משפט PCP והשימושים להוכחת קושי של בעיות קירוב

יהודה קליין

שלום קורא יקר. נראה שאתה מתעניין בסמינר שלי – מעולה, מישהו יקרא אותו! כמו שאמרה הכותרת, הסמינר יעסוק ב- "משפט PCP, ואיך אפשר להשתמש בו כדי להראות על בעיות קירוב מסוימות שהן NP קשות". ניסיון כלשהו, גם אם מועט, בהתעסקות עם מכונות טיורינג והערכת קושי של הכרעת שפות, יעזור מאוד בהבנת הסמינר. גם הכרה של מושגים באלגברה לינארית ותורת ההסתברות תועיל להבנת חלקים מסוימים בו.

בתור התחלה, לפני שנעמיק בנבכי הקסמים של הוכחה הסתברותית, העלאה של גרפים בחזקות ורדוקציות מבעיות קירוב קשות לבעיות קירוב קשות אחרות, כדאי שנסביר את מבנה הסמינר:

חלק א

בו ניזכר קצת במושגים שקשורים לחישוביות, קצת במושגים שקשורים לסיבוכיות, ונציג גם כמה מושגים שאולי לא היו מוכרים לפני כן.

חלק ב

בו נגדיר מערכות הוכחה הסתברותית, נציג את שתי הגרסאות של משפט ה- PCP, ונוכיח שהן שקולות.

חלק ג

בו ניגע בשני הנושאים הבאים: הראשון הוא גרפים מרחיבים, השני הוא שערים לוגיים בוחני השמה (מבחני השמה, באנגלית: assignment testers).

חלק ד

בו נציג את עיקרי הוכחת משפט PCP לפי המאמר של פרופ' אירית דינור מ- 2005.

חלק ה

בו נשתמש במשפט החדש שלנו כדי להראות על כמה בעיות נבחרות שהן קשות לקירוב.

את **חלק א**, החימום שלנו לפני שמתחילים לעבוד באמת, נתחיל ממש בעמוד הבא.

חלק א – חזרה קצרה על מושגים נבחרים מתורת החישוביות והסיבוכיות

המחלקה NP

המושג PCP קשור באופן הדוק להגדרה הפורמלית של המחלקה NP, ולכן כדאי להזכיר את הניסוח המדויק שלה.

המחלקה NP, שהיא מחלקת כל השפות הכריעות בזמן פולינומיאלי על ידי מחשב (או: מכונת טיורינג) לא דטרמיניסטי, מוגדרת להיות כל השפות L שעבורן מתקיים התנאי הבא:

קיימת מכונה דטרמיניסטית פולינומיאלית M, כך שאם $w \in L$ היא מילה תקינה בשפה, קיימת מחרוזת c ש- M תוכל לקבל ביחד עם w ולקבוע בעזרתה את שייכות w לשפה. אם L אינה מילה בשפה, c לא קיימת, ו- M תדחה את w עבור כל c' שנזין לה.

הרציונל שמאחורי ההגדרה, הוא ש- c אמורה להיות פתרון עבור מופעים של בעיות NP קשות. אנחנו מקלים על המכונה שלנו – במקום לפתור את הבעיה w, אנחנו נותנים לה את הפתרון מראש, והיא צריכה רק לוודא שהפתרון נכון. בניסוח אחר, c היא הוכחה לכך ש- w היא מילה ב- L, ותפקידה של M הוא לוודא שההוכחה אינה שקרית.

לדוגמה: השפה SAT היא NP שלמה. לכל נוסחת SAT ספיקה, קיימת איזושהי השמה למשתנים שבתוך הנוסחה. נייצג את ההשמה בתור המחרוזת c, וקל יהיה לכתוב תכנית שמקבלת את c ואת הנוסחה שהיא לכאורה מספקת, ומוודאת שהשמת המשתנים לפי c אכן מספקת את הנוסחה.

פתרון בעיות הסתברותי

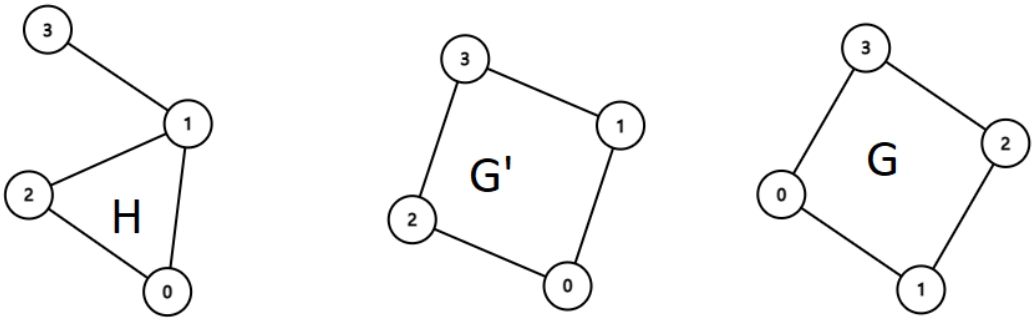
לפעמים הכרעה של בעיות מסוימות היא משימה אקספוננציאלית במשאבי זמן ומקום. לפעמים גם אם אפשר לחסום את המשאבים הנדרשים על ידי פולינום, הפתרון הדטרמיניסטי לבעיה פשוט יקר מדי – למשל בעיית הכרעת הראשוניות של מספר טבעי, שאלגוריתם AKS יודע לפתור פחות או יותר בזמן $O(n^6)$. זה נחמד, אבל לשימושים סטנדרטיים נעדיף את אלגוריתם מילר-רבין, שאומנם יכול לתת לנו תשובות שגויות, אבל בהסתברות נמוכה מספיק כדי שנעדיף את זמן הריצה $O(n^3)$ שלו.

באופן כללי, אלגוריתם הסתברותי מכיל רכיב אקראי לחלוטין (בדרך כלל הפיכת ביט בין 0 ל- 1 באופן רנדומלי, המקבילה החישובית להטלת מטבע) שבעזרתו ניתן להפיק פתרונות עם "ביטחון גבוה" בנכונות התשובה, במהירות רבה יותר מזמן הריצה של אלגוריתם שמבטיח פתרון מדויק. בדרך כלל אלגוריתם הסתברותי כזה מורכב ממבחן פשוט (לדוגמה, בדיקת השורשים של 1 עבור בסיס אקראי כלשהו – אם נחזור לדוגמא של מילר-רבין), ועל המבחן נחזור כמה פעמים כדי להוריד את ההסתברות לטעות מתחת לסף רצוי כלשהו.

פעמים רבות, נעדיף להגדיר את הטלת המטבע של המכונה מחדש בתור גישה למחרוזת אקראית מעל $\{0,1\}$. השקילות נובעת מהעובדה שאם לא היינו נותנים למכונה את המחרוזת האקראית, היא הייתה יכולה לייצר אותה בעצמה, ומצד שני מכונה דטרמיניסטית יכולה לקרוא את האות הבאה במחרוזת אקראית בתור תחליף להטלת מטבע.

מערכת הוכחה אינטראקטיבית

נתחיל מתיאור שפת הגרפים הלא איזומורפיים (GRAPH-NONISOMRPHISM, NONISO). נתונים לנו תיאורים של שני גרפים, G ו- H. לכל צומת בכל אחד מהגרפים יש ID ייחודי (מספר, שם או כל guid כזה או אחר). יכול מאוד להיות שבסידור מחדש של זהויות הצמתים בגרף G, נקבל תיאור מדויק של H.



הגרפים G ו- 'G איזומורפיים אחד לשני, אבל לא ל- H

הבעיה שלנו היא שאנחנו רוצים להיות בטוחים ש- G ו- H הם אכן גרפים שונים (נניח שיש לנו random graph generator ואנחנו רוצים לוודא שהוא אכן מייצר גרפים אקראיים ולא את אותו המבנה המסובך שחוזר על עצמו בשמות שונים כל פעם).

את הבעיה ההפוכה, GRAPH-ISOMORPHISM, שבה אנחנו מנסים להראות שהגרפים שווים, קל להכניס במחלקה NP. כל מה שאנחנו צריכים היא הוכחה c שתכיל את שמות זוגות הצמתים המקבילים ב- G ו- H, ומחשב שמוודא שלאחר החלפת השמות אכן מתקבל תיאור של אותו הגרף. בזמן כתיבת הסמינר, עוד לא ידוע אם NONISO נמצאת ב- NP.

עכשיו נעבור לסיפור קצר:

למיכאל *יש שני גרפים*, G ו- H. מסיבות שחורגות מהיקף הסמינר[1], למיכאל *חשוב מאוד* שהגרפים לא יהיו איזומורפיים. מסיבות אחרות שגם בן חורגות מהנושא שלנו, מסתובבים בינינו **מוכיחים** מרושעים, ומטרתם היחידה היא לשכנע את מיכאל שהגרפים שהוא מחזיק הם אכן לא איזומורפיים, למרות ש- G ו- H הם אותו הגרף. המוכיחים האלו הם בעצם מחשבים מהעתיד, וכל בעיה חישובית שנוכל לחשוב עליה, אותם המחשבים יכולים לפתור ב- 0 זמן.

מיכאל צריך מחשב חזק כדי לבדוק אם הגרפים שלו לא איזומורפיים, אבל הוא לא יכול לסמוך על המוכיחים שסביבו (נזכיר – הבעיה החישובית שהמוכיחים מתמודדים איתה כרגע היא "איך אפשר לשכנע את מיכאל שהגרפים האיזומורפיים שלו הם לא איזומורפיים").

מיכאל מחליט ליצור **פרוטוקול הוכחה אינטראקטיבית**. הוא פותח את הלפטופ שלו, וכותב את התכנית הבאה:

א. בחר באופן רנדומלי לגמרי (מחולל הרנדומליות היחיד שאף מוכיח לא יכול לנבא את פעולתו הוא מחולל רנדומלי אמיתי, לצורך הדיון מיכאל מחזיק כזה) אחד מהגרפים H או G. הגרף שנבחר יהיה K.

ב. בחר באופן רנדומלי לגמרי סידור מחדש של הצמתים בגרף K, ושלח אותו ביחד עם H ו- G אל מוכיח כלשהו. שאל את המוכיח אם K הוא סידור מחדש של H או של G.

[1] כאן הכוונה ב'חורגות מהיקף הסמינר', היא - לא היה לי כוח לחשוב על סיבות שכאלה. זה גם המקום לומר שמחשבים מעולם לא היו אמורים להתקיים מעבר להגדרתם המתמטית האבסטרקטית, ואולי גם להוסיף ש'שימושים מהעולם האמיתי' כפי שהם נקראים, הם באופן גורף וללא יוצאים מן הכלל – מעוררי דחייה.

אם תשובת המוכיח נכונה, בסבירות של לפחות חצי הגרפים לא איזומורפיים, כיוון שמוכיח הצליח להבדיל ביניהם בלי קשר לשמות הצמתים. אם המוכיח היה רק מנחש את התשובה היה לו סיכוי של 50% לטעות. אם התשובה לא נכונה, מיכאל מחזיק גרפים איזומורפיים, כיוון שהמוכיח ניחש את התשובה בלי לדעת בוודאות – וטעה.

ג. חזור על התהליך עד שהמוכיח שכנע את הלפטופ בהסתברות גבוהה מספיק שהגרפים לא איזומורפיים.

התיאור הקלאסי של **מערכת הוכחה אינטראקטיבית**, או **פרוטוקול מוכיח-מאמת**, היא מכונת טיורינג פולינומיאלית הסתברותית (המאמת) שמתקשרת עם ישות חישובית בעלת כוח חישוב בלתי מוגבל (המוכיח) עם מניעים לא ידועים. התקשורת מתבצעת על ידי סדרה של מסרים (מחרוזות) – המאמת מקבל מילה שנמצאת או לא נמצאת בשפה שאותה הוא אמור להכריע. על סמך המילה הזו המאמת מחשב חישוב פולינומיאלי, שולח שאלה למוכיח, קורא את התשובה, ועל סמך התשובה מחשב שאלה נוספת, כך הלוך וחזור, כאשר למאמת נוספות כל פעם תשובות נוספות של המוכיח להתבסס עליהן בחישוב השאלה הבאה. בסופו של דבר המאמת יקבל את המילה אם ההסתברות לטעות נמוכה מספיק.

תיאור פורמלי יהיה: המאמת הוא פונקציה עם קלט שמכיל מילה שצריך להכריע, ואת ההודעות שנשלחו מהמאמת למוכיח ובחזרה עד עכשיו. הפלט יכול להיות קבלה של המילה, דחייה של המילה, או שאלה חדשה למוכיח. המוכיח יהיה פונקציה עם קלט שמכיל את התקשורת בין המאמת למוכיח עד עכשיו, והפלט יהיה תשובה למאמת.

המונח PCP שבו נתעסק הוא עוד דוגמה למחלקת שפות שבה המאמת מנסה לוודא הוכחה חיצונית כלשהי, כשההבדל המרכזי הוא שכל ההוכחה נתונה מראש, ואין מוכיח שצריך לענות על שאלות. בהמשך נגדיר עוד מגבלות על המאמת כדי ליצור מחלקות PCP עם מגבלות שונות על "כמות הרנדומליות" ו- "כמות הגישה להוכחה".

בעיות קירוב קשות

בעיות קירוב בדרך כלל משמשות לקירוב פתרונות אופטימליים כלשהם, ולא כדאי לדבר עליהן בלי ההקשר המתאים – כלומר הגיע הזמן להציג את המושג **בעיות אופטימיזציה**. לשם כך נשתמש בשפה 3SAT המפורסמת. נזכיר ש- 3SAT היא גרסה של SAT שבה כל פסוקית מכילה 3 ליטרלים בדיוק, והאופרטורים הלוגיים המותרים הם $\{\neg\ \wedge,\vee\}$. נוסחה לוגית נתונה היא מילה בשפה 3SAT אם ניתן לתת ערך 1 או 0 לכל אחד מהמשתנים, כך שהנוסחה תקבל ערך אמת. נתבונן בשני המופעים הבאים (לא צריך להתעמק בנוסחאות, אין להן משמעות מיוחדת חוץ מהעובדה שהדוגמאות שבחרתי בהמשך מספקות חלק או את כל הפסוקיות בהן):

$$(a \vee b \vee c) \wedge (b \vee c \vee a) \wedge (a \vee \neg b \vee \neg c) \wedge (c \vee \neg a \vee \neg c) \wedge (\neg b \vee c \vee \neg b) \qquad (i)$$

$$(a \vee b \vee c) \wedge (a \vee b \vee c) \wedge (\neg a \vee \neg a \vee \neg a) \wedge (\neg b \vee \neg b \vee \neg b) \wedge (\neg c \vee \neg c \vee \neg c\ ) \qquad (ii)$$

המופע הראשון (i) ניתן לסיפוק על ידי ההשמה:

$$a = 1,\ \ b = 0,\ \ c = 0$$

ולכן הנוסחה היא מילה ב- 3SAT.

לעומתה הנוסחה השנייה (ii) אינה מסופקת על ידי אף השמה.

אבל, בעיית 3SAT מגדירה באופן מידי את בעיית Max-E3SAT, או: מה הוא המספר המקסימלי האפשרי של פסוקיות בנוסחה נתונה שניתן לספק בעזרת השמה של המשתנים ל- $\{1,0\}$. במקרה והנוסחה הנתונה היא (ii), קל לראות שניתן לספק ארבע מתוך חמשת הפסוקיות בעזרת השמה מסוג:

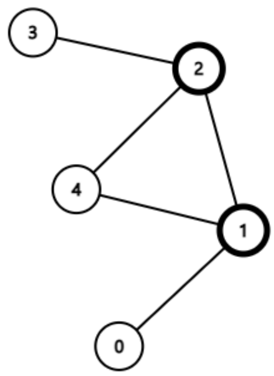
$$a = 1,\ \ b = 0,\ \ c = 0$$

מציאת המספר המקסימלי של פסוקיות הניתנות לסיפוק בנוסחה לוגית נתונה היא בעיה NP קשה. הרדוקציה מ- 3SAT ל- Max-E3SAT מיידית: אם הראשונה פתירה, השנייה תחזיר את מספר הפסוקיות בנוסחה עבור כל מופע של 3SAT.

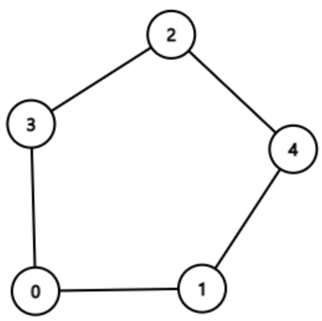
Max-E3SAT היא בעצם **בעיית אופטימיזציה**. אי אפשר לספק את כל הנוסחה, אבל בכל זאת נרצה לספק כמה שיותר פסוקיות ממנה. דוגמה נוספת לבעיית אופטימיזציה מפורסמת היא:

בעיית כיסוי הצמתים (Vertex-Cover).

בגרסתה המקורית, בעיית כיסוי הצמתים מנוסחת כך: בהינתן גרף ומספר k, האם ישנה קבוצה של k צמתים בגרף כך שלכל קשת בגרף, לפחות אחד הקצוות יהיה איבר בקבוצה. הבעיה ידועה להיות NP שלמה, ואנחנו לא מכירים לה פתרון פולינומיאלי דטרמיניסטי.



הצמתים 1 ו- 2 הם כיסוי



כאן אין כיסוי צמתים בגודל 2, הכיסוי המינימלי כולל שלושה צמתים

גרסת האופטימיזציה Min-Vertex-Cover של הבעיה מבקשת ממנו, לכל גרף, להוציא בתור פלט את מספר הצמתים המינימלי כך שלכל קשת בגרף, לפחות אחד הקצוות יהיה בקבוצה. למשל עבור הגרף השמאלי למעלה, הפלט שיתבקש הוא 3, כיוון שרק תת קבוצה של 3 ומעלה צמתים בגרף יכולה לכסות את כל הקשתות.

הנה אלגוריתם שרץ בזמן אקספוננציאלי ופותר את Min-Vertex-Cover:

בהינתן גרף G עם n צמתים ו- m קשתות, $G = (V, E) : |V| = n, |E| = m$:

עבור כל i מ- 1 עד n:

עבור כל קבוצת צמתים K בגודל i:

עבור כל קשת ב- E: בדוק האם אחד מקצוות הקשת נמצא ב- K.

אם כל הקשתות עברו את המבחן, החזר את i בתור גודל הכיסוי המינימלי.

הלולאה הפנימית "עבור כל קבוצת צמתים K בגודל i:" גורמת לכך שהאלגוריתם מבצע חישוב פולינומיאלי עבור כל תת קבוצה של צמתים ב- V, כלומר זמן הריצה הכולל הוא אקספוננציאלי. חשוב לציין שהאלגוריתם יכול באותה מידה להחזיר את קבוצת הקודקודים המינימלית שמהווה כיסוי צמתים במקום מספר, בלי תוספת רבה לחישוב.

Min-Vertex-Cover היא בעיה מצוינת להצגת הרעיון מאחורי חישובי אופטימיזציה, כיוון שיש לה אלגוריתם קירוב פשוט מאוד מסדר 2. האלגוריתם יעבוד כך:

כל עוד יש ב- G קשת לא מסומנת:

בחר קשת לא מסומנת e ב- E. סמן את e והוסף אותה לקבוצה H.

עבור כל אחד משני הקודקודים ש- e מחברת:

עבור כל קשת שצמודה לקודקוד, סמן את הקשת (בלי להוסיף ל- H).

החזר בתור פלט כל קודקוד שצמוד לקשת שנמצאת ב- H.

נקרא לקבוצת הקודקודים שהאלגוריתם מחזיר X.

האלגוריתם שלנו סימן את כל הקשתות, וכל קשת מסומנת צמודה לקודקוד ב- X, כלומר קיבלנו כיסוי קודקודים. כיוון שכל הקשתות שב- H רחוקות לפחות מרחק של קשת אחת מהשנייה, כיסוי מינימלי מחייב לפחות קודקוד נפרד לכל אחת. אנחנו הוספנו שני קודקודים לכל אחת, לכן הכיסוי שהחזרנו לכל היותר כפול בגודלו מהכיסוי המינימלי. האלגוריתם שלנו נחשב קירוב מסדר 2, כיוון שהמספר שהוא מחזיר הוא לכל היותר 2 פעמים הפתרון האופטימלי האמיתי. זהו לא קירוב טוב במיוחד, כל מה שעשינו בעצם היה לדלג על הגרף בקפיצות של 2 קשתות, ולסמן כל צומת שמצאנו – אבל מצד שני, הבעיה היא NP קשה וסביר שיהיה קשה לקרב אותה.

CSP – **C**onstraint **S**atisfaction **P**roblem

אחרי ההצלחה המטאורית של מיכאל נגד המוכיחים, העירייה רוצה שהוא יפתור עבורה בעיית גינון מסובכת במיוחד.

ישנן שתי חלקות פנויות בפארק העירוני, שיכולות להכיל שלושה עצים כל אחת. ספק הצמחייה מציע למכירה את הזנים האלה - {אקליפטוס, שקדייה, רוזמרין} אבל לרוע המזל, שלושת הזנים המסוימים האלה מייצרים שרשרת אלרגיות לא נעימה:

אקליפטוס לא יכול להישתל מצפון לשקדייה או אקליפטוס אחר.

שקדייה לא יכולה להישתל מצפון לרוזמרין או שקדייה אחרת.

רוזמרין לא יכול להישתל מצפון לאקליפטוס או עוד רוזמרין.

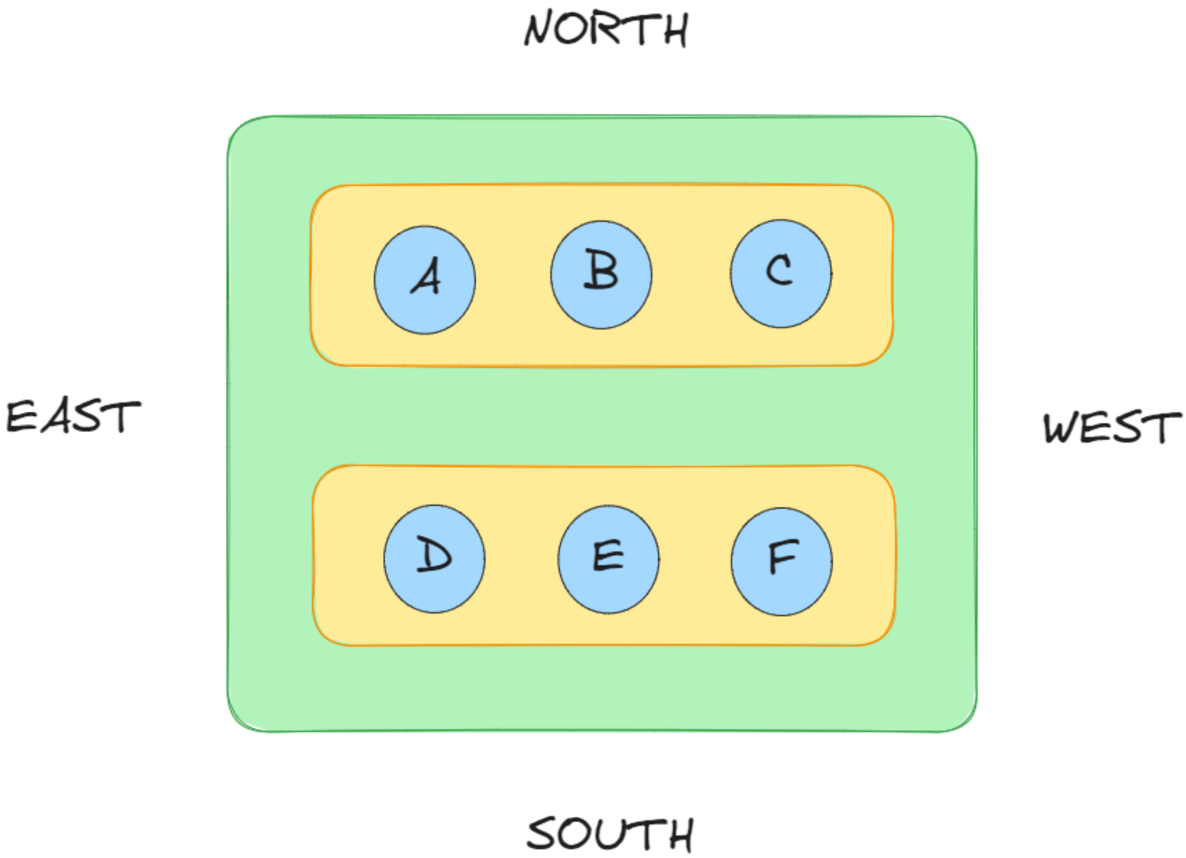
בנוסף, כל אחת מהחלקות הפנויות יכולה לספק 2 קוב מים ביום. צריכת המים היומית של הזנים השונים מוערכת כך:

אקליפטוס – 1.8 קוב

שקדייה – 1 קוב

רוזמרין – 0.5 קוב

מיכאל מקבל מפה של הפארק, שבה נמצאים בתור עיגולים המיקומים המדויקים עבור הצמחים שיש לשתול, ובתור מלבנים חומים החלקות המתאימות.



מיכאל, עם חוש הגננות המפותח שלו, יודע מייד להריח שיש כאן **CSP – בעיית סיפוק אילוצים**. עבור כל זוג מיקומים שנמצא במרחק שווה מהשמש (כלומר, אחד בדיוק מצפון לשני), קיים אילוץ אלרגני על הזנים שיכולים להישתל בו. עבור כל שלושת מיקומים שמהווה חלקה, יש אילוץ מימני על שלושת הצמחים הנשתלים.

הדבר הראשון שמיכאל עושה, זה לסדר את הבעיה בצורה פורמלית. יש לנו 6 משתנים – הקבוצה $\{A,B,C,D,E,F\}$. כל אחד מהמשתנים יכול לקבל איבר מהקבוצה $\{eucalyptus, almond, rosemary\}$ – זהו **התחום** של כל משתנה (בפארק כל התחומים שווים, אבל CSP אחר יכול להכיל תחומים שונים לכל משתנה). בנוסף, יש לנו את האילוצים – אותם נייצג בתור סדרות תקינות של השמות למשתנים. קודם כל, האלרגיות מונעות ממנו לשתול אקליפטוס מעל שקדייה וכן הלאה, ואנחנו נשארים עם:

$$c(A,D) = \{(eucalyptus, rosemary), (rosemary, almond), (almond, eucalyptus)\}$$

מה רשמנו כאן? האות c היא קיצור למילה $Constraint$, או אילוץ. זוהי הפונקציה מסדרות משתנים אל סדרות הערכים שהם יכולים לקבל ביחד. לפי הגדרת הבעיה, $c(A,D) = c(B,E) = c(C,F)$.

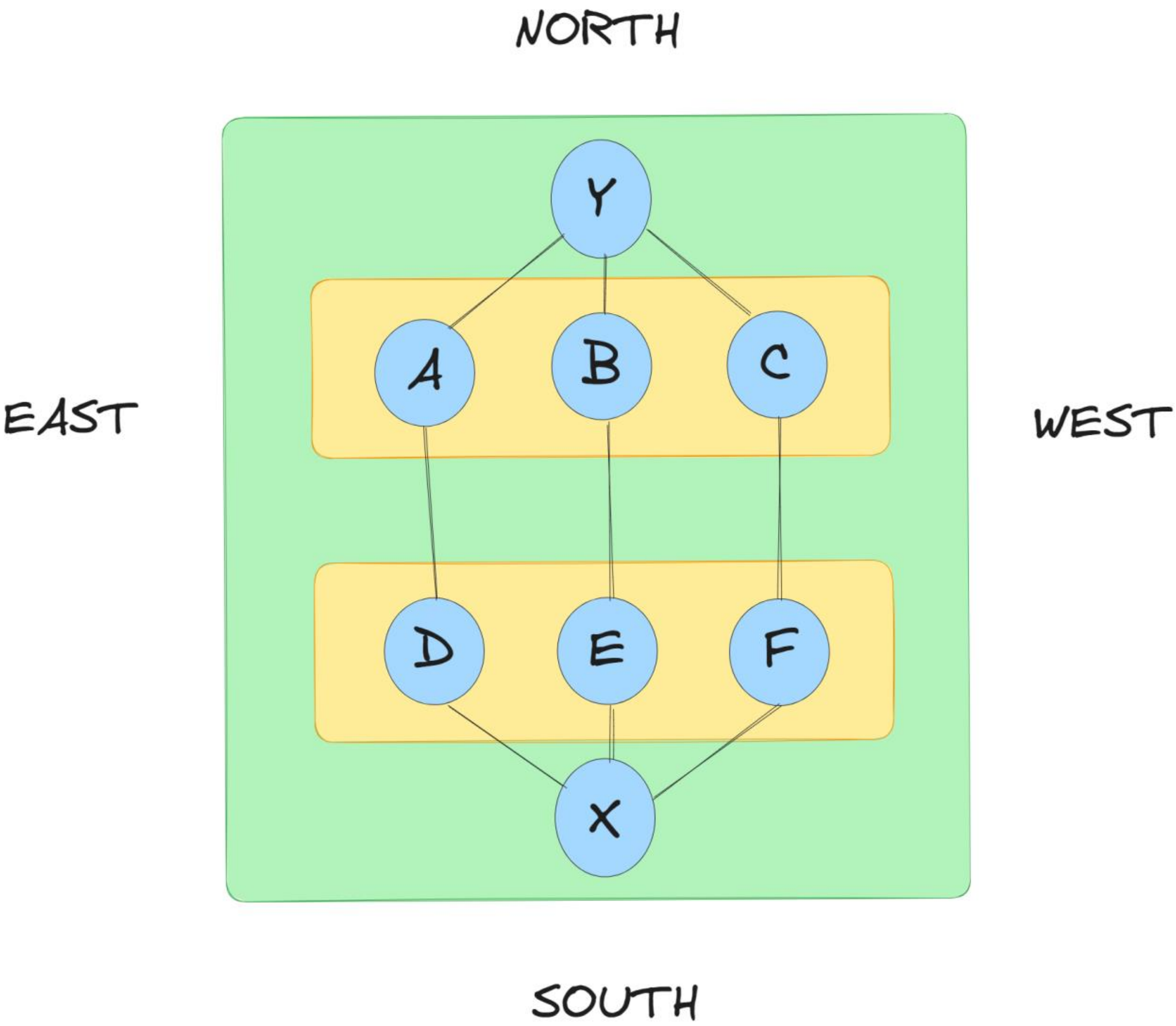
האילוצים על צריכת המים המשותפת לכל חלקה הם:

$$\begin{aligned} &c(A,B,C) = c(D,E,F) = \{\\ &(almond, rosemary, rosemary),\\ &(rosemary, almond, rosemary),\\ &(rosemary, rosemary, almond),\\ &(rosemary, rosemary, rosemary)\\ &\} \end{aligned}$$

או במילים אחרות, כל סידור של שתילים בחלקה, שמבטיח מספיק מים לכל צמח.

מיכאל מרגיש בתוך-תוכו שאת הבעיה המסוימת הזו הוא לא יוכל לפתור, אבל לפני שהוא מכריע שה- CSP אינו ספיק, הוא רוצה לנסות להפוך את הבעיה לפשוטה יותר – כזו שיש לה שתי משתנים לכל היותר בכל אילוץ. לשם כך הוא מגדיר את שני המשתנים X, Y. אנחנו נראה את שיטת הפעולה עם המשתנה X, שנועד לפשט את האילוץ הדרומי – Y עושה בדיוק את אותו הדבר מצפון:

התחום עבור המשתנים החדשים הוא שלשות מעל קבוצת העצים, שמהוות השמה תקינה לאילוצי המים, כלומר האילוץ על (D, E, F) הופך לתחום עבור המשתנה החדש. כל אחד מהמשתנים בחלקה מחליף את חלקו באילוץ המשולש, באילוץ אחר שמוצב עליו ועל X, ודורש שוויון בין הערך ש- X נותן למשתנה (כי הרי X בסופו של דבר נותן ערכים לכל שלשת המשתנים), לבין הערך שהמשתנה מקבל בסופו של דבר. זוהי רדוקציה מ- CSP כללי ל- CSP בינארי, שאנחנו יכולים לתאר בתור גרף עם צמתים שהם משתנים וקשתות שהן אילוצים:



לאחר הרדוקציה, מיכאל נזכר ש- CSP היא בעיה NP שלמה. הוא כמעט פורץ בבכי עד שהוא נזכר שיש לו רק 8 צמתים בגרף, שיכולים כל אחד לקבל עד 4 ערכים שונים, וגם שמעבד מודרני נותן 3 מיליארד דפיקות בשנייה. בעליזות רבה הוא מריץ אלגוריתם *BRUTE-FORCE* על הבעיה, מגלה שהיא אינה פתירה, וראש העיר מצליח בסופו של דבר להשיג זרעי תותים.

די קל להראות שבעיית ה- CSP היא NP שלמה. למעשה, היא מכלילה מספר רב של בעיות NP שלמות באופן מידי – למשל בעיית **GRAPH-3COLORING** שבה אנחנו מנסים לצבוע צמתים בגרף בשלושה צבעים כך שאף זוג צמתים שכן

לא מקבל את אותו הצבע. בגרסת ה- *CSP* הצבעים הם התחום עבור המשתנים, והאילוצים כוללים את כל הזוגות של ערכים שונים. ההוכחה שנראה למשפט ה- PCP היא בעיקרה: *איך אפשר לקחת בעיית* CSP *שאינה פתירה, ולעשות אותה **פחות** פתירה.* אני מבטיח שהמשפט הזה יהפוך להיות הרבה יותר הגיוני מבחינה תחבירית בהמשך.

קודים מתקני שגיאות

לפני שמדברים על קודים מתקני שגיאות, אנחנו צריכים לדבר על מרחק בין מחרוזות בינאריות, אז הנה שני מחרוזות רחוקות:

$$str_0 = 00000$$

$$str_1 = 11111$$

והמרחק ביניהן הוא – 5.

כפי שאפשר לנחש, אנחנו מודדים מרחק בין מחרוזות בתור "מספר הביטים שצריך לשנות כדי להפוך מחרוזת אחת לאחרת". במקרה שלמעלה, 5 פעולות *bit-flip* יהפכו סדרה של 5 אפסים לחמישה אחדות ולהפך. כמו רוב המדדים למרחק, גם כאן אפשר להשתכנע שהיחס סימטרי, ואנחנו נסמן מרחק בין מחרוזות עם הדלתא הסטנדרטית:

$$\Delta(str_0, str_1) = \Delta(str_1, str_0) = 5$$

כשאנחנו מסתכלים על קידוד **בינארי** של קבוצת סימבולים כלשהי, למשל $\Sigma = \{A, B, C, D\}$, אנחנו יודעים לומר שהקידוד המינימלי האפשרי עבורה כולל לפחות $\log_2 |\Sigma|$ ביטים, על מנת לוודא שכל שני סימבולים שונים לפחות בסיבית אחת – זהו מספר הביטים המינימלי שיאפשר לנו לספור את כל האיברים בסדרה. אם יהיו לנו פחות, בהכרח יש לנו יותר סימבולים לקודד מאשר מילות קוד אפשריות, ולפחות 2 מתוך הסימבולים יקודדו באותה צורה – מה שלא מאפשר להבדיל ביניהם. זו גם הסיבה שברוב הקידודים שלנו אנחנו פשוט סופרים, כמו בפונקציית הקידוד הבאה על Σ:

$$Enc: (A, B, C, D) \to (00,01,10,11)$$

תכונת ההבדלה הזו בין סימבולים שונים היא בעצם מרחק מינימלי של 1 בין סימבול אחד לאחר. למשל, המרחק של $Enc(A)$ משאר הקידודים נראה כך:

$$\Delta(A, B) = \Delta(0\mathbf{0}, 0\mathbf{1}) = 1$$

$$\Delta(A, C) = \Delta(\mathbf{0}0, \mathbf{1}0) = 1$$

$$\Delta(A, D) = \Delta(\mathbf{00}, \mathbf{11}) = 2$$

קודים מתקני שגיאות הם בסופו של דבר שיטות קידוד שונות שמסדרות את המרחקים בין מילות קוד כך שיהיה ניתן להבדיל בקלות רבה יותר בין תווים שונים – בדרך כלל בעזרת תוספת של סיביות נוספות. הדוגמה המפורסמת ביותר היא כנראה **קוד האמינג - (Hamming code)**, שמקודד *רביעיות* של ביטים בתור *שביעיות* של ביטים שמכילות את הארבע המקוריים, ועוד שלושה סיביות שמודדות זוגיות של תת-חלקים במילת הקוד. הקוד מצליח לשמור על מרחק מינימלי של 3 בין מילה למילה, ולכן אם חלה טעות ואחד הסימבולים עבר *bit-flip* מאיזושהי סיבה, תהיה מילה יחידה במרחק 1 ממנו, וכל השאר יהיו במרחק 2 לפחות – כלומר ניתן **לתקן שגיאות**, במקרים מסוימים.

הרבה פעמים יהיה לנו נוח יותר להתייחס למרחק היחסי של קוד, והכוונה היא פשוט למרחק בביטים חלקי האורך של מילת קוד. בדרך כלל יהיה ברור לאיזה סוג של מרחק אנחנו מתכוונים, אבל כדי להבדיל בקלות כדאי לזכור שמרחק רגיל הוא תמיד מספר שלם, ומרחק יחסי תמיד יהיה בין 0 ל- 1.

אותנו מעניין במיוחד קוד מתקן שגיאות בשם **קוד האדאמארד – Hadamard code.** בדרך כלל נוח לנו לחשוב על מחרוזות בתור מחרוזות – אבל הפעם, אנחנו רוצים מונחים של אלגברה לינארית. בסופו של דבר, אין הבדל בין מחרוזת באורך k של אפסים ואחדות, ובין וקטור כלשהו במרחב Z_2^k – אלה שני סידורים באותו אורך של איברים בינאריים.[2] אל קוד האדאמארד אפשר להתייחס בכל מיני צורות – אוסף של פונקציות לינאריות, XOR-ים ו-AND-ים על כל המחרוזות באורך כלשהו, וכנראה עוד כמה (יצא לי לשמוע על דבר שנקרא "מחרוזות מקופלות" בקונטקסט של הקוד הזה. בכנות, לא הבנתי במה מדובר, אבל אני חושב שהמושג אנלוגי ל"אנליזת פורייה של היפר-קוביות דיסקרטיות בינאריות" עם שם קצת פחות מלחיץ). בכל מקרה, אנחנו נשתמש במונחים של מכפלה סקלרית על וקטורים בינאריים.

אז איך עושים קוד האדאמארד? קוד האדאמארד מקודד מחרוזות בינאריות, ולכן כדי לקודד קבוצת סימבולים כלשהי אנחנו צריכים לתרגם אותה לקידוד בסיסי בינארי כלשהו. אנחנו כבר טיפלנו בצעד הזה, ולכן נשתמש בדוגמה הקודמת:

$$Enc: (A, B, C, D) \rightarrow (00,01,10,11)$$

כדי לקודד וקטור בינארי, אנחנו משרשרים את המכפלה הסקלרית שלו עם *כל הוקטורים האחרים* שנמצאים באותו המרחב. כל מכפלה כזו נותנת תוצאה מעל השדה המרחבי Z_2, כלומר 0 או 1. בתור דוגמה, ניקח את הסימבול B, ונעביר אותו את כל הדרך עד שנקבל מילת קוד האדאמארד.

צעד ראשון:

$$Enc(B) = "01"$$

צעד שני:

הוקטור שאנחנו מקודדים הוא המקבילה המרחבית ל- "01", כלומר $(0,1)$. אנחנו מחשבים את המכפלות הסקלריות שלו עם שאר הוקטורים במרחב, ומקבלים:

$$(0,1) \cdot (0,0) = (0 \cdot 0) + (1 \cdot 0) = 0$$

$$(0,1) \cdot (0,1) = (0 \cdot 0) + (1 \cdot 1) = 1$$

$$(0,1) \cdot (1,0) = (0 \cdot 1) + (1 \cdot 0) = 0$$

$$(0,1) \cdot (1,1) = (0 \cdot 1) + (1 \cdot 1) = 1$$

צעד שלישי:

משרשרים את כל התוצאות כדי לקבל מילת קוד יחידה:

“0”+”1”+”0”+”1”=”0101”

אם כך, קיבלנו $Had(10) = 0101$. יש כמה דברים שאנחנו רוצים לשים לב אליהם:

[2] אפילו מדד המרחק שלנו מיתרגם יפה לנורמת $l1$: $\Delta(str_0, str_1) = \|str_0 - str_1\|_{l1}$, למרות שזה פחות קריטי לנו.

א. כל וקטור במרחב שלנו מגדיר תו יחיד בקוד. יותר מכך, אם נתייחס לוקטורים במרחב שלנו בתור מספרים – כלומר 10=2 וכן הלאה, אנחנו מקבלים שיטה נוחה ממש לאנדוקס של תווים בקוד שלנו – אנחנו יכולים להתייחס לביט שהתקבל מהמכפלה עם הוקטור $(0,1,1)$ בתור הביט במקום השלישי, וכו' - ולכן מעכשיו אנחנו בסימון הבא:

הסיבית המתקבלת בקידוד הוקטור u על ידי הכפלה עם הוקטור v היא:

$$Had(u)[v]$$

ב. הקוד ארוך *מאוד*. כל וקטור מעל המרחב הבינארי שלנו מגדיר תו יחיד בקידוד, ואם ממד המרחב הוא k, יש לנו 2^k איברים שונים ב- Z_2^k.

ג. המרחק בין שתי מילות האדאמארד הוא חצי מהמילה, כלומר אם אנחנו מקודדים את הוקטורים u, v מממד k, וכל מילת קוד בעלת אורך של 2^k, אז $\Delta\big(Had(u), Had(v)\big) = 2^{k-1}$. במילים אחרות, לקוד האדאמארד יש מרחק יחסי של ½ .

תובנה ג לא הגיעה סתם כבה משום מקום, והמשימה הבאה שלנו היא להבין אותה. עבור המטרה הזו, אנחנו עומדים להגדיר את המושג "מצב סטטי" בתהליך המכפלה הסקלרית. כשאנחנו מכפילים את הוקטורים u, v, מצב סטטי הוא הקפאה של התהליך, כאשר הכפלנו רק חלק מהסיביות, הכפל עוד לא הסתיים, אבל אנחנו רוצים לבחון את הערכים שלנו בכל זאת. כיוון שמכפלה סקלרית היא סכום של מכפלות סיביות, וחיבור הוא קיבוצי, אנחנו נתעלם מסדר הסיביות בזוג וקטורים/מחרוזות כלשהו, ופשוט נגדיר אלו אינדקסים כבר השתתפו במכפלה (ונמצאים בסכום) ואלו עוד לא.

מעולה – עכשיו הגיע הזמן לקודד שני וקטורים שונים ב- Z_2^k. אנחנו נבחר את u ואת v, כי השמות שלהם הכי יפים. כיוון שאלו וקטורים שונים, אנחנו יודעים שיש להם אינדקס אחד לפחות שבו הם אינם שווים – אחד מחזיק 0 והשני 1. מפה לשם, אנחנו מגדירים את הקבוצות Eq ו- Neq. הראשונה היא כל האינדקסים i שבהם $u_i = v_i$, והשנייה היא כל האינדקסים שבהם $u_i \neq v_i$. אז אם למשל הוקטורים שלנו הם $v = (1,1,0) \quad u = (0,1,0)$, אנחנו נקבל:

$$Eq = \{1,2\}, \qquad Neq = \{0\}$$

אנחנו רוצים להתבונן בחלוקה מסוימת ש- Eq משרה לנו על המרחב: נניח שאנחנו בוחרים וקטור w כלשהו – יש קבוצה W של וקטורים במרחב ששווים לו בכל האינדקסים של Eq. w עצמו הוא איבר בקבוצה הזו, וגם כל הוקטורים ששונים ממנו אך ורק בסיביות של Neq, כלומר יש לנו בסך הכל $2^{|Neq|}$ וקטורים כאלה ב- W. הקבוצות W האלה הן "כיסוי" של המרחב, במובן שהאיחוד שלהם הוא כל המרחב, וכל 2 מהן זרות אחת לשנייה (כי יש לפחות סיבית אחת ב- Eq ששונה בין הקבוצות). השיטה שלנו להראות שהמרחק בין מילות קוד הוא חצי היא לחלק כל אחת מהקבוצות האלה ל- 2 חצאים:

$$W^{=} = \{w \in W : u \cdot w = v \cdot w\}$$

$$W^{\neq} = \{w \in W : u \cdot w \neq v \cdot w\}$$

אם נצליח במשימה הזו, התוצאה המיידית היא שבדיוק חצי מהמכפלות הפנימיות עם המרחב – כלומר חצי מהאינדקסים של הקוד – שווים בין $Had(u)$ ל- $Had(v)$, או במילים אחרות, הקוד הוא בעל מרחק קבוע של חצי בין כל שתי מילים.

כדי לבצע את החלוקה, אנחנו משתמשים במצבים הסטטיים שהגדרנו קודם. אנחנו רוצים לשים לב לנקודה הבאה: בכל מצב סטטי כזה, כאשר אנחנו מכפילים $u \cdot w$ ו- $v \cdot w$, ועברנו על חלק מהסיביות, אנחנו נמצאים באחת הקונפיגורציות הבאות[3]:

$$\begin{aligned}
&option\ 1\colon\ stat(u \cdot w) = 0, \qquad stat(v \cdot w) = 0\\
&option\ 2\colon\ stat(u \cdot w) = 0, \qquad stat(v \cdot w) = 1\\
&option\ 3\colon\ stat(u \cdot w) = 1, \qquad stat(v \cdot w) = 0\\
&option\ 4\colon\ stat(u \cdot w) = 1, \qquad stat(v \cdot w) = 1
\end{aligned}$$

שתיים מתוכן הן מצבים של שוויון (1,4) והשתיים האחרות הן מצבים של אי-שוויון (2,3).

על פי ההגדרות שלנו, הוקטורים u, v וכל W, כולם שווים בכל האינדקסים של Eq, ולכן אם נסתכל במצב הסטטי שלאחר הכפלת כל אותן הסיביות שב- Eq, אנחנו בקונפיגורציה של שוויון, לכל $w \in W$. נותר לנו להכפיל את כל הסיביות של Neq, ואנחנו יודעים ש- u ו- v *שונים* בכל אחת מהסיביות האלה. כשנכפיל את הסיבית הראשונה, יש בדיוק חצי מהוקטורים של W שמחזיקים 0 באותה הסיבית, והחצי השני מחזיק 1. בגלל השוני של u, v – כל הוקטורים עם 1 עוברים לקונפיגורציה סטטית של אי-שוויון, והאחרים נשארים כיוון שה- 0 לא משנה את ערכי המצבים הסטטיים. כאשר נכפיל את הסיבית הבאה, בדיוק חצי מהוקטורים שנשארו עם שוויון מחזיקים בה 1, ובדיוק חצי מהוקטורים שנמצאים באי-שוויון מחזיקים בה 1. כל אותם הוקטורים עוברים ממצב של שוויון למצב של אי-שוויון, או הפוך. כלומר אנחנו מחלקים את חצאי W לרבעים, ששניים מתוכם מחזיקים קונפיגורציות של שוויון והשניים האחרים אי-שוויונות. אנחנו יכולים להמשיך ולהכפיל את כל הסיביות ב- Neq, עד שנחלק את W ל- $2^{|Neq|}$ קבוצות – כלומר וקטורים יחידים. בכל חלוקה, אנחנו שומרים על בדיוק חצי מהוקטורים במצב סטטי של אי-שוויון, וכשנסיים – זוהי בדיוק התוצאה שחיפשנו, מרחק של חצי בדיוק בין מילות הקוד של u ו- v.

[3] הסימון $stat$ לא הוגדר לפני כן, והוא לא מוגדר גם עכשיו – אין סימון מתמטי כזה, זוהי פשוט הדרך שלי לומר "עצרנו את שתי המכפלות באותה הנקודה".

חלק ב – מערכות הוכחה הסתברותית ומשפט PCP.

בחלק א של העבודה, דיברנו על מחלקת הבעיות **IP**. הזכרנו שם את המושג מחלקות **PCP** ואמרנו שנתעסק בהן בהמשך – הגענו להמשך, ובדאי שנבין מהי בדיוק המשמעות של המילה.

PCP הוא קיצור של ***P**robabilistically-**C**heckable-**P**roof*, או "הוכחה ניתנת לבדיקה הסתברותית". כאשר דיברנו על מערכות מאמת מוכיח, התנסינו במכונות טיורינג שצריכות לבדוק איזושהי טענה מול אובייקט חיצוני (המוכיח), ולהגיע לתוצאה נכונה בהסתברות גבוהה. מכאן הדרך להגדרת ה- PCP קצרה – האובייקט החיצוני הוא לא מוכיח, אלא איזושהי הוכחה שצריך לבדוק. המחלקה PCP כוללת בתוכה שני פרמטרים חישוביים שהמחלקות הקלאסיות *P* ו- *NP* לא הכילו:

א. כמות הרנדומליות שמכונת חישוב מאפשרת. זה מושג שאנחנו מכירים מהמחלקות לפתרון בעיות הסתברותי, ומשמש גם בהוכחה הסתברותית. בעצם, המשמעות היא מספר הטלות המטבע שאנחנו מאפשרים למכונה, או אורך המחרוזת הרנדומלית שהיא מקבלת בתור קלט.
ב. כמות הגישה להוכחה. בהגדרה הקלאסית של המחלקה *NP*, אנחנו לא מוגבלים בפרמטר הזה, אבל אנחנו יכולים לגלות כל מיני דברים מעניינים אם ננסה לחקור את המשמעות של "הגבלת הגישה להוכחה" (משפט *PCP* למשל, הוא אחד הדברים המעניינים האלה).

אבל מה זה אומר "הגבלה על גישה להוכחה" – אם אני לא יכול לקרוא את כל ההוכחה, מי אני שאתיימר לבדוק אותה?

נניח שקיבלנו נוסחת 3SAT כלשהי ϕ. ביחד איתה קיבלנו גם את ההוכחה c, שאמורה להוות השמה תקינה כלשהי של ϕ. אנחנו יכולים לעבור על כל הפסוקיות של ϕ, ולוודא שכולן מסופקות על ידי אחת ההשמות למשתנה ב- c, כמו שהיינו עושים אם היינו רוצים לוודא ש- $3SAT \in NP$. אבל אנחנו נורא ממהרים, ואין לנו זמן לקרוא את כל ההוכחה. אנחנו רוצים לרפרף עליה, לקחת כמה נקודות, ובזמנינו הפנוי לעבור על הנקודות האלה ולוודא שהן הגיוניות. אז קיבלנו את ההוכחה ממישהו, רשמנו לעצמנו 3 השמות לליטרלים רנדומליים שמצאנו בהוכחה שלו, ושחררנו אותו לדרכו. כשחזרנו ל- ϕ, הסתכלנו בהשמות שרשמנו וגילינו שלפי ההוכחה $p_1 = 1,\ p_2 = 0,\ p_3 = 0$. אם נעבור עכשיו על ϕ ונגלה שאחת הפסוקיות בה היא $(p_1' \vee p_2 \vee p_3)$, נדע שמישהו מנסה לשקר לנו... וכיוון שהמוכיח לא יודע איזו שלשה של השמות רשמנו, הוא לא יכול להבטיח שהשלשה הזו לא מרכיבה את המשתנים של פסוקית כלשהי, ולא יכול למנוע לחלוטין את האפשרות שנעלה על התרמית. באופן סימטרי, אם לא מצאנו את הפסוקית המפלילה ב- ϕ, אנחנו יכולים להיות בטוחים קצת יותר שההוכחה תקינה – וקראנו רק 3 ביטים מתוכה!

למעשה, דרך הפעולה שלנו בהמשך תהיה דומה, עם הבדל מרכזי שהופך אותה ליעילה יותר. אנחנו לא נבחר 3 ליטרלים רנדומליים בהוכחה, אלא נבחר פסוקית רנדומלית מהנוסחה, ונסתכל בהשמות לאותם ליטרלים שמופיעים בה. מספר הגישות שלנו להוכחה נשאר אותו הדבר, אבל הפעם אנחנו *יודעים* שאם ההשמה לאותם המשתנים בעייתית, הפסוקית שההשמה מפריכה נמצאת בנוסחה.

משפחות ה- **PCP** השונות מנסות למצוא את הגבולות המתמטיים של אותן ודאויות חלקיות. אלה כמובן משליכות על ההבנה שלנו לגבי הבעיות ומודלי החישוב שאיתם נתעסק.

ועכשיו זמן טוב להכניס את האנוטציה הכבדה: משמעות הסימן $PCP_{c(n),s(n)}[r(n),q(n)]$ היא כזו:

א. *PCP* משמעו *Probabilistically-Checkable-Proof* או במילים שלנו: הוכחה עבור שייכות מילה w לשפה L, כאשר ההוכחה צריכה לעמוד במבחן תלוי-אקראיות כלשהו שמכונת טיורינג פולינומיאלית תייצר.
ב. n מבחינתנו מסמל את אורכה של מילת הקלט w, כלומר: $|w| = n$.
ג. $c(n)$ משמעו *completeness* (שלמות), או: כאשר ההוכחה נכשלה במבחן, מה מידת הביטחון שלנו שהמילה w באמת לא נמצאת ב- L. הדיון שלנו מתמקד במקרה שבו $c(n) = 1$, כלומר כל מילה ב- L, בוודאות תעבור את המבחן שלנו.

ד. $s(n)$ משמעו *soundness* (נאותות), ההגדרה המשלימה ל- *completeness*, או: כאשר ההוכחה **עברה** את המבחן, מה מידת הביטחון שלנו שהמילה w באמת ב- L. אנחנו נתמקד במקרה שבו $s(n) = \frac{1}{2}$, כלומר מילה שאינה ב- L תעבור את המבחן בהסתברות של לא יותר מחצי.

ה. $r(n)$ משמעו *randomness*, או: כאשר נתונה מילה w באורך n, מה מספר הטלות המטבע שהמכונה האקראית שלנו יכולה להטיל, כפונקציה תלויה ב- n. אפשר להתייחס למספר הזה בתור 'מידת האקראיות שאנחנו מאפשרים למכונה כאשר היא מקבלת קלט באורך n'. במילים אחרות, אפשר לומר שזהו האורך של המחרוזת הרנדומלית שהמכונה הדטרמיניסטית שלנו מקבלת, אם אנחנו לא רוצים להשתמש במושג 'הטלת מטבע'. אגלה כבר עכשיו שאני לא אוהב הטלות מטבע, ולכן נשתמש במחרוזות אקראיות במקומן.

ו. $q(n)$ משמעו *query*, או: מספר התווים מההוכחה שאנחנו מאפשרים למכונה שלנו לקרוא, כפונקציה של אורך w. כלומר, המכונה שלנו לא תקרא את כל ההוכחה, אלא תקבל גישה רק לחלקים מצומצמים ממנה, כפונקציה תלויה ב-n.[4]

כיוון שמבחינתנו השלמות והנאותות הדרושות הן תמיד 1 ו- 0.5, אנחנו נצמצם את הסימון שלנו כך שלא יכלול את c ואת s, כלומר: $PCP[r(n), q(n)]$.

הסימון שהגדרנו כרגע הוא סימון למשפחות של שפות, בדומה לסימונים P, NP, IP.

כאשר אנחנו אומרים $PCP[n^3, nlg(n)]$ אנחנו מתייחסים למשפחת כל השפות L שעומדות בתנאים שהגדרנו. כלומר: קיימת מכונה דטרמיניסטית ופולינומיאלית, כך שעבור כל מילה w ב- L, ניתן לכתוב הוכחה c, והמכונה – עם קלט c,w ומחרוזת רנדומלית באורך $|w|^3$ – קוראת $|w|\lg(|w|)$ תווים מ- c, ומחליטה לקבל את w. בנוסף, עבור כל מילה z שאינה ב- L, ***לא ניתן*** לכתוב את c, ועבור כל מחרוזת הוכחה c המכונה הדטרמיניסטית קוראת $|z|\lg(|z|)$ תווים מ- c (זהות התווים תלויה במחרוזת הרנדומלית), ודוחה את z בהסתברות של חצי לפחות.

עכשיו אנחנו מוכנים להכריז על המשפט המפוצץ הבא:

$$\boldsymbol{PCP\ theorem}:\ \ \boldsymbol{NP \subseteq PCP[O(lgn), O(1)]}$$

או: לכל שפה ב- **NP**, קיימת מכונה פולינומיאלית דטרמיניסטית, שמקבלת מילה w, הוכחה c ומחרוזת רנדומלית r באורך לוגריתמי ב- $|w|$, ומכריעה את השפה עם שלמות 1 ונאותות $\frac{1}{2}$ תוך קריאה של **מספר תווים קבוע מההוכחה, בלי תלות באורך w**.

במילים אחרות, ההתנהגות שלנו בדוגמה עם הנוסחה ϕ, כאשר הסתפקנו בשלושה ליטרלים בלבד, היא לאו דווקא עד כדי כך רשלנית. מסתבר שקריאת כמות קבועה של השמות תספיק עבור וודאות של לפחות 0.5 בנכונות ההוכחה. וכמו שאנחנו יודעים, ברגע שהגענו לנאותות של 0.5 ושלמות של 1, קל להגיע לנאותות נמוכה הרבה יותר באמצעות חזרה על חישובי המכונה שלנו מספר קבוע של פעמים.

משפט PCP נחשב לאחד המשפטים המרכזיים בתורת הסיבוכיות. ההוכחה המפורסמת הראשונה שלו הוצגה בשנות ה-90, והייתה תוצאה של סדרת מאמרים ועבודה של מספר חוקרים בתחום, ותשעה מתוכם אף זכו בפרס גדל לשנת 2001 בעקבות תרומתם. למשפט היו השלכות נרחבות בתחום הסיבוכיות של אלגוריתמי קירוב, שבחלקן נתעסק בהמשך. הוכחת המשפט נחשבת למסובכת במיוחד, וכתוצאה מכך חוקרים נוספים ניסו למצוא הוכחה קומבינטורית פשוטה יותר עבור אותה התוצאה. בשנות האלפיים פרופ' אירית דינור פרסמה מאמר (שגם הוא זכה בפרס גדל לשנת 2019) שבו היא מוכיחה את המשפט דרך השקילות שנמצאה בינו לבין הוכחת הקושי של קירוב בעיות מסוג **CSP**. החלקים הבאים של הסמינר יתעסקו בהוכחה המסוימת הזו. אנחנו לא נראה אותה בצורה ריגורוזית – אבל נגיע לא רחוק מזה, ובעיקר נקבל אינטואיציה לגבי שיטות העבודה בה.

[4] כדאי לשים לב שאפשר להתייחס ל- $q(n)$ בתור *מספר המקומות השונים שבהן ניתנת גישה להוכחה*, **ואז** נצטרך להגדיר גם כמה המכונה שלנו יכולה לקרוא בכל גישה. אנחנו נתייחס ל- $q(n)$ בתור מספר התווים הכולל שהמכונה קוראת.

קודם כל, עלינו להבין את הקשר בין המשפט לבין בעיות קירוב. לשם כך נציג את הבעיה **GAP-E3SAT**. כפי שניתן להבין מהשם, הבעיה עוסקת בנוסחאות 3CNF - אנחנו מכירים את 3SAT ואת גרסת האופטימיזציה Max-E3SAT, עכשיו נכיר את גרסת הקירוב.

בבעיית GAP-E3SAT, אנחנו מפרידים בין נוסחאות עם פתרונות טובים, לנוסחאות שאין להן פתרון טוב – פורמלית, אנחנו מתייחסים לפתרון האופטימלי OPT עבור נוסחת 3CNF נתונה (זה שמספק הכי הרבה פסוקיות), ואומרים על נוסחה בעלת m פסוקיות שהיא ב- $\text{GAP-E3SAT}_{c,s}$ אם OPT מספק לפחות cm פסוקיות, ושהיא לא בשפה אם OPT מספק פחות מ- sm פסוקיות. במקרה ש- OPT מספק מספר פסוקיות בין cm ל- sm, שייכות המילה לשפה אינה מוגדרת, ואלגוריתם שמכריע את GAP-E3SAT יכול להחזיר כן או לא, איך שמתאים לו.[5]

כדי להפוך את $\text{GAP-E3SAT}_{c,s}$ לבעיית קירוב סטנדרטית לגרסת האופטימיזציה, נקבע את $c = s$, ונקבל קבוע שחוסם מלמטה את מספר הפסוקיות המינימלי שעלינו לספק. אנחנו מעוניינים בגרסה אחרת, שבה השלמות שלנו $\boldsymbol{c}$ שווה ל-1.

הבעיה $\text{GAP-E3SAT}_{1,s}$ היא הבעיה שעבורה אלגוריתם צריך לדעת להבחין האם לפחות $(1-s)m$ מתוך הפסוקיות ב-ϕ **אינן** ניתנות לסיפוק בו זמנית. אם כל הנוסחה ספיקה, האלגוריתם יחזיר **כן**. אם פחות מ- sm פסוקיות ספיקות עבור כל השמה, הוא יחזיר **לא**. עבור כל נוסחה עם OPT שמספק חלק גדול מ- s וקטן מ- 1, אנחנו לא מגדירים מה חוזר. אם נראה שהגרסה הזו קשה, נוכל להראות שגם הגרסה עם $c = s$ קשה. אנחנו אפילו לא צריכים להגדיר רדוקציה, האלגוריתם שפותר את הגרסה הסטנדרטית פותר גם את הגרסה שלנו, אבל מחזיר **כן** עבור כל מקרה לא מוגדר.

ישנן גרסאות ***GAP*** לרוב הבעיות ה- NP שלמות, ובכולן הרעיון הוא לקבוע את s ואת c כך שהפרמטר המשמעותי בבעיה, בין אם זה מספר פסוקיות, גודל של קליקה, כיסוי צמתים או פרמטר אחר, ניתן להבדלה בין המקרים "יש כזה טוב יותר מ- c" ו- "אין כזה טוב יותר מ- s".

אנחנו יודעים מה אומר משפט PCP**,** והגדרנו את $\text{GAP-E3SAT}_{1,s}$. הקשר ביניהם מופיע בשקילות הבאה:

משפט PCP נכון אם ורק אם קיים s קטן (ממש) מ- 1 כך ש- $\text{GAP-E3SAT}_{1,s}$ היא בעיה NP קשה.

כלומר, אם המשפט נכון, קיים איזשהו קבוע **אוניברסלי** s כך שהקביעה האם מספר הפסוקיות המקסימלי שניתנות לסיפוק בנוסחת 3CNF נתונה גדול מ- sm, היא בעיה NP קשה. אנחנו יודעים שהקבוע קיים כי אנחנו יודעים שמשפט PCP הוכח. אבל אנחנו נעבוד בכיוון ההפוך – נוכיח שהקבוע קיים, ומכך נסיק את המשפט.

המשמעות של בעיית *GAP* שהיא NP קשה, היא שכל שפה L ב- NP יכולה לעבור רדוקציה פולינומיאלית כך שלכל $w \in L$, הרדוקציה של w לבעיית ה- *GAP* בעלת פתרון טוב יותר מ- c. לכל מילה אחרת, הרדוקציה מחזירה נוסחה שהפתרון הטוב ביותר עבורה גרוע יותר מ- s. במקרה של $\text{GAP-E3SAT}_{1,s}$,המשמעות היא כמובן שלאחר רדוקציה נקבל נוסחה ספיקה, או נוסחה שכל השמה לה נכשלת לפחות ב- $(1-s)m$ פסוקיות[6].

כל זה נחמד מאוד, אבל שקילויות צריך להוכיח:

<u>כיוון 1</u>: $\text{GAP-E3SAT}_{1,s}$ היא *NP* שלמה $\Longleftarrow$ משפט PCP

זהו הכיוון הפשוט יותר. נניח שיש לנו s אוניברסלי שעבורו $\text{GAP-E3SAT}_{1,s}$ היא NP קשה.

יש לנו בעיה $L \in \text{NP}$, ואנחנו רוצים להראות שמשפט PCP תקף לגביה – כלומר קיימת לה הוכחה שממנה נצטרך לקרוא רק מספר קבוע של תווים ולפיהם נוכל לקבל בוודאות של 1/2 .

[5] טכנית, גם כאן בהגדרת GAP-E3SAT, הקבועים c ו- s מייצגים שלמות ונאותות – אבל כאן מסתתר דיון על המטא של נכונות, ואנחנו לא ניפול במלכודת הזו.

[6] מעניין לשים לב להנחה הסמויה שטמונה בהגדרה כזו של NP שלמות – אנחנו מגדירים שאלגוריתם **פותר** את בעיית ה-GAP רק אם הוא חומק מאותו תחום הפכפך בין s ל- c, שבו הוא יכול להיות נכון בלי לדעת את התשובה. לכן כדי להוכיח שכל בעיה ב- NP קשה לפתרון לפחות כמו בעיית ה- GAP, הרדוקציה לא יכולה להשתמש באותו התחום כדי להיפטר ממופעים קשים של הבעיה.

מה אנחנו מקבלים: מופע של L, והשמה לנוסחת *3CNF*.

מה אנחנו עושים עם זה: כיוון ש- $L \in \mathrm{NP}$, ו- $\text{GAP-E3SAT}_{1,s}$ היא NP שלמה, נעביר את המופע שקיבלנו לנוסחת *3CNF*. ההשמה שקיבלנו היא עבור הנוסחה הזו.

הרדוקציה ל- $\text{GAP-E3SAT}_{1,s}$ מבטיחה לנו שאם המופע של L הוא אכן מילה בשפה, הנוסחה המתקבלת מהרדוקציה ספיקה לחלוטין.

אחרת, יש לנו חלק בגודל קבוע (*1-s*) מהפסוקיות שההוכחה שלנו - אותה השמה שקיבלנו עם המופע - בוודאות אינה מספקת. כתוצאה מכך, לבחירה רנדומלית של פסוקית בנוסחה יש הסתברות קבועה לכישלון, וחזרה על המבחן מספר קבוע של פעמים תברר בוודאות של לפחות $1/2$ שהמופע שקיבלנו הוא אכן מילה ב- L. כיוון שבכל מבחן אנחנו קוראים רק שלושה ליטרלים מההוכחה, מספר הקריאות שלנו קבוע.

<u>כיוון 2</u>: משפט PCP $\Leftarrow$ קיים s כך ש- $\text{GAP-E3SAT}_{1,s}$ היא NP קשה.

אנחנו רוצים להוכיח ששפה כלשהי היא NP קשה. מאז שנות השבעים השיטה שלנו לא השתנתה - רדוקציה משפה NP קשה שאנחנו כבר מכירים. המקרה הזה לא שונה, מלבד העובדה שיש לנו את משפט PCP המפלצתי לנצל.

אנחנו נבצע רדוקציה מ- 3COLOR אל $\text{GAP-E3SAT}_{1,s}$ בצורה הבאה:

משפט PCP מבטיח לנו שאנחנו יכולים לקבל מחרוזת אקראית באורך לוגריתמי והוכחה עבור 3COLOR**,** לבחור כמות קבועה של תווים מההוכחה, ובהסתברות של 1/2 לדחות כל גרף שאינו 3-צביע.

אנחנו נהיה חמדנים במיוחד, ונחליט לקרוא את כל הצירופים האפשריים עבור התווים האלה, כלומר: קיבלנו מחרוזת אקראית באורך לוגריתמי, ויש לה *רק מספר פולינומיאלי של קומבינציות אפשריות*. כל קומבינציה כזו מגדירה בחירה של תווים אקראיים בהוכחה[7], ואיזשהו הכרח לוגי שהמכונה שלנו מחשבת על היחסים בין התווים האלה – המבחן שלנו. את ההכרח הזה אנחנו יכולים להציג באמצעות מספר קבוע של פסוקיות 3CNF, שמבטאות אילו צירופים של תווים אלה עוברים את המבחן, ואילו צירופים יכשילו את המבחן לאחר קריאתם.

משפט PCP מבטיח לנו ***שלפחות חצי מהנוסחאות הללו יכשילו את המבחן***. זאת אומרת שלפחות חצי מהנוסחאות מכילות פסוקית אחת לפחות שאינה ניתנת לסיפוק ביחד עם שאר הנוסחה, ולכן אחת מתוך מספר קבוע של פסוקיות (מספר הפסוקיות המקסימלי בנוסחה) אינה ספיקה – עבור חצי מהנוסחאות. איחוד של הנוסחאות הללו לנוסחת 3CNF יחידה הוא הרדוקציה שחיפשנו עבור $\text{GAP-E3SAT}_{1,s}$**.**

בסך הכל: נתונה לנו מכונת טיורינג M**,** פולינומיאלית והסתברותית שמקבלת מופע של 3COLOR, והוכחה כלשהי, ביחד עם מחרוזת אקראית באורך $O(lgn)$. אנחנו בונים מכונה שעוקבת אחר החישוב של M עבור כל פרמוטציה של המחרוזת האקראית (בסך הכל חישוב פולינומיאלי, כפול פולינום של פרמוטציות), ואז מרכיבה נוסחת 3CNF בגודל קבוע m לפי הפרמוטציה. איחוד כל הנוסחאות הללו יוצר מופע של $\mathrm{GAP-E3SAT}_{1,\frac{1}{2m}}$, כאשר אם הגרף צביע, הנוסחה ספיקה, ואחרת – אחת מכל $2m$ פסוקיות היא שקרית בכל השמה.

[7] בדרך כלל נהוג לחשוב על המחרוזת הרנדומלית ממש בתור מספר קבוע של אינדקסים לקריאה מההוכחה, ואנחנו צריכים שכל אחד יהיה באורך לוגריתם בבסיס 2 של ההוכחה, כדי להיות מסוגלים להגיע לכל חלק ממנה – פורמלית, אנחנו לא מכריחים את המחרוזת להתנהג כך.

חלק ג – קצת רקע מסביב (גרפים מרחיבים, מבחני השמה)

חלק ג הוא בעצם שני חלקים – הראשון מתעסק בגרפים מרחיבים, השני במבחני השמה, ואין ביניהם ממש קשר חוץ מהעובדה שנצטרך להכיר את שני המושגים בהמשך. אבל, יהיה קשה מאוד להבין אותם בלי להכיר את הסיבה שאנחנו משתמשים בהם, ולכן נרצה קודם כל לראות סקירה כוללת מאוד של הוכחת משפט ה- PCP. אז מה אנחנו עושים פה בעצם? הזכרנו בקצרה שנשתמש בבעיות **CSP** כדי להוכיח את המשפט. השקילות שהראינו בחלק הקודם אומרת **GAP-E3SAT** היא NP קשה אם"ם $NP \subseteq PCP[O(lgn), O(1)]$. בעקרון, זו לא תוצאה שצריכה להפתיע אותנו במיוחד – אנחנו פחות או יותר אומרים:

"אם כל בעיה ב- NP ניתנת לייצוג בתור GAP-E3SAT, כלומר בעזרת לוגיקה פסוקית עם אחוז קבוע של סתירות פנימיות, אנחנו יכולים לבדוק את הטענה הלוגית בצורה סטטיסטית די מהר – מצד שני, אם אנחנו יכולים לבדוק את הטענה מהר, אנחנו יכולים לסדר את הבדיקות שלנו בתור טענות לוגיות ששקולות לבעיה עצמה." אלו בדיוק הצעדים שביצענו בהוכחת השקילות.

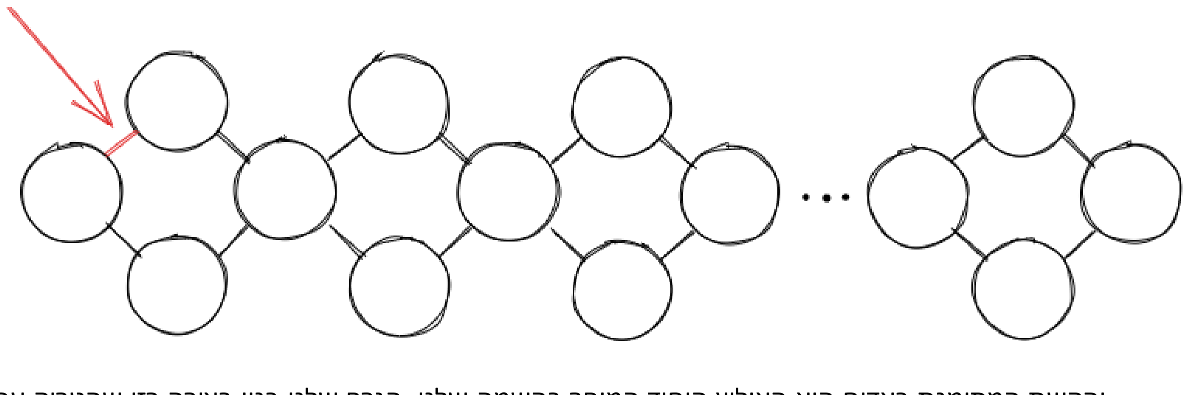
אבל עכשיו אנחנו אומרים שהטענות אינן רק שקולות, אלא גם נכונות – וזו טענה אחרת. עכשיו אנחנו אומרים:

"הבעיות הקשות ביותר ב- NP בדרך כוללות איזשהו מרכיב של תלויות פנימיות, שאנחנו לא מצליחים לתאר בצורה כללית והיררכית כלשהו (כלומר לכתוב אלגוריתם שמצליח לפסול קבוצות גדולות של פתרונות בעייתיים מהר). אנחנו יכולים (מהר, על ידי רדוקציה פולינומיאלית) לסדר מחדש את התלויות האלה כך שהן 'חסרות מצבי קצה' – הסתירות הפנימיות שהן מכילות (אם יש כאלה) יהיו באותו סדר גודל של הבעיה עצמה עבור כל פתרון אפשרי".

אנחנו גם רמזנו שאירית דינור מוכיחה את המשפט על ידי הצגת קבוע גלובלי כזה כך שבעיית ה- **GAP-CSP** היא NP קשה. איך נוכיח כזה דבר? על ידי רדוקציה, duh... אנחנו נראה רדוקציה פולינומיאלית מ- **CSP** אל **GAP-CSP**[8]. אבל למה דווקא CSP? הבעיה CSP היא כמעט תיאור מילולי של "תלויות פנימיות לא מסודרות" – יש משתנים, ויש עליהם אילוצים, וזהו. זה מאפשר לנו שיטה די פשוטה (כמעט גסה) לרדוקציה. אנחנו הופכים אילוצים לקשים יותר.

בעצם, כל מה שאנחנו צריכים לעשות זה לוודא שאם אילוץ אחד מופר, הוא יגרום להרבה אילוצים אחרים להישבר יחד איתו. הדרך הפשוטה ביותר לעשות מניפולציה שכזו היא לומר לכל אילוץ: "אתה נשאר ההכרח הלוגי שאתה, אבל גם צריך שכל השכנים שלך יסופקו, אחרת אתה נשבר ביחד איתם" – כלומר המשתנה שלנו הופך למשתנה של סדרת השכנים שלו, והאילוצים עליו כוללים את כל האילוצים על קבוצת המשתנים שהוא מכיל. בהנחה שכל פעם שמבצעים את הרדוקציה הזו, מספר האילוצים המופרים גדל פי איזשהו קבוע, לאחר מספר לוגריתמי של רדוקציות כאלה נקבל גידול פולינומיאלי של ה- **GAP** שלנו, והמספר המינימלי של הפרות אילוצים יתקרב לקבוע מתוך הגרף הכולל. כמו תמיד, הדרך הפשוטה היא לא באמת עד כדי כך פשוטה והשיטה שתיארנו מאוד לא מדויקת. איך בדיוק זה קורה נראה בהמשך, אבל כרגע מעניינות אותנו שתי הבעיות הבאות:

א. נניח שהגרף שלנו נראה ככה:



והקשת המסומנת באדום היא האילוץ היחיד המופר בהשמה שלנו. הגרף שלנו בנוי בצורה כזו שהטריק עם השכנים לא ממש יעבוד – הגידול ב- GAP יהיה לינארי במספר הרדוקציות, כי הקשת הבעייתית רחוקה מאוד

---

[8] GAP-CSP זו השפה של בעיות אילוצים עם אחוז קבוע של אילוצים מופרים. היא מוגדרת באופן דומה לאיך שהגדרנו את GAP-E3SAT.

מרוב הקשתות האחרות בגרף, וקבוצת האילוצים השכנים שלה יחסית מבודדת – למעשה, בחרנו להסתכל על מקרה קצה קיצוני במיוחד, שבו כל קבוצות השכנים של כל הקשתות בגרף יחסית מבודדות, אבל הוא מראה לנו מה קורה כאשר יש בגרף שלנו קבוצה "מסוגרת" של צמתים/קשתות – יכול להיות ש'רדוקציית השכנים' שלנו תהיה הרבה פחות יעילה ממה שתכננו. אנחנו רוצים לדחוף אילוצים שבורים לעבר כל הגרף, אבל צווארי בקבוק שכאלה מקשים עלינו מאוד.

ב. נניח שביצענו את הרדוקציה. עכשיו כל צומת שאנחנו צריכים לתת לה השמה, צריכה לתת השמה לעוד מספר קבוע של צמתים. כלומר, אם התחום המקורי של הצומת היה למשל $\{A, B, C, D\}$, והיא צריכה לתת השמה לעוד 7 צמתים עם תחום דומה, התחום החדש של הצומת יהיה הסדרות באורך 7 מעל התחום המקורי, או $\{A, B, C, D\}^7$ – התחום גדל בחזקה של קבוע. לאחר מספר לוגריתמי של רדוקציות כאלה, אנחנו נקבל שהתחום עבור כל צומת הוא בגודל $O\left(4^{\left(7^{logn}\right)}\right)$ – הסוגריים במקום הנכון, בכל איטרציה של הרדוקציה אנחנו מגדילים את התחום האפשרי פי 7 ממה שהיה **באיטרציה הקודמת**, ולא פי 7 ביחס לתחום המקורי. אם לא נטפל בגדילת התחומים, נסיים עם בעיה בגודל על-פולינומיאלי ביחס לקלט המקורי.

גרפים מרחיבים יעזרו לנו לפתור את הבעיה הראשונה, מבחני השמה וצמצום אלף-בית יעזרו לנו לפתור את השנייה.

גרפים מרחיבים – expander graphs

אז מה זה בעצם גרף מרחיב? התשובה הריגורוזית היא, שאין באמת דבר כזה – יש גרף שהוא מרחיב יותר, וגרף מרחיב פחות. ההגדרה המתמטית[9] קשורה למידת ההרחבה של גרף, ולא על היותו כן מרחיב או לא. גרף מרחיב הוא פשוט כינוי נוח לגרף **מרחיב מאוד**.

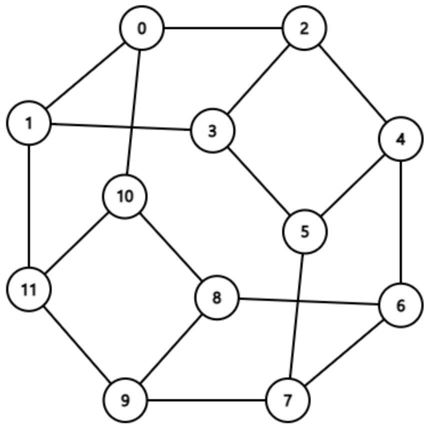
אוקיי, נעדכן את השאלה – מה זה גרף מרחיב מאוד? באופן גס, מידת ההרחבה של הגרף מודדת כמה קל להגיע מצומת אחת לצומת אחרת. מנקודת מבט אחרת, ההרחבה מודדת כמה קל לפרק את הגרף שלנו לחלקים, או כמה צווארי בקבוק יש לנו. המושג שאנחנו מגדירים נקרא **הרחבת-קשתות**, או **edge-expansion**, ומוגדר כך:

> ניקח את קבוצת הצמתים V בגרף $G = (V, E)$, ונחלק אותה לשתי קבוצות, לאו דווקא בגודל שווה. הקבוצה הקטנה תיקרא S והגדולה S'. לכל הקשתות שמחברות צומת מ- S אל צומת ב- S' נקרא $E(S, S')$[10].
>
> אנחנו נבצע את התהליך עבור כל החלוקות האפשריות של V, ולכל אחת מהן נחשב את היחס בין מספר הקשתות החוצות (כלומר $|E(S, S')|$), לבין גודל S. הערך המינימלי שנקבל הוא הרחבת הקשתות ϕ:

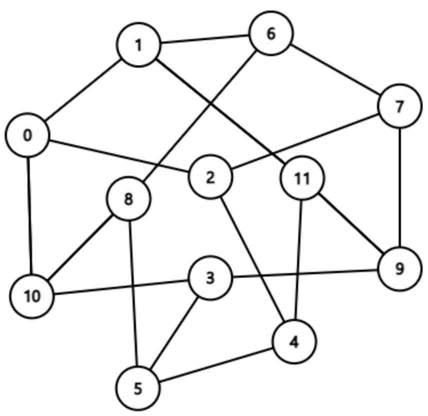
$$\phi(G) = \min_S \left(\frac{|E(S, S')|}{|S|} \right)$$

במילים אחרות, אנחנו עוברים על כל תת קבוצה של צמתים בגרף (בתנאי שאינה מכילה את רוב הגרף), ובודקים כמה קשתות יוצאות ממנה ביחס לגודל הקבוצה. על הערכים האלה אנחנו מחפשים מינימום – אם המינימום גדול, אז אי אפשר למצוא קבוצה של צמתים שלא "קשירה היטב" עם שאר הגרף, והגרף שלנו מרחיב מאוד.

מה זה אומר לנו? נניח שמצאנו את אותה חלוקה שנותנת את $\phi(G)$. S היא הקבוצה עם הקשירות הנמוכה ביותר לצמתים חיצוניים לה, אבל אם הגרף שלנו מרחיב מאוד, גם היא תהיה קשירה היטב לשאר – וקבוצת השכנים המיידיים שלה עוד יותר קשירה! כלומר, גם הצומת הכי נידח ב- G, יכול להגיע על מסלול באורך 1 או 2 ל'איזורים' רבים בגרף.



גרף מרחיב פחות (עם הרחבה 2/3)

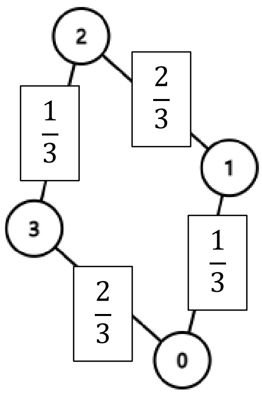


גרף מרחיב יותר (עם הרחבה 1 - אני יודע, כי בדקתי בעזרת מחשב)

כאן יש לנו שני גרפים, אחד יפה עם מבנה מרחבי ברור, ושני מכוער עם הרחבת קשתות גדולה. דרך נוספת להבין את הגרפים האלה היא במונחים של אנטרופיה - הגרף היפה אוצר יותר מידע, כי המבנה שלו פחות טבעי מאשר זה של הגרף המרחיב - תכונה מעניינת נוספת של גרפים מרחיבים היא ש-"גרף רנדומלי הוא מרחיב", כלומר גרף שבו התפלגות הקשתות קרובה לאחידה בין זוגות צמתים, יהיה בעל הרחבת קשתות גדולה בתוחלת.

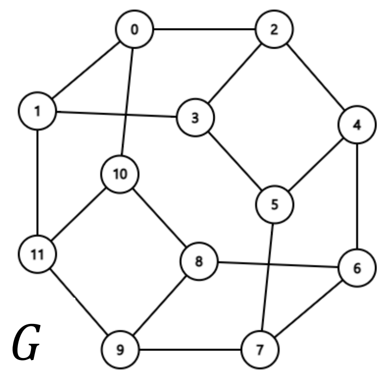
[9] יש כמה הגדרות ל"הרחבה", אנחנו נשתמש באחת פופולרית במיוחד.
[10] אפשר גם לומר חתך, אם זה יותר נוח...

אחת הדרכים המעניינות יותר להסתכל על תכונת הקשירות הזו (שהיא – לא במקרה – הסיבה העיקרית לדיון שלנו מלכתחילה), היא דרך עדשה של **מסלולים אקראיים – random walks**. במסלולים אקראיים אנחנו מתחילים מצומת מקור כלשהי, ובכל צעד הצומת הבא במסלול תלוי בהסתברות הבחירה בקשת המובילה אליו.

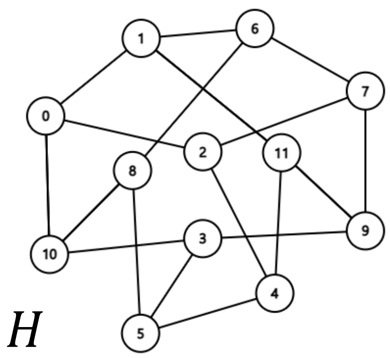


בגרף למעלה הקשתות ממושקלות בהתאם להסתברות הבחירה בהן בתור צעד. מסלול באורך 1 מצומת **0** מגיע בהסתברות $\frac{1}{3}$ לצומת **1**, ובהסתברות $\frac{2}{3}$ לצומת **3**. מסלול באורך 2 יגיע בהסתברות $\frac{4}{9}=\frac{2}{3}*\frac{2}{3}+\frac{1}{3}*\frac{1}{3}*\frac{2}{3}$ לצומת **2**, ובהסתברות $\frac{5}{9}=\frac{1}{3}*\frac{1}{3}+\frac{2}{3}*\frac{2}{3}$ בחזרה לצומת **0**.

כמובן שבבניית גרפים עם התפלגויות על המסלולים, חובה לדאוג שמכל צומת, ההסתברויות ליציאה מסתכמות ל- 1. בגרפים מסובכים זה יכול להיות קשה, או אפילו בלתי אפשרי אם הקשתות אינן מכוונות. אנחנו לא צריכים להתמודד עם הבעיה הזו, כיוון שהדיון שלנו נסוב סביב **גרפים רגולריים** (או באנגלית **regular graphs**) – שבהם דרגת כל הצמתים שווה. הגרף למעלה למשל הוא 2-רגולרי, כי מכל צומת יוצאות (או נכנסות) בדיוק 2 קשתות. השימוש בגרפים רגולריים מאפשר לנו להגדיר התפלגות אחידה על הקשתות, שבה לכל קשת יש ערך הסתברות הופכי לדרגת הגרף – בגרף 4-רגולרי ההסתברות תהיה $\frac{1}{4}$ וכן הלאה – וסך ההסתברויות של הקשתות בכל צומת יסתכם ל- 1. ובהפגנה של תחכום ייחודי שאני גאה בה מאוד, התמונה שראינו:



גרף מרחיב פחות (עם הרחבה 2/3)



גרף מרחיב יותר (עם הרחבה 1 - אני יודע, כי בדקתי בעזרת מחשב)

מכילה שני גרפים 3-רגולריים. זה חשוב, כי נשתמש בה כדי להסביר את הקשר בין הרחבת קשתות של גרף, ובין המטריצה המייצגת אותו (ובעיקר – הערך העצמי השני שלה). קודם כל, כדי שנוכל להמשיך, כדאי שניתן שמות לגרפים שלנו, אז מעבשיו הימני הוא G והשמאלי הוא H.

אנחנו נשרה על הקשתות שלהם התפלגות אחידה, והמטריצות המייצגות, בהתאמה, הן:

(אני לא מצפה מאף אחד להתעמק במטריצות האלה, הן לא כאן בשביל שיסתכלו עליהן, אלא בשביל החישובים שיגיעו בהמשך)

$$G=\begin{bmatrix}
0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0\\
\frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3}\\
\frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3}\\
\frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3}\\
0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0
\end{bmatrix}$$

$$H=\begin{bmatrix}
0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0\\
\frac{1}{3} & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & \frac{1}{3}\\
\frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0\\
0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & \frac{1}{3}\\
0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0\\
0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0\\
0 & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0\\
0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0\\
0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3}\\
\frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0\\
0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0
\end{bmatrix}$$

כדי לבחון מסלולים בגרף בעזרת מטריצה מייצגת, אנחנו צריכים לבחור צומת התחלה כלשהו, ולייצג אותו בעזרת וקטור מהבסיס הסטנדרטי שמכיל 1 באינדקס המתאים ו- 0 בכל השאר (למשל, הצומת **0** מיוצג על ידי הוקטור $(1,0,0,0,0,0,0,0,0,0,0,0)$). כשנכפיל את המטריצה בוקטור הזה, וקטור התוצאה יהיה וקטור העמודה המתאים במטריצה, כלומר התפלגות ההסתברות על הקשתות היוצאות מצומת (וקטור) זה, ובהתאם התפלגות צמתי היעד במסלול אקראי באורך 1. כדי למצוא את ההתפלגות על מסלולים באורך 2, אנחנו צריכים להכפיל שוב את המטריצה

בתוצאה (וכך נכפיל את הסתברות היציאה מכל צומת שכן, בהסתברות להגעה אל אותו צומת שכן מלכתחילה), וכן הלאה עד מסלול באורך n. אנחנו נבחן את ההתפלגויות על מסלולים באורך 1, 2 ו- 3 מצומת אקראי בגרפים (G, H), ונראה כיצד הגרף המרחיב יותר מוביל להתפלגות אחידה יותר – כיוון שהוא הרבה יותר קשיר.

אז קודם כל, אנחנו צריכים צומת אקראי. לשם כך נלחש את מילות הקסם[11] [12]:

"Import randomo,

Nodus equlaes randrange twelvus,

Imprintus Nodus"

התוצאה שהתקבלה היא 11, ואת החישוב אני אחסוך מכם לטובת הצגת התוצאות (אם כבר מדברים על קסמים, החישובים התבצעו בעזרת הספרייה *SymPy*):

$$Gx = \begin{pmatrix} 0 \\ 1/3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1/3 \\ 1/3 \\ 0 \end{pmatrix}, \quad GGx = \begin{pmatrix} 2/9 \\ 0 \\ 0 \\ 1/9 \\ 0 \\ 0 \\ 0 \\ 1/9 \\ 2/9 \\ 0 \\ 0 \\ 1/3 \end{pmatrix}, \quad GGGx = \begin{pmatrix} 0 \\ 2/9 \\ 1/9 \\ 0 \\ 0 \\ 2/27 \\ 1/9 \\ 0 \\ 0 \\ 2/9 \\ 7/27 \\ 0 \end{pmatrix}$$

$$Hx = \begin{pmatrix} 0 \\ 1/3 \\ 0 \\ 0 \\ 1/3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1/3 \\ 0 \\ 0 \end{pmatrix}, \quad HHx = \begin{pmatrix} 1/9 \\ 0 \\ 1/9 \\ 1/9 \\ 0 \\ 1/9 \\ 1/9 \\ 1/9 \\ 0 \\ 0 \\ 0 \\ 1/3 \end{pmatrix}, \quad HHHx = \begin{pmatrix} 1/27 \\ 5/27 \\ 2/27 \\ 1/27 \\ 5/27 \\ 1/27 \\ 1/27 \\ 2/27 \\ 2/27 \\ 5/27 \\ 2/27 \\ 0 \end{pmatrix}$$

וכמו שציפינו, ההתפלגות על הגרף המרחיב H אחידה הרבה יותר. אני עצמי הופתעתי כשראיתי שבגרף H, הצומת היחיד שלא נמצא במרחק 3 קשתות בדיוק מצומת **11** הוא הצומת **11** עצמו...

[11] פסודו-קוד בפסודו-לטינית, לזה לא ציפיתם...
[12] אם במקרה הגעתם לכאן תוך כדי חיפוש חומר על הוכחות באפס ידע – אני יודע, הציפייה שתסמכו עלי שהמספר רנדומלי באמת היא נושא כאוב במיוחד. ובכלל – השימוש במחולל לא קריפטוגרפי מעורר בכם גיחוך ו/או חלחלה. אף על פי כן, אם כבר קראתם עד לכאן, כולי תקווה שתסלחו לי על כמה פשרות בנושאי אנטרופיה.

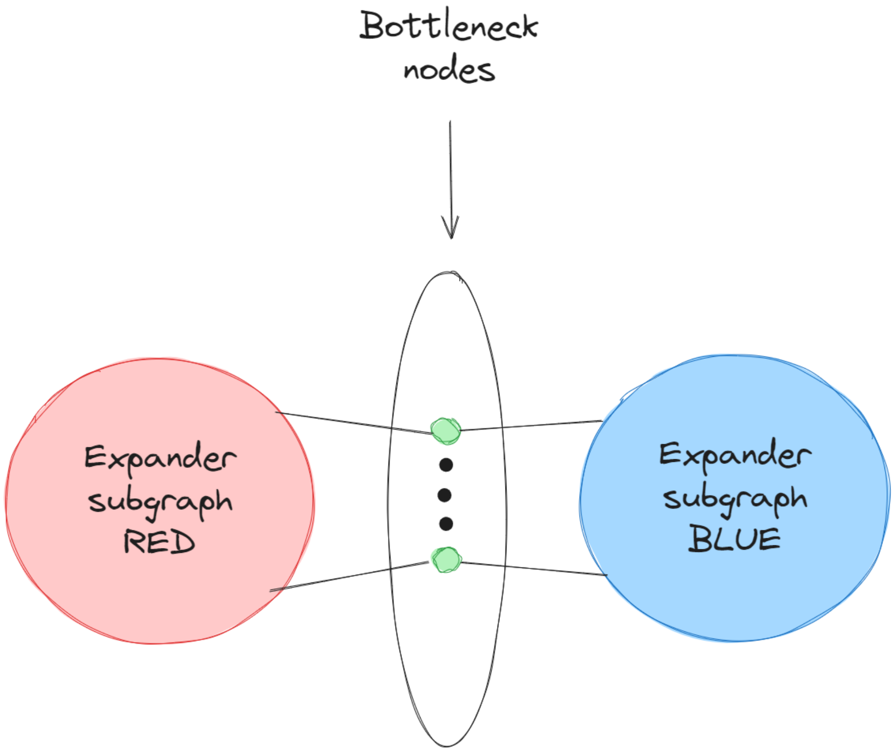
עבור המשך הדיון נתעלם מהאלמנט ההסתברותי שהצגנו עד עכשיו, ונתייחס למטריצות H ו- G בצורה הקלאסית – זאת אומרת, למטריצות יש ערך 1 בכל אינדקס שבו קיימת קשת (או - הכפלנו את המטריצות בסקלר 3).

בשלב הזה אנחנו כבר מבינים את הטריק: הכפלה של המטריצה המייצגת בעצמה n פעמים, ואז בוקטור צומת, סופרת את המסלולים באורך n המגיעים לכל צומת בגרף ושמה את הערך המתאים באינדקס המתאים לצמתי היעד. השימוש הקודם שלנו בטכניקה הזו היווה הכללה הסתברותית לספירה כזו, אבל התוצאה האמיתית שמעניינת אותנו היא היכולת להגדיר גרפים מרחיבים מחדש, באמצעות ערכים עצמיים של המטריצה המייצגת.

יש ערך עצמי אחד של גרף רגולרי שקל מאוד לחשב – עבור הוקטור $\vec{1}$, שבו הכפלה במטריצה סופרת את כל הקשתות היוצאות מכל הצמתים בגרף, אותה הכפלה גם סופרת את כל המסלולים באורך 1 **הנכנסים** לכל צומת. עבור גרף M d-רגולרי, המספר הזה הוא כמובן קבוע, ולכן $M(\vec{1}) = \vec{d}$, כלומר d הוא ערך עצמי המתאים לוקטור היחידה.

אנחנו מתייחסים למטריצה בתור 'טרנספורמציית מסלולים' לינארית על וקטור שמייצג תת קבוצה של צמתים. הטרנספורמציה היא על המרחב $\mathbb{R}^n$, ולכן הוקטור שלנו מייצג קונספט קצת יותר מורכב מתת-קבוצה של צמתים – הקבוצה יכולה להכיל חלקי צמתים, צמתים גדולים במיוחד או צמתים שליליים. לא קשה להשתכנע שאין ערכים עצמיים גדולים יותר מ- d, שהרי וקטור עצמי אחר של המטריצה, חייב לאזן ערכים חיוביים ושליליים כדי לשמור על אחידות בגדילת איברי הוקטור.

עכשיו, הגרפים H ו- G שראינו קטנים יחסית, ולכן לא ניכרות בהם תכונות של הרחבה (או חוסר של כאלו) באופן בולט מאוד[13]. אבל אנחנו נדמיין את הגרף ה-d-רגולרי K, שהוא גם גדול מאוד, וגם "ניתן לפירוק" – יש בו שני תתי גרפים, האדום והכחול, שבפני עצמם הם מאוד מרחיבים, אבל החיבור ביניהם הוא חלש יחסית, ומהווה צוואר בקבוק עבור מעבר מסלולים בין צמתים אדומים וכחולים:



אנחנו רוצים למצוא את הערך העצמי השני בגודלו בגרף, ולכן נתאר את הוקטור הבא:

כל הצמתים הכחולים מקבלים את הערך 1, כל הצמתים האדומים את הערך 1-, ואת ערכי קבוצת ה- *Bottleneck* אנחנו עוד לא קובעים.

[13] שיקרתי קצת: ניסיתי לחשב את הערכים העצמיים של G, H (שוב בעזרת $SymPy$), וקיבלתי שיעור מתסכל מאוד בנושא https://en.wikipedia.org/wiki/Casus_irreducibilis ...

עכשיו נעביר את הוקטור טרנספורמציית מסלולים – הכפלה במטריצה המייצגת. רוב הצמתים בגרף הכחול מקבלים את הערך d, ורוב אלה שבגרף האדום מקבלים את הערך $-d$. בכל גרף ישנה קבוצה קטנה של צמתים עם קשת החוצה לקבוצת *Bottleneck*, שערכם קצת יותר קטן. אנחנו יכולים לכוונן את הגרפים כך שבכל גרף, קבוצת הגבול היוצאת תקבל ערך קצת יותר קטן מ- 1, והערך של צומת כזו לאחר הטרנספורמציה יהיה כפול d ממה שהיה לפני. כמובן, זה אומר שערכי הצמתים השכנים לצומת קטנים יותר מ- d, כי אחת מהכניסות אליה קטנה מ- 1. גם את הצמתים האלה נצטרך לתקן, וגם את השכנים שלהם, עד שנכסה את הגרפים הכחול והאדום. בסופו של דבר, אפשר לתקן את ערכי הוקטור ההתחלתיים כך שצמתים אדומים וכחולים כולם גדלים בצורה אחידה לאחר הטרנספורמציה. קבוצת ה-*Bottleneck*, מקבלת מסלולים חיוביים מהגרף הכחול, ושליליים מהגרף האדום – בסך הכל, המסלולים האלה פחות או יותר מנטרלים את עצמם. אנחנו יכולים לתת לצמתים בצוואר הבקבוק ערכים קטנים מאוד, כך שהגדילה שלהם גם כן קבועה ביחד עם שאר הגרף (כמובן, זה דורש הכנסה של עוד תיקונים לגרף הכחול והאדום).

מה שעשינו בעצם, זה "לתקן את הזליגה" של שני תתי גרפים יחסית סגורים. בתוך כך, אנחנו מקבלים וקטור עצמי נוסף, שהערך העצמי המתאים לו קרוב מאוד ל- d, הערך הגדול ביותר.

הגרף K שראינו לא מאוד מרחיב - אבל M, עוד גרף מאוד גדול *ומאוד* מרחיב, הוא כל כולו זליגה אחת גדולה – ולכן הערך העצמי הבא בגודלו (אחרי d) ב- M יהיה קטן בהרבה. כבר אין לנו את האופציה לתיקונים קטנים, כדי שההכפלה במטריצה תהיה דומה להכפלה בסקלר, ניאלץ לשתף את *כל* הצמתים במידה פחות או יותר שווה, עם ערכים חיוביים ושליליים שמנטרלים זה את זה. מכאן אנחנו לומדים שלגרפים מרחיבים יש "מרווח ספקטרלי" גדול – כלומר ההפרש בין הערך העצמי הגדול (בערך מוחלט) d לזה הבא אחריו גדול יותר ככל שהגרף מרחיב יותר (כלומר ככל שיותר קשה לאפיין את אותן תת-קבוצות "מכונסות בעצמן" של צמתים).

אז עכשיו יש לנו דרך נוספת להגדיר הרחבה של גרף:

גרף d-רגולרי G עם ערכים עצמיים $\lambda_1 \cdots \lambda_n$, והערך $\lambda = \max\{|\lambda_i|: 1 \le i \le n \ and \ \lambda_i \ne d\}$ (הערך העצמי השני בערכו המוחלט), נקרא $(n, d, \lambda) - expander$. הערכים האלה מהווים אפיון טוב למידת ההרחבה של הגרף בשל אי-השוויון הבא (שלא נוכיח כאן):

$$\phi^2(G) = \min_S \left(\frac{|E(S, S')|}{|S|}\right)^2 \le \quad d - \lambda \quad \le 2 \min_S \left(\frac{|E(S, S')|}{|S|}\right) = 2\phi(G)$$

כלומר, יש לנו חסם עליון ותחתון שתלויים בגודלו של λ, וקושרים את ההגדרה הקודמת שלנו לגרף מרחיב, שכללה את מדידת הרחבת הקשתות המינימלית, למרווח הספקטרלי של המטריצה המייצגת: $d - \lambda$.

הדיון הארוך שלנו מוביל אותנו לטענה הבאה:

טענה:

> יהי $G = (V, E)$ גרף $(n, d, \lambda) - expander$, ותהי $F \subset E$ תת קבוצה של קשתות. עבור מסלול אקראי באורך t, שהצעד הראשון שלו הוא קשת אקראית ב- F, ההסתברות שהצעד האחרון גם כן נמצא ב- F (הקשתות הראשונה והאחרונה ב- F), היא לכל היותר:
>
> $$\frac{|F|}{|E|} + \left(\frac{\lambda}{d}\right)^{t-1}$$
>
> המשמעות של הטענה היא כזו – ככל שגרף מרחיב יותר, הערך $\left(\frac{\lambda}{d}\right)$ קטן יותר, והתפלגות צמתי היעד לאורך מסלול באורך כלשהו מתקרבת מאוד להתפלגות אחידה ממש על הצמתים ככל ש – t (אורך המסלול) גדול יותר.

הוכחה:

ההוכחה כוללת המון אלגברה, וממש אפשר לדלג עליה אם רוצים. מה שחשוב לנו זה שהוכחה תקפה גם עבור גרפים עם ריבוי קשתות (כמה קשתות בין אותו זוג צמתים). אם בחרתם לדלג, אחרי ההוכחה יש הערה קטנה על בניית גרפים רגולריים שצריך לראות.

תהי G המטריצה המייצגת של G. אנחנו נגדיר את $\mathbb{G}$ בתור המטריצה המייצגת בגרסה ההסתברותית, כלומר $\mathbb{G} = \frac{G}{d}$. המסלול שלנו מתחיל בקשת אקראית ב- F, ולכן צומת המקור שלנו מתואר על ידי התפלגות – יש לנו $2|F|$ קצוות של קשתות ב- F, וככל שיותר מהן יושבות על אותו הצומת, ההסתברות לבחור בו בתור צומת המקור גדלה.

s יהיה וקטור ההתפלגות עבור הצומת הראשונה במסלול, כאשר x_v הוא הוקטור המייצג של צומת v ו- $F(v)$ הוא הסימון עבור מספר הקשתות של v שהן איברים ב- F:

$$s = \sum_{v \in V} x_v \frac{F(v)}{2|F|}$$

עכשיו שיש לנו את ההתפלגות עבור הצומת הראשון, ההתפלגות עבור הצומת אליו נגיע לאחר $t-1$ צעדים היא $\mathbb{G}^{t-1}s$. ההסתברות P לכך שהקשת האחרונה שנעבור בה תהיה ב- F שווה להסתברות שהגענו לצומת כלשהו u אחרי $t-1$ צעדים, כפול מספר קשתות F היוצאות מ- u חלקי d (שיהוו הצעד האחרון – צעד מספר t). כלומר – המכפלה הפנימית בין ההתפלגות על הצומת אליה הגענו לפני הצעד האחרון, כפול הוקטור המייצג עבור כל צומת מה ההסתברות שקשת סמוכה לו נמצאת ב- F:

$$P = \langle \mathbb{G}^{t-1}s, \sum_{u \in V} u \frac{F(u)}{d} \rangle$$

ובכתיבה אחרת, אנחנו מוסיפים פקטור של $\left(\frac{|2F|}{d}\right)$ כדי להפוך את וקטור ההתפלגות על הקשת האחרונה לוקטור ההתפלגות על הצומת הראשונה:

$$P = \langle \mathbb{G}^{t-1}s, \sum_{u \in V} u \frac{F(u)}{2|F|} \rangle \left(\frac{2|F|}{d}\right) = \frac{2|F|}{d} \langle \mathbb{G}^{t-1}s, s \rangle$$

את הוקטור s ניתן לכתוב בתור הסכום $s = s_U + s^{\perp}$, כאשר s_U הוא וקטור ההתפלגות האחידה $(\frac{1}{n}, \cdots, \frac{1}{n})$, והוקטור $s^{\perp}$ מאונך אליו (כלומר $\langle s_U, s^{\perp} \rangle = 0$). זאת כיוון ש- s הוא וקטור התפלגות, כלומר סכום האיברים בו שווה ל-1. זה גורם לכך שסכום האיברים החיוביים ב- $s^{\perp}$ שווה לסכום השליליים, וכיוון ש- s_U ממשקל כל איבר בצורה שווה, כפל איבר-איבר מתאפס.

מכאן:

לינאריות באיבר הראשון:

$$\langle \mathbb{G}^{t-1}s, s \rangle = \langle \mathbb{G}^{t-1}s_U, s \rangle + \langle \mathbb{G}^{t-1}s^{\perp}, s \rangle$$

d הוא ערך-עצמי של G, ולכן 1 הוא ערך עצמי של $\mathbb{G}$ עבור הוקטור $\vec{1}$, ש- s_U פרופורציונלי אליו:

$$= \langle s_U, s \rangle + \langle \mathbb{G}^{t-1}s^{\perp}, s \rangle$$

$$= \langle s_U, s_U + s^{\perp} \rangle + \langle \mathbb{G}^{t-1}s^{\perp}, s \rangle$$

שוב לינאריות, בתוספת העובדה ש- $\langle s_U, s^{\perp} \rangle = 0$:

$$= \langle s_U, s_U \rangle + \langle s_U, s^{\perp} \rangle + \langle \mathbb{G}^{t-1}s^{\perp}, s \rangle$$

$$= \|s_U\|^2 + \langle \mathbb{G}^{t-1}s^{\perp}, s \rangle$$

$$= \frac{1}{n} + \langle \mathbb{G}^{t-1}s^{\perp}, s \rangle$$

לפי אי-שוויון קושי-שוורץ:

$$\leq \frac{1}{n} + \|\mathbb{G}^{t-1}s^{\perp}\| \cdot \|s\|$$

המעבר הבא נובע מנוסחת ריילי (**Rayleigh quotient**), שלא דיברנו עליה, ופחות או יותר קובעת ש- λ, הערך העצמי השני של G (בערך מוחלט), היא המקסימום מבין הערכים האלה: $\frac{|\langle Gx,x\rangle|}{\langle x,x\rangle}$, עבור וקטורים x מאונכים ל- $\vec{1}$ (או ל- s_U במקרה שלנו). אותה הטענה מתקיימת גם עבור $\mathbb{G}^{t-1}$, שבה λ הופך ל- $\left(\frac{\lambda}{d}\right)^{i-1}$. **אנחנו לא נתמקד בה, ונקבל בתור טענה נכונה את אי השוויון** $\|\mathbb{G}^{t-1}s^{\perp}\| \leq \left(\frac{\lambda}{d}\right)^{i-1} \|s^{\perp}\|$, ואנחנו ממשיכים:

$$\leq \frac{1}{n} + \left(\frac{\lambda}{d}\right)^{i-1} \|s^{\perp}\| \cdot \|s\|$$

$s^{\perp}$ הוא החיסור של הוקטור s_U מ- s, שניהם וקטורי התפלגות עם סכום 1 – והנורמה של s_U היא המינימלית מבין אלה. $s^{\perp}$ מפוזר במשקלי האיברים בצורה אחידה יותר מ- s, ולכן הנורמה שלו דומה יותר לזו של s_U, וקטנה מ- $\|s\|$ לכן:

$$\leq \frac{1}{n} + \left(\frac{\lambda}{d}\right)^{i-1} \|s\|^2$$

$$= \frac{1}{n} + \left(\frac{\lambda}{d}\right)^{i-1} \left\|\sum_{v\in V} x_v \frac{F(v)}{2|F|}\right\|^2$$

$$= \frac{1}{n} + \left(\frac{\lambda}{d}\right)^{i-1} \sqrt{\langle\sum_{v\in V} x_v \frac{F(v)}{2|F|}, \sum_{v\in V} x_v \frac{F(v)}{2|F|}\rangle}^2$$

$$= \frac{1}{n} + \left(\frac{\lambda}{d}\right)^{i-1} \sum_{v\in V}\left(\frac{F(v)}{2|F|}\right)^2$$

ה- $F(v)$ המקסימלי האפשרי הוא d, ולכן:

$$\leq \frac{1}{n} + \left(\frac{\lambda}{d}\right)^{i-1} \frac{d}{2|F|}\sum_{v\in V}\frac{F(v)}{2|F|}$$

והסכום בביטוי שקיבלנו הוא דרגת F של כל הצמתים בגרף, כלומר $2|F|$, חלקי $2|F|$:

$$\leq \frac{1}{n} + \left(\frac{\lambda}{d}\right)^{i-1} \frac{d}{2|F|}$$

נציב בביטוי שקיבלנו קודם עבור התפלגות הקשת האחרונה במסלול:

$$P = \frac{2|F|}{d}\langle\mathbb{G}^{t-1}s, s\rangle \leq \frac{2|F|}{d}\left(\frac{1}{n} + \left(\frac{\lambda}{d}\right)^{i-1} \frac{d}{2|F|}\right) = \frac{2|F|}{dn}\left(\frac{\lambda}{d}\right)^{i-1}$$

וכיוון ש- $|E| = \frac{dn}{2}$ (תוצאה מיידית של היות G d-רגולרי), הוכחנו את הטענה.

עבדנו קשה, הוכחנו את אי השוויון – למה זה טוב? בהמשך נשתמש בתכונה הזו, שבגרף מרחיב מסלול רנדומלי דומה להתפלגות אחידה, כדי "לערבב" *CSP* בעזרת מסלולים בגרפים. אולי יותר נכון לומר שנערבב אותו ואז נוכיח שערבבנו כמו שצריך... אבל אני קופץ קדימה.

יש עוד "תכונה" של גרפים מרחיבים שחשובה לנו – והיא שנורא קל לבנות אותם. יש לנו אלגוריתמים (בניות) להרכבת גרפים רגולריים ומרחיבים בגודל לא חסום, ובדרך כלל אם נצטרך גרף מרחיב כלשהו – הוא קיים. אנחנו לא נתעסק בבניות כאלה, וכשאשתמש בהן בהמשך כולנו נהנה ונעמיד פנים שאנחנו מבינים על מה מדובר -

טענה:

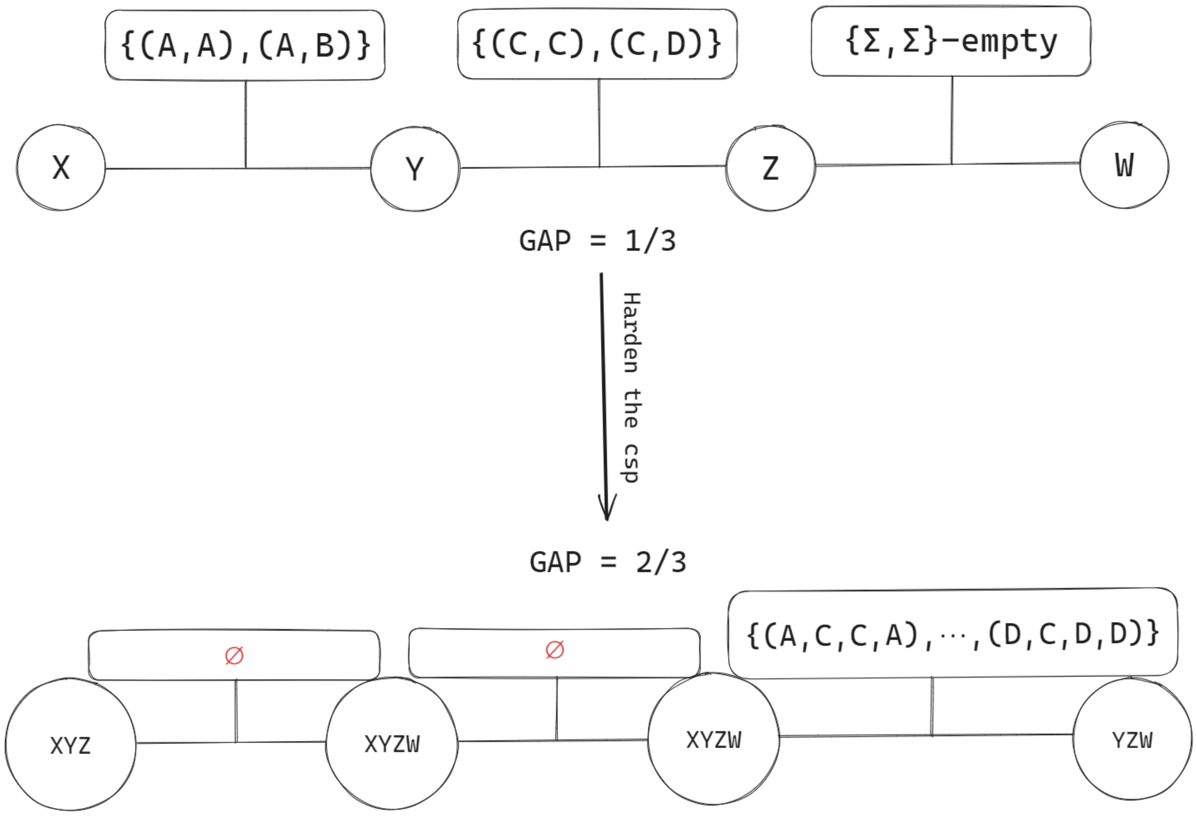
קיים $d \in \mathbb{N}$ ו- $d > \lambda > 0$ כך שלכל n, ישנה בנייה פולינומיאלית של גרף מרחיב d-רגולרי עם ערך עצמי שני קטן או שווה בערכו המוחלט ל- λ על n צמתים. הגרף המרחיב יכול להכיל לולאות עצמיות (מצומת לעצמו), וקשתות מקבילות (יותר מקשת אחת בין זוג צמתים).

על ההוכחה (אלגוריתם הבנייה עצמו) – נדלג.

מבחני השמה – Assignment Testers

דיברנו הרבה על גרפים מרחיבים, וסביר ששכחנו בינתיים מה התפקיד של מבחני השמה בכל המרק הזה, אז הנה תזכורת:

אנחנו לוקחים בעיית CSP, וגורמים לכל אילוץ (כל קשת) בין 2 צמתים להכיל בתוכו גם אילוצים של קשתות שכנות.



שני הגרפים שאנחנו רואים הם בעיות אילוצים בינאריות על ארבעה משתנים. $\{\Sigma,\Sigma\}$ הוא האילוץ הריק, שמסופק בכל השמה. הגרף העליון מעל המשתנים X,Y,Z,W בעל GAP של שליש, כיוון שהמשתנה Y יכול לקבל ערך שמספק אחד בלבד מהאילוצים המופעלים עליו (השמאלי על ידי A,B או הימני על ידי C). הפעולה $Harden\ the\ csp$ היא רדוקציה אל בעיית CSP אחרת שבה כל משתנה בפני עצמו מקבל סדרת ערכים עבור וקטור של כמה משתנים, ואל כל אילוץ מתווספת הדרישה לספק את האילוצים שלידו. כיוון שהאילוצים על (X,Y) ו- (Y,Z) גרמו לסתירה, שני האילוצים החדשים הכוללים אותם לא מסופקים באף השמה. האילוץ הימני ביותר בבעיה החדשה אדיש לאילוץ (X,Y) שהיה רחוק מדי מ- (Z,W) בבעיה המקורית, ולכן עדיין ספיק.

אם היה קיים פתרון עבור הבעיה הראשונה, אותה השמה למשתנים תפתור גם את הבעיה החדשה – לא הוספנו תנאים שלא היו קיימים לפני, רק סידרנו את אלה שהיו לנו מחדש. אם לא היה פתרון לבעיה ההיא, הפתרון האופטימלי עבורה עכשיו קצת פחות אופטימלי. זה מעולה עבור המטרה המקורית שלנו, להפוך את ה- *CSP* לבעל *GAP* גדול, אבל אם נפריז בשימוש בכלי המסוים הזה הייצוג הכולל של הבעיה יגדל מחוץ לשליטה. בכל פעם שהטריק שלנו מופעל, השמה למשתנה הופכת להשמה עבור סדרת משתנים, וכל אילוץ, במקום לכפות זוגות של השמות למשתנים, צריך לכפות סדרה של כאלה. בדוגמה למעלה, אילוץ על הזוג (Z,W) הפך לאילוץ על הסדרה (X,Y,Z,W). אם נפעיל שוב את $Harden\ the\ CSP$ נקבל אילוץ על רביעייה של רביעיות משתנים, וכן הלאה. באופן כללי, אם אילוץ גדל פי K בכל פעם שמבצעים את הרדוקציה, לאחר $\log(m)$ איטרציות נקבל אילוצים בסדר גודל $O(1)^{K^{\log(m)}}=\Omega(2^{\mathrm{m}})$, לא טוב לנו...

אז מה נשאר לנו לעשות? היה לנו גרף אילוצים, שהפכנו כל אילוץ בו להיפר-אילוץ על כמה משתנים. מה אם הייתה לנו דרך לקחת כל אחד מהיפר-האילוצים האלה ולהפוך אותו לבעיית *CSP* שקולה, על מספר אילוצים קטנים יותר? נוכל לחבר את כל הגרפים האלה לגרף יחיד ואולי כך להגדיל את ה- *GAP* שלנו, בלי גידול אקספוננציאלי של האילוצים - האם יש לנו תהליך כזה?

יש, ובגלל זה אנחנו כאן. אבל לפני כן, אנחנו צריכים לשים לב לשני דברים:

א. המטרה שלנו היא להגדיל את ה- *GAP*, ולכן פירוק של אילוץ לבעיית אילוצים מקבילה לא יכול להתבצע סתם כך, אנחנו צריכים לדאוג שה- *GAP* של הבעיה לא קטן מדי – אחרת כל מה שנשיג מהקשיה של האילוץ, ואז פירוק שלו, זו צריכה של משאבי חישוב.

ב. אנחנו צריכים לדאוג שאת תתי הבעיות המתקבלות מכל אילוץ, אנחנו יכולים לחבר מחדש לגרף אילוצים כולל, שמכיל את אותן התלויות שהיו קיימות בין האילוצים המקוריים. אי הספיקות של בעיית האילוצים שלנו היא תוצאה של התלויות האלה, והפרדה של האילוצים לבעיות נפרדות תמחק אותן לחלוטין.

ועכשיו, **מבחן השמה – *Assignment Tester***:

מבחן השמה של q שאילתות ושמירת מרחק γ בעל א"ב Σ הוא אלגוריתם רדוקציה שפועל כך:

הקלט:

נוסחה לוגית מעל קבוצת המשתנים הבינאריים X. האופרטורים הלוגיים המותרים בנוסחה הם $\{\wedge, \vee, \neg\}$.

הפלט:

קבוצת אילוצים C מעל המשתנים $X \cup Y$, כאשר X הם המשתנים מנוסחת הקלט ו- Y הם משתנים שהאלגוריתם הוסיף, ושמקבלים ערכים ב- Σ. בעיית האילוצים מקיימת את התנאים הבאים:

א. כל אילוץ תלוי ב- q משתנים לכל היותר. q הוא פרמטר של מבחן ההשמה ואינו תלוי בקלט.

ב. עבור כל השמה $a: X \to \{0,1\}$ למשתנים הבינאריים X:

1. אם ההשמה מספקת את הנוסחה הלוגית שהתקבלה בקלט, אנחנו יכולים להשלים אותה עם השמה $b: Y \to \Sigma$ כך שבעיית האילוצים שלנו מסופקת במלואה.
2. a היא השמה למשתנים בינאריים, ולכן אפשר להתייחס אליה בתור וקטור/מחרוזת מעל $\{0,1\}$ ולהשתמש במדד המרחק עבור קודים מתקני שגיאות.
תהי s ההשמה המספקת עבור נוסחת הקלט, שקרובה ביותר ל- a. אם המרחק היחסי של a, s הוא $\Delta(a,s) = \delta$, אז בכל השמה $b: Y \to \Sigma$, לפחות $\gamma\delta|C|$ מהאילוצים שהאלגוריתם מחזיר מופרים. אם *אין* השמה תקינה עבור נוסחת הקלט, אנחנו מתייחסים למרחק היחסי של a מהשמה תקינה בתור 1, כלומר לפחות $\gamma|C|$ מהאילוצים בבעיה שחוזרת מופרים, עבור כל השמה.
זאת אומרת שספיקות בעיית האילוצים שמבחן ההשמה מחזיר "שומרת על מרחק לינארי ב- γ" מסיפוק הנוסחה הלוגית על ידי ההשמה ל- X.

איך נשתמש במבחן כדי להשיג את המטרות שלנו?

נניח שעברנו את שלב היפר-האילוצים, והגדלנו את ה- *GAP* שלנו פי K. מעכשיו היפר-אילוץ הוא פשוט אילוץ.

צעד א:

אנחנו לוקחים את קבוצת הערכים שמשתנים בגרף יכולים לקבל (בשם אחר – א"ב האילוצים), ומקודדים אותה בקוד מתקן השגיאות $Code$ שיש לו מרחק יחסי של ¼ [14], וכל מילת קוד בו באורך m סיביות.

צעד ב:

כל משתנה הופך לסדרה של m משתנים בינאריים, שמקבלים ערכים ב- $\{0,1\}$. סדרת המשתנים הזו אמורה ליצור מילים ב- $Code$, כלומר סימבולים תקינים להשמה של משתנה. את האילוצים אנחנו הופכים לאילוצים על $2m$ משתנים – כל אילוץ על זוג משתנים, עכשיו מכריח את הסדרות המתאימות למשתנים אלה לקבל רק מילות קוד תקינות, ורק את אותן מילות קוד שמהוות השמה תקינה לאילוץ. בעיית האילוצים כבר לא בינארית, אבל זה לא מפריע לנו.

צעד ג:

[14] לא נבנה אותו, אבל יש כזה.

את האילוצים על $2m$ משתנים אנחנו הופכים לנוסחה לוגית עם האופרטורים $\{\neg, \vee, \wedge\}$ בלבד – קל לבנות נוסחת OR על קבוצה של נוסחאות AND שתתאים לאילוץ (נוסחת $2mDNF$), והיא תענה על התנאים שלנו. את הנוסחה שקיבלנו, אפשר להכניס למבחן השמה.

צעד ד:

מכניסים כל אחת מהנוסחאות המתקבלות למבחן ההשמה. הפלט שלנו הוא קבוצה של בעיות *CSP* שלא תלויות אחת בשנייה, ולא כולן בינאריות בהכרח, אבל בכולן

- $2m$ המשתנים מהנוסחה עדיין קיימים.
- גודל התחום לכל משתנה הוא לכל היותר $|\Sigma|$, פרמטר הא"ב של מבחן ההשמה.
- כל אילוץ תלוי לכל היותר ב- q משתנים, פרמטר השאילתות של מבחן ההשמה.

צעד ה:

הופכים את בעיות ה- *CSP* לבינאריות, באותה הדרך שהשתמשנו בה אי אז בהקדמה בחלק א, כשהצגנו את מושג ה-*CSP*. $2m$ המשתנים עדיין קיימים, וגודל התחום המקסימלי הוא עכשיו $2^{|\Sigma|}$. עכשיו הבעיות ניתנות להצגה בתור גרף. מעכשיו, אנחנו מעדיפים להתייחס לצעדים ד ו- ה בתור יחידה אחת שמהווה מבחן השמה של 2 שאילתות (במקום q).

צעד ו:

מחברים את הגרפים, לפי $2m$ המשתנים, לגרף יחיד. זאת אומרת, $2m$ המשתנים הבינאריים היו ונשארו מילות קוד עבור המשתנים המקוריים שהיו לנו לפני השימוש במבחן ההשמה. כל m צמתים, בכל אחד מהגרפים, שמקודדים את אותו המשתנה X, יאוחדו לסדרה יחידה של m משתנים.

לתהליך שתיארנו עכשיו יש כמה תכונות שאנחנו רוצים להוכיח.

למת הקומפוזיציה – *Composition lemma*

יהי AT מבחן השמה של 2 שאילתות, שמירת מרחק γ וא"ב Σ. יהי $\beta > 0$.

כל גרף אילוצים $G = (V, E, C)$ שבו האילוצים הם מעל Σ, יכול להפוך בחישוב פולינומיאלי לגרף אילוצים חדש $G' = (V', E', C')$ עם אילוצים מעל Σ_0, ששומר על התנאים הבאים:

1. $size(G') \le O(1) size(G)$, כלומר אפשר לחסום את הגדילה הכוללת של ייצוג הבעיה על ידי קבוע.
2. אם G היה ספיק, גם G' ספיק.
3. $Gap(G') \ge \beta \cdot Gap(G)$, כלומר היחס בין המספר המינימלי של אילוצים מופרים למספר האילוצים הכולל, יקטן בסדר גודל קבוע לכל היותר במעבר מ- G ל- G'.

הוכחה:

אנחנו מבצעים את צעדים א-ו שתיארנו על G.

כל אחד מהשלבים שמגדילים את הגרף (ב,ד,ה) עובדים על אילוצים יחידים. הגודל המקסימלי של אילוץ הוא $O(|\Sigma|^2)$, כלומר קבוע בעצמו. נניח ששילוב כל הצעדים האלה על אילוץ גורם לתפיחה של R בגודל הייצוג שלו, אז קיבלנו קבוע שחוסם את גדילת הגרף הכולל: $size(G') \le R \cdot size(G)$

קיבלנו שתנאי 1 של המשפט מתקיים.

נניח שקיימת השמה מספקת עבור המשתנים ב- G. את אותה ההשמה אפשר לתרגם למילות קוד, של הקוד שהשתמשנו בו בשלב א ($Code$).

ההשמה הזו תספק את האילוצים המקודדים של צעד ב. מכאן שגם הנוסחה הלוגית בצעד ג תהיה ספיקה. מבחן ההשמה מבטיח לנו שאם הנוסחה על המשתנים X ספיקה (כאן X הם $2m$ המשתנים של מילות הקוד), אפשר להשלים השמה מספקת עבורם להשמה מספקת עבור כל בעיית האילוצים שהוא מוציא כפלט – ולכן גם אחרי שלבים ד ו-ה הבעיה שנוצרת עדיין ספיקה. בשלב ו אנחנו צריכים לאחד את כל הסדרות באורך m שמייצגות

משתנה ספציפי, אבל כיוון שיש לנו מילת קוד יחידה שמספקת - עם ההשלמות המתאימות לקבוצות Y השונות - את כל הפלטים שמבחן ההשמה יצר מאילוצים על אותו המשתנה, הגרף עדיין ספיק.

אז גם תנאי 2 מתקיים.

תנאי 3 טיפה פחות טריוויאלי, ויצריך קצת תחכום. בשביל להוכיח אותו אנחנו רוצים להתבונן ב- 3 ההשמות הבאות:

ההשמה OPT היא ההשמה האופטימלית עבור G'. היא מפרה את מספר האילוצים הקטן ביותר בגרף שלאחר רדוקציית הקומפוזיציה.

מההשמה OPT אנחנו מחלצים את ההשמה σ עבור הגרף G. היא מוגדרת כך: בגרף G' יש m משתנים בינאריים שמתארים את המשתנה v מהגרף G. יש גם מילת קוד לכל ערך ש- v יכול לקבל. OPT לא מחויבת לתת ל- m המשתנים האלה מילת קוד תקינה, אבל יש איזושהי מילת קוד עם מרחק מינימלי מהערכים ש- OPT נתנה לסדרה – והערך של מילת הקוד הזו הוא הערך של σ עבור v. אנחנו רוצים להגדיר סימון נוח יותר להגדרה שתיארנו כרגע, ולכן:

בגרף G' ישנה סדרת משתנים שמתאימה ל- v ב- G. לסדרה הזו נקרא $[v]$. עכשיו אנחנו יכולים לתאר את הפסקה למעלה כך:

$$\sigma(v) = \min_{w \in Code(\Sigma)} \big(\delta(OPT[v], w)\big)$$

כיוון ש- σ היא השמה עבור G, היא מפרה לפחות $Gap(G)$ מהאילוצים ב- G. אנחנו נבחר אחד מהאילוצים האלה ונקרא לצמתים שהוא פועל עליהם u ו- v. ההשמה האחרונה שמעניינת אותנו היא ההשמה UV, שהיא השמה תקינה אך ורק עבור הגרף המתקבל מקומפוזיציה של האילוץ היחיד $c(u,v)$. זאת אומרת שהיא נותנת ל- $\big[[u],[v]\big]$ ערכים שהם מילות קוד, ושאפשר להשלים אותם עם $b: Y(u,v) \to \Sigma$ כדי לספק את כל בעיית האילוצים המתקבלת עבור הצמתים האלה[15].

הטענה שלנו היא:

$$\Delta(\ [UV([u]), UV([v])]\ ,\ \ [OPT([u]), OPT([v])]\ \ ) \geq \frac{1}{16}$$

ומה שאנחנו מתכוונים באמת לומר זה:

"עבור כל השמה OPT שנבחר עבור G' נוכל למצוא $Gap(G)$ פלטים של מבחן ההשמה, שאפשר לחסום את המרחק שלהם מסיפוק מלא על ידי $\frac{1}{16} \cdot \gamma$, המרחק של OPT מזוג סימבולים שמספק את האילוץ כפול פרמטר שמירת המרחק של מבחן ההשמה"

16 הוא לא מספר קסם, ומייד יהיה ברור מה הוא עושה בהוכחה שלנו.

קודם כל, אנחנו יודעים ש- σ לא מספקת את האילוץ על u, v – ולכן היא שונה מ- $[UV([u]), UV([v])]$. בפרט, $Code\big(\sigma(u)\big) \neq UV([u])$ או $Code\big(\sigma(v)\big) \neq UV([v])$. בלי הגבלת הכלליות, נניח שאי-שוויון מתקיים עבור $[u]$.

לקוד שלנו יש מרחק של ¼ , ולכן:

$$\frac{1}{4} \leq \Delta\left(Code\big(\sigma(v)\big), UV([v])\right)$$

עכשיו אנחנו צריכים רגע כדי להשתכנע שמרחק בקודים מתקני שגיאות שומר על אי שוויון המשולש.

[15] לא דאגנו להבטיח שיש צמתים u, v עבורם תת הגרף המתקבל ספיק, אבל במקרה שלנו זה לא משנה – אנחנו מראים שאם UV קיימת, OPT רחוקה ממנה. אם היא לא קיימת, מבחן ההשמה מסיק שהמרחק מקסימלי.

הרגע עבר, ואנחנו מקבלים:

$$\Delta\left(Code(\sigma(v)), UV([v])\right) \leq \Delta\left(Code(\sigma(v)), OPT([v])\right) + \Delta\big(OPT([v]), UV([v])\big)$$

בנוסף, הגדרנו את σ עם מרחק מינימלי מ- OPT, ולכן:

$$\Delta\left(Code(\sigma(v)), OPT([v])\right) \leq \Delta\big(OPT([v]), UV([v])\big)$$

כלומר:

$$\Delta\left(Code(\sigma(v)), OPT([v])\right) + \Delta\big(OPT([v]), UV([v])\big) \leq 2 \cdot \Delta\big(OPT([v]), UV([v])\big)$$

וכיוון ש- $\big[[u],[v]\big]$ ארוך פי שתיים בדיוק מ- $[u]$, המרחק היחסי שלו קטן בחצי לכל היותר:

$$\Delta\big(OPT([v]), UV([v])\big) \leq 2 \cdot \Delta\big(\ [UV([u]), UV([v])]\ ,\ \ [OPT([u]), OPT([v])]\ \big)$$

בסך הכל:

$$\begin{aligned}
&\frac{1}{4} \leq \Delta\left(Code(\sigma(v)), UV([v])\right) \\
&\leq \Delta\left(Code(\sigma(v)), OPT([v])\right) + \Delta\big(OPT([v]), UV([v])\big) \\
&\leq 2 \cdot \Delta\big(OPT([v]), UV([v])\big) \\
&\Longrightarrow \frac{1}{8} \leq \Delta\big(OPT([v]), UV([v])\big) \\
&\Longrightarrow \frac{1}{16} \leq \Delta\big(\ [UV([u]), UV([v])]\ ,\ \ [OPT([u]), OPT([v])]\ \big)
\end{aligned}$$ [16]

יש לנו לפחות $Gap(G)$ אילוצים $c(u,v)$ ש- σ מפרה, ומבחן ההשמה מבטיח לשבור $\frac{\gamma}{16}$ מהאילוצים בפלט המתאים לכל $c(u,v)$ כזה. אנחנו מקבלים ש- $Gap(G') \geq \frac{\gamma}{16} Gap(G)$, והצלחנו להוכיח גם את תנאי 3, ולהשלים את למת הקומפוזיציה.

[16] המספר 1/16 הוא פשוט המרחק של הקוד, כפול זה שאנחנו מודדים רק חצי מהזוג $[u],[v]$, כפול זה שאנחנו מתייחסים למקרה הגרוע שבו $OPT[v]$ היא בדיוק באמצע בין $Code(\sigma(v)), UV([v])$.

בנייה (בערך) של מבחן השמה

הקיום של אובייקט מתמטי כמו מבחן ההשמה לא טריוויאלי להוכחה, ולכן לא נבנה אחד כזה באופן מדויק – רק ננסה להשתכנע שהיינו יכולים לבנות אותו לו רצינו. בבנייה יש תוצאות שלא נוכיח, אבל התהליך שלנו ינסה להיות הגיוני וברור ככל האפשר.

ומה הוא בעצם "התהליך שלנו"? מבחן השמה הוא איזושהי רדוקציה מנוסחה לוגית לבעיית CSP – מה היא כוללת?

קודם כל, על נוסחאות לוגיות לא כיף לדבר. כדי שלא נירדם בדרך, אנחנו נתייחס לנוסחה שלנו בתור ⚡*מעגל לוגי*⚡. מבחינה מתמטית אין הבדל – הנוסחה שלנו כללה רק את האופרטורים $\{\neg, \vee, \wedge\}$, שמקבילים בדיוק לשלשה הקלאסית והטיורינג-שלמה $\{AND, OR, NOT\}$, אז המעבר היחיד שאנחנו עושים הוא תפיסתי – במקום אופרטורים ונוסחאות, יש לנו שערים וקלטים-פלטים.

אנחנו עוברים לייצוג על ידי שערים מסיבות אסתטיות לחלוטין – לשלב הראשון בתהליך שנבצע נהוג לקרוא **אריתמטיזציה של מעגל בוליאני – Boolean Circuit Arithmetization**, ואנחנו רוצים להשתמש בטרמינולוגיה המתאימה. באריתמטיזציה של מעגל, המטרה שלנו היא ליצור מערכת משוואות מעלה השדה Z_2 שמתארת את המעגל שלנו בצורה אריתמטית, במקום לוגית.

למעגל יש משתני כניסה, שערים עם קלט-פלט, וערך יציאה בוליאניים. אנחנו ניקח את משתני הכניסה, ואת הפלט של כל שער במעגל, ונגדיר עבורם את קבוצת המשתנים $\{z_i : 1 \leq i \leq k, z_i \in Z_2\}$, כלומר k משתנים בינאריים, אחד לכל ערך שמופיע במעגל שלנו. כל יציאה של כל שער תתואר בעזרת אחת מהנוסחאות הבאות:

שער AND שמקבל בתור קלט את הערכים z_1, z_2 ומוציא את הערך z_3 שקול לנוסחה הבאה:

$$z_1 z_2 - z_3 = 0$$

שער OR על המשתנים z_1, z_2 ופלט z_3 שקול ל-:

$$z_1 + z_2 + z_1 z_2 - z_3 = 0$$

שער NOT על z_1 שמוציא את z_3 יהפוך למשוואה:

$$z_1 + z_3 - 1 = 0$$

כיוון שאנחנו רוצים שהנוסחה תסופק, נגדיר עוד משוואה עבור שער היציאה של הנוסחה:

$$z_{out} - 1 = 0$$

את המעגל הלוגי אפשר לייצג בעזרת מערכת משוואות שקולה מעל קבוצת המשתנים Z. אם השער הלוגי ספיק, קיים וקטור z במרחב Z_2^k שפותר את מערכת המשוואות. אנחנו רוצים לבנות מבחן השמה שלוקח את מערכת המשוואות הזו, ומרכיב עליה בעיית CSP עם משתנים חיצוניים Y שנשברת ביחס לינארי γ למרחק של הוקטור $x \in Z_2^k$ מהוקטור הקרוב אליו ביותר שפותר את המשוואות.[17] איך? קצת קידוד, קצת אלגברה, וקצת מזל.

אז קודם כל, מערכת המשוואות שלנו צריכה שם, ואיזה שם מתאים יותר מ- P? קבוצת המשוואות שהתקבלה תהיה:

---

[17] יש כאן איזשהו קאץ' – בהגדרת מבחן ההשמה הבטחנו שהיחס הלינארי של שבירת האילוצים תלוי רק בקבוצה X, שהיא קבוצת משתני הכניסה למעגל. איך פתאום אנחנו משנים את הפרמטרים כך שהיחס תלוי בכל הוקטור z, שקבוצת הערכים בו גדולה יותר? זה עלול לגרום למבחן ההשמה שנבנה להיות חלש יותר מההגדרה שהשתמשנו בה! ולכן חשוב לשים לב שיש גבול למספר השערים שאפשר להכניס למעגל: שני שערי NOT עוקבים הם כמובן חסרי משמעות, ושני השערים האחרים לוקחים שני קלטים והופכים אותם לפלט יחיד. אנחנו יכולים להראות שגם אם נדחוף את כל השערים שאפשר לנוסחה שלנו, מספר המשתנים במערכת המשוואות יהיה לכל היותר גדול פי 4 ממספר משתני הכניסה.

$$P = \{equation\ P_i : 1 \le i \le number\ of\ equations\}$$

ועכשיו אנחנו יכולים לדבר על צירופים לינאריים של איברים ב- P. נניח שיש לנו השמה תקינה a כלשהי עבור המשתנים $\{z_1, \cdots, z_k\}$ – כיוון שפתרנו את המערכת, כל המשוואות מתאפסות. אנחנו יכולים לקחת את צד שמאל של כולן ולסכום אותו, וגם הסכום מתאפס (בתור סכום של אפסים):

$$\sum_{i=1}^{|P|} P_i = 0$$

אנחנו עומדים לסחוט מהנוסחה הזו אילוצים כמו מים, בעזרת כל תת הסכומים שאפשר להרכיב ממנה - צירוף לינארי של המשוואות על פי וקטור $v \in \{0,1\}^{|P|}$ נראה כך:

$$\sum_{i=1}^{|P|} v_i \cdot P_i = \ v_{|P|}(z_{out} - 1) + \sum_{i=1}^{|P|-1} v_i \cdot \begin{cases} z_j z_k - z_h & : P_i\ \ is\ AND \\ z_j + z_k + z_j z_k - z_h & : P_i\ \ is\ OR \\ z_j + z_h - 1 & : P_i\ \ is\ NOT \end{cases}$$

כלומר אנחנו בוחרים מחרוזת v שתקבע אילו מהמשוואות תשארנה בצירוף, ואילו יוכפלו ב- 0 ויתבטלו.

אם a פתרה את מערכת המשוואות, היא תאפס גם כל צירוף לינארי כזה – אלה עדיין יהיו סכומים של אפסים. הדבר הראשון שאנחנו רוצים להראות, זה שאם a *לא* פותרת את המערכת, בדיוק חצי מהצירופים האלה שווים ל- 1. כשדיברנו על קודים מתקני שגיאות, התעסקנו במיוחד בקוד האדאמארד – אם הוא לא יושב כמו שצריך, עכשיו זה הזמן להתרענן.

a היא פתרון שגוי עבור המערכת, ולכן יש לנו תת קבוצה כלשהי של משוואות שערכן הוא 1. אם נתייחס לערכי המשוואות בתור וקטור x כלשהו מעל $Z_2^{|P|}$, אנחנו שמים לב שהצירופים הלינאריים האלה מקודדים וקטור בקוד האדאמארד – אלה המכפלות הסקלריות של x עם הוקטורים v של הצירופים! אם a הייתה פותרת את המשוואות, היינו מקבלים $x = \vec{0}$, וכמובן $Had(\vec{0}) = longer\ \vec{0}$. אבל a לא מאפסת את כל המשוואות, ולקוד האדאמארד יש מרחק של ½ , כלומר חצי מהצירופים הלינאריים שווים ל- 1.

כדי להוציא מקסימום תנאים ממערכת המשוואות, אנחנו רוצים לרשום אותה מחדש. כל מחובר בכל משוואה במערכת הוא מרכיב סקלרי, לינארי או ריבועי מעל האיברים של ההשמה a, ולכן כולם מוכלים בקבוצה הבאה:

$$\{1,0\} \cup \{a_i \in a\} \cup \{a_i a_j \in a\}$$

עבור צירוף לינארי כלשהו, נרצה לפתוח את כל מערכת המשוואות, לכנס איברים דומים, ולהגיע לביטוי הבא:

$$\sum_{i=1}^{|P|} v_i \cdot P_i = a_{scalar} + \sum_{i=1}^{k} s_i a_i + \sum_{i=1,j=1}^{k} t_{ij} a_i a_j$$

זו עדיין אותה המערכת, אבל עכשיו יש לנו את כל אחד מאיברי ההשמה, וכל מכפלה שלהם, במחובר נפרד עם מקדם משל עצמו. מה שמעניין אותנו במיוחד, הוא הדמיון של המערכת לזוג מכפלות סקלריות מעל Z_2 – יש לנו איבר לא תלוי a_{scalar} שערכו 0 או 1 (בתלות לפי v), ואז מכפלה של $s \cdot a$, ועוד מכפלה של $t \cdot (a \otimes a)$ (כאן t הוא וקטור מעל $\{0,1\}^{k^2}$, ו- $a \otimes a$ הוא הטנזור של ההשמה עם עצמה, כלומר וקטור אחר מעל $\{0,1\}^{k^2}$).

ה- *CSP* שמבחן ההשמה יבנה, שאנחנו רוצים שיסופק רק אם קיימת השמה a שפותרת את מערכת המשוואות, ושיהיה בעל *GAP* גבוה אחרת, הוא בעצם סדרה של k משתנים בינאריים שצריכה להתמלא בקוד האדאמארד של השמה (וירטואלית, כיוון שאנחנו לא יודעים אם היא קיימת) a מספקת כלשהי, ועוד סדרה של k^2 משתנים שתתמלא בטנזור $a \otimes a$. איך אנחנו עושים את זה? אנחנו צריכים שה- *CSP* יכריח (עם *GAP* גבוה, כלומר הרבה אילוצים צריכים להישבר אם משהו נכנס לא כמו שצריך), את ארבעת התנאים הבאים:

א. לכל צירוף לינארי כזה של המשוואות P:

$$\sum_{i=1}^{|P|} v_i \cdot P_i = a_{scalar} + \sum_{i=1}^{k} s_i a_i + \sum_{i=1,j=1}^{k} t_{ij} a_i a_j$$

הסכום כולו, כלומר $a_{scalar} + s \cdot a + t \cdot a \otimes a$, שווה ל- 0. ב- *CSP* שלנו, זהו אילוץ על שני משתנים, אחד שמייצג את האינדקס s בקוד האדאמארד של a, ואחד שמייצג את אינדקס t בקוד האדאמארד של $a \otimes a$. כמובן, את האינדקסים s, t צריך לחשב לפני בניית הגרף, עבור כל צירוף לינארי אפשרי, וגם את הערכים האפשריים שלהם, שתלויים ב- a_{scalar}.

$Constraint\ set\ A$: $for\ each\ v \in Z_2^{|P|}, and\ the\ resulting\ s \in Z_2^k, t \in Z_2^{k^2}$

$$Had(a)[s] + Had(a \otimes a)[t] + a_{scalar} = 0$$

כפי שכבר ראינו, עבור כל השמה a שאינה מאפסת את המערכת, חצי מהאילוצים האלה מופרים. זה כמובן רק בתנאי שהמשתנים בגרף שמתאימים לאינדקסים s, t הם אכן האינדקסים של קודי האדאמארד המתאימים, ולכן מציבים גם את התנאים הבאים:

ב. ההשמה שניתנת למשתנים בגרף, אכן יוצרת קודי האדאמארד של וקטורים כלשהם ב- Z_2^k ו- $Z_2^{k^2}$.
אפשר להוכיח שהאילוץ שנציג עכשיו נכשל בחלק עבור כל וקטור שאינו קוד האדאמארד של וקטור אחר, ביחס לינארי למרחק שלו ממילת ההאדאמארד הקרובה ביותר – אנחנו נדלג על ההוכחה. האילוץ הוא על שלושה משתנים, והוא בעצם הטענה הבאה:
עבור וקטורים v, u, a מעל $\{0,1\}^k$, מתקיים השוויון:

$$v \cdot a + u \cdot a = \sum_{i=1}^{k} v_i a_i + \sum_{i=1}^{k} u_i a_i = \sum_{i=1}^{k} u_i a_i + v_i a_i = (v+u) \cdot a$$

כלומר, אנחנו נקבע אילוץ, עבור כל שלשת אינדקסים מתאימה $v, u, v+u$, שתכריח את סכום האינדקסים s, u בקוד של a להיות שווה לערך באינדקס המתאים לוקטור $s+u$. אותו המבחן יפעל גם עבור הקוד של הוקטור $a \otimes a$.

$Constraint\ set\ B$: $for\ each\ pair\ u, v\ in\ Z^k\ and\ w, x\ in\ Z^{k^2}$

$$Had(a)[v] + Had(a)[u] = Had(a)[u+v]$$

$$Had(a \otimes a)[w] + Had(a \otimes a)[x] = Had(a \otimes a)[w+x]$$

המבחן מבטיח שהשמנו למשתנים קודי האדאמארד, אבל יש תלות בין הקודים שאנחנו צריכים להבטיח ועוד לא בדקנו – במערכת המשוואות שלנו הוקטורים המקודדים הם a ו- $a \otimes a$, אבל בינתיים, שום דבר לא מונע ממנו למצוא וקטור אחר מעל $\{0,1\}^{k^2}$, במקום $a \otimes a$, כך שקוד ההאדאמארד שלו עונה על מספר רב מהאילוצים שהצבנו, למרות שהתנאים על סכום מערכת המשוואות אינם מתקיימים עבור ההשמה a. מכאן התנאי הבא:

ג. קוד ההאדאמארד של הוקטור מעל $\{0,1\}^{k^2}$ הוא אכן הקוד עבור $a \otimes a$, ולא קוד עבור איזשהו $w \in Z_2^{k^2}$ אחר.

כאן אנחנו בודקים את השוויון הבא (שוב, מעל הוקטורים u, v, a):

$$(v \cdot a)(u \cdot a) = \left(\sum_{i=1}^{k} v_i a_i\right)\left(\sum_{j=1}^{k} u_j a_j\right) = \sum_{i=1}^{k}\sum_{j=1}^{k}(v_i u_j)(a_i a_j) = (v \otimes u)(a \otimes a)$$

כלומר זהו גם בן אילוץ על שלושה משתנים: אנחנו אומרים שהמכפלה של האינדקסים u, v בקידוד של a תהיה שווה לערך באינדקס הטנזור שלהם בקידוד של $a \otimes a$. גם עבור המבחן הזה, אפשר להוכיח שהוא דוחה בחלק קבוע מהפעמים.

$Constraint\ set\ C:\quad for\ each\ pair\ u, v\ in\ Z^k$

$$Had(a)[v] \cdot Had(a)[u] = Had(a \otimes a)[u \otimes v]$$

אם כך, אנחנו בונים גרף אילוצים על משתנים ב- $\{0,1\}$ שבו צריך למלא את קוד ההאדאמארד של השמה תקינה a שפותרת את מערכת המשוואות שקיבלנו, ואם אין השמה תקינה הגרף יהיה בעל *GAP* גבוה. נשמע טוב? לא! יש לנו חור!

מבחנים א ו- ג תלויים בכך שהערכים בגרף התלויות מהווים קוד האדאמארד כלשהו. לא הצלחנו להבטיח שהם עדיין עושים את העבודה כאשר הערכים בגרף הם לא קוד האדאמארד תקין. מבחן ב אמור לוודא שערכים שאינם מילת קוד אינם אפשריים, אבל הוא *תלוי במרחק* ממילת קוד תקינה. יכול להיות שאפשר עדיין להשים לגרף 'כמעט' מילות קוד, שלא יכשילו את מבחן ב יותר מדי, וגם יעברו את שאר המבחנים בהצלחה יחסית.[18]

כאן מצרפים למשחק את תכונת תיקון השגיאות של קוד האדאמארד. במקום לקרוא אינדקס v כלשהו מהקוד, אנחנו נחשב את הערך שלו מתוך שני אינדקסים אחרים:

$$(v + u) \cdot a - u \cdot a = v \cdot a$$

הפרוצדורה הזו יכולה לטעות רק אם $v + u, u$ הם אינדקסים שגויים של מילת הקוד. מה שבעצם יש לנו כאן, זו דרך לקרוא את קוד ההאדאמארד גם אם הוא לא נתון לנו. מבחן ב ייכשל *מאוד* אם המילים שהכנסנו לגרף אינן קרובות למילות האדאמארד. המבחנים האחרים ייכשלו *מאוד*, אם הם קוראים קוד האדאמארד שאינו השמה תקינה. אנחנו נשנה את מבחנים א ו-ג כך שיבדקו את כל התיקונים של v במקום את הוקטור עצמו, כלומר מחשבים את v משני הוקטורים $v + u, u$. זה כמובן גורם לאילוץ במבחן ג לפעול על 6 משתנים, במקום על 3.

$Constraint\ set\ A, redefined:\ for\ each\ v \in Z_2^{|P|}, and\ the\ resulting\ s \in Z_2^k, t \in Z_2^{k^2},$
$and\ each\ u \in Z_2^k, w \in Z_2^{k^2}$

$$Had(a)[s + u] - Had(a)[u] + Had(a \otimes a)[t + w] - Had(a \otimes a)[w] + a_{scalar} = 0$$

$Constraint\ set\ C, redefined:\quad for\ each\ 2\ pairs\ u, v\ and\ x, z\ in\ Z_2^k, and\ each\ w\ in\ Z_2^{k^2}$

$$(Had(a)[v + x] - Had(a)[x]) \cdot (Had(a)[u + z] - Had(a)[z]) = Had(a \otimes a)[u \otimes v + w] - Had(a \otimes a)[w]$$

בשלב האחרון, מבחן ההשמה מבטיח ש- X, קבוצת משתני הכניסה של המעגל הלוגי, קיימים גם בבעיית האילוצים שהוא מייצר, וככל שהם רחוקים יותר מסיפוק המעגל הבעיה פחות ספיקה. עד עכשיו, לא ראינו את המשתנים האלה, ובטח שלא הצבנו עליהם תנאים.

[18] טכנית, יש משהו בטענה ש- "מבחן ג מגן על עצמו". כיוון שהוא פועל על כל קומבינציות האינדקסים האפשריים, גם עם המילה הנתונה אינה מילת האדאמארד, היא קרובה לכזו, ורוב האינדקסים שמבחן ג יקרא יהיו אינדקסים של מילה תקינה. אף על פי כן, יש תלות בין הוקטורים בממד k והוקטורים בממד k^2 שהמבחן בודק, שעלולים לאפשר שילוב של 2 'כמעט מילות קוד' עם תוצאות קצת בעייתיות. מבחן א לא בודק את כל שילובי האינדקסים, ולכן שימוש בתיקון שגיאות הכרחי כדי לוודא שהוא לא קורא דווקא אינדקסים פגומים.

למזלנו, כבר דאגנו לוודא שקוד ההאדאמארד של a הוא באמת קוד האדאמרד, ולכן נשאר רק לבדוק שזהו קוד ההאדאמארד של אותו ה- a שנכנס לצמתים המייצגים את X. כיוון שזהו קוד האדאמרד (או לפחות, הקריאה המתקנת שלנו קרובה מאוד לקוד האדאמארד), האינדקסים עבור וקטורי הבסיס הסטנדרטי צריכים להיות שווים בדיוק ל- a:

$$a \cdot e_j = \sum_{i=1}^{k} \begin{cases} 0 : i \neq j \\ a_j : i = j \end{cases}$$

ולכן התנאי:

ד. גרף האילוצים שלנו בוחן את המשתנים X.

$Constraint\ set\ D: for\ each\ standard\ basis\ vector\ e_j\ and\ u\ in\ Z_2^k$

$$a_j = Had(a)\left[e_j + u\right] - Had(a)[u]$$

ואיתו אנחנו משלימים את הבנייה.

עכשיו, חזרה קצרה על המבחן מנקודת מבט אחרת – אנחנו מכניסים השמה שגויה z למשתנים X, שמרחקה מההשמה המספקת הקרובה ביותר הוא δ, ורואים מה שברנו:

אם הערכים שניתנו למשתנים בגרף הם קודי האדאמארד של $z, z \otimes z$ - אז חצי מהאילוצים של מבחן א נשברים.

אחרת, אם הם קודי האדאמארד של z, ואיזשהו וקטור אחר $u \otimes v$ – אז מספר קבוע מהאילוצים של מבחן ג נשברים.

אחרת, אם הם קודי האדאמארד של סתם וקטורים כלשהם $b, b \otimes b$ ולא של ההשמה z – $\gamma\Delta(b, z)$ מהאילוצים של מבחן ד נשברים.

אחרת, אם הם אינם קודי האדאמארד בכלל, אז ככל שהם רחוקים יותר מקודי האדאמארד, חלק גדול יותר מהאילוצים של מבחן ב נשבר. אם הם קרובים למילות קוד תקינות, תיקון השגיאות שלנו מאפשר בדיקה של מילות הקוד האלה והכשלה של המבחנים האחרים.

כדי לוודא שכל המבחנים מאזנים אחד את השני, יכול להיות שנרצה לשכפל את האילוצים של חלקם (אלה יהיו פשוט קשתות מרובות בגרף האילוצים). לא קשה לחשב את המכפלה הנדרשת עבור כל מבחן, ואם נהיה ממש עצלנים נוכל להביא את מספר האילוצים בכל מבחן למכפלת כל האילוצים של כולם.

אנקדוטה:

בדרך כלל, מבחני השמה מתוארים במודל מוכיח-מאמת, שבו המוכיח צריך לבנות הוכחה עבור הנוסחה הלוגית שמבחן ההשמה מקבל, והמאמת צריך לוודא את נכונות ההוכחה בוודאות קבועה תוך קריאת מספר קבוע של תווים ממנה. במקרה או שלא, מבחן ההשמה עצמו נשמע דומה מאוד לסיבה שבגללה התכנסנו כאן מלכתחילה: אנחנו רוצים להוכיח שמשפחת השפות NP מוכלת ב- PCP$[O(lgn), O(1)]$. הפלט של מבחן ההשמה הוא בסופו של דבר, שפה ב- PCP$[poly(n), O(1)]$. אנחנו בודקים הוכחה באורך אקספוננציאלי לספיקות הנוסחה (קודי ההאדאמרד, שאורכם $\Omega(2^k)$). מספר התווים הרנדומליים שאנחנו צריכים כדי לקרוא חלקים רנדומליים מתוכה כבר לא לוגריתמי, אבל זוהי עדיין *Probabilistically Checkable Proof*.

חלק ד – הוכחת משפט PCP על ידי הגברת תכונת ה- *GAP*

סוף סוף, אחרי עשרות עמודים של עבודה מקדימה, הגיע הזמן להוכיח את המשפט שלנו. אבל היתרון בעשרות עמודים של הכנה, הוא שאנחנו מגיעים מוכנים.

המטרה שלנו היא להשתמש בשקילות שהוכחנו בחלק ב:

$$\boldsymbol{GAP-E3SAT\ \ is\ \ NP\ hard\ \ \Leftrightarrow PCP\ theorem}:\ \ \boldsymbol{NP} \subseteq \boldsymbol{PCP}[\boldsymbol{O}(\boldsymbol{lgn}), \boldsymbol{O}(\boldsymbol{1})]$$

כדי להוכיח את צד שמאל שלה – כלומר, אנחנו צריכים להראות ש- GAP-E3SAT היא NP קשה, ולהסיק את משפט PCP. אבל למי יש כוח להתעסק עם נוסחאות $3CNF$? לא רק שהן משעממות נורא, הן גם לא באמת מספיק כלליות כדי להוות כלי נוח. אנחנו מעדיפים להתעסק בבעיות *CSP* – שמאפשרות לנו לתאר באופן מדויק משפחה גדולה יותר של בעיות, בלי להשתמש ברדוקציות והתאמות אחרות.

אז הדבר הראשון שאנחנו צריכים לעשות הוא להגדיר את הבעיה $\text{GAP-CSP}_{1,s}$ – *לא עשינו את זה עד עכשיו?! אבל השתמשנו בה בלי הפסקה כשדיברנו על מבחני השמה?!*

לא צריך לדאוג, ההגדרה של $\text{GAP-CSP}_{1,s}$ היא בדיוק מה שציפינו, אין כאן הפתעות:

> אלגוריתם שמכריע את השפה $\text{GAP-CSP}_{c,s}$ הוא אלגוריתם שמקבל כקלט בעיית סיפוק אילוצים עם m אילוצים, ומחזיר:
>
> כן – אם יש השמה למשתנים בבעיה, שעבורה לפחות cm אילוצים מסופקים.
>
> לא – אם לכל השמה, פחות מ- sm אילוצים מסופקים.
>
> מה שבא לו – אם יש השמה שמספקת בין sm ל- cm אילוצים.
>
> וכמו במקבילה הלוגית, רדוקציה לבעיה הזו תקפה רק אם היא נמנעת מהתחום $[sm, cm]$, ונותנת תשובה אמיתית לכל מילה שתורגמה לבעיית אילוצים.

אנחנו נשתמש בשקילות: GAP-CSP היא NP קשה אם"ם משפט PCP. האם אנחנו יכולים סתם ככה לדלג מתוצאה על נוסחאות 3CNF לתוצאה על בעיות אילוצים?

התשובה הקצרה היא – לא.

התשובה הנכונה היא, שהדרישות היחידות שבאמת היינו צריכים מבעיית GAP-E3SAT הן:

א. כל פרמטר בבעיה (כלומר פסוקית), היה צריך להיות תלוי במספר קבוע של משתנים, שיכולים לקבל מספר קבוע של ערכים – במקרה של GAP-E3SAT, 3 הוא מספר המשתנים, $\{1,0\}$ הם הערכים האפשריים. כך מתאפשרת בדיקה של פרמטר יחיד בבעיה (פסוקית, אילוץ וכו') על ידי קריאה של מספר תווים קבוע מההוכחה.

ב. הבעיה המקורית, בגרסת ההחלטה – במקרה הזה היא 3SAT – הייתה צריכה להיות חזקה מספיק כדי לתאר במונחים של כן ולא את התוצאה של מבחן יחיד עבור בעיה ב- NP. זה מאפשר לנו לתאר איטרציה אחת של מבחן רנדומלי בתור בעיה כזו שגודלה קבוע, ולאחד את כל האיטרציות האפשריות לבעיה עם *GAP* משמעותי.

אם אנחנו מסתכלים על מקרה פרטי של *CSP*, שבו תחום האילוצים תמיד קבוע וסופי, וכל אילוץ תלוי במספר קבוע של משתנים (במקרה הבינארי שלנו – 2 משתנים), ברור שהצלחנו לענות על התנאים. המקרה הפרטי מספיק לנו – לא נפרט פה הוכחה לשקילות, אבל כדאי לקרוא לפחות חלק קבוע מההוכחה בחלק ב ולוודא שאני לא משקר חחחחחח משחק מילים חחחחחח פרוטוקול מאמת מוכיח חחחחחח מדהים כמה שאני מצחיק...

מצטער על זה, חזרנו ועכשיו אפשר להתחיל לתכנן.

מה אנחנו רוצים?

להוכיח שקיים s גלובלי שלא תלוי באף מופע של הבעיה, עבורו $\text{GAP-CSP}_{1,s}$ היא NP-קשה. מעבשיו נשמיט את 1 ואת s, ונכתוב פשוט GAP-CSP.

איך נעשה את זה?

שאלה טובה – שכבר ענינו עליה באופן מעורפל, אז כיוון יש לנו.

אנחנו רוצים רדוקציה פולינומיאלית מבעיה NP-קשה כלשהי, לבעיית GAP-CSP. הרדוקציה שלנו צריכה לענות על התנאים הבאים:

- אם התקבלה מילה $w \in L$, אנחנו צריכים להוציא בעיית אילוצים בינארית ספיקה, עם תחום קבוע לכל אילוץ.
- אם התקבלה מילה $w \notin L$, אנחנו צריכים להוציא בעיית דומה, שאינה ספיקה – ויותר מזה, יש לה *GAP* של לפחות s, כלומר אי אפשר לספק בה יותר מאחוז קבוע של אילוצים.

הבעיה ה- NP קשה שנבחר היא זו:

> נתון לנו גרף אילוצים $G = ((V, E), C)$ שבו מספר הקשתות (והאילוצים) הוא m (V היא קבוצת הצמתים, E הקשתות ו- C האילוצים). הוא מגדיר עבורנו בעיית *CSP* בינארי, אבל גם את הבעיה $\text{GAP-CSP}_{1,1/m}$ - כלומר האם בכל פתרון לבעיה, לפחות אילוץ אחד מופר (האם ה- *CSP* ***אינו*** ספיק היא כמובן בעיה שקולה באופן מידי לשאלה אם הוא ספיק[19]).
>
> בפרט, אנחנו רוצים להתחיל עם תחומי משתנים סופיים, אז בעיית הקלט שלנו תהיה $\text{GAP-3COLOR}_{1,1/m}$. היא NP שלמה ,היא מקרה פרטי של GAP-CSP, ויש בה רק 3 צבעים.

בסופו של דבר, אנחנו רוצים לגרום לכך שאם קיים *GAP* כלשהו במופע של הבעיה (ורק אם הוא קיים), הרדוקציה תחזיר לנו בעיה עם *GAP* בסדר גודל קבוע – תוך ביצוע מספר פולינומיאלי של צעדי חישוב.

זה יפה מאוד, אבל לא ענינו על השאלה "איך?"

הממם... אנחנו יכולים לנסות להמציא טכניקה שבה מוסיפים עוד אילוצים לגרף, כאלה שמועדים להישבר. אבל האם נוכל להבטיח שגרף ספיק יישאר ספיק, כשאנחנו מוסיפים אילוצים בלי לפתור קודם לכן את הבעיה? כנראה שיהיה יותר פשוט להשתמש באילוצים הקיימים איכשהו.

הנה הגישה הפשוטה ביותר: נבצע רדוקציה כזו, כך שכל צומת בגרף מקבל וקטור של ערכים עבור ***כל*** הצמתים בגרף, וכל אילוץ מכריח את שני הצמתים שלו לקבל השמה שווה, וכזו שהייתה מספקת את הגרף המקורי. אנחנו מקבלים שגרף ספיק הופך לגרף ספיק (כל משתנה מקבל השמה תקינה עבור כל המשתנים בגרף), וגרף עם *GAP* כלשהו הופך לגרף עם *GAP=1* (אף משתנה לא יכול לקבל השמה שפותרת את הבעיה). הבעיה כאן היא כמובן, שניאלץ לפתור את ה- *CSP* כדי לחשב את האילוצים, ואת זה אנחנו לא יודעים לעשות. (בסוגריים, כיוון שאנחנו כבר מכירים את הקונספט, אנחנו יכולים להוסיף שתחומי האילוצים ברדוקציה הזו ענקיים, הרבה מעבר לחסם הפולינומיאלי – לא נוכל לקרוא אותם, ובטח שלא לחשב אותם).

יש לנו כאן איזשהו "שיווי משקל חישובי" שאנחנו צריכים לשמור עליו. הרדוקציה שלנו צריכה להיות מוגבלת בגודלה, אבל גם להיות נכונה – אנחנו לא יכולים סתם לזרוק אילוצים לכל מקום, לקוות שה- *GAP* שלנו יגדל, אבל גם שהגרף שלנו יישאר ספיק אם הוא צריך להיות כזה. איך נשמור על שיווי משקל כזה? נעבוד לאט, בזהירות ובשלבים.

כדי שבעיית אילוצים ספיקה תישאר ספיקה, אנחנו רוצים להשתמש רק באילוצים שכבר קיימים בה. כדי שבעיית אילוצים לא ספיקה תהפוך לבעיה עם *GAP* גדול, אנחנו רוצים לסדר מחדש את האילוצים האלה כך שהם תלויים אחד

[19] אנחנו מתעסקים בבעיית סיבוכיות, לולאות אינסופיות וההבדל בין הכרעה לזיהוי לא יפריעו לנו (חוץ מזה ש- CSP מעל תחומים סופיים היא בבירור כריעה – כשהתחומים נעשים אינסופיים ולא ניתנים למניה זה כבר לא תמיד נכון, אבל גלשנו קצת מהנושא).

בשני באופן הדוק יותר. ראינו דוגמה כללית מאוד לאופרציה כזו בחלק ג – ואפילו ראינו תרשים של "הגברת *GAP*" לפני שהתחלנו את הדיון על מבחני השמה – אבל עכשיו אנחנו רוצים להגדיר פעולה על גרף אילוצים:

העלאה של גרף אילוצים בחזקת t:

בהינתן גרף אילוצים **d-רגולרי** מעל אלף-בית סופי $G = ((V,E),\Sigma,C)$, שהפרמטרים בו הם אלה:

1. (V,E) הוא 'הגרף עצמו', כלומר הצמתים V והקשתות E – התיאור המרחבי של בעיית האילוצים.
2. Σ היא א"ב האילוצים – זהו מושג שבעבר התייחסנו אליו בתור "תחומי משתנים", אבל כיוון שאנחנו חוסמים את גדלי התחומים, נעדיף להשתמש בשם א"ב – המשמעות דומה, כל משתנה ב- V יכול לקבל אחד מהערכים בקבוצה Σ.
3. C היא קבוצת האילוצים – ישנה התאמה חח"ע ועל בין E ל- C, שקובעת איזה $c \in C$ הוא האילוץ המתאים לקשת $e \in E$. אותו ה- $c \in C$ הוא קבוצה של זוגות סדורים מעל Σ שמתאימים לזוגות הצמתים $e = (u,v)$.

אז בהינתן הגרף הזה, אנחנו מגדירים את הגרף $G^t = \left((V,E^t),\Sigma^{d^{\lceil t/2\rceil}},C^t\right)$ כך:

1. הצמתים: קבוצת הצמתים V של G^t היא בדיוק קבוצת הצמתים V של G.
2. הקשתות: עבור כל מסלול באורך t בין $u,v \in V$, קיימת קשת $e^t \in E^t$ $(u,v) =$. אמרנו *עבור* ***כל*** *מסלול*, וזה מאפשר יותר מקשת אחת בין u ל- v ב- E^t. אם מספר המסלולים ביניהם (באורך t) הוא k, יהיו k קשתות מקבילות $(u,v) \in E^t$.
3. הא"ב: אנחנו מגדירים "שכונה" של כל צומת בגרף G. השכונה של v היא כל הצמתים במרחק $\lceil t/2\rceil$ ממנו, והיא מסומנת ב- $\Gamma(v)$. כל צומת u שמסלול באורך $\lceil t/2\rceil$ מחבר בינו לבין v הוא אחד מהאיברים ב- $\Gamma(v)$:
$$\Gamma(v) = \{u \in V : there\ is\ a\ \lceil t/2\rceil\ length\ path\ between\ u,v\}$$
בגרף d-רגולרי $d^{\lceil t/2\rceil}$ הוא חסם עליון על גודל שכונה - כל צעד במסלול שאורכו $\lceil t/2\rceil$ יכול לבחור בדיוק אחת מבין d קשתות, ובהנחה שאף שניים מהמסלולים לא מתלכדים נגיע לכל היותר ל- $d^{\lceil t/2\rceil}$ צמתים.
הא"ב שלנו הוא סדרות מעל σ, באורך שווה לגודל המקסימלי של שכונה: $\Sigma^{d^{\lceil t/2\rceil}}$, והוא מאפשר לכל צומת לתת ערך ב- Σ לכל אחד מהשכנים שלו.
4. האילוצים: אנחנו מגדירים אילוץ פשוט יחסית: כל קשת ב- E^t מחברת שני צמתים ב- V, ולכל אחד מהצמתים האלה יש שכונה. אנחנו מסתכלים על האיחוד של שתי השכונות. זהו תת גרף של G שאנחנו יכולים לחסום את הגודל שלו – הגודל של כל שכונה הוא איזשהו מספר סופי של צמתים שתלוי ב- d וב- t – אם נניח ששני הערכים האלה הם $O(1)$ (כיוון שהם לא תלויים במספר הקשתות ב- G), אז איחוד השכונות מחזיר תת-גרף של G שמהווה *CSP* בגודל קבוע, ופתיר בסיבוכיות זמן ומקום קבועות. האילוץ שלנו הוא שההשמה לכל שני צמתים מחוברים פותרת את הבעיה עבור תת הגרף הזה. פורמלית, האילוץ שלנו נראה כך:
תהי $e^t = (u,v) \in E^t$. קבוצת הקשתות $E \cap \left(\Gamma(u)\times\Gamma(v)\right)$ היא תת-בעיית-אילוצים בגודל קבוע, והאילוץ $c^t(e^t)$ מסופק על ידי השמות ל- u ו- v אם ההשמות מספקות את תת הבעיה (נדרשת כמובן הסכמה בין u ו- v על הערך הניתן לכל צומת w שנמצאת בחיתוך שתי השכונות).

המטרה של העלאת גרף בחזקה היא להגביר את ה- *GAP*. לפני שננתח את הפעולה בצורה מדויקת, כדאי לנסות להבין מה הרציונל שמאחוריה.

התחלנו עם גרף G בגודל כלשהו, כשהגודל תלוי בקבוצת הקשתות $|E|$. העברנו אותו לגרף G^t שגדול ממנו – כמה גדול? אנחנו יכולים לחשב במדויק את מספר הקשתות ב- E^t: כל מסלול ב- E^t מתחיל ונגמר באיזושהי קשת, ולכל קשת יש בדיוק d^{t-1} מסלולים כאלה שמתחילים או נגמרים בה, ולכן $|E^t| = \frac{|E|}{2}\cdot d^{t-1}$.

אם G היה ספיק, אז ההשמה σ שמספקת אותו יכולה גם לספק את כל תת הבעיות ב- G^t, ולכן שמרנו על הספיקות של גרף ספיק מלכתחילה.

אם הגרף G לא היה ספיק, אנחנו יכולים להסתכל על השמה σ אופטימלית (כזו שמספקת את מספר האילוצים המקסימלי, או בשם אחר שכבר הכרנו – OPT). ל- σ יש לפחות קשת אחת ב- G שהיא שוברת, נקרא לה e, ואנחנו יכולים להעריך באופן גס (מאוד) שכיוון ש- σ אופטימלית, בגרף G^t החדש כל "תת בעיה" שכוללת את e אינה פתירה.

כמה בעיות כאלה יש? בערך $d^{t-1} \cdot d^{t/2}$ כאלה, בחירה בין בערך $d^{t/2}$ צמתים ש- e נמצאת בשכונה שלהם, ועוד d^{t-1} צמתים שמגדירים אילוץ עם הצומת הראשון שנבחר. כיוון שהגרף כולו גדל בפקטור של d^{t-1}, האם הצלחנו להגדיל את ה- *GAP* בסדר גודל של $d^{t/2}$? אולי, אבל את החסם הזה לא נוכיח.

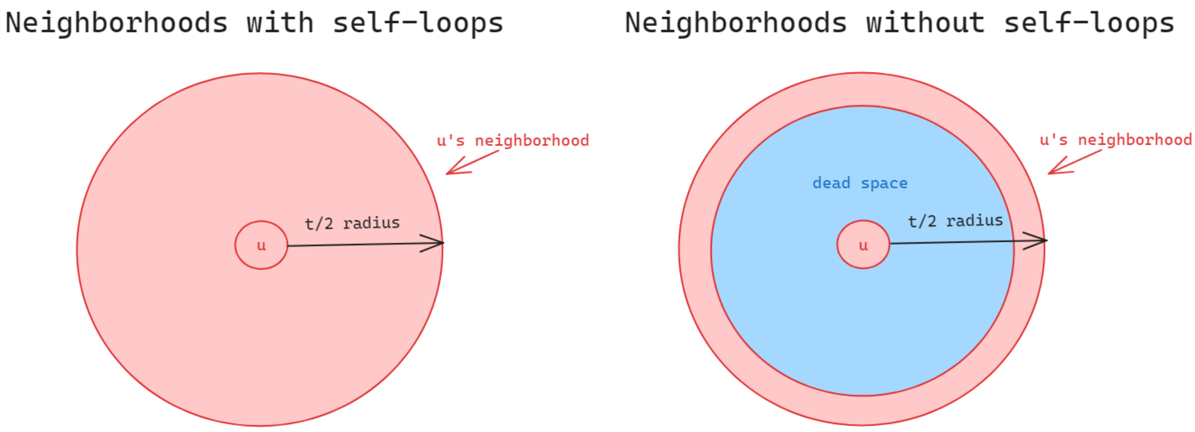
בכל מקרה, בהערכה שלנו יש כמה פרמטרים מאוד לא מדויקים – הראשון הוא ההנחה שאם תת-בעיה מכילה את הקשת הבעייתית e, היא אינה פתירה. הקשת בעייתית כיוון שהגרף G כולו בעייתי, ואין שום דבר שמחייב תת בעיות בגרף להתייחס לבעיה הכוללת G – בינתיים. הפרמטר השני הוא הניפוח שלנו מקשת בעייתית אחת אל תת-בעיה שלמה. בתוך אותה תת הבעיה יכולות להיות כל הקשתות הבעייתיות שב- G, והחפיפה שנוצרת במקרה הזה פוגעת משמעותית בהגדלת ה- *GAP*.

הפתרון עבור שני הבעיות הוא שימוש בתכונות של גרפים מרחיבים – כשדיברנו עליהם, ראינו איך הם גורמים לקשירות רחבה בין כל חלקי הגרף, באופן שמקשה על חלוקה שלו לאגפים נפרדים. מכאן אנחנו מסיקים שתי מסקנות:

- אם תת בעיה כלשהי בגרף שמוגדרת על ידי הצמתים z ו- w תשנה את ההשמה המתקבלת עבורה כדי לספק קשת בעייתית e, יהיו מספיק בעיות חופפות במידה כזו או אחרת אליה, שייאלצו גם כן לשנות את הפתרונות שלהן כדי להתאים לשינויים ש- z ו- w קבעו. אם הקשת הבעייתית e היא כזו שמגיעה מהשמה אופטימלית, השינויים האלה קרוב לוודאי ירעו את ההשמה הכוללת לבעיה. באיזשהו מובן, אם לבעיה יש תכונות של הרחבה – גם לאחר חלוקה שלה לתת בעיות, אנחנו יכולים להניח שיש בעיה אחת כללית שמנהלת את כולן.
- בגרף מרחיב, לא ייווצר מצב שחלק גדול מהקשתות הבעייתיות עבור השמה מסוימת יימצאו באזור מסוים בגרף – פשוט כי אין אזורים כאלה, וכל ההתפלגויות עבור מיקומי 'קשתות בעייתיות' מקבלות צורה של התפלגות פחות או יותר אחידה.

ומכאן אנחנו מגיעים לעוד הערכה להגברת ה- *GAP*, אחת שכן ננסה להוכיח: נניח שבגרף G אנחנו בוחרים באופן רנדומלי t קשתות. מה ההסתברות שאחת מהן לפחות בעייתית? עבור $Gap(G) = x$, נקבל tx. בהנחה ש- G היה מרחיב, מסלול רנדומלי באורך t יהיה קרוב לבחירה בהתפלגות אחידה של t קשתות. זאת אומרת שלכל קשת שהגדרנו ב- E^t, ההסתברות *לעבור ממש* בתוך קשת בעייתית היא בערך $t \cdot Gap(G)$.

לפי ההגדרה שלנו, העלאת גרף בחזקה מוסיפה מסלולים בתור קשתות, אבל המסלול כולו לאו דווקא נוכח בתוך תת הבעיה שהוא מייצר – הכבוד הזה שמור אך ורק לצמתים במרחק $\lceil t/2 \rceil$ מקצות המסלול. בגלל שלחשוב על פתרונות מתוחכמים עושה כאב ראש, אנחנו פשוט נדאג להוסיף קשתות עצמיות בכל צומת בגרף – עכשיו כל מסלול באורך קטן מ- $\lceil t/2 \rceil$ יכול להיחשב ארוך יותר, אם נאמר שחלק מהצעדים בו הם מצומת לעצמה. כמובן, מסלול באורך t הוא פשוט שני מסלולים באורך $\lceil t/2 \rceil$ שנפגשים באמצע.



אם מסלול עובר בקשת בעייתית e, הלולאות העצמיות מכריחות את אותה הקשת להיות נוכחת בתת הבעיה שהוא מגדיר. אם הגרף מרחיב מספיק, ההסתברות שמסלול עובר בקשת כזו היא בערך $t \cdot Gap(G)$, ואנחנו יכולים להעריך ש- $Gap(G^t) \geq \sim tGap(G)$.

טכנית, אנחנו נוכיח חסם חלש יותר: $Gap(G^t) \geq O(\sqrt{t}) \cdot Gap(G)$, אבל בשביל הטכניקה הכוללת שנשתמש בה, החסם הזה מספיק.[20] [21]

מה אנחנו יכולים לעשות עם החסם הזה? העלאת גרף בחזקה גורמת לעלייה בסדר גודל קבוע (תלוי ב- d ו- t) של ייצוג הבעיה. את ה- *GAP* היא מגבירה גם כן בפאקטור קבוע, של $O(\sqrt{t})$. נניח שקיבלנו גרף G עם *GAP* של קשת אחת ($1/m$), ואנחנו רוצים להפוך אותו לגרף H עם *GAP* של $1/s$ עבור s קבוע. אנחנו מקבלים את המשוואה:

$$\frac{1}{m} \cdot O(\sqrt{t})^x = \frac{1}{s}$$

כאשר x הוא מספר הפעמים שנצטרך לבצע העלאת גרף בחזקה. מהנוסחה הזו אנחנו יכולים לחשב:

$$O(\sqrt{t})^x = \frac{m}{s}$$
$$\Longrightarrow x = \log_{O(\sqrt{t})}\left(\frac{m}{s}\right)$$
$$\Longrightarrow x = O(\log(m))$$

כל רדוקציה של העלאת גרף בחזקה מגדילה את הבעיה המתקבלת בפאקטור קבוע של d^{t-1}, ולכן לאחר x איטרציות אנחנו מקבלים:

$$size(H) = size(G) \cdot (d^{t-1})^{O(\log(m))} = poly(size(G))$$

כל אחד מהשלבים ברדוקציה שלנו יהיה חישוב פולינומיאלי על גודל הבעיה, וכיוון שהבעיה עצמה תמיד נשארת בגודל פולינומיאלי של הבעיה המקורית (הקלט לרדוקציה שלנו), הרדוקציה שלנו כולה פולינומיאלית – יש!

כמה סיוגים:

א. העלאת גרף בחזקה דורשת קלט d-רגולרי, ולא מובטח לנו כזה.
ב. החסם $O(\sqrt{t})$ על הגברת ה- *GAP* דורש גרף רגולרי ומרחיב עם לולאות עצמיות בכל צומת – לא מובטח לנו כזה.

שני הבעיות האלה קלות יחסית לפתרון, ועוד מעט נטפל בהן.

ג. בכל החישוב המסובך שלנו התעלמנו לגמרי מהגדילה של א"ב האילוצים, מ- $|\Sigma|$ אל $\left|\Sigma^{d^{\lceil t/2 \rceil}}\right|$. הבעיה הזו לא פשוטה בכלל – אבל כבר טיפלנו בה, כשהראינו איך מבחן השמה נותן לנו להקטין את הא"ב לגודל קבוע, תוך שמירה על ה- *GAP* קטן לכל היותר בסדר גודל קבוע. הא"ב שלנו אומנם גדל באופן משמעותי, אבל הניפוח עדיין תלוי רק ב- d, t ו- $|\Sigma|$ – מכאן אנחנו מסיקים שהעלאת גרף בחזקה מגדילה את **כל** הייצוג שלו לכל היותר פי:

$$d^{t-1} \cdot \left|\Sigma^{d^{\lceil \frac{t}{2} \rceil}}\right|$$

---

[20] החסם שאנחנו מראים הוא אותו החסם שאירית דינור מוכיחה, אבל אנליזה מתקדמת יותר של ג'איקומר ראדהקרישנאן הצליחה לשפר את החסם של על הגברת ה- GAP לפאקטור של $O(t)$

[21] החסם $O(d^{t/2})$ מצריך ניתוח של תלויות בין משתנים באופן שלא פיתחנו עבורו כלים - קשה לומר באופן כללי מתי קשת נמצאת בתת בעיה שמפרה אותה. במסלול, הרבה יותר קל לנתח את הסבירות שצומת ההתחלה והסיום מפרות בעצמן את הקשת המדוברת. גם החסם $O(\sqrt{t})$ מצריך ניתוח לא טריוויאלי, אבל הוא פותר כמה בעיות תלות בכך שהוא מונע את קיומן מלכתחילה.

ובהנחה ש- Σ נשאר א"ב בגודל קבוע, אנחנו עדיין מקבלים $size(G^t) = O(1) \cdot size(G)$

עדיין - אנחנו צריכים לחשב מחדש את הפרמטרים של הרדוקציה, תוך התייחסות לבעיות שהעלינו. אבל החל מעכשיו, אנחנו מתחילים לעבוד בצורה פורמלית, עם הוכחות אמיתיות. בתור התחלה, אנחנו צריכים להגדיר כמו שצריך כמה טענות.

למת העיבוד המקדים – Preprocessing lemma:

יהי $G = \big((E,V),\Sigma,C\big)$ גרף אילוצים. קיימים קבועים λ, d, β_1 כאשר $\lambda < d$, ורדוקציה פולינומיאלית בגודל G כך שהפעלת הרדוקציה על G מחזירה גרף אילוצים $G' = \big((V',E'),\Sigma',C'\big)$ עם הפרמטרים הבאים:

א. G' הוא d-רגולרי עם קשתות עצמיות בכל צומת (יש קשת מכל צומת אל עצמו, ועוד $d-2$ קשתות אל צמתים אחרים, כך שדרגת כל צומת היא d). בנוסף, הגרף מרחיב – הערך העצמי השני (בערך מוחלט) של המטריצה המייצגת של G קטן או שווה ל- λ, כלומר:
$$\lambda(G') \le \lambda < d$$

ב. א"ב האילוצים של G' לא משתנה, כלומר $\Sigma' = \Sigma$.

ג. הגודל של G' לינארי בגודל G, כלומר $size(G') = O\big(size(G)\big)$.

ד. ערך ה- *GAP* של G' קטן לכל היותר פי β_1 מזה של G: $\beta_1 \cdot Gap(G') \ge Gap(G)$ – כלומר חלה ירידה לינארית לכל היותר באי-ספיקות הגרף.

ה. אם G ספיק גם G' ספיק.

אנחנו נשתמש בלמה הזו כדי לטפל בסיוגים א ו- ב שהעלינו. את ההוכחה שלה נשמור לשלב מאוחר יותר.

למת ההגברה – Amplification lemma:

יהיו λ, d קבועים, $\lambda < d$, ו- Σ א"ב אילוצים סופי. קיים קבוע $\beta_2 > 0$ שתלוי רק ב- $|\Sigma|, d, \lambda$ כך שלכל גרף אילוצים $G = \big((V,E),\Sigma,C\big)$ d-רגולרי, עם ערך עצמי שני $\lambda(G) \le \lambda$, וקשתות עצמיות בכל צומת, נקבל לאחר העלאת הגרף בחזקת t:

$$Gap(G^t) \ge \beta_2\sqrt{t} \cdot \min\left(\frac{1}{t}\ , Gap(G)\right)$$

הלמה הזו מבטיחה לנו את הגברת ה- *GAP* בפאקטור של $O\big(\sqrt{t}\big)$. את ההוכחה שלה גם כן נדחה בינתיים.

למת הקומפוזיציה – Composition lemma:

יהי AT מבחן השמה של 2 שאילתות, שמירת מרחק γ וא"ב Σ'. יהי $\beta_3 > 0$.

כל גרף אילוצים $G\big((V,E),\Sigma,C\big)$, יכול להפוך בזמן פולינומיאלי לגרף אילוצים חדש $G'\big((V',E'),\Sigma',C'\big)$, ששומר על התנאים הבאים:

1. $size(G') \le O(1)size(G)$, כלומר אפשר לחסום את הגדילה הכוללת של ייצוג הבעיה על ידי קבוע.
2. אם G היה ספיק, גם G' ספיק.

3. $Gap(G') \geq \beta_3 \cdot Gap(G)$, כלומר היחס בין המספר המינימלי של אילוצים מופרים למספר האילוצים הכולל, יקטן בסדר גודל קבוע לכל היותר במעבר מ- G ל- G'.

זוהי אותה למת הקומפוזיציה שהוכחנו בחלק ג, עם כמה שינויים בסימני הפרמטרים, ואיחוד השלבים ד ו- ה מאלגוריתם הקומפוזיציה ליצירת מבחן השמה של 2 שאילתות. היא תפתור לנו את סיוג ג.

שלושת הלמות הללו מבטיחות לנו שלושה אלגוריתמי רדוקציה שונים, עם הבטחות לגבי גודל הפלט וגודל ה- *GAP* שלו:

- כל האלגוריתמים שומרים על גדילה לינארית לכל היותר של הפלט ביחס לקלט.
- למת העיבוד המקדים לוקחת בעיית *CSP* בינארי כלשהי, והופכת אותה לבעיה מרחיבה d-רגולרית ומרחיבה עם לולאות עצמיות בכל צומת, תוך שמירה על ספיקות הגרף וירידה של פי β_1 לכל היותר בערך ה- *GAP*.
- למת ההגברה לוקחת את הבעיה הרגולרית והמרחיבה עם לולאות עצמיות שמובטחת על ידי למת העיבוד המקדים, והופכת אותה לבעיה עם *GAP* גדול פי $\beta_2\sqrt{t}$ (לכל t טבעי שנבחר), עד נקודת רוויה של $1/t$. תוך כך, היא מגדילה את א"ב האילוצים פי איזשהו קבוע.
- למת הקומפוזיציה לוקחת את הבעיה המוגברת, ומקטינה את הא"ב בחזרה לגודל קבוע, תוך שמירה על ירידה של β_3 לכל היותר בערך ה- *GAP*.

את שלושת האלגוריתמים האלה אנחנו יכולים להרכיב ביחד ל**צעד ההגברה** שלנו – אלגוריתם שלוקח גרף אילוצים, מגדיל את תכונת ה- *GAP* שלו פי $O(\sqrt{t})$, תוך שמירה על אותו א"ב אילוצים וגדילה לינארית לכל היותר בגרף:

משפט ההגברה:

קיים א"ב סופי Σ_0, כך שלכל א"ב סופי Σ, קיימים קבועים $K > 0$ ו- $1 > \alpha > 0$, כך שלכל גרף אילוצים מעל Σ $G = ((V,E),\Sigma,C)$, אפשר בזמן פולינומיאלי לבנות גרף אילוצים $G' = ((V',E'),\Sigma_0,C')$ שמקיים את התנאים הבאים:

א. $size(G') \leq K \cdot size(G)$ – הבעיה כולה גדלה לכל היותר באופן לינארי.
ב. אם הגרף G ספיק גם G' ספיק.
ג. $Gap(G') \geq \min(2Gap(G), \alpha)$ – אי הספיקות של G' הכפילה את עצמה, עד לנקודת רוויה של α.

בפשטות, המשפט הזה מתקבל מהרכבת הלמות שלנו אחת על השנייה לפי הסדר. לפני שאנחנו מוכיחים את המשפט, אנחנו רוצים לעבור על כל הפרמטרים בו ולהבין את המשמעות של כל אחד.

המשפט מתחיל ב- "קיים א"ב סופי Σ_0..." כך שגרף אילוצים מעל א"ב כלשהו הופך לגרף אילוצים מעל Σ_0. הטענה כאן היא שבעזרת מבחן השמה מתאים, בעיה מעל Σ הופכת לבעיה מעל Σ_0 – כלומר Σ_0 הוא פרמטר שתלוי במבחן השמה כלשהו – אנחנו יודעים שמבחן ההשמה קיים, אבל הוא לא רלוונטי לטענה הפורמלית של המשפט.

אנחנו ממשיכים בהכרזה על שני קבועים, α ו- K:

K הוא החסם על הגדילה המקסימלית של ייצוג הבעיה במעבר מ- G ל- G'. הוא מייצג את מכפלת הגדילה בכל אחד מהצעדים שהבטחנו בשלושת הלמות שלנו – כל אחת מהן בתורה מגדילה את הבעיה פי קבוע כלשהו, נניח שאלו הקבועים K_1, K_2, K_3. אנחנו רוצים לבצע את האלגוריתמים המובטחים אחד אחרי השני, ולחסום את הגדילה שלנו על ידי $K = K_1 \cdot K_2 \cdot K_3$.

α הוא 'נקודת הרוויה' של צעד ההגברה. באיזשהו שלב ה- *GAP* של הבעיה גדול מספיק כדי שהעלאת הגרף בחזקת t כבר לא תוכל לשפר אותו (זה קורה כאשר 'אזור צפוף בקשתות בעייתיות' הוא כל הגרף שלנו). עד שאנחנו מגיעים לנקודת הרוויה הזו, אנחנו מבטיחים אלגוריתם פולינומיאלי שמגדיל את ה- *GAP* פי 2 לפחות – כלומר, אנחנו באופן

משתמע מכריזים על $t \in \mathbb{N}$ שעבורו העלאת גרף בחזקה מבצעת הגברת *GAP* מספיקה כדי שכל הירידות הלינאריות ב- *GAP* ששלושת הלמות הזהירו אותנו מהן, עדיין לא יוכלו להוריד את $O\left(\sqrt{t}\right)$ מתחת לסף של 2.

הוכחת משפט ההגברה:

יהי $Prep$ האלגוריתם המובטח בלמת העיבוד המקדים, AT מבחן השמה של 2 שאילתות מעל Σ_0, ו- t קבוע טבעי שנחשב בהמשך. אנחנו מגדירים:

$$G' = AT \circ \left(Prep(G)\right)^t$$

כלומר G' הוא ההרכבה של פונקציית מבחן ההשמה על $Prep(G)$ בחזקת t.

יהיו K_1, K_2, K_3 קבועים החוסמים את הגדילה הלינארית של הפלט עבור $Prep, G^t, AT$ בהתאמה. נשים לב ש- K_2 תלוי ב- t.

אנחנו יודעים ש:

$$\begin{aligned} Gap(G') &= Gap\left(AT \circ \left(Prep(G)\right)^t\right) \\ &\geq \beta_3 \cdot Gap(Prep(G)^t) \\ &\geq \beta_3\beta_2\sqrt{t} \cdot \min\left(Gap\big(Prep(G)\big), \frac{1}{t}\right) \\ &\geq \beta_3\beta_2\sqrt{t} \cdot \min\left(\beta_1 Gap(G), \frac{1}{t}\right) \end{aligned}$$

כאשר אי השוויונות נובעים לפי הסדר מלמת העיבוד המקדים, למת ההגברה ולמת הקומפוזיציה. מאותן הלמות אנחנו מסיקים שאם G ספיק גם G' ספיק – כלומר תנאי ב של משפט ההגברה מוכח.

אבל עכשיו, הוכחת המשפט הופכת למציאת פתרונות α ו- t עבור אי השוויונות:

$$\beta_3\beta_2\sqrt{t}\beta_1 \geq 2$$

$$\beta_3\beta_2\sqrt{t} \cdot \frac{1}{t} \geq \alpha$$

ומייד אנחנו רואים שקביעת:

$$t = \left\lceil \left(\frac{2}{\beta_3\beta_2\beta_1}\right)^2 \right\rceil$$

$$\alpha = \beta_3\beta_2\sqrt{t} \cdot \frac{1}{t} = \frac{(\beta_3\beta_2)}{\sqrt{t}}$$

מוכיחה עבורנו את תנאי ג של משפט ההגברה.

עכשיו שקבענו את t, גם K_2 – קבוע הגדילה של העלאת גרף בחזקת t – נתון לנו. מכאן נקבע את K להיות $K_1K_2K_3$ ונקבל:

$$size(G') \leq K \cdot size(G)$$

כיוון ששלושת הלמות מבטיחות רדוקציות פולינומיאליות, קיבלנו שהאלגוריתם המובטח על ידי משפט ההגברה הוא שרשור של אלגוריתמים פולינומיאליים, כלומר פולינומיאלי בעצמו.

הדרך ממשפט ההגברה אל הוכחת משפט PCP לא ארוכה – מבצעים את הרדוקציה $\log(m)$ פעמים, עד שמקבלים בעיית אילוצים עם *GAP* של α, ומשתמשים בשקילות הידועה. אבל עדיין חסרות לנו למת העיבוד המקדים ולמת ההגברה – השלב הבא הוא להוכיח אותן.

למת העיבוד המקדים – הוכחה:

המטרה שלנו היא להראות רדוקציה פולינומיאלית, שלוקחת גרף אילוצים $G = \big((V,E),\Sigma,C\big)$ והופכת אותו לגרף אילוצים d-רגולרי ומרחיב, עם לולאות עצמיות. הרדוקציה צריכה לשמור על הפרמטרים הבאים:

- א"ב האילוצים נשאר כמו שהוא.
- הייצוג הכולל של הגרף המתקבל גדול פי קבוע K_1 לכל היותר.
- ערך ה- *GAP* של הגרף המתקבל קטן פי קבוע β_1 לכל היותר.
- הרדוקציה שומרת על ספיקות של גרף ספיק.

הרדוקציה שלנו מורכבת משני חלקים. חלק א גורם לגרף להפוך לרגולרי, וחלק ב לוקח את הגרף הרגולרי המתקבל והופך אותו למרחיב עם לולאות עצמיות. כשדיברנו על גרפים מרחיבים, הזכרנו שיש לנו בניות פולינומיאליות עבורם – עכשיו אנחנו צריכים את הבניות האלה. הציטוט המדויק:

"קיים $\mathrm{d} \in \mathbb{N}$ ו- $\mathrm{d} > \lambda > 0$ כך שלכל n, ישנה בנייה פולינומיאלית של גרף מרחיב d-רגולרי עם ערך עצמי שני קטן או שווה בערכו המוחלט ל- λ על n צמתים. הגרף המרחיב יכול להכיל לולאות עצמיות (מצומת לעצמו), וקשתות מקבילות (יותר מקשת אחת בין זוג צמתים)."

המונח "ערך עצמי שני בערך מוחלט" קצת מסורבל, אז מעכשיו נאמר "ע"ע שני של גרף", כאשר הכוונה היא לערך המוחלט שלו.

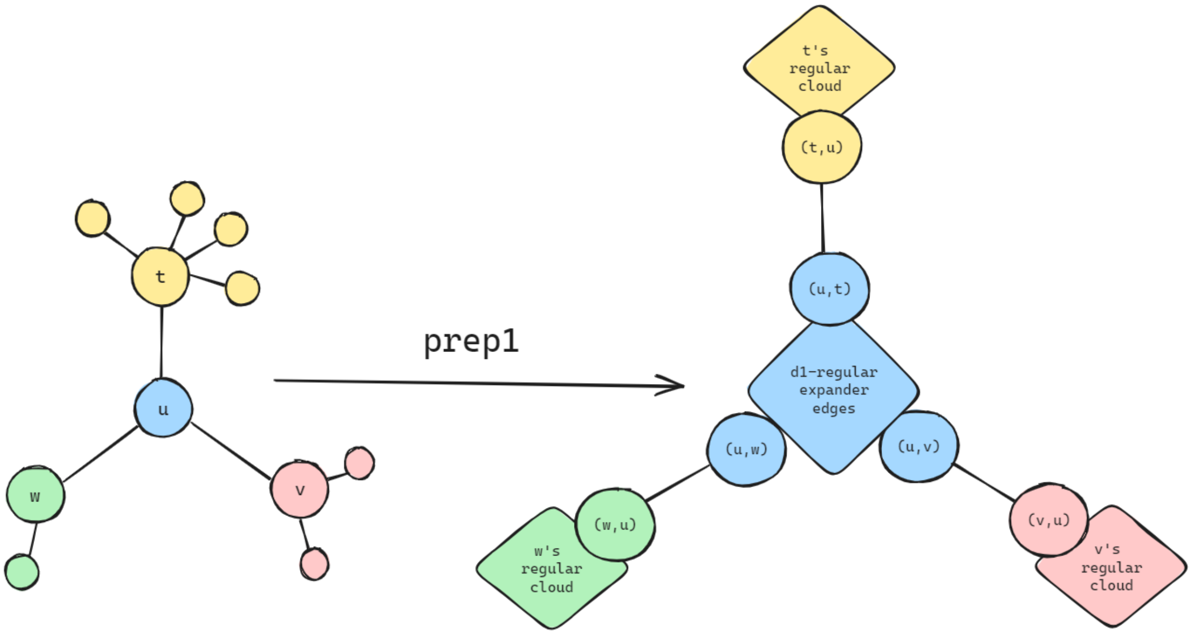
האלגוריתם $prep_1$ – Expander Replacement of Papadimitriou and Yannakakis:

אנחנו לוקחים את $G = \big((V,E),\Sigma,C\big)$ ובונים ממנו את $G_1 = \big((V_1,E_1),\Sigma,C_1\big)$ כך:

לכל צומת $v \in V$ אנחנו מגדירים את הגרף G_v כך: אם בגרף G קיימת הקשת $e = (v,u)$, ב- G_v e מייצג צומת. זאת אומרת שב- G_v מספר הצמתים הוא הדרגה של צומת v. כדאי לשים לב שאם ב- G הייתה לנו קשת עצמית (v,v), בגרף G_v היא הופכת לשני צמתים נפרדים. על קבוצת הצמתים המתקבלת אנחנו פורשים גרף מרחיב d-רגולרי עם ע"ע שני קטן מ- λ, ומקבלים את G_v. אנחנו נקרא לגרף הזה 'הענן של v'. אנחנו מגדירים אילוץ של שוויון על כל הקשתות באותו ענן – כדי להכריח את הענן כולו לייצג ערך יחיד, כמו ש- v היה מייצג ב- G.

את כל העננים המתקבלים אנחנו מאחדים על ידי 'קשתות חוצות-ענן', שהן בעצם הקשתות מהגרף המקורי G. לדוגמה, הקשת $e = (u,v)$ מ- G הפכה לשני צמתים: (v,u) בענן של v ו- (u,v) בענן של u. אנחנו מקשרים את העננים עם קשת $e_1 = \langle (v,u),(u,v) \rangle$, שנושאת את האילוץ $c_1(e_1) = c(e)$ – כלומר האילוץ המקורי על u ו- v. במקרה של הקשת העצמית (v,v), הקשת חוצת-הענן נשארת בתוך אותו הענן של v.

איחוד העננים המתקבל הוא הגרף G_1 שלנו. אנחנו טוענים שהוא רגולרי, גודלו הכולל חסום על ידי גודל G כפול קבוע כלשהו, הוא ספיק אם G ספיק וערך ה- *GAP* שלו קטן בקבוע לכל היותר מזה של G.



הרעיון ברדוקציה שביצענו היה די פשוט – אנחנו לוקחים כל צומת, והופכים אותו לענן של צמתים. בין העננים אנחנו שומרים על אותם היחסים שהיו לנו בגרף המקורי. כדי לדאוג שכל ענן מתנהג באמת בתור יחידה אחת, אנחנו דואגים להשרות אילוצי שוויון בין הצמתים, והרחבה על הענן – כולם צריכים לקבל את אותו הערך, ואין אף קבוצה של צמתים שמסוגרת מספיק כדי שמספר אילוצי השוויון עליה יהיה זניח.

אנחנו יכולים להשתכנע מהר ש- G_1 הוא $(d+1)$-רגולרי: כל צומת בו הוא מרכיב בגרף d-רגולרי של הענן המתאים, והוספנו לכל צומת קשת חוצת-ענן יחידה.

באותה המידה אנחנו יכולים להשתכנע שגודל הגרף לא גדל ביותר מקבוע: אם ב- G מספר הקשתות $|E|$ היה m, ב- G_1 יש לנו $2m$ צמתים $(d+1)$-רגולריים, כלומר $\frac{2m(d+1)}{2}$ קשתות סך הכל. גודל האילוצים עדיין חסום על ידי המקסימום ש- Σ מאפשר.

אם G ספיק, אנחנו יכולים לתת את אותה ההשמה שמספקת אותו לעננים שלמים ב- G_1 – ולענות גם על אילוצי השוויון בתוך העננים, וגם על האילוצים המקוריים מ- G בין העננים.

הטענה המעניינת היא החסם הלינארי שמקבלים על ה- *GAP*. אנחנו רוצים להוכיח שקיים β כזה כך ש:

$$Gap(G_1) \geq \beta \cdot Gap(G)$$

כדי להוכיח את הטענה, אנחנו מסתכלים על השמה אופטימלית (OPT) ל- G_1. ממנה אנחנו מחלצים השמה σ ל- G כך: בכל ענן של צומת v, הערך הפופולרי ביותר מ- Σ בענן הוא ההשמה ש- σ נותנת ל- v ב- G. בכתיב מתמטי, נסמן את קבוצת הצמתים שנוצרת מ- v בתור $cloud(v)$ ונקבל את σ:

$$\sigma(v) = \arg\max_{a\in\Sigma} |\{z : z \in cloud(v) \ \ and \ \ OPT(z) = a\}|$$

בתור השמה לגרף עם *GAP*, σ מפרה לפחות $Gap(G)$ מהאילוצים על קשתות ב- G. את כל הקשתות האלה נסמן ב- $F \subseteq E$. אותו הדבר נעשה עבור OPT – כל האילוצים שהיא מפרה ב- G_1 יסומנו על ידי F_1. עבור כל קשת ב- F, יש קשת חוצת ענן מקבילה ב- G_1, והקצוות שלה מקבלים מ- OPT את אותה ההשמה ש- σ הביאה להם ב- G, או השמה אחרת (אחד לפחות מהקצוות מקבל ערך שונה).

אם הקצוות קיבלו את אותם הערכים, נסיק שהאילוץ חוצה הענן נשבר, כיוון שב- σ אותם הערכים בדיוק הובילו לשבירה של אותו האילוץ בדיוק. אחרת, אחד הקצוות מקבל ערך *שאינו* הערך הפופולרי בענן שלו, ואנחנו יכולים לצפות לשבירה של אילוצי שוויון.

לכל ענן $cloud(v)$, אנחנו מגדירים את S_v בתור קבוצת הצמתים שלא קיבלו את הערך פופולרי, כלומר ערך אחר מזה שעבר ל- σ. כל S_v מורכבת מ- $|\Sigma|-1$ קבוצות של תווי א"ב אפשריים, וכל קבוצה כזו תסומן עם הערך המתאים, כלומר:

$$S_v[b] = \{z \in S_v : OPT(z) = b, b \in \Sigma \; is \; not \; \sigma(v)\}$$

אנחנו מאחדים את כל הקבוצות S_v לקבוצה יחידה S, ושמים לב לאי השוויון הבא – הגדרנו את F, קבוצת הקשתות השבורות ב- G לפי σ, ואת F_1, קבוצת הקשתות השבורות ב- G_1 לפי OPT. אנחנו מקבלים:

$$|F_1| + |S| \geq |F|$$

כיוון שכל קשת ב- F מקבילה לקשת חוצת-ענן שבורה, או לקשת שמוסיפה צומת ל- S.

עכשיו אנחנו מחלקים למקרים:

מקרה א – שבו $|S| \leq \frac{|F|}{2}$:

במקרה הזה אנחנו מייד מקבלים חסם לינארי ל- *GAP* - לפי אי השוויון $|F_1| + |S| \geq |F|$:

$$S \leq \frac{|F|}{2} \Longrightarrow |F_1| \geq \frac{|F|}{2}$$

ואנחנו יודעים שמספר הקשתות ב- G_1 לינארי בזה של G (שבו $|E|$ הוא מספר הקשתות), אז:

$$Gap(G_1) = \frac{|F_1|}{|E_1|} \geq \frac{1}{d+1} \cdot \frac{1}{2} \cdot \frac{|F|}{|E|} = \frac{1}{2(d+1)} \cdot Gap(G)$$

מקרה ב – שבו $|S| \geq \frac{|F|}{2}$:

כאן אנחנו משתמשים בתכונת ההרחבה של העננים. כזכור, יש לנו קשר בין שתי ההגדרות להרחבה, זו של הרחבת הקשתות וזו של הערך העצמי השני:

$$d - \lambda \; \leq 2 \cdot \left(\min_{\substack{S \subseteq V \\ |S| \leq \frac{|V|}{2}}} \left(\frac{|E(S, S')|}{|S|} \right) \right)$$

שאומר לנו מה המספר המינימלי של קשתות יוצאות מכל תת קבוצה קטנה מחצי של צמתים בגרף מרחיב עם ע"ע שני λ. אנחנו נגדיר את h בתור החסם הזה, כלומר:

$$h = \frac{(d-\lambda)}{2}$$

ועכשיו, אנחנו מסתכלים על ענן $cloud(v)$ כלשהו. בתוך הענן הזה הגדרנו את הקבוצות $S_v[b]$ בתור קבוצות הצמתים שמקבלות את הערך הלא-פופולרי b, לכל $b \in \Sigma$. כיוון שהערכים האלה לא פופולריים, כל קבוצה כזו עונה על התנאי "אני קטנה מחצי הצמתים בענן", והחסם h תקף לגבי כל אחת. אנחנו גם יודעים שכל קשת שיוצאת מתת קבוצה כזו יוצאת אל תת קבוצה עם ערך אחר – ואלו אילוצי השוויון השבורים שלנו.

כל הקבוצות $S_v[b]$ ביחד מהוות את הקבוצה S, ואנחנו יכולים לחסום את מספר האילוצים שצמתים ב- S שוברים על ידי סכום מספר האילוצים שכל אחת מהקבוצות האלה שוברת:

$$\sum_{v\in V}\sum_{b\in\Sigma} h\cdot|S_v[b]| = h\cdot|S|$$

אנחנו צריכים לחלק את הערך המתקבל ב- 2, כיוון שאם הקבוצה $S_v[b]$ שוברת אילוץ עם הקבוצה $S_v[c]$ באותו הענן, הסכום שלנו סופר את האילוץ הזה פעמיים. אחרי החלוקה, אנחנו מחשבים את אי השוויון הבא עבור מקרה ב, כלומר $|S|\ge\frac{|F|}{2}$:

$$Gap(G_1)\ge\frac{1}{2}\cdot\frac{(h\cdot|S|)}{|E_1|}\ge h\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{|F|}{(d+1)|E|}\ge\frac{(d-\lambda)}{2}\cdot\frac{1}{4}\cdot\frac{|F|}{(d+1)|E|}=\frac{d-\lambda}{8(d+1)}Gap(G)$$

אז קיבלנו את החסם β שלנו עבור הערך המינימלי משני המקרים:

$$\beta=\min\left(\frac{1}{2(d+1)},\frac{d-\lambda}{8(d+1)}\right)$$

האלגוריתם $prep_2$:

אין כאן שום תחכום – יש לנו את הגרף G_1. אנחנו לוקחים את קבוצת הצמתים שלו, בונים עליה גרף מרחיב d_2-רגולרי בשם $Expander$ עם ע"ע שני $\lambda_2<d_2$ ומדביקים את הקשתות שנוצרו בחזרה על G_1. על כל הקשתות של $Expander$ שנוספו אנחנו מגדירים אילוצים ריקים, שמסופקים בכל השמה. לכל צומת אנחנו מוסיפים קשת עצמית עם אילוץ ריק. לגרף המתקבל נקרא G_2.

הגרף G_1 היה $(d+1)$-רגולרי, ולכן G_2 הוא $(d+1+d_2+2)$-רגולרי.

לכל צומת יש קשת עצמית.

אם G_1 ספיק גם G_2 ספיק, כי לא נוספו בו אילוצים שהשמה כלשהי יכולה לשבור.

הגודל של $Expander$ חסום בקבוע שתלוי ב- d_2 ובמספר הצמתים ב- G_1, ולכן G_2 גדל באופן חסום לינארית.

ה- GAP יכול לרדת רק באופן לינארי כיוון שהוספנו מספר לינארי של קשתות (ואילוצים).

כדי לחשב את הערך העצמי השני של G_2 אנחנו משתמשים בנוסחת ריילי[22]:

$$\lambda=\max_{x\in\mathbb{R}^n,x\cdot\vec{1}=0,x\neq0}\frac{|\langle Ax,x\rangle|}{\langle x,x\rangle}$$

עבורנו, A תהיה המטריצה המייצגת של G_2. היא מורכבת מסכום של המטריצות הבאות:

- המטריצה המייצגת של G_1, בעלת ע"ע שני λ_1 כלשהו.
- המטריצה המייצגת של הגרף המרחיב שהשרינו על G_1, עם ע"ע שני λ_2.
- המטריצה $2I$ של הלולאות העצמיות.

אנחנו בוחרים את ה- x שממקסם את נוסחת ריילי עבור G_2, ומשתמשים בלינאריות המכפלה הפנימית:

$$\frac{|\langle G_2(x),x\rangle|}{\langle x,x\rangle}=\frac{|\langle G_1(x),x\rangle|}{\langle x,x\rangle}+\frac{|\langle Expander(x),x\rangle|}{\langle x,x\rangle}+\frac{|\langle 2I(x),x\rangle|}{\langle x,x\rangle}$$

[22] הזכרנו אותה בקצרה בחלק ג – גם הפעם לא נתעמק בה יותר מדי.

מנוסחת ריילי אנחנו מקבלים:

$$
\begin{aligned}
\lambda(G_2) &= \frac{|\langle G_2(x), x\rangle|}{\langle x, x\rangle} \\
&= \frac{|\langle G_1(x), x\rangle|}{\langle x, x\rangle} + \frac{|\langle Expander(x), x\rangle|}{\langle x, x\rangle} + \frac{|\langle 2I(x), x\rangle|}{\langle x, x\rangle} \\
&\le \lambda_1 + \lambda_2 + \lambda(2I)
\end{aligned}
$$

כאשר אי השוויון בשורה האחרונה נובע מכך שבחרנו x ממקסם עבור G_2, ולא אף מטריצה אחרת. כיוון שהע"ע השני בערך מוחלט של גרף רגולרי קטן (או שווה, במקרה של $2I$) מדרגת הרגולריות, בסך הכל נקבל:

$$\lambda(G_2) \le \lambda_1 + \lambda_2 + \lambda(2I) < (d + 1 + d_2 + 2)$$

ואנחנו מקבלים חסם קבוע על מידת ההרחבה של G_2.

החישוב שדורש את נוסחת ריילי נכנס פה רק כדי שנוכל להמשיך לדבר במונחים של ערכים עצמיים – אם חושבים על זה, האלגוריתם $prep_2$ הוא:

"לוקחים כל תת קבוצה קטנה מחצי של צמתים בגרף ומוסיפים לה קשתות יוצאות כדי להגביר את ההרחבה".

כשאנחנו מרכיבים את שני האלגוריתמים $prep_2$ ו- $prep_1$, אנחנו מקבלים את הרדוקציה שהבטחנו בלמת העיבוד המקדים, ובזה הוכחנו אותה. הפרמטרים הלינאריים שלה (ירידת ה- *GAP* וגדילת הייצוג) הופכים למכפלות של הפרמטרים המתאימים בכל אחד מהאלגוריתמים, והתוצאה המתקבלת מוכנה להעלאה בחזקה והגברת *GAP*.

למת ההגברה – הוכחה:

למת ההגברה טוענת:

> יהיו γ, d קבועים, $\lambda < d$, ו- Σ א"ב אילוצים קבוע. קיים קבוע $\beta_2 > 0$ שתלוי רק ב- $|\Sigma|$, λ, d, כך שלכל גרף אילוצים $G = \left((V,E),\Sigma,C\right)$ d-רגולרי, עם ערך עצמי שני $\lambda(G) \le \lambda$, וקשתות עצמיות בכל צומת, נקבל לאחר העלאת הגרף בחזקת t:
>
> $$Gap(G^t) \ge \beta_2\sqrt{t} \cdot \min\left(\frac{1}{t}\ , Gap(G)\right)$$

הנה הטכניקה הסטנדרטית[23] שלנו כשאנחנו רוצים לחסום מלמטה את שינוי ה- *GAP* של פעולה על גרפים: אנחנו מסתכלים על ההשמה האופטימלית עבור הגרף הנוצר, וגוזרים ממנה השמה לגרף המקור, בצורה שתאפשר לנו לחסום את ה- *GAP* הנוצר מה- *GAP* שכבר היה. כשהוכחנו את למת הקומפוזיציה, השגנו ערך למשתנה לפי מרחק מינימלי ממילת קוד תקינה. כשבנינו את האלגוריתם $prep_1$, הסתכלנו על השמת 'רוב הענן קובע'. עכשיו נסתכל על השמת 'מה השכנים שלי חושבים'.

כדי להוכיח את הלמה, אנחנו עוברים לעולם המושגים של חישובי הסתברות. כזכור, העלאת גרף בחזקה היא בעצם יצירת גרף חדש, שבו הקשתות הן מסלולים באורך t. כל קשת כזו יוצרת 'תת-בעיה' בתוך הבעיה הכוללת שהגרף מייצג, שמורכבת מאותם האילוצים שנמצאים על צמתים במרחק $t/2$ מתחילת וסוף המסלול. (טכנית, הצמתים הם במרחק $[t/2]$, אבל כדי למנוע סיבוך מיותר של הסימונים, אנחנו נניח ש- t הוא זוגי).

בגרף G^t כל צומת v צריך לתת ערך לכל הצמתים במרחק $t/2$ ממנו. באופן סימטרי, כל הצמתים האלה צריכים לתת ערך בחזרה ל- v, היות שהוא במרחק $t/2$ מהם. אנחנו רוצים להתבונן בהשמה הדומה ביותר ל- OPT של G^t, ולנסות להשתמש בדמיון הזה כדי לחסום את ה- *GAP* בגרף החזקה. את הדמיון נקבל כך: אם $\sigma(v) = a$, אז a הוא הערך שממקסם את ההסתברות לצאת למסלול אקראי באורך $t/2$ מ- v ולהגיע לצומת שנותן ל- v את הערך a:

> יהי $G = \left((V,E),\Sigma,C\right)$ גרף אילוצים. לאחר העלאת G בחזקת t נקבל את $G^t = \left((V,E^t)\ ,\Sigma^{d^{t/2}}, C^t\right)$, ואת ההשמה האופטימלית עבורו OPT. ממנה מחלצים ממנה את ההשמה σ עבור הגרף G כך:
>
> $$\sigma(v) = \max\arg_{a\in\Sigma}\ P(A\ random\ t/2\ \ length\ walk\ from\ v\ reaches\ a\ vertex\ w\ with\ OPT(w)[v] = a)$$
>
> כאשר הסימון $OPT(w)[v]$ משמעו הערך ש- w נותן ל- v לפי OPT.

כיוון ש- σ היא השמה לגרף G, היא מפרה לפחות $Gap(G)$ מהאילוצים בו. אנחנו משתמשים בה כדי לחסום את מספר האילוצים ש- OPT מפרה ב- G^t. לשם כך אנחנו מגדירים בתור F את קבוצת הקשתות ש- σ מפרה. אם $\frac{|F|}{|E|} > \frac{1}{t}$, אנחנו מוותרים על חלק מהקשתות בה כדי לקבל מספר שקטן מנקודת הרוויה שלנו.

קשתות ב- E^t הן מסלולים באורך t בגרף G. אנחנו נרצה להתייחס אליהן ככאלה, כלומר מבחינתנו אפשר לסמן את הקשת $e^t = (u,v) \in E^t$ בתור $e^t = (u = v_0, v_1,\cdots, v_t = v)$. עם הסימון החדש אנחנו מגדירים **מסלול שנפגע של ידי הקשת ה- i שלו** אם:

א. $(v_{i-1}, v_i) \in F$

ב. $OPT(v_0)[v_{i-1}] = \sigma(v_{i-1})$ וגם $OPT(v_t)[v_i] = \sigma(v_i)$

במקרה הזה, ברור שהאילוץ המוטל על הקשת $(v_0,\cdots,v_t)$ מופר, כי הצמתים v_0, v_t נותנים לצמתים v_{i-1}, v_i ערכים שמפרים את האילוץ שמוטל עליהם ב- G.

[23] כמה נחמד שכבר יש לנו טכניקה סטנדרטית :)

מתכונת הרגולריות של הגרף אנחנו יכולים להסיק חוסר-תלות בין מספר המסלולים שעובר דרך קשת, לבין הקשת. אנחנו מגדירים אלגוריתם היפותטי לבחירת מסלול, שתלוי בקשת מסוימת, ומראים שהאלגוריתם הוא פשוט תיאור אחר לבחירה אחידה מעל המסלולים ב- G.

טענה: יהי G גרף d-רגולרי, ויהי R אלגוריתם הבחירה הבא למסלול אקראי באורך t ב- G, לכל $0 \le i \le t$:

1. בוחרים קשת רנדומלית (u,v) ב- G.
2. בוחרים מ- u מסלול אקראי באורך $i-1$: $(u,u_{i-2},u_{i-3},\cdots,u_1,u_0)$
3. בוחרים מ- v מסלול אקראי באורך $t-i$: $(v,v_{i+1},v_{i+2},\cdots,v_{t-1},v_t)$
4. מחזירים את המסלול: $(u_0,u_1,\cdots,u_{i-2},u,v,v_{i+1},\cdots,v_{t-1},v_t)$

המסלול ש- R מחזיר שקול לבחירה אחידה מעל המסלולים באורך t.[24]

הוכחה: אין פה הרבה מה להוכיח, יש $|E| \cdot d^{t-1}$ מסלולים כאלה בגרף, ובדיוק $\frac{1}{|E|}$ מהם כוללים את (u,v) בתור הקשת ה- i. על אלה בחרנו באופן אחיד אחת מהאפשרויות לתחילת המסלול, אחת מהאפשרויות לסיום, ובסוף החזרנו את שרשור הבחירות.

עם הטענה האחרונה, אנחנו מסוגלים לחשב באופן לא תלוי את ההסתברות לבחור באופן אחיד קשת ב- G, ואת ההסתברות שאם היא שבורה לפי σ, היא תפגע במסלול כלשהו שעובר דרכה. בלעדיה, היה קשה לחשב באופן אחיד עבור המסלולים ב- G^t את ההסתברות להיפגע.

צריך לשים לב שמסלול כלשהו יכול להיפגע על ידי צעד i רק אם צמתי הקצה v_0,v_t יכולים לתת ערכים לקשת ה- i במסלול. צומת יכול לתת ערכים רק עד מרחק $t/2$, ולכן אנחנו מסיקים שהקשת הפוגעת צריכה להיות בערך באמצע המסלול, כשה- 'בערך' נעשה פחות ופחות מדויק ככל שהמסלול כולל יותר קשתות עצמיות. אז אנחנו מגדירים את I בתור קבוצת מספרי הצעדים הקרובה לאמצע המסלול:

$$I = \left\{ i \in \mathbb{N} : t/2 - \sqrt{t/2} < i \le t/2 + \sqrt{t/2} \right\}$$

השאלה שמייד קופצת היא: למה בעצם רצינו ששכונה של צומת תהיה עד הטווח $t/2$? מה אנחנו מרוויחים עם הטווח הקצר יותר שלא היה לנו אם היינו מגדירים את הטווח להיות שווה לאורך הקשתות, כלומר t?

אז קודם כל – הבטחנו קשתות עצמיות בכל צומת, ולכן מסלולים באורך $t/2$ הם גם מסלולים באורך t אם נתאמץ מספיק – בעיקרון, היינו יכולים להגדיר שכונה גדולה יותר. *אבל, המשמעות האמיתית של השכונות בגרף G^t היא ההשמה σ*. אנחנו רוצים להסיק השמה עבור הגרף G שתעזור לנו לשבור את הגרף G^t, ולערך $t/2$ יש יתרון בגלל הלמה הבאה:

למה על התפלגויות בינומיות דומות:

התפלגות בינומית מודדת לנו את ההסתברות לקבל בדיוק k הצלחות בניסוי שמבצעים n פעמים, שבו הסתברות p להצלחה והסתברות $1-p$ לכשלון, עם הנוסחה הבאה:

$$P\left[B_{n,p} = k\right] = \binom{n}{k} p^k (1-p)^{n-k}$$

הלמה שלנו קובעת –

לכל $p \in (0,1)$ ו- $c > 0$, קיימים l_0 ו- $0 < \tau < 1$ כך שלכל l_1, l_2 שקרובים מספיק וגדולים מספיק כדי לקיים את אי השוויון:

$$l_0 < l_1 - \sqrt{l_1} \le l_2 \le l_1 + \sqrt{l_1}$$

[24] האלגוריתם R אינו חלק מהרדוקציה שלנו – אנחנו משתמשים בו בשביל ניתוח הסתברותי, כי זו פשוט דרך נוחה לומר שבגרף רגולרי אין תלות בין הקשת במקום i לבין המסלול שכולל אותה.

יש לנו טווח K של ערכי k:

$$K = \left\{k : |k - p \cdot l_1| \le c\sqrt{l_1}\right\}$$

עבורם מקבלים:

$$\tau \le \frac{P\left[B_{l_1,p} = k\right]}{P\left[B_{l_2,p} = k\right]} \le \frac{1}{\tau}$$

אנחנו לא נוכיח את הלמה הזו, אבל אנחנו מאוד רוצים לדעת מה היא אומרת, ולמה היא נכונה. קודם כל, אנחנו רוצים להבין את אי השוויון האחרון, החסם שהלמה נותנת:

$$\tau \le \frac{P\left[B_{l_1,p} = k\right]}{P\left[B_{l_2,p} = k\right]} \le \frac{1}{\tau}$$

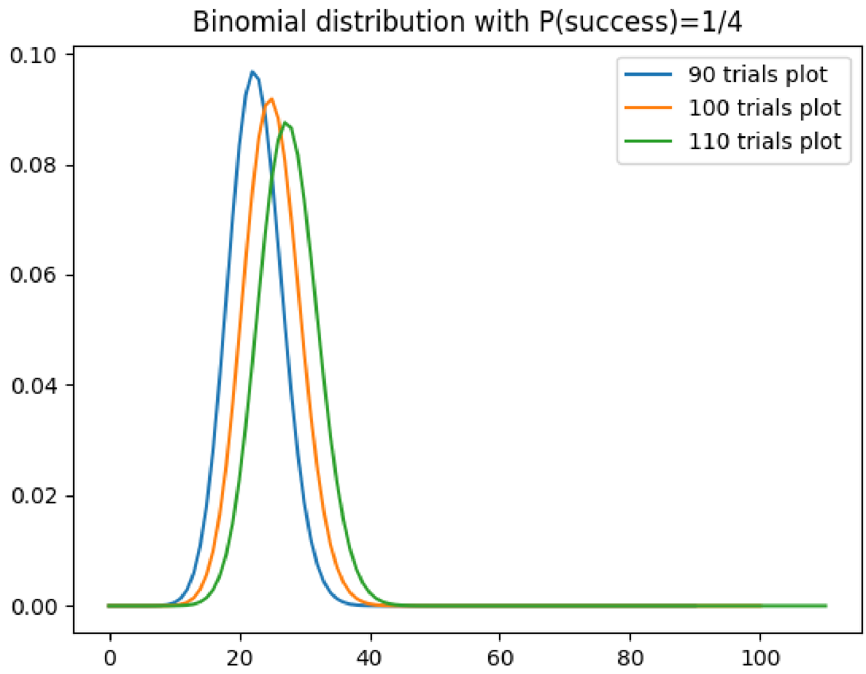
הוא אומר לנו, שאם נבצע את הניסוי l_1 או l_2 פעמים, תחת האילוצים המתאימים נקבל התפלגויות דומות עבור מספר ההצלחות – יש לנו חסם על היחס בין ההסתברות לקבל k הצלחות לאחר l_1 או l_2 פעמים, לפי הפרמטר τ שתלוי רק ב- p וב- c.

מה הם האילוצים האלה? הראשון הוא, שהערך k שאנחנו מכוונים אליו נמצא קרוב לתוחלת של ההתפלגות הבינומית לפי l_1:

$$K = \left\{k : |k - p \cdot l_1| \le c\sqrt{l_1}\right\}$$

תוחלת מספר ההצלחות לפי הסתברות p לאחר l_1 ניסויים היא כמובן $p \cdot l_1$, ואנחנו אומרים שאי-השוויון תקף לכל k שמרחקו מהתוחלת קטן מ- $c\sqrt{l_1}$, כלומר מרחק לינארי בשורש מספר הניסויים.

האילוץ השני הוא שמספר הניסויים l_2 קרוב למספר הניסויים l_1, גם כן לפי הפרמטר $\sqrt{l_1}$, וששני המספרים גדולים מספיק כדי למנוע התפלגויות עם מספר קטן של ניסויים, שעלולות למנוע את הופעת התכונה שהלמה מתארת – ריכוז פונקציית ההסתברות של משתנה בינומי באזור התוחלת. כדי להבין מה היא התכונה הזו, יש לנו תרשימים של התפלגות בינומית עבור סיכויי הצלחה של ¼ לאחר 100 ניסויים, ולאחר 90 ו- 110 ניסויים ($\pm\sqrt{100}$):



החפיפה שאנחנו רואים באזור התוחלות של כל אחת מההתפלגויות היא אותה תכונת ריכוז – היא זו שאומרת לנו שבש- $P\left[B_{l_1,p}=k\right]$ גדול, $P\left[B_{l_2,p}=k\right]$ לא יכול להיות קטן מדי, ולכן היחס ביניהם ניתן לחסימה. כמובן, ככל שאנחנו מגדילים את c, כלומר את הטווח של קבוצת הערכים K, τ ייאלץ להיות קטן יותר על מנת לאפשר יחסים גבוהים יותר, אבל כמו שכבר הדגמנו הרבה פעמים – ברגע שיש לנו איזשהו חסם קבוע, לא ממש אכפת לנו כמה הוא טוב.

מבחינתנו הלמה הזו מסמנת את הדמיון שבין מסלולים אקראיים באורך $t/2$ למסלולים באורך אחר, אבל קרוב. כיוון שיש לנו בכל צעד הסתברות של $1/d$ לעבור דרך קשת עצמית ובאיזשהו אופן לקצר את המסלול שלנו, אנחנו מקבלים שעבור t מספיק גדול, מסלולים בטווח של $\left[\frac{t}{2}-\sqrt{\frac{t}{2}},\ \frac{t}{2}+\sqrt{\frac{t}{2}}\right]$ הם פחות או יותר מסלולים באותו אורך. הרשינו צמתים עם יותר מקשת עצמית אחת בגרף G, אז בשביל התפלגות בינומית אמיתית אנחנו בוחרים קשת עצמית יחידה בכל צומת שהיא **ה**קשת העצמית שלו, ואליה אנחנו מתייחסים כשאנחנו מדברים על צעד עצמי.

פלאשבק אחורה – שאלנו למה פעולת ההעלאה בחזקה שהגדרנו מקצרת את רדיוס השכונה של משתנה לחצי מאורך קשת. אנחנו זוכרים שהערך $\sigma(v)$ נקבע על ידי המשתנים בקצה השכונה של v (צמתים במרחק $t/2$). עם הלמה שראינו עכשיו, נוכל להתייחס לכל הצמתים באמצע המסלול בתור צמתים שבהסתברות גבוהה, לקחו חלק בקביעת ההשמה σ שלנו עבור הגרף G. זה מאפשר לנו להתייחס לקשת שבורה באזור אמצע המסלול, ולחסום מלמטה את ההסתברות שהיא פוגעת, כלומר שהצמתים v_0, v_t בקצות המסלול מפרים ביניהם את האילוץ המוטל עליה.

בחזרה לאיפה שהיינו – הגדרנו את I בתור "קבוצת האמצע":

$$I=\left\{i\in\mathbb{N}:t/2-\sqrt{t/2}<i\leq t/2+\sqrt{t/2}\right\}$$

מעליה מגדירים את המשתנה האקראי $N(e^t)$ שסופר עבור בחירה אקראית ואחידה של מסלול ב- E^t כמה פעמים הוא נפגע בתחום I שלו:

$$N(e^t)=\left|\left\{i\in I: e\ is\ hit\ by\ its\ i^{th}\ edge\right\}\right|$$

הרעיון הוא, שקשת עבורה $N(e^t) > 0$, דוחה את ההשמה OPT. אנחנו משתמשים בשיטת המומנט השני כדי לחסום מלמטה את ההסתברות ש- $N(e^t) > 0$.

שיטת המומנט השני:

זוהי שיטה קלאסית שבה משתמשים כדי להראות שמשתנה אקראי עם תוחלת גבוהה, הוא חיובי בהסתברות גבוהה. אנחנו רוצים לוודא שהשונות של המשתנה אינה גבוהה מדי – למשל, אם X הוא משתנה אקראי במרחב הבא:

$$P(X=0)=\frac{99}{100},\qquad P(X=10^9)=\frac{1}{100}$$

נקבל תוחלת גבוהה, אבל אנחנו לא יכולים להסיק מכך שההסתברות לקבל ערך חיובי היא גבוהה.

השיטה היא בעצם החסם הבא על היחס בין ריבוע התוחלת לבין התוחלת של הריבוע:

$$P(X>0)\geq\frac{\mathbb{E}^2[X]}{\mathbb{E}[X^2]}$$

שאותו אפשר להסיק די בקלות אם מתייחסים לתוחלת בתור אופרטור מכפלה פנימית במרחב הסתברות – מבחינתנו זה פשוט נכון.

אז אנחנו משתמשים באי השוויון הבא:

$$P(N(e)>0)\geq\frac{\mathbb{E}^2[N(e^t)]}{\mathbb{E}[N^2(e^t)]}\geq\Omega(\sqrt{\mathrm{t}})\cdot\frac{|F|}{|E|}$$

כדי להראות שההסתברות לכך ש- $N(e^t)$ חיובי גבוהה מפאקטור של $\sqrt{t}$ כפול $Gap(G)$, ולכן (כיוון ש- $N(e^t)$ מקבל ערך לפי בחירה אחידה של מסלולים) – $Gap(G^t)$ גדול כפי שרצינו.

לשם כך אנחנו צריכים להראות:

$$\mathbb{E}[N(e^t)]\geq\Omega(\sqrt{t})\frac{|F|}{|E|}$$

$$\mathbb{E}[N^2(e^t)]\leq\mathrm{O}(\sqrt{t})\frac{|F|}{|E|}$$

ולקבל:

$$P(N(e)>0)\geq\frac{\mathbb{E}^2[N(e^t)]}{\mathbb{E}[N^2(e^t)]}\geq\frac{\Omega(t)\left(\frac{|F|}{|E|}\right)^2}{\mathrm{O}(\sqrt{t})\frac{|F|}{|E|}}\geq\Omega(\sqrt{\mathrm{t}})\cdot\frac{|F|}{|E|}\geq\Omega(\sqrt{t})\min\left(\frac{1}{t}\ ,Gap(G)\right)$$

כדי להעריך את $\mathbb{E}[N(e^t)]$, אנחנו מגדירים את האינדיקטור $N_i(e^t)$ שמשמעו "מה ההסתברות שהמסלול ש- $N(e^t)$ מקבל נפגע על ידי הצעד ה- i שלו". כמובן, $N(e^t)=\sum_{i\in I}N_i(e^t)$, ואנחנו יכולים להשתמש בלינאריות של התוחלת כדי לקבל:

$$\mathbb{E}[N(e^t)]=\sum_{i\in I}\mathbb{E}[N_i(e^t)]$$

את $\mathbb{E}[N_i(e^t)]$ נעריך בעזרת האלגוריתם R שלנו, שמחזיר לנו מסלול באורך t בהתפלגות אחידה, *ומבטיח לנו שקצות המסלול תלויים אך ורק בקשת ה- i, שנבחרה באופן אחיד מעל E.*

עבור מסלול e^t שחוזר מ- R, עבור $i \in I$ כלשהו, אנחנו מקבלים:

$$P(N_i(e^t) = 1) = \frac{|F|}{|E|} \cdot p_{v_0} \cdot p_{v_t}$$

כאשר $\frac{|F|}{|E|}$ זו פשוט ההסתברות שהקשת הראשונה שנבחרה על ידי האלגוריתם מקבלת השמה שבורה מ- σ, והרכיבים p_{v_0}, p_{v_t} הם ההסתברות שהצומת הראשון והאחרון במסלול קיבלו מ- OPT השמה שמסכימה עם σ לגבי הצמתים v_i, v_{i-1}.

אנחנו מחשבים את p_{v_t} (המקרה של p_{v_0} סימטרי כי כמו שאמרנו, צמתי הקצה תלויים אך ורק בבחירת הקשת הראשונה).

לפי הגדרת σ, אם אורך המסלול p_{v_i} ל- p_{v_t} הוא בדיוק $t/2$, ההסתברות ש- $OPT(p_{v_t})[p_{v_i}] = \sigma(p_{v_i})$ היא לפחות $\frac{1}{|\Sigma|}$.

אנחנו מגדירים מחדש מסלול אקראי מצומת לצומת בצורה הבאה: במקום לבחור בכל צעד אחת מ- d הקשתות, אנחנו בוחרים מספר l כלשהו של צעדים שבהם נצעד בקשת עצמית, ואת המיקומים שלהם לאורך המסלול. את שאר הצעדים אנחנו בוחרים בתנאי שהם לא כוללים קשת עצמית. זו הגדרה שקולה למסלול אקראי, אבל כדי שיהיה יותר נוח אנחנו נבחר את כל הצעדים שבהם צועדים בקשת *לא* עצמית – זו כמובן, עוד דרך שקולה להגדרת מסלול.

עכשיו הגיע הזמן להגדיר כמה משתנים אקראיים, מהסוג שאפשר לכתוב בנוסחאות:

המשתנה $X_{u,l}$ מקבל ערך $a \in \Sigma$ באותה ההסתברות שמסלול אקראי באורך l מ- u מגיע לצומת w עבורה $OPT(w)[u] = a$. במונחים אלה הערך p_{v_t} מנוסחת ההסתברות של האינדיקטור הוא:

$$p_{v_t} = P\left[X_{v_i,t-i} = \sigma(v_t)\right]$$

והחסם שמקבלים אם הקשת i היא בדיוק אמצע המסלול הוא:

$$P\left[X_{u,t/2} = \sigma(u)\right] \geq \frac{1}{|\Sigma|}$$

המשתנה $X'_{u,l}$ דומה ל- $X_{u,l}$ אבל ההסתברות שהוא מחשב היא של מסלול ללא לולאות עצמיות. אותו אנחנו יכולים לשלב עם התפלגות בינומית לספירת הצעדים במסלול באורך l שבהם *לא* צועדים בלולאה כך:

$$P\left[X_{u,l} = a\right] = P\left[X'_{u,k} = a\right] \cdot P\left[B_{l,\frac{d-1}{d}} = k\right]$$

כלומר ההסתברות להגיע לצומת w כלשהו, היא בדיוק ההסתברות לבחור k צעדים מתוך ה- l שאינם לולאה, ובהם לעשות את המסלול מ-u ל- w במסלול חסר לולאות.

עכשיו אפשר להשתמש במה שאנחנו יודעים על התפלגויות בינומיות דומות:

עבור $p = \frac{d-1}{d}$, אנחנו יכולים לבחור $c > 0$, כך שלכל l שקרוב מספיק ל- $t/2$ (בהנחה ש- $t/2$ גדול מספיק):

$$\frac{t}{2} - \sqrt{\frac{t}{2}} \leq \ l \ \leq \frac{t}{2} + \sqrt{\frac{t}{2}}$$

יש לנו טווח K של ערכי k באזור $\frac{(d-1)\left(\frac{t}{2}\right)}{d}$:

$$K = \left\{ k : \left| k - \frac{(d-1)(t/2)}{d} \right| \le c\sqrt{t/2} \right\}$$

עבורם קיים ערך τ שמקיים:

$$\tau \le \frac{P[B_{t/2,p} = k]}{P[B_{l,p} = k]} \le \frac{1}{\tau}$$

אנחנו בוחרים c גדול מספיק, כדי שההסתברות לקבל מההתפלגות הבינומית לבחירת מעגלים ערך שאינו ב- K עבור $t/2$ ניסויים, קטנה מ- $\frac{1}{2|\Sigma|}$.[25] בזה נשתמש כדי בשביל סדרת המעברים הבאה:

יש לנו 2 הגדרות שקולות למסלול באורך l שאחת מהן כוללת בחירה של צעדים בהם מבצעים צעד עצמי. מספר האופציות לכמות הצעדים העצמיים גדולה מ- $|K|$, אז מתקבל אי השוויון:

$$P\left(X_{v_i,l} = \sigma(v_i)\right) \ge \sum_{k\in K} P\left[B_{l,\frac{d-1}{d}} = k\right] \cdot P\left(X'_{v_i,k} = \sigma(v_i)\right)$$

אנחנו משתמשים בהתפלגויות דומות כדי לקבל סכום על מסלולים באורך $t/2$, עבורם ידוע שההסתברות $OPT(v_t)[v_i] = \sigma(v_i)$ גדולה מ- $\frac{1}{|\Sigma|}$:

$$P\left(X_{v_i,l} = \sigma(v_i)\right) \ge \tau \sum_{k\in K} P\left[B_{t/2,\frac{d-1}{d}} = k\right] \cdot P\left(X'_{v_i,k} = \sigma(v_i)\right)$$

אם סכום ההסתברויות היה מ- $i = 1$ עד $t/2$ היינו מקבלים בדיוק את ההסתברות להגיע ל- $\sigma(v_i)$ לאחר מסלול באורך $t/2$. הסכום שלנו כולל פחות אינדקסים, אבל בחרנו את c כך שיהיו מספיק מהם כדי לקבל:

$$P\left(X_{v_i,l} = \sigma(v_i)\right) \ge \tau \cdot \left(P\left(X_{v_i,\frac{t}{2}} = \sigma(v_i)\right) - \frac{1}{2|\Sigma|}\right) \ge \tau \cdot \left(\frac{1}{|\Sigma|} - \frac{1}{2|\Sigma|}\right) = \frac{\tau}{2|\Sigma|} = Constant!$$

עכשיו אנחנו יכולים לחזור למשוואות:

$$\mathbb{E}[N(e^t)] = \sum_{i\in I} \mathbb{E}[N_i(e^t)]$$

$$P(N_i(e^t) = 1) = \frac{|F|}{|E|} \cdot p_{v_0} \cdot p_{v_t}$$

הצלחנו לחסום את $p_{v_0} \cdot p_{v_t}$ מלמטה עם הערך $\left(\frac{\tau}{2|\Sigma|}\right)^2$. מכאן אנחנו מסיקים:

$$\mathbb{E}[N(e^t)] = \sum_{i\in I} \mathbb{E}[N_i(e^t)] \ge |I| \frac{|F|}{|E|} \left(\frac{\tau}{2|\Sigma|}\right)^2 = \Omega(\sqrt{t}) \frac{|F|}{|E|} \left(\frac{\tau}{2|\Sigma|}\right)^2 = \Omega(\sqrt{t}) \frac{|F|}{|E|}$$

[25] כיוון שערכי ה- k שאנחנו מכוונים אליהם נמצאים באזור התוחלת, ההסתברות יכולה להיות נמוכה גם עבור טווח קטן יחסית של ערכים.

כדי להשלים את השימוש בשיטת המומנט השני, אנחנו צריכים להראות:

$$\mathbb{E}[N^2(e^t)] \leq O(\sqrt{t})\frac{|F|}{|E|}$$

במילים אחרות, אנחנו רוצים להראות שהתוחלת שחישבנו בשלב הקודם גבוהה – לא בגלל נוכחות מספר מצומצם של מסלולים שנפגעים מהרבה קשתות (מה שיגרום לערך הריבועי של המשתנה לטפס מעל ריבוע התוחלת), אלא פשוט בגלל שההסתברות להיפגע גבוהה. הגרף שלנו מרחיב, ולכן אנחנו יודעים שהקשתות הפוגעות שלנו לא יכולות להיות מרוכזות באזור כלשהו של הגרף, אבל עדיין חסרה דרך פורמלית לתאר את הטענה הזו. לא הגענו בידיים ריקות, את רוב העבודה כבר עשינו – אנחנו צריכים להיזכר בטענה הבאה שהוכחנו כשדיברנו על הרחבה בגרפים:

> יהי $G=(V,E)$ גרף $(n,d,\lambda)-expander$, ותהי $F\subset E$ תת קבוצה של קשתות. עבור מסלול אקראי באורך t, שהצעד הראשון שלו הוא קשת אקראית ב- F, ההסתברות שהצעד האחרון גם כן נמצא ב- F (הקשתות הראשונה והאחרונה ב- F), היא לכל היותר:
>
> $$\frac{|F|}{|E|}+\left(\frac{\lambda}{d}\right)^{t-1}$$

אנחנו נחסום את $\mathbb{E}[N^2(e^t)]$ בעזרת משתנה אקראי חדש, $Z(e^t)$. המשתנה החדש סופר את מספר הפעמים שהמסלול e^t עובר ב- F כשהוא צועד בקבוצת האמצע I. שלא כמו בספירה של $N(e^t)$, הקשת לא צריכה לפגוע (כלומר הוא סופר קשתות שבורות לפי σ ב- G, בלי להתייחס בכלל ל- OPT). כל קשת ש- $N(e^t)$ סופר גם $Z(e^t)$ סופר, אז מייד מתקבל אי השוויון:

$$N(e^t)\leq Z(e^t)$$

ולכן המשימה החדשה לנו היא לחסום את $Z(e^t)$ ב- $O(\sqrt{t})\frac{|F|}{|E|}$. שוב אנחנו משתמשים באינדיקטורים:

$$Z(e^t)=\sum_{i\in I}Z_i(e^t)$$

ובלינאריות של התוחלת כדי לקבל:

$$\mathbb{E}[Z^2(e^t)]=\sum_{i,j\in I}\mathbb{E}\left[Z_i(e^t)\cdot Z_j(e^t)\right]=\sum_{i\in I}\mathbb{E}[Z_i(e^t)]+2\sum_{\substack{i,j\in I\\ i<j}}\mathbb{E}\left[Z_i(e^t)\cdot Z_j(e^t)\right]$$

קודם כל, אנחנו רוצים לחשב את המחובר $\sum_{i\in I}\mathbb{E}[Z_i(e^t)]$ – בעיקר, אנחנו מעוניינים להראות שבחירת הקשת ה- i במסלול אקראי שקולה לבחירה אחידה מעל הקשתות e ב- G. מהאלגוריתם R שלנו, זה נובע באופן מידי. יש בדיוק $\frac{1}{|E|}$ מהמסלולים שבהם קשת מסוימת היא הקשת ה- i. אנחנו יכולים לבחור בעזרת R מסלול באופן אחיד, אבל R בעצמו בוחר באופן אחיד מעל E את הקשת ה- i במסלול – אז בחירה של קשת במיקום ספציפי של מסלול שנבחר באופן אחיד, שקולה לבחירה אחידה של קשתות. אם כך, $P(Z_i(e^t)=1)=\frac{|F|}{|E|}$, ואנחנו מקבלים:

$$\mathbb{E}[Z^2(e^t)]=|I|\frac{|F|}{|E|}+2\sum_{\substack{i,j\in I\\ i<j}}\mathbb{E}\left[Z_i(e^t)\cdot Z_j(e^t)\right]$$

עכשיו, כדי לחשב את $\sum_{\substack{i,j\in I\\ i<j}} \mathbb{E}\left[Z_i(e^t)\cdot Z_j(e^t)\right]$ אנחנו מחשבים את ההסתברות:

$$\begin{aligned}\mathbb{E}\left[Z_i(e^t)\cdot Z_j(e^t)\right] &= P\left(Z_i(e^t)\cdot Z_j(e^t)=1\right)\\ &= P(Z_i(e^t)=1)\cdot P\left(Z_j(e^t)=1\middle|Z_i(e^t)=1\right)\\ &= \frac{|F|}{|E|}P\left(Z_j(e^t)=1\middle|Z_i(e^t)=1\right)\end{aligned}$$

מה היא ההסתברות $P\left(Z_j(e^t)=1\middle|Z_i(e^t)=1\right)$? זוהי בדיוק ההסתברות שמסלול באורך $j-i$ שמתחיל בצעד i של המסלול e^t, יסתיים ב- F – בתנאי שהצעד ה- i עובר ב- F. כמו שאמרנו, הקשת ה- i בבחירה אחידה של e^t שקולה לבחירה אחידה של קשת ב- E, ולכן אנחנו יכולים להשתמש בטענה שלנו על גרפים מרחיבים – המקטע $(v_{i-1},\cdots,v_j)$ ממסלול e^t אקראי, הוא מסלול אקראי באורך $j-i$ שמתחיל בבחירה אחידה של קשתות, והטענה שלנו על גרפים מרחיבים נותנת:

$$P\left(Z_j(e^t)=1\middle|Z_i(e^t)=1\right)=P\left((v_{j-1},v_j)\in F|(v_{i-1},v_i)\in F\right)\le \frac{|F|}{|E|}+\left(\frac{\lambda}{d}\right)^{j-i-1}$$

אנחנו מתחילים להציב במשוואות שחישבנו ומקבלים:

$$\begin{aligned}\mathbb{E}\left[Z_i(e^t)\cdot Z_j(e^t)\right] &= \frac{|F|}{|E|}P\left(Z_j(e^t)=1\middle|Z_i(e^t)=1\right)\le\frac{|F|}{|E|}\left(\frac{|F|}{|E|}+\left(\frac{\lambda}{d}\right)^{j-i-1}\right)\\ &\Rightarrow\\ \mathbb{E}[Z^2(e^t)] &= \sum_{i\in I}\mathbb{E}[Z_i(e^t)]+2\sum_{\substack{i,j\in I\\ i<j}}\mathbb{E}\left[Z_i(e^t)\cdot Z_j(e^t)\right]\\ &= |I|\frac{|F|}{|E|}+2\sum_{\substack{i,j\in I\\ i<j}}\frac{|F|}{|E|}\left(\frac{|F|}{|E|}+\left(\frac{\lambda}{d}\right)^{j-i-1}\right)\\ &= O\left(\sqrt{t}\right)\frac{|F|}{|E|}+2\sum_{\substack{i,j\in I\\ i<j}}\frac{|F|}{|E|}\left(\frac{|F|}{|E|}+\left(\frac{\lambda}{d}\right)^{j-i-1}\right)\\ &= O\left(\sqrt{t}\right)\frac{|F|}{|E|}+\frac{2|F|}{|E|}\sum_{\substack{i,j\in I\\ i<j}}\left(\frac{|F|}{|E|}+\left(\frac{\lambda}{d}\right)^{j-i-1}\right)\\ &= O\left(\sqrt{t}\right)\frac{|F|}{|E|}+\frac{2|F|}{|E|}\left(|I|^2\frac{|F|}{|E|}+\sum_{\substack{i,j\in I\\ i<j}}\left(\frac{\lambda}{d}\right)^{j-i-1}\right)\\ &= O\left(\sqrt{t}\right)\frac{|F|}{|E|}+\frac{2|F|}{|E|}\left(|I|^2\frac{|F|}{|E|}+\sum_{i=1}^{2\sqrt{t}}\left(\frac{\lambda}{d}\right)^{i}\right)\end{aligned}$$

$\left(\frac{\lambda}{d}\right)$ הוא ערך קטן ממש מ- 1, ולכן הוא חוסם כל סדרה הנדסית שמשתמשת בו כבסיס, ואנחנו מקבלים:

$$O(\sqrt{t})\frac{|F|}{|E|}+\frac{2|F|}{|E|}\left(|I|^2\frac{|F|}{|E|}+\sum_{i=1}^{2\sqrt{t}}\left(\frac{\lambda}{d}\right)^i\right)=O(\sqrt{t})\frac{|F|}{|E|}+\frac{2|F|}{|E|}\left(|I|^2\frac{|F|}{|E|}+O(1)\right)$$
$$=O(\sqrt{t})\frac{|F|}{|E|}+2|I|^2\left(\frac{|F|}{|E|}\right)^2+O\left(\frac{2|F|}{|E|}\right)$$

בנוסף, קבענו[26] את $\frac{|F|}{|E|}$ להיות תמיד קטן או שווה ל- $\frac{1}{t}$, אז:

$$O(\sqrt{t})\frac{|F|}{|E|}+2|I|^2\left(\frac{|F|}{|E|}\right)^2+O\left(\frac{2|F|}{|E|}\right)=O(\sqrt{t})\frac{|F|}{|E|}+O(\sqrt{t})^2\cdot O\left(\frac{1}{t^2}\right)+O\left(\frac{|F|}{|E|}\right)$$
$$=O(\sqrt{t})\frac{|F|}{|E|}$$
$$\geq \mathbb{E}[N^2(e^t)]$$

בכך אנחנו משלימים את השימוש בשיטת המומנט השני, משלימים איתה את הוכחת למת ההגברה ועם הלמה משלימים את משפט ההגברה.

וסוף סוף, אנחנו מוכיחים את משפט PCP:

משפט ההגברה מבטיח רדוקציית הגברה פולינומיאלית שלוקחת מופע של CSP בינארי $G=\big((V,E),\Sigma,C\big)$ מעל א"ב Σ סופי כלשהו, והופך אותה למופע אחר של CSP בינארי $G'=\big((V',E'),\Sigma_0,C'\big)$ עם הפרמטרים הבאים:

א. $size(G')\leq K\cdot size(G)$ – הבעיה כולה גדלה לכל היותר באופן לינארי.
ב. אם הגרף G ספיק גם G' ספיק.
ג. $Gap(G')\geq \min(2Gap(G),\alpha)$ – אי הספיקות של G' הכפילה את עצמה, עד לנקודת רוויה של α.

אנחנו מראים ש- GAP-CSP$_{1,\alpha}$ היא NP קשה על ידי רדוקציה מ- 3COLOR. קודם כל, אנחנו מייצגים את 3COLOR בתור CSP. לאחר מכן אנחנו מבצעים את רדוקציית ההגברה x פעמים, עבור ה- x המינימלי שפותר את אי השוויון:

$$\frac{1}{|E|}\cdot 2^x\geq\alpha$$

כיוון ש- α ניתן לכתיבה בתור חלק קבוע כלשהו מהקשתות ב- G, כלומר $\frac{s}{|E|}$, אנחנו מקבלים:

$$\frac{1}{|E|}\cdot 2^x\geq\frac{s}{|E|}\Longrightarrow 2^x\geq s\Longrightarrow x=O(\lg(|E|))=O\big(\lg\big(size(G)\big)\big)$$

אנחנו יכולים לחסום את גודל הגרף המקסימלי שהרדוקציה נתקלת בו על ידי $size(G)\cdot K^x=poly\big(size(G)\big)$, ולכן הרדוקציה שתיארנו עכשיו פולינומיאלית, כלומר GAP-CSP$_{1,\alpha}$ היא NP קשה. מהשקילות בין התוצאה הזו למשפט PCP, נובע המשפט. ■

[26] זו לא איזו תאונה מבורכת – החסם על תוחלת הריבוע $N^2(e^t)$ הוא חסם פורמלי על הריכוז המקסימלי של קשתות בעייתיות באזור מסוים בגרף האילוצים, והוא יכול לעבוד רק עד נקודת הרוויה של הגרף.

חלק ה – משפט PCP: מה עושים איתו?

הצלחנו! יש לנו את משפט PCP! ואיתו שפע תוצאות קושי-של-קירוב עבור מספר גדול של בעיות NP-שלמות שאנחנו יודעים להכליל לבעיות אילוצים מעל א"ב קבוע, ביניהן:

- בעיות צביעה בגרפים
- בעיות סיפוק נוסחאות CNF
- ...

אבל אם נחזור רגע לניסוח המקורי של המשפט: $NP \subseteq PCP[O(lgn), O(1)]$, כל כך הרבה רעש עשינו סביבו – "אחת התוצאות החשובות ביותר בסיבוכיות... כולם מקבלים פרס גדל..." – עד עכשיו, רק הקביעה שבעיית האילוצים היא NP-קשה לקירוב נתנה לנו משהו, ובכלל לא השתמשנו במשפט PCP כדי להוכיח אותה. אולי כדאי שנראה איזו תוצאה מעניינת של המשפט כמו שהוא נוסח במקור?

קבוצת השפות ה- NP-שלמות מכילה 3 בעיות קומבינטוריות בגרפים שקשורות באופן הדוק אחת לשנייה. בהינתן הגרף $G = (V, E)$ ומספר טבעי k, אנחנו מגדירים את שלושת הבעיות הבאות.

בעיית הקליקה – CLIQUE:

אנחנו רוצים לקבוע האם בגרף יש **קליקה** בגודל k, כלומר k צמתים שכל זוג מהם מחובר בקשת.

בעיית הקבוצה הלא-תלויה – INDEPENDENT-SET:

אנחנו רוצים לקבוע האם בגרף יש קבוצה בלתי-תלויה בגודל k, כלומר k צמתים שביניהם אין אף קשת בגרף.

בעיית כיסוי הצמתים – VERTEX-COVER:

את הבעיה הזו אנחנו כבר מכירים בגרסת האופטימיזציה – מהי הקבוצה המינימלית של צמתים שמכסה את כל הקשתות בגרף? בגרסת ההחלטה אנחנו רוצים לדעת האם יש קבוצה מכסה בגודל k. בחלק א, הראינו לבעיה הזו אלגוריתם קירוב מסדר 2, שמחזיר קבוצה בגודל כפול לכל היותר מהכיסוי המינימלי.

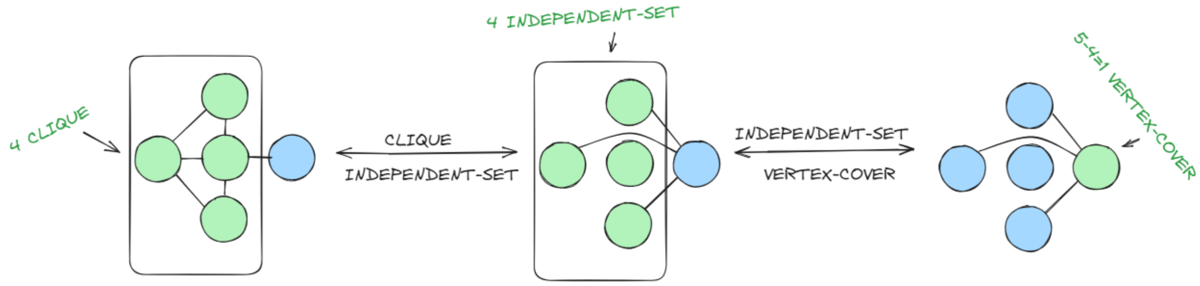
הקשר בין הבעיות עובר דרך פעולת **ההשלמה** של גרף – שבה אנחנו שמים קשת בכל מקום שלא הייתה אחת, ומורידים את כל הקשתות הקיימות. הפעולה הזו נותנת לנו רדוקציות פולינומיאליות פשוטות בין 3 הבעיות.

רדוקציה מ- CLIQUE ל- INDEPENDENT-SET, ובחזרה:

משלימים את הגרף – כל קליקה הופכת לקבוצה בלתי-תלויה, וכל קבוצה בלתי תלויה הופכת לקליקה. אם הייתה קליקה בגודל k, לאחר ההשלמה היא תהפוך לקבוצה בלתי-תלויה, והפוך.

רדוקציה מ- INDEPENDENT-SET ל- VERTEX-COVER, ובחזרה:

כאן לא צריך את ההשלמה – אם S היא קבוצה בלתי-תלויה בגרף, אין אף קשת בין שני צמתים ב- S, ולכן $V \setminus S$ היא כיסוי צמתים – כלומר יש לנו כיסוי צמתים בגודל $|V| - k$.



אם הרדוקציה כל כך פשוטה, בוודאי אפשר לצפות לאלגוריתם קירוב כלשהו עבור בעיית הקליקה שמבוסס על הקירוב המוכר ל- VERTEX-COVER, עם כמה התאמות ושינויים קטנים?

אז נגדיר גרסת *GAP* לבעיית הקליקה – הבעיה GAP-CLIQUE_s מוגדרת כך:

בהינתן גרף ומספר k, אלגוריתם שפותר את GAP-CLIQUE_s צריך להחזיר את התוצאות הבאות:

1. כן, אם בגרף יש קליקה בגודל k.
2. לא, אם אין בגרף אף קליקה בגודל $s \cdot k$ (s הוא פרמטר הקירוב של הבעיה).
3. מה שבא לו, אם יש קליקה שמכילה יותר מ- $s \cdot k$ אבל פחות מ- k צמתים.

הדבר הראשון שנראה, הוא ש- GAP-CLIQUE היא NP קשה. כמובן, GAP-CLIQUE תהיה קשה עבור s מסוים והלאה – הקירוב שלנו עבור VERTEX-COVER טוב רק עד פי 2 מהפתרון האופטימלי, ואנחנו עדיין יכולים לקוות שיצא ממנו קירוב כלשהו עבור בעיית הקליקה, עבור s קטן יחסית.

רדוקציית הגברת ה- *GAP* שביצענו כדי להראות ש- GAP-CSP היא NP-שלמה לא כל כך מתאימה. היא לא נותנת לנו דרך להפוך קליקה בגודל $k-1$ לקליקה קטנה מ- sk, כי CLIQUE אינה בעיית CSP באופן מובן מאליו. למרות זאת, הרדוקציה עבור 3SAT מובילה אותנו לתוצאת קושי-של-קירוב עבור בעיית הקליקה באופן כמעט מידי. כדי לראות איך זה עובד, אנחנו רוצים לבחון את הרדוקציה הקלאסית מ- 3SAT אל קליקה בגרסאות ההחלטה.

רדוקציה מ- 3SAT אל CLIQUE:

הקלט: נוסחת 3CNF עם m פסוקיות – אנחנו רוצים להחליט אם היא ספיקה.

הפלט: גרף $G=(V,E)$, שיש בו קליקה בגודל m אם"ם הנוסחה ספיקה.

הרדוקציה:

עבור כל פסוקית בנוסחה $\phi=(a \vee b \vee c)$:

בנה שורה של 7 צמתים. הצמתים מייצגים את שבעת ההשמות השונות ל- $\{a,b,c\}$ שמספקות את הפסוקית. לפסוקית OR מעל שלושה משתנים יש השמה לא-מספקת יחידה (שלילת כל הליטרלים בפסוקית). למשל, $(\neg x_1, x_2, x_3)$ מסופקת על ידי כל השמה מלבד $(1,0,0)$.

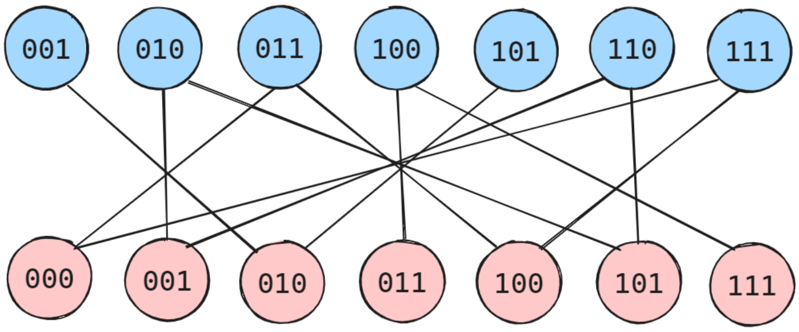
לאחר יצירת כל שורות הצמתים, הגדר קשת בין כל שני צמתים שעונים על התנאים הבאים:

- הצמתים לא נמצאים באותה השורה.
- אין קשת בין זוג צמתים שנותנים השמה שונה לאותו הליטרל.

לדוגמה, הנוסחה (הספיקה) הבאה מעל 2 פסוקיות:

$$(x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee \neg x_2 \vee \neg x_3)$$

תהפוך לגרף הבא מעל 14 צמתים:



בגרף יש 2 שורות, אחת עבור כל פסוקית. שתי הפסוקיות מתייחסות למשתנים x_2, x_3, ולפי אלה נקבעות הקשתות – יש קשת בין כל שני צמתים בשורות שונות, שמשלימות את ההשמה אחת של השנייה ביחס לשני הערכים מימין, בהתאם ליחס ההשלמה הקיים בפסוקיות עצמן. כיוון שהנוסחה ספיקה בגרף יש קליקה בגודל 2 (מספר הפסוקיות). כיוון שאין קשתות משורה לעצמה, זוהי גם הקליקה המקסימלית – לא יכולה להיות קליקה שמערבת 2 צמתים מאותה השורה.

שבו קיימת קליקה בגודל 2 – צומת אחד מכל שורה.

לא קשה להוכיח את תקינות הרדוקציה:

אם $\phi \in 3SAT$, קיימת השמה תקינה לכל המשתנים בנוסחה. כל שורה בגרף מכילה צומת שמייצג את אותו החלק מההשמה שמתייחס למשתנים בפסוקית שהשורה מייצגת. כיוון שההשמה תקינה, כל הצמתים האלה לא סותרים אחד השני, ובין כל זוג בהם קיימת קשת – זוהי קליקה בגודל מספר השורות m.

אם $\phi \notin 3SAT$, אין השמה תקינה לנוסחה, ולכן לא נוכל למצוא m צמתים בשורות שונות שמסכימים ביניהם על כל אחד מהליטרלים – אז אין קליקה בגודל m.

הרדוקציה הזו מעניינת במיוחד במקרה שלנו, בגלל היחס הישר שהיא מקיימת בין פסוקיות לצמתים. אם יש m פסוקיות בנוסחה, יש $7m$ צמתים בגרף. זה כשלעצמו אולי לא מאוד מיוחד, אבל ביחד עם אותו יחס הישר אנחנו מקבלים את התוצאה הבאה:

אם יש תת-קבוצה של $n < m$ פסוקיות בנוסחה שביניהן מייצרות תת נוסחה ספיקה, אנחנו מקבלים קליקה בגודל n בגרף.

עכשיו שאנחנו יודעים ש- $\text{GAP-E3SAT}_{1,s}$ היא NP-קשה, אנחנו יכולים להשתמש באותה הרדוקציה בדיוק כדי לקבל ש- GAP-CLIQUE_s קשה – אם הנוסחה ספיקה, ישנה קליקה בגודל m בגרף. אם אין תת-נוסחה של sm פסוקיות ספיקות בנוסחה, הקליקה המקסימלית בגרף תהיה קטנה מ- sm.

אז הצלחנו להראות שקשה לקרב את CLIQUE עד כדי s – אבל לא נעצור כאן! הגיע הזמן להשתמש במשפט PCP.

גרף $FGLSS$[27]:

לפי משפט PCP אנחנו יודעים ש- $3SAT \in PCP_{1,1/2}[O(lgn), O(1)]$. זאת אומרת שיש לנו מאמת V והוכחה c שהמאמת יכול לבדוק עם מחרוזת אקראית בסדר גודל לוגריתמי, תוך קריאת מספר תווים קבוע בהוכחה, עם שלמות של 1 ונאותות של חצי. כמו שעשינו בעבר, אנחנו מסתכלים על כל האופציות שהמחרוזת האקראית שלנו יכולה לקבל – כל אחת מגדירה "הוכחה חלקית", מספר קבוע של תווים מ- c שיגרמו ל- V לקבל את המבחן עבור מחרוזת אקראית יחידה.

[27] FGLSS אלה ראשי התיבות לשמות קבוצת החוקרים שגילתה את הקשר בין קירוב קליקות לבדיקה הסתברותית.

הגרף שאנחנו בונים מורכב גם כן משורות. כל מחרוזת אקראית תקבל שורה של צמתים, שבה כל צומת מייצג את קומבינציה יחידה של תווים שבקריאתם V מקבל, אם הוא מריץ את המבחן שהמחרוזת האקראית מגדירה. גם כאן, אנחנו מגדירים קשת בין כל זוג צמתים משורות שונות, שלא מגדירים באותו האינדקס של ההוכחה c שני תווים שונים.

יהי $q \in O(1)$ מספר התווים המקסימלי ש- V קורא בכל מבחן. אנחנו יכולים לומר שבכל שורה יש לכל היותר 2^q צמתים[28] ושיש לכל היותר מספר פולינומיאלי של שורות בגרף – כלומר הבנייה שלו פולינומיאלית.

בעיקר, אנחנו יודעים שעבור נוסחת $3CNF$ ספיקה, תהיה איזושהי הוכחה c שאותה המאמת מקבל בוודאות, ובגרף ה- $FGLSS$ אותה ההוכחה תחבר קליקה בין כל השורות. אם הנוסחה אינה ספיקה, הקליקה המקסימלית קטנה מחצי מספר השורות, כיוון שאין הוכחה שתעבור יותר מחצי המבחנים.

אבל, עם המאמת V אנחנו יכולים להגדיר את המאמת V^x, שחוזר על המבחן של V x פעמים. המאמת V^x הוא זה שמאפשר לנו להוריד את הנאותות של הוכחה נבדקת הסתברותית לכל סף, ובזכותו נקבל שלכל x טבעי, $3SAT \in PCP_{1,\left(\frac{1}{2^x}\right)}[O(lgn), O(1)]$. את זה כבר ידענו, אבל גרף $FGLSS$ הופך את התוצאה הזו לתוצאת קושי-של-קירוב עבור בעיית הקליקה. לכל $\epsilon = \left(\frac{1}{2^x}\right)$, המאמת V^x מאפשר לנו לבנות גרף שבו הקליקה המקסימלית מכילה רק ϵ ממספר השורות, כלומר קיבלנו רדוקציה מ- 3SAT אל GAP-CLIQUE **עבור כל קבוע חיובי – אי אפשר לקרב את קליקה במקרה הכללי בשום סדר גודל**.

לבעיית VERTEX-COVER, ראינו בהקדמה קירוב מסדר גודל של 2. לבעיות אחרות יש קירובים טריוויאליים אחרים (למשל, לא קשה להשתכנע שבכל נוסחת 3CNF לפחות חצי מהנוסחאות ספיקות). אנחנו יודעים לומר שלקליקה אין קירוב שכזה, אלא אם כן $P = NP$.

מכאן הדרך ממשיכה לחיפוש קירובים אחרים, שמוותרים על יחס קבוע בין הקירוב לבין הפתרון האופטימלי – למשל, קירוב מסדר $\ln(n)$ עבור בעיית SET-COVER, וכן הלאה. אנחנו מסיימים כאן, אבל משפט PCP פותח דרכים רבות במחקר הקירוב – והקושי שלו.

[28] אנחנו עדיין מניחים א"ב בינארי של ההוכחה.

דברים/אנשים שעזרו לי בדרך:

1. *The PCP theorem by Gap Amplification, Irit Dinur 2007*
https://www.wisdom.weizmann.ac.il/~dinuri/mypapers/combpcp.pdf

2. הקורס המעולה של ונקאטסאן גורוסוואמי וראיין אודונל:
https://courses.cs.washington.edu/courses/cse533/05au

3. בונה הגרפים https://csacademy.com/app/graph_editor
שמאפשר להוריד תרשימי גרפים בתור *PNG*, וגם הועיל מאוד בתכנון הגרף המרחיב בחלק ג – מפתיע כמה שמסובך לבנות גרף מאוד לא מסודר עם 12 צמתים בלבד...

4. פרויקט *Excalidraw*. אם חשבתם 'הו, זה ציור נחמד...' במהלך הקריאה – זה בזכותם.
https://github.com/excalidraw/excalidraw, https://excalidraw.com

5. *Python, numpy, sympy, matplotlib…*

6. החברים מהאוניברסיטה הפתוחה, שהקשיבו בסבלנות לנאומים מבולבלים על הרחבה ותיקון שגיאות, ובאמת באמת ניסו לגרום לי להבין מה אני רוצה מהחיים שלהם.

7. המנחה שלי, ד"ר אלעזר בירנבאום, שהסביר לי בעדינות מתי המילים שאני כותב לא מתחברות אחת לשנייה כמו שצריך...