neg w_2 \vee w_3), \ldots, (\neg w_{k_j} \vee w_1)$. כל משתנה מופיע לכל היותר פעם אחת בפסוקיות המקוריות, ופעמיים בפסוקיות הנוספות ־ בסך הכל 3 פעמים. ∎

טענה 21.8 יש $\gamma' < 1$ וb קבוע כך שהבעיה $Gap_{[\gamma',1]} \max 3SAT(b)$ היא NP־קשה.

הוכחה: היינו רוצים להשתמש באותה רדוקציה שהראינו מ$3SAT$ ל$3SAT(b)$ כרדוקציה מ$Gap_{[\gamma,1]} \max 3SAT$ ל$Gap_{[\gamma',1]} \max 3SAT(b)$ ־ אבל האם היא משמרת gap? לא, כי לדוגמה נניח x_1 מופיע בכל פסוקית של φ, ב$\frac{m}{2}$ חיובי וב$\frac{m}{2}$ שלילי, נבחר השמה לφ' כך: המופעים החיוביים T, המופעים השליליים F, אז כל הפסוקיות המקוריות יסתפקו, וגם כמעט כל הפסוקיות הנוספות יסתפקו ־ כל הגרירות מחיובי לחיובי או שלילי לשלילי יסתפקו, גם משלילי לחיובי, רק גרירה אחת מחיובי לשלילי לא תסתפק. ואז, מתוך כל $4m$ הפסוקיות נספק $4m-1$. כלומר, הרדוקציה מעבירה

$$Gap_{[\gamma m,m]} \max 3SAT \le Gap_{[4m-1,4m]} \max 3SAT(3)$$

כלומר הgap נעלם ־ היינו רוצים $[\gamma' \cdot 4m, 4m]$ כאשר $\gamma' < 1$ קבוע.

לכן, נרצה לתקן את הרדוקציה. נעבור על כל n המשתנים, עבור i, נניח שx_i מופיע k_i פעמים, ונבנה גרף $G_i = (V, E)$ b־רגולרי שיש בו k_i קודקודים וקשת (w_1, w_2) תגיד $w_2 \iff w_1$, ונוסיף את הקשתות שלו כקשתות עקביות נוספות (כלומר, במקום מעגל מכוון, נרצה גרף יותר קשיר ־ אקספנדר!). יש בנוסחה החדשה $3m$ משתנים ו

$$m + \sum_{i=1}^{n} k_i \cdot b \leq (3b+1)\, m$$

פסוקיות, וכל משתנה מופיע בφ' לכל היותר $2b+1$ פעמים.

אם φ ספיק עם $a_1, \ldots, a_n$ אז אם לכל המשתנים שבאו מi נבחר השמה a_i אז φ' גם ספיק.

אבל אם $\varphi \in NO$, אז לכל השמה $a = a_1, \ldots, a_n$ לφ לפחות $(1-\gamma)\, m$ מהפסוקיות לא מסתפקות. נניח שבנינו את הגרף G_i כך שלכל $A \subseteq V$ עם $|A| \leq \frac{|V|}{2}$ מתקיים $|\Gamma(A)| > |A|$ כאשר $\Gamma(A)$ קבוצת כל השכנים של קודקודים בA (לא נראה איך בונים כזה, בשביל זה היה צריך לבוא לסמינר בתג"ס).

נראה שבהשמה האופטימלית המשתנים עקביים ־ אכן, אם היריב בחר השמה לφ' ועבור משתנה מסוים x_i חלק מהמשתנים המתאימים קיבלו T וחלק F, אז אחת הקבוצות היא קבוצת מיעוט, ואם הוא יחליף את הערך של קבוצת המיעוט לערך של השאר הוא יפסיד לכל משתנה בA לכל היותר פסוקית אחת מהפסוקיות המקוריות (בסך הכל $|A|$ פסוקיות), אבל הוא ירוויח $|\Gamma(A)|$ קשתות בגרף העקביות (קשתות שהיו מהצורה $F \implies T$) ־ ו$|A| < |\Gamma(A)|$ כי $|A| \leq \frac{|V_i|}{2}$, ולכן עדיף להחליף.

כלומר, ההשמה האופטימלית עקבית ולכן היא מספקת לכל היותר $\gamma m + 3bm$ מתוך $m + 3bm$ פסוקיות.

כלומר, קיבלנו רדוקציה משמרת gap:

$$Gap_{[\gamma,1]} \max 3SAT \leq Gap_{\left[\frac{3b+\gamma}{3b+1},1\right]} \max 3SAT\,(b)$$

■

סוף.