

ניסוי 79 ECC - דוח מסכם חלק ב'

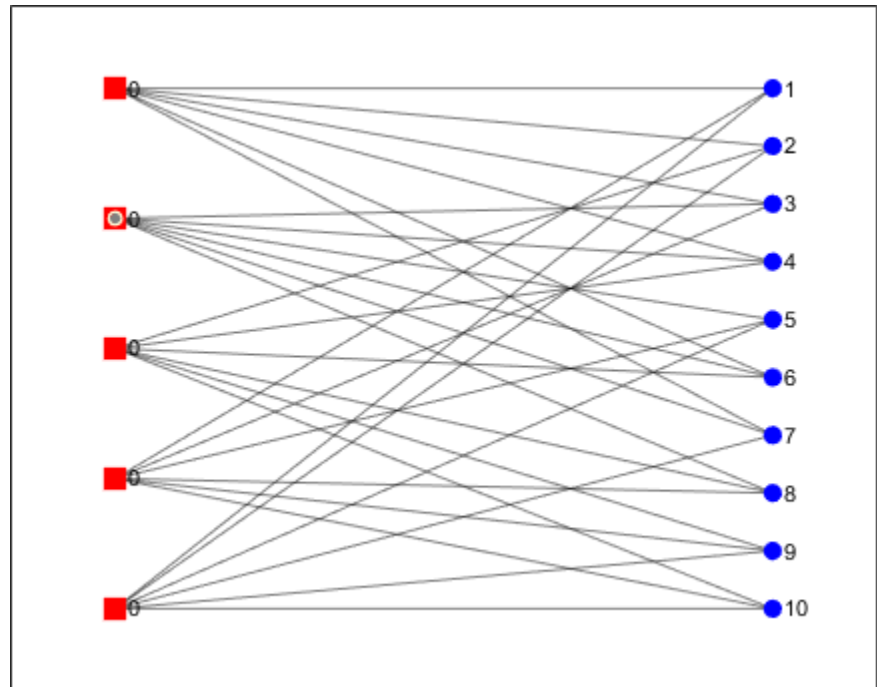
מגישים:

כריסטיאן שקור, 208157826, christian.s@campus.technion.ac.il

לארין עטאללה, 208653543, lareine.at@campus.technion.ac.il

ניסוי 5:

שאלה 1:



גרף טאנר הוא גרף המקשר בין המילות השונות של הקוד (העמודות של המטריצה הבודקת H) לבין קודי הזוגיות שלו (השורות של מטריצה H). כל צומת שמאפיין מילת קוד הוא נקרא צומת משתנה $variable\ node$ וכל צומת של בדיקת זוגיות הוא צומת הנקרה צומת בדיקה $check\ node$. הקשר בין כל $variable\ node$ לבין $check\ node$ הוא קיום קשת בגרף טאנר ביניהם אם $H(c,v)=1$ כאשר c הוא $check\ node$ והוא שורה במטריצה H ו- v הוא $variable\ node$ והוא עמודה במטריצה h .

שאלה 2:

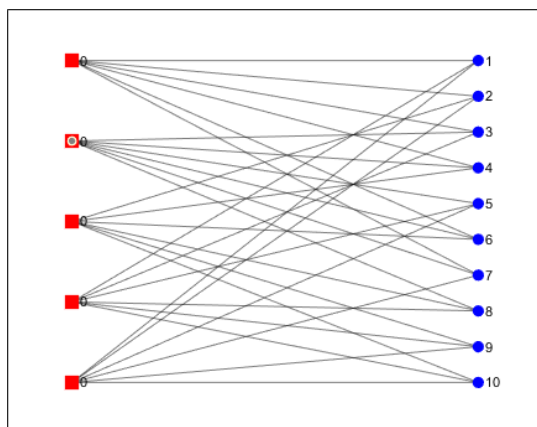
>> y

$y =$

2 1 0 1 1 1 1 2 1 0

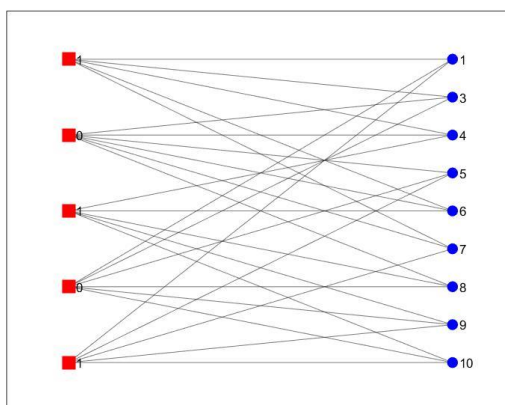
המילה עברה בתוך ערוץ BEC שהוא מוחק ביטים. מרחב הכניסה הוא $[0,1]$ ומרחב היציאה הוא $[0,1,2]$ כלומר המספר 2 מסמל ביטים אשר נמחקו. אי אפשר לנחש מה הייתה המילה קודם.

שאלה 3:



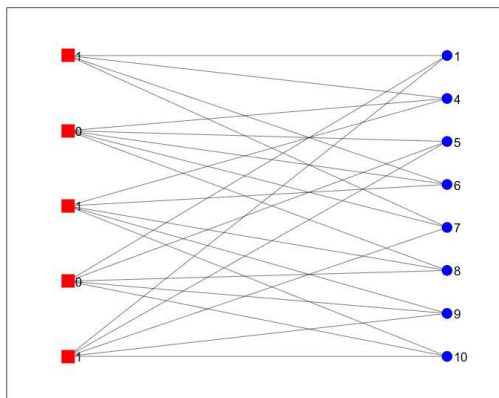
גרף טאנר 'מלא', כאשר מאתחלים את צמתי הבדיקה להיות 0. נעבור בכל שלב על צמתי המשתנה שאינם מחזקים ונבצע פעולת XOR על מנת לקבוע את צמתי בדיקה בסיום פעולת XOR נמחק את צומת המשתנה.

1:



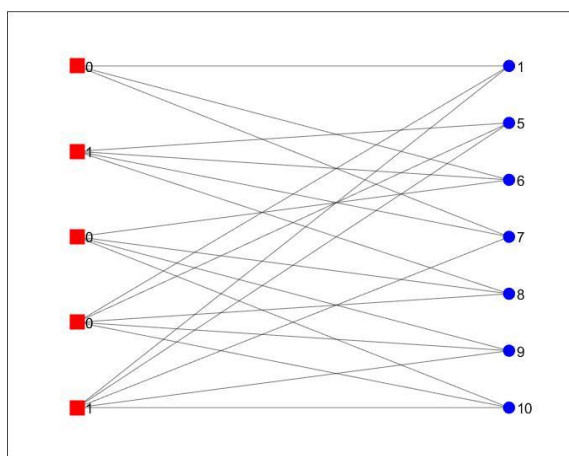
בשלב הראשון, בגרף חסר צומת משתנה 2, וזאת מפני שביצענו את פעולות XOR עם צמתי הבדיקה המחוברים לצומת 2. ניתן לראות שצמתי הבדיקה 1,3,5 עם ערך 1, לכן נסיק כי צומת משתנה 2 קושר אל צמתי הבדיקה 1,3,5 וערכו 1.

:2



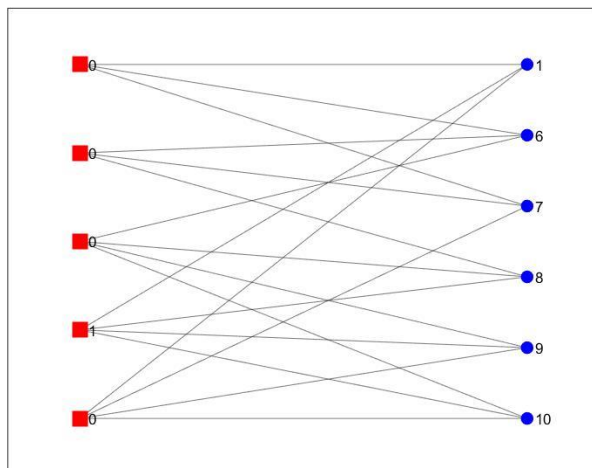
בשלב השני, בגרף חסר צומת משתנה 3. צומת זו קשורה אל צמתי הבדיקה 1,2,4 וניתן לראות כי ערכי הצמתיים הללו לא השתנו, לכן נסיק כי צומת משתנה 3 הוא עם ערך 0.

:3



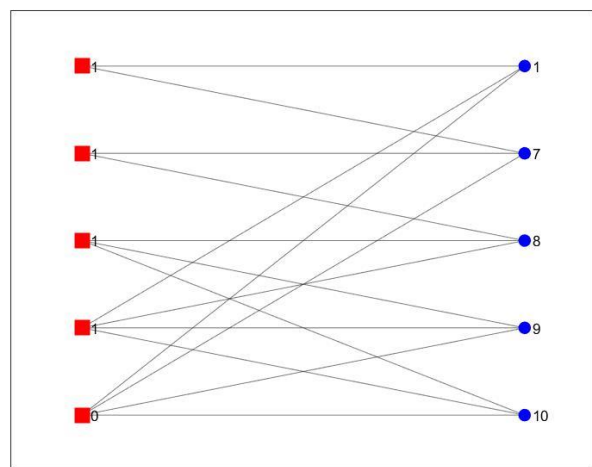
בשלב השלישי, בגרף חסר צומת משתנה 4. צומת זו קשורה אל צמתי הבדיקה 1,2,3 וניתן לראות כי ערכי הצמתיים הללו השהפכו, לכן נסיק כי צומת משתנה 4 הוא עם ערך 1.

:4



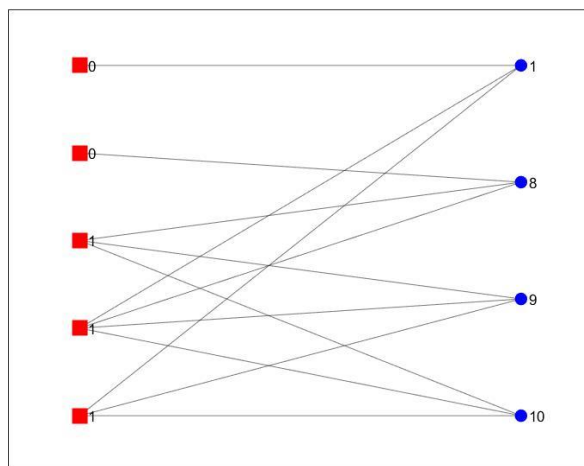
בשלב הרביעי, בגרף חסר צומת משתנה 5. צומת זו קושרה אל צמתי הבדיקה 2,4,5 וניתן לראות כי ערכי הצמתיים הללו התהפכו, לכן נסיק כי צומת משתנה 5 הוא עם ערך 1.

:5



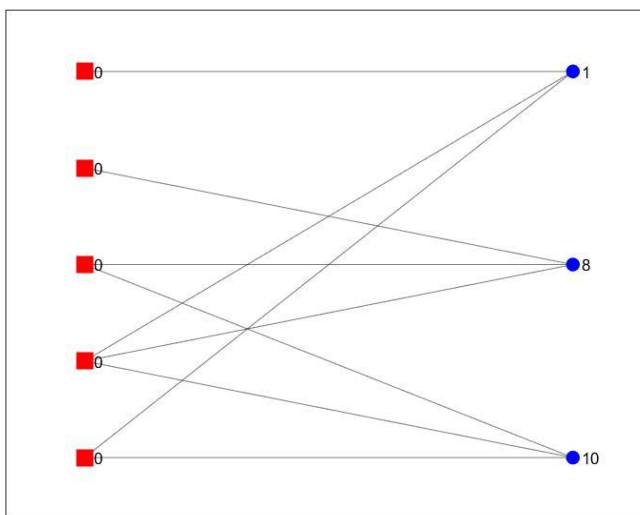
בשלב החמישי, בגרף חסר צומת משתנה 6. צומת זו קושרה אל צמתי הבדיקה 1,2,3 וניתן לראות כי ערכי הצמתיים הללו התהפכו, לכן נסיק כי צומת משתנה 6 הוא עם ערך 1.

:6



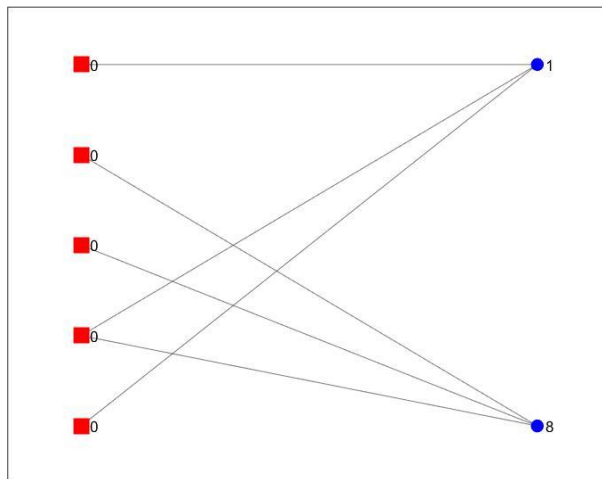
בשלב השישי, בגרף חסר צומת משתנה 7. צומת זו קושרה אל צמתי הבדיקה 1,2,5 וניתן לראות כי ערכי הצמתים הללו התהפכו, לכן נסיק כי צומת משתנה 7 הוא עם ערך 1.

:7



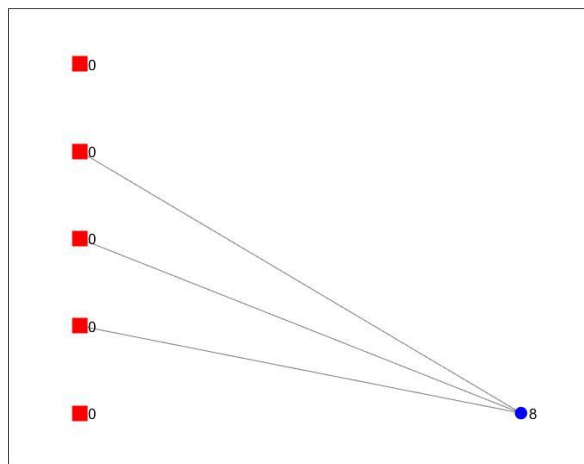
בשלב השביעי, בגרף חסר צומת משתנה 9. צומת זו קושרה אל צמתי הבדיקה 3,4,5 וניתן לראות כי ערכי הצמתים הללו התהפכו, לכן נסיק כי צומת משתנה 9 הוא עם ערך 1.

:8



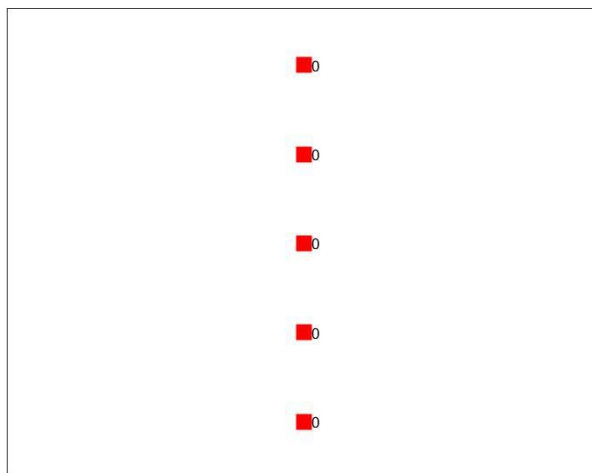
בשלב השמיני, בגרף חסר צומת משתנה 10. צומת זו קושרה אל צמתי הבדיקה 3,4,5 וניתן לראות כי ערכי הצמתיים הללו לא השתנו, לכן נסיק כי צומת משתנה 10 הוא עם ערך 0.

:9



כעת השתמש בערכים של צמתי הבדיקה על מנת להעריך את צמתי המשתנה 1,8 (המחוקים) התחלנו מצומת משתנה 1, נסיק כי ערכו הוא 0 על מנת לקבל 0 בפעולת XOR עם צמתי הבדיקה 1,5

10:



נמשיך עם צומת משתנה 8, נסיק כי ערכו הוא 0 על מנת לקבל 0 בפעולת XOR עם צמתי הבדיקה 2,3,4

שאלה 4:

```
>> cDecML
```

```
cDecML =
```

```
0 1 0 1 1 1 1 0 1 0
```

```
>> cDecPeeling
```

```
cDecPeeling =
```

```
0 1 0 1 1 1 1 0 1 0
```

קיבלנו תוצאות זהות והפענוח אכן הצליח.

אלגוריתם ML הוא אלגוריתם מאוד מובטח כי הוא לוקח בחשבון גם התלויות בין המילים השונות ומפענח בצורה יותר מובטחת. אבל, עם סיבוכיות זמן חישוב ארוכה יותר וחישובים יותר מסובכים (בודק תלויות וכו'). לעומת זאת, באלגוריתם ה-peeling החישובים יותר קלים עם סיבוכיות זמן חישוב סבירה יותר (פולינומית) אבל לא תמיד מובטח למצוא פענוח (כי לפעמים צריך להסתכל על התלויות בין המילים כדי להבחין).

שאלה 5:

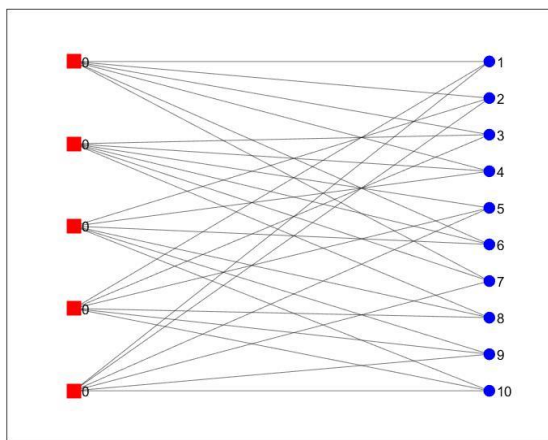
האות המורעש:

$y =$

0 2 0 0 0 2 2 1 1 0

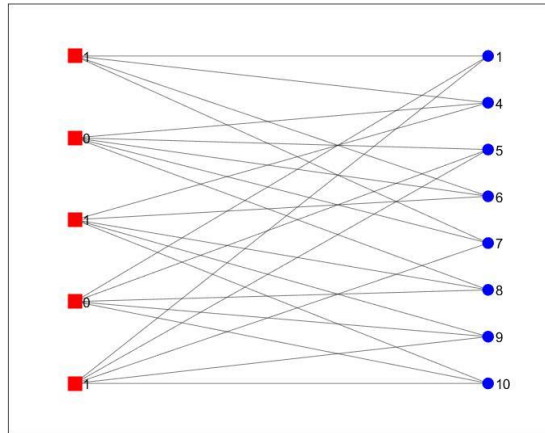
הביטים המורעשים באינדקסים 2,6,7. לא ניתן לנחש את ערכם

:1



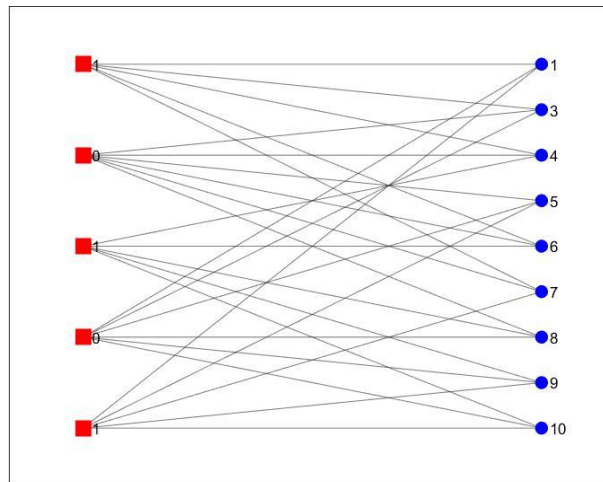
בשלב הראשון, בגרף חסר צומת משתנה 1, וזאת מפני שביצענו את פעולות XOR עם צמתי הבדיקה שמחוברים אל צומת 1. ניתן לראות שכל צמתי הבדיקה נותרו עם ערך 0 ולכן נסיק צומת משתנה 1 עם ערך 0

:2



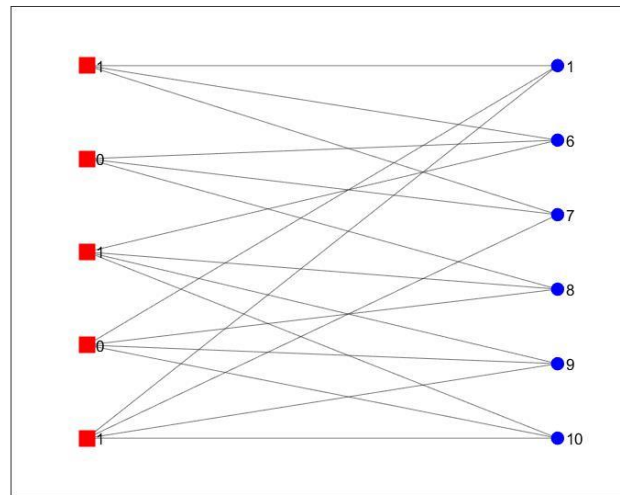
בשלב השני, בגרף חסר צומת משתנה 3. צומת זו קושרה אל צמתי הבדיקה 1,2,4 וניתן לראות כי ערכי הצמתיים הללו לא השתנו, לכן נסיק כי צומת משתנה 3 הוא עם ערך 0.

:3



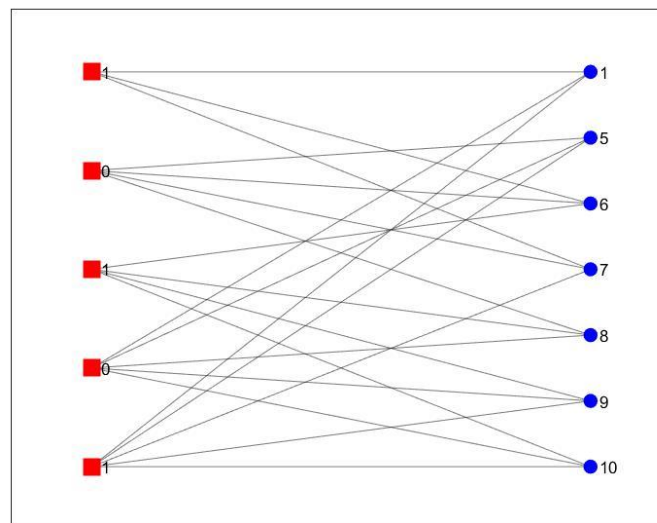
בשלב השלישי, בגרף חסר צומת משתנה 4. צומת זו קושרה אל צמתי הבדיקה 1,2,3 וניתן לראות כי ערכי הצמתיים הללו התהפכו, לכן נסיק כי צומת משתנה 4 הוא עם ערך 1.

:4



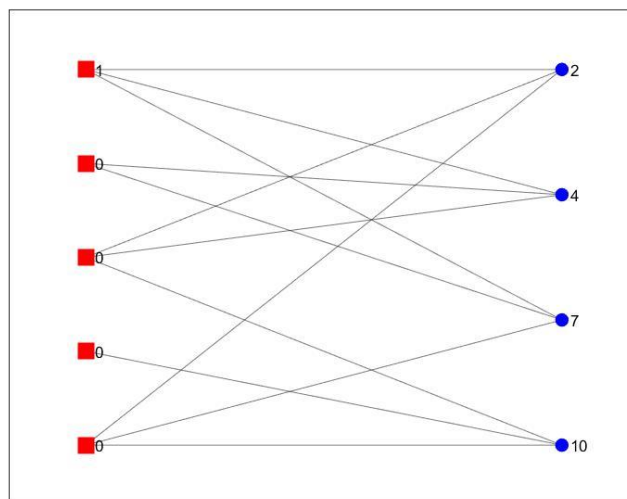
בשלב הרביעי, בגרף חסר צומת משתנה 5. צומת זו קושרה אל צמתי הבדיקה 2,4,5 וניתן לראות כי ערכי הצמתיים הללו התהפכו, לכן נסיק כי צומת משתנה 5 הוא עם ערך 1.

:5



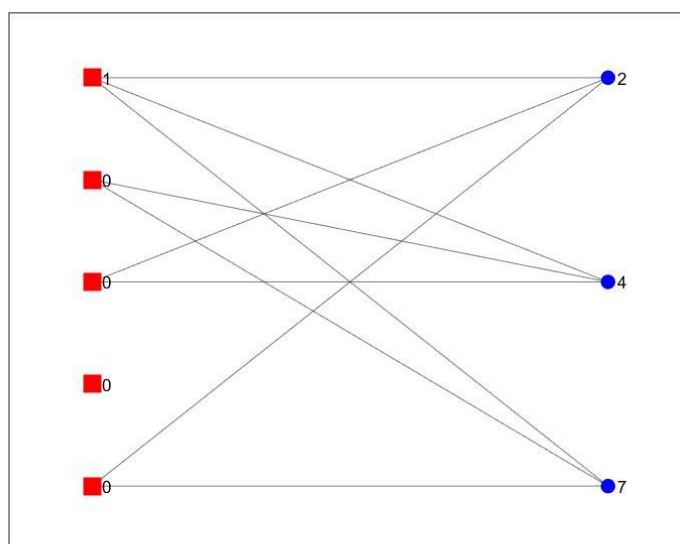
בשלב החמישי, בגרף חסר צומת משתנה 8. צומת זו קושרה אל צמתי הבדיקה 2,3,4 וניתן לראות כי ערכי הצמתיים הללו התהפכו, לכן נסיק כי צומת משתנה 8 הוא עם ערך 1.

:6



בשלב השישי, בגרף חסר צומת משתנה 9. צומת זו קושרה אל צמתי הבדיקה 3,4,5 וניתן לראות כי ערכי הצמתים הללו התהפכו, לכן נסיק כי צומת משתנה 9 הוא עם ערך 1.

7:



בשלב השביעי, בגרף חסר צומת משתנה 10. צומת זו קושרה אל צמתי הבדיקה 3,4,5 וניתן לראות כי ערכי הצמתים הללו לא השתנו, לכן נסיק כי צומת משתנה 10 הוא עם ערך 0.

מכיוון שקיבלנו מצב בו כל צמתי הבדיקה הם מדרגה 2 או יותר, לא ניתן לפענח את ערכי צמתי המשתנה באמצעות האלגוריתם.

cDecML	[0,1,0,0,0,1,0,1,1,0]
cDecPeeling	[0,2,0,0,0,2,2,1,1,0]
erasureLocations	[0,1,0,0,0,1,1,0,0,0]
Fig	7
FigList	18x1 Figure
G	5x10 double
H	5x10 double
k	5
n	10
successML	1
successPeeling	0
t	3
u	[1,0,1,1,0]
y	[0,2,0,0,0,2,2,1,1,0]

קיבלנו תוצאות שונות. הפענוח בעזרת ML הצליח ואילו הפענוח בעזרת PEELING נכשל.

שאלה 6:

מספר האלמנטים הכללי במטריצה זו הוא : $30000 * 60000 * 1.8 = 9^{10}$

מתוכם, מספר האלמנטים עם ערך 1 הוא: $5^{10} * 1.8$

הטיפוס של המשתנה H הוא sparse double , הזיכרון שהוא תופס הוא 3360008 בתים המשתנה Hfull תופס 13.4Gb

מסקנה: עדיף להשתמש במשתנה מסוג sparse במקום במערך דו-מימדי כדי לייצג מטריצה כדי לחסוך בזכרון (וגם בזמן החישוב כי לא צריך לעבור על כל איברי המטריצה)

HFullsize:

>>					
Name	Size	Bytes	Class	Attributes	
H	30000x60000	3360008	double	sparse	
Name	Size	Bytes	Class	Attributes	
HFull	30000x60000	14400000000	double		

שאלה 7:

מספר המחיקות בערוץ הוא 23918, והחלק היחסי שלהם $p = 0.4 = 60000/23918$, כלומר זוהי ההסתברות למחיקה בערוץ. הפענוח הצליח וזאת לפי המשתנה במטלב `successPeeling = 1`

```
>> whos fracErasedBits
```

Name	Size	Bytes	Class	Attributes
fracErasedBits	1x23918	191344	double	

ניסוי 6:

שאלה 1:

Name ▲	Value
strIn	'46733'
strOut	'54766'

הקלט הוא `strIn = 46733`, הפלט הוא `strOut = 54766`

נפלה שגיאה ב-4 ביטים, אופי השגיאה הוא שהביט שהשתנו הפכו לביטים במרחק של עד 3 תווים, כפי שהוגדר (5 'הפך' ל-4, 6 'הפך' ל-3, 3 'הפך' ל-6). הסתברות השגיאה שהוגדרה קטנה מחצי ואכן קיבלנו שתי שגיאות מתוך 5 'אפשרויות' ולכן הסתברות זו תואמת להגדרה של ההסתברות.






שאלה 2:

הרצנו את הסקריפט כמה פעמים בחלק הפענוח הצליח ובחלק אחר נכשל.

Success

nstrOut	7x5 char
repEnc	7x5 char
strIn	'46733'
strOut	'54766'
strOutDec	'46733'

Fail

Name ▲	Value
 nstrOut	7x5 char
 repEnc	7x5 char
 strIn	'46733'
 strOut	'54766'
 strOutDec	'43723'

שאלה 3:

אורך מילת קוד 3, וקצב המקודד הוא $\frac{1}{3}$.
המחרוזת המקודדת:

```
strEnc =
    '188818881181181'
```

שאלה 4:

המחרוזת המורעשת:

```
strNoised =
    '178817864281173'
```

שאלה 5:

המחרוזת המתקבלת (לפי צורת הקידוד החדשה)

```
strDec =
    '188818881181181'
```

מחרוזת זו זהה למחרוזת המקודדת של שאלה 3

שאלה 6:

המחרוזת המתקבלת:

```
strDec2 =
    '46733'
```

המפענח הצליח

שאלה 7:

המחרוזת המתקבלת:

```
strIn =  
  
    'hello, my cellphone number is 555046205, thanks.'  
  
>> strOut  
  
strOut =  
  
    'helli, my vellogonw bymber oa 555037105. thsnkd,'
```

שאלה 8:

נבחר את התווים כך שקיימים 2 מספרים המפרידים ביניהם (נתחיל מן הקצוות), כך שמפענח יכול להתאים כל אות אל האות המקורית שלה. התווים הללו הם

1,4,7,0

q,r,u,p

;,a,f,j

/,z,v,m

בעזרת תו הבחנה בודד ניתן לייצר 16 מילות קוד, בהתאם ל 16 האפשרויות שקיבלנו. לא ניתן לקודד כל תו מקלדת בעזרת תו הבחנה בודד משום שלא ניתן לשחזר בצורה חד משמעית את אות המקלדת המקורית. לדוגמא עבור התווים e,r,t כולם מותאמים לתו r

בעזרת שני תווי הבחנה ניתן לייצר $2^{16} = 256$ מילים. אורך מילת הקוד יהיה 2. קצב המקודד הוא $\frac{1}{2}$.

שאלה 9:

encoderOut

```
encoderOut = {
  '1','4','7','0','q','r','u','p','a','f',...%10 elements
  'j',';','z','v','m1','m4','m7','m0','mq','mr',...%10 elements
  'mu','mp','ma','mf','mj','m;','mz','mv','/1','/4'...%10 elements
  '/7','/0','/q','/r','/u','/p','/a','/f','/j','/;','...'...%10 elements
  ' '};
```

נחשב את האורך הממוצע:

$$((14*1)+(26*2))/(40) = 1.65$$

אם מילת הקוד מתחילה ב m או ב / , המקלט ימתין לתו נוסף כך שנקבל מילת קוד באורך 2, אחרת מילת הקוד היא באורך 1

שאלה 10:

המחרוזת המקודדת:

strEnc =

```
'm;z/l/lmq/f /am4 /qz/l/lmr;m/q/pz /pm7/a/uzv m0mp qqqlf0r4fq/f m1m;mu/pmvmmp/j'
```

קיימים 76 תווים במחרוזת זו.

שאלה 11:

המחרוזת המתקבלת:

strNoised =

```
',;z/l.2nq/f /s,4 /qx/l.lmem;nq/pz .pm7/a/izv m9mp qqqd0t3gq/g m1m1,i/omv,o.j'
```

שאלה 12:

לפי הרמז, נחלק את האלגוריתם לשני שלבים. השלב הראשון נסיר את הרעש ע"י שימוש בעובדה שהשתמשנו בתווי הבחנה בלבד בקידוד ולכן נוכל להבטיח פענוח נכון. בשלב השני נשתמש במיפוי ההפוך שאיתו קודדנו את המחרוזת.

שאלה 13:

המחרוזת שהתקבלה, המחרוזת המקורית ללא שגיאות:

```
strIn =  
  
    'hello, my cellphone number is 555046205, thanks.'  
  
>> strDec2  
  
strDec2 =  
  
    'hello, my cellphone number is 555046205, thanks.'
```

ניסוי 7ב:

שאלה 1:

כי האילוץ לא התקיים.

הפונקציה שבודקת את האילוץ בודקת אם יש רצף של אחדים, ואצלנו קיים רצף של אחדים.

```
Error using ConstChannel (line 3)  
Input to channel does not satisfy channel constraint
```

```
Error in Ex7b (line 10)  
ChannelOut=ConstChannel(binFile);
```

```
function out = ConstChannel(in)  
if (sum(in.*[0 in(1:end-1)])>0)  
    error("Input to channel does not satisfy channel constraint");  
else  
    out = in;  
end  
end
```

שאלה 2:

עבור קוד באורך n מוסיפים ביט 0 לכל ביט כלומר במקודד יש באות n ביטים. כלומר, 12000 ביטים. לכל ביט אחד יוצא אותו ביט עם 0 לידו כלומר $1 \leftarrow 2$ ומכאן קצב השידור הוא $\frac{1}{2}$.

שאלה 3:

אין שגיאה.

קיבלנו אותו אות שזהה ל-`binFile`.

שאלה 4:

```
DecIn = {'000', '001', '010', '011', '100', '101', '110', '111'};
```

```
DecOut = {'00000', '10000', '01000', '00100', '00010', '10100', '01010', '10010'};
```

מבחינת קצב, לכל 3 ביטים המקודד מוציא 5 ביטים \leftarrow קצב השידור הוא 60%.

אם גודל האות הוא n אז האות המקודד הוא בגודל 5 n . כלומר עבור ה-`binFile` שהוא בגודל 6000, גודל המקודד יהיה 10000 ביטים.

```
>> whos binFileEnc
```

Name	Size	Bytes	Class	Attributes
binFileEnc	1x10000	20000	char	

שאלה 5:

```
>> whos binFileEnc
```

Name	Size	Bytes	Class	Attributes
binFileEnc	1x9024	72192	double	

בשיטה הזאת כמות הביטים המשודרים תלויה במידע עצמו, בהינתן מידע שכולו אחדים נקבל כמות ביטים כפולה לשידור ועל כן קצב שידור קטן יותר, עבור מידע חסר אחדים נשדר כמות קטנה יותר של ביטים מאשר בשיטה שהצענו בסעיף 4 בו כמות הביטים קבועה וחסרת תלות במידע עצמוץ

ניסוי 8:

שאלה 1:

התמונה שהתקבלה:



המערך שמייצג את התמונה הוא מטיפוס double וגודלו 2097152 משתנים מסוג double. משתנה בודד מסוג double מכיל 64 ביטים על כן נכפול ב 64 ונקבל את מספר הביטים שמייצגים את התמונה

$$134217728 = 64 * 2097152$$

שאלה 2:

התמונה שהתקבלה:



נראה שהתמונה שהתקבלה בשאלה 2 זהה לתמונה שהתקבלה בשאלה 1

שאלה 3:



זוהי תמונה 'רועשת', סימנו שני פיקסלים שעברו הרעשה. זיהינו כל פיקסל כזה מכיוון שצבעו שונה ביחס לכל הפיקסלים שמסביבו (לכל הכיוונים ולא לכיוון בודד, כמו למשל ליד הבניינים ששם הפיקסלים משתנים 'בחדות'). כמו כן ראינו כבר את התמונה המקורית ואנו יודעים שבאזור זה אין פיקסלים לבנים וכן התמונה לא מאוד רועשת כך שבאמת ניתן לזהות שזוהי התמונה המקורית (לא רועשת מדי). בנוסף, לפי ההיכרות שלנו עם התמונה אנו יכולים להסיק שבאזור זה הפיקסל נדרש להיות כהה יותר, כלומר קרוב יותר לערך 0, אם כי אנו לא יכולים לשער את הערך המדויק שלו.

שאלה 4:

המערך שמייצג את התמונה (CI) הוא מטיפוס double וגודלו 332599 משתנים מסוג double. משתנה בודד מסוג double מכיל 64 ביטים על כן נכפול ב 64 ונקבל את מספר הביטים שמייצגים את התמונה

$$332599 * 64 = 21286336$$

חישוב של פקטור הדחיסה

$$2678368 / 332599 = 8.0528$$

Name	Size	Bytes	Class	Attributes
CI	1x1	2678368	struct	

שאלה 5:

התמונה שהתקבלה:



התמונה מעוותת מאוד, ולא ניתן לשחזר את התמונה המקורית.

דחיסה היא אזורית יותר כך שכל פיקסל מייצג אזור יותר גדול מהתמונה המקורית. כאשר מייחסים פיקסל בודד למפות אזור יחסית גדול מהתמונה המקורית, הרבה מידע אבד כך מקשה עלינו לשחזר את התמונה המקורית.

שאלה 6:

ערכי k האפשריים הם 11,7,5

$$100 * \frac{15 * \left\lceil \frac{332\,599}{k} \right\rceil - 332\,599}{332\,599}$$

על מנת לחשב את היתירות נציע את הקשר הבא

כך החישוב יניב

עבור 36% , k=11 יתירות, יכולת התיקון היא 1

עבור 114% , k=7 יתירות, יכולת התיקון היא 2

עבור 200% , k=5 יתירות, יכולת התיקון היא 3

שאלה 7:

נבחר $n=127$, $k=106$. התמונה שהתקבלה:



על פי החישוב של שאלה 6 היתירות היא 19%. קצב הקוד הוא $106/127 = 0.834$. גודל המערך שייצג את התמונה הוא 332605 משתנים מסוג double וביטים 21286720.

שאלות מחשבה:

ניסוי 6:

(א) קוד החזרות מסעיף 2 נכשל מכיוון שההסתברות אמנם קטנה מחצי אך כן גדולה מאפס, ולכן ייתכנו מספר רב של שגיאות כך שלא יהיה ניתן לשחזר את המחרוזת המקורית.

(ב

[illegible]

ג) תווים ברי הבחנה הם תווים שמובטח כי השחזור שלהם מתבצע בצורה מדויקת אם נופלת שגיאה.
ככל שישנם יותר תווים כאלו, כך משתפרת יכולת השחזור באמצעות מילות קוד קטנות.
הממוצע שקיבלנו הינו 1.65. אם היינו משתמשים בקוד בלוק, הממוצע היה זהה לאורך כל מילה, כלומר 2.

(ד) הממוצע שקיבלנו הינו 1.65. אם היינו משתמשים בקוד בלוק, הממוצע היה זהה לאורך כל מילה, כלומר 2.

(ה) אם לא היינו משתמשים בתווים ברי הבחנה לא היה ניתן לבצע שחזור מושלם, מכיוון שקוד תיקון שגיאות לא מבטיח תיקון של כל השגיאות.

ניסוי 8:

א) אפשר לזהות בעין פיקסל רועש מכל מיני סיבות, כמו צבע, צורה וכו'. וזאת מכיוון שערך הפיקסל המקורי יהיה זהה בקירוב טוב לערכי הפיקסלים שסביבו. ובגלל שאלו תמונות טבעיות שיכולים לנחש בקלות מה הערך שצריך להיות נוכל לקבל אינטואיציה על מה היה הצבע המקורי.

ב) לא ניתן לשחזר את התמונה מכיוון שהדחיסה גורמת לאיבוד מידע רב.

(ג) מספר הביטים שהועברו בערוץ הכולל: 21,288,192.
מספר הביטים שהועברו בערוץ ללא קוד וללא דחיסה: 134,217,728
כלומר בערך פי 7. כלומר, הקידוד והדחיסה חסכו לנו משמעותית בזכרון ובגודל האות המשודר וכך גם שיפר את קצב השידור.

(ד) אם נעשה כך, נתקשה מאוד לפענח את הדחיסה בגלל הרעש שהתווסף לה, וזאת מכיוון שלא יהיה ברשותנו הקידוד שאמור לזהות שגיאות.

