אביב 2002

מיתוג ומערכות ספרתיות
קודים

1 מבוא

<u>הגדרה</u> : יהיו A , B קבוצות סופיות. קידוד של אברי קבוצה A ע'י קבוצה B היא פונקציה חח'ע $f : A \to B$.
הקידוד של התו $a \in A$ הוא $f(a) \in B$, הקידוד של הרצף $a_1 a_2 \ldots a_n$ הוא $f(a_1)f(a_2)\ldots f(a_n)$.
<u>הערות</u>

1. תנאי הכרחי שניתן יהיה לקודד את A בעזרת B הוא ש-$|A| \leq |B|$.
2. ישנן הגדרות שבהן לא דורשים כי לכל מילה ב-A תתאים מילה יחידה ב-B, במקרה זה ההגדרה תשתנה מעט ונדרוש כי קיום של פונקציה $f : A \to 2^B$ המקיימת שלכל $a_1 \neq a_2$ יתקיים $f(a_1) \cap f(a_2) = \emptyset$. כלומר, שלכל איבר a ב- A אותו רוצים לקודד תתאים תת-קבוצה של איברים ב-B ובלבד ששתי תתי-קבוצות כאלו, המתאימות לאיברים שונים, לא תחתכנה.
3. נסמן $\mathcal{C} = f(A) = \{f(a) : a \in A\}$, זהו ספר הקוד.

אנו נעסוק בקודים בינריים בהם $B = \{0,1\}^n$ עבור $n \geq 1$ ושלם.
<u>הגדרה</u> : הקצב ($Rate$) של קוד בינרי הוא $R = n^{-1}\log_2(|A|)$, היחס בין מספר הביטים המינימלי הנדרש ליצג את אברי A ובין מספר הביטים שבהם הקוד מקודד את A. הקצב מודד עד כמה הקוד בזבזני.
<u>טענה</u> : $R \leq 1$.
<u>הוכחה</u> : בקודים בינריים $|B| = 2^n$ ולכן מחח'ע מתקבל $|A| \leq 2^n$. ניקח $\log$ על שני האגפים ונקבל $\log(|A|) \leq n$ ומכאן $R = n^{-1}\log_2(|A|) \leq 1$.

<u>שימושים של קודים</u> :

1. יצוג אוביקטים. לדוג' יצוג ספרות עשרוניות, אותיות מסרים וכו'.
2. גילוי ותיקון שגיאות.
3. דחיסה.
4. הצפנה.

2 יצוג

2.1 קידוד ספרות

נרצה לקודד ספרות עשרוניות בעזרת ביטים. מכיון שבעזרת שלושה ביטים ניתן ליצג עד שמונה אוביקטים יש צורך בארבעה ביטים לפחות. במקרה זה $R = 4^{-1}\log_2 10 \approx 0.83$.

הגדרה : קוד ממושקל קוד שבו לכל ביט יש משקל קבוע $w_3w_2w_1w_0$, וצרוף הביטים $b_3b_2b_1b_0$ מקודד את המספר $x = \sum_{i=0}^{3} b_i w_i$.
הגדרה : קוד משלים לעצמו קוד של ספרות עשרוניות שבו המשלים ל-9 של כל ספרה מקודד ע'י המשלים ל-1 של היצוג שלה. כלומר היצוג של המשלים ל-9 של ספרה מסוימת מתקבל ע'י היפוך הביטים ביצוג של הספרה.
טענה : תנאי הכרחי שקוד משוקלל ישלים לעצמו הוא שסכום משקלו יהיה 9.
דוגמאות :

- קוד Binary Coded Decimal (BCD) בקוד זה מקודדים את הספרות ע'י היצוג שלהם בבסיס 2, זהו למעשה קוד עם המשקולות 1 , 2 , 4 , 8 ואינו משלים לעצמו.
- קוד Excess-3 מתקבל מקוד BCD ע'י הוספת 3(0011) לכל מילת קוד.

טבלאות הקודים :

no.	BCD	Excess-3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

הגדרה : קוד ציקלי הוא קוד שבו מילת קוד של ספרה כלשהיא שונה ממילת הקוד של הספרה הקודמת והעוקבת בביט אחד בדיוק (לקודים אלו קצב 1).
דוגמא :
קוד עם $n = 3$ ביטים :

b	g
0	000
1	001
2	011
3	010
4	110
5	111
6	101
7	100
0	000

הגדרה : קוד גריי הוא קוד ציקלי המוגדר באופן הבא. תהי $b_nb_{n-1}\ldots b_1b_0$ מילה בינרית אותה רוצים לקודד ($b_i \in \{0,1\}$), נגדיר את המילה המקודדת אותה $g_ng_{n-1}\ldots g_1g_0$ בקוד גריי באופן הבא :

$$\begin{aligned} g_n &= b_n \\ g_i &= b_i \oplus b_{i+1} = (b_i + b_{i+1}) \bmod 2 \qquad \forall i = 0\ldots n-1 \end{aligned}$$

<u>טענה</u> : קוד גרי הוא חח'ע, כלומר מילים שונות מקודדות באופן שונה.
<u>הוכחה</u> : אם שתי מילים שונות לראשונה (כאשר עוברים על הביטים לפי סדר אינדקסים יורד) בביט ה-i, אזי ע'פ הגדרה הן יהיו שונות בביט ה-i בקידוד שלהן.
<u>מסקנה</u> : קוד גרי הוא על.
<u>הוכחה</u> : מספר הביטים הנדרש לקוד זהה למספר הביטים של המילים המקודדות, ולכן מספר המילים בקוד זהה למספר המילים המקודדות (=2 בחזקת מספר הביטים). מכיון שהקוד הוא העתקה חח'ע בין קבוצות זהות מתקבל הדרוש.

<u>דוגמא</u> : נקודד את המילה $b_4b_3b_2b_1b_0 = 10110$. ע'פ הגדרה :

$$\begin{array}{lclclcl} g_4 & = & b_4 & & & = & 1 \\ g_3 & = & b_3 \oplus b_4 & = & 0 \oplus 1 & = & 1 \\ g_2 & = & b_2 \oplus b_3 & = & 1 \oplus 0 & = & 1 \\ g_1 & = & b_1 \oplus b_2 & = & 1 \oplus 1 & = & 0 \\ g_0 & = & b_0 \oplus b_1 & = & 0 \oplus 1 & = & 1 \end{array}$$

ולכן $g(10110) = 11101$
<u>טענה</u> : אלגוריתם פענוח.
תהי $g_ng_{n-1}\dots g_1g_0$ מילה בקוד גרי, אזי הפענוח שלה (המילה $b_nb_{n-1}\dots b_1b_0$ שקודדה) נתון באופן הבא. אם מספר ביטי ה-1 שמשמאל לביט ה-i זוגי אזי $b_i = g_i$, אחרת $b_i = g'_i$.
<u>דוגמא</u> : נפענח את המילה $g_4g_3g_2g_1g_0 = 11101$. בשורה הראשונה רשומה המילה שרוצים לפענח, בשורה השניה רשומים מספר ביטי ה-1 שמשמאל לכל ביט, ובשורה השלישית ביט הזוגיות של מספר זה (0 - זוגי, 1 - אי זוגי), בשורה האחרונה רשום הביט המפוענח b_i שהינו XOR של הביטים המתאימים בשורות הראשונה והשלישית.

$$\begin{array}{ccccc} 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 2 & 3 & 3 \\ 0 & 1 & 0 & 1 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 \end{array}$$

ולכן $g^{-1}(11101) = 10110$
<u>הערה</u> : חשוב להבדיל בין המרת מספר עשרוני לבינרי לבין קידודו. לדוגמא, המרת המספר 15 מבסיס 10 לבסיס 2 תתן $(1111)_2 = (15)_{10}$ בעוד שקידוד מספר זה בשיטת BCD תתן 0001 0101, ואילו המרת הרצף 00010101 מבסיס 2 לבסיס 10 תתן $1 + 4 + 16 = 21$.

2.2 קידוד אלפאנומרי

ישנם מספר קודים מוכרים ליצוג אותיות וסימנים שונים בינהם :

- ASCII - American Standard Code for Information Interchange
  קוד בן שבע ביטים (ועוד ביט זוגיות לאיתור שגיאות).
- EBCDIC - External BCD Interchange Code
  קוד בן שמונה ביטים.

3 איתור ותיקון שגיאות

כאשר מעבירים מידע בינרי בקו תקשורת (בטלפון, שידור אלחוטי, בתוך המחשב וכו') עלול להווצר מצב שבו המידע שנקלט שונה מן המידע ששודר (עקב תקלה או חבלה). נתאר מספר קודים שמטרתם להתגבר על בעיה זו. לקודים אלו שתי מטרות :

1. גילוי/איתור השגיאות - לקבוע האם ישנם ביטים שהשתבשו ברצף.
2. תיקון השגיאות - לקבוע אילו ביטים השתבשו ברצף, ומה היה ערכם קודם השיבוש.

כל הקודים לאיתור ותיקון שגיאות עושים שימוש ביתירות של הביטים, כלומר משדרים ביטים נוספים על ביטי האינפורמציה. ביטים אילו מאפשרים לקבוע, בתנאים מסוימים, האם והיכן היתה שגיאה.

3.1 קוד חזרות עם k ביטים

הקוד מקודד ביטים בודדים, כל ביט משודר k פעמים. הפענוח נעשה ע'י הצבעת רוב בין הביטים שהתקבלו. נקח לדוגמא $k = 4$. במקרה זה הקידוד הינו $1 \to 1111$ $\quad 0 \to 0000$. קידוד הרצף 0101 הינו 0000111100001111 נניח כי ארעו תקלות והתקבל הרצף 0001101010110000. נחלק את הרצף לרביעיות ונקח את רוב הביטים בכל רבעיה נקבל :

0000	1111	0000	1111
0001	1010	1011	0000
0	?	1	0

ברביעיה הראשונה קיבלנו 0001 ויש רוב של 3 אפסים מול 1 אחדים, ולכן הכרענו ששודר הביט 0. ברביעיה השניה קיבלנו 1010 ויש שני אחדים ושני אפסים, אי אפשר להכריע מה שודר אולם אנו יודעים שמשהו משובש (כיון שאין מילה עם שני אחדים ושני אפסים). ברביעיה השלישית קיבלנו 1011 ויש רוב של 1 אפסים מול 3 אחדים, ולכן הכרענו ששודר הביט 1 למרות שבפועל שודר הביט 0. ברביעיה הרביעית קיבלנו 0000 ויש רוב של 4 אפסים מול 0 אחדים, ולכן הכרענו ששודר הביט 0, למרות שבפועל שודר הביט 1. נסכם :

- ניתן לאתר עד 3 שגיאות (אם כל הביטים התהפכו לא נוכל לאתר את השגיאה, כי ההנחה שאין טעויות כלל סבירה יותר מן ההנחה שבכל הביטים נפלה טעות).
- ניתן לתקן עד שגיאה אחת (משום שאם לפחות מחצית הביטים ישתבשו אזי הכרעת הרוב תשתבש אף היא).

נכליל את הטענות האחרונות לקוד חזרות כללי בגודל k.
<u>משפט</u> : קוד חזרות שבו כל ביט משודר k פעמים מקיים :

- ניתן לאתר $k-1$ שגיאות (משום שאם יש k שגיאות נעבור למילה חוקית).
- ניתן לתקן עד $\lfloor \frac{k-1}{2} \rfloor$ שגיאות (משום שאם לפחות מחצית הביטים ישתבשו אזי הכרעת הרוב תשתבש אף היא).

<u>טענה</u> : הקצב של קוד חזרות שבו כל ביט משודר k פעמים הינו $R = k^{-1}\log_2(2) = \frac{1}{k}$.

3.2 הכללת משפט איתור ותיקון השגיאות

נכליל את המשפט האחרון עבור קודים בינריים כלשהם, לשם כך יש צורך במספר מושגים.
<u>הגדרה</u> : יהיו $a = a_n a_{n-1} \dots a_0$, $b = b_n b_{n-1} \dots b_0$ מילים בקוד בינרי $a_i, b_i \in \{0,1\}$ נגדיר את

מרחק המינג בין המילים להיות מספר הביטים השונות בין המילים. פורמלית

$$\begin{aligned} d_H(a,b) &= |\{i : a_i \neq b_i\}| \\ &= \#\{i : a_i \neq b_i\} \end{aligned}$$

<u>טענות - אקסיומות המרחק</u> :

1. $d_H(a,a) = 0$.
2. $d_H(a,b) = d_H(b,a)$.
3. $d_H(a,b) + d_H(b,c) \geq d_H(a,c)$.

<u>דוגמאות</u> :

1. $d_H(0011, 0101) = 0 + 1 + 1 + 0 = 2$.
2. $d_H(010101, 010010) = 0 + 0 + 0 + 1 + 1 + 1 = 3$.

<u>הגדרה</u> : יהי $\mathcal{C}$ ספר קוד בינרי. נגדיר את המרחק של הקוד להיות

$$\rho(\mathcal{C}) = \min_{c_1 \neq c_2 \in \mathcal{C}} d_H(c_1, c_2)$$

<u>דוגמאות</u> :

1. $\mathcal{C} = \{0000, 1111\}$.
   $d_H(0000, 1101) = 4$.
   $\Rightarrow \rho(\mathcal{C}) = 4$
2. $\mathcal{C} = \{0000, 0011, 1100\}$.
   $d_H(0000, 0011) = 2 \quad , d_H(0000, 1100) = 2 \quad , d_H(0011, 1100) = 4$.
   $\Rightarrow \rho(\mathcal{C}) = 2$

נכליל כעת את המשפט שהוכחנו עבור קוד חזרות.
<u>משפט</u> : יהי $\mathcal{C}$ ספר קוד עם מרחק $\rho = \rho(\mathcal{C})$ אזי :

1. ניתן לגלות בקוד עד $\rho - 1$ שגיאות.
2. ניתן לתקן בקוד עד $\lfloor \frac{\rho-1}{2} \rfloor$ שגיאות.

כאשר בהנתן מילה x נשיך לה את המילה הקרובה לה ביותר מתוך ספר הקוד $\mathcal{C}$, דהיינו,

$$\arg\min_{c \in \mathcal{C}} d_H(c, x)$$

<u>דוגמא</u> : נתבונן בספר הקוד $\mathcal{C} = \{0000, 0011, 1100\}$.
קל לראות כי המרחק של הקוד הוא $\rho = 2$.
נניח שידרנו את המילה $c = 0011$ וקיבלנו את המילה $c' = 0111$, כדי לפענח נחשב את מרחק המינג בין המילה שקיבלנו לכל אחת מן המילים בקוד ונקבל :
$d_H(0000, 0111) = 3 \quad , d_H(0011, 0111) = 1 \quad , d_H(1100, 0111) = 3$.
קיבלנו כי יש מועמד אחד להיות המילה ששודרה וזו אכן המילה $c = 0011$. זהו פענוח נכון.
נניח כעת כי שידרנו את המילה $c = 0000$ וקיבלנו את המילה $c' = 0001$, כדי לפענח נחשב את מרחק המינג בין המילה שקיבלנו לכל אחת מן המילים בקוד ונקבל :
$d_H(0000, 0001) = 1 \quad , d_H(0011, 0001) = 1 \quad , d_H(1100, 0001) = 3$.
קיבלנו כי יש שני מועמדים למילה ששודרה ואין את היכולת להכריע - טעות.

3.3 קוד זוגיות עם k ביטים

הקוד מקודד רצפים של k ביטים. לכל רצף מוסיפים ביט אחד בדיוק באופן שבו מספר ביטי ה-1 בכל רצף באורך $k+1$ הוא זוגי. פורמלית נסמן את הרצף לקידוד ב-$a = (a_1,\ a_2,\ \ldots\ , a_k)$ אזי הביט הנוסף יהיה :

$$b = \oplus_{i=1}^{k} a_i = a_1 \oplus a_2 \oplus \ldots \oplus a_k$$

והרצף שנשלח יהיה

$$a_1,\ a_2,\ \ldots, a_k\ , b$$

<u>טענה</u> : הקצב של קוד זוגיות עם k ביטים הינו $R = k + 1^{-1}\log_2(2^k) = \frac{k}{k+1}$.
<u>טענה</u> : המרחק של קוד זוגיות עם k ביטים הינו 2.
<u>הוכחה</u> יהיו v , w שתי מילות קוד באורך $k+1$ השונות בביט אחד בדיוק מתוך k הביטים הראשונים. כיון שהמילים שונות בביט אחד בדיוק נקבל כי גם ביט הזוגיות שלהם שונה, ולכן המרחק בינהם הוא בדיוק 2, ולכן המרחק של הקוד הוא לפחות 2. אולם, כל זוג מילים שלא נכנס בקטגוריה הקודמת מקיים שהמרחק בין k הביטים הראשונים הוא לפחות 2, ולכן המרחק בין המילים הוא לפחות 2, ולכן המרחק של הקוד הוא בדיוק 2.
מכאן עולה כי ניתן לגלות לכל היותר שגיאה אחת, אך לא לתקן אפילו שגיאה אחת.

3.4 קוד המינג

נציג קידוד של ארבעה ביטים בעזרת שבעה ביטים.
נגדיר פונקציה $f : \{0,1\}^4 \to \{0,1\}^7$ באופן הבא. נסמן $x = (I_4 I_3 I_2 I_1)$ ($I_i \in \{0,1\}$). נגדיר

$$f(x) = (I_4 I_3 I_2 C_3 I_1 C_2 C_1) \qquad (1)$$

כאשר

$$\begin{aligned} C_1 &= I_1 \oplus I_2 \oplus I_4 \\ C_2 &= I_1 \oplus I_3 \oplus I_4 \\ C_3 &= I_2 \oplus I_3 \oplus I_4 \end{aligned} \qquad (2)$$

לדוגמא $f(1101) = (1100110)$ כי

$$\begin{aligned} C_1 &= 1 \oplus 0 \oplus 1 = 0 \\ C_2 &= 1 \oplus 1 \oplus 1 = 1 \\ C_3 &= 0 \oplus 1 \oplus 1 = 0 \end{aligned}$$

נשים לב כי בהנתן $f(x)$ קל לחשב את x מתוכו. נניח כי ארעה תקלה והביט I_3 (השישי מימין) בדוגמא התהפך, כלומר נתון לנו המספר

$$y' = (1000110) = (I'_4 I'_3 I'_2 C'_3 I'_1 C'_2 C'_1)$$

נראה כיצד לחשב את המספר המקורי x למרות התקלה. בהנתן מספר בן שבע ספרות $y' = (I'_4 I'_3 I'_2 C'_3 I'_1 C'_2 C'_1)$ נחשב כמקודם :

$$\begin{aligned} C''_1 &= I'_1 \oplus I'_2 \oplus I'_4 \\ C''_2 &= I'_1 \oplus I'_3 \oplus I'_4 \\ C''_3 &= I'_2 \oplus I'_3 \oplus I'_4 \end{aligned}$$

בדוגמא שאנו עוסקים בה המספר הינו (1000110) ומכאן :

$$\begin{aligned} C''_1 &= 1 \oplus 0 \oplus 1 = 0 \\ C''_2 &= 1 \oplus 0 \oplus 1 = 0 \\ C''_3 &= 0 \oplus 0 \oplus 1 = 1 \end{aligned}$$

נחשב את ה- xor של הוקטור $(C'_3C'_2C'_1) = (010)$ עם הוקטור $(C''_3C''_2C''_1) = (100)$ ונקבל $(0 \oplus 1, 1 \oplus 0, 0 \oplus 0) = (110)$. נמיר את המספר שקבלנו לבסיס עשרוני ונקבל 6, זהו בדיוק מספר הביט שהתהפך. נהפוך אותו בחזרה ומכאן נוכל לשחזר שוב את x.
נציג כעת את תהליך הקידוד והפענוח :
קידוד - בהנתן x נשדר את $y = f(x)$ כפי שנוגדר במשוואה מס' 1 .
פענוח - בהנתן y' שהתקבל. אנו מניחים כי אין בו יותר משגיאה אחת. נסמן את הערך של המספר המשובש ב-

$$y' = (I'_4I'_3I'_2C'_3I'_1C'_2C'_1) \tag{3}$$

נגדיר :

$$\begin{aligned} C''_1 &= I'_1 \oplus I'_2 \oplus I'_4 \\ C''_2 &= I'_1 \oplus I'_3 \oplus I'_4 \\ C''_3 &= I'_2 \oplus I'_3 \oplus I'_4 \end{aligned} \tag{4}$$

ולבסוף נגדיר :

$$\begin{aligned} b_1 &= C'_1 \oplus C''_1 \\ b_2 &= C'_2 \oplus C''_2 \\ b_3 &= C'_3 \oplus C''_3 \end{aligned} \tag{5}$$

אם $(b_3b_2b_1)_2 = 0$ אזי נקבע כי לא היתה שגיאה, ונשחזר את x מתוך $y = y'$. אחרת, נסמן $(n)_{10} = (b_3b_2b_1)_2$, נהפוך את הביט ה-n (מימין) ב-y' ומכאן נחלץ את $x = I_4I_3I_2I_1$.

אנו טוענים כל האלגוריתם נכון, כלומר כי :

<u>משפט</u>

1. אם $y = f(x)$ ו- $y' = y$ אזי $(b_3b_2b_1)_2 = (0)_{10}$.
2. אם $y = f(x)$ ו- y' שונה מ y בביט שמיקומו מימין שווה ל- n אזי $(b_3b_2b_1)_2 = (n)_{10}$.

כאשר $(C_3C_2C_1)$ חולץ מן הוקטור y לפי משוואה 1 , ו-$(C'_3C'_2C'_1)$ חושב מן הוקטור y' לפי משוואה 2. כלומר יש להראות כי מיקום הביט שבו ארעה השגיאה ניתן ע'י הביטוי $(C'_3 \oplus C''_3, C'_2 \oplus C''_2, C'_1 \oplus C''_1)_2$ (ואם לא ארעה שגיאה, ערך הביטוי הינו 0).

<u>הוכחה</u>

ראשית נוכיח את חלקו הראשון של המשפט. ממשואות (1) ו- (3) נובע :
$y = y' \Rightarrow (I_4I_3I_2C_3I_1C_2C_1) = (I'_4I'_3I'_2C'_3I'_1C'_2C'_1)$. ולכן ממשואות (2) ו-(4) נובע כי :

$$\begin{aligned} C''_1 &= I'_1 \oplus I'_2 \oplus I'_4 = I_1 \oplus I_2 \oplus I_4 = C_1 = C'_1 \\ C''_2 &= I'_1 \oplus I'_3 \oplus I'_4 = I_1 \oplus I_3 \oplus I_4 = C_2 = C'_2 \\ C''_3 &= I'_2 \oplus I'_3 \oplus I'_4 = I_2 \oplus I_3 \oplus I_4 = C_3 = C'_3 \end{aligned}$$

ומכאן, בצרוף משוואה (5) נובע כי $r = 1,2,3 \quad b_r = 0$, כנדרש. נוכיח כעת את חלקו השני של המשפט. מתוך ההגדרה של מעבר בין הבסיסים 2 ו-10 נובע כי חלקו השני של המשפט שקול לשלוש הטענות :

$$\begin{aligned} b_1 = 1 &\Leftrightarrow n = 1,3,5,7 \\ b_2 = 1 &\Leftrightarrow n = 3,4,6,7 \\ b_3 = 1 &\Leftrightarrow n = 4,5,6,7 \end{aligned}$$

מדוע? אם לדוגמא $b_1 = 1$ אזי אנו טוענים כי המיקום n של הביט בו ארעה השגיאה הוא מן הצורה $(i \ j \ 1)$ (אי זוגי) כלומר מקבל אחד מן הערכים $1,3,5,7$. ולהפך, אם אנו טוענים כי $n = 1,3,5,7$ אזי אנו בעצם טוענים כי הביט השמאלי ביותר ביצוג של n בבסיס 2 שווה ל-1.
נראה כעת את הראשונה מבין שלוש הטענות, הוכחת הטענות האחרות שקולה :
עלינו להראות כי $b_1 = 1 \Leftrightarrow n = 1,3,5,7$. ממשוואה (5) נובע כי $b_1 = 1 \Leftrightarrow C'_1 \neq C''_1$, וממשוואה (1) נובע כי $n = 1,3,5,7 \Leftrightarrow$ השגיאה באחד הביטים C_1, I_1, I_2, I_4. ולכן עלינו להראות כי $C'_1 \neq C''_1 \Leftrightarrow$ השגיאה באחד הביטים C_1, I_1, I_2, I_4.
רעיון ההוכחה :
$C'_1 \neq C''_1$ בשני מקרים :

1. הביט C_1 השתבש (כלומר $n = 1$). במקרה זה $C''_1 = C_1 \neq C'_1$. השויון נובע מכך שהביטים I_4, I_2, I_1 לא השתבשו.
2. אחד מן הביטים I_4, I_2, I_1 השתבש. במקרה זה $C''_1 \neq C'_1 = C_1$. משום שאם אחד מן הביטים I_4, I_2, I_1 השתבש אזי הביט C_1 לא השתבש, ולכן $C'_1 = C_1$. אי השויון מתקבל מתוך משואות (3) ו-(4).

להוכחת הטענה :
מתוך ההגדרות במשוואה (3) עולה כי הביט C_1 שגוי $\Leftrightarrow C_1 \neq C'_1 \Leftrightarrow C_1 \oplus C'_1 = 1$, חישוב זהה נכון עבור שאר הביטים. לכן, תוך שימוש בהנחה שרק אחד הביטים שגוי, נקבל כי השגיאה היא באחד הביטים $C_1 , I_1 , I_2 , I_4 \Leftrightarrow (C_1 \oplus C'_1) \oplus (I_1 \oplus I'_1) \oplus (I_2 \oplus I'_2) \oplus (I_4 \oplus I'_4) = 1$.
נפתח את הביטוי השמאלי :

$$\begin{aligned}
& (C_1 \oplus C'_1) \oplus (I_1 \oplus I'_1) \oplus (I_2 \oplus I'_2) \oplus (I_4 \oplus I'_4) \\
= \; & C_1 \oplus (I_1 \oplus I_2 \oplus I_4) \oplus C'_1 \oplus (I'_1 \oplus I'_2 \oplus I'_4) \\
= \; & C_1 \oplus C_1 \oplus C'_1 \oplus C''_1 \\
= \; & C'_1 \oplus C''_1
\end{aligned}$$

השויון הראשון נובע מאסוציטיבות וקומטטביות של פעולת ה-$\oplus$, השני מתוך משואות (2) ו-(4) והשלישי מתוך הטענה $x \oplus x = 0$.
קיבלנו לכן כי השגיאה היא באחד הביטים $C_1 , I_1 , I_2 , I_4 \Leftrightarrow C'_1 \oplus C''_1 = 1 \Leftrightarrow C'_1 \neq C''_1$, כנדרש.
בכך הוכחנו את הראשונה מבין שלוש הטענות. שתי הטענות האחרות מוכחות באותו אופן, ובכך מסתיימת הוכחת המשפט.