

שיעור 10

NP שלמות

10.1 המחלקות NPC ו- NPH

הגדרה 10.1 NP-hard

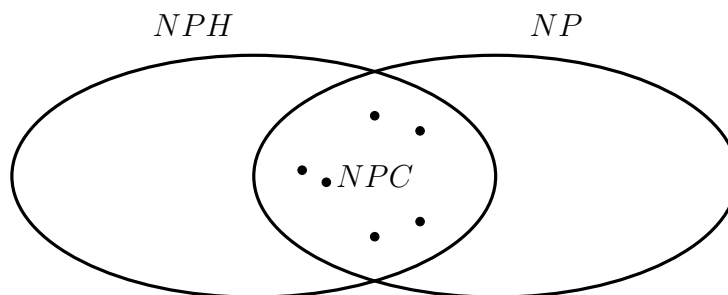
בעייה B נקראת NP קשה אם לכל בעייה $A \in NP$ קיימת רדוקציה $A \leq_p B$.

הגדרה 10.2 NP-complete

בעייה B נקראת NP שלמה אם

$$B \in NP \quad (1)$$

(2) לכל בעייה $A \in NP$ קיימת רדוקציה $A \leq_p B$.



משפט 10.1

אם $B \in NP$ שלמה וגם $B \in P$ אזי $P = NP$.

הוכחה:

• הוכחנו כבר ש- $P \subseteq NP$.

• נוכיח כי $NP \subseteq P$.

לכל בעייה $A \in NP$ קיימת רדוקציה $A \leq_p B$ ומכיון ש- $B \in P$, ממשפט הרדוקציה מתקיים $A \in P$.

מסקנה 10.1

אם $A \leq_p B$ אזי $\bar{A} \leq_p \bar{B}$.

משפט 10.2

אם $A \leq_p B$ וגם $B \leq_p C$ אזי $A \leq_p C$.

הוכחה:

משפט 10.3

תהי B בעייה NP -שלמה. אזי לכל בעייה $C \in NP$, אם $B \leq_p C$ אזי גם C היא NP שלמה.

הוכחה: מכיוון ש- B היא NP -שלמה, לכל בעייה $A \in NP$ קיימת רדוקציה $A \leq_p B$. מכיוון ש- $B \leq_p C$ מהטרנזיטיביות מתקיים $A \leq_p C$ לכל בעייה $A \in NP$. ולכן C היא NP -שלמה.

10.2 בעיית הספיקות

הגדרה 10.3

נוסחת CNF ϕ היא נוסחה בוליאנית מעל n משתנים x_1, x_2, \dots, x_n המכילה m פסוקיות C_1, C_2, \dots, C_m , כאשר כל פסוקית מכילה אוסף של ליטרלים $(x_i \vee \bar{x}_i)$ המחוברים ע"י OR (\vee) בוליאני והפסוקיות מחוברים ע"י AND (\wedge) בוליאני.

לדוגמה

$$\phi = \left(x_1 \vee \bar{x}_2 \vee x_4 \vee \bar{x}_7 \right) \wedge \left(x_3 \vee x_5 \vee \bar{x}_8 \right) \wedge \dots$$

הגדרה 10.4 נוסחת CNF ספיקה

נוסחת CNF ϕ היא ספיקה אפ קימת השמה למשתנים x_1, x_2, \dots, x_n ע"י $T \setminus F$ כך ש- ϕ מקבלת ערך T , כלומר בכל פסוקית ישנו לפחות ליטרל אחד שקיבל ערך T .

10.3 בעיית SAT

הגדרה 10.5 בעיית SAT

קלט: נוסחת CNF ϕ .
פלט: האם ϕ ספיקה?

$$SAT = \{ \langle \phi \rangle \mid \text{נוסחת } CNF \text{ ספיקה} \}$$

משפט 10.4 $SAT \in NP$

$$SAT \in NP.$$

הוכחה: נבנה אלגוריתם אימות V עבור SAT .

$$V = \langle \langle \phi \rangle, y \rangle \text{ על קלט}$$

(1) בודק האם y היא השמה למשתנים x_1, x_2, \dots, x_n .

• אם לא $3CNF \Leftarrow$ דוחה.

(2) בודק האם השמה זו מספקת את ϕ .

• אם כן \Leftarrow מקבל.

• אם לא \Leftarrow דוחה.

■

10.4 משפט קוק לוין

משפט 10.5 (1973) משפט קוק לוין

הבעיית SAT היא NP - שלמה.

רעיון ההוכחה:

לכל $A \in NP$, $A \leq_p SAT$.
לכל $w \in \Sigma^*$:

$$w \in A \Leftrightarrow f(w) \in SAT,$$

כאן $f(w) = \langle \phi_w \rangle$.

מסקנה 10.2

$$P = NP \Leftrightarrow SAT \in P.$$

10.5 גרסאות של $kSAT$

ישנן לכל היותר k ליטרלים בכל פסוקית:

• $1SAT \in P$.

$$\phi = x_1 \wedge \bar{x}_2 \wedge x_3 \wedge \dots$$

• $2SAT \in P$.

$$\phi = (x_1 \vee \bar{x}_2) \wedge (x_2 \vee x_4) \wedge \dots$$

• $3SAT$ היא NP - שלמה.

10.6 בעיית $3SAT$

הגדרה 10.6 בעיית $3SAT$

קלט: נוסחת $3CNF$, ϕ .
פלט: האם ϕ ספיקה?

$$3SAT = \{ \langle \phi \rangle \mid \phi \text{ נוסחת } 3CNF \text{ ספיקה} \}$$

משפט 10.6 $3SAT$ היא NP שלמה. $3SAT$ היא NP שלמה.

הוכחה:

יש לקיים את השני תנאים הבאים:

$$3SAT \in NP \quad (1)$$

ניתן לבנות אלגוריתם אימות עבור $3SAT \in NP$ דומה לאלגוריתם האימות עבור SAT שבנינו בהוכחה של המשפט קוק-לוין 10.5 למעלה.

$$3SAT \text{ היא } NP \text{ קשה ע"י רדוקציה} \quad (2)$$

$$SAT \leq_p 3SAT.$$

ואז בגלל ש- SAT היא NP שלמה (לפי משפט קוק-לוין 10.5) ומכיון ש- $3SAT \in NP$ אז לפי משפט האסימפטוטית 10.2 גם $3SAT$ היא NP -שלמה.

$$\underline{\text{קיום פונקצית הרדוקציה } SAT \leq_p 3SAT}$$

כעת נבנה את פונקציה הרדוקציה מ- SAT ל- $3SAT$.

ראשית נציין כי כל נוסחה בוליאנית ϕ ניתנת לרשום בצורה CNF בזמן פולינומיאלי.

בהינתן נוסחת CNF , ϕ (הקלט של SAT) נבנה בזמן פולינומיאלי נוסחת $3CNF$, ϕ' (הקלט של $3SAT$) ואז נוכיח שמתקיים

$$\langle \phi' \rangle \in 3SAT \iff \langle \phi \rangle \in SAT.$$

לכל פסוקית C ב- ϕ המכילה יותר מ-3 ליטרלים, ניצור אוסף C' ב- ϕ' של פסוקיות כך שכל פסוקית ב- C' תכיל 3 ליטרלים. למשל בהינתן הפסוקית C הבאה של ϕ :

$$C = x_1 \vee x_2 \vee x_3 \vee \bar{x}_4 \vee \bar{x}_5$$

ניצור את הפסוקית C' הבאה ב- ϕ' :

$$C' = (x_1 \vee x_2 \vee y_1) \wedge (\bar{y}_1 \vee x_3 \vee y_2) \wedge (\bar{y}_2 \vee \bar{x}_4 \vee \bar{x}_5).$$

באופן כללי, לכל פסוקית $C = a_1 \vee a_2 \vee \dots \vee a_k$ המכיל $k > 3$ ליטרלים, ניצור אוסף C' של פסוקיות שבו כל פסוקית מכילה 3 ליטרלים, ע"י הוספת $k-3$ משתנים y_1, y_2, \dots, y_{k-3} :

$$C' = (a_1 \vee a_2 \vee y_1) \wedge (\bar{y}_1 \vee a_3 \vee y_2) \wedge \dots \wedge (\bar{y}_{i-2} \vee a_i \vee y_{i-1}) \wedge \dots \wedge (\bar{y}_{k-3} \vee a_{k-1} \vee a_k).$$

בפרט, עבור כל פסוקית $C = (a_1 \vee a_2 \vee \dots \vee a_k)$ נניח $a_i = 1$ הוא הליטרל הראשון ששווה ל-1. אז

$$\bullet \text{ נשים } y_j = 1 \text{ לכל } 1 \leq j \leq i-2,$$

$$\bullet \text{ ונשים } y_j = 0 \text{ לכל } i-1 \leq j \leq k-3.$$

סיימנו להגדיר את הפונקציה הרדוקציה.

כעת נוכיח כי הפונקציה הזאת מקיימת את התנאי ההכרחי

$$\langle \phi' \rangle \in 3SAT \iff \langle \phi \rangle \in SAT.$$

כיוון \Leftarrow :

נניח כי $\langle \phi \rangle \in SAT$ ותהי X השמה המספקת את ϕ .

נוכיח שקיימת השמה X' מתאימה המספקת את ϕ' .

- בכל פסוקית C של ϕ , עבור הליטרלים a_1, a_2, \dots, a_k ניתן אותם ערכים כמו ב- X .
- מכיוון ש- X מספקת את ϕ , בכל פסוקית $C = (a_1 \vee a_2 \vee \dots \vee a_k)$ יש לפחות ליטרל אחד שקיבל ערך 1. נניח $a_i = 1$. אז על פי ההגדרה של פונקצית הרדוקציה:

$$* \text{ נשים } y_j = 1 \text{ לכל } 1 \leq j \leq i - 2,$$

$$* \text{ ונשים } y_j = 0 \text{ לכל } i - 1 \leq j \leq k - 3.$$

באופן הזה אנחנו ניצור אוסף C' של פסוקיות עם המבנה הבא:

$$\left(a_1 \vee a_2 \vee \overset{1}{y_1} \right) \wedge \left(\overset{0}{\bar{y}_1} \vee a_2 \vee \overset{1}{y_2} \right) \wedge \dots \wedge \left(\overset{0}{\bar{y}_{i-3}} \vee a_{i-1} \vee \overset{1}{y_{i-2}} \right) \wedge \left(\overset{0}{\bar{y}_{i-2}} \vee \overset{1}{a_i} \vee \overset{0}{y_{i-1}} \right) \wedge \left(\overset{1}{\bar{y}_{i-1}} \vee a_{i+1} \vee \overset{0}{y_i} \right) \\ \wedge \dots \wedge \left(\overset{1}{\bar{y}_{k-3}} \vee a_{k-1} \vee a_k \right)$$

ולכן השמה זו מספקת את C' ולכן $\langle \phi' \rangle \in 3SAT$.

כיוון \Rightarrow :

נניח כי $\langle \phi' \rangle \in 3SAT$ ותהי X' השמה המספקת את ϕ' .
נוכיח שקיימת השמה X המספקת את ϕ .

נסתכל על פסוקית $C = (a_1 \vee a_2 \vee \dots \vee a_k)$.
נניח בשלילה שלא קיימת השמה X המספקת את C . אז בהכרח

$$a_1 = a_2 = \dots = a_k = 0$$

לפי זה, באוסף פסוקיות C' שנקבל על פי ההגדרה של פונקצית הרדוקציה, $y_j = 1$ לכל $1 \leq j \leq k - 3$.
כלומר מתקיים $y_1 = y_2 = \dots = y_{k-3} = 0$. לכן

$$C' = \left(\overset{0}{a_1} \vee \overset{0}{a_2} \vee \overset{1}{y_1} \right) \wedge \left(\overset{0}{\bar{y}_1} \vee \overset{0}{a_3} \vee \overset{1}{y_2} \right) \wedge \dots \wedge \left(\overset{0}{\bar{y}_{i-2}} \vee \overset{0}{a_i} \vee \overset{1}{y_{i-1}} \right) \wedge \dots \wedge \left(\overset{0}{\bar{y}_{k-3}} \vee \overset{0}{a_{k-1}} \vee \overset{0}{a_k} \right)$$

הפסוקית האחרונה $\left(\overset{0}{\bar{y}_{k-3}} \vee \overset{0}{a_{k-1}} \vee \overset{0}{a_k} \right)$ אינה מסופקת.
לכן C' אינה מסופקת, בסתירה לכך ש- X' מספקת את C' .

ולכן $\langle \phi \rangle \in SAT$.

הוכחנו שקיימת הרדוקציה $SAT \leq 3SAT$.

כעת נוכיח כי הרדוקציה הזו היא זמן פולינומיאלית.

סיבוכיות

החישוב של הפונקציה מתבצע בזמן פולינומיאלי. ספציפי, אם האורך של הנוסחה ϕ הוא $n = |\phi|$ אז הרדוקציה היא $O(n)$.



10.7 הוכחת משפט קוק לוין*

משפט 10.7 משפט קוק לויין

הבעיית SAT היא NP - שלמה.

הוכחה:

חשיפה מלאה: ההוכחה הבאה מתבססת על ההוכחה שנתונה בהספר של Sipser.

על פי הגדרה 10.2 יש להוכיח ששני התנאים הבאים מתקיימים:

תנאי 1: $SAT \in NP$.תנאי 2: $A \leq_p SAT$ לכל $A \in NP$.ראשית נוכיח כי $SAT \in NP$:

כדי להוכיח כי SAT שייכת ל-NP, נוכיח כי אישור המורכב מהשמה מספקת עבור נוסחת קלט ϕ ניתן לאימות בזמן פולינומיאלי.

נניח כי $|\phi| = n$. כלומר ב- ϕ מופיעים n ליטרלים. ז"א השמה כלשהי דורשת n משתני בוליאניים לכל היותר.

- אלגוריתם האימות מחליף כל משתנה בנוסחה בערך המתאים לו על פי ההשמה. השלב הזה הוא $O(n)$.

- אחר כך האלגוריתם מחשב את ערכו של הביטוי:

* נניח כי הנוסחה ϕ מכילה k דורות של סוגריים בתוך סוגריים.

* החישוב מתחיל עם החישובים של הביטויים בתוך הסוגריים הכי בפנים.

* יש n סוגריים הכי-בפנים לכל היותר, וכל אחד של הסוגריים האלה מכיל n ליטרלים לכל היותר. לכן החישוב הזה הוא $O(n^2)$.

* יש k דורות של סוגריים לכן החישוב כולו הוא $O(kn^2)$

- בסה"כ הסיבוכיות זמן הריצה היא

$$O(n) + O(kn^2) = O(n^2)$$

לפיכך אישור של השמה כלשהי מתבצע בזמן פולינומיאלי.

- אם ערכו של הביטוי הוא 1 הנוסחה ספיקה.

הוכחנו כי $SAT \in NP$. עכשיו נוכיח כי $A \leq_p SAT$.

תהי N מ"ט אי-דטרמיניסטית זמן-פולינומיאלית שמכריעה שפה A כלשהי בזמן $O(n^k)$ עבור k טבעי. התרשים למטה מראה טבלה של קונפיגורציות של N . ברשימה הבאה רשומות ההגדרות של הטבלה:

- כל שורה מראה את תוכן הסרט בשלב מסוים של מסלול אחד של N .
- בשורה הראשונה יש את הקונפיגורציה ההתחלתית.
- אנחנו מניחים כי האורך של המילה, כלומר אורך הקלט הוא n .
- הסימנים w_1, \dots, w_n מסמנים את התווים של הקלט.

- בתא הראשון בכל שורה יש $\#$, ואחר כך רשומה הקונפיגורציה של N .
בסוף הקונפיגורציה בכל שורה יש $\#$.
- אחרי ה- $\#$ בקצה הימין של המילה, בכל תא יש תו רווח עד הסוף של השורה.
התווי רווח לפני המשבצת הראשונה של הקונפיגורציה לא מופיעים בטבלה.
- האורך של כל שורה הוא בדיוק n^k תאים.
- בטבלה יש בדיוק n^k שורות לסיבה הבאה:
 - המכונת טיורינג מבצעת n^k צעדים לכל היותר.
 - בכל צעד המ"ט עוברת לקונפיגורציה חדשה.
 - בכל שורה רשומה קונפיגורציה.
 - בסה"כ יש n^k שורות עבור ה- n^k קונפיגורציות שונות האפשריות.

#	q_0	w_1	w_2	\dots	w_n	\sqsubset	\dots	\sqsubset	#						
#	q_0								#						
#	q_0								#						
				<table><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>											
#									#						

אנחנו אומרים כי טבלה שלהי היא **טבלה המקבלת** אם באחת השורות יש קונפיגורציה אשר N מקבלת אותה.

בעזרת הטבלה נתאר את הרדוקציה זמן-פולינומיאלית f משפה A כלשהי ל- SAT .

הפונקציות הרדוקציה

f מקבלת קלט w ומחזירה נוסחה $\phi = f(w)$, אשר לפי ההגדרה של פונקצית הרדוקציה, עומדת בתנאי הבא:

$$w \in A \quad \Leftrightarrow \quad f(w) \in SAT \text{ .}$$

יהיו Q קבוצת המצבים ו- Γ האלפיבית של הסרט של N . נגדיר

$$C = Q \cup \Gamma \cup \{\#\} \text{ .}$$

נסמן ב- s איבר כלשהו של C .

עבור כל תא ה- (i, j) של הטבלת הקונפיגורציות נגדיר משתנה בוליאני $x_{i,j,s}$ לכל $1 \leq i, j \leq n^k$. המשתנה $x_{i,j,s}$ מוגדר על פי התנאי

$$x_{ijs} = 1$$

אם בתא ה- ij של הטבלה יש $s \in C$. למשל, אם בתא ה- $(2, 5)$ של הטבלה מופיע התו a אז

$$x_{2,5,a} = 1$$

בעוד

$$x_{2,5,b} = 0 .$$

במובן הזה, התכנים של כל התאים של הטבלה מסומנים על ידי המשתנים של ϕ .

עכשיו נבנה נוסחה ϕ על סמך התנאי שהשמה מספקת של ϕ תהיה מתאימה לטבלה המקבלת של N . נגדיר

$$\phi = \phi_{\text{cell}} \wedge \phi_{\text{start}} \wedge \phi_{\text{move}} \wedge \phi_{\text{acc}} . \quad (10.1)$$

אנחנו נסביר את כל הנוסחאות ϕ_{cell} , ϕ_{start} , ϕ_{move} ו- ϕ_{acc} אחד אחד למטה.

• הנוסחה ϕ_{cell}

כפי שמצויין לעיל, אם המשתנה $x_{i,j,s}$ "דולק", כלומר אם $x_{i,j,s} = 1$, זאת אומרת שיש סימן s בתא ה- ij של הטבלה. אנחנו רוצים להבטיח שהשמה כלשהי בנוסחה אשר מתאימה לקונפיגורציה של הטבלה, מדליקה בדיוק משתנה אחד לכל תא של הטבלה. למטרה זו נגדיר ϕ_{cell} כך:

$$\phi_{\text{cell}} = \bigwedge_{1 \leq i, j \leq n^k} \left[\left(\bigvee_{s \in C} x_{i,j,s} \right) \wedge \left(\bigwedge_{\substack{s, t \in C \\ s \neq t}} (\bar{x}_{i,j,s} \vee \bar{x}_{i,j,t}) \right) \right] \quad (10.2)$$

* האיבר הראשון בסוגריים מרובעים, $\bigvee_{s \in C} x_{i,j,s}$ מבטיח שלכל תא של הטבלה, לפחות משתנה אחד דולק.

* האיבר השני $\bigwedge_{\substack{s, t \in C \\ s \neq t}} (\bar{x}_{i,j,s} \vee \bar{x}_{i,j,t})$ מבטיח שעבור כל תא של הטבלה, משתנה אחד לכל היותר דולק.

לפיכך כל השמה מספקת עומדת בתנאי שיהיה בדיוק סימן אחד, s , בכל תא של הטבלה.

• הנוסחה ϕ_{start}

נוסחה ϕ_{start} מבטיחה ששורה הראשונה של הטבלה היא הקונפיגורציה ההתחלתית של N על הקלט w :

$$\begin{aligned} \phi_{\text{start}} = & x_{1,1,\#} \wedge x_{1,2,q_0} \wedge x_{1,3,w_1} \wedge x_{1,4,w_2} \\ & \wedge \dots \wedge \\ & x_{1,n+2,w_n} \wedge x_{1,n+3,\sqcup} \wedge \dots \wedge x_{1,n^k-1,\sqcup} \wedge x_{1,n^k,\#} \end{aligned} \quad (10.3)$$

• הנוסחה ϕ_{acc}

הנוסחה ϕ_{acc} מבטיחה שקיימת טבלה קונפיגורציה אשר המ"ט N מקבלת אותה.

בפרט ϕ_{acc} מבטיחה שהסימן q_{acc} מופיע בתא אחד של הטבלה דרך התנאי שלפחות אחד המשתנים $x_{i,j,q_{\text{acc}}}$ דולק:

$$\phi_{\text{acc}} = \bigvee_{1 \leq i, j \leq n^k} x_{i,j,q_{\text{acc}}} \quad (10.4)$$

• הנוסחה ϕ_{move}

הנוסחה ϕ_{move} מבטיחה שכל שורה של הטבלה היא "שורה חוקית".

כלומר בכל שורה, הקונפיגורציה היא כך שאפשר להגיע אליה על ידי תזוזה חוקית של N מהקונפיגורציה הקודמת שמופיעה בשורה אחת למעלה.

תזוזה חוקית בין כל שתי קונפיגורציות נקבעת על ידי הפונקציה המעברים של המ"ט N .

בשפה פורמלית, אם c_i הקונפיגורציה של שורה i , ו- c_{i+1} הקונפיגורציה של השורה $i + 1$ אחת למטה, אז ϕ_{move} מבטיחה כי לכל $1 \leq i \leq n^k - 1$ מתקיים

$$c_i \vdash_N c_{i+1}.$$

במונחי הטבלה, אפשר להגדיר תזוזה חוקית בין כל שתי שורות על ידי תת-טבלה מסדר 2×3 שמכילה 3 תאים מתאימים של שתי שורות שכנות.

מכאן ואילך אנחנו נקרא לתת-טבלה כזאת "חלון".

למטה יש דוגמאות של חלונות חוקיים:

a	q_1	b
q_2	a	c

a	q_1	b
a	a	q_2

a	a	q_1
a	a	b

#	b	a
#	b	a

a	b	a
a	b	q_2

b	b	b
c	b	b

החלונות האלה למטה הם דוגמאות לחלונות לא חוקיים:

a	b	a
a	a	a

a	q_1	b
q_1	a	a

b	q_1	b
q_2	b	q_2

הנוסחה ϕ_{move} קובעת שכל חלון של הטבלה חוקי. בפרט, כל חלון מכיל 6 תאים. לכן ϕ_{move} קובעת שהתכנים של ה-6 תאים של כל חלון מהווה חלון חוקי. ז"א

$$\phi_{\text{move}} = \bigwedge_{\substack{1 \leq i \leq n^k \\ 1 \leq j \leq n^k}} (\text{חלון ה- } i, j \text{ חוקי}) \quad (10.5)$$

אנחנו מציבים בטקסט "חלון ה- i, j חוקי" את הנוסחה הבאה, כאשר a_1, \dots, a_6 מסמנים את התכנים של ה-6 תאים של כל חלון:

$$\bigvee_{\substack{\{a_1, a_2, a_3, a_4, a_5, a_6\} \\ \text{חלון חוקי}}} (x_{i,j-1,a_1} \wedge x_{i,j,a_2} \wedge x_{i,j+1,a_3} \wedge x_{i+1,j-1,a_4} \wedge x_{i+1,j,a_5} \wedge x_{i+1,j+1,a_6}) \quad (10.6)$$

עד כה הוכחנו שקיים רדוקציה מכל שפה $A \in NP$ ל- SAT . כעת נוכיח כי הרדוקציה זו חישובית בזמן-פולינומיאלי.

הטבלה של N היא מסדר $n^k \times n^k$ ולכן היא מכילה n^{2k} תאים.

נחשב את הסיבוכיות של כל הנוסחאות $\phi_{\text{move}}, \phi_{\text{acc}}, \phi_{\text{start}}, \phi_{\text{cell}}$.

- הנוסחה ϕ_{cell}

הנוסחה (10.2) של ϕ_{cell} מכילה n^{2k} נוסחאות עם 3 ליטרלים. לכן

$$\phi_{\text{cell}} = O(n^{2k}) .$$

- הנוסחה ϕ_{start}

הנוסחה (10.3) של ϕ_{start} מכילה בדיוק n^k ליטרלים. לכן

$$\phi_{\text{start}} = O(n^k) .$$

- הנוסחה ϕ_{acc}

הנוסחה (10.4) של ϕ_{acc} מכילה בדיוק n^k ליטרלים. לכן

$$\phi_{\text{acc}} = O(n^k) .$$

- הנוסחה ϕ_{move}

הנוסחה (10.5,10.6) של ϕ_{move} מכילה n^{2k} נוסחאות עם 6 ליטרלים. לכן

$$\phi_{\text{move}} = O(n^{2k}) .$$

לכן בסה"כ

$$\phi = O(n^{2k}) + O(n^k) + O(n^k) + O(n^{2k}) = O(n^{2k}) .$$

לפיכך קיימת רדוקציה חישובית בזמן פולינומיאלי מכל שפה $A \in NP$ ל- SAT .

