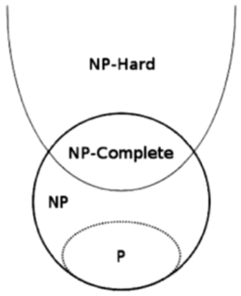
1: NP Completeness

P & NP



השוואה בין בעיות חישוב – לפי סדר גודל של סיבוכיות זמן ומקום. אנחנו יודעים לתת ניתוח כזה לאלגוריתם ספציפי.

אבל נרצה לאפיין את הבעיה עצמה. להוכיח שעבור בעיה מסויימת, כל פתרון ידרוש סיבוכיות זמן או מקום מסויימים.

לדוגמה אנחנו יודעים לנתח את אלגוריתם *Dijkstra*. אבל נרצה לאפיין את בעיית *shortest-path* עצמה.

בהינתן 2 בעיות, נרצה לקבוע מי יותר מסובכת. לדוגמה בעיות *max-flow, shortest-path*.

אנחנו יודעים שהאלגוריתמים הידועים ל-*max-flow* מסובכים יותר מאלה של *shortest-path*. העובדה הזו חסרת משמעות.

הגדרות גסות:

P (polynomial) – קבוצת כל ה**בעיות שיש להן** אלגוריתם פולינומי. כמו בעיית *max-flow, shortest-path*.

NP (nondeterministic polynomial) – קבוצת כל הבעיות שיש להן אלגוריתם פולינומי **אם** משתמשים באי-דטרמיניזם.

NPC (NP-complete) – קבוצת כל הבעיות שנמצאות ב-NP, ו**קשות לפחות כמו** כל הבעיות ב-*NP*.

אם נמצא אלגוריתם פולינומי לאחת הבעיות שם, אז זה יוכיח ש- $P = NP$.

NPH (NP-hard) – קבוצת כל השפות שקשות לפחות כמו כל הבעיות ב-*NP*.

Self-reductions – רדוקציות עצמיות

בעיות הכרעה ובעיות חיפוש: **בעיית חיפוש** – מחפשים פיתרון כלשהו. **בעיית הכרעה** – בהינתן פיתרון, האם הוא נכון.

באופן כללי – בנושא של שלמות-*NP*, מספיק לדבר על בעיות הכרעה ולא צריך לדבר על בעיות חיפוש.

נדגים באמצעות בעיית *k-clique*: בהינתן גרף G, נגדיר $\omega(G)$ את גודל הקליקה הכי גדולה שלו. אז השפה:

$$\text{k-CLIQUE} \coloneqq \{G : \omega(G) \geq k\}$$

היא קבוצת הגרפים שיש בהם קליקה בגודל k.

בעיית החיפוש היא הבעיה של מציאת הקליקה K_k בגרף (אם היא קיימת). **בעיית ההכרעה** היא הבעיה של בדיקה האם $G \in \text{k-CLIQUE}$.

בעיית החיפוש שייכת ל-*P*: נעבור על כל ה- $\binom{n}{k} \leq n^k$ קבוצות של k קודקודים. לכל אחת נבדוק אם היא קליקה. זה זמן פולינומיאלי.

ובאופן טריוויאלי, זה גם פתרון פולינומי לבעיית ההכרעה: אם מצאנו קליקה, אז היא קיימת.

טריוויאלי שבעיות הכרעה לא קשות יותר מבעיות חיפוש. במקרה הזה גם ההפך נכון: בעיית החיפוש *search k-clique* לא קשה יותר מבעיית ההכרעה:

נניח שיש לנו אלגוריתם פולינומי לבעיית ההכרעה – A. הוא מחזיר 0 או 1. נגדיר בעזרתו אלגוריתם פולינומי לבעיית החיפוש:

קלט: גרף G. **פלט:** קבוצת קודקודים $V' \subseteq V(G)$ שהיא K_k.

1. אם $A(G) = 0$, החזר ϕ.
2. כל עוד $|V(G)| \geq k$ (יש בגרף לפחות k קודקודים):
   2.1 נבחר קודקוד שרירותי $v \in V(G)$.
   2.2 אם $A(G - v) = 1$ (כלומר, יש *k-clique* ב-G בלי v) אז נגדיר $G \coloneqq G - v$.
3. נחזיר את G.

אם בשלב 1 לא החזרנו ϕ, אז זה אומר שיש קליקה. בשלב 2, כל פעם נבדוק קודקוד. אם בלעדיו עדיין יש קליקה, נוריד אותו.

אם בלעדיו אין קליקה, נשאיר אותו ונבדוק את הקודקוד הבא. ככה עד שנגיע ל-k קודקודים. ואז מצאנו קליקה בגודל k.

Deterministic Turing Machines (DTMs) – מכונת טיורינג דטרמיניסטית

מודל חישובי (אוטומט) המורכב מ:

- א"ב סופי Σ.
- סרט אינסופי חד צדדי, שיש לו "מיקומים".
  - כל מיקום יכול להכיל אות אחת מתוך הא"ב.

- ראש קריאה/כתיבה.
- קבוצה סופית של מצבים Q.
  - מצב התחלתי ייחודי q_{start},
  - מצב עצירה ייחודי q_{halt}.
- פונקציית מעברים $\delta: Q \times \Sigma \to \{\text{left}, \Delta, \text{right}\} \times Q \times \Sigma$.

תהליך החישוב של DTM:

- החישוב מתחיל עם הקלט כתוב על הסרט, המכונה במצב q_{start}, הראש מעל המיקום הראשון בסרט.
- כל עוד המכונה לא הגיעה ל- q_{halt}, בצע:
  - יהי $q \in Q$ המצב הנוכחי של המכונה.
  - נקרא אות $\sigma \in \Sigma$ מהמיקום הנוכחי.
  - לפי הפונקציה δ:
    - נכתוב אות σ' במיקום,
    - נישאר במקום או נזוז ימינה או שמאלה,
    - המכונה תעבור למצב q'.
- אם הגענו ל- q_{halt}: אם המילה $accept$ כתובה על הסרט, הקלט התקבל ע"י ה-DTM. אחרת, נדחה.
  - אם ה-DTM לעולם לא עוצרת, נאמר הקלט נדחה.

$L(M)$ – שפה של DTM. יהי DTM כלשהו M. אזי השפה של M היא קבוצת כל המילים שהמכונה מקבלת, ומוגדרת:

$$L(M) \coloneqq \{\omega \in \Sigma^* : M \text{ accepts } \omega\}$$

חישוב פונקציה ב-DTM: יהי DTM כלשהו M. אזי נאמר ש-M **מחשבת** פונקציה $f: \{0,1\}^* \to \{0,1\}^*$ אם לכל $x \in \{0,1\}^*$,

המכונה עוצרת ובסוף הריצה המילה $f(x)$ כתובה על הסרט. כלומר בהינתן הקלט x על הסרט, הפלט בסוף יהיה y.

מעבר אחד לפי δ נקרא **צעד חישובי** של ה-DTM. מספר הצעדים ש-DTM מבצעת בהינתן קלט הוא הזמן הנדרש לחישוב הקלט. הוא יכול להיות אינסופי.

חישוב ב-T זמן: יהיו פונקציות $T: \mathbb{N} \to \mathbb{N}$, $f: \{0,1\}^* \to \{0,1\}^*$.

נאמר ש-DTM כלשהי M מחשבת את f בזמן T אם לכל $n \in \mathbb{N}$ ולכל $x \in \{0,1\}^n$, בהינתן קלט x, M עוצרת עם $f(x)$ כתוב על הסרט, ב-$T(n)$ צעדים.

עבור פונקציה $T: \mathbb{N} \to \mathbb{N}$, נגדיר $DTIME\big(T(n)\big)$ את קבוצת כל הפונקציות הבוליאניות $f: \{0,1\}^* \to \{0,1\}$ (כלומר שפות),

שעבורן קיימת DTM שמחשבת את f בזמן $O\big(T(n)\big)$. הפונקציות האלה נקראות $O\big(T(n)\big)$-DTM-$computable$.

נגדיר את P באופן פורמלי: המחלקה P מכילה את כל השפות L שעבורן קיים DTM כלשהי M כך ש: $L = L(M)$. כלומר:

$$P \coloneqq \bigcup_{c \geq 1} DTIME(n^c)$$

כל השפות שהן $O\big(T(n)\big)$-DTM-$computable$ עבור $T(n) = n^c$ כלשהו, שזה כל הזמנים הפולינומיאלים.

מכונת טיורינג אי-דטרמיניסטית (להלן $NDTM$), בנויה באופן זהה למכונת טיורינג, אבל המעברים לא נקבעים ע"י פונקציה, אלא ע"י היחס:

$$\delta \subseteq (Q \times \Sigma) \times (Q \times \Sigma \times \{\text{left}, \Delta, \text{right}\})$$

כלומר, δ היא בעצם זוגות של: קלט ומצב, וקלט ומצב ותזוזה. כלומר לכל זוג $(q, \sigma) \in Q \times \Sigma$, יש מספר מעברים תקינים שהמכונה יכולה לבצע.

לכן המכונה אי-דטרמיניסטית.

מילה x מעל Σ מקיימת $x \in L(M)$ אם **קיימת** סדרת מעברים שהמכונה יכולה לבצע בהינתן x, כך שמגיעים ל- q_{halt} והמילה $accept$ כתובה על הסרט.

אם אין סדרה כזו, אז $x \notin L(M)$.

אם נתעלם מהגבלות זמן, DTM שקולים ל-$NDTM$ בכך שלכל $NDTM$ יש DTM שמדמה אותה.

נאמר ש-$NDTM$ כלשהי M רצה בזמן $T(n)$ אם לכל קלט $x \in \{0,1\}^n$ ולכל סדרת מעברים, M מגיעה ל- q_{halt} (בין אם מתקבלת או לא) תוך $T(n)$ צעדים.

עבור פונקציה $T : \mathbb{N} \to \mathbb{N}$, נגדיר: $NTIME\big(T(n)\big)$ מכילה את כל השפות $L \subseteq \{0,1\}^*$ שעבורן יש $NDTM$ שרצה בזמן $O\big(T(n)\big)$,

כך ש- $L(M) = L$.

1: NP Completeness

המחלקה NP מכילה את כל השפות שעבורן קיים $NDTM$ כלשהי M כך ש $L(M) = L$, כלומר:

$$NP \coloneqq \bigcup_{c\geq 1} NTIME(n^c)$$

Verification Algorithms – אלגוריתם אימות

תהי $L \subseteq \{0,1\}^*$ שפה שמקיימת $L = L(M)$ עבור $NDTM$ **פולינומיאלי** כלשהו M. כלומר, $L \in NTIME\big(p(n)\big)$ עבור פולינום כלשהו p.

אזי, לכל $x \in L$ קיימת **תעודה כלשהי** y, (שמייצגת את הבחירות עבור M) כך שבהינתן x, אם נכריח את M ללכת לפי y, M תעצור עם הפלט 1 על הסרט.

בנוסף, מתקיים $|y| \leq p(|x|)$. כלומר זה מגביל את מספר הצעדים של-M (כי M היא מכונה בזמן פולינומי).

אם $x \notin L$, אז לא קיימת תעודה y שיכולה לגרום ל-M לעצור מ-x עם 1 על הסרט.

זה מוביל אותנו להגדרה אלטרנטיבית ל- NP. שפה $L \subseteq \{0,1\}^*$ היא ב-NP אם קיימים:

פולינום $p : \mathbb{N} \to \mathbb{N}$, ו-$NDTM$ כלשהו M (אלגוריתם), כך ש: לכל $x \in \{0,1\}^*$,

מתקיים $x \in L$ **אם ורק אם** קיים $y \in \{0,1\}^{\leq p(|x|)}$ כך ש-M בהינתן x ו-y עוצר עם $accept$, תוך $p(|x| + |y|)$ צעדים.

כלומר התנאים:

1. האלגוריתם מחזיר אמת עבור y אמ"מ x שייך לשפה,
2. הגודל של y פולינומי בגודל של x,
3. האלגוריתם רץ בזמן פולינומי ב- $|x| + |y|$.

ה-$NDTM$ מההגדרה הזו נקרא **אלגוריתם אימות** עבור L.

טענה: $\text{k-CLIQUE} \in NP$. **הוכחה** – נגדיר אלגוריתם אימות:

קלט: גרף G, וקבוצת קודקודים X. הקבוצה X היא **העד.** אם $G[X] \cong K_k$ (k-קליקה) נחזיר 1. אחרת, 0.

האלגוריתם עומד בתנאים: אם $G \in \text{k-CLIQUE}$, אז אכן קיים X מתאים. והגודל שלו פולינומי בגודל הגרף. אחרת, לא קיימת קבוצה X כזו.

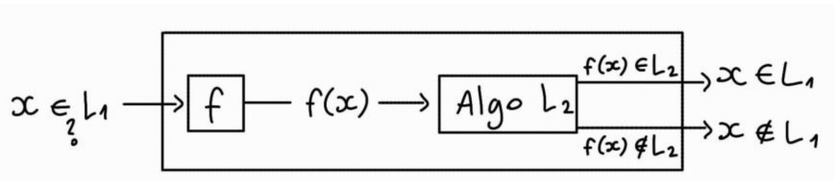
Polynomial Time Reductions

יהיו שפות $L_1, L_2 \subseteq \{0,1\}^*$. נסמן $L_1 \leq_p L_2$ אם קיימות: פונקציה $f: \{0,1\}^* \to \{0,1\}^*$ ופולינום $p: \mathbb{N} \to \mathbb{N}$ כך ש:

1. מתקיים $x \in L_1 \Leftrightarrow f(x) \in L_2$,
2. לכל $x \in L_1$, אפשר לחשב את $f(x)$ בזמן $p(|x|)$.

אם $L_1 \leq_p L_2$, נאמר שניתן לעשות רדוקציה מ- L_1 ל- L_2 בזמן פולינומי. הפונקציה f נקראת **רדוקציה פולינומית**.

במקרה זה נאמר ש L_1 קשה לכל היותר כמו L_2. למה? כי אם ידוע ש $L_1 \leq_p L_2$, ונתון אלגוריתם פולינומי עבור L_2, אז יש לנו אלגוריתם פולינומי ל- L_1:



שפה $L \subseteq \{0,1\}^*$ תיקרא $NP\text{-}hard$ אם $L' \leq_p L$ לכל $L' \in NP$.

כלומר, אם מכל שפה ב-NP אפשר לעשות רדוקציה בזמן פולינומי ל-L. זה אומר שהבעיה של L קשה **לפחות כמו** L'.

בדרך כלל כדי להוכיח ששפה L היא NPH, ניקח שפה L^* שאנחנו יודעים שהיא NPH, ונעשה רדוקציה ממנה ל-L.

NPC זה כל השפות ב- $NP \cap NPH$. אם נוכיח $P \cap NPC \neq \phi$, זה יוכיח ש $P = NP$.

The Cook-Levine Theorem

הזיהוי הראשון של שפה NPC. הקדמה – $CNF\text{-}SAT$:

יהיו משתנים בוליאניים $x_1, \dots, x_n$. לכל משתנה יש את הליטרלים שלו: החיובי x_i, השלילי $\bar{x}_i$.

1: NP Completeness

אם x_i קיבל את הערך אמת, אז $\bar{x}_i$ יקבל שקר. אם x_i קיבל שקר, אז $\bar{x}_i$ מקבל אמת.

נוסחה בוליאנית φ תיקרא *CNF* (*conjunctive normal form*) אם היא מורכבת מפסוקיות שביניהן יש רק $\vee$ (*OR*),

ובתוך כל פסוקית הליטרלים מופרדים על ידי $\wedge$ (*AND*). "מכפלה של סכומים":

$$\varphi = (x_1 \vee \bar{x}_4 \vee x_3 \vee \bar{x}_5) \wedge x_2 \wedge (\bar{x}_2 \vee \bar{x}_5)$$

נוסחה כזו תיקרא **ספיקה** אם קיימת השמה למשתנים שלו כך שהנוסחה יוצאת אמת. השמה כזו תיקרא **השמה מספקת.**

לדוגמה, ההשמה: $x_1 = x_2 = T, x_3 = x_4 = x_5 = F$. אז:

$$\varphi = (T \vee \bar{F} \vee F \vee \bar{F}) \wedge T \wedge (\bar{T} \vee \bar{F}) = (T \vee T \vee F \vee T) \wedge T \wedge (F \vee T) = T \wedge T \wedge T = T$$

נגדיר את השפה:

$$\text{CNF-SAT} := \{\varphi : \varphi \text{ is a satisfiable Boolean formula in CNF}\}$$

כלומר, כל הנוסחאות הבוליאניות שהן נוסחת *CNF* ספיקה. משפט *Cook-Levine* – השפה *CNF-SAT* היא ב-*NPC*.

עבור מספר טבעי k, נגדיר:

$$\text{k-CNF-SAT} \coloneqq \{\varphi\text{: satisfiable CNF formulas with } k \text{ literals in each clause}\}$$

כלומר ביטויי *CNF* עם k ליטרלים בכל פסוקית.

טענה: *3-CNF-SAT* היא *NPC*. קשה לפחות כמו כל בעיה ב-*NP*. המעבר מ-2 ל-3 העביר אותנו מ-*P* ל-*NPC*.

הוכחה: נוכיח שהיא ב-*NP* וגם *NPH*. כדי להוכיח שהיא ב-*NP*, נגדיר אלגוריתם אימות לבעיה:

קלט: φ, נוסחת *3-CNF*. a, השמה למשתנים של φ. אם a מספקת את φ, נחזיר אמת. אחרת, שקר.

האלגוריתם מקיים את 3 התנאים הבאים באופן טריוויאלי:

1. האלגוריתם מחזיר אמת **אם ורק אם** $\varphi \in \text{3-CNF-SAT}$.
2. הגודל של a פולינומי בגודל של φ.
3. האלגוריתם רץ בזמן $|\varphi| + |a|$.

כדי להוכיח שהשפה ב-*NPH*, צריך להראות ש $\forall L \in NP,\ L \leq_p \text{3-CNF-SAT}$. למעשה, נוכיח ש $L^* \leq_p \text{3-CNF-SAT}$ עבור $L^* \in NPH$ כלשהי.

נשתמש ב *CNF-SAT* כי אנחנו יודעים שהיא *NPH*. (אמרנו שהיא *NPC*, וזה גם *NPH*). נצטרך להראות ש $\text{CNF-SAT} \leq_p \text{3-CNF-SAT}$. נמצא פונקציה f כך שבהינתן נוסחה φ שהיא *CNF*, יתקיים:

1. $f(\varphi)$ היא נוסחת *3-CNF*.
2. מתקיים: $f(\varphi)$ ספיקה **אם ורק אם** φ ספיקה.
3. ניתן לחשב את $f(\varphi)$ בזמן פולינומי בגודל של φ.

הבהרה: הפונקציה הזו לא פותרת את בעיית *CNF-SAT* או *3-CNF-SAT*. היא רק מתרגמת ביניהן. אם נמצא אלגוריתם פולינומי ל-*3-CNF-SAT*, אז יהיה אלגוריתם פולינומי ל-*CNF-SAT*.

נגדיר את הפונקציה: נחליף כל פסוקית $C = \ell_1 \vee \ell_2 \vee \cdots \vee \ell_m$ **בגאדג'ט הפסוקית** (*clause gadget*) לפי המקרים:

אם $m = 3$, נעתיק את הפסוקית כמו שהיא.

אם $m < 3$, נחזור על ליטרלים כדי להשלים ל-3:

$$C = \ell_1 \longrightarrow C' = \ell_1 \vee \ell_1 \vee \ell_1, \qquad C = \ell_1 \vee \ell_2 \longrightarrow C' = \ell_1 \vee \ell_2 \vee \ell_2$$

אם $m > 3$, נגדיר $m - 3$ משתנים חדשים: $y_1^C, y_2^C, \dots, y_{m-3}^C$.

לכל C כזו, הפונקציה f תגדיר גאדג'ט לאותו C ומשתנים אלו יופיעו רק בגאדג'ט של C. הגאדג'ט עבור הפסוקית C יוגדר:

$$(\ell_1 \vee \ell_2 \vee y_1) \wedge (\overline{y_1} \vee \ell_3 \vee y_2) \wedge (\overline{y_2} \vee \ell_4 \vee y_3) \wedge \cdots \wedge (\overline{y_{m-3}} \vee \ell_{m-1} \vee \ell_m)$$

דוגמאות:

$$\varphi = \underbrace{(x_1 \vee \overline{x_2})}_{C_1, m=2} \wedge \underbrace{(x_3 \vee x_4 \vee \overline{x_2} \vee x_1)}_{C_2, m=4} \longrightarrow f(\varphi) = \underbrace{(x_1 \vee \overline{x_2} \vee \overline{x_2})}_{G_1} \wedge \underbrace{\left(x_3 \vee x_4 \vee y_1^{C_2}\right) \wedge \left(\overline{y_1^{C_2}} \vee \overline{x_2} \vee x_1\right)}_{G_2}$$

$$\varphi = (x_1 \vee x_2 \vee \overline{x_3} \vee x_4 \vee x_5 \vee \overline{x_6} \vee x_7)$$
$$\longrightarrow f(\varphi) = (x_1 \vee x_2 \vee y_1) \wedge (\overline{y_1} \vee \overline{x_3} \vee y_2) \wedge (\overline{y_2} \vee x_4 \vee y_3) \wedge (\overline{y_3} \vee x_5 \vee y_4) \wedge (\overline{y_4} \vee x_6 \vee x_7)$$

$$\varphi = \underbrace{(x_1 \vee \overline{x_2} \vee x_3 \vee \overline{x_4})}_{C_1} \wedge \underbrace{(\overline{x_1} \vee \overline{x_3} \vee \overline{x_5} \vee x_4)}_{C_2}$$
$$\longrightarrow f(\varphi) = \underbrace{\left(x_1 \vee \overline{x_2} \vee y_1^{C_1}\right) \wedge \left(\overline{y_1^{C_1}} \vee x_3 \vee \overline{x_4}\right)}_{G_1} \wedge \underbrace{\left(\overline{x_1} \vee \overline{x_3} \vee y_1^{C_2}\right) \wedge \left(\overline{y_1^{C_2}} \vee \overline{x_5} \vee y_4\right)}_{G_2}$$

נוכיח ש-f מקיימת את 3 התנאים לכל φ שהיא נוסחת CNF:

1. $f(\varphi)$ היא נוסחת 3-CNF. (טריוויאלי, כי לפי הגדרת הבנייה כל G תהיה באורך 3).

3. ניתן לחשב את $f(\varphi)$ בזמן פולינומי – כי מספר המשתנים החדשים פולינומי ב- $|\varphi|$, ומספר הפסוקיות החדשות פולינומי ב-$|\varphi|$.

הוכחת תנאי 2:

כיוון ראשון: נניח ש φ ספיקה. תהי a השמה המספקת את φ.

נזכור ש $f(\varphi)$ מכילה משתנים חדשים שאינם ב-φ המקורית, ולכן יכול להיות ש-a אינה השמה תקינה עבור $f(\varphi)$.

נגדיר a' השמה מספקת עבור $f(\varphi)$: לכל x שהוא משתנה מקורי של φ, $a'(x) = a(x)$.

לכל פסוקית $C = \ell_1 \vee \ell_2 \vee \cdots \vee \ell_m$ באורך $m \geq 4$, יש משתנים לוקאליים y. למשתנים אלו אין השמה ב-a. ניתן להם השמה:

מכיוון ש-a מספקת את φ, קיים $i \in [m]$ כך ש- $\ell_i = 1$ (יש ליטרל חיובי כלשהו ב-C). ואותו ליטרל הוא משתנה מקורי של φ.

לכל $j \in [m-3]$: אם $j < i$ זה משתנים שמופיעים לפני ℓ_i בגאדג'ט, אם $j \geq i$ זה משתנים שמופיעים אחרי. נגדיר:

$$a'\left(y_j^C\right) = \begin{cases} 1, & j < i \\ 0, & j \geq i \end{cases}$$

נוכיח שזו השמה מספקת:

כל הפסוקיות בגאדג'ט שאינן מכילות את ℓ_i ומופיעות לפניו, מסופקות כי $a\left(y_j^C\right) = 1$. הפסוקית שמכילה את ℓ_i מסופקת כי $\ell_i = 1$.

כל הפסוקיות אחרי ℓ_i מסופקות בגלל ה- $\bar{y}_j$.

כיוון שני: נניח ש $f(\varphi)$ ספיקה, ותהי a' השמה מספקת עבורה. נגדיר השמה a המספקת את φ: לכל x שהוא משתנה מקורי של φ, $a'(x) = a(x)$.

נוכיח שזו השמה מספקת: נב"ש שלא, כלומר קיימת פסוקית $C = \ell_1 \vee \ell_2 \vee \cdots \vee \ell_m$ של φ כך ש: $\ell_1 = \ell_2 = \cdots = \ell_m = 0$.

עבור $m \leq 3$, זה אותם משתנים אז ההשמה לא מספקת גם את $f(\varphi)$, סתירה.

עבור $m \geq 4$, נבחר את הגאדג'ט המתאים ל-C ב- $f(\varphi)$: בגלל שאנחנו יודעים ש-a' מספקת את $f(\varphi)$, ו-a לא מספקת את φ חייב להתקיים:

$$(\ell_1 \vee \ell_2 \vee y_1) \wedge (\overline{y_1} \vee \ell_3 \vee y_2) \wedge (\overline{y_2} \vee \ell_4 \vee y_3) \wedge \cdots . \wedge (\overline{y_{m-3}} \vee \ell_{m-1} \vee \ell_m)$$

(המסומנים בירוק הם 1, כל השאר 0). כי כל ה-ℓ הם 0, אז y_1 חייב להיות 1. אז $\overline{y_1}$ הוא 0, אז y_2 חייב להיות 1...

כל פעם, זה מכריח את ה-y הבא להיות 1. ובפסוקית האחרונה, נקבל שהכל 0.

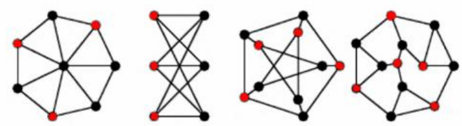
כלומר בכל מקרה הפסוקית האחרונה תהיה לא מסופקת, בסתירה לכך ש-a' מספקת את $f(\varphi)$.

בסך הכל, נקבל ש הפונקציה f שלנו תקינה, כלומר 3-CNF-SAT $\leq_p$ CNF-SAT אז *3-CNF-SAT* היא ב-*NPH*.

ומכיוון שהראנו שהיא *NP*, בסך הכל קיבלנו שהיא *NPC*, כנדרש.

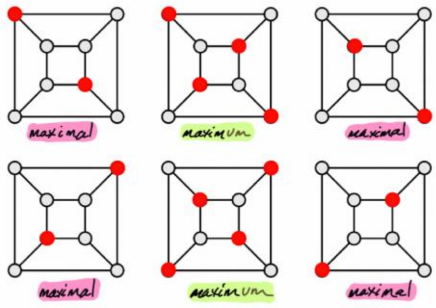
Independent Sets in Graphs

יהי גרף G. קבוצת קודקודים $I \subseteq V(G)$ תיקרא בלתי תלויה אם היא לא פורשת אף צלע (אין אף צלע בין כל שני קודקודים בקבוצה).



נסמן $\alpha(G)$ את גודל הקבוצה הבת"ל הגדולה ביותר (מקסימום).

מקסימום זאת הקבוצה הכי גדולה שאפשר למצוא. **מקסימלי** זו קבוצה שלא ניתן להגדיל. כל מקסימום הוא מקסימלי, ולא בהכרח הפוך.



הבעיה של מציאת קבוצה מקסימלית היא ב-P. נפעיל אלגוריתם חמדן: נבחר קודקוד ונוריד את השכנים שלו, ככה לכל הקודקודים.

הבעיה של מציאת קבוצת מקסימום היא NPC. נגדיר את השפה:

$$IS \coloneqq \{(G,k) : \alpha(G) \geq k\}$$

נוכיח שהיא NPC. **תחילה נוכיח שהיא ב-*NP*.** נגדיר אלגוריתם אימות:

קלט: גרף G, קבוצת קודקודים $V' \subseteq V(G)$. לכל $u,v \in V'$, אם קיימת צלע uv בגרף, נחזיר 0. אם בדקנו את כל הזוגות ואין צלע, נחזיר 1.

האלגוריתם מקיים את התנאים (באופן טריוויאלי):

1. האלגוריתם מחזיר 1 אם ורק אם $(G,|V'|) \in IS$. 2. הגודל של V' פולינומי ב- $|V|$. 3. האלגוריתם רץ בזמן $O(|G| + |V'|^2)$.

נוכיח שהשפה *NPH*: נראה ש $3\text{-CNF-SAT} \leq_p IS$. כלומר נראה רדוקציה - פונקציה f כך שבהינתן נוסחה φ שהיא *3-CNF*, יתקיים:

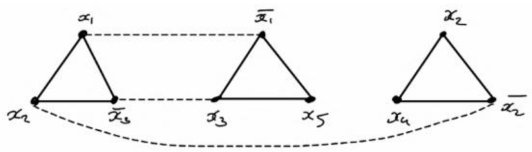
1.$f(\varphi)$ הוא זוג סדור (G_φ, k). 2. $\alpha(G_\varphi) \geq k$ **אם ורק אם** φ ספיקה. 3. ניתן לחשב את $f(\varphi)$ בזמן פולינומי.

נגדיר את f: בהינתן φ נוסחת *3-CNF* עם m פסוקיות, נגדיר גרף G_φ כך:

לכל פסוקית $C \coloneqq \ell_1 \vee \ell_2 \vee \ell_3$ ב-φ נגדיר משולש עם קודקודים שהם הליטרלים.

בנוסף, קודקודים שנמצאים על משולשים שונים יתחברו בצלע אם הליטרלים שלהם משלימים. לדוגמה:

עבור $\varphi = (x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_5 \vee x_3) \wedge (x_4 \vee \overline{x_2} \vee x_2)$, נקבל:



הפונקציה f מחזירה את (G_φ, m). נוכיח שהיא עומדת בתנאים:

1. מתקיים לפי הבנייה.

3. מספר המשולשים בתחילת הבנייה שווה למספר הפסוקיות ב-φ. מספר הצלעות גם פולינומי בגודל של φ, כי הוא חסום ב- $(3m)^2$.

נוכיח את 2:

כיוון ראשון: נניח ש φ ספיקה, ותהי a השמה מספקת כלשהי עבור φ. כלומר a מספקת לפחות ליטרל אחד בכל פסוקית.

נבחר ליטרל מסופק אחד (שרירותית) מכל פסוקית. כל ליטרל כזה מתאים לקודקוד בגרף.

קבוצת הקודקודים הזו היא קבוצה בת"ל בגודל m בגרף G_φ. למה?

בחרנו ליטרל אחד מכל פסוקית, אז זה נותן לנו בדיוק m קודקודים. נב"ש שהקבוצה תלויה, כלומר שקיימים שני קודקודים שיש ביניהם צלע.
לפי הגדרת הבנייה, זה קורה אמ"מ הם משלימים. אבל בחרנו ליטרלים מסופקים, אז לא יכול להיות שבחרנו את שניהם. מש"ל.

כיוון שני: נניח ש $\alpha(G_\varphi) \geq m$, ותהי $I \subseteq V(G_\varphi)$ קבוצה בת"ל בגודל m ב-G_φ.
הקבוצה I חייבת להכיל בדיוק קודקוד אחד מכל משולש (כי אם יש 2 ממשולש, הקבוצה לא בת"ל. וצריך m קודקודים).
כלומר: לכל משולש ב-G_φ, יש בדיוק קודקוד אחד ששייך ל-I.
נגדיר השמה a' שמספקת את φ, בצורה הבאה:
כל ליטרל שנמצא ב-I, יסופק (יקבל ערך *TRUE*). עבור שאר המשתנים, ניתן השמה שרירותית, שאינה סותרת.
אז לכל פסוקית יש לפחות ליטרל אחד שקיבל *TRUE*. ואין סתירות, כי אין ליטרלים ב-I שמופיעים גם המקורי וגם המשלים ב-φ.
(כי אחרת הייתה ביניהם צלע, וזו סתירה לכך ש-I בת"ל).
בסה"כ, הוכחנו $IS \in NP \cap NPH = NPC$.

Graph Coloring

k-coloring (k-צביעה): עבור גרף G, פונקציה $f: V(G) \to [k]$ תיקרא k-צביעה של G אם לכל $(u,v) \in E(G)$, מתקיים $f(u) \neq f(v)$.
Chromatic Number (מספר כרומטי): נסמן ב- $\chi(G)$ את ה-k הכי קטן שעבורו קיימת k-צביעה של G.
אבחנה: עבור k-צביעה כלשהי, כל "מחלקת צבע" (קבוצה של קודקודים שכולם באותו הצבע, $\{v \in V(G): f(v) = i\}$) היא קבוצה בת"ל.
(לפי הגדרת k-צביעה). לכן, k-צביעה של גרף שקולה לחלוקת הקודקודים ל-k קבוצות בת"ל.
גרף G הוא 2-צביע אמ"מ הוא דו-צדדי (*Bipartite*). שאפשר לחלק את הקודקודים שלו ל-2 קבוצות שאין ביניהן צלע.
נגדיר את השפות:

$$2COL \coloneqq \{G: \chi(G) \leq 2\}, \qquad 3COL \coloneqq \{G: \chi(G) \leq 3\}$$

באופן טריוויאלי, *2COL* היא ב-*P* כי זה שקול לבדיקה האם גרף הוא דו צדדי (בדיקה האם יש בגרף מעגלים אי זוגיים).
עוד אבחנה: נסמן $n = V(G)$, אזי מתקיים:

$$\alpha(G) \geq n/\chi(G)$$

גודל הקבוצה הבת"ל הכי גדולה היא לפחות מספר הקודקודים חלקי המספר הכרומטי. אם הגרף הוא 3-צביע,
אז אפשר למצוא 3 קבוצות בת"ל וזה אומר שהקבוצה הבת"ל הגדולה היא **לפחות** בגודל שליש הקודקודים.
טענה: *3COL* היא ב-*NPC*. כמו עם *k-CNF-SAT*, המעבר מ-2 ל-3 מעביר את הבעיה מ-*P* ל-*NPC*. כדי להוכיח את זה, נציג עוד שפה:

NAE-SAT

בהינתן נוסחת *k-CNF* כלשהי φ, נאמר שהיא בתצורת *NAE-SAT* (או *"Not All Equal" Satisfiable*),
אם קיימת לה השמה מספקת כך שבכל פסוקית קיים לפחות ליטרל אחד לא מסופק. (מעצם ההגדרה שההשמה מספקת, יש לה לפחות גם ליטרל מסופק אחד).
נגדיר את השפה *NAE-k-CNF-SAT*, כל הנוסחאות במבנה *k-CNF* שהן גם *NAE-SAT*:

$$\text{NAE-k-CNF-SAT} \coloneqq \{\varphi: \varphi \text{ is an NAE-satisfiable k-CNF formula}\}$$

השפה *NAE-3CNF-SAT* היא *NPC*.

כעת, נוכיח ש-*3-COL* היא ב-*NPC*. קודם **נוכיח שהיא ב-*NP*** על ידי הגדרת אלגוריתם אימות:
קלט: גרף G, 3-צביעה f. לכל צלע, נבדוק אם $f(u) = f(v)$. אם זה מתקיים באחת הצלעות, נחזיר 0. אם בכל הצלעות זה לא מתקיים, נחזיר 1.
מתקיים באופן טריוויאלי: האלגוריתם מחזיר 1 אמ"מ הגרף הוא 3-צביע, הגודל של f (התחום) פולינומי בגודל של G, והאלגוריתם פולינומי ב- $|G| + |f|$.

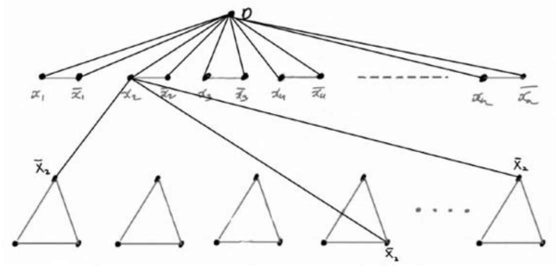
נוכיח שהשפה ב-*NPH*: מספיק להראות ש: $\text{NAE-3-CNF-SAT} \leq_p 3COL$, ע"י רדוקציה.
נתאר פונקציה f שבהינתן נוסחה $\varphi \in \text{3-CNF}$:
1. $f(\varphi)$ הוא גרף G_φ, 2. מתקיים: $\chi(G_\varphi) \leq 3$ **אמ"מ** $\varphi \in \text{NAE-3-CNF-SAT}$, 3. ניתן לחשב את $f(\varphi)$ בזמן פולינומי.

הגדרת f: בהינתן φ נוסחת $3\text{-}CNF$, נגדיר גרף G_φ כך:

לכל פסוקית $C := \ell_1 \vee \ell_2 \vee \ell_3$ ב-φ, נגדיר משולש כמו ברדוקציה הקודמת (נקרא לזה **"הגאדג'ט של הפסוקיות"**).

בנוסף, לכל משתנה x_i ב-φ נגדיר צלע $e = (x_i, \bar{x}_i)$ (**"הגאדג'ט של המשתנים"**). ונוסיף קודקוד D (*don't care*) לגרף.

נחבר את D עם כל קודקוד בגאדג'טים של המשתנים. ונחבר כל ליטרל בגאדג'טים של המשתים עם המשלים שלו בגאדג'טים של הפסוקיות:



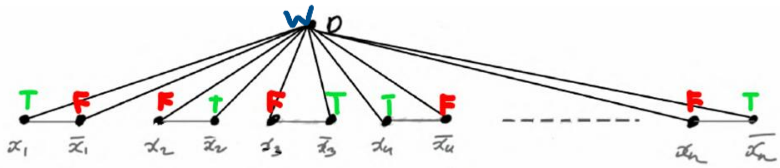
נוכיח ש-f מקיימת את התנאים:

1 טריוויאלי לפי הגדרת הבנייה, 3 מתקיים כי מספר הגאדג'טים והצלעות פולינומי במספר המשתנים ומספר הפסוקיות.

נוכיח את 2 – נראה ש $\varphi \in \text{NAE-3-CNF-SAT} \Leftrightarrow \chi(G) = 3$. (פורמלית היינו מוכיחים $\chi(G) \leq 3$, אבל יש משולשים אז השוויון טריוויאלי).

כיוון ראשון: נניח ש φ בצורת $NAE\text{-}SAT$, ותהי a השמה מספקת NAE עבורה.

נגדיר 3-צביעה עבור G_φ באופן הבא: הצבעים יהיו T, F, W, ונצבע את הקודקודים בגאדג'טים של המשתנים בהתאם ל-a, לדוגמה:



נגדיר את הצביעה עבור הגאדג'טים של הפסוקיות (המשולשים):

עבור פסוקית $C := \ell_1 \vee \ell_2 \vee \ell_3$, נסרוק אותה משמאל לימין ונעצור בליטרל המסופק הראשון. נצבע אותו T.

מכיוון שזו השמת NAE, יש ליטרל אחר שלא מסופק ב-C. נצבע אותו ב-F. ונותר עוד ליטרל אחד ב-C, נצבע אותו W.

נותר להראות שהצביעה היא 3-צביעה תקינה (שאין שני קודקודים סמוכים באותו הצבע):

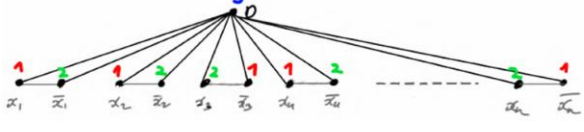
נשים לב ש-W לא בעייתי כי הוא מופיע רק פעם אחת במשולש ובקודקוד D. המשולשים לא מחוברים ישירות ל-D ואין צלעות בין המשולשים.

כמו כן, אין שני ליטרלים באותה השמה באותו משולש או בגאדג'ט של המשתנים.

אז הבעיה היחידה שיכולה להיות היא האיור הימני:

אבל כל החיבורים האלה הם רק מליטרל למשלים שלו (כמו באיור השמאלי), ולפי הצביעה לא יהיה מצב כזה ששניהם באותו צבע.

כיוון שני: נתון $\chi(G_\varphi) = 3$. תהי ψ פונקציית ה-3-צביעה. נניח בה"כ שהצבעים 1,2 מופיעים בגאדג'טים של המשתנים ולכן D נצבע ב-3.



נגדיר השמה בתצורת $NAE\text{-}SAT$ שמספקת את φ: היא תהיה בהתאם לצביעה כאשר 1 זה T, 2 זה F.

נראה שההשמה תקינה – כל משתנה בעל ערך יחיד: בגאדג'טים של המשתנים כל משתנה מופיע בדיוק פעמיים – ליטרל והמשלים שלו.

פעם אחת חיובי, ופעם ושלילי. אז אין סתירות בהשמה.

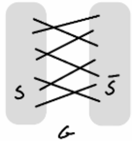
נראה שהיא בתצורת $NAE\text{-}SAT$: מכיוון שהצביעה היא 3-צביעה, בכל משולש יש שימוש בכל הצבעים. ובפרט צבעים 1,2 כלומר בכל פסוקית יש F,T.

בסה"כ, הוכחנו ש-$3COL$ היא NP וגם NPH ולכן NPC.

The Max-Cut Problem

בהינתן גרף G, קבוצת קודקודים $S \subseteq V(G)$, והקבוצה המשלימה $\bar{S} = V(G) \setminus S$, נסמן את קבוצת הצלעות החוצות מ-S ל-$\bar{S}$ כך:

$$E_G(S, \bar{S}) = \{uv \in E(G) : u \in S, v \in \bar{S}\}$$



החלוקה הזו נקראת **חתך** (*cut*). בנוסף, נסמן את גודל החתך: $e_G(S, \bar{S}) = |E_G(S, \bar{S})|$. גודל החתך המקסימום בגרף G:

$$\sigma(G) \coloneqq \max_{S \subseteq V(G)} e_G(S, \bar{S})$$

זוג $(S, \bar{S})$ שמקיים $e_G(S, \bar{S}) = \sigma(G)$ ייקרא **חתך מקסימום** של G.

נגדיר את השפה *MAX-CUT*, קבוצת הזוגות: $\{(G, k): \sigma(G) \geq k\}$, גרפים שהחתך המקסימום שלהם לפחות בגודל k. טענה: היא *NPC*.

נגדיר אלגוריתם אימות כדי להראות שהיא *NP*:

קלט: גרף G, חלוקה של הקודקודים $(S, \bar{S})$, מספר טבעי k.

נגדיר $x = 0$. לכל צלע uv, אם הצלע מחברת בין הקבוצות $S, \bar{S}$, נוסיף 1 ל-x. אחרי שעברנו על הצלעות: אם $x \geq k$ נחזיר אמת. אחרת, שקר.

הנכונות של האלגוריתם טריוויאלית – זו ספירה של מספר הצלעות בחתך. הפולינומיות של הגודל של העד וזמן הריצה גם טריוויאלית.

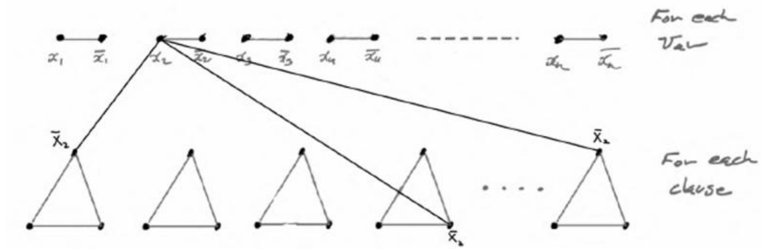
זמן הריצה הוא $O(|E(G)|)$.

כדי להראות שהשפה *NPH*, נראה ש $\text{NAE-3-CNF-SAT} \leq_p \text{MAX-CUT}$ באמצעות הפונקציה f, שצריכה לקיים:

1. $f(\varphi) = (G_\varphi, k)$, 2. $\sigma(G_\varphi) \geq k \Leftrightarrow \varphi \in \text{NAE-3-CNF-SAT}$, 3. אפשר לחשב את $f(\varphi)$ בזמן פולינומי

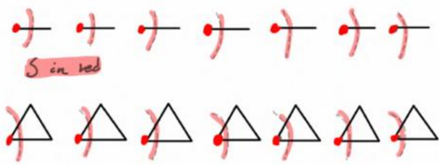
נגדיר את הגרף שהפונקציה מחזירה:

בהינתן φ נוסחת *3-CNF*, נגדיר גרף G_φ בדומה לרדוקציה של $\text{NAE-3-CNF-SAT} \leq_p 3COL$, אבל בלי הקודקוד D:



נגדיר את ה-k שהיא מחזירה:

נניח שיש ב-φ m פסוקיות ו-n משתנים. ראשית, אינטואיציה. אם נגדיר את S כדלהלן:

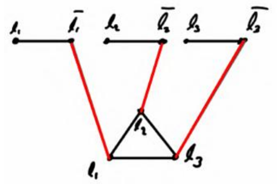


אז כל צלע בגאדג'ט של המשתנים תורמת צלע לחתך, וכל משולש תורם 2 צלעות לחתך. ובינתיים נקבל:

$$\sigma(G) = (\text{number of literals}) + 2(\text{number of clauses}) = n + 2m$$

נסתכל על הצלעות "באמצע" של G_φ, כלומר אלו שמחברות בין הגאדג'טים של המשתנים לגאדג'טים של הפסוקיות. כמה כאלו יש?

אם נתבונן בפסוקית שמיוצגת ע"י משולש, לכל קודקוד בה (שהוא ליטרל), יש שכן ייחודי (המשלים שלו) בגאדג'טים של הפסוקיות:



אז עבור m פסוקיות, נקבל $3m$ צלעות באמצע.

אנחנו לא יודעים איזה או כמה מהצלעות האלו חוצות את החתך. אבל בסה"כ, יש לכל היותר $3m + 2m + n = 5m + n$ צלעות בחתך.

נשים לב שאם נמצא $S \subseteq V(G_\varphi)$ כך ש $e_{G_\varphi}(S, \bar{S}) \geq n + 5m$, אז יתקיים:

$$e_{G_\varphi}(S, \bar{S}) = n + 5m$$

כי זה הגודל המקסימום של חתך אפשרי. כי כדי לקבל חתך גדול יותר, משולש יצטרך לתרום יותר משתי צלעות לחתך. וזה לא אפשרי. כלומר,

$$\sigma(G_\varphi) = e_{G_\varphi}(S, \bar{S})$$

נחזור להגדרת הרדוקציה: בהינתן φ, נגדיר $f(\varphi) = (G_\varphi, 5m + n)$.

נבדוק שהפונקציה מקיימת את התנאים הנדרשים:

1 מתקיים באופן טריוויאלי מתהליך הבנייה. 3 מתקיים בדומה להסבר ברדוקציה של NAE-3-CNF-SAT $\leq_p 3COL$.

נוכיח את 2 – נראה שמתקיים $\varphi \in$ NAE-3-CNF-SAT $\Leftrightarrow (G_\varphi, 5m + n) \in$ MAX-CUT:

כיוון ראשון: נניח ש $(G_\varphi, 5m + n) \in$ MAX-CUT, ויהי $(S, \bar{S})$ חתך המקיים $e_{G_\varphi}(S, \bar{S}) = n + 5m$. (נוכל לדרוש שוויון בעקבות המסקנה לעיל).

לפי אותה מסקנה, בגלל ש $e_{G_\varphi}(S, \bar{S}) = n + 5m$, נובע ש $\sigma(G_\varphi) = e_{G_\varphi}(S, \bar{S})$.

בפרט, חייב להתקיים עבור החתך שכל צלע בגאדג'ט של המשתנים היא צלע חוצה (קצה אחד ב-S והשני ב-$\bar{S}$).

נגדיר השמה a שתספק את φ ותהיה בתצורת $NAE\text{-}SAT$ כך:

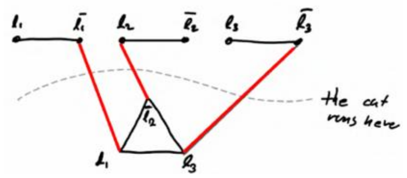
נתמקד בגאדג'טים של המשתנים. לכל צלע בהם נוודא ש-a מספקת את הליטרל שנמצא ב-S (ובהתאמה, לא מספק את הליטרל המשלים שנמצא ב-$\bar{S}$):

ההשמה תקינה (כל משתנה בעל ערך יחיד) כי קבענו את הערכים לפי הגאדג'טים של המשתנים וכל משתנה הופיע שם פעם אחת (עם המשלים).

נראה שהיא בתצורת $NAE\text{-}SAT$:

לכל פסוקית בגאדג'טים של הפסוקיות (המשולשים) יש 2 צלעות החוצות את החתך $(S, \bar{S})$.

בנוסף, כל צלע אמצעית ב-G_φ חוצה את החתך, ובפרט הם מהצורה $(\ell, \bar{\ell})$ - ליטרל והמשלים שלו.



אז לכל פסוקית יש ליטרל מ-S וליטרל מ-$\bar{S}$.

תהי C פסוקית כלשהי בגאדג'ט הפסוקיות, ויהי ℓ ליטרל המופיע בה.

אם $\ell \in S$, אזי הוא מחובר לליטרל $\bar{\ell} \in \bar{S}$ בגאדג'ט המשתנים, ולפי ההשמה a הוא יקבל את הערך T.

אם $\ell \in \bar{S}$, אזי הוא מחובר לליטרל $\bar{\ell} \in S$ בגאדג'ט המשתנים, ולפי ההשמה a הוא יקבל את הערך F.

מכיוון שכל פסוקית מכילה ליטרלים מ-S ומ-$\bar{S}$, אז בכל פסוקית יש ליטרל מסופק וליטרל לא מסופק. מש"ל.

כיוון שני: נניח ש φ בתצורת $NAE\text{-}SAT$ ותהי a השמת NAE שמספקת אותה. נמצא חתך ב-G_φ בגודל $5m + n$.

נגדיר את $S \subseteq V(G_\varphi)$ להיות קבוצת כל הקודקודים שהליטרלים שלהם מסופקים תחת a.

נשים לב ש-S מכילה קודקודים גם מהגאדג'טים של המשתנים וגם מהגאדג'טים של הפסוקיות.

נספור את הצלעות שחוצות את החתך:

מכיוון ש-a היא השמת NAE מספקת, בכל משולש בגאדג'ט הפסוקיות יש גם ליטרל שהוא T וגם ליטרל שהוא F. כלומר יש קודקוד מ-S וגם מ-$\bar{S}$.

לכן, כל פסוקית תורמת לפחות 2 צלעות החוצות את החתך.

בנוסף, מכיוון ש-a עקבית, כל צלע בגאדג'ט המשתנים חוצה את החתך (כי היא מוגדרת ע"י ליטרל והמשלים שלו).

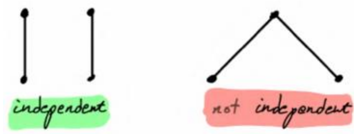
לבסוף, כל צלע "אמצעית" ב-G_φ היא מהצורה $(\ell, \overline{\ell})$ ולכן קודקוד אחד מסופק והשני לא, אז אחד ב-S והשני ב-$\bar{S}$. כלומר חוצה את החתך.

בסה"כ, יש לפחות $5m + n$ צלעות החוצות את החתך, ולפי הטענות שהוכחנו, יתקיים $\sigma(G_\varphi) \geq 5m + n$.

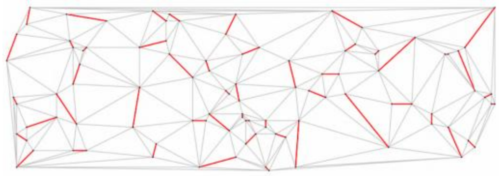
כלומר הוכחנו את התנאי השני של הפונקציה, אז הוכחנו ש $\text{MAX-CUT} \in NPH$ ולכן היא NPC.

König's Theorem

Independent Edges – נאמר ש-2 צלעות הן בת"ל אם אין להן קודקוד משותף:



Matching – קבוצה $M \subseteq E(G)$ תיקרא **שידוך** אם כל הצלעות בה בת"ל:



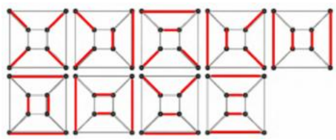
נסמן $V(M)$ את הקודקודים **המרווים** ע"י M (הקודקודים ש-M "תופס").

לפעמים נתייחס ל-M בתור תת גרף של G, המיוצג ע"י $(V(M), M)$.

שידוך M המקיים $V(M) = V(G)$ ייקרא **תת-גרף פורש** של G (תת-גרף שמכיל את כל הקודקודים). לדוגמה - $(V(G), \phi)$.

1-factor – שידוך מושלם

עבור גרף G, קבוצה $M \subseteq E(G)$ הפורשת את G תיקרא שידוך מושלם:

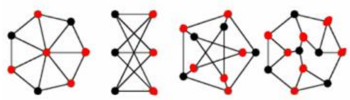


בעיית ***Max Matchings*:** בהינתן גרף, נמצא שידוך בגודל מקסימום. הבעיה הזו נמצאת ב-P. כל שידוך מושלם הוא בפרט גם שידוך מקסימום.

בגרף כללי אפשר לפתור ע"י אלגוריתם *Edmonds* או *Hopcroft-Karp*. בגרף דו"צ אפשר ע"י *Max-flow* או השיטה ההונגרית (בהמשך).

Vertex Covers – כיסוי קודקודים

עבור גרף G, תת-קבוצה $V \subseteq V(G)$ תיקרא כיסוי בקודקודים של G אם לכל צלע $(u, v) \in E(G)$ מתקיים: $u \in V$ או $v \in V$.



באופן טריוויאלי, הקבוצה $V(G)$ היא כיסוי בקודקודים של G.

בעיית ***Min. Vertex-Cover***: בהינתן גרף, נרצה למצוא כיסוי של G במינימום קודקודים.

עבור גרף כללי ללא הגבלות, הבעיה היא *NPC*.

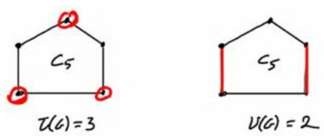
הקשר בין כיסוי קודקודים מינימום ושידוך מקסימום

גודל השידוך המקסימום בגרף מסומן $\nu(G)$ (נו, *nu*).   גודל הכיסוי המינימום מסומן $\tau(G)$ (טאו, *tau*).

באופן טריוויאלי, מתקיים ש $\tau(G) \geq \nu(G)$.

כי בהינתן שידוך מקסימום, כל כיסוי (ובפרט המינימום) יצטרך להשקיע לפחות קודקוד אחד בכל צלע בשידוך כדי לכסות אותן.

ייתכנו גרפים שבהם $\tau(G) > \nu(G)$, לדוגמה:



לכל משפחת גרפים $\mathcal{G}$ שעבורה נוכיח ש:

$$\forall G \in \mathcal{G}, \qquad \tau(G) = \nu(G)$$

יהיה לנו אלגוריתם פולינומי ל-*Min. Vertex-Cover*.

כלומר התנאי הזה מביא את הבעיה מ-NPC ל-P.

האם קיימת משפחה (מעניינת מספיק, שימושית מספיק) שמקיימת את התנאי?

König, Hall, & Frobenius

König's Theorem:

יהי G גרף דו"צ. אזי, $\tau(G)=\nu(G)$.

לדוגמה, עבור $K_{3,3}$:

Hall's Theorem:

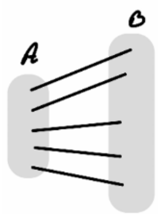
יהי $G=(A\cup B,E)$ גרף דו"צ.

אזי, ניתן לשדך את A ל-B **אמ"מ** לכל S תת-קבוצה של A, מספר השכנים של S הוא לפחות כמו מספר הקודקודים של S. נסמן:

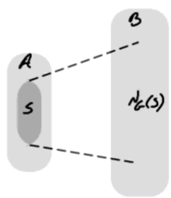
$$A \circlearrowleft B \iff \forall S\subseteq A: |N_G(S)|\geq |S|$$

צד ימין נקרא תנאי הול (*Hall condition*).

$A \circlearrowleft B$ הכוונה ש-A משתדך לתוך B: כל הצלעות שיש בין A ל-B מהוות שידוך:



תנאי הול: $\forall S\subseteq A: |N_G(S)|\geq |S|$:



Forbenius' Theorem:

יהי $G=(A\cup B,E)$ גרף דו"צ.

אזי, קיים ב-G שידוך מושלם **אמ"מ** B,A שווים בגודלם וגם אחד מהם מקיים את תנאי הול.

טענה: 3 המשפטים שקולים.

הטענה $Forbenius \rightarrow Hall$ היא טריוויאלית, כי $Forbenius$ הוא מקרה פרטי של $Hall$. (זה בעצם משפט החתונה של הול).

נוכיח ש *König* → *Forbenius*. נניח שמשפט $Forbenius$ נכון ונוכיח שמשפט $König$ נכון.

יהי גרף דו-צדדי $G=(A\cup B,E)$. ע"פ ההנחה, קיים ב-G שידוך מושלם **אמ"מ** B,A שווים בגודלם וגם אחד מהם מקיים את תנאי הול.

נרצה להוכיח ש $\tau(G)=\nu(G)$.

באופן טריוויאלי מתקיים $\tau(G)\geq\nu(G)$, אז מספיק להוכיח $\tau(G)\leq\nu(G)$.

נותר להוכיח ש $König \Leftrightarrow Hall$.

כיוון ראשון: נניח $Hall$ ונוכיח את $König$.

יהי גרף דו-צדדי $G=(A\cup B,E)$. לפי משפט הול, $A \circlearrowleft B \iff \forall S\subseteq A: |N_G(S)|\geq |S|$.

נרצה להוכיח ש $\tau(G)=\nu(G)$. באופן טריוויאלי מתקיים $\tau(G)\geq\nu(G)$, אז מספיק להוכיח $\tau(G)\leq\nu(G)$.

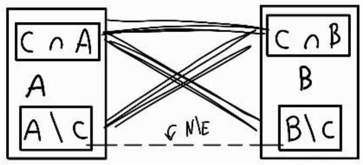
אם נראה שידוך ב-G בגודל לפחות $\tau(G)$, אז מכיוון ש $\nu(G)$ זה השידוך המקסימום, זה יגרור $\tau(G)\leq\nu(G)$.

יהי $C\subseteq V(G)$ כיסוי מינימום בקודקודים של G. כלומר $|C|=\tau(G)$.

3: König's Theorem

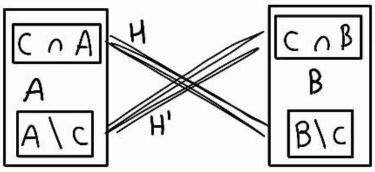
ניתן לחלק את A ל: $A \cap C,\ A \setminus C$. ואת B ל: $B \cap C,\ B \setminus C$.

נשים לב שאין צלע בין $A \setminus C$ ל-$B \setminus C$, כי C לא מכסה אותה:



נגדיר 2 תתי-גרפים של G:

$$H \coloneqq G[A \cap C, B \setminus C], \qquad H' \coloneqq G[B \cap C, A \setminus C]$$



נשים לב שמתקיים:

$$\tau(G) = |C| = |A \cap C| + |B \cap C|$$

כי כל קודקוד ב-C שייך גם ל-A או ל-B.

כלומר כדי להראות שקיים שידוך בגודל לפחות $\tau(G)$, נרצה להראות שיש שידוך מושלם בשני הגרפים H, H'.

מספיק להראות ש:

$$\nu(H') \geq |B \cap C|, \qquad \nu(H) \geq |A \cap C|$$

נתמקד ב- $A \cap C$ ו-H. (בה"כ. אותו טיעון יעבוד גם על $B \cap C$ ו- H').

אנחנו בעצם רוצים להראות שאפשר לשדך את כל $A \cap C$ לתוך $B \setminus C$.

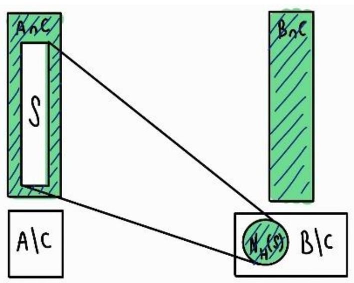
לפי משפט הול, מספיק להראות שבגרף H, הקבוצה $A \cap C$ מקיימת את תנאי הול.

נב"ש ש-$A \cap C$ לא מקיימת את הול, ולכן קיימת $S \subseteq A \cap C$ כך ש- $|N_G(S)| < |S|$.

נגדיר את הכיסוי בקודקודים הבאים:

$$X \coloneqq \big(N_H(S) \cup (B \cap C)\big) \cup \big((A \cap C) \setminus S\big)$$

כך (השטח המקושקש בציור):



נשים לב ש-X מתקבלת מ-C ע"י הורדת S והוספת $N_H(S)$ שהיא קבוצה קטנה יותר.

היא עדיין כיסוי בקודקודים, כי השינוי משפיע רק על צלעות שיוצאות מ-S, וכל הצלעות האלה מחוברות רק ל-$N_H(S)$.

לכן נקבל:

$$|X| < |C| = \tau(G)$$

וזו סתירה למינימליות של C.

כיוון שני: נניח $König$ ונוכיח את $Hall$.

יהי גרף דו-צדדי $G = (A \cup B, E)$. נרצה להוכיח ש: $A \circlearrowleft B \iff \forall S \subseteq A: |N_G(S)| \geq |S|$

הכיוון ← טריוויאלי כי אם A משתדך לתוך B, אז בהכרח $\forall S \subseteq A: |N_G(S)| \geq |S|$.

(כי אם יש קבוצה שיש לה פחות שכנים מאשר קודקודים, אז A לא משתדכת לתוך B).

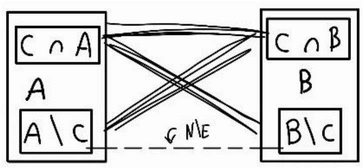
בכיוון →: נניח ש-A מקיים את תנאי הול. נוכיח ש-A משתדך לתוך B.

מספיק להוכיח ש $\tau(G) \geq |A|$. למה? כי לפי קניג, אם הגרף דו"צ אז $\tau(G) = \nu(G)$. ומכאן ינבע ש $\nu(G) \geq |A|$.

3: König's Theorem

זה יראה שקיים בגרף שידוך בגודל לפחות $|A|$, וזה מוכיח ש-A משתדכת לתוך B (כי כל צלע מ-A עוברת רק ל-B).

יהי $C \subseteq V(G)$ כיסוי מינימום בקודקודים של G.



נרצה להוכיח ש $|A| \geq |C|$.

נתבונן ב- $N_G(A \setminus C)$. היא מוכלת ב- $B \cap C$ כי אין צלעות מ- $A \setminus C$ לשום מקום אחר. אז $|N_G(A \setminus C)| \geq |B \cap C|$.

נזכור כי לפי ההנחה, A מקיימת את תנאי הול ולכן $|A \setminus C| \geq |N_G(A \setminus C)|$.

אז נקבל ש $|A \setminus C| \geq |N_G(A \setminus C)| \geq |B \cap C|$. ובסה"כ,

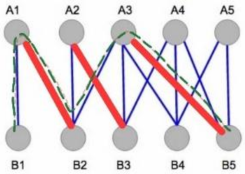
$$\tau(G) = |C| = |A \cap C| + |B \cap C| \geq |A \cap C| + |A \setminus C| = |A|$$

כנדרש.

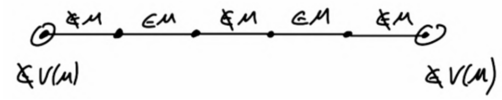
Characterizing Max Matchings

אפיון שידוך מקסימום בגרף כללי.

עבור גרף G ושידוך M בגרף, מסלול ב-G ייקרא **מסלול M-מתחלף** (*M-Alternating path*) אם צלעותיו מתחלפות בין M ל-$E(G) \setminus M$:



מסלול M-מתחלף ב-G ייקרא **M-משפר** (*M-Augmenting path*) אם קודקודי הקצה לא מרווים ע"י M (כלומר לא שייכים ל-$V(M)$).



בפרט, צלעות הקצה לא יהיו שייכות ל-M.

זה נקרא מסלול משפר כי נוכל לזרוק מהשידוך את הצלעות במסלול ששייכות ל-M ולהוסיף למסלול ולהוסיף לשידוך את הצלעות במסלול שאינן ב-M,

ובכך להגדיל את השידוך M כולו (כי במסלול יש צלע אחת יותר שלא מהשידוך מאשר צלעות מהשידוך).

לפיכך, אם קיים בגרף מסלול M משפר, אז M הוא לא שידוך מקסימלי ($|M| < \nu(G)$).

משפט ברג' יוכיח שזה אמ"מ.

תזכורת – בהינתן 2 קבוצות X,Y, **ההפרש סימטרי** שלהן מוגדר: $X \Delta Y \coloneqq (X \setminus Y) \cup (Y \setminus X)$.

כל מה שנמצא רק באחת הקבוצות ולא בשתיהן.

הפרש סימטרי של שידוכים: בהינתן 2 שידוכים M,N בגרף G, ההפרש הסימטרי שלהם הוא הגרף:

$$M \Delta N \coloneqq \big(V(G), (M \setminus N) \cup (N \setminus M)\big)$$

כלומר, תת-גרף פורש עם צלעות מההפרש הסימטרי.

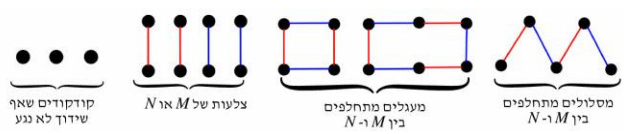
איך נראה הגרף $M\Delta N$?

הדרגה המקסימום בגרף היא לכל היותר 2. למה?

הדרך היחידה לקבל קודקוד עם דרגה 3 ומעלה, היא אם מאחד השידוכים הגיעו 2 צלעות לפחות שנגעו בקודקוד הזה – בסתירה להגדרת שידוך.

רכיבי הקשירות של הגרף הם רק מהצורה:

1. קודקוד בודד, 2. מסלולים (כולל צלע בודדת), 3. מעגלים.



אם $M\Delta N$ מכיל מעגל, הוא בהכרח באורך זוגי (כי הוא חייב להיות מתחלף).

כל המסלולים חייבים להיות מסלולים מתחלפים בין M ל-N. גם מסלול באורך 1 נחשב מתחלף.

Berg's Theorem

משפט ברג: יהי G גרף, ויהי $M \subseteq E(G)$ שידוך ב-G. אזי, לא קיים ב-G מסלול M-משפר **אמ"מ** M הוא שידוך מקסימום.

נוכיח את השלילה של המשפט, כלומר: קיים ב-G מסלול משפר אמ"מ M הוא לא שידוך מקסימום.

כיוון ראשון: אם קיים מסלול משפר, נוכל להגדיל את השידוך אז הוא לא מקסימום. מש"ל.

כיוון שני: נניח ש-M אינו שידוך מקסימום, ויהי N שידוך מקסימום (כלומר $|N| = \nu(G) > |M|$).

נתבונן ברכיבים של הגרף $N\Delta M$:

מעגלים, או קודקודים בודדים, או מסלול עם מספר שווה של צלעות מ-M ו-N – כולם תורמים מספר שווה ל-M ו-N.

בסה"כ יש יותר צלעות מ-N מאשר מ-M. כלומר חייבים להיות רכיב (לפחות אחד) שמכיל יותר צלעות של N מאשר M. (יכול להיות גם צלע בודדת).

כל רכיב כזה הוא מסלול מתחלף. כדי שיהיו בו יותר צלעות של N מאשר של M, הוא חייב להתחיל ולהסתיים בצלע מ-N. כלומר הוא מסלול M-משפר.

The Hungarian Method

אלגוריתם למציאת שידוכים מקסימום בגרפים דו"צ. הוא מסתמך על משפט ברג:

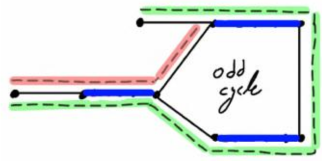
האלגוריתם מוגדר על גרף כללי, בהמשך נתאר את הקשר לגרף דו"צ.

קלט: גרף G. פלט: שידוך מקסימום M ב-G.

1. נאתחל $M = \emptyset$.
2. כל עוד קיים ב-G מסלול M-משפר:
   נמצא מסלול M-משפר ונשפר את M לאורך P.
3. נחזיר את M.

אז הבעיה שלנו מצטמצמת: בהינתן גרף G ושידוך M, נרצה לבדוק האם קיים מסלול M-משפר ב-G. ואם קיים, למצוא אותו.

זו לא בעיה פשוטה. לדוגמה, עבור השידוך הצבוע בכחול:

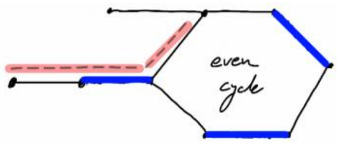


המסלול הירוק (הארוך יותר) הוא מסלול M-משפר.

המסלול האדום הוא מסלול M-מתחלף שצלעות הקצה שלו לא ב-M אבל הוא לא מסלול M-משפר, כי קודקוד הקצה שלו מרווה ע"י M.

איך נדע, אלגוריתמית, להימנע מהמסלול האדום?

אפשר לראות שהבעיה נפתרת כאשר המעגל זוגי, ואז המסלול האדום הוא טוב:



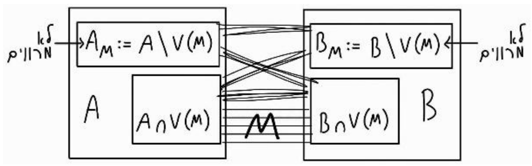
ננסה לפתור את הבעיה בגרף דו"צ.

Finding Augmenting Paths in Bipartite Graphs

הפתרון: בהינתן גרף דו"צ $G = (A \cup B, E)$ ושידוך $M \subseteq E(G)$, נגדיר:

$$A_M \coloneqq A \setminus V(M), \qquad B_M \coloneqq B \setminus V(M)$$

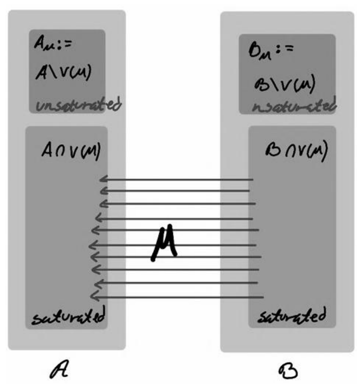
כלומר קבוצות הקודקודים של A ו-B (בהתאמה) שאינן מרווים (נוגעים) בשידוך M. לכן ניתן לחלק את הגרף בצורה הבאה:



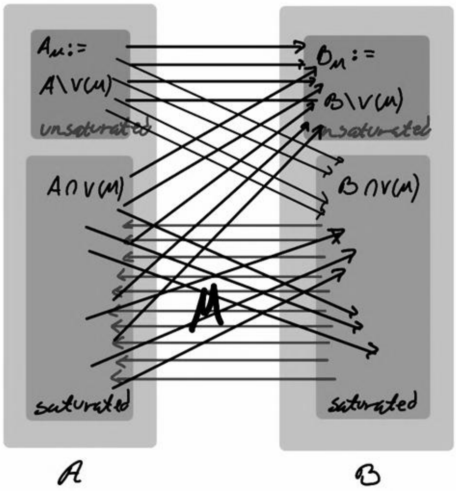
נשים לב שאם יש צלע בין A_M ל-B_M אז היא מסלול M-משפר ונוכל להוסיף אותה לשידוך.

אבל לא נוכל להסתמך על כך שיהיו צלעות כאלה. איך נזהה מסלול M-משפר?

נבנה גרף עזר, גרף מכוון באופן הבא: כל צלע מהשידוך M נהפוך לצלע מכוונת מ-B ל-A:



וכל צלע שאינה בשידוך נהפוך לצלע מכוונת מ-A ל-B:



נשים לב שיכולות להיות צלעות בין $A\cap V(M)$ ל- $B\cap V(M)$ שאינן בשידוך M, ולכן גם אותן נכוון מ-A ל-B.

אבחנה: כל מסלול מ-A_M ל-B_M הוא מסלול M-משפר.

הוכחה: יהי P מסלול מכוון בין A_M ל-B_M. מספיק שנוכיח כי P הוא מסלול M-מתחלף וקודקודי הקצה שלו אינם מרווים ע"י M.

טריוויאלי שקודקודי הקצה אינם מרווים ע"י M, כי הם שייכים ל-A_M ו-B_M.

נראה שהמסלול הוא M-מתחלף: נשים לב ש-P חייב להיות באורך אי-זוגי לפי מבנה הגרף – או 1, או 3.

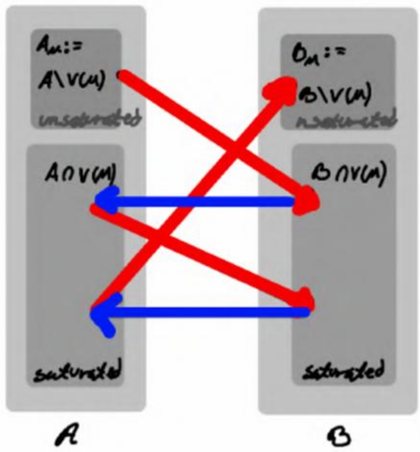
אם הוא באורך 1, אז הטענה טריוויאלית.

אחרת, המסלול באורך לפחות 3.

אזי P מתחיל בצלע מכוונת שאינה בשידוך מ-A_M ל-$B\cap V(M)$, וממשיך עם צלע בשידוך מ-$B\cap V(M)$ ל- $A\cap V(M)$.

ומכאן יכול להיות מסלול מתחלף הלוך חזור בין $A\cap V(M)$ ל-$B\cap V(M)$ שמסתיים ב-$A\cap V(M)$, כלומר הצלע האחרונה הייתה בשידוך.

לבסוף, המסלול יסתיים בצלע שאינה בשידוך מ-$A\cap V(M)$ ל-B_M:



האלגוריתם:

<u>קלט</u>: גרף דו"צ $G=(A\cup B,E)$.   <u>פלט</u>: שידוך מקסימום M ב-G.

1. נחלק את A ו-B לקודקודים המרווים ע"י M (כלומר $A\cap V(M),\ B\cap V(M)$), ולקודקודים שאינם מרווים ע"י M (A_M,B_M).
2. ניצור את הגרף המכוון המתואר לעיל.
3. כל עוד ריצת DFS או BFS מוצאה מסלול בין A_M ל-B_M, זה יהיה גם מסלול M-משפר.
   3.1. נשפר את M לאורך המסלול.
   3.2. נבנה מחדש את הגרף המכוון כמתואר לעיל.
4. נחזיר את M.

***Tutte's Theorem* – מאפיין התנהגות של זיווג מושלם בגרף כללי**

משפט *Frobenius* – בהינתן גרף דו"צ $G=(A\cup B,E)$, מתקיים:

קיים שידוך מושלם ב-G **אמ"מ** $|A|=|B|$ וגם A או B מקיימים את תנאי הול.

נשאל – מה אם G אינו גרף דו"צ? נרצה למצוא תנאי לקיום שידוך מושלם בגרפים כלליים.

הסרת קודקודים מגרף

שיטת $G-S$: יהי G גרף ותהי $S\subseteq V(G)$ תת-קבוצת קודקודים.

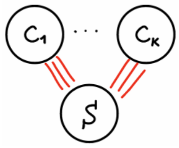
נסמן $\boldsymbol{G-S}$ את תת-הגרף המתקבל מ-G לאחר הסרת כל הקודקודים שב-S והצלעות המחוברות אליהם. הוא תת-גרף של G.

כעת נוכל להציג את G באמצעות רכיבי הקשירות מ- $G-S$ יחד עם הקבוצה S והצלעות שיוצאות מ-S.

כי הקבוצה S והצלעות שלה זה בדיוק מה שהורדנו כדי לקבל את $G-S$, אז אם נחזיר אותם נקבל את G.

כל הצלעות יהיו בין S לרכיבי הקשירות, בתוך רכיבי הקשירות עצמם, ובתוך S עצמה. בין רכיבי הקשירות אין צלעות, כי אחרת הם היו אותו רכיב.

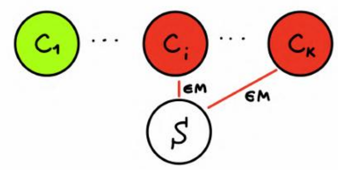
נקבל את המבנה:



בהינתן S, נחלק את רכיבי הקשירות של $G-S$ לאלו עם כמות זוגית של קודקודים ולאלו עם כמו אי זוגית של קודקודים.

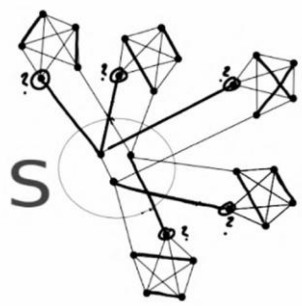
נניח שקיים שידוך מושלם ב-G. נקרא לו M.

ברכיבי הקשירות מסדר אי זוגי (השניים הימניים באיור), בהכרח יישאר לפחות קודקוד אחד ללא שידוך בתוך הרכיב ולכן הוא יתחבר ל-S:



כלומר M חייב להשקיע לפחות צלע אחת בין S לכל רכיב קשירות מסדר אי-זוגי.

לדוגמה באיור הבא, בכל אחד מהמחומשים נשאר קודקוד שחייב להיות מחובר ל-S.



נשים לב שאם יש פחות קודקודים ב-S מאשר רכיבי קשירות אי-זוגיים, אז יש קודקודים ב-S שיצטרכו להתחבר ליותר מצלע אחת.

מסקנה: אם קיימת קבוצה S כזו, אין בגרף שידוך מושלם.

Tutte's Theorem

נסמן $\boldsymbol{C_o(G)}$ את מספר רכיבי הקשירות מדרגה אי-זוגית שיש ב-G.

קיים ב-G שידוך מושלם **אמ"מ** לכל $S\subseteq V(G)$ מתקיים $C_o(G-S)\le|S|$. (צד שמאל נקרא תנאי *Tutte*).

נשים לב ש-S יכולה להיות הקבוצה הריקה ϕ.

הוכחה – **כיוון ראשון:** כבר הראנו שתנאי *Tutte* הכרחי.

כיוון שני – נוכיח שהתנאי מספיק:

5: Tutte's Theorem

גרף G ייקרא ***Factorizable*** אם יש לו שידוך מושלם. נב"ש שקיים גרף G שמקיים את תנאי *Tutte* ואינו *Factorizable*.

אבחנה 1: יהי גרף G המקיים את תנאי *Tutte* ויהי $e \notin E(G)$. אזי הגרף $G' \coloneqq G + e$ גם מקיים את תנאי *Tutte*.

נוכיח את האבחנה: נרצה להראות שלאחר הוספת e, לא קיימת $S \subseteq V(G') = V(G)$ כך ש- $C_o(G' - S) > |S|$.

תהי קבוצה S: $S \subseteq V(G') = V(G)$ (לא הוספנו קודקודים חדשים אלא רק צלעות).

מההנחה, G מקיים את תנאי *Tutte* בפרט עבור S, כלומר $C_o(G - S) \le |S|$.

נעבור על כל האפשרויות של הוספת e לגרף G ונראה שבכל המקרים, מספר רכיבי הקשירות האי-זוגיים לא עולה. אז התנאי עדיין מתקיים.

אפשרות 1 – הוספת e בין שני רכיבי קשירות מדרגה זוגית. אזי נקבל רכיב קשירות חדש מדרגה זוגית כתוצאה מאיחוד 2 רכיבי הקשירות. ולכן נקבל:

$$C_o(G' - S) = C_o(G - S) \le |S|$$

אפשרות 2 – הוספת e בין שני רכיבי קשירות מדרגה אי-זוגית.

אזי נקבל רכיב קשירות חדש מדרגה זוגית כתוצאה מאיחוד 2 רכיבי הקשירות, ונאבד 2 רכיבי קשירות מדרגה אי-זוגית:

$$C_o(G' - S) < C_o(G - S) \le |S|$$

אפשרות 3 – הוספת e בין רכיב קשירות מדרגה אי-זוגית לרכיב מדרגה זוגית.

אזי נקבל רכיב קשירות חדש מדרגה אי-זוגית כתוצאה מהאיחוד, ונאבד רכיב קשירות 1 מדרגה אי-זוגית:

$$C_o(G' - S) = C_o(G - S) \le |S|$$

אפשרות 4 – הוספת e בתוך S או אחד מרכיבי הקשירות. זה לא משפיע על הגודל של S או על מספר רכיבי הקשירות.

בסה"כ, לכל קבוצה S ולכל סוג של הוספת צלע, תנאי *Tutte* נשמר.

אבחנה 2: יהי G גרף המקיים את תנאי *Tutte*. אזי $|V(G)|$ זוגי.

הוכחה: מכיוון ש-G מקיים את תנאי *Tutte*, הוא בפרט מקיים אותו עבור $S = \phi$.

אזי, $C_o(G - \phi) \le |S| = 0$ כלומר אין ב-G רכיבי קשירות דרגה אי-זוגית, אז לא יכול להיות מספר אי-זוגי של קודקודים.

נמשיך בהוכחת משפט *Tutte*: הנחנו בשלילה שקיים גרף G המקיים את תנאי *Tutte* ואינו *Factorizable*.

נדמיין שנוסיף את כל הצלעות עד שנקבל קליקה ואז קיים שידוך מושלם. ובכל השלבים של הוספת צלע, תנאי *Tutte* ממשיך להתקיים (לפי אבחנה 1).

כלומר, יש שלב שבו יש לנו G שהוא דוגמה נגדית, וכל הוספת צלע תגרום לו להפסיק להיות דוגמה נגדית.

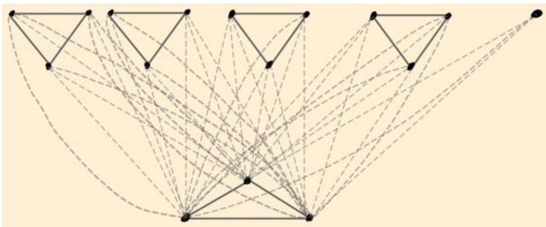
לכן, אפשר להניח כי G הינו דוגמה נגדית מקסימלית בצלעות. כלומר שאם נוסיף לה צלע, היא כבר לא תהיה דוגמה נגדית.

לפי אבחנה 2, ל-G יש מספר זוגי של קודקודים. אם G שלם, אז הוא גרף שלם עם מספר זוגי של קודקודים ויש לו שידוך מושלם. אז G לא שלם.

המטרה שלנו – להוכיח שגרף כמו G אינו קיים.

נגדיר: גרף *SNF*

גרף G שאין בו שידוך מושלם, אבל **כן** מקיים ש- $G + e$ הוא *Factorizable* לכל צלע $e \notin E(G)$, נקרא *Saturated non-factorizable graph*.

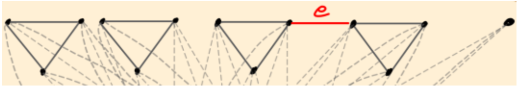


נסביר את מבנה הגרף ב<u>דוגמה</u>: בין המשולשים והקודקוד היחיד למעלה אין צלעות בכלל. בינם לבין המשולש למטה, כל הצלעות קיימות.

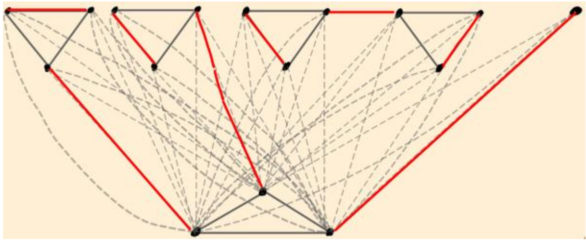
הוא לא מקיים את תנאי *Tutte*: אם ניקח את S להיות המשולש התחתון, אז מספר הרכיבים האי-זוגיים בגרף גדול מ- $|S|$.

לכן, אין בו שידוך מושלם. (כי כבר הראנו שהתנאי הכרחי לשידוך מושלם).

כל צלע שנוסיף לגרף תגרום לו להיות *Factorizable*. לדוגמה, אם נוסיף את e:



אז יש שידוך מושלם:



לגרף SNF יש מבנה מיוחד (לא נוכיח), ומקיים את הטענות הבאות: יהי G גרף SNF:

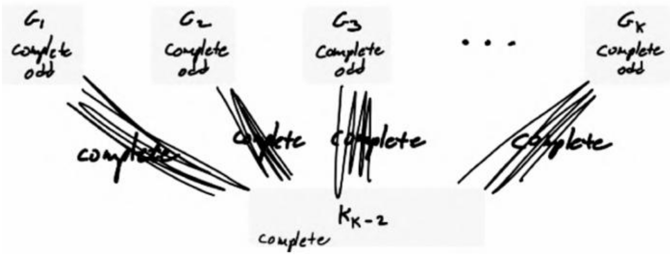
אם יש בו מספר אי-זוגי של קודקודים, הוא הגרף השלם על n קודקודים: K_n.

ל- K_n עם n אי-זוגי אין שידוך מושלם. ומתקיים: כל צלע שניתן להוסיף תגרום לזיווג מושלם (באופן ריק, כי אין צלעות כאלו).

אם יש בו מספר זוגי של קודקודים, אז יש לו את המבנה הבא:

קיימת חלוקה של הקודקודים ל-k קבוצות כך שיש עותק אחד של K_{k-2} (נקרא לו הגרף התחתון),

וכל שאר הקבוצות (הגרפים העליונים) הן גרף שלם עם מספר אי-זוגי של קודקודים. וכל הצלעות בין הגרף התחתון לעליונים, קיימות:



בפרט, גרף SNF לא מקיים את תנאי $Tutte$. למה?

ראשית, כי לכל גרף שמקיים את תנאי $Tutte$ יש מספר זוגי של קודקודים (הוכחנו). אז אם יש לו מספר אי זוגי, הוא לא מקיים את תנאי $Tutte$.

אם יש לגרף מספר זוגי של קודקודים, אז לפי המבנה שתיארנו, תנאי $Tutte$ לא מתקיים (בגלל שיש יותר רכיבים אי-זוגיים מאשר קודקודים בגרף התחתון).

נחזור להוכחת משפט $Tutte$: נוכיח שתנאי $Tutte$ הכרחי.

יהי G דוגמה נגדית מקסימלית בצלעות (מקיים את תנאי $Tutte$ ולא $factorizable$).

לפי מה שהוכחנו, הוא גרף SNF.

גרף SNF לא מקיים את תנאי $Tutte$.

סתירה.

בסה"כ, הוכחנו שתנאי $Tutte$ הוא הכרחי ומספיק לקיום שידוך מושלם בגרף.

אפיון מבנה של גרפים

הגדרה: נסמן $\boldsymbol{d_G(u,v)}$ את אורך המסלול הקצר ביותר (מספר הצלעות במסלול) שיש בין u ל-v בגרף.

בלוקים של גרפים

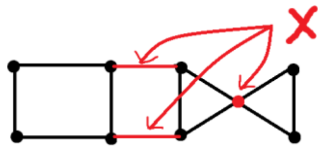
עבור גרף G, נסמן $\boldsymbol{C(G)}$ את מספר מרכיבי הקשירות של הגרף.

הגדרה: קבוצה $X \subseteq V(G) \cup E(G)$ שמקיימת:

$$C(G-X) > C(G)$$

נקראת ***disconnector*** (מנתק). X יכול להיות מורכב מקודקודים וצלעות.

למשל בגרף הבא יש מרכיב קשירות יחיד:

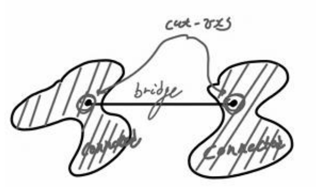


אם נוריד את הקודקוד והצלעות האדומים (הקבוצה X), יהיו 3 רכיבי קשירות. X היא *disconnector*.

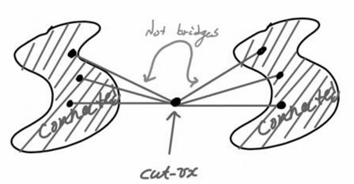
אם $X \subseteq V(G)$, היא נקראת *vertex-disconnector*. אם X בגודל 1 היא נקראת **קודקוד חתך** (*cut-vertex*).

אם $X \subseteq E(G)$, היא נקראת *edge-disconnector*. אם X בגודל 1 היא נקראת **גשר** (*bridge*).

הקצוות של גשרים הם קודקודי חתך:



מצד שני, הצלעות שמחוברות לקודקוד חתך הן לא בהכרח גשרים:



אבחנה: צלע בגרף מהווה גשר **אמ"מ** היא לא נמצאת על אף מעגל בגרף.

הוכחה: נוכיח את השלילה – צלע היא לא גשר אמ"מ היא כן על מעגל.

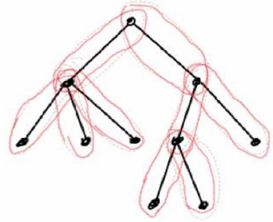
<u>כיוון ראשון</u>: תהי צלע uv על מעגל בגרף. מהגדרת מעגל יש גם מסלול $u \rightsquigarrow v$. אז אם נוריד את הצלע, עדיין אפשר לעבור בין v ל-u.

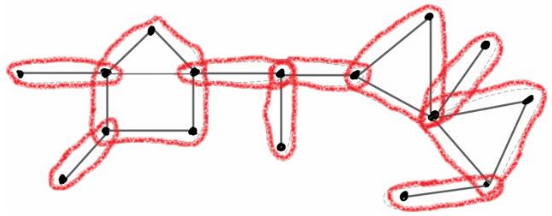
<u>כיוון שני</u>: תהי צלע uv שאינה גשר. כלומר שאפשר לנתק אותה ועדיין להגיע מ-u ל-v.

אם יש מסלול $v \rightsquigarrow u$, אז ביחד עם הצלע, יש מסלול $u \to v \rightsquigarrow u$, שזה מעגל.

תת-גרף מקסימלי של G (שאי אפשר להוסיף לו קודקודים או צלעות) שאין בו קודקודי חתך ייקרא **בלוק** של G.

דוגמאות. נשים לב שהצלעות של עץ הן בלוקים (כל אחת), כי עבור כל צלע, כל קודקוד שנוסיף יהפוך את זה לתת-גרף שיש בו קודקוד חתך.





משפט וויטני – *Whitney's Theorem*

יהי G גרף קשיר, עם $|V(G)| \geq 3$. ב-G אין קודקודי חתך **אמ"מ** כל שני קודקודים חולקים מעגל.

כיוון ראשון: נניח שכל שני קודקודים חולקים מעגל. נב"ש שיש ב-G קודקוד חתך, v.

כלומר יש לפחות 2 קודקודים שאם ננתק את v, הם יהפכו להיות ברכיבי קשירות שונים. סתירה לכך שכל שני קודקודים חולקים מעגל.

כיוון שני: נניח שאין קודקודי חתך. נוכיח באינדוקציה על $d_G(u, v)$ שכל שני קודקודים חולקים מעגל.

<u>בסיס</u>: יהיו שני קודקודים u, v שמקיימים: $d_G(u, v) = 1$. כלומר קיימת צלע $(u, v) \in E(G)$.

נוכל להניח שהיא לא גשר (כי אז הקודקודים שלה היו קודקודי חתך). כלומר היא על מעגל. כלומר המעגל מכיל את u ואת v.

<u>צעד</u>: נניח שהטענה נכונה עבור כל זוגות הקודקודים x, y שמקיימים $d_G(x, y) < k$.

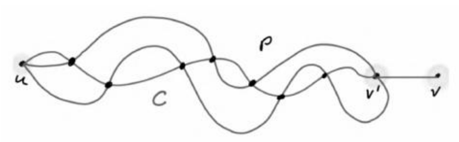
כלומר, כל שני קודקודים שהמרחק ביניהם קטן מ-k, חולקים מעגל.

נשקול זוג קודקודים u, v שמקיים $d_G(u, v) = k$. נראה שהם חולקים מעגל.

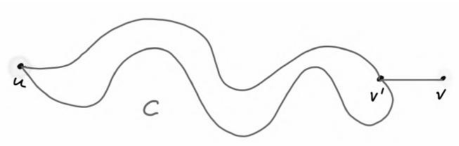
יהי P המסלול הקצר ביותר בין u ל-v (הוא באורך k). ויהי v' הקודקוד הקודם ל-v במסלול. (מכיוון ש $k > 1$, יש קודקוד כזה והוא לא v).

כלומר יש את המבנה: $v \rightarrow v' \rightsquigarrow u$. מתכונות תת-מסלול קצר, מתקיים $d_G(u, v') = k - 1$.

אז מהנ"א, u, v' חולקים מעגל, נקרא לו C.



אם $v \in V(C)$, סיימנו. נניח ש- $v \notin V(C)$. כלומר המצב הוא:



נשים לב ש-u ו-v' מחלקים את המעגל לשתי קשתות: $uCv', v'Cu$. הראשון בכיוון כלשהו – בה"כ המסלול התחתון, והשני ממשיך באותו כיוון (העליון).

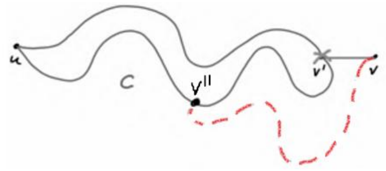
כך שאם נלך על uCv' ואז נמשיך עם $v'Cu$, לא נלך מיד על הצלע האחרונה שהיינו בה.

הנחנו שאין ב-G קודקודי חתך, אז בפרט v' הוא לא קודקוד חתך. כלומר, $G - v'$ הוא קשיר. אז ב- $G - v'$ יש אחד משני מסלולים:

אפשרות 1, המסלול: $(v, uCv' - v')$.

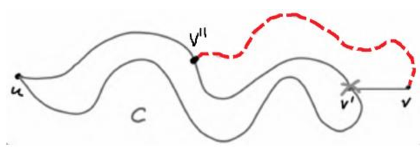
כלומר, נתחיל מ-v, ונלך עד שנפגוש את המסלול $uCv' - v'$ (מה שנשאר מהמסלול uCv' אחרי שהורדנו את v').

נקודת המפגש יכולה (אבל לא חייבת) להיות u. נסמן אותה v'':



בגרף G, המסלול הבא הוא מעגל: נלך מ-u על uCv' עד v'', נלך על P' עד v, נלך בצלע vv', ונלך על $v'Cu$.

אפשרות 2, המסלול $(v, v'Cu - v')$:

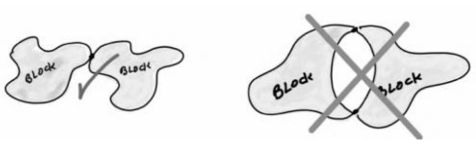


בגרף G, המסלול הבא הוא מעגל: נלך מ-u על $v'Cu$ עד v'', נלך על P' עד v, נלך בצלע vv', ונלך על uCv' (הפעם "בכיוון ההפוך").

בשני המקרים, מצאנו מעגל שמכיל את u ואת v, כנדרש.

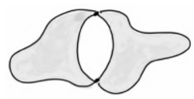
השלכות של משפט וויטני

טענה: יהי G גרף. כל שני בלוקים בגרף נפגשים בלכל היותר קודקוד אחד:



הוכחה:

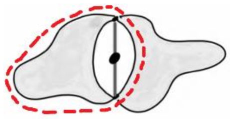
נניח (בשלילה) שיש שני קודקודי מפגש:



אם אין ביניהם צלע – נוסיף אותה:

נוסיף קודקוד באמצע הצלע:

נוכיח שבכל אחד מהצדדים אין קודקוד חתך. נתמקד בה"כ בצד שמאל:



בתוך כל בלוק מקורי אין קודקוד חתך (מהגדרת בלוק). זה כולל את שני קודקודי החיבור.

וגם הקודקוד החדש באמצע, אם נוריד אותו עדיין אפשר להגיע לשני הקודקודים שהוא מחובר אליהם.

עכשיו נפעיל את משפט וויטני על כל אחד מהצדדים. בה"כ, על צד שמאל.

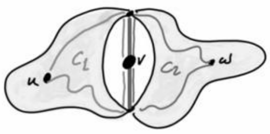
יש בו לפחות 3 קודקודים (השניים של המפגש והנוסף באמצע) והוכחנו שאין בו קודקודי חתך. אז לפי משפט וויטני, כל שני קודקודים בו חולקים מעגל.

ובפרט, שני קודקודי המפגש, או כל קודקוד בתוך הבלוק. וכמובן שבתוך הבלוק יש לפחות קודקוד אחד חוץ מקודקודי המפגש (כי אחרת אין בלוק).

אז נקרא לקודקוד הזה u, ויש מעגל שהוא חולק עם כל אחד מקודקודי המפגש.

כלומר יש מסלול מ-u לקודקוד מפגש, מהקודקוד מפגש לקודקוד מפגש השני, וחזרה ל-u.

אותו דבר מתקיים גם לצד ימין, ובסה"כ נקבל:



עכשיו אפשר להתעלם מהצלע והקודקוד שהוספנו, ועדיין יש מעגל $C_1 \cup C_2$: מ-u לקודקוד מפגש, ל-w, לקודקוד מפגש השני, ובחזרה ל-u.

כלומר כל זוג קודקודים בגרף יושבים על מעגל משותף, אז שוב לפי משפט וויטני אין בגרף קודקודי חתך. אז הוא בעצם בלוק אחד, סתירה.

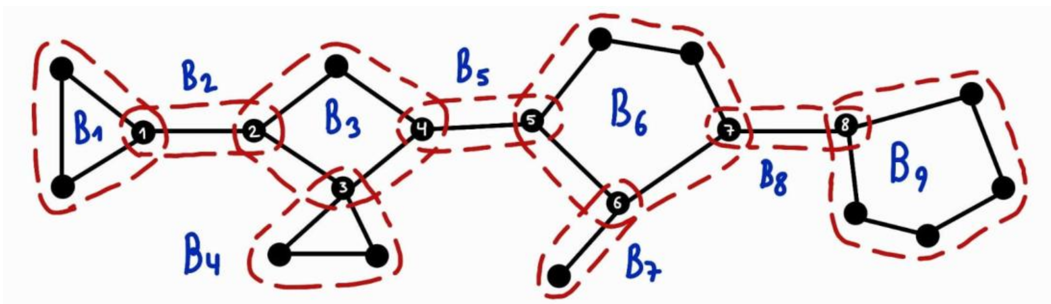
מתוך זה שהבלוקים נפגשים בלכל היותר קודקוד אחד, נובע שבלוקים לא נפגשים בצלעות בכלל.

כלומר, הם מהווים **חלוקה** של $E(G)$, ומגדירים יחס שקילות על הצלעות. כלומר, שתי צלעות יקיימו $e \sim f$ אם הן באותו בלוק.

Block Trees

בהינתן גרף G, נוכל להגדיר גרף עזר. נסמן:

$$B(G) \coloneqq \{B : B \subseteq G \text{ is a block of } G\}, \qquad C(G) \coloneqq \{v \in V(G) : v \text{ is a cut-vx of } G\}$$



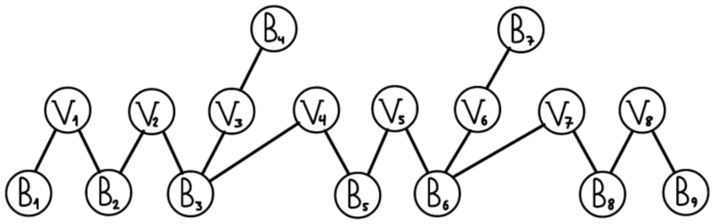
הקודקודים הממוספרים הם קודקודי חתך.

ונגדיר את $BC(G)$ להיות הגרף שקודקודיו הם $B(G) \cup C(G)$, והצלעות הן:

$$\{\{B,v\} : B \in B(G), v \in C(G), v \in B\}$$

כלומר, נחבר צלעות רק בין בלוקים וקודקודי חתך שמחוברים אליהם (אז הגרף דו"צ, קבוצה אחת זה הבלוקים והשנייה היא קודקודי החתך).

נבנה את $BC(G)$:



משפט: אם G קשיר, אז $BC(G)$ הוא עץ.

הוכחה: ראשית, נוכיח שבגרף בלוקים אין מעגלים:

כל מעגל ב-G מוכל בבלוק יחיד. כי בתת-גרף שהוא המעגל, אין קודקודי חתך.

יהי גרף G, ויהי $BC(G)$ גרף הבלוקים שלו. נב"ש שקיים מעגל ב- $BC(G)$.

מהגדרת הבנייה, $BC(G)$ הוא דו"צ. אז כל מעגל שקיים יהיה באורך לפחות 4, והמבנה: $(B_1, v_1, B_2, v_2, \dots, B_i, v_i, B_1)$.

כאשר B_i, v_i יכולים להיות v_2 (במקרה הזה לא נכתוב את ה-"$\dots v_i$").

אז ב-G, יש מסלול מ-v_1 ל-v_2 (דרך B_2) ומסלול מ-v_2 עד v_i (דרך כל הקודקודים והבלוקים שמחוברים בדרך), ומסלול מ-v_i ל-v_1.

כלומר ב-G, v_1, v_i היו על מעגל משותף. אז הם צריכים להיות על אותו בלוק. סתירה.

אז $BC(G)$ הוא גרף חסר מעגלים. נוכיח שאם G קשיר, אז $BC(G)$ קשיר:

אם G קשיר, זה אומר שאפשר להגיע מכל קודקוד לכל קודקוד אחר. כלומר מכל בלוק לכל בלוק אחר.

בין שני בלוקים מחבר בהכרח קודקוד חתך יחיד.

אז אם אפשר להגיע מבלוק אחד לבלוק אחר, יש ביניהם קודקוד חתך (או רצף של בלוקים וקודקודי חתך).

אז יהיה מסלול ביניהם ב-$BC(G)$.

וכל קודקוד חתך חייב להיות מחובר לבלוק, אז אם הבלוקים קשירים גם קודקודי החתך קשירים.

אז $BC(G)$ הוא גרף קשיר חסר מעגלים, שזה עץ.

Characterizing Local Connectivity

עבור גרף G, נגדיר את $\boldsymbol{\kappa(G)}$ (קאפא, *kappa*).

נתחיל עם ההגדרה לפי דיסטל (*Diestel*). יהיו בה חסרונות מסויימים. נתאר הגדרה לפי בונדי ומרטי (*Bondy & Murty*).

משפט מנגר יגיד שהן אותה הגדרה, ואז נשלב ביניהן.

ההגדרה של דיסטל

בהינתן $0 \leq k \in \mathbb{Z}$, וגרף G שמקיים $\nu(G) > k$. (זה מספר הקודקודים בגרף, לא גודל השידוך המקסימום $(\nu(G)$.

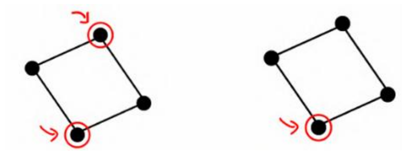
אם לכל $X \subseteq V(G)$ כך ש $|X| \leq k-1$, תת-הגרף $G - X$ הוא קשיר, אז G ייקרא ***k-connected***.

הסבר אינטואיטיבי א: אם אפשר להוריד כל קבוצה של עד $k-1$ קודקודים והגרף עדיין קשיר.

הסבר אינטואיטיבי ב: אם אין שני קודקודים שניתן לנתק אותם אחד מהשני על ידי הסרה של **פחות מ-**k קודקודים בגרף.

הסבר אינטואיטיבי ג: כדי לנתק שני קודקודים אחד מהשני, נצטרך להסיר לפחות k קודקודים.

לדוגמה, ניתוק כל קודקוד יחיד משאיר את הגרף קשיר. אבל אפשר לנתק אם מסירים שני קודקודים. הגרף הוא *2-connected*:



מה שמפריע לנו בהגדרה של דיסטל, זה החלק שבו מגדירים את G להיות עם $\nu(G) > k$. למה דורשים את זה?

אם אין את הדרישה הזו, אז גרף עם $\nu(G) \leq k-1$ קודקודים יהיה k-קשיר. כי אפשר פשוט להסיר את כל הגרף. לכן דורשים $\nu(G) > k$.

מצד שני, נשקול את K_1 ("הגרף השלם" על קודקוד יחיד). הרבה פעמים נרצה שהוא ייחשב גרף קשיר. אבל בהגדרה של דיסטל, הוא לא 1-קשיר.

את זה נתקן בהמשך. בנתיים, נגדיר את $\kappa(G)$.

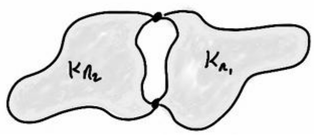
הגדרה: ה-$0 \leq k \in \mathbb{Z}$ הגדול ביותר שעבורו G הוא k-קשיר, יסומן $\boldsymbol{\kappa(G)}$. זה ה-*vertex-connectivity* של G.

כל גרף שאינו קשיר, מקיים $\kappa(G) = 0$ (כי מספיק להוריד 0 קודקודים כדי שיהיה לא קשיר).

כל גרף קשיר (חוץ מ-K_1) הם *1-connected*, כלומר מקיימים $\kappa(G) \geq 1$. גם $\kappa(K_1) = 0$.

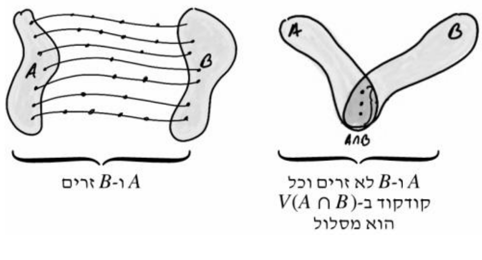
נשקול מעגלים. בגלל שמדובר בגרפים פשוטים, לכולם יש לפחות 3 קודקודים. כל מעגל הוא 2-קשיר. כלומר, לכל $3 \leq \ell \in \mathbb{N}$, מתקיים $\kappa(C_\ell) = 2$.

עבור K_{n_1}, k_{n_2} שמחוברים בשני קודקודים, יש $\kappa(G) = 2$:



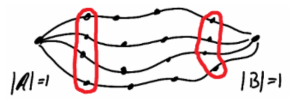
לכל $r \in \mathbb{N}$, $\kappa(K_r) = r - 1$. כי אפשר להוריד $r-2$ קודקודים לפני שהגרף הופך ל-K_1, שהוא לא 1-קשיר.

הגדרה: יהי גרף G, ויהיו $A, B \subseteq V(G)$. נסמן $\varrho_G(A, B)$ את מספר המסלולים **הזרים בקודקודים** מ-A ל-B.



אם $|A| = 1$ או $|B| = 1$, נספור את מספר המסלולים הזרים **פנימית** בקודקודים (כלומר לא נתחשב בקצוות):

7: Menger's Theorem



ההגדרה של בונדי ומרטי

גרף G ייקרא *k-connected* אם $\varrho_G(u,v) \geq k$ לכל שני קודקודים $u \neq v$ **שונים.**

עם ההגדרה הזו, כל גרף קשיר יהיה 1-קשיר, ו-K_1 יהיה קשיר (כי התנאי מתקיים באופן ריק).

נעדכן את ההגדרה לפי דיסטל:

בהינתן $0 \leq k \in \mathbb{Z}$, וגרף G שמקיים $v(G) > k > 1$.

אם לכל $X \subseteq V(G)$ כך ש $|X| \leq k-1$, תת-הגרף $G - X$ הוא קשיר, אז G ייקרא k-קשיר.

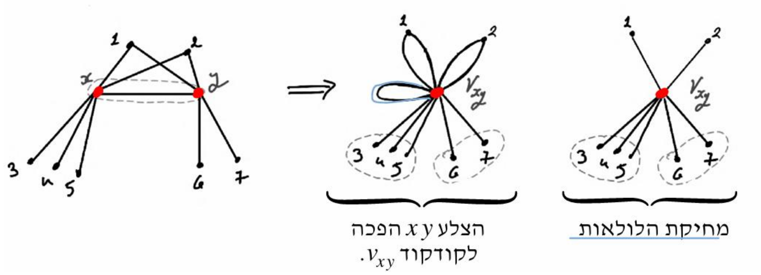
אם $v(G) = 1$, אז G הוא 1-קשיר.

נציג את הרעיון של **כיווץ צלעות** (*edge contractions*).

נתבונן בצלע xy בגרף. כיווץ x ו-y "מחבר" אותם, ומייצר קודקוד חדש V_{xy}. נקבל מזה מולטי-גרף (*multigraph*):

- כל קודקוד שהיה שכן רק של x או רק של y, יהיה מחובר ל-V_{xy}.
- כל קודקוד שהיה מחובר ל-x וגם ל-y, יהיה מחובר בצלע כפולה ל-V_{xy}.
- הצלע xy הופכת להיות לולאה על V_{xy}.

אם נרצה לעבוד עם גרפים פשוטים: נסיר את הלולאה, וכל צלע כפולה נהפוך לצלע יחידה. וזה מחזיר אותנו לגרף רגיל.



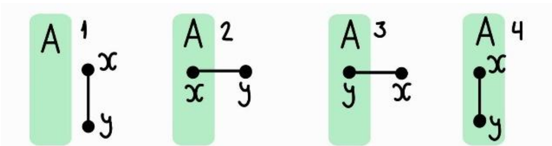
בהינתן צלע $e = \{x,y\} \in E(G)$, נגדיר G/e (כמו חלוקה, *quotient*) את הגרף הנוצר ע"י כיווץ e (אחרי החזרה לגרף פשוט).

נניח שיש לנו 2 קבוצות קודקודים, $A, B \subseteq V(G)$.

בגרף G/e, יכול להיות שהשפענו על הקבוצות A או B.

נגדיר את $A_e \subseteq V(G/e)$ כך:

1. אם $\{x,y\} \cap A = \phi$ (הצלע לא קשורה ל-A), נגדיר: $A_e := A$.
2. אם $x \in A, y \notin A$ (רק x שייכת ל-A), נגדיר: $A_e := (A \setminus \{x\}) \cup \{V_e\}$. כלומר נסיר את הקודקוד שבפנים ונוסיף את החדש.
3. אם $y \in A, x \notin A$ (רק y שייכת ל-A), נגדיר: $A_e := (A \setminus \{y\}) \cup \{V_e\}$.
4. אם $\{x,y\} \subseteq A$ (שניהם בתוך A), נגדיר: $A_e := (A \setminus \{x,y\}) \cup \{V_e\}$. כלומר נסיר את שני הקודקודים שבפנים ונוסיף את החדש.



ונגדיר את B_e באותה צורה.

משפט מנגר

יהי גרף G, ויהיו $A, B \subseteq V(G)$ לא ריקות. נסמן $\kappa_G(A, B)$ להיות הגודל של החתך-קודקודים המינימום בין A ל-B.

כלומר, המספר הקטן ביותר של קודקודים שנצטרך להסיר כדי להפריד בין A ל-B.

אזי, $\kappa_G(A, B) = \varrho_G(A, B)$. זה בעצם משפט *min-cut-max-flow*.

הכיוון הראשון פשוט: מתקיים $\kappa_G(A, B) \geq \varrho_G(A, B)$, כי על כל מסלול זר בקודקודים, נצטרך להשקיע לפחות קודקוד אחד כדי לנתק אותם.

נוכיח את הכיוון השני: במקום $\varrho_G(A, B),\ \ \kappa_G(A, B)$, נרשום κ, ϱ. אנחנו רוצים להוכיח $\kappa \leq \varrho$.

נוכיח באינדוקציה על $e(G)$.

<u>בסיס</u>: עבור $e(G) = 0$, אז $\varrho = |A \cap B|$ לפי הגדרה, וזה גם מספר הקודקודים שצריך להסיר כדי להפריד בין A ל-B.

<u>צעד</u>: נניח ש $e(G) > 0$, ותהי $e = xy \in E(G)$.

נניח שבגרף G/e יש κ מסלולים זרים בקודקודים בין A_e ל- B_e.

כל מסלול שלא מכיל את V_e, קיים גם ב-G.

אם אף מסלול לא מכיל את V_e, סיימנו. כי זה מראה שיש κ מסלולים זרים בקודקודים בתוך G.

נניח שיש מסלול שהכיל את e (יכול להיות רק אחד, כי המסלולים זרים בקודקודים ובפרט זרים בצלעות).

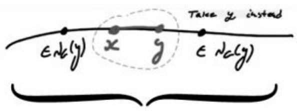
"נפתח" את e חזרה ל-x ו-y. יש 4 מצבים שהמסלול יכול להיות בהם אחרי הפתיחה, מבחינת הסדר של x, y, ושני הקודקודים הסמוכים אליהם:

אם x מופיע לפני y במסלול, והקודקוד לפני x היה שכן של x ב-G, והקודקוד אחרי y היה שכן של y ב-G,

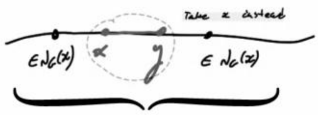
או ש-y מופיע לפני x במסלול, והקודקוד לפני y היה שכן של y ב-G, והקודקוד אחרי x היה שכן של x ב-G,

אז המסלול בסדר, אנחנו יודעים שהמסלול הזה קיים גם ב-G.

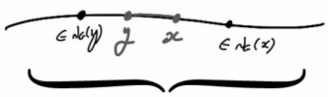
אם יש שכנים של y לפני ואחרי, אז פשוט ניקח רק את y, וזה מסלול שקיים גם ב-G. ובאופן דומה אם יש רק שכנים של x.



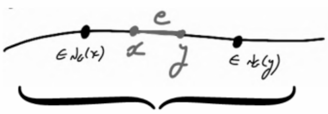
מקרה ד'



מקרה ג'



מקרה ב'



מקרה א'

אפשר גם להסתכל על זה כך: נשאל (לפני הפתיחה), מה יש לפני ואחרי הקודקוד המכווץ במסלול.

ואז לפי המצב נחליט מה לעשות עם הצלע אחרי שנפתח אותה: או שנשתמש בה, או שנשתמש רק באחד הקודקודים.

בכל מקרה, אם בגרף G/e יש κ מסלולים זרים בקודקודים בין A_e ל- B_e, סיימנו.

עכשיו, נניח שבגרף G/e אין κ מסלולים זרים בקודקודים בין A_e ל- B_e.

ניזכר בהנחת האינדוקציה: $\kappa = \varrho$ עבור גרף כלשהו עם פחות צלעות מאשר $e(G)$. בפרט, הגרף G/e.

כלומר, לפי ההנחה שלנו $\kappa_{G/e}(A_e, B_e) < \kappa$.

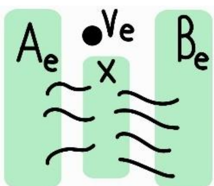
נסתכל על החתכים המינימומים שמפרידים בין A_e ל- B_e ב- G/e:

כל חתך בין A_e ל- B_e חייב להכיל את V_e. כי אחרת, יש ב-G חתך מפריד בין A ל-B (ב-G) שהוא קטן מ-κ.

וזה לא אפשרי כי ההנחה שלנו היא ש- $\kappa_G(A, B) = \kappa$.

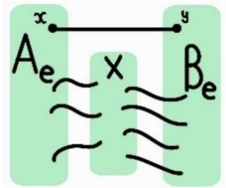
למה אם יש חתך בין A_e ל- B_e שלא מכיל את V_e יש ב-G חתך מפריד בין A ל-B (ב-G) שהוא קטן מ-κ?

ניקח חתך כזה, X, ב-G/e. כל הקודקודים ב-X קיימים גם ב-G:

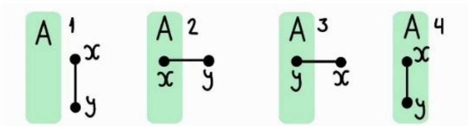


לפי הנ"א, $|X| < \kappa$. אם X מפרידה בין A ל-B ב-G, אז יש חתך מפריד בגודל קטן מ-κ. אז נניח שהיא לא מפרידה.

איך זה יכול לקרות? רק אם הצלע xy המקורית חיברה בין A ל-B ב-G (כי זה ההבדל היחיד בין G ל- G/e):



אבל נשים לב: ניזכר ב-4 המצבים ש-xy יכולה להיות ביחס ל-A,B:



אם הצלע היא מ-A ל-B, זה מצב 2 עבור A ומצב 3 עבור B. כלומר יתקיים $A_e \coloneqq (A \setminus \{x\}) \cup \{V_e\}, B_e \coloneqq (B \setminus \{y\}) \cup \{V_e\}$.

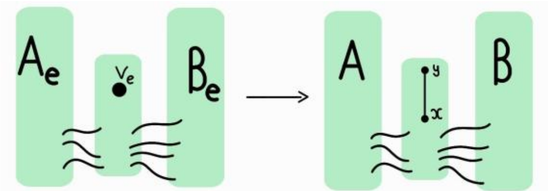
אז $V_e \in A_e \cap B_e$, כלומר היא חייבת להיות חלק מכל חתך ביניהם, ובפרט חלק מ-X. סתירה.

מזה נובע, שכל חתך בין A_e ל- B_e ב- G/e הוא בגודל $\kappa - 1$.

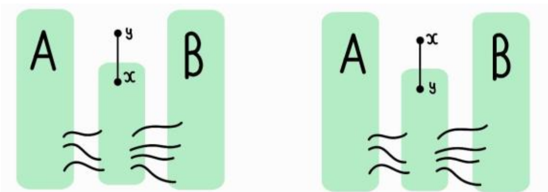
כי הוא מכיל את V_e, אז גם אחרי הפתיחה אפשר להוסיף לו רק קודקוד אחד. אז אם הוא היה קטן יותר, לא היינו יכולים להגיע ל-$\kappa_G(A,B) = \kappa$.

וניזכר שאנחנו בהנחה ש $\kappa_{G/e}(A_e, B_e) < \kappa$. אז חייב להיות $\kappa_{G/e}(A_e, B_e) = \kappa - 1$.

אזי אנחנו יודעים ש-X בגודל $\kappa - 1$, מכילה את V_e, ופורשת את xy – כלומר, כשנפתח את e, X תכיל את x ואת y, ככה:



ולא ככה:



כי אם לא נכיל את x ואת y, אז בפתיחה, "איבדנו" את V_e ו"הרווחנו" את x או y, אבל זה 1 תמורת 1.

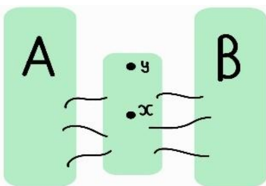
אז החתך נשאר אותו גודל, $\kappa_G(A,B) = \kappa - 1$. וזה סתירה להנחת העבודה שלנו ש $\kappa_G(A,B) = \kappa$.

אז ב-G, יש את החתך שלנו X, הוא בגודל κ, והוא פורש את e. (יכול להיות ש $X = A$ או B, זה לא משנה).

נתבונן בגרף $G - e$ (נוריד את הצלע, הקודקודים נשארים). גודל החתך המינימום בין A ל-X הוא לפחות κ, וגם בין X ל-B:

$$\kappa_{G-e}(X,A) \geq \kappa, \qquad \kappa_{G-e}(X,B) \geq \kappa$$

כי אם יש חתך קטן יותר באחד מהם, זה מהווה חתך קטן יותר בין A ל-B:



אז מהנ"א, בגלל שב- $G - e$ יש פחות צלעות, נוכל לומר שבין A ל-X יש κ מסלולים, זרים בקודקודים. וכנ"ל בין B ל-X.

ואנחנו יודעים שכל מסלול בין A ל-B חייב לעבור בחתך (אחרת הוא לא היה חתך).

בסה"כ יש לנו κ מסלולים בין A ל-B, כנדרש.

סיכום ההוכחה

נגדיר $\kappa := \kappa_G(A,B), \varrho := \varrho_G(A,B)$. מספיק להוכיח ש $\varrho \ge \kappa$ (שבין A ל-B יש לפחות κ מסלולים זרים בקודקודים).

נוכיח באינדוקציה על $e(G)$. עבור $e(G) = 0$, אז $\varrho = |A \cap B| = \kappa$.

נניח ש- $e(G) > 0$, הנחת האינדוקציה שלנו היא ש $\kappa = \varrho$ עבור גרף שיש לו פחות קודקודים מאשר $e(G)$.

נבדוק מה קורה אם מכווצים את הצלע e ומנסים להפעיל את הנ"א על G/e.

(אחרי שנבין את החתכים של G/e, נגיע למסקנה שיש חתך כלשהו בין A ל-B ב-G, שפורש את הצלע המכווצת.)

לפני כן, בגרף G/e, הגדרנו את A_e, B_e, לפי היחס לקודקוד החדש V_e.

בשלב הראשון, אמרנו ש**אם** יש ב-G/e, κ מסלולים זרים בקודקודים בין A_e, B_e, סיימנו. כי כל מסלול שלא מכיל את V_e, קיים גם ב-G.

וזה נותן לפחות $\kappa - 1$ מסלולים. ואם יש מסלול עם V_e, הראנו איך ע"י פתיחת הכיווץ נוכל לשחזר מסלול בין A ל-B שגם זר בקודקודים.

אז נבדוק את המקרה שב- G/e אין κ מסלולים. אז לפי הנ"א, גם גודל החתך קטן ממש מ-κ. כי מהנ"א, $\varrho = \kappa$.

הראנו שכל חתך מינימום בין A_e, B_e מכיל את V_e, ושהוא חייב להיות בגודל $\kappa - 1$, ושהוא פורש את xy.

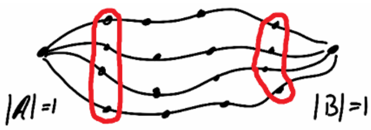
עכשיו, סיימנו לעבוד עם G/e ועוברים ל- $G - e$, ומפעילים את האינדוקציה ע"י פעולה אחרת על הגרף.

והראנו שלא צריך את xy כדי לבנות κ מסלולים בין A ל-B.

השלכות של משפט מנגר

מסקנה: בגרף G כלשהו, $\kappa_G = \min\{\varrho_G(u,v) : u,v \in V(G), u \neq v\}$.

זה בעצם מה שאמרנו בהתחלה, שאם $|A| = |B| = 1$, אפשר להתייחס לשכנים שלהם במסלולים הזרים מ-A ל-B.



מספר המסלולים בין הקודקודים הוא מספר הקודקודים שצריך להסיר כדי לנתק אותם.

The Extension Lemma

יהי G גרף k-קשיר. הגרף H מתקבל ע"י הוספת קודקוד w וחיבורו ל-k קודקודים ב-G, שרירותית.
אזי, H הוא k-קשיר.

הוכחה: נשקול קבוצת קודקודים $S \subseteq V(H)$ כלשהי שמקיימת $|S| = k - 1$. יש שני מקרים:

אם $w \in S$, אז הסרת S מ-H מסירה את w (אז הוא לא צריך להיות קשיר) ומסירה $k - 2$ קודקודים מקוריים מ-G. הגרף נשאר קשיר כי G היה k-קשיר.

אם $w \notin S$, אז הסרת S מסירה $k - 1$ קודקודים שהיו ב-G. G היה k-קשיר אז כל הקודקודים המקוריים עדיין קשירים, ו-w היה מחובר ל-k קודקודים.

אז אפילו אם הוא היה מחובר לכל S, עדיין נשאר לו קודקוד אחד שהוא מחובר אליו.

דרך שנייה:

מכיוון ש-G הוא k-קשיר, מתקיים $\kappa(G) \ge k$. לכן, לכל שני קודקודים $u \neq v$, מתקיים $\varrho_G(u,v) \ge k$.

בגרף H, הטענה מתקיימת לכל שני קודקודים שהם לא w, כי רק הוספנו צלעות. אז אם היו k' מסלולים מ-u ל-v ב-G, גם ב-H יש.

אז מספיק להוכיח שהטענה מתקיימת עבור v כלשהו עם w. כלומר שלכל $v \in V(G)$, מתקיים $\varrho_H(v,w) \ge k$.

ניזכר ש-w מחובר ל-k קודקודים.

אם v הוא אחד מהם, אז יש מסלול vw, שהוא זר בקודקודים פנימיים לכל מסלול שקיים ב-G. (כי אין לו קודקודים פנימיים).

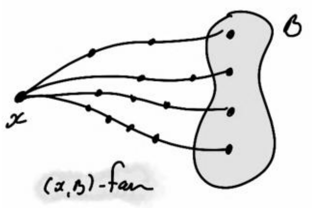
ו-w מחובר ל- $k - 1$ קודקודים אחרים. ומכל קודקוד כזה יש k מסלולים ל-v, כולם קיימים ב-G וזרים בקודקודים פנימיים.

אם v לא אחד מהם, אז w מחובר ל-k קודקודים אחרים, ומכל קודקוד כזה יש k מסלולים ל-v, כולם קיימים ב-G וזרים בקודקודים פנימיים.

בשני המקרים, קיבלנו שיש k מסלולים זרים בקודקודים פנימיים בין w ו-v, כנדרש.

למת המניפה

בהינתן קודקוד $x \in V(G)$, וקבוצת קודקודים $B \subseteq V(G)$ כך ש- $x \notin B$, נסמן $(x, B)\text{-}fan$ את הגרף:



קבוצת מסלולים זרים בקודקודים (חוץ מ-x) שמתחילים ב-x ומסתיימים בקודקוד כלשהו ב-B.

טענה: בהינתן G גרף k-קשיר, $x \in V(G), B \subseteq V(G) \setminus \{x\}$. (קודקוד בגרף וקבוצת קודקודים שלא מכילה אותו).

אז ל-G יש מניפה (x, B) בגודל $\min\{k, |B|\}$. (k מסלולים, או מסלול לכל קודקוד ב-B).

הוכחה: זו מסקנה ישירה של משפט מנגר: הגרף הוא k-קשיר, אז לכל $A, B \subseteq V(G)$ לא ריקות מתקיים $\kappa_G(A, B) = \varrho_G(A, B)$.

ניקח את $A \coloneqq \{x\}, B \coloneqq B$. ברור שמתקיים $\kappa_G(A, B) = \min\{k, |B|\}$ (או שנוריד k קודקודים, או שנוריד את כל B).

אז $\varrho_G(A, B) = \min\{k, |B|\}$, כנדרש.

משפט דיראק – *Dirac's Theorem*

הכללה של משפט וויטני (שאמר שבגרף 2-קשיר, כל 2 קודקודים יושבים על מעגל).

בהינתן $2 \leq k \in \mathbb{N}$, וגרף G שהוא k-קשיר, וקבוצה $S \subseteq V(G)$ כך ש- $2 \leq |S| \leq k$.

אזי, G מכיל מעגל שמכיל את S כולה. כלומר, בגרף k-קשיר, כל k קודקודים יושבים על מעגל.

הוכחה: באינדוקציה על k.

בסיס: עבור $k = 2$, זה משפט וויטני.

צעד: נבחר x, קודקוד שרירותי ב-S. נסמן $T = S \setminus \{x\}$.

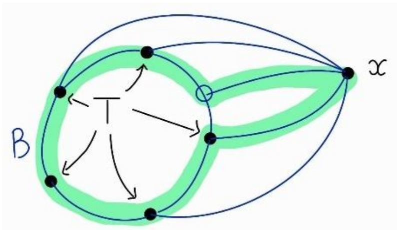
באופן טריוויאלי, $\kappa(G - x) \geq k - 1$ (כי הוא היה k-קשיר. הורדנו קודקוד אחד, אז אם זה "קידם" אותנו לכיוון אי-קשירות, עכשיו אנחנו ב- $k - 1$).

מכיוון ש- $|T| = k - 1$, אז לפי הנ"א יש מעגל בגרף שמכיל את כל T. נגדיר את כל קודקודי המעגל הזה כקבוצה B.

נגדיר את x בתור מקור של מניפה ונקבל שהמניפה (x, B) היא בגודל $k - 1$ (ישירות מלמת המניפה).

יש $k - 1$ מסלולים שיוצאים מ-x ומגיעים למעגל. הקבוצה T מחלקת את המעגל ל- $k - 1$ קשתות.

מכיוון ש- $k \geq 3$, מתקיים $k - 1 \geq 2$, אז יש לפחות 2 קשתות ויש לפחות 2 מסלולים זרים מ-x ל-T. ונוכל להשתמש במסלולים האלה כדי לייצר מעגל:



כלומר x גם חלק מהמעגל. אז בסה"כ יש k קודקודים ב- $T \cup \{x\}$, והם על מעגל. כנדרש.

8: Hamiltonicity

נסמן $\deg_G v$ את הדרגה (מספר השכנים) של קודקוד v, ו-$\delta(G)$ את $\min\limits_{v} \deg_G v$, הדרגה המינימום בגרף.

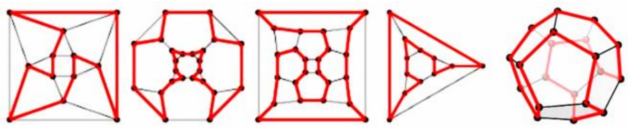
מסלול המילטון (*Hamilton path*) – מסלול <u>פשוט</u> הפורש את כל $V(G)$ (עובר בכל הקודקודים פעם אחת).

מעגל המילטון (*Hamilton cycle*) – מעגל <u>פשוט</u> הפורש את כל $V(G)$ (עובר בכל הקודקודים פעם אחת ומגיע חזרה להתחלה).

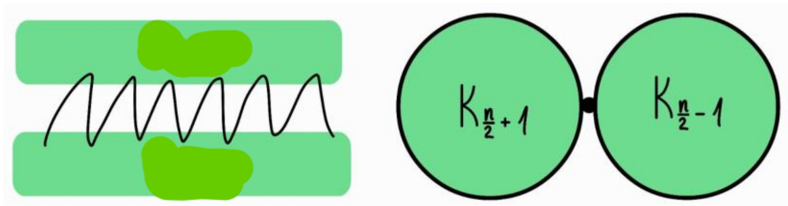
גרף ייקרא *Traceable* (ניתן למעקב) אם יש בו מסלול המילטון, ו-*Hamiltonian* (המילטוני) אם יש בו מעגל המילטון.

באופן כללי, בעיית ההכרעה של קיום מעגל המילטון בגרף היא *NPC*. נרצה להבין מתי בעיית הכרעה יכולה להיות ב-*P*.

דוגמאות לגרפים המילטונים:



דוגמאות לגרפים שלא יכולים להיות המילטונים: (בשניהם יש סה"כ n קודקודים. נניח ש-n זוגי, או ניקח את $\left\lfloor \frac{n}{2} \right\rfloor, \left\lceil \frac{n}{2} \right\rceil$).



שני גרפים $K_{\frac{n}{2}+1}, K_{\frac{n}{2}-1}$ שחולקים קודקוד אחד: ברגע שעברנו מצד אחד לצד השני, לא נוכל לחזור.

גרף דו-צדדי שלם עם צד אחד גדול יותר, $K_{\frac{n}{2}+1,\frac{n}{2}-1}$: כל מעבר חייב להיות רק בין שני הצדדים, ואנחנו ניתקע בצד הקטן.

שתי הדוגמאות הן גרפים מלאים יחסית: $\delta(G) = \frac{n}{2} - 1$. ועדיין, אין מעגל המילטוני.

משפט דיראק *Dirac's Theorem*

יהי גרף G עם $v(G) \geq 3$, ו- $\delta(G) \geq \frac{n}{2}$. אזי, G הוא המילטוני (מכיל מעגל המילטון).

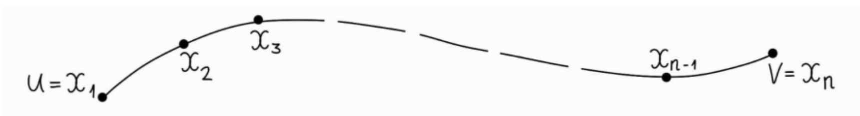
הוכחה: נב"ש שהמשפט לא נכון, ויהי G דוגמה נגדית מקסימלית בצלעות. כלומר, G לא המילטוני, אבל $G+e$ המילטוני לכל $e \notin E(G)$.

מכיוון ש-G לא המילטוני, הוא לא מלא (כי בגרף המלא יש מעגל המילטון). כלומר, קיימת צלע שלא בגרף: $\exists uv \in \binom{V(G)}{2} \setminus E(G)$.

מכיוון שב- $G+uv$ יש מעגל המילטון, ב- G יש מסלול-uv המילטוני. למה?

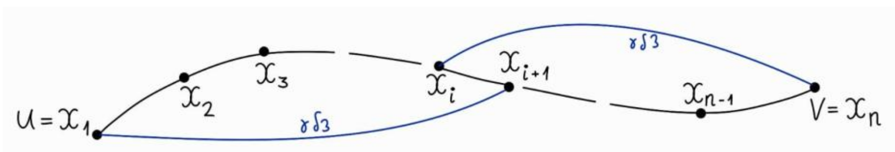
כי אם הוספת uv גרמה לכך שיש מעגל המילטון היא בהכרח חלק מהמעגל. אז אפשר להגיע מ-u ל-v בלי לחזור על קודקודים, ואז לעבור מ-v ל-u.

כלומר גם בלי הצלע uv, אפשר להגיע מ-u ל-v בלי לחזור על קודקודים, שזה מסלול המילטון. נקרא לו P:

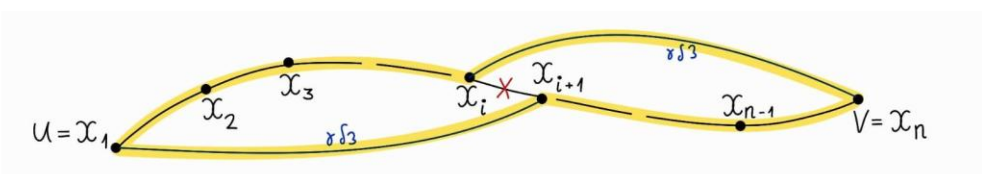


נציג את הרעיון של ***Rerouting:***

אם יש צלע מ-u ל-x_{i+1} כלשהו, וגם צלע מ-v ל- x_i:



אז נוכל לבצע **ניתוב מחדש**, ולזרוק את הצלע x_i, x_{i+1}:



וזה נותן לנו מעגל המילטון. נלך מ-u ל-x_{i+1}, נמשיך מ-x_{i+1} ל-v, נלך מ-v ל- x_i, ומ- x_i ל- u.

כלומר, אם יש *rerouting*, יש מעגל המילטון.

אינטואיציה: במקרה של $K_{\frac{n}{2}+1}, K_{\frac{n}{2}-1}$ למעלה, אפשר לראות שאי אפשר לעשות *rerouting*, כי אי אפשר "לעקוף" את הקודקוד באמצע.
אבל במקרה של $K_{\frac{n}{2}+1,\frac{n}{2}-1}$, קשה יותר לראות למה אי אפשר לעשות *rerouting*.

נגדיר:

$$S \coloneqq \{i \in [n] : ux_{i+1} \in E(G)\}, \qquad T \coloneqq \{i \in [n] : vx_i \in E(G)\}$$

T זה פשוט האינדקסים של הקודקודים שהם שכנים של v. S זה כל הקודקודים **הקודמים** במסלול לקודקודים שהם שכנים של u.

נשים לב ש $n \notin S$ (כי x_{n+1} לא קיים). ולפי ההנחה, אין צלע uv. וגם, מכיוון ש $ux_2 \in E(G)$ (הצעד הראשון במסלול), אז $1 \in S$.

בסה"כ, הקודקוד הראשון במסלול מחובר ל-u ותורם 1 ל-S, וכל קודקוד אחר במסלול שמחובר ל-u יתרום גם 1. אז $|S| \geq \deg_G(u)$.

נשים לב גם ש $n \notin T$, כי הגרף פשוט (אין לולאות). וגם, לפי הגדרתו, $|T| \geq \deg_G(v)$.

אם $S \cap T \neq \phi$, כלומר קיים i ששייך לשניהם, אז: $i \in S$ נותן לנו שיש צלע ux_{i+1}. $i \in T$ נותן לנו שיש צלע vx_i.

זה מאפשר *rerouting*, כמו שראינו בדוגמה לעיל. וזה אומר שיש מעגל המילטוני.

אז נניח $S \cap T = \phi$, כלומר $|S \cap T| = 0$. אזי, מהכלה והדחה והמסקנות על $|S|, |T|$:

$$|S \cup T| + |S \cap T| = |S| + |T| \geq \deg_G(v) + \deg_G(u)$$

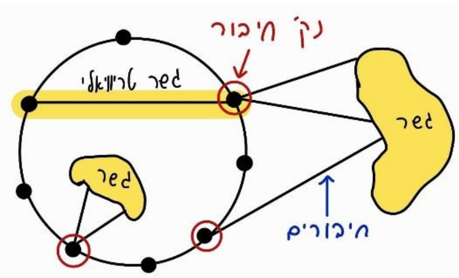
ומתקיים $\deg_G(v), \deg_G(u) \geq n/2$ כי $\delta(G) \geq \frac{n}{2}$. אזי, $\deg_G(v) + \deg_G(u) \geq n$. בסה"כ, $|S \cup T| \geq n$.

אבל ראינו ש $n \notin S \cup T$, שזו סתירה.

גשרים

יהי C מעגל בגרף G. **גשר** (*bridge*) ב-C הוא אחד משני דברים:

1. **גשר טריוויאלי** הוא מיתר במעגל – צלע בין שני קודקודים במעגל שאינה חלק מהמעגל.
2. או, רכיב קשירות של $G - C$ (קבוצות שאם נסיר את C מהגרף, הם יהיו רכיב קשירות):

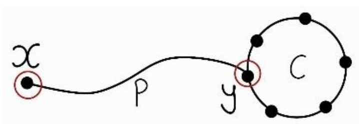


הצלעות שמחברות את הגשר עם המעגל נקראות **חיבורים** (*attachments*) של הגשר.

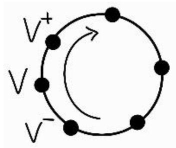
הקודקודים על המעגל שמתחברים לגשר נקראים **נקודות חיבור** (*points of attachments*).

Lollipops

תת-גרף $C \cup P$ שבנוי ממעגל C ומסלול $P \coloneqq x \rightsquigarrow y$ (כאשר נקודת המפגש היחידה של P עם C היא נקודת החיבור y) ייקרא (x, y)-*lollipop*.



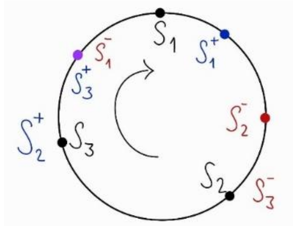
בהינתן מעגל C, ניתן לו **אוריינטציה** (כיוון). בהתחשב בכיוון הזה, נוכל לומר עבור קודקוד v איזה קודקוד הגיע לפני ואיזה אחריו.
נסמן v^- את הקודקוד לפניו, ו-v^+ את הקודקוד אחריו:



בהינתן קבוצת קודקודים על מעגל $S \subseteq V(C)$, נסמן:

$$S^- := \{s^- : s \in S\}, \qquad S^+ := \{s^+ : s \in S\}$$

את קבוצות הקודקודים שבאים לפני ואחרי הקודקודים של S.

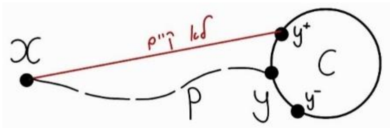


נשים לב שקודקוד יכול להיות שייך ליותר מקבוצה אחת. לדוגמה $S^- \cap S^+$, או ל- $S \cap S^+$.

Lollipop Lemma

יהי C המעגל הארוך ביותר ב-G. יהי $C \cup P := (x,y)\text{-}lollipop$ ב-G. נכוון את C. אזי, $xy^-, xy^+ \notin E(G)$.

כלומר, אין צלע מ-x לקודקודים שלפני או אחרי y במעגל:



נוכיח: נב"ש שיש – אזי מצאנו מעגל ארוך יותר מ-C בגרף: נתחיל מ-y^+, נלך בכיוון המעגל עד y, נלך ל-x, ונלך ל-y^+.

זה מעגל ארוך יותר מ-C כי איבדנו רק צלע אחת (את yy^+) והרווחנו לפחות 2 צלעות: הצלע xy^+, ואת הצלעות של P (לפחות אחת).

Erdős – Chvátal Theorem

נזכיר: $\alpha(G)$ זה גודל הקבוצה הבת"ל הגדולה ב-G, $\kappa(G)$ זה ה-k-קשירות של G (מספר הקודקודים המינימלי שצריך כדי לנתק זוג קודקודים כלשהו).

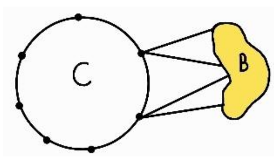
יהי G עם $\nu(G) \ge 3$, $\kappa(G) \ge \alpha(G)$. אזי G הוא המילטוני.

הוכחה:

נב"ש שהטענה שגויה. יהי גרף G עם התכונות לעיל ויהי C המעגל הארוך ביותר ב-G.

מההנחה ש-G לא המילטוני, נקבל ש $V(G) \setminus V(C) \neq \phi$. יש קודקודים (לפחות אחד) בגרף שלא נמצאים על המעגל.

אזי, יש ב-G גשר לא טריוויאלי עם 2 נקודות חיבור:



למה? נבחן את מרכיבי הקשירות של $G - C$: כבר אמרנו שיש קודקוד שלא על המעגל, אז גשר קיים, אולי אין לו 2 נקודות חיבור?

אם אין לו בכלל נקודות חיבור, אז מכיוון שיש לפחות קודקוד אחד בגשר ואחד במעגל, זה שני קודקודים שהם בת"ל. כלומר $\alpha(G) \ge 2$.

אז מההנחה, $\kappa(G) \ge 2$. אבל הגרף לא קשיר. סתירה.

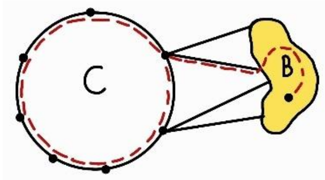
אם יש לו נקודת חיבור אחת: המעגל הוא מעגל פשוט, כלומר יש לו לפחות 3 קודקודים. אז אחד הקודקודים בגשר (נקרא לו v) מחובר בצלע לאחד מהם.

יש עוד 2 קודקודים במעגל (נקרא לאחד מהם w), שלא מחוברים בצלע לאף קודקוד בגשר. אז $\{v, w\}$ היא קבוצה בת"ל, כלומר $\alpha(G) \ge 2$.

אז גם פה $\kappa(G) \ge 2$, אבל אפשר לנתק את הגשר מהמעגל על ידי הסרת קודקוד יחיד. גם סתירה.

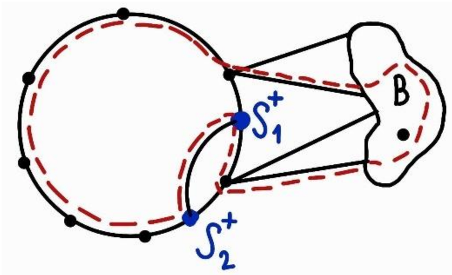
אז יהי B גשר כמו שאמרנו. נסמן S את קבוצת נקודות החיבור של B עם C. אנחנו יודעים ש $|S| \ge 2$.

לכל $b \in B, s \in S$, נקבל (b,s)-$lollipop$:



נכוון את C. מתקיים ש- S^+ היא קבוצה בת"ל, כי אחרת:

נב"ש שלא, כלומר יש צלע $s_1^+ s_2^+$ (המספרים הם בה"כ). אזי, נוכל למצוא מעגל גדול יותר, לדוגמה:



כי נאבד 2 צלעות מ-C, ונרוויח את הצלע $s_1^+ s_2^+$ ואת הצלעות שמחברות את B ל-C.

פורמלית: יהי Q מסלול $s_2 \rightsquigarrow s_1$ זר בקודקודים פנימיים, שמוכל ב-B. בהכרח קיים כי: יש ל-s_1 שכן ב-B (לכן הוא ב-S) וגם ל-s_2.

ויש בתוך B מסלול זר בקודקודים בין השכנים של s_1, s_2 (כי הוא רכיב קשירות). אזי, יש מעגל:

$$Q \cup s_2 C s_1^+ \cup \{s_1^+ s_2^+\} \cup s_2^+ C s_1$$

שארוך יותר מ-C בלפחות צלע אחת, כי הוא תופס את כל הקודקודים ב-C, ומרוויח לפחות קודקוד מ- $V(Q) \cap V(B)$.

בפרט, זה מראה ש $s_1^+ \neq s_2$, כלומר S לא יכולה להכיל קודקודים סמוכים במעגל.

נזכיר את ה- $lollipop$-$lemma$: אם יש (x,y)-$lollipop$, אז אין צלע xy^+ או xy^-. כפי שאמרנו, לכל $b \in B, s \in S$, יש (b,s)-$lollipop$.

אזי $bs^+ \notin E(G)$ לכל $b \in B, s \in S$. כלומר אין אף צלע בין b ל-S^+, לכל $b \in B$.

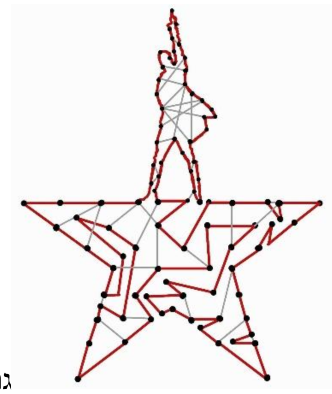
כלומר, $S^+ \cup \{b\}$ היא קבוצה בת"ל לכל $b \in B$. אז $\alpha(G) \geq |S^+| + 1$.

וניזכר ש-S היא קבוצת נקודות החיבור של B עם C. היא חתך בגרף (כי אם נסיר אותה, B יתנתק מ-C). כלומר $\kappa(G) \leq |S|$.

ומתקיים $|S^+| = |S|$ (כי לכל קודקוד $s \in S$ יש קודקוד $s^+ \in S^+$ ייחודי), כלומר:

$$\kappa(G) \leq |S| = |S^+| < |S^+| + 1 \leq \alpha(G)$$

סתירה להנחה ש $\kappa(G) \geq \alpha(G)$. ■



גרף המילטוני

נזכיר: $\chi(G)$ הוא המספר הכרומטי של G (מספר הצבעים המינימום שצריך כדי שתהיה k-צביעה תקינה), ו- $\Delta(G)$ הוא הדרגה המקסימום בגרף.

אלגוריתם צביעה חמדן

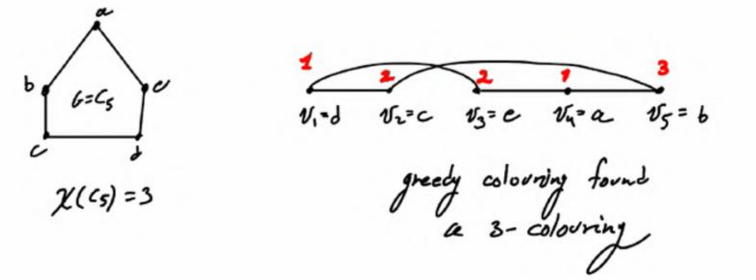
קלט: גרף G, וסידור כלשהו של קודקודי הגרף: $v_1, v_2, \dots, n_n$.   פלט: צביעה חוקית של קודקודי הגרף.

נעבור על כל הקודקודים לפי הסדר הנתון.

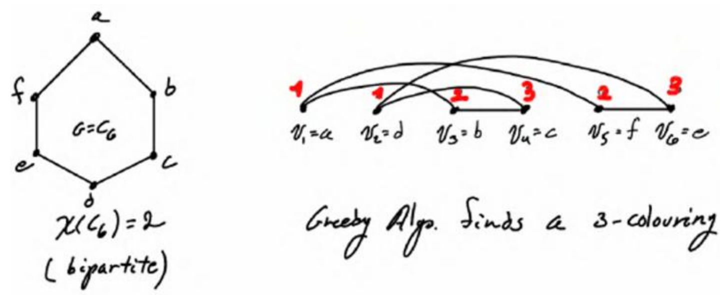
לכל v_i, ניתן לו את המספר הנמוך ביותר שאפשרי (כלומר נעבור על כל השכנים שלו, ונראה איזה צבעים כבר בשימוש).

קל לראות שהצביעה תקינה, כי לכל קודקוד נותנים צבע שונה מהשכנים שלו.

לדוגמה:



אבל עבור הגרף:



האלגוריתם הזה יעזור לנו להוכיח את הטענה הבאה:

$$\forall G, \qquad \chi(G) \le \Delta(G) + 1$$

רעיון ההוכחה: נסתכל על הקודקוד בעל דרגה $\Delta(G)$. השתמשנו בלכל היותר $\Delta(G)$ צבעים בשביל השכנים שלו, אז בשבילו נצטרך רק עוד צבע אחד.

יש גרפים שעבורם מתקיים $\chi(G) = \Delta(G) + 1$. לדוגמה מעגל אי-זוגי, $\chi = 3, \Delta = 2$. או גרף שלם: $\chi(K_r) = r,\ \Delta(K_r) = r - 1$.

משפט ברוקס – *Brook's Theorem*

יהי G גרף קשיר, שהוא לא גרף שלם או מעגל אי-זוגי. אזי, $\chi(G) \le \Delta(G)$.

נשים לב שזה רחוק מאוד מחסם אופטימלי: יהי $n = 2k$ עבור $k \in \mathbb{N}$ כלשהו. אזי, $\Delta(K_{n,n}) = k$ אבל $\chi(K_{n,n}) = 2$.

הוכחה

נוכל להניח ש $\Delta(G) \ge 3$. למה?

אם $\Delta(G) \le 2$, אז העובדה ש-G קשיר ולא מעגל אי-זוגי מכריחה אותו להיות מעגל זוגי או מסלול פשוט. בשניהם מתקיים $\chi(G) \le \Delta(G)$.

נבחין בין 2 מקרים משלימים:

קיים קודקוד שיש לו דרגה קטנה ממש מהדרגה המקסימום: $\exists v \in V(G)\ s.t\ \deg_G(v) < \Delta(G)$.

לא קיים, כלומר G הוא $\Delta(G)$-רגולרי. תזכורת – גרף הוא k-רגולרי אם מתקיים $\deg(v) = k$ לכל $v \in V(G)$.

במקרה הראשון:

קיים בגרף עץ פורש T, שמושרש ב-v (פשוט כי הגרף קשיר. זה לא אומר שהגרף כולו עץ. זה רק עץ פורש, תת-גרף).

נגדיר סידור של הקודקודים לפי *post-order*: כך שאחד העלים ראשון, ו-v אחרון. בסידור הזה, לכל קודקוד (חוץ מ-v) יש את אחד משכניו אחריו בסידור.

אם נפעיל את האלגוריתם החמדן על הסידור הזה, נקבל $\Delta(G)$-צביעה. למה?

לכל הקודקודים (כולל v) יש לכל היותר $\Delta(G)-1$ שכנים לפניהם. כי לכל אחד יש לכל היותר $\Delta(G)$ שכנים, ואחד מהם מופיע אחריו.

ול-v יש לכל היותר $\Delta(G)-1$ שכנים, וכולם לפניו.

במקרה השני:

אם אין קודקוד שיש לו פחות שכנים מ-$\Delta(G)$, הטיעון לא עובד עבור השורש (כל השאר כן מתקיימים).

נבחן את הקשירות של הגרף:

אם $\kappa(G)=1$, זה אומר שיש קודקוד חתך:

וכזכור, הגרף הוא $\Delta(G)$-רגולרי. נפריד בין חלקי הגרף:

מתקיים: $\deg_{G_1}(v), \deg_{G_2}(v) < \Delta(G)$. (כי הוא היה בעל דרגה $\Delta(G)$, ועכשיו הוא מחובר רק לחלק).

כל שאר הקודקודים עדיין מחוברים לכל השכנים שלהם: $\forall x \in V(G_1): \deg_{G_1}(x)=\Delta(G),\ \ \forall y \in V(G_2): \deg_{G_2}(y)=\Delta(G)$.

עכשיו, בכל חלק, יש לנו קודקוד בעל דרגה קטנה ממש מ- $\Delta(G)$. אז אנחנו **במקרה הראשון** עבור כל חלק.

לכל צד נמצא עץ פורש ונבצע את הצביעה. נקבל שתי צביעות: ψ_1,ψ_2. שכל אחת מהן משתמשת ב- $\Delta(G)$ צבעים.

אם $\psi_1(v)=\psi_2(v)$, סיימנו. גם אם לא, נבין שאין משמעות ל"שמות" (או המספרים) של הצבעים – העיקר זה שיש חלוקה.

אז נוכל לבצע החלפות כך ש- $\psi_1(v)=\psi_2(v)$. נקבע את הצביעה לפי ψ_1. נגדיר:

$$A \coloneqq \{x \in V(G_2): \psi_2(x)=\psi_2(v)\}, \qquad B \coloneqq \{x \in V(G_2): \psi_2(x)=\psi_1(v)\}$$

כלומר, A זאת מחלקת השקילות של v ב-G_2, ו-B הם הקודקודים שיש להם את הצבע שאנחנו רוצים בשביל v. נחליף בין הצבעים שלהם.

אז אם יש קודקוד חתך, מצאנו צביעה ב- $\Delta(G)$ צבעים. אז נוכל להניח שאין קודקוד חתך, כלומר $\kappa(G)\geq 2$.

סיכום ביניים: הגרף G מקיים: $\kappa(G)\geq 2$, הוא $\Delta(G)$-רגולרי, $\Delta(G)\geq 3$, הוא לא מעגל אי-זוגי, והוא לא הגרף השלם.

אם היינו יכולים למצוא 3 קודקודים x, y, z כך ש:

$$xy, xz \in E(G), \qquad yz \notin E(G), \qquad G-\{y,z\}\ is\ connected$$

אז נוכל להגדיר סידור של הקודקודים כך:

יהי T העץ הפורש של $G-\{y,z\}$, מושרש ב-x. נסדר את הקודקודים לפי העץ כמו שעשינו קודם.

נשים בהתחלה את y, z. אנחנו יודעים שיש להם צלע ל-x. (יכול להיות שיש צלעות בין y או z לקודקוד פנימי).

נפעיל את האלגוריתם החמדן על הסידור הזה. נקבל: $\psi(y)=\psi(z)=1$, כי אין ביניהם צלע.

לכל קודקוד אחר בעץ (חוץ מ-x), יש שכן אחריו בסידור. אז יש לכל היותר $\Delta(G)-1$ קודקודים לפני, אז נצטרך לכל היותר $\Delta(G)-1$ צבעים.

ל-x יש $\Delta(G)$ קודקודים לפניו. אבל לשניים מהם (y, z) יש את אותו הצבע. אז יש לנו צבע "פנוי" מתוך ה-$\Delta(G)$ צבעים.

נותר להראות שאכן יש 3 קודקודים כאלה.

טענה: יהיו $3 \leq k \in \mathbb{N}$, גרף G k-רגולרי ו-2-קשיר, לא מלא. אזי קיימים 3 קודקודים x, y, z כך ש:

$$xy, xz \in E(G), \qquad yz \notin E(G), \qquad G-\{y,z\}\ is\ connected$$

הוכחה: ראשית, יש 3 קודקודים כי הוא k-רגולרי עם $k \geq 3$.

עכשיו, נניח שקיים קודקוד v כך ש $\kappa(G-v)=1$. כלומר הוא לא קודקוד חתך, אבל אם נוריד אותו, אז יהיו בגרף קודקודי חתך.

נתבונן בגרף הבלוקים של $G-v$ (אפשר לחשב עם DFS).

יהיו בו לפחות 2 בלוקים שהם עלים. כל קודקודי החתך שיש ב $G-v$, הם לא קודקודי חתך ב-G (כי $\kappa(G) \geq 2$).

מה יכול לגרום לכך שב-G אין קודקודי חתך? זה קורה רק אם v היה שכן של קודקוד פנימי בכל אחד מהבלוקים שהם עלה.

ניזכר שיש לפחות 2 בלוקים שהם עלים. אז יש לפחות 2 קודקודים פנימיים כאלה. נקרא להם y, z. אין ביניהם צלע (כי הם בבלוקים נפרדים). ו-v יהיה ה-x שלנו.

ובנוסף, y, z הם קודקודים פנימיים בבלוקים. אז בוודאי $G-\{y,z\}$ קשיר. בסה"כ, קיבלנו 3 קודקודים x, y, z כמו שרצינו.

נניח כעת ש $\kappa(G-v) \geq 2$ לכל $v \in V(G)$. מכיוון ש-G לא שלם, קיימים 3 קודקודים x, y, z כך ש:

$$xy, xz \in E(G), \qquad yz \notin E(G)$$

צריך להראות רק ש $\kappa(G-y-z) \geq 1$. ואנחנו יודעים ש $\kappa(G-y) \geq 2$ (מההנחה שאין קודקודי חתך), אז אכן מתקיים.

סיכום ההוכחה:

צ"ל: יהי G גרף קשיר, שהוא לא גרף שלם או מעגל אי-זוגי. אזי, $\chi(G) \leq \Delta(G)$.

זה מתקיים עבור $\Delta(G) \leq 2$, אז נניח ש $\Delta(G) \geq 3$.

במקרה ש $\exists v \in V(G)\ s.t\ \deg_G(v) < \Delta(G)$, תיארנו סידור של הקודקודים שמאפשר צביעה לפי האלגוריתם.

במקרה המשלים, ראינו שאם יש קודקוד חתך, אז אפשר למצוא את אותה צביעה (בשני חלקים ואז לאחד).

אז נניח שאין קודקודי חתך.

אמרנו שאם נמצא 3 קודקודים: $xy, xz \in E(G),\ \ yz \notin E(G),\ \ G-\{y,z\}\ is\ connected$, אז נוכל לתאר סידור בשביל הצביעה.

ראינו שאם יש קודקודים שאם נסיר אותם הגרף יהיה 1-קשיר, נוכל למצוא 3 קודקודים מתאימים (דרך גרף הבלוקים).

ואם לכל קודקוד שנסיר הגרף עדיין 2-קשיר, אז באופן מיידי יש 3 קודקודים מתאימים.

בסה"כ, עברנו על כל המקרים ותמיד הצלחנו להראות סידור שמאפשר צביעה לפי האלגוריתם.

Vizing's Theorem

צביעת צלעות ב-k צבעים היא פונקציה $\varphi: E(G) \to [k]$. צביעה בצלעות תיקרא *proper* (תקינה) אם צלעות שחולקות קודקוד צבועות בצבעים שונים.

גרף G ייקרא ***k-edge-colorable*** אם יש לו צביעה תקינה ב-k צבעים.

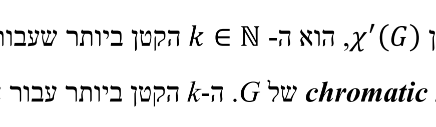
האינדקס הכרומטי – *chromatic index* של G מסומן $\chi'(G)$, הוא ה- $k \in \mathbb{N}$ הקטן ביותר שעבורו G הוא *k-edge-colorable*.

לא להתבלבל עם $\chi(G)$, **המספר הכרומטי – *chromatic number*** של G. ה-k הקטן ביותר עבור צביעת **קודקודים.**

גרף הצלעות – *line graph* של G מסומן $L(G)$, הוא גרף שקודקודיו הם $E(G)$. יש צלע בין שני קודקודים ב-$L(G)$ אם הם היו צלעות סמוכות ב-G.

אפשר לדמיין את זה ככה: נשים קודקוד על כל צלע, ואם הצלעות סמוכות – נחבר אותן:

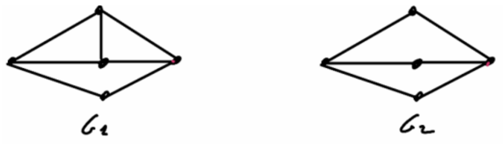
יוצא ש $L(K_3)=K_3$.   עוד דוגמה:



נשים לב ש: $L(K_{1,r})=K_r$. הגרף $K_{1,3}$ נקרא *claw graph*.

9: Graph Colorings

נביט בגרפים G_1, G_2:



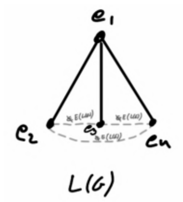
הם מכילים הרבה עותקים של $K_{1,3}$ בתור תתי-גרפים. אבל, רק ב-G_2 העותק מופיע בתור **תת-גרף-מושרה** (*induced subgraph*).

תת-גרף-מושרה הוא תת-גרף שמתקבל ע"י לקיחת תת-קבוצה של קודקודי G, ואת **כל** הצלעות שיש בקודקודים האלו ב-G.

נשים לב: לכל G, $L(G)$ לא מכיל *claw* מושרה. למה?

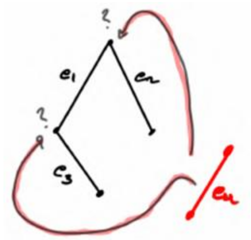
נב"ש שקיים $L(G)$ שיש בו $K_{1,3}$ מושרה.

כלומר, יש קודקודים: e_1, e_2, e_3, e_4. יש צלעות: $e_1e_2,\ e_1e_3,\ e_1e_4$. **ואין** את הצלעות $e_2e_3,\ e_2e_4,\ e_3e_4$:



איך G נראה? הצלעות e_1, e_2 סמוכות, נחבר אותן בקודקוד v_1. גם e_3 סמוכה ל-e_1, אבל לא ל-e_2. נחבר אותה ל-e_1 עם קודקוד v_2.

כרגע, e_1 סמוכה ל-e_2 מצד אחד ול- e_3 מהצד השני. גם e_4 סמוכה ל-e_1, אבל לא ל- e_2 או e_3. אין איפה לחבר אותה:



סתירה.

אבחנות:

המספר הכרומטי $\chi(L(G))$ של גרף הצלעות, שווה לאינדקס הכרומטי $\chi'(G)$ של הגרף המקורי: $\boldsymbol{\chi(L(G)) = \chi'(G)}$.

כי פשוט המרנו את המילה "קודקוד" במילה "צלע".

האינדקס הכרומטי גדול מהדרגה המקסימום: $\boldsymbol{\chi'(G) \geq \Delta(G)}$.

כי אם לקודקוד יש Δ שכנים, יש לו (מן הסתם) Δ צלעות, אז זה Δ צלעות שחולקות קודקוד וכל אחת צריכה צבע אחר.

הדרגה המקסימום של גרף הצלעות חסומה ב: $\boldsymbol{\Delta(L(G)) \leq 2 \cdot (\Delta(G) - 1) = 2\Delta - 2}$.

למה? נתבונן בצלע כלשהי $uv \in E(G)$. הדרגה שלה ב-$L(G)$ היא: $\deg_{L(G)}(uv) = \deg_G(u) + \deg_G(v) - 2$.

כי כל צלע שמחוברת ל-u או v תהיה מחוברת ל-uv בגרף הצלעות. ומורידים 2 כי ספרנו את הצלע uv עצמה פעמיים,

פעם אחת מ-u ופעם אחת מ-v.

אז הדרגה המקסימום של קודקוד ב-$L(G)$ היא $2(\Delta(G) - 1) = 2\Delta(G) - 2$.

לכן, אם $L(G)$ לא שלם או מעגל אי-זוגי, אז לפי ברוקס נקבל $\chi(L(G)) \leq \Delta(L(G)) \leq 2(\Delta(G) - 1)$. ובסה"כ:

$$\Delta(G) \leq \chi'(G) \leq 2(\Delta(G) - 1)$$

A Theorem of König

אם G גרף דו-צדדי, אז $\chi'(G) = \Delta(G)$.

הוכחה: ניזכר, שבגרף שהוא k-רגולרי (לכל קודקוד יש דרגה k) יש שידוך מושלם (הוכחה בתרגול).

אם נסיר את הצלעות של שידוך מושלם מגרף דו-צדדי k-רגולרי, נקבל גרף דו"צ $(k-1)$-רגולרי.

למה? כי השידוך הוא בדיוק צלע אחת לכל קודקוד. אז הגרף שנשאר הוא $(k-1)$-רגולרי.

הגרף היה דו"צ כלומר אין לו מעגל אי זוגי. והסרת צלעות לא תייצר מעגל חדש, ובפרט לא מעגל אי זוגי. אז הגרף המתקבל הוא גם דו"צ.

אז גם בגרף החדש יש שידוך מושלם. אז נוכל בעצם להוריד שידוכים אחד אחרי השני, ולכל שידוך נגדיר צבע.

נבצע את זה $\Delta(G)$ פעמים, וכל פעם נצטרך צבע חדש כי כל צלע חולקת קודקוד עם צלע אחרת מכל שאר השידוכים.

זה מוכיח את הטענה עבור גרף דו-צדדי k-רגולרי.

כדי להוכיח עבור גרף דו"צ שאינו k-רגולרי, מספיק להוכיח שכל גרף דו"צ G הוא תת-גרף של גרף G^S שהוא דו"צ $\Delta(G)$-רגולרי.

ואז, נקבל $\Delta(G)$-צביעה של G^S כמו שתיארנו, וזה נותן לנו $\Delta(G)$-צביעה על G.

נתון ש- $G=(A\cup B,E)$ הוא דו"צ. אם $|A|\neq|B|$, נוסיף קודקודים לקטן יותר עד שהם שווים בגודלם. נקרא לגרף הזה G'. עכשיו, נבצע:

כל עוד G' הוא **לא** $\Delta(G)$-רגולרי, אז לשני הצדדים יש קודקוד v שמקיים $\deg_{G'}(v)<\Delta(G)$. למה? ראשית, בוודאות קיים בגרף.

אם יש רק בצד אחד, אז אם נספור את הצלעות מכל צד נקבל מספרים שונים. ולא הגיוני שיוצאות יותר צלעות מצד אחד מאשר מה שמגיע לצד השני.

אז נבחר x, y כאלה ונוסיף את הצלע xy ל- G'.

ככה, בכל שלב הגרף נשאר דו"צ, והוא בסוף יהיה $\Delta(G)$-רגולרי.

זה מוכיח שכל גרף דו"צ שאינו k-רגולרי, הוא תת-גרף של גרף G^S שהוא דו"צ $\Delta(G)$-רגולרי. כנדרש.

משפט ויזינג

לכל G מתקיים $\chi'(G)\le\Delta(G)+1$.

נובע מכך (יחד עם החסם התחתון הטריוויאלי $\Delta(G)\le\chi'(G)$ שראינו), ש- $\Delta(G)\le\chi'(G)\le\Delta(G)+1$ לכל גרף G.

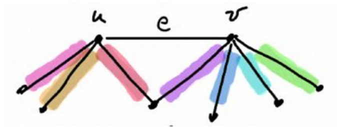
בהינתן גרף לא דו"צ, בעיית ההכרעה האם $\chi'(G)$ שווה $\Delta(G)$ או $\Delta(G)+1$, היא NPC.

רעיון ההוכחה:

הוכחה באינדוקציה על מספר הצלעות, $e(G)$. אפשר להניח $e(G)>0$. נבחר $e\coloneqq uv\in E(G)$, ונגדיר $G'\coloneqq G-e$.

מהנ"א, $\chi'(G')\le\Delta(G')+1$. כלומר קיימת בו $(\Delta(G)+1)$-צביעה לצלעות, נקרא לה φ. וגם, $\Delta(G')\le\Delta(G)$ באופן טריוויאלי.

עכשיו, נחזיר את uv ונצבע אותה. הבעיה שלנו היא שאם הצלעות שיוצאות מ-u ו-v משתמשות ב-$\Delta(G)+1$ צבעים, אז נצטרך להוסיף עוד צבע.



אבל אנחנו יודעים שיש $\Delta(G)+1$ צבעים, והדרגה המקסימום היא $\Delta(G)$.

כלומר, כל קודקוד $x\in V(G')=V(G)$, **רואה** $\deg_{G'}(x)\le\Delta(G')\le\Delta(G)$ צבעים ב-φ (קודקוד רואה את הצלעות שמחוברות אליו).

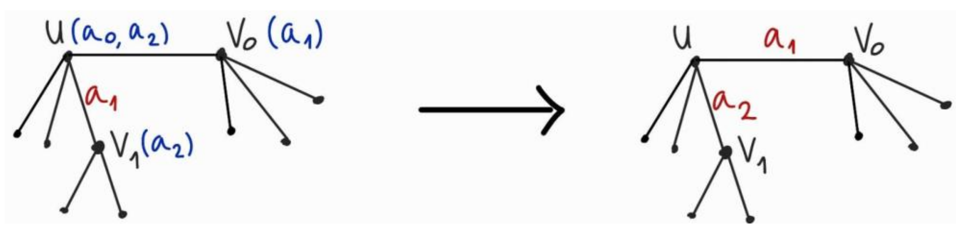
אזי, לכל $x\in V(G)$, קיים $k\in[\Delta(G)+1]$ כך ש-x לא רואה את הצבע k תחת φ. אז במקרה הבעייתי, ל-u יש צבע שהוא לא רואה. וכנ"ל v.

אם זה היה אותו צבע לשניהם, סיימנו.

אם זה צבעים שונים, נסמן a_0 את הצבע ש-u לא רואה, ונסמן a_1 את הצבע ש- $v\coloneqq v_0$ לא רואה.

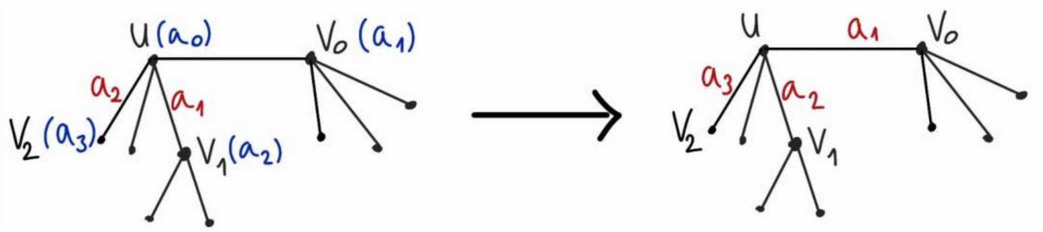
כלומר, קיים $v_1\in N_G(u)$ כך שהצלע uv_1 צבועה ב-a_1. וגם ל-v_1 יש צבע שהוא לא רואה, נקרא לו a_2.

אם u לא רואה את a_2, נצבע את uv_1 ב- a_2, ואת uv_0 ב- a_1, וזה נותן לנו צבע ל-uv:



אם כן u רואה את a_2, אז זה אומר שיש קודקוד v_2 כך ש uv_2 צבועה ב-a_2. כלומר יש צבע a_3 ש-v_2 לא רואה.

אם u לא רואה את a_3, אז נצבע את uv_2 ב-a_3, נעביר את a_2 ל- uv_1, ונעביר את a_1 ל-uv_0:



אם u כן רואה את a_3, נמשיך את התהליך עם כל השכנים של-u (יש צלע שצבועה a_3, אז הקודקוד לא רואה את a_4...).

אם התהליך נגמר בלי שהצבע a_i חזר על עצמו, אז ההעברות הצליחו לתת צביעה של G. כי יש לכל היותר $\Delta(G)$ שכנים, אז כל השכנים משתמשים ב- $\Delta(G)$ צבעים, ונשאר אחד לצלע החדשה.

אז נניח שיש חזרות באמצע. יהי ℓ האינדקס הראשון כך ש-v_ℓ לא רואה את $a_{\ell+1}$ אבל $a_{\ell+1} \in \{a_1, \dots, a_\ell\}$. כלומר קיים $k \in [\ell]$ כך ש $a_{\ell+1} = a_k$:

1. אם v_ℓ לא רואה את a_0, נצבע את uv_ℓ ב-a_0 ונבצע את ההעברות החל מ-v_ℓ ומטה, ונקבל צביעה תקינה.

2. אם v_ℓ כן רואה את a_0: יהי P מסלול שמתחיל ב-v_ℓ, שהצלעות שלו צבועות a_0 עד $a_k = a_{\ell+1}$. יש מסלול יחיד כזה ב-G.
   a. אם $v_{k-1} \in P$, אז P עוצר ב- v_{k-1} כי v_{k-1} לא רואה את a_k. אז P מתחיל ומסתיים בצבע a_0, ונוכל למצוא צביעה כך:
      i. נבצע את ההעברות מ- v_{k-1} והלאה.
      ii. נצבע את uv_{k-1} ב- a_0.
      iii. נחליף את הצבעים לאורך P (צלע צבועה ב- a_0 תיצבע $a_{\ell+1}$, והפוך).

   אז נניח ש $v_{(k-1)} \notin P$:

   b. אם $v_k \in P$, אז P מסתיים ב-u וכולל את הצלע uv_k שצבועה ב- a_k.
   ונקבל את הצביעה של G כך:
      i. נבצע את ההעברות מ-v_k והלאה.
      ii. נחליף צבעים לאורך P, זה כולל צביעת uv_k ב- a_0.

   אז נניח ש $v_k, v_{k-1} \notin P$:

   c. אם $u \in P$, אז P מגיע ל-u דרך צלע שצבועה ב- a_k.
   מכיוון ש- uv_k צבועה ב- a_k, ו- $v_k \notin P$, זה לא אפשרי כי זה מכריח שתי צלעות שצבועות a_k להיות סמוכות ל-u.

   d. נוכל להניח ש- P לא מבקר בקבוצה $\{u, v_k, v_{k-1}\}$, ומסתיים בקודקוד שלא בקבוצה $\{u, v_\ell, v_k, v_{k-1}\}$.
   אז נקבל צביעה של G כך:
      i. נבצע את ההעברות מ- v_ℓ והלאה.
      ii. נצבע את uv_ℓ ב- a_0.
      iii. נחליף צבעים לאורך P.

תכנון (תכנות) לינארי – *Linear Programming*

בהינתן: מטריצה $A \in M_{m\times n}(\mathbb{R})$, ווקטורים $c \in \mathbb{R}^n, b \in \mathbb{R}^m$,

ופונקציית מטרה: $x \mapsto c^T x, x \in \mathbb{R}^n$ (ממפה את x למכפלה הסקלרית מעל $\mathbb{R}^n$, *inner product*),

המטרה היא למקסם (*maximize*) או למנמן/למזער (*minimize*) את פונקציית המטרה, תחת אילוצים לינארים הנתפסים ע"י A ו-b.

נבחין בין 2 סוגים של תכניות (לא קוד, אלא תכנית פעולה):

תכנון בשלמים – *IP, Integer Programming*

בדרך כלל *NPC*. נרצה למקסם (בדוגמה) או למזער, משהו מהצורה:

$$\max_{x\in\mathbb{Z}}\{c^T x : Ax \leq b\}$$

יכולים להיות יותר אילוצים. ההשוואה בין Ax ל-b היא לפי אינדקס, כלומר נדרוש במקרה הזה שהכניסה ה-i של Ax תהיה קטנה-שווה לכניסה ה-i של b.

תכנון לינארי – *LP, Linear Programming*

בדרך כלל *P*. אותו דבר, אבל $x \in \mathbb{R}^n$.

לדוגמה בעיית תרמיל הגב. בעיית תרמיל הגב בשלמים היא בעיה של *IP*, בעיית תרמיל הגב בשברים היא *LP*.

הרבה מהבעיות שלנו יהיו *IP*. אז מה שנרצה לעשות, זה לקחת בעיה של גרפים ולצקת אותה בתור בעיית *IP* (שהיא *NPC*).

ואז, נעשה רילקסיציה (*relaxation*) לגרסת *LP*, נמצא פתרון אופטימלי (בשברים) ל-*LP*, ונאמר שזה מקרב (איכשהו) לפתרון אופטימלי *IP*.

***IP* ו-*LP* בבעיית שידוך מקסימום**

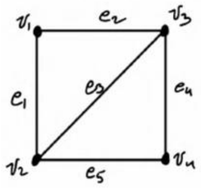
מטריצת הסמיכויות (*Incidence matrix*) של גרף היא מטריצה שמתארת את הקשר בין הצלעות והקודקודים (לא מטריצת שכנויות!).

כל עמודה מתארת צלע, כל שורה היא קודקוד. עבור גרף G, נסמן:

$$M \coloneqq M(G) = \begin{array}{c|c} & e_1 \quad e_2 \quad \cdots \quad e_m \\ \begin{matrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{matrix} & \begin{bmatrix} & \cdots & \\ \vdots & \in \{0,1\} & \vdots \\ & \cdots & \end{bmatrix} \end{array}, \qquad M_{ve} \coloneqq \begin{cases} 1, & v \in e \\ 0, & v \notin e \end{cases}$$

כלומר, בשורה של הקודקוד בעמודה של הצלע, נשים 1 אם הצלע הזו מחוברת לקודקוד.

לדוגמה:



$$\begin{array}{c|c} & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

לכל עמודה יש בדיוק 2 אחדות. לכל שורה יש $\deg_G(v)$ אחדות.

בהינתן גרף עם n קודקודים ו-m צלעות, נחפש וקטור בוליאני $x \in \{0,1\}^m$ שהוא **הווקטור האופייני** (*characteristic vector*) של שידוך, כלומר:

$$x_i = \begin{cases} 1, & e_i \in Matching \\ 0, & e_i \notin Matching \end{cases}$$

שיש לו כמות מקסימום של אחדות (שידוך מקסימום). כלומר למקסם את:

$$\sum_{i=1}^{m} x_i \Rightarrow \max(1^T x)$$

הכוונה ב-"1" היא ווקטור האחדות באורך המתאים.

נשתמש ב- $M \coloneqq M(G)$ כדי לתפוס את האילוצים של השידוך. האילוץ שלנו הוא שלכל קודקוד יש לכל היותר צלע אחת שנוגעת בו בשידוך.

בהינתן ווקטור אופייני של שידוך $u \in \{0,1\}^m$, נביט ב- *Mu*: המכפלה של המטריצה M כפול u:

$$\begin{array}{cc} & \begin{array}{cccc} e_1 & e_2 & \cdots & e_m \end{array} \\ \begin{array}{c} V_1 \\ V_2 \\ \vdots \\ V_n \end{array} & \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 1 \\ 0 & 1 & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}}_{\substack{M \\ n\times m}} \end{array} \underbrace{\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix}}_{\substack{u \\ m\times 1}} = \underbrace{\begin{bmatrix} \\ (Mu)_i \\ \\ \end{bmatrix}}_{\substack{Mu \\ n\times 1}}$$

כאשר $(Mu)_i := \langle v_i, u\rangle$, כמות הצלעות שיהיו מחוברות לקודקוד. לכן נדרוש $Mu \leq 1$ (בכל מקום לכל היותר 1). בסה"כ, הניסוח של הבעיה:

$$IP\ for\ max\ matching: \max_{x\in\mathbb{Z}^m}\{1^T x : Mx \leq 1, x \geq 0\}$$

והניסוח המתאים ב-*LP*, *corresponding LP relaxation*:

$$\max_{x\in\mathbb{R}^m}\{1^T x : Mx \leq 1, x \geq 0\}$$

במקרה הזה זה לא באמת משנה, כי $max\text{-}matching \in P$.

IP ו-LP בבעיית כיסוי קודקודים מינימום

נרצה למצוא פיתרון לבעיית כיסוי הקודקודים (*vertex cover, VC*). היא בעיה ב-*NPC*.

בהינתן גרף *G* עם *n* קודקודים ו-*m* צלעות, אנחנו מחפשים וקטור אופייני עבור ה-*VC*: $y \in \{0,1\}^n$, כך ש $y_i = 1$ אם $v_i \in VC$, ו-0 אחרת.

ואנחנו רוצים למזער את הסכום, שזה מקביל למזעור המכפלה של הווקטור ב-1:

$$\sum_i y_i \Rightarrow \min 1^T y$$

אנחנו צריכים לתפוס את האילוצים של *VC* ע"י מטריצת הסמיכויות. לכל צלע, לפחות אחד הקצוות שלה חייב להיות ב-*VC*. נשקול את M^T.

$$\begin{array}{cc} & \begin{array}{cccc} V_1 & V_2 & \cdots & V_n \end{array} \\ \begin{array}{c} e_1 \\ e_2 \\ \vdots \\ e_m \end{array} & \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 1 \\ 0 & 1 & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}}_{\substack{M^T \\ m\times n}} \end{array} \underbrace{\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix}}_{\substack{y \\ n\times 1}} = \underbrace{\begin{bmatrix} \\ (M^T y)_e \\ \\ \end{bmatrix}}_{\substack{M^T y \\ m\times 1}}$$

כאשר $(M^T y)_e$ זה מספר הקודקודים בכיסוי ששייכים ל-*e*. אנחנו דורשים שלכל *e*, $(M^T y)_e \geq 1$. כלומר הניסוח של *IP* ו-*LP*:

$$\min_{y\in\mathbb{Z}^m}\{1^T y : M^T y \geq 1, y \geq 0\}, \qquad \min_{y\in\mathbb{R}^m}\{1^T y : M^T y \geq 1, y \geq 0\}$$

הניסוח ב-*LP* הוא ב-*P* במקום *NPC*.

משפט קניג בניסוח LP

ניזכר במשפט קניג: אם גרף *G* הוא דו"צ, אז $\tau(G) = \nu(G)$. גודל השידוך המקסימום ν שווה לגודל הכיסוי המינימום τ.

בניסוח *IP*, התרגום הוא: אם *G* דו"צ, אזי:

$$\max_{x\in\mathbb{Z}^m}\{1^T x : Mx \leq 1, x \geq 0\} = \min_{y\in\mathbb{Z}^m}\{1^T y : M^T y \geq 1, y \geq 0\}$$

ומסתבר שגם:

$$\max_{x\in\mathbb{R}^m}\{1^T x : Mx \leq 1, x \geq 0\} = \min_{y\in\mathbb{R}^m}\{1^T y : M^T y \geq 1, y \geq 0\}$$

כלומר המעבר ל-*LP* משאיר את השוויון של הפתרונות האופטימליים. ההוכחה לזה היא לא טריוויאלית.

בבעיית השידוך, אפשר גם להוכיח שבפתרון האופטימלי ב-*LP*, הווקטור *x* עדיין שייך בעצם ל-$\mathbb{Z}^m$.

כלומר הפיתרון האופטימלי גם ב-$\mathbb{R}$ הוא בשלמים. גם לא טריוויאלי.

אם G לא דו"צ, אז בעיית השידוך של IP נשארת ב-P, אבל בעיית הכיסוי היא NPC.

צורות של LP

ראינו איך אפשר לצקת בעיות בגרפים לתוך תבניות של IP או LP. עכשיו בעצם יש לנו עניין (בלי קשר לבעיות בגרפים), לפתור בעיות במבנה הבא:

$$(1)\ \max\{c^Tx : Ax \le b\},\qquad (2)\ \max\{c^Tx : x\ge 0, Ax\le b\},\qquad (3)\ \max\{c^Tx : x \ge 0, Ax = b\}$$

$$(4)\ \min\{c^Tx : Ax \ge b\},\qquad (5)\ \min\{c^Tx : x\ge 0, Ax\ge b\},\qquad (6)\ \min\{c^Tx : x \ge 0, Ax = b\}$$

ראינו את (2) בשידוך מקסימום, ואת (5) בכיסוי מינימום.

מבחינת LP, כולן שקולות, ונראה איך לעבור בין פורמטים.

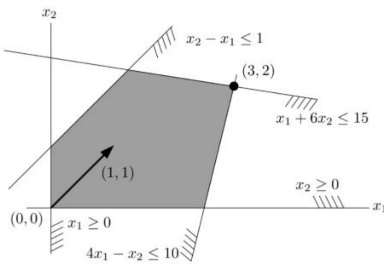
(3) ו-(6) הם במבנה **תצורת שוויון** – *equational form*, והפתרונות האופטימלים שלהם הם הכי קלים להסברה.

האילוצים $x \ge 0$ נקראים **אילוצי אי-שליליות** – *non-negativity constraints*.

דוגמה

צריך למקסם את $x_1 + x_2$, כך ש:

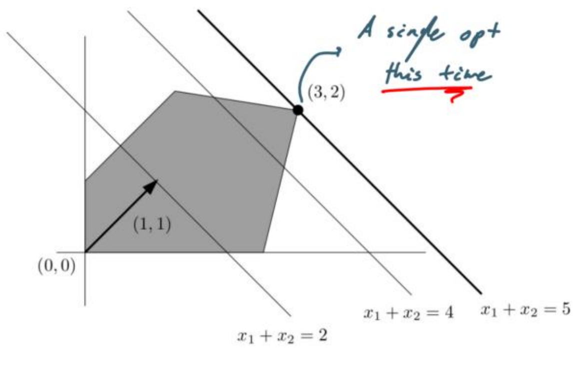
$$-x_1 + x_2 \le 1,\qquad x_1 + 6x_2 \le 15,\qquad 4x_1 - x_2 \le 10,\qquad x_1 \ge 0,\qquad x_2 \ge 0$$



נמיר את הבעיה לתצורת LP:

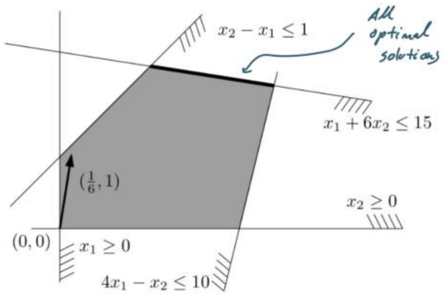
$$\max_{x\in\mathbb{R}^2}\left\{(1,1)\cdot\begin{pmatrix}x_1\\x_2\end{pmatrix} : \begin{pmatrix}-1 & 1\\ 1 & 6\\ 4 & -1\end{pmatrix}\cdot\begin{pmatrix}x_1\\x_2\end{pmatrix} \le \begin{pmatrix}1\\15\\10\end{pmatrix},\qquad \begin{pmatrix}x_1\\x_2\end{pmatrix} \ge \begin{pmatrix}0\\0\end{pmatrix}\right\}$$

פתירת הבעיה בצורה גיאומטרית:



נצייר את האילוצים. נצייר את $c = (1,1)$ ונדמיין את המכפלה שלו עם $\begin{pmatrix}x_1\\x_2\end{pmatrix}$, עד שנגיע לנקודה שבה המכפלה מקסימלית.

במקרה הזה קיבלנו פיתרון אופטימלי יחיד. יש בעיות שיתנו יותר, לדוגמה אם נרצה למקסם את $\frac{x_1}{6} + x_2$:



נקבל שכל הישר למעלה הוא פתרונות אופטימלים.

נרצה למצוא דרך שיטתית לפתור בעיות כאלה גם במימדים גבוהים.

נשקול את בעיית ה-*LP* הבאה. יש לה "אילוצים מעורבבים" – *mixed constraints*. היא לא באף אחת מהתצורות שראינו למעלה:

$$\max(3x_1 - 2x_2)\,, \qquad 2x_1 - x_2 \le 4, \qquad x_1 + 3x_2 \ge 5, \qquad x_2 \ge 0$$

נרצה להמיר את זה לתצורה (3): $\max\{c^T x : x \ge 0, Ax = b\}$. נטפל בכל אחד מהאילוצים.

האילוץ $2x_1 - x_2 \le 4$, אנחנו רוצים שהוא יהיה בתצורת שוויון. נגדיר משתנה חדש, $x_3 \ge 0$.

הוא נקרא ה- *slack variable*, כלומר הוא מכסה על הפער שייווצר לנו. נחליף את האילוץ המקורי ע"י:

$$2x_1 - x_2 + x_3 = 4, \qquad x_3 \ge 0$$

ב- $x_1 + 3x_2 \ge 5$, הבעיה היא שהכיוון הפוך. אז נכפיל הכל ב- -1 כדי שיהיה $\le$, ונחליף ב:

$$-x_1 - 3x_2 + x_4 = -5, \qquad x_4 \ge 0$$

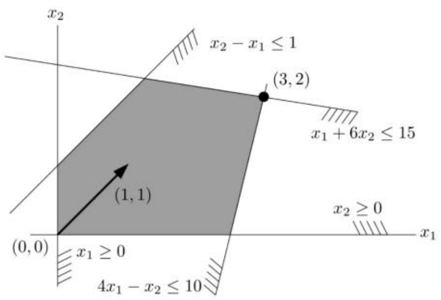
ונוסיף אילוץ *non-negativity* ל- x_1. כל מספר ממשי הוא הפרש בין שני ממשיים חיוביים, אז במקום x_1 נרשום:

$$x_1 = y_1 - z_1, \qquad y_1 \ge 0, \qquad z_1 \ge 0$$

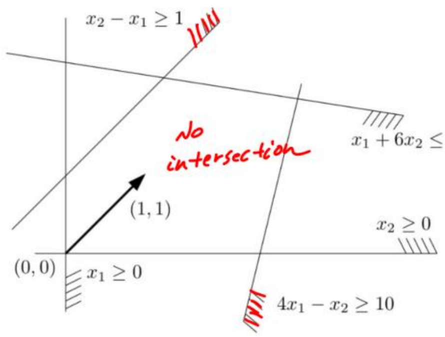
התצורה הסופית של בעיית ה-*LP* היא:

$$\max(3y_1 - 3z_1 - 2x_2)\,, \qquad 2y_1 - 2z_1 - x_2 + x_3 = 4, \qquad -y_1 + z_1 - 3x_2 + x_4 = 5, \qquad y_1, z_1, x_2, x_3, x_4 \ge 0$$

נחזור לבעיה הראשונה שראינו:

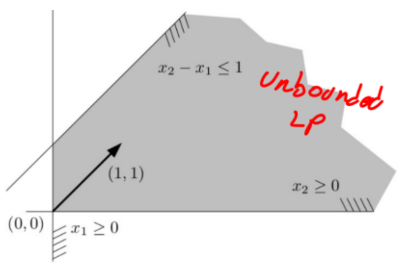


מה אם "נהפוך" את $-x_1 + x_2 \le 1,\ 4x_1 - x_2 \le 10$ ל- $-x_1 + x_2 \ge 1,\ 4x_1 - x_2 \ge 10$:



עכשיו, אין פיתרון. זאת בעיית *LP* שהיא *infeasible*, לא ישימה.

ואם נמחק את האילוצים $x_1 + 6x_2 \le 15,\ 4x_1 - x_2 \le 10$, הבעיה היא *feasible yet unbounded*, ישימה אבל אין מקסימום:



לכל בעיית *LP* שהיא *feasible* ולא *unbounded*, יש פיתרון אופטימלי.

תכנון לינארי בתצורת שוויון

בהינתן מטריצה $A \in M_{m\times n}(\mathbb{R})$ בעלת $rank(A) = m$ (כלומר $n \geq m$), אנחנו נחקור את הפתרונות של:

$$\max\{c^T x : Ax = b, \ \ x \geq 0\}$$

בדוגמה נראה מקסימום, כמובן שהרעיון עובד גם למינימום. העניין החשוב הוא האילוצים (שנדרש שוויון ואי-שליליות).
וראינו שאפשר לעבור מתצורה אחרת לתצורה הזו.

נרצה ללמוד איך נראה פיתרון אופטימלי. נציג את הרעיון של **פיתרון בסיסי** – *basic solution*.

בהינתן $I \subseteq [n]$ (קבוצה שמייצגת אינדקסים של עמודות), נגדיר A_I את המטריצה המתקבלת ע"י לקיחת רק העמודות של A לפי I.

וקטור $x \in \mathbb{R}^n$ שמקיים $Ax = b$ ייקרא בסיסי אם קיים $B \subseteq [n]$ כך ש $|B| = m$, וגם:

A_B היא מטריצה לא-סינגולרית (העמודות שלה בת"ל), ו- $x_j = 0$ לכל $j \notin B$.

נשים לב שמכיוון של-A יש m שורות, אז A_B היא ריבועית.

נשים לב גם שפתרון בסיסי לא בהכרח מקיים אי-שליליות. אם הוא בנוסף מקיים, הוא ייקרא *basic feasible solution*, פתרון בסיסי ישים.

אבחנה טריוויאלית – אבחון מתי פתרון הוא בסיסי

בהינתן $x \in \mathbb{R}^n$ כך ש $x \geq 0$, $Ax = b$: x יהיה בסיסי **אמ"מ** העמודות של A_K בת"ל עבור $K \coloneqq \{j \in [n] : x_j > 0\}$.

כלומר, ניקח את העמודות של A רק באינדקסים שבהם x גדול ממש מ-0. אם העמודות האלה בת"ל, הפיתרון בסיסי.

הוכחה: זה מאוד דומה להגדרה עצמה.

<u>כיוון ראשון</u>: נניח ש-x בסיסי, כלומר קיים $|B| = m$, $B \subseteq [n]$ כך ש A_B לא סינגולרית ו- $x_j = 0$ לכל $j \notin B$.

מכיוון ש- $x \geq 0$, אז כל אינדקס שהוא לא 0 יהיה גדול ממש מ-0. נקבל ש $K \subseteq B$, אז העמודות של A_K יהיו בת"ל.

<u>כיוון שני</u>: נניח שהעמודות של A_K בת"ל עבור $K \coloneqq \{j \in [n] : x_j > 0\}$.

נרחיב את העמודות של A_K לקבוצה בת"ל מקסימלית בעמודות של A. יש m עמודות כאלה כי $rank(A) = m$ לפי הנחה.

נסמן B את הקבוצה הזו. מכיוון ש $x \geq 0$, ו-K זה כל המקומות שבהן $x_j > 0$, אז כל המקומות של x שהן לא ב-K הן 0.

ומכיוון ש $K \subseteq B$, נקבל ש $x_j = 0$ לכל $j \notin B$.

האבחון הזה נותן לנו אלגוריתם לבדיקה מתי פיתרון ישים הוא בסיסי:

<u>קלט</u>: $x \in \mathbb{R}^n$ כך ש $x \geq 0, Ax = b$.   <u>פלט</u>: 0 או 1, האם הפיתרון בסיסי.

נגדיר $K \coloneqq \{j \in [n] : x_j > 0\}$. אם העמודות של A_K בת"ל, נחזיר 1. אחרת, 0.

אנחנו רוצים פתרונות בסיסיים, כי אין הרבה כאלו. זה מקטין את המרחב שבו צריך לחפש פיתרון.

טענה: לכל קבוצה $|B| = m$, $B \subseteq [n]$, יש לכל היותר פתרון ישים בסיסי אחד שמתאים לה.

כלומר, אם נמצא פיתרון בסיסי לפי B, זה הפיתרון הבסיסי היחיד לפי ה-B הזו.

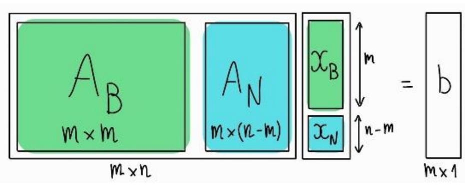
הוכחה: בהינתן $|B| = m$, $B \subseteq [n]$, נגדיר $N \coloneqq [n] \setminus B$.

כל פיתרון ישים $x \in \mathbb{R}^n$ מקיים $Ax = b$.

אם x מתאים ל-B, נוכל לכתוב:

$$A_B x_B + A_N x_N = b, \qquad and, \qquad x_j = 0 \ \ \forall j \notin B$$

כאשר $x_B \in \mathbb{R}^m$ הוא הווקטור x רק באינדקסים של B, ו- $x_N \in \mathbb{R}^{n-m}$ הוא x רק באינדקסים של N.



11: Linear Programming II

כמובן שהעמודות יכולות לשנות מקום, הסדר שלהן לא משנה.

אם x מתאים ל-B, אז כל הקורדינטות שלא ב-B הן 0, כלומר $x_N = 0$. אז מתקיים $A_B x_B = b$.

ומכיוון ש- A_B לא סינגולרית, יש ל- $A_B y = b$ לכל היותר פיתרון אחד.

אז יש לכל היותר $\binom{n}{m}$ פתרונות ישימים בסיסיים. אבל, זה לא מבטיח שאחד מהפתרונות האלה הוא **אופטימלי.**

טענה: תהי בעיית LP: $\max\{c^T x : x \geq 0, Ax = b\}$ ישימה וחסומה. אזי, לכל פתרון ישים x_0 קיים פתרון ישים בסיסי $\tilde{x}$ כך ש: $c^T \tilde{x} \geq c^T x_0$.

אם ניקח x_0 אופטימלי, לפי הלמה נקבל $\tilde{x}$ אופטימלי.

זה נותן לנו אלגוריתם בדיד (אבל אקספוננציאלי) לפתירת כל בעיית LP (מתוך ה-6 צורות שראינו):

1. נעבור לתצורת שוויון.
2. לכל $B \subseteq [n], |B| = m$ כך ש- A_B לא סינגולרית,
   a. ננסה לפתור את $A_B x = b$,
   b. אם יש פתרון נסמן את ה-B הזה **כמועמד**.
3. נבחר את הפיתרון הטוב ביותר מבין המועמדים.

הגיאומטריה של תכנון לינארי

מה שבעצם קורה מאחורי הקלעים של LP, היא שכל האילוצים מגדירים פוליהדרון (פאון, *polyhedron*) קמור (*convex*). בתמונה, השמאלי קמור:

כל פיתרון ישים בסיסי יהיה בקצוות, ובפרט בקודקודים. כלומר האלגוריתמים שלנו ינסו להתחיל מפיתרון ישים כלשהו (על אחת הפאות), ולפי הטענה לעיל (עם קצת שינויים) ימירו את הפיתרון בפתרונות ישימים בסיסיים עד שיגיעו לפינה. ואז יקחו את הפינה הכי טובה.

יש שני אלגוריתמים קלאסיים, שעוברים על הפוליגון בצורה חכמה (כמובן שהפוליגון לא באמת קיים, אבל זה עוזר לנו לדמיין):
אלגוריתם *Simplex*, ואלגוריתם *Ellipsoid*.
יש הוכחה ש-*Ellipsoid* הוא פולינומי. *Simplex* הוא לא בהגדרה פולינומי, אבל בכל מקרה פרקטי הוא הרבה יותר פשוט (צריך לבנות את הבעיות האלה בצורה מאוד ספציפית ומלאכותית כדי שתהיה בעיה שגורמת ל-*simplex* להיות מעריכי). ולכן הוא יותר נפוץ.

דואליות

בעיות LP מגיעות בזוגות: ***Primal LP***, ***Dual LP***.

כבר ראינו את הרעיון של דואליות: במשפט קניג, ראינו שהשידוך המקסימום שווה לכיסוי המינימום (בגרף דו"צ).

במשפט מנגר, ראינו שהחתך המינימום בין קודקודים שווה למספר המקסימום של מסלולים זרים בקודקודים ביניהם.

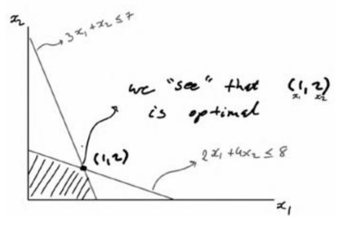
נראה דרך טכנית ויחסית פשוטה לקבל את הבעיה הדואלית של בעיה *primal* נתונה. לפני כן, נשאל מה התפקיד של דואליות ולמה זה עוזר:

1. עוזר להוכיח אופטימליות של פתרונות ישימים בתכנית המקורית.
2. עוזר לתכנן אלגוריתמים שמבוססים על LP אבל לא משתמשים ב *simplex* או *ellipsoid* שהם כבדים ויקרים. (דיקסטרה עובד ככה).

דוגמה

נשקול את ה-LP הבא: $\max(5x_1 + 5x_2),\ 3x_1 + x_2 \leq 7,\ 2x_1 + 4x_2 \leq 8,\ x_1, x_2 \geq 0$

אפשר לראות באיור איפה הנקודה האופטימלית:



איך נוכיח שזה אכן הפיתרון האופטימלי?

עבור 1,2, פונקציית המטרה נותנת: $5x_1 + 5x_2 = 5 \cdot 1 + 5 \cdot 2 = 15$. כל פיתרון ישים מקיים: $3x_1 + x_2 \leq 7,\ \ 2x_1 + 4x_2 \leq 8$.

אם נחבר את המשוואות האלה, נקבל בדיוק $5x_1 + 5x_2 \leq 15$, כלומר הסכום שלהם לא יכול לעבור את 15. אז 1,2 זה פתרון אופטימלי.

אבל זה עובד רק במקרה הזה. אם המשוואות של האילוצים לא נסכמות בדיוק לפונקציית המטרה, זה לא עובד. לדוגמה:

$$\max(7x_1 + 4x_2), \qquad 3x_1 + x_2 \leq 7, \qquad 2x_1 + 4x_2 \leq 8, \qquad x_1, x_2 \geq 0$$

התשובה היא, שכל פתרון ישים מקיים:

$$3x_1 + x_2 \leq 7 \overset{\times 2}{\Longrightarrow} 6x_1 + 2x_2 \leq 14, \qquad 2x_1 + 4x_2 \leq 8 \overset{\times 0.5}{\Longrightarrow} x_1 + 2x_2 \leq 4$$

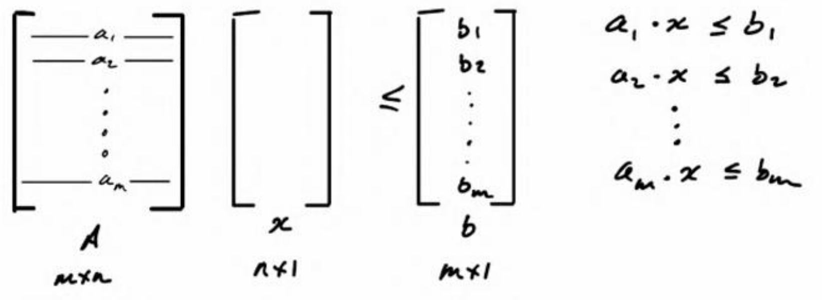
כלומר, כל פתרון ישים מקיים:

$$7x_1 + 4x_2 \leq 18$$

וזה נותן לנו ש- 2,1 אופטימלי. (ולא 1,2)

הכללה של הדוגמאות

נדגים את ההכללה על LP מהצורה: $\max\{c^T x : Ax \leq b\}$.



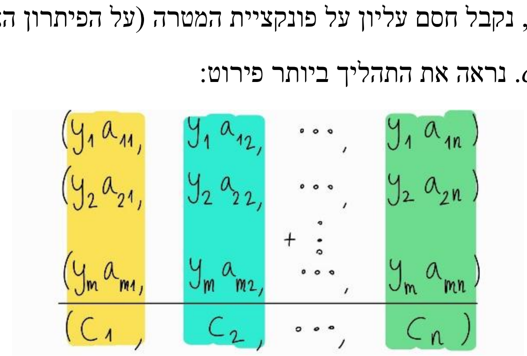
כל שורה $a_i \cdot x \leq b_i$ היא אילוץ של ה-LP הראשונית. נכפיל כל אילוץ במשתנה דואלי $y_i \geq 0$.

אחרי כל ההכפלות נראה שסכום האילוצים הוא פונקציית המטרה:

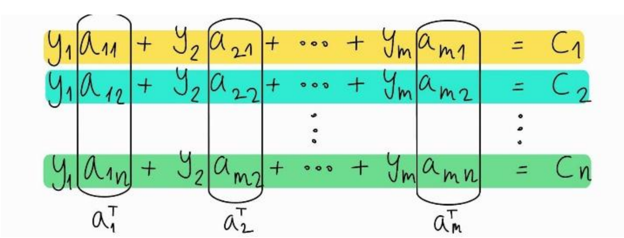
$$\begin{array}{rc} & y_1 \cdot (a_1 \cdot x \leq b_1) \\ & y_2 \cdot (a_2 \cdot x \leq b_2) \\ + & \vdots \\ & y_m \cdot (a_m \cdot x \leq b_m) \\ & = \\ & c^T x \leq y^T b \end{array}$$

זה מה שהיינו רוצים לקבל. אם נצליח למצוא y כזה, נקבל חסם עליון על פונקציית המטרה (על הפיתרון האופטימלי).

אנחנו מכפילים את $a_i \cdot y_i$, ומקבלים את הווקטור c. נראה את התהליך ביותר פירוט:



כל עמודה נסכמת ל- c_j. נכתוב בתור שורות:



בסה"כ, כל העמודות כאן נותנות את A^T. זה בעצם מבטא את $A^T y$. ואנחנו רוצים למצוא y כך ש $A^T y = c$.

11: Linear Programming II

נשים לב שעבור: $A \in M_{m\times n}(\mathbb{R}),\ c \in \mathbb{R}^n,\ b \in \mathbb{R}^m,\ 0 \le y \in \mathbb{R}^m$. אם $A^Ty = c$ וגם $Ax \le b$, אז $c^Tx \le y^Tb$.

כלומר, אם מצאנו y מתאים, אז פונקציית המטרה c^Tx חסומה ב- y^Tb לכל וקטור אפשרי ($Ax \le b$).

הוכחה: אנחנו יודעים ש $b \ge Ax$, אז $b^Ty \ge (Ax)^Ty$. ואנחנו יודעים ש $(Ax)^Ty = x^TA^Ty$ (חוקי כפל מטריצות ושחלוף).

וכבר אמרנו ש $A^Ty = c$, אז בסה"כ: $b^Ty \ge (Ax)^Ty = x^TA^Ty = x^Tc$.

משפט הדואליות החלש – *The Weak Duality Theorem*

אם שני הבעיות LP הן ישימות וחסומות, אזי:

$$\max\{c^Tx : Ax \le b\} \le \min\{y^Tb : A^Ty = c,\ \ y \ge 0\}$$

אנחנו מנסים לחסום את הבעיה הראשונית ע"י הבעיה הדואלית. במעבר מהראשוני לדואלי, כל אילוץ מקבל משתנה דואלי y_i כך ש:

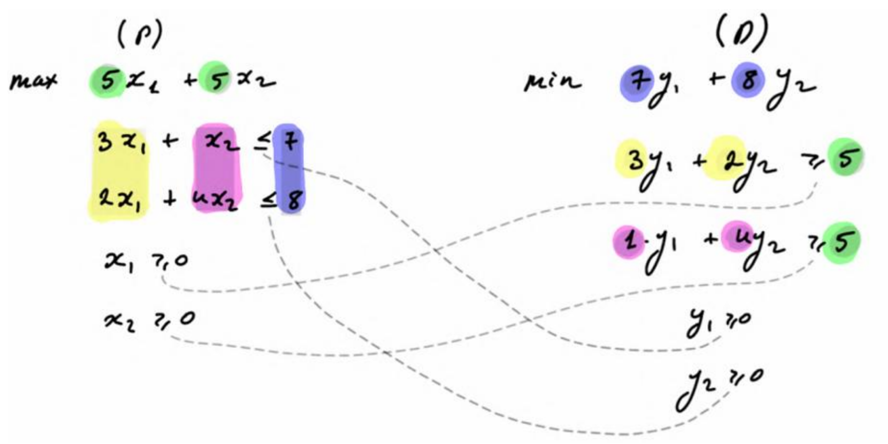
$$\begin{aligned} &primal, && dual \\ &\langle a_i, x\rangle \le b_i \implies && y_i \ge 0 \\ &\langle a_i, x\rangle = b_i \implies && y_i \in \mathbb{R} \\ &\langle a_i, x\rangle \ge b_i \implies && y_i \le 0 \end{aligned}$$

נשים לב שאם האילוץ הראשוני היה שוויון, במשתנה הדואלי אין אילוץ.

במעבר הזה, כל משתנה ראשוני נהיה אילוץ דואלי:

$$\begin{aligned} &primal, && dual \\ &x_i \ge 0 \implies && y_1a_{1i} + y_2a_{2i} + \cdots + y_ma_{mi} \ge c_i \\ &x_i \le 0 \implies && y_1a_{1i} + y_2a_{2i} + \cdots + y_ma_{mi} \le c_i \\ &x_i \in \mathbb{R} \implies && y_1a_{1i} + y_2a_{2i} + \cdots + y_ma_{mi} = c_i \end{aligned}$$

דוגמה:



לדואלי יש בעצמו דואלי. הדואלי של הדואלי הוא הראשוני.

משפט הדואליות החזק

יהיו (P) ו-(D) ראשוני והדואלי שלו. יש בדיוק 4 אופציות, ובדיוק אחת מהן אפשרית. אנחנו נעסוק רק ברביעית.

1. שניהם לא ישימים,
2. הראשוני ישים והדואלי לא,
3. הראשוני לא ישים והדואלי כן,
4. שניהם ישימים, חסומים, ויש חפיפה בין החסימות – כלומר המקסימום שווה למינימום, יש שוויון ולא רק חסימה.

הגדרות

האיברים של $[n]$ מייצגים יחידות (או נקודות) מידע. זה לא חייב להיות במרחב אוקלידי – צריך רק שיהיה מושג של מרחב בין 2 נקודות, ושיתקיים:

$D := (d_{ij}) \in M_{n\times n}$ היא מטריצת המרחקים. d_{ij} זה המרחק בין נקודה i לנקודה j.

לכל $i,j,k \in [n]$ צריך להתקיים: $d_{ij} \geq 0$, וגם $d_{ij} + d_{jk} \geq d_{ik}$.

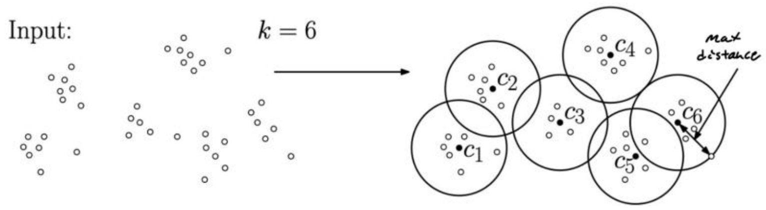
אפשר לראות את D בתור גרף מלא, שקודקודיו הם $[n]$, והמשקל של כל צלע ij הוא d_{ij}.

עבור $i \in [n], S \subseteq [n]$, נגדיר $d(i,S)$ את המרחק של הנקודה i מהקבוצה S: $\min_{j\in S} d_{ij}$. אם $i \in S$, אז $d(i,S) = 0$.

The K-centers Problem

בהינתן מספר $k \in \mathbb{N}$, קבוצה $[n]$, ומטריצת מרחקים $D := (d_{ij})$ שמקיימת את אי"ש המשולש,

נרצה למצוא $S \subseteq [n], |S| = k$ שממזערת את $\max_{i\in[n]} d(i,S)$.



נהוג לקרוא ל-$\max_{i\in[n]} d(i,S)$ **הרדיוס** של S. הנקודות של S הן *centroids* שמסביבן נרצה לקבץ את שאר הנקודות.

בעיית *k-center* היא *NPH*. להלן פסודו-קוד לאלגוריתם חמדן:

1. נתחיל עם S שמכילה קודקוד שרירותי.
2. כל עוד $|S| < k$, נמצא את הקודקוד שהכי רחוק מ-S ונוסיף אותו ל-S.

טענה: האלגוריתם הזה הוא 2-מקרב עבר בעיית *k-center*.

אלגוריתם *f*-מקרב

תהי בעיית מינימיזציה Π כלשהי, ויהי A אלגוריתם עבור הבעיה. האלגוריתם ייקרא f-קירוב (*f-approximation*) עבור Π אם:

$$A(I) \leq f \cdot OPT(I)$$

לכל מופע I של Π. כלומר, אם הערך ש-A מחזיר הוא לכל היותר f פעמים הערך האופטימלי.

למה הקירוב כפלי (כפול f) ולא חיבורי (ועוד f)? כי אין כמעט בעיות שעומדות בתנאי הזה, אז זה לא שמיש.

אותה הגדרה עובדת עבור בעיית מקסימום, אבל f יהיה שבר כלשהו (קטן מ-1) והסימן יהיה הפוך: $A(I) \geq f \cdot OPT(I)$.

אז עבור בעיית *k-center*, נאמר שהאלגוריתם הוא 2-מקרב. כלומר, אם:

$$OPT(I) := \min_{S\subseteq[n],|S|=k}\left(\max_{i\in[n]}\big(d(i,S)\big)\right)$$

אנחנו טוענים שאם $S \subseteq [n], |S| = k$ היא הפלט של האלגוריתם החמדן, אז:

$$\max_{i\in[n]}\big(d(i,S)\big) \leq 2 \cdot OPT(I)$$

כלומר המרחק המקסימום מהקבוצה לנקודה כלשהי, הוא לכל היותר פעמיים המרחק המקסימום שמתקבל באלגוריתם האופטימלי.

הוכחה: תהי $S^* := \{j_1, j_2, \dots, j_k\} \subseteq [n]$ שהיא פתרון אופטימלי. נסמן $r^* := \max_{i\in[n]}\big(d(i,S^*)\big)$ את הרדיוס של הפיתרון.

הקבוצה S^* מגדירה חלוקה על הדאטא: $V_1 \cup V_2 \cup \dots \cup V_k = [n]$ כך:

נשים נקודה $\ell \in [n]$ ב- V_i אם מתוך $\{j_1, j_2, \dots, j_k\}$, היא הכי קרובה ל- j_i. (אם זה תיקו נבחר שרירותית ביניהם).

אבחנה: נביט במקבץ (*cluster*) ה-i עבור $i \in [k]$, ויהיו $x, y \in V_i$. אזי, $d_{xy} \leq 2 \cdot r^*$. למה? כי:

$$d_{xy} \leq d_{xj_i} + d_{j_iy} \leq 2r^*$$

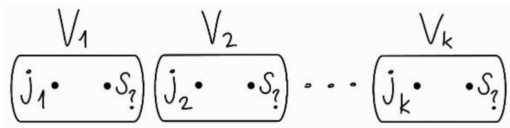
המרחק בין x ל-y הוא לכל היותר סכום המרחקים בין x, y למרכז של המקבץ. שהוא לכל היותר פעמיים הרדיוס.

אז המרחק בין 2 נקודות בתוך מקבץ, הוא לכל היותר פעמיים הרדיוס.

יהי $S \coloneqq \{s_1, s_2, \dots, s_k\} \subseteq [n]$ הפתרון החמדן, ונסמן $r \coloneqq \max_{i\in[n]}\big(d(i,S)\big)$ את הרדיוס של הפתרון החמדן. אנחנו רוצים להוכיח: $r \leq 2 \cdot r^*$.

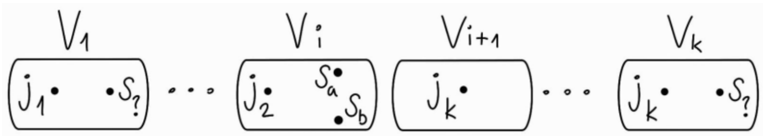
נחלק לשני מקרים:

<u>במקרה הראשון</u>: $|S \cap V_i| = 1$ לכל $i \in [k]$. כלומר, בכל קבוצה לפי החלוקה של האופטימלי, יש נקודה אחת מהמרכזים של החמדן:



במקרה הזה, לכל נקודה שנבחר, היא נמצאת במקבץ עם נקודה מ-S. אז לפי האבחנה לעיל, הטענה מתקיימת.

<u>במקרה השני</u>, קיים $i \in [k]$ כך ש $|S \cap V_i| \geq 2$. כלומר יש קבוצה אחת בחלוקה של האופטימלי, שיש בה לפחות 2 מרכזים של החמדן:



יהיו $s, s' \in S$ בתוך V_i.

בה"כ, החמדן בחר קודם את s ואז את s'.

בגלל שהן באותו מקבץ, לפי האבחנה לעיל, $d_{ss'} \leq 2 \cdot r^*$.

כאשר s' נבחר ע"י החמדן, הוא היה הנקודה הרחוקה ביותר מ-S הנוכחית. ו-s כבר הייתה ב-S.

כלומר, כל שאר הנקודות הן במרחק לכל היותר $2r^*$ מ-S, כנדרש.

The k-suppliers Problem

בהינתן $A \cup B \coloneqq [n]$, כך ש $|A| \geq k$ עבור $k \in \mathbb{N}$ כלשהו. A מייצגת שירותים או ספק כלשהם, ו-B היא הלקוחות.

נרצה למצוא $S \subseteq A, |S| = k$ שממזערת את $\max_{b\in B}\big(d(b,S)\big)$.

נציע את האלגוריתם הבא:

עבור לקוח $b \in B$, נסמן $a_b \in A$ את הספק הכי קרוב.

נפעיל את *k-center* (אלגוריתם 2-מקרב) על B. תהי S' הקבוצה המוחזרת.

נחזיר את $S \coloneqq (a_b : b \in S')$.

אם $|S| < k$, נוסיף ל-S נקודות באקראי.

כלומר, נקבל k צרכנים שהם פיזור טוב בין שאר הצרכנים, וניקח את הספק שהכי קרוב לכל אחד מהם.

טענה: האלגוריתם המוצע הוא 3-מקרב.

הוכחה: נסמן $r^* \coloneqq \min_{S^*\subseteq A, |S^*|=k}\left(\max_{b\in B}\big(d(b,S^*)\big)\right)$ את הרדיוס האופטימלי עבור *k-suppliers*.

בפרט, אנחנו מניחים ש**קיימת** קבוצה $Y \subseteq A, |Y| = k$ כך שכל נקודה ב-B קרובה אליה עד כדי r^* (זה הפיתרון האופטימלי).

ניקח $b \in B$ כלשהו. צ"ל: $d(b,S) \leq 3r^*$.

מקרה ראשון: קיים $b' \in S'$ כך ש: $d_{bb'} \le 2r^*$. (כלומר לכל b שנבחר, יש b' ב-S' שקרוב אליו עד כדי $2r^*$).

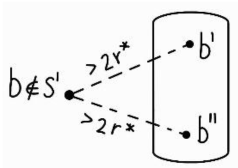
אזי לפי אי"ש המשולש, מתקיים:

$$d_{ba_{b'}} \le d_{bb'} + d_{b'a_{b'}} \le 3r^*$$

כי $d_{bb'} \le 2r^*$ לפי ההנחה. ו- $d_{b'a_{b'}} \le r^*$ כי $a_{b'}$ היא הנקודה הכי קרובה ל- b' מתוך כל A. אז $d(b,S) \le 3r^*$, כנדרש.

מקרה שני: $d_{bb'} > 2r^*$ לכל $b' \in S'$. אזי, בפרט $b \notin S'$. ניזכר שהחמדן תמיד לקח את הנקודה הכי רחוקה מ-S' הנוכחית.

אם $b \notin S'$ וגם $d(b,S') > 2r^*$:



אז זה אומר שהמרחק בין כל 2 נקודות ב-S' הוא יותר מ- $2r^*$. למה?

כי כשבחרנו את b'', היא הייתה הנקודה הרחוקה ביותר מכל שאר הנקודות ב-S'.

אז בפרט היא הייתה רחוקה יותר מ- b' מאשר b.

נשים לב שמתקיים: $|S' \cup \{b\}| = k+1$. וגם, בקבוצה הזו, כל שתי נקודות הן במרחק לפחות $2r^*$.

כי המרחק בין כל 2 נקודות ב-S' הוא יותר מ- $2r^*$, וגם b רחוקה לפחות $2r^*$ מכל נקודה ב- S'.

כלומר, קיימת קבוצה $X \subseteq B, |X| = k+1$ כך ש: $d_{uv} > 2r^*$ לכל $u, v \in X$.

(קבוצה ב-B שכל האיברים שלה רחוקים לפחות $2r^*$ אחד מהשני).

אזי, אין $Y \subseteq A, |Y| = k$ שיכולה לתת איבר a שקרוב עד כדי r^* לכל $b \in X$. למה?

נב"ש שיש קבוצה Y כזו. מכיוון ש $|X| = k+1$, אז לפחות שני איברים של X, נגיד $u,\ v$ מקיימים:

$$d(u,Y) = d_{ua}, \qquad d(v,Y) = d_{va}$$

כלומר המרחק שלהם מ-Y נקבע לפי אותה נקודה ב-Y, כי $|Y| = k$ (שובך היונים).

וגם, $d_{ua}, d_{va} \le r^*$ כי הנחנו ש-Y יכולה לתת שירות ל-X. ובסה"כ, נקבל:

$$d_{uv} \le d_{ua} + d_{av} \le 2r^*$$

סתירה.

כלומר אין קבוצה ב-A שנותנת נקודה שקרובה עד כדי r^* לכל נקודה ב-B. סתירה להנחה מההתחלה – בעצם הראינו שהפתרון האופטימלי לא אפשרי.

שזו סתירה, כי התחלנו בהנחה שהפתרון האופטימלי נותן רדיוס r^*.

כלומר המקרה השני לא אפשרי, אז רק המקרה הראשון מתקיים. מש"ל.

בעיית הסוכן הנוסע המטרית

הקבוצה $[n]$ מייצגת ערים. יש מטריצת עלויות סימטרית: $C \coloneqq (c_{ij} \geq 0)$. $c_{ii} = 0$ לכל i.

הבעיה נקראת ה-*TSP* ***metric*** בגלל ההנחה שהמטריצה מקיימת את אי"ש המשולש. אפשר לראות את הבעיה בתור גרף ממושקל מלא.

המטרה היא למזער את העלויות (או המשקל) של מעגל המילטון בגרף.

אבחנה מרכזית: אם ניקח את המעגל של ה-*TSP* (להלן *tour*, סיור), נקבל עץ פורש.

ולכן, המשקל של הסיור האופטימלי הוא לפחות המשקל של העץ פורש מינימום.

Nearest Extension Algorithm

אתחול:

1. ניקח את $i, j \in [n]$, שני הקודקודים הכי קרובים.
2. נגדיר את הסיור להיות הגרף המכוון: $T \coloneqq i \longrightarrow j$.

צעד *extension*. לא נוסיף קודקודים לפני i או אחרי j, אלא רק ביניהם:

3. נניח ש $V(T) \neq [n]$.

4. תהי $y \in [n] \setminus V(T)$ הנקודה שהכי קרובה למסלול, ו-x הנקודה שהגדירה את המרחק למסלול (הכי קרובה ל-y):
$$(x, y) \coloneqq argmin\{c_{xy} : x \in V(T), \ \ y \notin V(T)\}$$
אפשר לחשוב על זה כך: הגדרנו חתך שבו צד אחד זה $V(T)$, וכל פעם ניקח את הקודקוד שמחובר ע"י הצלע הקלה ביותר שחוצה את החתך.

5. "נבלע" (*absorb*) את y בצורה הבאה:
נסתכל על הקודקוד הבא במסלול אחרי x (נגיד z), ונעשה עיקוף: במקום $x \to z$, נעשה $x \to y \to z$.
נשים לב: לכאורה אין לנו שליטה על המשקל שמוסיפים עם yz. אבל, נוכל לרסן את זה עם אי"ש המשולש.
   a. אם $x = j$, ניקח את הקודקוד שלפניו במקום אחריו.

סיום:

6. כאשר $V(T) = [n]$, יש לנו סיור T שמתחיל ב-i ועובר בכל הקודקודים ומסתיים ב-j.
נזכור, שב-*extension* הראשון זרקנו את הצלע ij. אז נחזיר את הצלע ji.

טענה: האלגוריתם הזה נותן 2-קירוב עבור בעיית *Metric-TSP*.

הוכחה: יהי G הגרף הממושקל המלא, ונסמן F' את קבוצת הצלעות שהחמדן בחר (הצלע שהייתה מינימלית בחתך בכל שלב).

נגדיר $F \coloneqq F' \cup \{ij\}$. נשים לב ש $([n], F)$ הוא עפ"מ (לפי תהליך הבנייה, אלגוריתם פרים – *prim*).

כלומר, כל צלע של F' היא צלע מינימלית של חתך. הגודל של F' הוא $|F'| = n - 2$, אז $|F| = n - 1$.

וקיימת הרצה של פרים לפי הסדר של הצלעות שהאלגוריתם בחר, שתיתן את F בתור עפ"מ.

ולכן, אם נסמן *OPT* את העלות של הסיור האופטימלי, נקבל:

$$OPT \geq \sum_{f \in F} c(f)$$

כי אנחנו יודעים שהמשקל של הסיור האופטימלי הוא לפחות המשקל של העץ פורש המינימום.

יהי $T + ij$ הסיור שמוחזר מהאלגוריתם. הסיור הזה התקבל ע"י סדרת מעקפים.

במעקף $u \to v \implies u \to w \to v$, העלות עלתה ב:

$$+c_{uw} + c_{wv} - c_{uv} \geq 0$$

בגלל אי"ש המשולש, זה לפחות 0. ואנחנו יודעים גם:

$$c_{wv} \leq c_{wu} + c_{uv}$$

אז נוכל לרשום:

$$increase \coloneqq c_{uw} + c_{wv} - c_{uv} \le c_{uw} + \underbrace{c_{wu} + c_{uv}}_{\le c_{wv}} - c_{uv} = 2c_{wu}$$

וניזכר ש: $uw \in F'$, ושהיא הייתה בחירה חמדנית. כלומר העלות של כל הסיור היא: לכל היותר, העלות של כל המעקפים ב- F', ועוד c_{ij}:

$$cost\ of\colon\ T + ij \le [cost\ of\ all\ bypasses\ in\ F'] + c_{ij} \le c_{ij} + \sum_{f \in F'} 2 \cdot c(f) \le$$

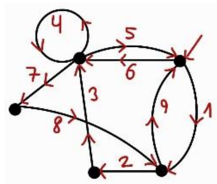
נכפיל את c_{ij} ב-2 כדי להכניס אותו גם לסכום.

כזכור, הצלעות של F מהוות MST. ואמרנו שה-MST האופטימלי הוא לכל היותר ה-$tour$ האופטימלי (נסמן OPT), כלומר נוכל לרשום:

$$\le 2 \cdot \underbrace{\sum_{f \in F} c(f)}_{MST} \le 2 \cdot OPT$$

Doubling Trees Algorithm

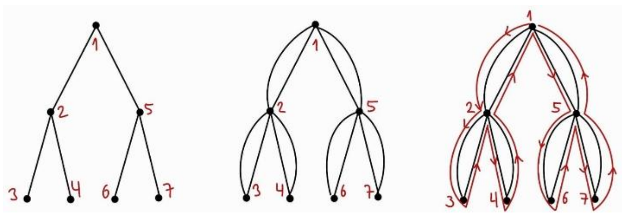
תזכורת: **מעגל אוילר** (*Euler tour*) על מולטיגרף הוא הילוך סגור שעובר על כל צלע בדיוק פעם אחת:



גרף שיש בו מעגל כזה נקרא אוילרי, או אוילריאני. מציאת מסלול אוילר היא בעיה ב-P.

משפט: מולטיגרף קשיר יהיה אוילרי **אמ"מ** לכל הקודקודים יש דרגה זוגית. אינטואיטיבית, כי צריך לצאת ולהיכנס מכל קודקוד פעם אחת.

מה הקשר בין בעיית TSP למעגל אוילר? אבחנה: אם T הוא עץ, אז הכפלת כל צלע נותנת לנו גרף אוילרי:



כי אין מעגלים, והגרף קשיר. אז אם מכפילים כל צלע, לכל קודקוד יש דרגה זוגית. זה נותן לנו את האלגוריתם הבא למציאת סיור:

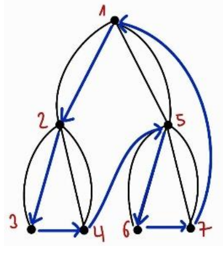
1. ניקח את T, עפ"מ על G.
2. נכפיל כל צלע ב-T (עם המשקלים), ויהי $\mathcal{T}$ המולטיגרף המתקבל.
3. נמצא מעגל אוילר ε ב- $\mathcal{T}$.
4. נבנה סיור TSP מתוך ε ע"י $shortcutting$.

ביצוע $shortcutting$ (קיצור דרך):

ניקח את הקודקודים לפי הסדר שהם מופיעים ב-ε בפעם הראשונה, והקודקוד הראשון חוזר על עצמו. בדוגמה לעיל יש לנו:

$$\varepsilon = (\bar{1}, \bar{2}, \bar{3}, 2, \bar{4}, 2, 1, \bar{5}, \bar{6}, 5, \bar{7}, 5, 1) \implies T \coloneqq (1, 2, 3, 4, 5, 6, 7, 1)$$

ונקבל את הסיור:



טענה: האלגוריתם $doubling\ trees$ הוא 2-מקרב.

הוכחה: נגדיר: OPT את העלות של סיור TSP אופטימלי, L את הסיור המתקבל מהאלגוריתם, T את ה-MST בתהליך ההרצה, ו-$\mathcal{T}$ את העץ ה"מוכפל". מתקיימים:

$$c(\mathcal{T}) = 2 \cdot c(T) \leq 2 \cdot OPT$$

כי $\mathcal{T}$ זה פשוט T עם כל הצלעות פעמיים, כולל המשקלים. ואנחנו יודעים שמשקל ה-MST חסום בעלות הסיור האופטימלי.

בגלל שהמשקלים מקיימים את אי"ש המשולש, כל קיצור דרך לא מעלה את העלות. אז המשקל הסופי של הסיור הוא לכל היותר מה שהתחלנו איתו. התחלנו עם $\mathcal{T} \leq 2 \cdot OPT$, אז המשקל הסופי הוא גם לכל היותר $2 \cdot OPT$.

Christofides Algorithm

בניית גרף אוילר ע"י הכפלת כל הצלעות של MST זה תהליך בזבזני. הקודקודים הבעייתיים הם רק אלה עם דרגה אי"ז.

אבחנה: בכל גרף, מספר הקודקודים עם דרגה אי"ז הוא תמיד זוגי.

אז אם G' גרף קשיר, ו-M הוא שידוך מושלם של <u>קודקודי G' שבעלי דרגה אי"ז</u> (עם צלעות שאולי לא ב- $E(G')$! יכול להיות שנוסיף צלעות).

אז זה בעצם הוסיף 1 לכל קודקוד אי"ז, ועכשיו לכל הקודקודים יש דרגה זוגית, והגרף $G' + M$ הוא אוילרי.

לפי זה, נקבל אלגוריתם:

1. יהי T עפ"מ של G. (ה-T הזה יהיה ה- G' מהטענה לעיל).
2. תהי O קבוצת הקודקודים ב-T שבעלי דרגה אי"ז.
3. יהי M שידוך מושלם בעל מחיר מינימום בגרף $G[O]$ (הגרף המלא על הקודקודים ב-O). מציאת ה-M המינימום היא בעיה ב-P.
4. נוסיף את M ל-T כדי לקבל את $\mathcal{T}$, שהוא אוילרי לפי הטענה.
5. נמצא סיור אוילר ε ב- $\mathcal{T}$.
6. נבנה סיור TSP מ- ε ע"י קיצורי דרך.

טענה: האלגוריתם הוא 1.5-מקרב.

למה: יהי OPT המחיר של סיור TSP אופטימלי. תהי קבוצה $W \subseteq V(G)$ כך ש- $|W|$ זוגית.

ויהי M שידוך מושלם במחיר מינימום ב- $G[W]$.

אזי, $c(M) \leq OPT/2$.

הוכחה: נוכיח את הטענה על האלגוריתם תחת ההנחה שהלמה נכונה. ואז נוכיח את הלמה ונסיים.

יהי L הסיור המתקבל מהאלגוריתם. אזי, העלות של L היא לכל היותר כל ה-MST, ועוד הצלעות שהוספנו בשידוך (בגלל אי"ש המשולש).

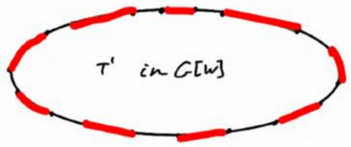
כלומר לפי הלמה ($c(M) \leq OPT/2$) ולפי הטענה מההתחלה ($c(T) \leq OPT$) נקבל:

$$c(L) \leq c(T) + c(M) \leq OPT + OPT/2 = 1.5 \cdot OPT$$

נוכיח את הלמה: יהי T סיור TSP אופטימלי, ויהי T' סיור TSP ב- $G[W]$ שהתקבל ע"י קיצורי דרך ב-T.

בגלל אי"ש המשולש, נקבל $c(T') \leq c(T)$.

מכיוון ש- $|W|$ זוגי, T' הוא מעגל זוגי, כלומר הוא איחוד של שני שידוכים מושלמים (זרים בצלעות) של $G[W]$:



מכיוון ש-M הוא במחיר מינימום ב- $G[W]$, המשקל שלו הוא לכל היותר חצי המשקל של T' (כי T' מורכב משני שידוכים): $c(M) \leq c(T')/2$.

אזי:

$$c(M) \leq c(T')/2 \leq c(T)/2 = OPT/2$$

כנדרש.

14: Approximating min VC

נזכיר: בגרף G, קבוצה $C \subseteq V(G)$ תיקרא **כיסוי בקודקודים** של G אם לכל צלע $uv \in E(G)$, מתקיים $u \in C$ או $v \in C$.

אלגוריתם אקראי

1. נאתחל $C = \emptyset$.
2. כל עוד $E \neq \emptyset$, נבצע:
   a. נבחר $e = uv \in E$ שרירותית.
   b. $C \coloneqq C + v$
   c. $E \coloneqq E \setminus \{e \in E : v \in e\}$. נוריד מ-$E$ את כל הצלעות שמחוברות ל-v.
3. נחזיר את C.

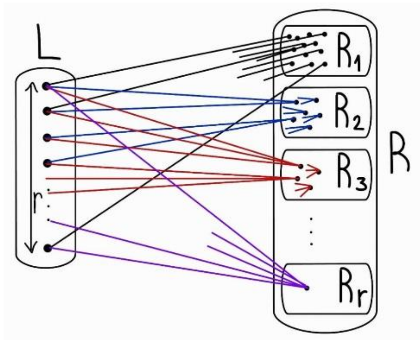
נשקול את הביצועים של האלגוריתם על הגרף הדו"צ הבא:

$$G = (L \cup R, E), \qquad R \coloneqq R_1 \cup R_2 \cup \ldots \cup R_r, \qquad r = |L|, \qquad \forall i \colon |R_i| = \lfloor r/i \rfloor$$

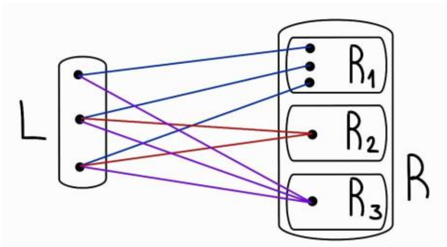
כלומר: יש r קודקודים ב-L, ו-R מחולקת ל-r קבוצות. הגודל של כל קבוצה i הוא r/i (מעוגל כלפי מטה).

בנוסף, בתוך כל R_i, אין שני קודקודים עם שכן משותף. ולכל קודקוד ב-R_i, יש i שכנים ב-L.

המבנה הכללי:



ודוגמה עבור $r = 3$: $|R_1| = 3/1 = 3,\ |R_2| = \lfloor 3/2 \rfloor = 1,\ |R_1| = \lfloor 3/3 \rfloor = 1$



כמה צלעות יהיו בגרף? נבדוק כמה קודקודים יהיו ב-R. לפחות:

$$\sum_{i=1}^{r} \lfloor r/i \rfloor = \Theta\left(r \cdot \sum_{i=1}^{r} 1/i\right) = \Theta(r \cdot \log r)$$

(חסם על הטור ההרמוני). מספר הקודקודים ב-R הוא חסם תחתון למספר הצלעות שיש בגרף. אז מספר הצלעות הוא ב- $\Omega(r \cdot \log r)$.

בגרף שלנו, L מהווה פתרון VC אופטימלי יחיד (אין עוד פתרון מינימום אחר).

אבל האלגוריתם יכול להחזיר את R (אם בכל פעם שהוא בוחר צלע שרירותית, הוא בוחר קודקוד מ-R). ואז, יחס הקירוב הוא:

$$\frac{|R|}{|L|} = \frac{\Omega(r \cdot \log r)}{r} = \Omega(\log r)$$

תלוי בגודל הגרף. זה יחס קירוב רע מאוד (כי r הוא פונקצייה של n).

אלגוריתם חמדן

האלגוריתם כמעט זהה לקודם, אבל במקום לבחור שרירותית, נבחר בצורה חמדנית:

1. נאתחל $C = \emptyset$.
2. כל עוד $E \neq \emptyset$, נבצע:
   a. נבחר $v \in V$ בעל דרגה מקסימום (אם יש יותר מאחד, נבחר ביניהם שרירותית).
   b. $C \coloneqq C + v$
   c. $E \coloneqq E \setminus \{e \in E : v \in e\}$. נוריד מ-$E$ את כל הצלעות שמחוברות ל-v.
3. נחזיר את C.

כל פעם נבחר קודקוד שיש לו כמה שיותר צלעות, אז ברעיון נכסה את כל הצלעות עם פחות קודקודים.

אבל עדיין, בגרף שהראינו, אפשר לגרום לאלגוריתם להחזיר את R. כי בבחירה הראשונה נוכל לבחור את הקודקוד ב-R_r,

ואז זה מוריד צלע מכל אחד מהקודקודים של L. ובכל פעם שבחרנו קודקוד מ-R_i כלשהו, לא הורדנו צלעות מאף R_j אחר.

אז בכל פעם נוכל לבחור קודקוד מ-R.

אלגוריתם 2-מקרב, מבוסס על שידוך מקסימום

ניזכר ששידוך מקסימום היא בעיה ב-P. וגם, שגודל הכיסוי המינימום $\tau(G)$ הוא לפחות גודל השידוך המקסימום $\nu(G)$.

אז נוכל למצוא שידוך מקסימום בגרף, ולקחת את הקודקודים של הצלעות של השידוך.

כי זה בוודאות מכסה את כל הצלעות (אם יש צלע שלא מכוסה, אפשר להוסיף אותה לשידוך).

מספר הצלעות הוא לכל היותר כגודל הכיסוי ($\nu(G) \leq \tau(G)$), ולקחנו 2 קודקודים לכל צלע.

כלומר לקחנו לכל היותר $2 \cdot \tau(G)$ קודקודים. פי 2 מגודל הכיסוי המינימום, כלומר 2-קירוב.

הבעיה באלגוריתם הזה, היא שהוא דורש את אלגוריתם *Edmonds* (מציאת שידוך מקסימום בגרף כללי), שהוא מסובך ויקר בזמן ריצה. (פולינומי, אבל עדיין לא משהו).

***Savage's Algorithm* – אלגוריתם 2-מקרב בזמן לינארי**

<u>קלט</u>: גרף G קשיר. (אם הוא לא קשיר, נפעיל את האלגוריתם על כל רכיב קשירות בנפרד).

1. יהי T עץ *DFS* של G.
2. נחזיר את הקודקודים הפנימיים של T (הכל חוץ מהעלים).

באופן טריוויאלי, כל הצלעות מכוסות. כי אין צלע בין שני עלים: כשביקרנו בעלה הראשון מביניהם, אם הייתה צלע היינו עוברים בה (ואז זה לא היה עלה).

כדי להוכיח שהוא גם 2-מקרב, מספיק להוכיח:

למה: יהי T עץ *DFS* של גרף קשיר, ונסמן $I(T)$ את הקודקודים הפנימיים. אזי, T מכיל שידוך בגודל לפחות $|I(T)|/2$.

אם הלמה נכונה, אז נקבל ש $\tau(G) \geq |I(T)|/2$, כי הכיסוי צריך לקחת לפחות קודקוד אחד מכל צלע בשידוך.

האלגוריתם מחזיר קבוצה בגודל $|I(T)|$, אז הוא 2-מקרב.

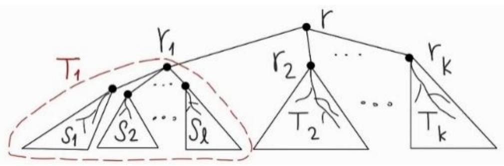
נוכיח את הלמה: באינדוקציה על כמות הצלעות. (נתעלם מעצים שהם רק קודקוד יחיד, כי אין צלעות).

<u>בסיס</u>: עבור $e(T) = 1$, נקבל: $|I(T)| = 0$, ואכן T מכיל שידוך בגודל $0 \leq 1$.

<u>צעד</u>: נניח שהטענה מתקיימת עבור כל תת-עץ ממש של T, ויהי T עץ *DFS* של גרף קשיר.

נסמן r את שורש העץ. ונסמן $T_1, T_2, \dots, T_k$ את תתי-העצים המושרשים בילדים של r. r_i הוא השורש של T_i.

נסמן $S_1, S_2, \dots, S_\ell$ את תתי-העצים של T_1 (מושרשים ב- r_1):



נחשב את גודל השידוך שיש ב-T:

בכל S_i, יש שידוך בגודל לפחות $|I(S_i)|/2$ (לפי הנ"א). כלומר ב- T_1 יש שידוך בגודל לפחות:

$$\frac{1}{2} \cdot \sum_{i=1}^{\ell} |I(S_i)|$$

בכל $T_2, T_3, \dots, T_k$ יש שידוך בגודל לפחות $|I(T_i)|/2$ (לפי הנ"א). כלומר ב- $T_2 \cup T_3 \cup \dots \cup T_k$ יש שידוך בגודל לפחות:

$$\frac{1}{2} \cdot \sum_{i=2}^{k} |I(T_i)|$$

ואנחנו יודעים שלא השתמשנו ב r או r_1. אז נוסיף את rr_1 לשידוך. בסה"כ קיבלנו שיש שידוך ב-T בגודל לפחות:

$$\geq 1+\frac{1}{2}\cdot\sum_{i=1}^{\ell}|I(S_i)|+\frac{1}{2}\cdot\sum_{i=2}^{k}|I(T_i)| = 1+\frac{\sum_{i=1}^{\ell}|I(S_i)|+\sum_{i=2}^{k}|I(T_i)|}{2}$$

ונשים לב ש: $\sum_{i=1}^{\ell}|I(S_i)|+\sum_{i=2}^{k}|I(T_i)|$ זה מספר הקודקודים הפנימיים בעץ, בלי r, r_1. אז:

$$\sum_{i=1}^{\ell}|I(S_i)|+\sum_{i=2}^{k}|I(T_i)| = |I(T)|-2$$

נציב את השוויון הזה בחישוב, ונקבל שב-T יש שידוך בגודל לפחות:

$$\geq 1+\frac{|I(T)|-2}{2}=\frac{2+|I(T)|-2}{2}=\frac{|I(T)|}{2}$$

כנדרש.

Minimum Weight Vertex Cover

בהינתן גרף G, **ופונקציית משקל** $w: V(G)\to\mathbb{R}^+$, על הקודקודים של G. נרצה למצוא כיסוי $C\subseteq V(G)$ שממזער את $\sum_{v\in C} w(v)$.

אף אחד מהאלגוריתמים שראינו לא עובד עם משקלים. באופן כללי, כי משקלים לא קשורים למבנה הגרף. נצטרך להיעזר בתכנון לינארי.

נתחיל ב- *Integer Programming*. נקודד את *min-w-VC* לבעיית *IP*, שהיא *NPC*:

יהיה לנו וקטור x שמייצג איזה קודקודים לוקחים לכיסוי. המיקום ה-v ב-x יהיה 1 אם $v\in C$, ו-0 אחרת. ואנחנו נרצה למזער את:

$$\sum_{v\in V(G)} w(v)\cdot x_v$$

תחת האילוצים: $x_u+x_v\geq 1$ לכל $uv\in E(G)$, ו- $x_v\in\{0,1\}$ לכל v.

נעבור ל-*relaxation* של הבעיה ל-*LP*. נקודד את *min-w-VC* לבעיית *LP*, שהיא *P*:

הכל אותו דבר, אבל האילוץ של x הופך להיות $x_v\geq 0$.

יכולנו לרשום $0\leq x_v\leq 1$, אבל זה אותו דבר.
כי אנחנו נרצה למזער את הסכום שתלוי ב-x, וברגע שיש יותר מ-1 הקודקוד כבר מכוסה (אז היה אפשר להגיע לאותו כיסוי עם 1).

האלגוריתמים *Simplex*, *Ellipsoid* יכולים למצוא פתרון אופטימלי בזמן פולינומי.

אבל איך יודעים שהפתרון שנמצא ב-*LP* פותר את הבעיה שתיארנו ב-*IP*?

נציע אלגוריתם שמבוסס על *LP*, למציאת כיסוי בקודקודים במשקל מינימום:

1. יהי x^* הפיתרון האופטימלי לבעיית ה-*LP*.
2. נחזיר: $C := \{v\in V(G) : x_v^*\geq 0.5\}$.

טענה: C היא אכן כיסוי בקודקודים.

הוכחה: יהי $uv\in E(G)$. אזי:

$$x_v^*+x_u^*\geq 1 \implies x_v^*\geq 0.5 \;\; or \;\; x_u^*\geq 0.5$$

כלומר, בפיתרון של ה-*LP* דרשנו $x_v^*+x_u^*\geq 1$ לכל uv. ואם התנאי הזה מתקיים, אז אחד הקודקודים קיבל לפחות חצי.
כלומר נבחר לכיסוי לפחות קודקוד אחד מכל צלע, כנדרש.

עכשיו שהראנו שהפתרון הוא אכן כיסוי בקודקודים, נראה עד כמה הוא מתרחק מהמשקל המינימום (האופטימלי).

טענה: האלגוריתם הוא 2-מקרב.

הוכחה: נסמן *OPT* את המשקל האופטימלי של הפתרון ל-*IP*, ו-*OPTf* את המשקל האופטימלי של הפתרון ל-*LP*.

האילוץ $0\leq x_v\leq 1$ גורר ש: $OPTf\leq OPT$. כי אותם קודקודים ייבחרו, ובגרסת ה-*LP* יתקיים $0\leq x_v\leq 1$ כמו שאמרנו.

לכל קודקוד שנמצא ב-C, היה משקל לפחות חצי ב- x^*. כלומר מתקיים:

$$\sum_{v\in C}\frac{w(v)}{2}\le\sum_{v\in v(G)}w(v)\cdot x_v^*$$

כלומר:

$$\sum_{v\in C}w(v)\le 2\cdot\underbrace{\sum_{v\in v(G)}w(v)\cdot x_v^*}_{OPTf}=2\cdot OPTf\le 2\cdot OPT$$

כנדרש.

אלגוריתם מקרי לכיסוי קודקודים מינימום (בלי משקלים)

קלט: גרף $G\coloneqq(V,E)$

1. נאתחל $C=\emptyset$.
2. נסדר את E בסדר שרירותי.
3. כל עוד $E\neq\emptyset$, נבצע:
   a. נבחר את ה- $e=uv\in E$ הבא, לפי הסדר שקבענו.
   b. נבחר $x\in\{u,v\}$ באופן מקרי ואחיד.
   c. $C\coloneqq C+x$
   d. $E\coloneqq E\setminus\{e\in E : x\in e\}$.
4. נחזיר את C.

האלגוריתם תמיד יחזיר VC תקין, כי מורידים צלע רק אם בחרנו קודקוד שנוגע בה.

טענה: התוחלת של יחס הקירוב היא 2. (כלומר נצפה שהאלגוריתם יהיה 2-מקרב, שהאלגוריתם מחזיר כיסוי בגודל לכל היותר פעמיים הגודל האופטימלי).

הוכחה: נסמן C^* כיסוי אופטימלי כלשהו. נסמן C את הכיסוי שקיבלנו מהאלגוריתם. אז $c\coloneqq|C|$ הוא משתנה מקרי. נוכיח ש: $\mathbb{E}[c]\le 2|C^*|$.

האלגוריתם מטיל סה"כ c מטבעות הוגנים. כל הטלה תקבע האם הקודקוד ייכנס ל-C.

הטלת מטבע תיקרא **טובה** אם הקודקוד שהכנסנו נמצא גם ב- C^*.

ניזכר שלכל צלע uv, לפחות אחד מתוך הקודקודים נמצא ב- C^*. כלומר לכל זריקה יש הסתברות לפחות חצי להיות טובה.

מספר ההטלות הטובות חסום ב- $|C^*|$. (כי לא נוכל לבחור יותר מ- $|C^*|$ קודקודים ששייכים ל- C^*. אחרי $|C^*|$ הטלות טובות, נקבל $C^*\subseteq C$).

נסמן X את מספר ההטלות שנדרשות כדי לקבל $|C^*|$ הטלות טובות. (זה גם מ"מ).

אזי, $c\le X$. כי ברגע ש- $c=X$, זה אומר שהטלנו X פעמים, אז קיבלנו $|C^*|$ הטלות טובות, יתקיים $C^*\subseteq C$ ונסיים.

אז גם $\mathbb{E}[c]\le\mathbb{E}[X]$. אם נראה ש- $\mathbb{E}[X]\le 2|C^*|$, זה יוכיח את הטענה.

נסמן X_1 את מספר ההטלות לפני ההטלה הטובה הראשונה. ונסמן X_2 את מספר ההטלות מההטלה הטובה הראשונה, עד ההטלה הטובה השנייה.

באופן כללי: X_i זה מספר ההטלות בין ההטלה הטובה ה- $i-1$ ועד ההטלה הטובה ה- i.

אז מתקיים: $X=X_1+X_2+\cdots+X_{|C^*|}$, כי אחרי $|C^*|$ הטלות נקבל $C^*\subseteq C$.

מלינאריות התוחלת, מתקיים:

$$\mathbb{E}[X]=\sum_{i=1}^{|C^*|}\mathbb{E}[X_i]$$

בנוסף, לכל i: $X_i\sim Geo(0.5)$. אז $\mathbb{E}[X_i]=1/0.5=2$. (תוחלת של $Geo(p)$ זה $1/p$).

כלומר, $\mathbb{E}[X]=2\cdot|C^*|$, כנדרש.

16: Approximating Max Cut

בעיית *max-cut*: בהינתן גרף G, נרצה למצוא חלוקה של הקודקודים לשתי קבוצות זרות $(S,V(G)\setminus S)$ שממקסמת את $e_G(S,V(G)\setminus S)$.

משפט: בגרף G, גודל החתך המקסימום הוא לפחות $e(G)/2$.

נגזר מכך, שלכל גרף יש תת-גרף דו"צ פורש, עם לפחות $e(G)/2$ צלעות.

כלומר אפשר למצוא חלוקה של קודקודי הגרף לשתי קבוצות ותת-קבוצה בגודל $e(G)/2$ של הצלעות, כך שכל הצלעות הן רק בין הקבוצות.

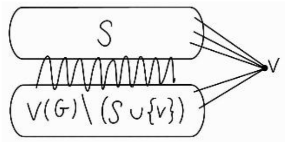
הוכחה: באינדוקציה על $v(G)$.

בסיס: עבור קודקוד 1 או 2, הטענה טריוויאלית.

צעד: נניח שהטענה מתקיימת עבור כל גרף G' עם $v(G')<n$, ונוכיח עבור גרף G עם $v(G)=n$:

נבחר קודקוד $v\in V(G)$ שרירותי, ונתבונן ב- $G-v$:

ניקח את החלוקה שקיימת ב- $G-v$ לפי הנ"א: $\big(S,V(G)\setminus(S\cup\{v\})\big)$. יש בין הקבוצות $e(G-v)/2$ צלעות, לפי הנ"א.



נוסיף את v חזרה, לקבוצה שבה יש לו פחות שכנים. כלומר, מוסיפים לחתך לפחות $(\deg_G v)/2$ צלעות.

אז גודל החתך שנוצר אחרי ההוספה הוא לפחות:

$$\frac{e(G-v)}{2}+\frac{\deg_G v}{2}=\frac{e(G-v)+\deg_G v}{2}=\frac{e(G)}{2}$$

כנדרש.

ההוכחה הזו נותנת לנו אלגוריתם פולינומי שהוא 0.5-מקרב לבעיית *max-cut*. האלגוריתם מגיע דרך תהליך ***de-randomization***.

התהליך של *de-randomization* הוא בניית אלגוריתם מקרי (שזה תהליך הרבה יותר קל) ואז המרת האלגוריתם למבנה דטרמיניסטי.

בנייה רנדומית

נבנה אלגוריתם אקראי ל-*max-cut*:

1. נאתחל $A\coloneqq\emptyset,\ B\coloneqq\emptyset$.
2. לכל קודקוד $v\in V(G)$, נטיל מטבע הוגן (כל ההטלות בת"ל):
   a. אם יצא עץ, נשים את הקודקוד ב-A. אחרת, ב-B.

ניתוח האלגוריתם: מה ההסתברות שהאלגוריתם מוציא חתך בגודל לפחות $e(G)/2$.

טענה: ההסתברות היא לפחות $1/\left(\frac{e(G)}{2}+1\right)$. זו לא הסתברות טובה, אבל האלגוריתם הזה הוא רק הבסיס.

הוכחה: נצטרך כמה טענות עזר:

עבור צלע $e\in E(G)$, נגדיר את האינדיקטור לכך ש-e חוצה את החתך:

$$X_e\coloneqq\begin{cases}1, & e\in E_G(A,B)\\ 0, & e\notin E_G(A,B)\end{cases}$$

עבור צלע $e\coloneqq uv$, מתקיים:

$$\mathbb{E}[X_e]=\mathbb{P}[u\in A\ \wedge\ v\in B]+\mathbb{P}[u\in B\ \wedge\ v\in A]=1/4+1/4=1/2$$

כי המיקומים נקבעים לפי מטבעות הוגנים בת"ל. נוכל לרשום:

$$e_G(A,B)=\sum_{e\in E(G)}X_e$$

כלומר, הגודל של החתך יהיה מספר הצלעות שהאינדיקטור שלהן יצא 1.

אזי, לפי לינאריות התוחלת נקבל:

$$\mathbb{E}[e_G(A,B)] = \sum_{e\in E(G)} \mathbb{E}[X_e] = \frac{e(G)}{2}$$

נחזור להוכחה: אנחנו רוצים לחסום מלמטה את:

$$p \coloneqq \mathbb{P}[e_G(A,B) \geq e(G)/2]$$

נמצא את הביטוי הזה בתוך התוחלת. מתקיים:

$$e(G)/2 = \mathbb{E}[e_G(A,B)] =$$

נחשב את התוחלת לפי הגדרה, ונפצל את הסכום לשניים:

$$= \sum_{i\leq\frac{e(G)}{2}-1} i\cdot\mathbb{P}[e_G(A,B)=i] + \sum_{i\geq\frac{e(G)}{2}} i\cdot\mathbb{P}[e_G(A,B)=i] \leq$$

ננתח את צד שמאל: אם נשתמש רק ב-i הכי גדול בו, נקבל ביטוי שחוסם אותו מלמעלה:

$$\leq \left(\frac{e(G)}{2}-1\right) \underbrace{\sum_{i\leq\frac{e(G)}{2}-1} \mathbb{P}[e_G(A,B)=i]}_{1-p} + \sum_{i\geq\frac{e(G)}{2}} i\cdot\mathbb{P}[e_G(A,B)=i] \leq$$

כל החלק המסומן, זה ההסתברות המשלימה למצב שאנחנו מחפשים. כלומר זה $1-p$.

נעשה דבר דומה גם בצד ימין: הערך הכי גדול ש-i יכול לקבל זה $e(G)$:

$$\leq \left(\frac{e(G)}{2}-1\right)(1-p) + e(G)\cdot \underbrace{\sum_{i\geq\frac{e(G)}{2}} \mathbb{P}[e_G(A,B)=i]}_{p} = \left(\frac{e(G)}{2}-1\right)(1-p) + e(G)p$$

בסה"כ, הראנו ש:

$$\frac{e(G)}{2} \leq \left(\frac{e(G)}{2}-1\right)(1-p) + e(G)p$$

נבודד את p ונקבל שאכן הוא חסום מלמטה כמו שרצינו.

קצת הסתברות

במרחב הסתברות כשלהו, נאמר שאירוע $\mathcal{E}$ הוא **מדיד** (*measurable*) אם $\mathbb{P}[\mathcal{E}]$ מוגדרת.

ההסתברות $\mathbb{P}[\mathcal{E}]=0$ היא מוגדרת היטב.

עבור מאורעות B,A כך ש $\mathbb{P}[B]>0$ נרשום:

$$\mathbb{P}[A|B] \coloneqq \frac{\mathbb{P}[A\cap B]}{\mathbb{P}[B]}, \qquad \mathbb{P}[B]\cdot\mathbb{P}[A|B] \coloneqq \mathbb{P}[A\cap B]$$

ועבור חלוקה בת מניה: $B = B_1 \cup B_2 \cup \ldots$, מתקיים:

$$\mathbb{P}[A] = \sum_{i:\,\mathbb{P}[B_i]>0} \mathbb{P}[A|B_i]\cdot\mathbb{P}[B_i] = \sum_i \mathbb{P}[A\cap B_i]$$

עבור מ"מ Y,X בדידים, וערך y בטווח של Y:

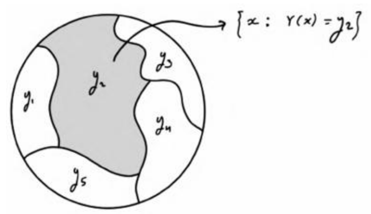
$$\mathbb{E}[X\mid Y=y] \coloneqq \sum_x x\cdot\mathbb{P}[X=x\mid Y=y]$$

$\mathbb{E}[X\mid Y]$ זה הפונקציה:

$$y \longmapsto \mathbb{E}[X\mid Y=y]$$

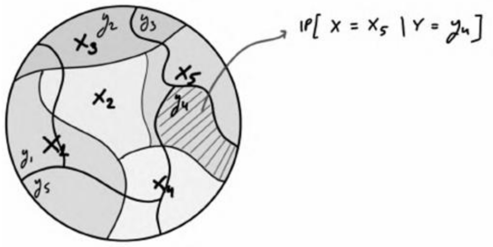
לפעמים נרשום $\mathbb{E}[X\mid Y](y)$ במקום $\mathbb{E}[X\mid Y=y]$.

Y מחלק את העולם לקבוצות *level-sets*:

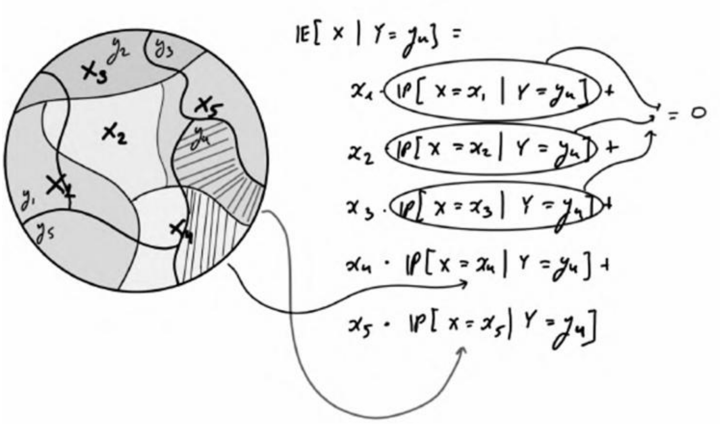


ההסתברות (*measure*) של *level-set* היא: $\mathbb{P}[Y = y_i]$.

נוסיף את X, עם ה- *level-sets* שלו, ונשים לב שהחלוקות של X ו-Y מעדנות אחת את השנייה:



אז התוחלת המותנית:



וחוק התוחלת השלימה:

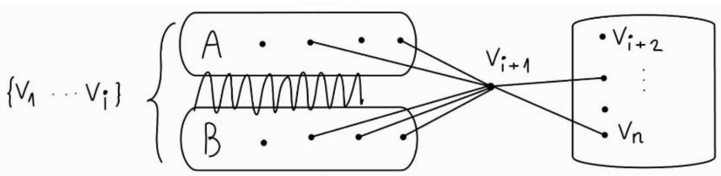
$$\mathbb{E}[X] = \sum_{y \in supp(Y)} \mathbb{E}[X \mid Y = y] \cdot \mathbb{P}[Y = y]$$

חזרה ל-Max-cut, ביצוע de-randomization

היה לנו אלגוריתם אקראי, וחסמנו מלמטה את ההסתברות שהוא מצליח. נשתמש בתוחלת המותנית כדי להוציא מהאלגוריתם את האקראיות.

יהי $\{v_1, v_2, \dots v_n\}$ את הסדר שבו עברנו על הקודקודים באלגוריתם הרנדומי.

נתבונן במצב שבו $\{v_1, v_2, \dots v_i\}$ כבר מוקמו בקבוצות A ו-B. נשקול את הרגע שבו מטילים מטבע עבור v_{i+1}:



לפי נוסחת התוחלת השלימה, התוחלת של מספר הצלעות בחתך בהינתן שכבר מיקמנו את $\{v_1, v_2, \dots v_i\}$:

$$\mathbb{E}[e_G(A,B) \mid v_1, \dots, v_i\ placed] =$$

$$= \underbrace{\mathbb{P}[v_{i+1} \in A]}_{0.5} \cdot \mathbb{E}[e_G(A,B) \mid v_1, \dots, v_i\ placed\ \wedge\ v_{i+1} \in A] + \underbrace{\mathbb{P}[v_{i+1} \in B]}_{0.5} \cdot \mathbb{E}[e_G(A,B) \mid v_1, \dots, v_i\ placed\ \wedge\ v_{i+1} \in B] =$$

$$= 0.5 \cdot \left(\underbrace{\mathbb{E}[e_G(A,B) \mid v_1, \dots, v_i\ placed\ \wedge\ v_{i+1} \in A]}_{E_A} + \underbrace{\mathbb{E}[e_G(A,B) \mid v_1, \dots, v_i\ placed\ \wedge\ v_{i+1} \in B]}_{E_B} \right)$$

נסמן את התוחלת $\mathbb{E}$. יש לנו את המבנה: $2\mathbb{E} = E_A + E_B \implies \mathbb{E} = (E_A + E_B)/2$.

כלומר, $E_A \geq \mathbb{E}$ או $E_B \geq \mathbb{E}$. (כי אם שניהם קטנים יותר, הסכום שלהם לא יעלה על $2\mathbb{E}$).

כלומר: לפחות אחת האופציות **לא מקטינה** את התוחלת המותנית. לפי העיקרון הזה, נבנה אלגוריתם דטרמיניסטי:

1. יהי $v_1, v_2, \dots v_n$ סדר כלשהו של הקודקודים.
2. נאתחל $A = B = \emptyset$
3. עבור $i = 1$ עד n:
   a. נחשב את E_A, E_B. (אח"כ נסביר איך).
   b. אם $E_A > E_B$, נשים את v_i ב-A. אחרת, נשים אותו ב-B.

נוכיח שהאלגוריתם לעיל תמיד מייצר חתך בגודל לפחות $e(G)/2$:

כבר הראנו ש- $e(G)/2 \leq \mathbb{E}[e_G(A, B)]$.

והראינו שכל צעד לא מקטין את התוחלת המותנית. כלומר:

$$\mathbb{E}[e_G(A,B)] \leq \mathbb{E}[e_G(A,B) \mid v_1 \textit{ is placed}] \leq \mathbb{E}[e_G(A,B) \mid v_1, v_2 \textit{ are placed}] \leq \cdots \leq \mathbb{E}[e_G(A,B) \mid v_1, v_2 \dots, v_n \textit{ are placed}]$$

והשלב האחרון זה אחרי שמיקמנו את כל הנקודות. בסה"כ, קיבלנו שהתוחלת של $e_G(A,B)$ היא לפחות $e(G)/2$, גם בסוף האלגוריתם.

חישוב E_A, E_B

נגדיר את $\widetilde{A}, \widetilde{B}$ ונפריד ביניהם לבין B,A:

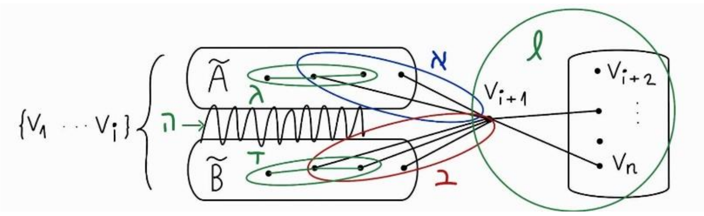
בשלב אחרי שמיקמנו את $v_1, \dots, v_i$, נאמר ש- $\widetilde{A}, \widetilde{B}$ זה מה שאנחנו רואים ב-B,A באותו הרגע. A ו-B הם בשביל המשתנה המקרי $e_G(A,B)$.

נשקול מה קורה עבור v_{i+1}: נרצה לחשב את E_A, E_B כדי שנדע לאיזה צד להכניס אותו. על מה בעצם לוקחים תוחלת מותנית?

$\widetilde{A}, \widetilde{B}$ כבר מקובעים, ואנחנו יודעים איזה צלעות יש ביניהם, ואיזה צלעות יש בין v_{i+1} לכל אחד מ- $\widetilde{A}, \widetilde{B}$.

מה שעדיין לא קבוע, הם הצלעות שמחוברות ל- $v_{i+2}, \dots, v_n$. נסמן את הכמות של הצלעות האלה ℓ:

$$\ell \coloneqq e(G) - \underbrace{\deg_G(v_{i+1}, \widetilde{A})}_{\text{א}} - \underbrace{\deg_G(v_{i+1}, \widetilde{B})}_{\text{ב}} - \underbrace{e_G(\widetilde{A})}_{\text{ג}} - \underbrace{e_G(\widetilde{B})}_{\text{ד}} - \underbrace{e_G(\widetilde{A}, \widetilde{B})}_{\text{ה}}$$



E_A זה התוחלת של גודל החתך, עבור המצב הנוכחי ותחת ההנחה ששמנו את v_{i+1} ב-A.

שזה שווה למספר הצלעות הנוכחי בחתך, ועוד מספר הצלעות שיש בין v_{i+1} ל- $\widetilde{B}$,
ועוד התוחלת של מספר הצלעות הלא-קבועות שיחצו את החתך – שזה חצי מהן.

$$E_A \coloneqq \mathbb{E}[e_G(A,B) \mid v_1, \dots, v_i \textit{ are placed } \wedge \ v_{i+1} \in A] = e_G(\widetilde{A}, \widetilde{B}) + \deg_G(v_{i+1}, \widetilde{B}) + \ell/2$$

$$E_B \coloneqq \mathbb{E}[e_G(A,B) \mid v_1, \dots, v_i \textit{ are placed } \wedge \ v_{i+1} \in B] = e_G(\widetilde{A}, \widetilde{B}) + \deg_G(v_{i+1}, \widetilde{A}) + \ell/2$$

אז ההבדל בין E_A ל- E_B הוא רק ההבדל בין הדרגות:

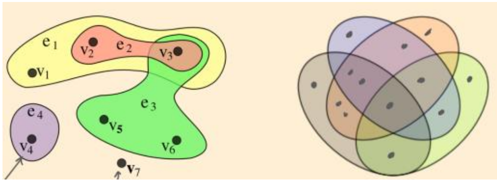
$$|E_A - E_B| = \left|\deg_G(v_{i+1}, \widetilde{B}) - \deg_G(v_{i+1}, \widetilde{A})\right|$$

אז אפשר לבדוק רק איזה מהדרגות גדולה יותר. וזה בדיוק האלגוריתם הראשון שראינו.

היפרגרף הוא זוג סדור: $H \coloneqq (V,\mathcal{E})$ כאשר $\mathcal{E} \subseteq \mathcal{P}(V)$. כלומר במקום צלעות רגילות, יש קבוצות של קודקודים.

אם כל האיברים של $\mathcal{E}$ הם זוגות, נאמר ש-H הוא גרף (רגיל). האיברים של H נקראים **היפר-צלעות** (*hyperedges*).

אם $|e| = k$ לכל $e \in \mathcal{E}$, אז H נקרא k-אחיד (*k-uniform*). אז גרף רגיל הוא 2-אחיד.



בהיפרגרף השמאלי: e_4 הוא לולאה. v_7 הוא קודקוד מבודד, יש לו דרגה 0.

בעיית *minimum set-cover*

כשנדבר על היפר-גרף, במקום לכתוב "היפר-צלע" נכתוב "צלע".

בהינתן היפרגרף $H \coloneqq (V,\mathcal{E})$, נרצה למצוא תת-קבוצת צלעות $C \subseteq \mathcal{E}$ כך ש: $\bigcup_{e\in C} e = V$, תחת האילוץ למזער את $|C|$.

אנחנו בעצם מחפשים כיסוי בצלעות מינימום של H. נסמן את הגודל של הכיסוי המינימלי $\varrho(H)$.

בעיית *minimum edge-cover* היא ב-P. בעיית *min SC* היא *NPC*. נוכיח ע"י רדוקציה מבעיית *min VC*. נרצה להראות ש $VC \le_p SC$:

$$VC \coloneqq \{(G,k) : \tau(G) \le k\}, \qquad SC \coloneqq \{(H,k) : H \text{ has a hyperedge cover} \le k\}$$

נבנה פונקציה פולינומית שמקבלת (G,k_1), ומחזירה (H,k_2) כך ש: $(H,k_2) \in SC \iff (G,k_1) \in VC$.

עבור קודקוד $u \in V(G)$, נגדיר:

$$S_u \coloneqq \{e \in E(G) : v \in e\}$$

את קבוצת כל הצלעות שנוגעות ב-u.

בהינתן (G,k), נגדיר היפרגרף H באופן הבא:

$$V(H) \coloneqq E(G), \qquad \mathcal{E}(H) \coloneqq \{S_u : u \in V(G)\}$$

כלומר, כל צלע של G היא קודקוד ב-H, והקבוצות (היפר-צלעות) הן צלעות של G שחולקות קודקוד. ואז:

$$f\big((G,k)\big) = (H,k)$$

נוכיח את נכונות הרדוקציה

הפונקציה פולינומית, כי מספר הקודקודים ב-H הוא מספר הצלעות ב-G. ומספר ההיפר-צלעות חסום במספר תתי-הקבוצות של הקודקודים.

נוכיח ש $(H,k) \in SC \iff (G,k) \in VC$:

יהי קודקוד $v \in V(G)$. הקבוצה S_v מכילה את כל הצלעות ב-G ש-v נוגע בהן.

ב-H, ההי"צ S_v מכסה את כל הקודקודים ב-H שמייצגים צלעות ש-v מכסה ב-G.

אז אם יש k קודקודים ב-G שמכסים את כל הצלעות, אז ה-S של אותם קודקודים יכסו ב-H את כל הקודקודים.

וגם הפוך, אם יש k היפר-צלעות ב-H שמכסות את כל הקודקודים של H, אז הקודקודים המתאימים ב-G מכסות את כל הצלעות.

אלגוריתם חמדן

עבור מערכת קבוצות A, נסמן: $\bigcup A \coloneqq \bigcup_{X\in A} X$.

האלגוריתם:

1. נאתחל $A \coloneqq \emptyset$.
2. כל עוד $\bigcup A \neq V$:
   a. $e \coloneqq argmax\{|(\bigcup A) \cup e| : e \in \mathcal{E}\}$. נבחר את הצלע שאם ניקח אותה, זה ממקסם את מספר האיברים ב-A.
   b. $A \coloneqq A \cup \{e\}$.

יש דוגמאות שבהן האלגוריתם החמדן לא מוצא את הפתרון האופטימלי.

נשאל, מהי איכות הקירוב? בשביל זה, נגדיר את הגודל של הצלע הכי גדולה:

$$m \coloneqq \max\{|e| : e \in \mathcal{E}\}$$

נסמן $n \coloneqq |V(H)|$. באופן טריוויאלי, $m \le n$ (הצלע הכי גדולה לא מכסה יותר קודקודים ממה שקיים).

נוכיח שהאלגוריתם הוא $O(\ln n)$-מקרב, ואז נשפר ונראה שהוא $O(\ln m)$-מקרב.

הוכחת קירוב $O(\ln n)$

אבחנה: האלגוריתם הוא איטרטיבי. וגודל הכיסוי שהוא מייצר שווה למספר האיטרציות, כי בכל איטרציה נוסיף צלע אחת לכיסוי.

נגדיר: A_i את קבוצת הצלעות בכיסוי של הפתרון החמדן בהתחלה של האיטרציה ה-i. במהלך האיטרציה ה-i, האלגוריתם עובר מפתרון A_i לפתרון A_{i+1}.

מספר הקודקודים החדשים שכוסו במהלך האיטרציה ה-i: $|\cup A_{i+1}| - |\cup A_i|$. ונשים לב, שכל הקודקודים החדשים הם מ- $V(H) \setminus \cup A_i$.

נרצה לתת חסם תחתון למספר הקודקודים החדשים שיכוסו.

נגדיר: OPT את גודל הכיסוי המינימלי של H, ויהי פיתרון אופטימלי כלשהו $p \coloneqq \{e_1, e_2, \dots, e_{OPT}\}$.

בכל איטרציה i, הפיתרון p מכיל צלע e_j שמקיימת:

$$\left|e_j \cap (V(H) \setminus \cup A_i)\right| \ge \frac{|V(H) \setminus \cup A_i|}{OPT}$$

כלומר, יש צלע כלשהי שהחיתוך שלה עם כל הקודקודים שעוד לא כוסו, הוא לפחות מספר הקודקודים הזה, חלקי גודל הפיתרון האופטימלי.

למה? מעיקרון שובך היונים. אם אין אפילו צלע אחת כזו, זה אומר שגם אם ניקח את כל צלעות הפתרון האופטימלי, הן לא יכסו את כל הקודקודים.

הצלע הזו היא לא חלק מהפתרון החמדן הנוכחי (כי יש לה חיתוך עם המשלים של הפתרון הנוכחי).

אז במעבר מ-A_i ל-A_{i+1}, ניקח את הצלע שמכסה הכי הרבה קודקודים חדשים, אז ניקח צלע שמכסה לפחות $\frac{|V(H) \setminus \cup A_i|}{OPT}$ קודקודים. אז השיפור יהיה לפחות:

$$|\cup A_{i+1}| - |\cup A_i| \ge \frac{|V(H) \setminus \cup A_i|}{OPT}$$

נעביר אגף ונקבל ש:

$$|\cup A_{i+1}| \ge \frac{|V(H) \setminus \cup A_i|}{OPT} + |\cup A_i|$$

ברגע שהגדרנו את A_{i+1}, כמה קודקודים עוד צריך לכסות?

$$|V(H) \setminus \cup A_{i+1}| \overset{\text{א}}{=} |V(H)| - |\cup A_{i+1}| \overset{\text{ב}}{\le} |V(H)| - \left(\frac{|V(H) \setminus \cup A_i|}{OPT} + |\cup A_i|\right) \overset{\text{א}}{=} |V(H)| - \left(\frac{|V(H)|}{OPT} - \frac{|\cup A_i|}{OPT} + |\cup A_i|\right) \overset{\text{ג}}{=}$$
$$= |V(H)| - \frac{|V(H)|}{OPT} - |\cup A_i| + \frac{|\cup A_i|}{OPT} \overset{\text{ד}}{=} \left(1 - \frac{1}{OPT}\right)(|V(H)| - |\cup A_i|) \overset{\text{א}}{=} \left(1 - \frac{1}{OPT}\right)|V(H) \setminus \cup A_i|$$

א. גודל של הפרש קבוצות.
ב. נציב את מה שקיבלנו לעיל.
ג. פתיחת סוגריים והחלפת סדר.
ד. גורם משותף.

כלומר כמות הקודקודים הלא-מכוסים אחרי האיטרציה ה- i, היא לכל היותר הכמות שהייתה בתחילת האיטרציה, כפול $\left(1 - \frac{1}{OPT}\right)$.

אז יש תבנית רקורסיבית: נגדיר U_i את הקודקודים הלא-מכוסים באיטרציה ה-i, אז: $U_0 = V(H)$, $U_i = V(H) \setminus \cup A_i$.

נגדיר $u_i = |U_i|$, ומתקיים:

$$u_i \le \left(1 - \frac{1}{OPT}\right)u_{i-1} \le \left(1 - \frac{1}{OPT}\right)\left(1 - \frac{1}{OPT}\right)u_{i-2} \le \left(1 - \frac{1}{OPT}\right)^3 u_{i-3} \le \cdots \le \left(1 - \frac{1}{OPT}\right)^i u_0$$

נוכיח פורמלית באינדוקציה:

<u>בסיס</u>: באופן טריוויאלי מתקיים:

$$u_0 \le \left(1 - \frac{1}{OPT}\right)^0 u_0$$

צעד: נניח שהטענה נכונה עבור כל u_i עם $i < n$, ונוכיח עבור u_n:

$$u_n \leq \left(1 - \frac{1}{OPT}\right) u_{n-1} = \left(1 - \frac{1}{OPT}\right)\left(1 - \frac{1}{OPT}\right)^{n-1} u_0 = \left(1 - \frac{1}{OPT}\right)^{n} u_0$$

ניזכר באי-שוויון: $1 - x \leq \exp(-x)$ לכל $x \in (0,1]$. כלומר:

$$u_i \leq \left(1 - \frac{1}{OPT}\right)^{i} u_0 \leq \exp\left(\frac{-i}{OPT}\right) \cdot n$$

הסדרה $u_0, u_1, \dots$ היא סדרה יורדת, בשלמים. נשאל, מה ה-i הכי קטן שאפשר לשים כדי שיתקיים: (כלומר שלא נשארו קודקודים לכסות)

$$\exp\left(\frac{-i}{OPT}\right) \cdot n < 1 \;\Rightarrow\; n < \exp\left(\frac{i}{OPT}\right) \;\Rightarrow\; \ln n < \frac{i}{OPT} \Rightarrow OPT \cdot \ln n < i$$

אז כדי שלא יישארו קודקודים, צריך i כלשהו שהוא גדול ממש מ- $OPT \cdot \ln n$. אז $OPT \cdot \ln n + 1$ יעבוד.

כלומר, אחרי $OPT \cdot \ln n + 1$ איטרציות, נכסה את כל $V(H)$. ומספר האיטרציות זה הגודל של הפיתרון החמדן.

כלומר האלגוריתם החמדן הוא $O(\ln n)$-מקרב.

הוכחת קירוב $O(\ln m)$

אנחנו יודעים ש- $OPT \leq u_0$ (ברור שמספר הצלעות בכיסוי האופטימלי קטן ממספר הקודקודים שצריך לכסות).

אז בגלל שה-u הם סדרה יורדת, מתקיים:

$$\exists i^* \; s.t \;\; u_{i^*} \geq OPT > u_{i^*+1}$$

כי היא מתחילה מעל OPT ומסתיימת ב-0.

כלומר, אחרי האיטרציה ה- $i^* + 1$, יש לכל היותר $OPT - 1$ קודקודים שצריך לכסות.

מאותו רגע, נצטרך לבצע לכל היותר עוד $OPT - 1$ איטרציות.

גם אם בכל איטרציה נכסה רק קודקוד אחד, זה יספיק.

אז נוכל לרשום:

$$size\text{-}of\text{-}greedy\text{-}solution \leq i^* + 1 + OPT - 1 = i^* + OPT$$

נוכל להעריך את $i*$. אנחנו יודעים ש:

$$OPT \leq u_{i^*} \leq \exp\left(\frac{-i^*}{OPT}\right) \cdot n \;\;\Rightarrow\;\; \frac{OPT}{n} \leq \exp\left(\frac{-i^*}{OPT}\right) \;\;\Rightarrow\;\; \exp\left(\frac{i^*}{OPT}\right) \leq \frac{n}{OPT} \;\;\Rightarrow\;\; i^* \leq \ln\left(\frac{n}{OPT}\right) \cdot OPT$$

אז:

$$size\text{-}of\text{-}greedy\text{-}solution \leq \ln\left(\frac{n}{OPT}\right) \cdot OPT + OPT = \left(1 + \ln\left(\frac{n}{OPT}\right)\right) \cdot OPT$$

ונשים לב ש:

$$m \coloneqq \max\{|e| : e \in \mathcal{E}\} \geq \frac{n}{OPT}$$

כי כל פתרון אופטימלי מכיל צלע שמכסה לפחות n/OPT קודקודים (שובך היונים). אז בוודאי שיש צלע בגודל כזה. אזי מתקיים:

$$\ln\left(\frac{n}{OPT}\right) \leq \ln(m)$$

ובסה"כ נקבל:

$$size\text{-}of\text{-}greedy\text{-}solution \leq (1 + \ln(m)) \cdot OPT$$

Min Cost Set Cover

בהינתן היפר-גרף H, ופונקציית משקל $c: E(H) \to \mathbb{R}^+$. נרצה למצוא תת-קבוצה של צלעות $C \subseteq E(H)$ שמכסה את כל הקודקודים, שממזערת את:

$$\sum_{e \in C} c(e)$$

ראינו אלגוריתם חמדן עבור הגרסה הלא-ממושקלת. הוא לא מתאים לגרסה הממושקלת. ננסה להתאים את האלגוריתם לגרסה הממושקלת:

נשקול את האיטרציה ה-i של האלגוריתם. נניח ש- $V(H) \setminus \cup A_i \neq \emptyset$, כלומר יש עוד קודקודים שלא מכוסים.

עבור צלע $e \in V(H) \setminus A_i$, נתבונן ביחס:

$$\frac{|e \setminus \cup A_i|}{|V(H) \setminus \cup A_i|}$$

כלומר: קודקודים ב-e שעדיין לא מכוסים, חלקי כל הקודקודים שעדיין לא מכוסים. ככל שהיחס גבוה יותר, הצלע משמעותית יותר.

אז דרך חדשה לתאר את הבחירה החמדנית היא הבחירה של הצלע הכי משמעותית.

בשביל הגרסה הממושקלת, נרצה להתחשב במשקלים:

$$\frac{c(e \setminus \cup A_i)}{|V(H) \setminus \cup A_i|}$$

הבעיה היא, שזה לא מוגדר היטב. כי $e \setminus \cup A_i$ לא בהכרח תהיה צלע.

ננסה משהו אחר. נשקול את היחס:

$$\frac{1}{|e \setminus \cup A_i|}$$

ככל שהוא נמוך יותר, זה מכסה יותר קודקודים חדשים. אם נבחר את הצלע שממזערת את זה, זה שקול לאלגוריתם החמדן המקורי.

בגרסה הממושקלת, היחס יהיה:

$$\frac{c(e)}{|e \setminus \cup A_i|}$$

המשקל של e מתחלק על כל הקודקודים שהיא תוסיף לכיסוי.

עבור קבוצת קודקודים $U \subseteq V(H)$, וצלע $e \in E(H)$, נגדיר את **היעילות** של e:

$$\text{eff}_U(e) := \begin{cases} \infty, & e \subseteq U \\ \dfrac{c(e)}{|e \setminus \cup A_i|}, & else \end{cases}$$

אנחנו נרצה למזער את זה. אז אם הצלע כולה מוכלת בקבוצה U, יש לה "יעילות אינסופית" ואז לא ניקח אותה.

אז האלגוריתם החמדן למשקלים:

3. נאתחל $A := \emptyset$.
4. כל עוד $\cup A \neq V$:
   a. $e := argmin\{\text{eff}_{\cup A}(e) : e \in E(H)\}$. הצלע עם מינימום יעילות ביחס לקודקודים שכבר מכוסים.
   b. $A := A \cup \{e\}$.

נרצה לנתח את יחס הקירוב של האלגוריתם. בגרסה הקודמת, השתמשנו בזה שבכל איטרציה מוסיפים צלע אחת.

עכשיו אין לנו את האפשרות הזו, כי אנחנו לא יודעים מה המשקל שמוסיפים כל פעם. נצטרך להשתמש בניתוח לשיעורין (*amortized analysis*).

אינטואיציה: יהי p פתרון אופטימלי, שעלותו OPT. קיימת צלע $e \in p$ שמקיימת:

$$\frac{c(e)}{|e|} \leq \frac{OPT}{n}$$

למה? נב"ש שלכל צלע $e \in p$ מתקיים:

$$\frac{c(e)}{|e|} > \frac{OPT}{n}$$

אזי,

$$OPT = \sum_{e\in p} c(e) =^{א} \sum_{e\in p}\sum_{v\in e}\frac{c(e)}{|e|} >^{ב} \sum_{e\in p}\sum_{v\in e}\frac{OPT}{n} = \frac{OPT}{n}\underbrace{\sum_{e\in p}\sum_{v\in e} 1}_{\geq n} \geq^{ג} \frac{OPT}{n}\cdot n = OPT$$

א. ניתן לכל קודקוד בצלע את המשקל הממוצע בצלע. ביחד זה יוצא שהמשקל של הצלע זהה.
ב. לפי ההנחה בשלילה.
ג. כי סכום הקודקודים שיש בצלעות של כל פתרון הוא לפחות n (כי הפתרון מכסה את כל הקודקודים).

קיבלנו ש-OPT גדול יותר מעצמו, שזו כמובן סתירה.

אז תמיד קיימת צלע בפתרון האופטימלי, שמכסה את הקודקודים שלה בעלות לכל היותר OPT/n.

ניתוח לשיעורין באלגוריתם החמדן

נתבונן בבחירה הראשונה. האלגוריתם בוחר צלע e שממזערת את:

$$\frac{c(e)}{|e\setminus\emptyset|} = \frac{c(e)}{|e|}$$

אנחנו הוכחנו שיש צלע שמקיימת $\frac{c(e)}{|e|} \leq \frac{OPT}{n}$. והצלע הראשונה שבחרנו ממזערת את היחס הזה, כלומר הצלע e הראשונה מקיימת את האי-שוויון הזה.

נרצה להכליל את הטענה לכל שלב.

נסמן p פתרון אופטימלי בעלות OPT. ונסמן A את הצלעות המכוסות ברגע כלשהו במהלך האלגוריתם.

נגדיר את קבוצת כל הצלעות מ-p, שיש בהן קודקודים שעוד לא מכוסים. כלומר עוד לא נבחרו בחמדן:

$$T \coloneqq \{e\in p : e\setminus \cup A \neq \emptyset\} \subseteq p$$

נשים לב שהקודקודים שלא מכוסים עדיין, מוכלים בקודקודים ב-T:

$$(V(H)\setminus \cup A) \subseteq \cup T$$

נוכיח: נב"ש שקיים קודקוד $v\in V(H)\setminus \cup A$ כך ש: $v\notin \cup T$, ותהי $e\in p$ כך ש- $v\in e$.

אזי, $e\setminus \cup A\neq\emptyset$, אז $e\in T$. ואז $v\in\cup T$, סתירה.

טענה על T: קיימת צלע $e\in T$ כך ש:

$$\frac{c(e)}{|e\setminus \cup A|} \leq \frac{OPT}{|V(H)\setminus \cup A|}$$

ונשים לב שהטענה שאמרנו לעיל: $\frac{c(e)}{|e|} \leq \frac{OPT}{n}$, היא מקרה פרטי של הטענה הזו עבור $\cup A = \emptyset$, האיטרציה הראשונה.

הוכחה: נב"ש שלכל $e\in T$,

$$\frac{c(e)}{|e\setminus \cup A|} > \frac{OPT}{|V(H)\setminus \cup A|}$$

אז באופן דומה למה שעשינו למעלה:

$$OPT =^{א} \sum_{e\in p} c(e) \geq^{ב} \sum_{e\in T} c(e) =^{ג} \sum_{e\in T}\sum_{v\in e\setminus\cup A}\frac{c(e)}{|e\setminus\cup A|} >^{ד} \sum_{e\in T}\sum_{v\in e\setminus\cup A}\frac{OPT}{|V(H)\setminus\cup A|} = \frac{OPT}{|V(H)\setminus\cup A|}\sum_{e\in T}\sum_{v\in e\setminus\cup A} 1 \geq OPT$$

א. הגדרת OPT, סכום העלות של הפתרון האופטימלי.
ב. כי $T\subseteq p$.
ג. פיצול כמו שעשינו מקודם, לפי הגדרת T.
ד. לפי ההנחה בשלילה.

סתירה.

איכות הקירוב של האלגוריתם החמדן

נוסיף חלק באלגוריתם בשביל בניתוח:

1. נאתחל $A \coloneqq \emptyset$.
2. כל עוד $\cup A \neq V$:
   a. $e \coloneqq argmin\{\text{eff}_{\cup A}(e) : e \in E(H)\}$.
   b. עבור כל קודקוד $v \in (e \setminus \cup A)$, (הקודקודים של e, שעוד לא מכוסים):
      i. נגדיר $ac(v) \coloneqq \text{eff}_{\cup A}(e)$. (*ac = amortized cost*)
   c. $A \coloneqq A \cup \{e\}$.

למה: המחיר של הפתרון החמדן שווה ל: $\sum_{v \in V} ac(v)$.

הוכחה: יהי A פתרון חמדן, וניתן איזשהו סדר לבחירות שלו:

$$A \coloneqq \{e_{i1}, e_{i2}, \dots, e_{im}\}$$

נסמן A_j את הפתרון החמדן החלקי - כלומר נעצור אותו בשלב כלשהו:

$$A_j \coloneqq \{e_{i1}, e_{i2}, \dots, e_{ij}\}$$

באופן דומה למה שכבר עשינו פעמיים, נשים לב שהעלות של החמדן מקיימת:

$$\sum_{e \in A} c(e) = \sum_{j=1}^{m} c(e_{ij}) \overset{\text{א}}{=} \sum_{j=1}^{m} \sum_{v \in (e_{ij} \setminus \cup A_{j-1})} \frac{c(e_{ij})}{|e_{ij} \setminus \cup A_{j-1}|} \overset{\text{ב}}{=} \sum_{j=1}^{m} \sum_{v \in (e_{ij} \setminus \cup A_{j-1})} ac(v) \overset{\text{ג}}{=} \sum_{v \in V} ac(v)$$

א. נחלק את המחיר בין הקודקודים. אפשר לחשוב על זה בתור חלוקה והכפלה ב- $|e_{ij} \setminus \cup A_{j-1}|$.
ב. לפי הגדרת ac.
ג. מעבר על כל הקודקודים החדשים בכל שלב, זה פשוט כל הקודקודים.

הוכחת קירוב $O(\ln n)$

נסמן A, A_j כמו לעיל.

כל איטרציה מכסה לפחות קודקוד אחד חדש.

באיטרציה ה-j, הצלע e_{ij} נבחרת, ומכסה קודקודים חדשים. נסמן אותם: $k(j)$ זה מספר הקודקודים החדשים שכוסו באיטרציה ה-j.

$$v_1^j, v_2^j, \dots, v_{k(j)}^j$$

נסדר את כל הקודקודים ב-V לפי הסדר שבו הם כוסו ע"י האלגוריתם: m הוא גודל הפתרון החמדן.

$$V \coloneqq \{v_1, v_2, \dots, v_n\} = \{v_1^1, v_2^1, \dots, v_{k(1)}^1, v_1^2, \dots, v_{k(2)}^2, \dots, v_1^m, v_2^m, \dots, v_{k(m)}^m\}$$

לכל $j \in [m]$, מתקיים:

$$ac(v_1^j) = ac(v_2^j) = \dots = ac\left(v_{k(j)}^j\right) = \text{eff}_{\cup A_{j-1}}(e_{ij}) = \frac{c(e_{ij})}{|e_{ij} \setminus \cup A_{j-1}|}$$

נקבע איטרציה j כלשהי ויהי $v_k \in \{v_1, \dots, v_n\}$ הקודקוד הראשון שכוסה באיטרציה הזו. כלומר, $|\cup A_{j-1}| = k - 1$.

כל פתרון אופטימלי מכיל צלע e שמוסיפה קודקודים חדשים:

$$e \setminus \cup A_{j-1} \neq \emptyset$$

וגם, לפי ה"טענה על T" שהוכחנו לעיל,

$$\text{eff}_{\cup A_{j-1}}(e) \leq \frac{OPT}{|V(H) \setminus \cup A_{j-1}|} = \frac{OPT}{n-(k-1)} = \frac{OPT}{n-k+1}$$

באיטרציה ה-j, בחרנו את e_{ij}. מכיוון ש $e \setminus \cup A_{j-1} \neq \emptyset$, הצלע e גם הייתה מועמדת. כלומר בחרנו את e_{ij} על פני e. כלומר:

$$\mathrm{eff}_{\cup A_{j-1}}\left(e_{ij}\right) \le \mathrm{eff}_{\cup A_{j-1}}(e) \le \frac{OPT}{n-k+1}$$

אזי,

$$ac(v_k) = \mathrm{eff}_{\cup A_{j-1}}\left(e_{ij}\right) \le \frac{OPT}{n-k+1}$$

כל קודקוד $v_{k+\ell}$ (עבור $\ell \ge 0$) שמכוסה ביחד עם v_k מקיים:

$$ac(v_{k+\ell}) \le \frac{OPT}{n-k+1} \le \frac{OPT}{n-(k+\ell)+1}$$

אזי לכל $r \in [n]$, מתקיים:

$$ac(v_r) \le \frac{OPT}{n-r+1}$$

כלומר, המחיר של החמדן חסום ב:

$$COST \overset{\text{א}}{=} \sum_{i=1}^{n} ac(v_i) \overset{\text{ב}}{\le} \sum_{i=1}^{n} \frac{OPT}{n-i+1} \overset{\text{ג}}{=} OPT \sum_{i=1}^{n} \frac{1}{n-i+1} \overset{\text{ד}}{=} OPT \sum_{i=1}^{n} \frac{1}{i} \overset{\text{ה}}{=} O(\ln n)$$

א. הוכחנו, זה בדיוק הרעיון של ac.
ב. האי-שוויון שבדיוק הוכחנו.
ג. נוציא גורם משותף.
ד. נשים לב שהסכימה הפכה להיות בעצם מ-n עד 1.
ה. טור הרמוני.

כנדרש.

עבור היפר-גרף $H \coloneqq (V, \mathcal{E})$, פונקציית משקל $c: \mathcal{E} \to \mathbb{R}^+$. נמצא תת-קבוצה של צלעות $C \subseteq \mathcal{E}$ שמכסה את כל הקודקודים, שממזערת את $\sum_{e \in C} c(e)$.

הגדרת בעיית ה-*IP*

יהיה לנו וקטור x שמייצג את הפתרון. כל אינדקס בווקטור מייצג צלע:

$$x_e \coloneqq \begin{cases} 1, & e \in C \\ 0, & else \end{cases}$$

ואנחנו נרצה למזער את $\sum_{e \in \mathcal{E}} c(e) x_e$, תחת האילוצים:

נגדיר לכל $v \in V$ את קבוצת הצלעות שמכסות אותו: $C_v \coloneqq \{e \in \mathcal{E} : v \in e\}$.

נדרוש שיתקיים $\sum_{e \in C_v} x_e \geq 1$ לכל $v \in V$. כלומר, לכל קודקוד, לפחות צלע אחת שמכסה אותה תהיה חלק מהכיסוי.

בעיית ה-*IP* היא *NPC*. נבצע *relaxation* ל-*LP*:

הגדרת בעיית ה-*LP*

הכל אותו דבר, חוץ מזה ש- $0 \leq x_e \leq 1$. כרגיל, אפשר פשוט לדרוש $x_e \geq 0$, כי אם יש $x_e > 1$, היינו מקבלים פתרון טוב יותר עם $x_e = 1$.

אז דרישת המינימום תיתן לנו את ההגבלה $x_e \leq 1$.

עיגול דטרמיניסטי של ה-*LP* הראשונית (*Deterministic Rounding of the Primal LP*)

עבור $v \in V$, נסמן את התדירות (*frequency*) שלו – גודל **הכוכב** שמכיל אותו. כלומר מספר הצלעות שמכסות אותו: $f_v \coloneqq |C_v|$.

ונגדיר $f \coloneqq \max_{v \in V} f_v$ את התדירות המקסימלית.

אז האלגוריתם לבעיית *SC*:

1. נמצא פתרון אופטימלי לבעיית ה-*LP*. נסמן את הפיתרון $(x_e^*)_{e \in \mathcal{E}}$.
2. נחזיר את: $A \coloneqq \{e \in \mathcal{E} : x_e^* \geq 1/f\}$

טענה: האלגוריתם הוא *f*-מקרב.

הוכחה: ראשית, נראה ש-*A* היא אכן כיסוי.

יהי $v \in V$ כלשהו. לפי הפיתרון שקיבלנו מה-*LP*, מתקיים: $\sum_{e \in C_v} x_e^* \geq 1$.

כלומר קיימת $e \in C_v$ כך ש:

$$x_e^* \overset{\text{א}}{\geq} \frac{1}{|C_v|} \overset{\text{ב}}{=} \frac{1}{f_v} \overset{\text{ג}}{\geq} \frac{1}{f}$$

א. שובך היונים.
ב. הגדרת f_v.
ג. כי $f \geq f_v$ לכל v.

אזי לפי הגדרת *A*, הצלע *e* שמכסה את *v* תהיה בכיסוי.

נוכיח את איכות הקירוב

נגדיר ווקטור *z* שמזכיר את הווקטור האופייני של *A*. לכל $e \in \mathcal{E}$, נגדיר:

$$z_e \coloneqq \begin{cases} 1, & e \in A \\ x_e^*, & e \notin A \end{cases}$$

נשים לב שלכל $e \in \mathcal{E}$, מתקיים:

$$z_e \leq f \cdot x_e^*$$

עבור $e \in A$, מתקיים $x_e^* \geq 1/f$ (כי זה מה שגרם ל-*e* להיכלל ב-*A*). אז מתקיים ש:

$$f \cdot x_e^* \geq f \cdot (1/f) = 1 = z_e$$

עבור $e \notin A$, ניזכר שמתקיים $f \geq 1$ (באופן כללי, כי ברור שלכל קודקוד יש לפחות צלע אחת שמכסה אותו כי אחרת אין כיסוי). ונקבל:

$$z_e = x_e^* \leq f \cdot x_e^*$$

נוכל עכשיו להעריך את עלות הפתרון שחזר מהאלגוריתם. נסמן אותו $c(A)$. מתקיים:

$$c(A) \leq^{\text{א}} \sum_{e \in \mathcal{E}} z_e \cdot c(e) \leq^{\text{ב}} f \cdot \sum_{e \in \mathcal{E}} x_e^* \cdot c(e) =^{\text{ג}} f \cdot \mathrm{OPT}_f \leq^{\text{ד}} f \cdot OPT$$

א. כי אם $e \in A$, אז $z_e = 1$, וזה מוסיף לסכום בדיוק את המשקל של הצלע.
אם $e \notin A$, אז היא מוסיפה 0 לסכום, וזה בוודאי פחות מ- $x_e^* \cdot c(e)$.
ב. לפי הטענה לעיל, $z_e \leq f \cdot x_e^*$.
ג. הסכום זה בדיוק העלות של הפתרון האופטימלי השברי.
ד. עלות הפתרון השברי היא לכל היותר עלות הפתרון בשלמים (כי בשברי $x_e \leq 1$)

כנדרש.

עיגול אקראי של ה-*LP* הראשונית (*Randomized Rounding of the Primal LP*)

באלגוריתם הקודם, ביצענו עיגול דטרמיניסטי כלפי מעלה: לקחנו כל צלע שמקיימת $x_e^* \geq 1/f$, כלומר בעצם עבורן קבענו ש- $x_e^* \geq 1$.

נשתמש בערכים של הפיתרון האופטימלי בתור הסתברויות:

1. נמצא פתרון אופטימלי לבעיית ה-*LP*. נסמן את הפיתרון $(x_e^*)_{e \in \mathcal{E}}$.
2. לכל צלע $e \in \mathcal{E}$, נצרף אותה לכיסוי A בהסתברות x_e^*.

נראה ש-A מהווה כיסוי חוקי. זאת תהיה טענה הסתברותית.

למה: (בהוכחה משתמשים בקבוע אוילר, e. נסמן אותו $\bar{e}$ כדי לא להתבלבל עם הצלע e).

לכל $v \in V$, מתקיים:

$$\mathbb{P}[v \in \cup A] \geq 1 - \frac{1}{\bar{e}}$$

הוכחה: ניזכר ב- $C_v \coloneqq \{e \in \mathcal{E} : v \in e\}$. אזי:

$$\mathbb{P}[v \notin \cup A] = \mathbb{P}[no\ e \in C_v\ was\ chosen] =^{\text{א}} \prod_{e \in C_v} (1 - x_e^*) \leq^{\text{ב}} \prod_{e \in C_v} \exp(-x_e^*) =^{\text{ג}} \exp\left(-\sum_{e \in C_v} x_e^*\right) \leq \exp(-1) = \frac{1}{\bar{e}}$$

א. כי כל בחירה של צלע היא בת"ל.
ב. כרגיל, עבור $x \in (0,1]$ מתקיים $1 - x \leq \exp(-x)$.
ג. נהפוך מחפלה של חזקות לחזקה של סכום.
ד. ניזכר ש $\sum_{e \in C_v} x_e^* \geq 1$, לפי האילוץ של הבעיה.

מסקנה: ההסתברות ש-A לא מהווה כיסוי חוקי, לפי חסם איחוד:

$$\mathbb{P}[A\ is\ not\ a\ valid\ SC] \leq \sum_{v \in V} \mathbb{P}[v \notin \cup A] \leq \sum_{v \in V} \frac{1}{\bar{e}} = \frac{n}{\bar{e}}$$

נצטרך להגביר (*amplify*) את ההסתברות להצלחה. כרגיל עם אלגוריתמים הסתברותיים, נחזור על הפעולה הפשוטה הרבה פעמים.

הגברת הסתברות – *Probability Amplification*

עבור $k \in \mathbb{N}$, נריץ את האלגוריתם k פעמים בת"ל ונקבל $A_1, A_2, \dots, A_k$ מועמדים. נגדיר את הפיתרון המאוחד, כל הצלעות שנבחרות:

$$\cup_k \coloneqq \bigcup_{i=1}^{k} A_i$$

ועכשיו לכל $v \in V$ מתקיים:

$$\mathbb{P}[v \notin \cup_k] \leq \exp(-k)$$

טענה: קיים קבוע $D > 0$ כך שאם $k \geq D \cdot \ln n$ (כלומר אם נבצע $\Omega(\ln n)$ הרצות), אז:

$$\mathbb{P}[\cup_k \ \ is\ not\ a\ valid\ SC] \leq 1/4$$

הוכחה: עבור $v \in V$ מתקיים:

$$\mathbb{P}[v \notin \cup_k] \leq \exp(-k) \leq \exp(-D \cdot \ln n) = \exp(\ln n^{-D}) = \exp\left(\ln \frac{1}{n^D}\right) = \frac{1}{n^D} \leq^{א} \frac{1}{4n}$$

א. אם D מספיק גדול.

אזי ההסתברות ש- $\cup_k$ לא מהווה כיסוי, לפי חסם איחוד היא:

$$\mathbb{P}[\cup_k \ \ is\ not\ a\ valid\ SC] \leq \sum_{v \in V} \mathbb{P}[v \notin \cup_k] \leq \frac{1}{4}$$

יחס הקירוב

בשביל ההוכחה, נשתמש באי"ש מרקוב. יהי X מ"מ ממשי אי-שלילי, ויהי $t \in \mathbb{R}^+$, אזי מתקיים:

$$\mathbb{P}[X \geq t] \leq \frac{\mathbb{E}[X]}{t}$$

למה: יהי OPT המחיר האופטימלי, ויהי k טבעי. אזי,

$$\mathbb{P}[c(\cup_k) \geq 4 \cdot k \cdot OPT] \leq 1/4$$

הוכחה: נמצא את התוחלת של העלות. בריצה אחת של האלגוריתם, עבור $e \in \mathcal{E}$, נגדיר:

$$X_e \coloneqq \begin{cases} 0, & e \notin A \\ c(e), & e \in A \end{cases}$$

אזי,

$$\mathbb{E}[X_e] = c(e) \cdot \mathbb{P}[X_e \neq 0] = c(e) \cdot x_e^*$$

ומכיוון ש: $c(A) = \sum_{e \in \mathcal{E}} X_e$, נוכל לרשום (לפי לינאריות התוחלת):

$$\mathbb{E}[c(A)] = \mathbb{E}\left[\sum_{e \in \mathcal{E}} X_e\right] = \sum_{e \in \mathcal{E}} \mathbb{E}[X_e] = \sum_{e \in \mathcal{E}} c(e) \cdot x_e^* = \mathrm{OPT}_f$$

כלומר,

$$\mathbb{E}[c(\cup_k)] \leq k \cdot \mathrm{OPT}_f$$

אז לפי אי"ש מרקוב:

$$\mathbb{P}[c(\cup_k) \geq 4 \cdot k \cdot OPT] \leq \frac{k \cdot \mathrm{OPT}_f}{4 \cdot k \cdot OPT} \leq \frac{k \cdot OPT}{4 \cdot k \cdot OPT} \leq \frac{1}{4}$$

מסקנה: אם נחזור על האלגוריתם $\Omega(\ln n)$ פעמים, נקבל כיסוי בעלות לכל היותר $O(\ln n \cdot OPT)$, בהסתברות לפחות חצי.

למה? כי עבור $k = \Omega(\ln n)$, מתקיים:

$$\mathbb{P}[\cup_k \ \ is\ a\ valid\ SC \quad \wedge \quad c(\cup_k) \leq 4 \cdot k \cdot OPT] \geq 1 - \mathbb{P}[c(\cup_k) > 4 \cdot k \cdot OPT] - \mathbb{P}[\cup_k \ \ is\ not\ a\ valid\ SC] \geq 1/2$$

אז יש לנו שני אלגוריתמים: ההסתברותי שראינו עכשיו, ודטרמיניסטי שנותן (בוודאות) קירוב $f \cdot OPT$. מה עדיף?

נשים לב ש-f (התדירות המקסימלית) ניתנת לחישוב בזמן לינארי עבור H נתון.

אז אם $f = o(\ln n)$, עדיף את הדטרמיניסטי. אם הדרגות גבוהות, עדיף את ההסתברותי.

הוכחת קירוב של האלגוריתם החמדן ע"י ה-*dual*

הבעיה הדואלית של גרסת ה-*LP* של *min-SC*: יהיה וקטור $y \geq 0$ שיש לו אינדקס לכל $v \in V$. אנחנו נרצה למקסם אותו, תחת ההגבלה:

$$\sum_{v \in e} y_v \leq c(e), \qquad \forall e \in \mathcal{E}$$

נשים לב שזו בעצם בעיית אריזה (*packing*). זה נפוץ מאוד שבעיות כיסוי ובעיות אריזה הן דואליות. לדוגמה *min-VC* ובעיית *max-match*.

ניזכר באלגוריתם החמדן:

1. נאתחל $A \coloneqq \emptyset$.
2. כל עוד $\bigcup A \neq V$:
   a. $e \coloneqq argmin\{\text{eff}_{\bigcup A}(e) : e \in E(H)\}$.
   b. עבור כל קודקוד $v \in (e \setminus \bigcup A)$, (הקודקודים החדשים ש-*e* מכסה):
      i. נגדיר $ac(v) \coloneqq \text{eff}_{\bigcup A}(e)$. (*ac = amortized cost*)
   c. $A \coloneqq A \cup \{e\}$.

טענה: יהי $n \in \mathbb{N}$. נסמן את הטור ההרמוני של *n*: $\mathcal{H}_n \coloneqq \sum_{i=1}^{n} 1/i$. נגדיר את הווקטור:

$$y \coloneqq \left(y_v \coloneqq \frac{\text{ac}(v)}{\mathcal{H}_n}\right)_{v \in V}$$

כלומר, יש אינדקס עבור כל קודקוד, והערך בכל אחד הוא ה-*ac* חלקי הטור ההרמוני של *n*.

אזי, הווקטור הזה הוא פתרון ישים עבור הבעיה הדואלית.

הוכחה:

נסמן את הסדר שבו הקודקודים כוסו ע"י האלגוריתם החמדן: $\{v_1, v_2, \dots, v_n\}$.

כבר הוכחנו שלכל $i \in [n]$. מתקיים:

$$\text{ac}(v_i) \leq \frac{OPT}{n-i+1}$$

בהינתן $e \in \mathcal{E}$ בגודל *k*, נגדיר את הסדר שבו הקודקודים של *e* כוסו: $\{v_1, v_2, \dots, v_k\}$.

(לא בהכרח אותו סדר שיש למעלה! לא אמרנו ש-*e* היא בחירה של החמדן).

נקבע קודקוד כלשהו ב-*e*: $v_i \in e = \{v_1, v_2, \dots, v_k\}$. ונתבונן ברגע לפני שהוא מכוסה.

אם *A* הוא הפתרון החמדן הנוכחי, אז מספר הקודקודים ב-*e* שעוד לא כוסו הוא:

$$|e \setminus \bigcup A| = |\{v_i, v_{i+1}, \dots, v_k\}| = |e| - i + 1 = k - i + 1$$

ננתח את ה- *amortized cost* של v_i. בגלל שאנחנו עומדים לכסות אותו, זה אומר שיש צלע כלשהי e' שהחמדן בחר לכסות, ו- $v_i \in e'$:

$$\text{ac}(v_i) = \text{eff}_{\bigcup A}(e') \overset{\text{א}}{\leq} \text{eff}_{\bigcup A}(e) \overset{\text{ב}}{=} \frac{c(e)}{|e \setminus \bigcup A|} \overset{\text{ג}}{=} \frac{c(e)}{k-i+1}$$

א. כי בחרנו את הצלע שממזערת את *eff*.
ב. הגדרת *eff*.
ג. לפי השוויון למעלה.

בסה"כ:

$$\text{ac}(v_i) \leq \frac{c(e)}{k-i+1}, \qquad |e| = k$$

עכשיו, אנחנו צריכים להראות ש-*y* הוא פתרון ישים לבעיה הדואלית.

ראשית, כל כניסה בווקטור גדולה מ-0 לפי ההגדרה של *y*. אז מתקיים $y \geq 0$, כנדרש.

צריך רק להוכיח שלכל $e \in \mathcal{E}$: $\sum_{v \in e} y_v \leq c(e)$.

נקבע $e \in \mathcal{E}$ כלשהי, ויהי $\{v_1, v_2, \ldots, v_k\}$ הסדר שבו הקודקודים שלו כוסו ע"י החמדן. לפי הגדרת y, מתקיים:

$$\sum_{v\in e} y_v = \sum_{v\in e}\frac{\mathrm{ac}(v)}{\mathcal{H}_n} \leq^{\text{א}} \sum_{i=1}^{k}\frac{c(e)}{\mathcal{H}_n(k-i+1)} =^{\text{ב}} \frac{c(e)}{\mathcal{H}_n}\sum_{i=1}^{k}\frac{1}{k-i+1} =^{\text{ג}} \frac{c(e)}{\mathcal{H}_n}\sum_{i=1}^{k}\frac{1}{i} =^{\text{ד}} \frac{c(e)}{\mathcal{H}_n}\cdot\mathcal{H}_k = \frac{\mathcal{H}_k}{\mathcal{H}_n}\cdot c(e) \leq^{\text{ה}} c(e)$$

א. נסכום לפי סדר הכיסוי, עם הטענה הקודמת שהוכחנו.
ב. נוציא החוצה מה שלא תלוי ב-i.
ג. נשים לב שהסכום הוא פשוט מ-k עד 1.
ד. הסכום הזה הוא הטור ההרמוני של k.
ה. מתקיים $\mathcal{H}_k/\mathcal{H}_n \leq 1$, כי $k \leq n$.

כנדרש.

ניגש להוכחת הטענה המרכזית: שהאלגוריתם החמדן הוא $\mathcal{H}_n$ מקרב עבור *min-cost SC*. (הטור ההרמוני הוא $O(\ln n)$).

נשתמש בטענה שאומרת ש-y הוא פתרון ישים לבעיה הדואלית. ראשית, אפשר להעביר אגפים בהגדרת y ולקבל:

$$y_v \coloneqq \frac{\mathrm{ac}(v)}{\mathcal{H}_n} \quad \Longrightarrow \quad \mathcal{H}_n\cdot y_v = \mathrm{ac}(v)$$

מתקיים שהעלות של הפתרון החמדן הוא סכום ה-ac, אז:

$$cost\ of\ greedy = \sum_{v\in V}\mathrm{ac}(v) = \sum_{v\in V}\mathcal{H}_n\cdot y_v = \mathcal{H}_n\sum_{v\in V} y_v$$

ואמרנו ש-y הוא פתרון ישים. ניזכר במשפט הדואליות החלש:

$$\max\{c^T x : Ax \leq b\} \leq \min\{y^T b : A^T y = c,\ \ y \geq 0\}$$

הפתרון המקסימלי של הדואלי (y) חסום בפתרון המינימלי של הראשוני. כלומר,

$$\mathcal{H}_n\sum_{v\in V} y_v \leq \mathcal{H}_n\cdot\mathrm{OPT}_f \leq^{\text{א}} \mathcal{H}_n\cdot OPT$$

א. כי האופטימלי של הראשוני (LP) חסום באופטימלי של ה-IP.

כנדרש.

יהיו $\mathcal{E}_1, \dots, \mathcal{E}_n$ אירועים "רעים" (באותו מרחב הסתברות). אנחנו רוצים להוכיח שההסתברות שכל האירועים לא קורים היא גדולה ממש מ-0:

$$\mathbb{P}\left[\bigcap_{i=1}^{n} \bar{\mathcal{E}}_i\right] > 0$$

נזכור שאם אירועים A, B הם בת"ל, אזי $\mathbb{P}[A \cap B] = \mathbb{P}[A] \cdot \mathbb{P}[B]$. וגם, $\bar{A}, \bar{B}$ בת"ל:

$$\begin{aligned}\mathbb{P}[\bar{A} \cap \bar{B}] &= 1 - \mathbb{P}[\bar{A} \cup \bar{B}] = 1 - (\mathbb{P}[A] + \mathbb{P}[B] - \mathbb{P}[A \cap B]) = 1 - \mathbb{P}[A] - \mathbb{P}[B] + \mathbb{P}[A \cap B] \\ &= 1 - \mathbb{P}[A] - \mathbb{P}[B] + \mathbb{P}[A] \cdot \mathbb{P}[B] = (1 - \mathbb{P}[A])(1 - \mathbb{P}[B]) = \mathbb{P}[\bar{A}] \cdot \mathbb{P}[\bar{B}]\end{aligned}$$

אם האירועים $\mathcal{E}_1, \dots, \mathcal{E}_n$ היו בת"ל בזוגות, אז גם האירועים המשלימים $\overline{\mathcal{E}_1}, \dots, \overline{\mathcal{E}_n}$ הם בת"ל בזוגות.

אם בנוסף, $\mathbb{P}[\mathcal{E}_i] < 1$ לכל $i \in [n]$, אז, **אם השוויון הראשון מתקיים** היינו מקבלים:

$$\mathbb{P}\left[\bigcap_{i=1}^{n} \bar{\mathcal{E}}_i\right] = \prod_{i=1}^{n} \mathbb{P}[\bar{\mathcal{E}}_i] = \prod_{i=1}^{n} \underbrace{(1 - \mathbb{P}[\mathcal{E}_i])}_{>0} > 0$$

אבל, בת"ל בזוגות לא בהכרח גורר את השוויון הראשון. בשביל זה, צריך משהו חזק יותר, **אי-תלות הדדית** (*mutual independence*).

אי-תלות הדדית של משתנים מקריים

מאורעות $\mathcal{E}_1, \dots, \mathcal{E}_n$ ייקראו בלתי תלויים הדדית אם לכל $\emptyset \neq I \subseteq [n]$ מתקיים:

$$\mathbb{P}\left[\bigcap_{i \in I} \mathcal{E}_i\right] = \prod_{i \in I} \mathbb{P}[\mathcal{E}_i]$$

ניסוח אחר:

נאמר שאירוע A הוא בת"ל הדדית מ- $\mathcal{E}_1, \dots, \mathcal{E}_n$ אם לכל $\emptyset \neq I \subseteq [n]$ מתקיים:

$$\mathbb{P}\left[A \mid \bigcap_{i \in I} \mathcal{E}_i\right] = \mathbb{P}[A]$$

אז אם כל $\mathcal{E}_i$ הוא בת"ל הדדית מ- $\{\mathcal{E}_1, \dots, \mathcal{E}_n\} \setminus \{\mathcal{E}_i\}$, נאמר ש- $\mathcal{E}_1, \dots, \mathcal{E}_n$ בת"ל הדדית.

עוד ניסוח:

נאמר שאירוע A הוא בת"ל הדדית מ- $\mathcal{E}_1, \dots, \mathcal{E}_n$ אם לכל $i \in [n]$, עבור $B_i \in \{\mathcal{E}_i, \bar{\mathcal{E}}_i\}$ מתקיים:

$$\mathbb{P}\left[A \mid \bigcap_{i=1}^{n} B_i\right] = \mathbb{P}[A]$$

כלומר, זה לא משנה האם האירוע $\mathcal{E}_i$ קרה או לא, זה לא משפיע על ההסתברות של A.

כמו עם אי-תלות בזוגות, אם $\mathcal{E}_1, \dots, \mathcal{E}_n$ בת"ל הדדית אז גם המשלימים $\overline{\mathcal{E}_1}, \dots, \overline{\mathcal{E}_n}$ הם בת"ל הדדית.

עקרון אי-תלות הדדית – *The Mutual Independence Principle*

תהי סדרה של ניסויים בת"ל בזוגות: $\mathcal{X} := (X_1, \dots, X_m)$. ויהיו קבוצת מאורעות $\{A_1, \dots, A_n\}$. כאשר כל מאורע נקבע ע"י קבוצת ניסויים $F_i \subseteq \mathcal{X}$.

בהינתן $I \subseteq [n]$, $j \in [n] \setminus I$ (ניקח קבוצה של מאורעות, ועוד מאורע שלא בקבוצה), אם מתקיים:

$$F_j \cap (\cup_{i \in I} F_i) = \emptyset$$

אז A_j בת"ל הדדית מ- $(A_i)_{i \in I}$.

זאת דרך לבדוק אי-תלות דרך הניסויים עצמם, בלי צורך לדעת את ההסתברויות.

מה אם $\mathcal{E}_1, \dots, \mathcal{E}_n$ כן תלויים אחד בשני באופן מסויים? **למת הלוקאליות של לובאס** נותנת לנו פתרון חלקי.

למת הלוקאליות של לובאס

יש כמה גרסאות. אנחנו נשתמש ב**גרסה הסימטרית.** נתמקד בשימוש של הלמה ולא בהוכחה.

בשביל הלמה, נשתמש ב**גרף תלויות** – ***Dependency Graph:***

עבור מאורעות $\mathcal{E}_1, \dots, \mathcal{E}_n$, **גרף התלויות** שלהם $D \coloneqq D(\mathcal{E}_1, \dots, \mathcal{E}_n)$ הוא **כל גרף** שהקודקודים שלו הם $\mathcal{E}_1, \dots, \mathcal{E}_n$, ושמקיים את התכונה:

לכל $i \in [n]$, המאורע $\mathcal{E}_i$ הוא בת"ל הדדית מכל המאורעות שאינם סמוכים לו בגרף: $\{\mathcal{E}_j : ij \notin E(D)\}$.

נשים לב: לא בנינו את הגרף במיוחד בצורה מסויימת עם צלעות לפי התלויות. כל גרף שמקיים את התכונה, הוא גרף תלויות.

בגרסאות א-סימטריות של הלמה, משתמשים גרסה מכוונת של הגרף.

הלמה עצמה:

יהיו מאורעות $\mathcal{E}_1, \dots, \mathcal{E}_n$ כך שמתקיימים כל הדברים הבאים:

1. **סימטריות:** מתקיים $\mathbb{P}[\mathcal{E}_i] \le p$ לכל $i \in [n]$. (בגרסה הא-סימטרית, זה p_i שיכול להיות שונה לכל מאורע).
2. **תלות מוגבלת:** בגרף תלויות של המאורעות, יש חסם על הדרגה המקסימלית: $\Delta\big(D(\mathcal{E}_1, \dots, \mathcal{E}_n)\big) \le d$.
3. **חסם על p, d:** מתקיים: $e \cdot p \cdot (d+1) \le 1$. (כאן, e זה קבוע אוילר). לפעמים לוקחים חסם קשוח יותר: $4pd \le 1$.

אזי,

$$\mathbb{P}\left[\bigcap_{i=1}^{n} \bar{\mathcal{E}}_i\right] > 0$$

הסבר אינטואיטיבי: אם יש לנו מאורעות "רעים", שאנחנו לא רוצים שיקרו, מה ההסתברות שבאמת כולם לא קרו?

הלמה של לובאס לא נותנת לנו את ההסתברות, אבל הוא יכולה להבטיח שזו הסתברות גדולה מאפס. כלומר יש סיכוי שמשהו טוב קרה.

הלמה בעצם תופסת גם למקרים שיש תלויות, אבל לא יותר מדי. זה התנאי השני.

התנאי הראשון מחייב שההסתברות לכל מאורע לא גדולה מדי.

התנאי השלישי משלב ביניהם.

בעיית **הספיקות** – מתי פסוק ניתן לסיפוק?

נזכיר: פסוק *k-CNF* היא במבנה: $\underbrace{(x_1 \vee x_2 \vee x_5 \vee \dots)}_{k} \wedge \underbrace{(\overline{x_1} \vee x_2 \vee \overline{x_7} \vee \dots)}_{k} \wedge \dots$

למה: תהי φ נוסחת *k-CNF* שבה **אין** משתנה שמופיע ביותר מ- $2^k/4k$ פסוקיות. אזי, φ ניתנת לסיפוק.

הוכחה: נסמן ב-m את מספר הפסוקיות ב-φ.

לכל משתנה ב-φ, נקבע ערך ב- $\{0,1\}$ באופן מקרי ואחיד ובת"ל.

לכל $i \in [m]$, נגדיר $\mathcal{E}_i$ את המאורע "פסוקית i לא מסופקת".

אנחנו בעצם רוצים להראות ש $\mathbb{P}\left[\bigcap_{i\in[m]} \bar{\mathcal{E}_i}\right] > 0$. וזה מתאים בדיוק ללמת המקומיות של לובאס.

נצטרך לבנות גרף תלויות שמתאים למאורעות, ולהראות ש-3 התנאים מתקיימים:

1. סימטריות: מתקיים $\mathbb{P}[\mathcal{E}_i] \le p$ לכל $i \in [n]$.
2. תלות מוגבלת: $\Delta\big(D(\mathcal{E}_1, \dots, \mathcal{E}_n)\big) \le d$ (הדרגה המקסימלית).
3. חסם על p, d: $4pd \le 1$.

בניית הגרף *D*

$$V = V(D) \coloneqq \{\mathcal{E}_i : i \in [m]\}, \qquad E = E(D) \coloneqq \big\{(\mathcal{E}_i, \mathcal{E}_j) : \textit{clauses } i,j \textit{ share common variables}\big\}$$

הקודקודים הם המאורעות של "פסוקית לא מסופקת". בין שני קודקודים, תהיה צלע אם יש משתנה משותף בין הפסוקיות המתאימות.

מאורע $\mathcal{E}_i$ הוא בת"ל הדדית מכל המאורעות $\mathcal{E}_j$ אם לפסוקיות שלהם אין משתנים משותפים, כי אז ההשמה של כל פסוקית j לא תשפיע על ההשמה של i.

אז הגרף הוא אכן גרף תלויות, כי כל מאורע יהיה בת"ל במכל המאורעות שלא סמוכים אליו בגרף.

<u>נקבע את *d*</u>: לפי ההנחה, אין משתנה שמופיע **ביותר מ-** $2^k/4k$ פסוקיות.

אז המקרה הגרוע ביותר שיכול להיות זה אם יש פסוקית שבה יש k משתנים שונים, וכל אחד מהם מופיע ב- $2^k/4k$ פסוקיות:

$$\Delta(D) \le k \cdot \frac{2^k}{4k} = \frac{2^k}{4} = 2^{k-2} = d$$

<u>נקבע את *p*</u>: ההסתברות שפסוקית לא תהיה מסופקת, היא ההסתברות שכל אחד מהמשתנים שלה לא מסופקים:

$$\mathbb{P}[\mathcal{E}_i] \le 2^{-k}, \qquad \forall i \in [m]$$

<u>נבדוק האם החסם על *p, d* מתקיים</u>:

$$4pd \le 4 \cdot 2^{k-2} \cdot 2^{-k} = 2^2 \cdot 2^{-2} = 1$$

אזי לפי למת המקומיות של לובאס, מתקיים $\mathbb{P}\left[\bigcap_{i=1}^{n} \bar{\mathcal{E}_i}\right] > 0$. כלומר יש הסתברות חיובית ממש שכל הפסוקיות מסופקות. כנדרש.

אלגוריתם מקרי לספיקות – *A randomized Algorithm for Satisfiability*

הלמה של לובאס מבטיחה קיום של השמה מספקת. נרצה להשתמש בלמה כדי למצוא השמה כזו.

הראינו שאם כל משתנה מופיע בלכל היותר $2^k/4k$ פסוקיות, אז φ ספיקה. נרשום עבור $\alpha < 1$ כלשהו:

$$\frac{2^k}{4k} = \frac{2^k}{2 \cdot 2 \cdot k} = \frac{2^{k-2}}{k} = 2^{\alpha k}$$

נרצה למצוא השמה ספציפית. בשביל זה נצטרך עוד הנחות לגבי α. יהיו שני שלבים:

<u>בשלב הראשון</u>, נייצר השמה חלקית אקראית, ונוכיח (עם הלמה של לובאס) שבהסתברות חיובית ההשמה החלקית ניתנת להרחבה להשמה מספקת.

<u>בשלב השני</u>, נראה שבהסתברות ששואפת ל-1, יש **גרף תלויות** שההשמה החלקית מפרקת אותו לרכיבי קשירות "קטנים".

ואז, נחשוב על כל רכיב קשירות בתור תת-נוסחה של φ.

ואם בכל מרכיב יש מעט מאד משתנים, נוכל למצוא השמה מספקת בזמן שהוא אקספוננציאלי בגודל המרכיב, אבל בגלל שהרכיבים קטנים הוא כמעט קבוע.

<u>התוצאה המרכזית שנגיע אליה</u>:

לכל k זוגי מספיק גדול וקבוע, קיים $\alpha \coloneqq \alpha(k) > 0$ כך שמתקיים:

> תהי φ נוסחת k-CNF עם m פסוקיות, כך שכל משתנה מופיע בלכל היותר $2^{\alpha k}$ פסוקיות.
>
> אזי בהסתברות 1, אפשר למצוא השמה מספקת ל-φ, כך שתוחלת זמן הריצה הוא פולינומי ב-m.

<u>הגדרות</u>:

- φ היא נוסחת k-CNF עבור $k \geq 4$ זוגי.
- המשתנים הם $x_1, \dots, x_\ell$.
- הפסוקיות הן $C_1, \dots, C_m$.
- כל משתנה מופיע בלכל היותר $2^{\alpha k}$ פסוקיות.
- $0 < \alpha < 1$, נקבע אותו בהמשך.

<u>הנחות</u>:

- אין פסוקית שמכילה זוגות של ליטרלים משלימים כמו $x, \bar{x}$. (אם יש, אז אותה פסוקית בוודאות מסופקת, ונוכל פשוט להוריד את הפסוקית הזו).
- אין פסוקית עם ליטרלים שחוזרים על עצמם. (כי אם יש, אפשר פשוט לאחד אותם).

כלומר באופן כללי, בכל פסוקית מספר הליטרלים שווה למספר המשתנים.

ההנחות האלה נותנות לנו שבכל פסוקית יהיו בדיוק k משתנים לאורך ההוכחה. אפשר להכליל את ההוכחה כך שבכל פסוקית i יש k_i ליטרלים.

השמה חלקית אקראית (שלב א)

נגדיר: פסוקית C תיקרא **מסוכנת** אם יש לה <u>בדיוק</u> $k/2$ ליטרלים שקיבלו ערכים והיא עדיין לא מסופקת.

1. נעבור על המשתנים $x_1, \dots, x_n$. לכל x_i:
   a. אם קיימת פסוקית מסוכנת שמכילה את x_i, אז לא ניתן ל-x_i השמה ונעבור למשתנה הבא.
   b. אחרת, אז ניתן ל-x_i ערך 0 או 1 באופן מקרי ואחיד ובת"ל מכל השאר.

בכל איטרציה אנחנו קובעים השמה למשתנה אחד בלבד.

משתנה שלא קיבל ערך במהלך שלב א ייקרא משתנה **דחוי**.

נאמר שפסוקית **שרדה** את שלב א אם היא לא מסופקת ע"י ההשמה החלקית שהוגדרה במהלך שלב א.

פסוקית שורדת לא בהכרח חייבת להיות מסוכנת: אם יש לה מספיק משתנים משותפים עם פסוקית מסוכנת, אז יכול להיות שיש לה פחות מ- $k/2$ משתנים שקיבלו ערכים.

לפסוקית שורדת תמיד יהיו לכל היותר $k/2$ משתנים שקיבלו ערך (נקבעו). למה?

ראשית, נשים לב שפסוקית מסופקת לא תחזור להיות לא-מסופקת. כי יש "או" בין המשתנים של הפסוקית.

אם פסוקית מוגדרת שורדת, היא בהכרח לא מסופקת.

ואם יש לה יותר מ- $k/2$ משתנים שנקבעו, זה אומר שבשלב מסויים היו לה בדיוק $k/2$ משתנים שנקבעו, והיא הייתה לא מסופקת.

אז באותו רגע היא הייתה מסוכנת. וברגע שפסוקית נהיית מסוכנת, בהגדרה אף משתנה אצלה לא יקבל ערך.

אז לא יכול להיות שיש לה יותר מ- $k/2$ משתנים שקיבלו ערך.

וגם נשים לב, שאם יש פסוקית שכל המשתנים שלה אף פעם לא היו משותפים עם פסוקית מסוכנת, אז כל המשתנים שלה יקבלו ערכים.

כלומר, כל פסוקית שורדת היא או מסוכנת בעצמה, או שיש לה משתנה דחוי שמשותף עם פסוקית מסוכנת.

כלומר היה שלב שבו היו לה בדיוק $k/2$ ליטרלים שקיבלו ערכים והיא עדיין לא מסופקת. אז היא הייתה מוגדרת מסוכנת.

בסה"כ, לכל פסוקית שורדת יש לפחות $k/2$ משתנים דחויים.

השלמה להשמה מספקת (המשך שלב א)

טענה: לכל k זוגי מספיק גדול וקבוע, קיים $\alpha > 0$ כך שקיימת השמה למשתנים הדחויים שמספקת את כל הפסוקיות השורדות.

הוכחה: נשתמש בלמת המקומיות כמו שעשינו בהתחלה, עם כמה שינויים:

<u>מקור האקראיות</u>: לא נשתמש בהשמה האקראית מחלק א. אלא, נקבע לכל משתנה דחוי ערך 0 או 1 באופן מקרי ואחיד ובת"ל מכל השאר.

עבור פסוקית שורדת C, נגדיר: $\mathcal{E}_C$ את המאורע "C לא סופקה על ידי ההשמה האקראית השנייה".

<u>גרף תלויות</u>: הקודקודים יהיו כל ה-$\mathcal{E}_C$. שני קודקודים $\mathcal{E}_C$ ו- $\mathcal{E}_{C'}$ יהיו סמוכים אם יש ל- C, C' משתנה דחוי משותף.

ההוכחה ש-D הוא אכן גרף תלויות זהה למה שעשינו בפעם הראשונה.

ההסתברות ש-C לא מסופקת: $\mathbb{P}[\mathcal{E}_C] \le 2^{-k/2}$. כי צריך שכל אחד מהמשתנים הדחויים לא יסופק.

הדרגה המקסימלית: $\Delta(D) \le k \cdot 2^{\alpha k}$. כי יש עד k משתנים דחויים בפסוקית שורדת, וכל אחד מהם נמצא בלכל היותר $2^{\alpha k}$ פסוקיות שורדות אחרות.

החסם על d, p (ההסתברות ל- $\mathcal{E}_C$ והחסם על הדרגה המקסימלית):

$$4dp = 4 \cdot \Delta(D) \cdot 2^{-k/2} \le 4 \cdot k \cdot 2^{\alpha k} \cdot 2^{-k/2} = k \cdot 2^{\alpha k + 2 - k/2} = 2^{\alpha k + 2 + \log_2 k - k/2} \le 1$$

כאשר המעבר האחרון מתקיים עבור k מספיק גדול ו-α מספיק קטן (כי אז ה- $(-k/2)$ יותר משמעותי משאר החלקים, והחזקה שלילית).

עכשיו, למת המקומיות אומרת שיש הסתברות חיובית ממש שכל פסוקית שורדת תסופק.

מרכיבי הקשירות של גרף התלויות (שלב ב)

כל רכיב קשירות של D מגדיר תת-נוסחה של φ (אוסף של חלק מהפסוקיות). ובין רכיבי קשירות, אין משתנים דחויים משותפים.

מכיוון שיש m פסוקיות, יש לכל היותר m רכיבים כאלה.

אם נצליח להראות שכל אחד מהרכיבים האלה בגודל $O(\log m)$, אז יש לכל היותר $O(k \log m)$ משתנים בפסוקיות של הרכיבים האלה.

ואז, בגלל שאנחנו יודעים ש-φ ספיקה, אז חיפוש *brute-force* על כל $2^{O(k \log m)} = m^{O(k)}$ ההשמות האפשריות ייתן השמה מספקת.

אז אם k קבוע, האלגוריתם הוא פולינומי.

המטרה שלנו היא להוכיח שכל זה קורה בהסתברות ששואפת ל-1. (נאמר שזה קורה <u>כמעט בוודאות</u>. *a.a.s. – asymptotically almost surly*)

למה: לכל k כמו שאמרנו קיים $\alpha \coloneqq \alpha(k) > 0$ ו- $C \coloneqq C(k, \alpha) > 0$ כך שכמעט בוודאות כל הרכיבים של D בגודל לכל היותר $C \cdot \ln m$.

הוכחה: כדי להוכיח, נשתמש בגרף D' שהוא דומה לגרף תלויות, רק שהקודקודים שלו הם לא מאורעות, אז אין מקריות.

$V(D')$ הם הפסוקיות של φ. 2 קודקודים יהיו סמוכים אם בין הפסוקיות יש משתנים משותפים.

כל הקודקודים של D (פסוקיות שלא מסופקות אחרי ההשמה השנייה) מיוצגים גם ב-D'.

ואם יש צלע בין שני קודקודים ב-D (כי יש להם משתנה דחוי משותף), אז יש גם צלע ב-D' כי יש משתנה משותף. כלומר $D \subseteq D'$, אז $\Delta(D) \le \Delta(D')$.

וגם, $\Delta(D') \le k \cdot 2^{\alpha k}$ (כי יש k משתנים בכל פסוקית, וכל אחד נמצא בלכל היותר $2^{\alpha k}$ פסוקיות אחרות). נסמן $\Delta \coloneqq \Delta(D') \le k \cdot 2^{\alpha k}$.

4-Trees

כדי לנתח את הרכיבים של D', נשתמש בגרף עזר מסוג "עץ-4".

נגדיר:

R תת-גרף קשיר של D' (יכול להיות רכיב קשירות).

אז, עץ-4 של R הוא עץ מושרש (כלומר שיש לו קודקוד מסוים שמוגדר בתור שורש) S, שמקיים:

1. $V(S) \subseteq V(R)$
2. כל שני צמתים ב-S הם במרחק לפחות 4 צלעות ב- D'.
3. שני קודקודים הם סמוכים ב-S אם ב-D' המרחק שלהם הוא בדיוק 4.
4. כל קודקוד של R הוא או ב-S או במרחק לכל היותר 3 מ-$V(S)$ ב-R. כלומר ב-R, אפשר להגיע מכל קודקוד לקודקוד ששייך גם ל-S ב-3 צעדים.

נשים לב – S לא חייב להיות בעצמו תת-גרף של R.

ננסח 2 טענות, נשתמש בהן, ואז נוכיח בסוף.

טענה א: יהי S עץ-4 של תת-גרף קשיר של D'. אזי, $V(S) \subseteq V(D)$ (כלומר כל הפסוקיות של S שרדו ב-D) בהסתברות לכל היותר:

$$\mathbb{P} \leq \left((\Delta+1) \cdot 2^{-k/2}\right)^{v(S)} = \left(\left(k \cdot 2^{\alpha k}+1\right) \cdot 2^{-k/2}\right)^{v(S)} \approx$$

אם כל הפסוקיות של S שרדו, נאמר ש-S שרד.

נשים לב ש $\Delta = k \cdot 2^{\alpha k}$. וה-1 לא משמעותי במיוחד (אם כל שאר הדברים גדולים מספיק). אז זה בערך:

$$\approx \left(2^{\alpha k+\log_2 k-k/2}\right)^{v(S)}$$

ודומה למה שהראינו מקודם, אם k מספיק גדול אז החזקה תהיה שלילית. אז יש לנו מספר קטן מ-1 בבסיס החזקה, שזה טוב בשביל חסם על ההסתברות. כאשר $v(S)$ יגדל, נוכל לומר שכמעט בוודאות עץ בגודל כזה לא שורד.

טענה ב: יהי R תת-גרף קשיר של D', ויהי S עץ-4 של R בגודל מקסימום. אזי, $v(S) \geq v(R)/\Delta^3$.

כלומר אם ב-R יש הרבה קודקודים, אז ב-S יש הרבה קודקודים.

ולפי טענה א, אם ב-S יש הרבה קודקודים, אז ההסתברות שהפסוקיות שלו שרדו מאוד נמוכה.

כזכור, אנחנו רוצים להוכיח:

לכל k כמו שאמרנו קיים $\alpha := \alpha(k) > 0$ ו- $C := C(k,\alpha) > 0$ כך שכמעט בוודאות כל הרכיבים של D בגודל לכל היותר $C \cdot \ln m$.

נשתמש בטענה א כדי להוכיח שכמעט בוודאות, D' לא מכיל עץ-4 בעל לפחות $C \ln m/\Delta^3$ קודקודים ששרד ב-D.

השרידות הזו היא מה שמביא את האקראיות.

מטענה ב נקבל שאם תת-גרף קשיר R של D' בגודל לפחות $C \ln m$ כלשהו שרד ב-D, אז יש לו עץ-4 בגודל לפחות $C \ln m/\Delta^3$ ששורד ב-D.

אבל לפי טענה א, עצים בגודל כזה כמעט בוודאות לא שורדים ב-D.

אז כמעט בוודאות, אין תת-גרף קשיר של D' כזה ששרד ב-D. אז כל מרכיבי הקשירות של D הם בגודל לכל היותר $C \cdot \ln m$ (כמעט בוודאות).

הוכחת טענה א

תהי C פסוקית של φ. נשאל מה ההסתברות שהיא שורדת ב-D.

כזכור, פסוקית שורדת היא או מסוכנת בעצמה או חולקת לפחות משתנה דחוי אחד עם פסוקית מסוכנת.

אם היא חולקת לפחות משתנה דחוי אחד עם פסוקית מסוכנת, אז בגרף D', היא סמוכה לפסוקית מסוכנת. אז ההסתברות ש-C שורדת:

$$\mathbb{P}[C\ survives] \leq \underbrace{\mathbb{P}[C\ is\ dangerous]}_{\leq 2^{-k/2}} + \underbrace{\mathbb{P}[\geq 1\ of\ its\ neighbors\ in\ D'\ is\ dangerous]}_{\leq \Delta \cdot 2^{-k/2}} \leq (\Delta+1) \cdot 2^{-k/2}$$

פסוקיות במרחק 2 (או יותר) ב-D' לא חולקות משתנים. אז המאורעות שפסוקיות כאלה הן מסוכנות, הם בלתי תלויים הדדית.

יהי $u \in V(S)$. כל קודקוד ב-R שיכולים לגרום ל-u לשרוד נמצאים במרחק 1 מ-u (ב-R).

ולכן הם במרחק לפחות 2 מכל הקודקודים האחרים של $V(S)$ (כי כל שני צמתים ב-S הם במרחק לפחות 4 ב-R).

כלומר, מאורעות השרידות של $V(S)$ כולם בת"ל הדדית. אז ההסתברות שכולם שרדו:

$$\mathbb{P}[V(S) \subseteq V(D)] \leq \left((\Delta+1) \cdot 2^{-k/2}\right)^{v(S)}$$

כנדרש לטענה א.

הוכחת טענה ב

ניזכר בטענה: כל עץ-4 של תת-גרף קשיר R הוא בגודל לפחות $v(R)/\Delta^3$.

נב"ש שגודל העץ-4 המקסימום ב-R (נקרא לו S) קטן ממש מ- $v(R)/\Delta^3$.

בהגדרה, כל קודקוד ב- $V(R)$ הוא במרחק לכל היותר 3 מ- $V(S)$.

יהי $v \in V(D')$ כלשהו. נסמן n_3 את מספר הקודקודים ב-D' במרחק לכל היותר 3 מ-v:

$$n_3 \le \Delta + \Delta(\Delta-1) + \Delta(\Delta-1)(\Delta-2) = \Delta + \Delta^2 - \Delta + \Delta^3 - \Delta^2 - 2\Delta^2 + 2\Delta \le \Delta^3 - 1$$

- ה-Δ זה בגלל החסם על מספר השכנים.
- ה-$\Delta(\Delta-1)$ זה חסם על מספר הקודקודים במרחק 2 (Δ שכנים של v, ולכל אחד לכל היותר $\Delta-1$ שכנים כי הם לא שכנים של v).
- ובאופן דומה עבור קודקודים במרחק 3.

אז בגלל שכל הקודקודים ב-R הם במרחק לכל היותר 3 מקודקודי S, נקבל:

$$v(R) \le v(S)\cdot n_3 <$$

ולפי ההנחה בשלילה, כל עץ-4 קטן ממש מ-$v(R)/\Delta^3$, אז:

$$< \frac{v(R)}{\Delta^3}\cdot(\Delta^3-1) < v(R)$$

סתירה.

גודל המרכיבים של D

ניזכר: רצינו להשתמש בטענה א כדי לומר שכמעט בוודאות אין ב-D' עץ-4 בגודל $C\log m/\Delta^3$ ששרד ב-D.

נגדיר $r := C\ln m$.

נגדיר $\mathcal{E}$ את המאורע שעץ-4 בגודל לפחות $C\ln m/\Delta^3$ שרד ב-D. מה ההסתברות לזה? לפי חסם איחוד:

$$\mathbb{P}[\mathcal{E}] \le (\#of\ 4\text{-}trees\ in\ D'\ of\ size\ r/\Delta^3)\cdot(max\ prob.\ that\ a\ 4\text{-}tree\ survives)$$

מספר העצים ב-D' בגודל r/Δ^3 הוא לכל היותר:

$$m\cdot\Delta^{4r/\Delta^2}$$

נוכיח: נספור את מספר הדרכים לבנות עץ-4 בגודל r/Δ^3. בעצם בהינתן מבנה של עץ, נבחר איזה קודקוד לשים בכל צומת.

בהינתן תבנית, נספור את מספר הדרכים לבנות את העץ בצורה סדורה – כאשר יש משמעות לסדר בין הילדים.

במקום לספור עצים, נספור את מספר מעגלי האויילר על הקודקודים של העץ.

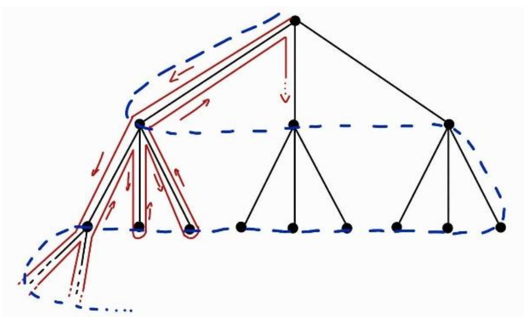
נספור מעגלים שמתחילים בשורש של העץ והולכים לפי הסדר שקבענו (נגיד $preorder$).

זה יהיה חסם עליון על מספר הדרכים.

יש m דרכים לבחור את השורש (כל הקודקודים ב-D', הפסוקיות מ-φ).

כל צלע בעץ מסמלת מסלול באורך 4. אז כדי לבחור את הקודקוד הבא, יש לכל היותר Δ^4 אפשרויות.

נשים לב שאנחנו סופרים כל פעם שחוזרים לצומת, למרות שבבנייה הצומת כבר קבועה:



תהליך הבנייה זה הקו הרצוף, לפי החיצים. המסלול אויילר זה הקו המקווקו, שזה פחות מהמסלול של הבנייה.

אז מספר מסלולי אוילר האפשריים הוא לכל היותר:

$$m\cdot(\Delta^4)^{\Delta\cdot r/\Delta^3} = m\cdot\Delta^{4r/\Delta^2}$$

כי מכל אחד מה- r/Δ^3 מקומות, יוצאים לכל היותר Δ צלעות. ולכל צלע יש לכל היותר Δ^4 אפשרויות.

אזי, ההסברות של $\mathcal{E}$:

$$\mathbb{P}[\mathcal{E}] \leq (\#of\ 4\text{-}trees\ in\ D'\ of\ size\ r/\Delta^3\ ) \cdot (max\ prob.\ that\ a\ 4\text{-}tree\ survives) \leq m \cdot \Delta^{4r/\Delta^2} \cdot \left((\Delta+1)\cdot 2^{-k/2}\right)^{r/\Delta^3}$$
$$= \exp\left(\ln m + \ln\left(\Delta^{4r/\Delta^2}\right) + \ln\left(\left((\Delta+1)\cdot 2^{-k/2}\right)^{r/\Delta^3}\right)\right)$$
$$= \exp\left(\ln m + \frac{4r}{\Delta^2}\ln\Delta + \ln\left((\Delta+1)^{r/\Delta^3}\cdot\left(2^{-k/2}\right)^{r/\Delta^3}\right)\right)$$
$$= \exp\left(\ln m + \frac{4r}{\Delta^2}\ln\Delta + \ln\left((\Delta+1)^{r/\Delta^3}\cdot 2^{-kr/2\Delta^3}\right)\right) = \exp\left(\ln m + \frac{4r}{\Delta^2}\ln\Delta + \frac{r}{\Delta^3}\ln(\Delta+1) - \frac{kr}{2\Delta^3}\ln 2\right) =$$

ניזכר שהצבנו: $\Delta \leq k\cdot 2^{\alpha k}$.

וניזכר ש $r := C\ln m$

$$\leq \exp\left(\ln m + \frac{4r}{\Delta^2}\ln\left(k\cdot 2^{\alpha k}\right) + \frac{r}{\Delta^3}\ln\left(k\cdot 2^{\alpha k}+1\right) - \frac{kr}{2\Delta^3}\ln 2\right) \leq \exp\left(\ln m + \frac{4r\ln(2k)\,\alpha k}{\Delta^2} + \frac{r\ln(2k+1)\,\alpha k}{\Delta^3} - \frac{kr}{2\Delta^3}\right) \leq$$
$$\leq \exp\left(\ln m + \frac{4r\ln(2k)\,\alpha k}{\Delta^2} + \frac{r\ln(4k)\,\alpha k}{\Delta^3} - \frac{k\cdot C\ln m}{2\Delta^3}\right)$$

אם α תהיה מספיק קטנה, וניקח C מספיק גדול, אז הביטוי השלילי בסוף יהיה חזק יותר מכל השאר, ונקבל שכל זה שואף לאפס, או $o(1)$.

כנדרש.

לסיכום

טענה:

לכל k זוגי מספיק גדול וקבוע, קיים $\alpha := \alpha(k)$ כך שמתקיים:

תהי φ נוסחת *k-CNF* עם m פסוקיות, כך שכל משתנה מופיע בלכל היותר $2^{\alpha k}$ פסוקיות.

אזי בהסתברות 1, אפשר למצוא השמה מספקת ל-φ בזמן שבתוחלת הוא פולינומי ב-m.

ההוכחה:

השלב הראשון נותן פירוק של φ לפסוקיות קטנות בגודל $O(\ln m)$. (בהסתברות ששואפת ל-1). זה נובע מטענות א ו-ב.

כלומר בהסתברות לפחות $1 - o(1)$ אנחנו מצליחים לפרק את גרף התלויות לחלקים מספיק קטנים.

ההתפלגות של הניסיונות לפירוק זו התפלגות $Geom\left(1-o(1)\right)$.

אז בתוחלת, תוך $O(1)$ ניסיונות של שלב א נקבל פירוק תקין.

והראינו שכל פירוק שנקבל, אפשר להרחיב להשמה מספקת.

אז נוכל ע"י *brute force* לבדוק את כל ה- $2^{O(k\log m)} = m^{O(k)}$ השמות האפשריות, שעבור k קבוע זה זמן פולינומי ב-m.

מש"ל. ■

20: Solving 2-Satisfiability Using Markov Chains

ניזכר בבעיה מתחילת הקורס, השאלה האם נוסחת *2-CNF* היא ספיקה. ראינו אלגוריתם דטרמיניסטי.

נציע אלגוריתם אקראי, וננתח אותו ע"י **שרשראות מרקוב** – *Markov Chains*

תהליך סטוכסטי (*Stochastic Process*) הוא סדרה: $\boldsymbol{X} \coloneqq \{X_t : t \in T\}$, כאשר:

T היא קבוצה שממספרת את הזמן. יכולה להיות אינסופית.

לכל $t \in T$, X_t הוא משתנה מקרי.

אפשר לחשוב על הסדרה $\boldsymbol{X}$ כולה בתור משתנה מקרי X, כאשר לכל $t \in T$, X_t הוא המצב של X בזמן t.

אם T בת-מניה, אז נקרא ל-$\boldsymbol{X}$ תהליך דיסקרטי (*discrete time process*).

אם כל ה- X_t מקבלים ערכים מקבוצה סופית משותפת כלשהי, אז $\boldsymbol{X}$ נקרא **תהליך סופי.**

אנחנו נתמקד בסוג מסויים של תהליך סטוכסטי: **שרשראות מרקוב.**

דוגמה קלאסית לשרשראות מרקוב היא סדרה של מ"מ בת"ל: $(X_n : n \in \mathbb{N})$.

יש לה את התכונה הבאה: לכל זמן $n \in \mathbb{N}$ ולכל ערך אפשרי של t, מתקיים:

אפשר לחשב את $\mathbb{P}[X_n = t]$ בצורה בת"ל מההיסטוריה שהובילה אותנו לזמן n.

תמונה שאפשר לחשוב עליה היא חלקיק (*particle*), שנמצא במצב כלשהו וקופץ בין מצבים.

התהליך הסטוכסטי מנסה להבין את ההתנהגות של החלקיק, ובעיקר להבין את ההתנהגות באינסוף – לאן הוא מתכנס.

שרשרת מרקוב

סדרה של מ"מ $(X_n : n \in \mathbb{N})$ תיקרא שרשרת מרקוב אם לכל נקודת זמן $n \in \mathbb{N}$ מתקיים:

התהליך העתידי $(X_m : m > n)$ הוא בת"ל מהתהליכים בעבר $(X_m : m < n)$, והאי-תלות הזו מתקבלת בתנאי שאנחנו יודעים את X_n.

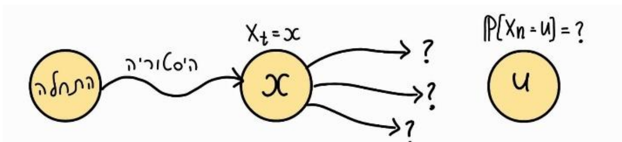
נשים לב: יכולה להיות תלות בין העתיד לעבר, בתנאי שכל התלויות האלה נתפסות דרך X_n.

לדוגמה, הילוך אקראי על גרף:

בכל קודקוד שנמצאים בו, בוחרים שכן באופן מקרי ואחיד ועוברים אליו.

אם נרצה לחשב את $\mathbb{P}[X_n = u]$ (קרי: ההסתברות שבזמן n נהיה בקודקוד u) נצטרך לקחת בחשבון את כל ההילוכים האפשריים מקודקוד ההתחלה ל-u, שהם באורך לכל היותר n. בחישוב כזה, כל ההיסטוריה של ההילוך היא קריטית.

לעומת זאת, אם נתנה את החישוב בכך שבזמן t נמצאים בקודקוד x (שניתן להגיע אליו מהקודקוד ההתחלתי), אז:



המאורע $\{X_n = u \mid X_t = x\}$ הוא בת"ל מהסדרה $X_0, X_1, \dots, X_{t-1}$.

אפשר לחשב את $\mathbb{P}[X_n = u \mid X_t = x]$ בלי לדעת את ההיסטוריה עד $t - 1$.

אפיון שרשראות מרקוב

משפט: יהי $\boldsymbol{X} \coloneqq (X_n : n \in \mathbb{Z}_{\geq 0})$ תהליך דיסקרטי. ותהי S קבוצת המצבים האפשריים של X_n. אזי,

$\boldsymbol{X}$ הוא שרשרת מרקוב **אמ"מ** לכל נקודת זמן $n \in \mathbb{Z}_{\geq 0}$ ולכל מצב $i_0, \dots, i_{n+1} \in S$ מתקיים:

$$\mathbb{P}[X_{n+1} = i_{n+1} \mid X_0 = i_0, X_1 = i_1, \dots, X_n = i_n] = \mathbb{P}[X_{n+1} = i_{n+1} \mid X_n = i_n]$$

כלומר, במקום לשאול על כל ההיסטוריה, מספיק לשאול רק על השלב הקודם.

השוויון הזה נקרא **תכונת מרקוב** או **תכונת חוסר זיכרון** של השרשרת. (ייקרא גם תהליך מרקובי).

וההסתברות $\mathbb{P}[X_{n+1} = i_{n+1} \mid X_n = i_n]$ נקראת **הסתברות המעבר** (*transition probability*).

הסתברות משותפת בשרשראות מרקוב

תכונת מרקוב מאפשרת לנו לכתוב:

$$\mathbb{P}[X_0 = i_0, X_1 = i_1, \dots, X_n = i_n] \overset{\text{א}}{=} \mathbb{P}[X_0 = i_0] \cdot \mathbb{P}[X_1 = i_1, \dots, X_n = i_n \mid X_0 = i_0] \overset{\text{א}}{=}$$

$$= \mathbb{P}[X_0 = i_0] \cdot \mathbb{P}[X_1 = i_1 \mid X_0 = i_0] \cdot \mathbb{P}[X_2 = i_2, \dots, X_n = i_n \mid X_0 = i_0, X_1 = i_1] \overset{\text{א}}{=}$$

$$= \mathbb{P}[X_0 = i_0] \cdot \mathbb{P}[X_1 = i_1 \mid X_0 = i_0] \cdot \underline{\mathbb{P}[X_2 = i_2 \mid X_0 = i_0, X_1 = i_1]} \cdot \mathbb{P}[X_3 = i_3, \dots, X_n = i_n \mid X_0 = i_0, X_1 = i_1, X_2 = i_2] \overset{\text{ב}}{=}$$

$$= \mathbb{P}[X_0 = i_0] \cdot \mathbb{P}[X_1 = i_1 \mid X_0 = i_0] \cdot \underline{\mathbb{P}[X_2 = i_2 \mid X_1 = i_1]} \cdot \mathbb{P}[X_3 = i_3, \dots, X_n = i_n \mid X_0 = i_0, X_1 = i_1, X_2 = i_2] = \cdots$$

א. הגדרת הסתברות משותפת.
ב. החלקים המסומנים שווים, בגלל תכונת מרקוב.

נוכל להמשיך את אותו התהליך, ובסוף נקבל:

$$\cdots = \mathbb{P}[X_0 = i_0] \cdot \mathbb{P}[X_1 = i_1 \mid X_0 = i_0] \cdot \ldots \cdot \mathbb{P}[X_n = i_n \mid X_{n-1} = i_{n-1}]$$

מטריצת מעברים של שרשרת מרקוב

שרשרת מרקוב תיקרא **הומוגנית בזמן** (*time homogeneous*) אם הסתברויות המעבר לא תלויות בזמן.

כלומר, לא אכפת לי **מתי** אני נמצא במצב כלשהו. כל פעם שאני נמצא במצב א, יש לי הסתברות מסויימת לעבור למצב ב.

שרשראות כאלה מוגדרות לחלוטין ע"י **מטריצת המעברים**, שמוגדרת:

$$P_{ij} := \mathbb{P}[X_t = j \mid X_{t-1} = i]$$

כלומר המיקום ה-ij זה ההסתברות לעבור למצב j ממצב i.

מטריצה כזו תהיה סופית אם מרחב המצבים סופי. אחרת היא אינסופית.

נניח שמרחב המצבים הוא $[n]$. אזי, סכום כל שורה במטריצה יהיה 1:

$$\forall i \in [n], \qquad \Sigma_{j=1}^{n} P_{ij} = 1$$

$$P = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/3 & 1/3 & 0 & 1/3 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 0 & 1/2 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

כי סכום ההסתברויות של המעברים חייב להיות 1.

נשים לב שהאלכסון הראשי הוא "מעבר" ממצב אחד לאותו מצב. זה גם תקין.

P היא **מטריצה סטוכסטית** – מטריצה שמתארת תהליך סטוכסטי.

שימוש במטריצת המעברים

יהי $[n]$ מרחב המצבים של שרשרת הומוגנית. עבור זמן t ומצב $i \in [n]$, נגדיר: $P_i(t)$ את ההסתברות שבזמן t השרשרת במצב i.

אז וקטור השורה מייצג את ההתפלגות של המצבים בזמן t:

$$P(t) := \left(P_1(t), \dots, P_n(t)\right)$$

לפי נוסחת ההסתברות השלמה, $P_i(t)$ הוא:

$$P_i(t)=\sum_{j=1}^{n}\mathbb{P}[X_{t-1}=j]\cdot\mathbb{P}[X_t=i\mid X_{t-1}=j]=\sum_{j=1}^{n}P_j(t-1)\cdot P_{ij}$$

כלומר זה הסכום של ההכפלות של המקומות של הוקטור שורה במקומות במטריצה. הכפלה של וקטור במטריצה. כלומר בעצם:

$$\underbrace{P(t)}_{row}=\underbrace{P(t-1)}_{row}\cdot\underbrace{P}_{matrix}$$

נשים לב: ההתפלגות של השרשרת בזמן t כן יכולה להיות פונקציה של הזמן. רק המעברים לא יכולים להיות.

מעברי m-צעדים

בהינתן $m\geq 1$, ההסתברות:

$$P_{ij}^{(m)}\coloneqq\mathbb{P}[X_{t+m}=j\mid X_t=i]$$

נקראת הסתברות מעבר m-צעדים (*m-step transition probability*). (ה-(m) זה לא חזקה, זה רק סימון).

אז מחוק ההסתברות השלמה ותכונת מרקוב נקבל:

$$P_{ij}^{(m)}=\sum_{k=1}^{n}\mathbb{P}[X_{t+1}=k\mid X_t=i]\cdot\mathbb{P}[X_{t+m}=j\mid X_{t+1}=k,X_t=i]=$$
$$=\sum_{k=1}^{n}\mathbb{P}[X_{t+1}=k\mid X_t=i]\cdot\mathbb{P}[X_{t+m}=j\mid X_{t+1}=k]=$$
$$=\sum_{k=1}^{n}P_{ik}\cdot P_{kj}^{(m-1)}$$

אז אם נגדיר:

$$P^{(m)}\coloneqq\left(P_{ij}^{(m)}\right)_{i,j\in[n]}$$

אפשר להוכיח באינדוקציה שמתקיים:

$$\forall m\geq 0,\qquad P^{(m)}=P^m$$

אז כמו שתיארנו את המעבר בצעד אחד בזמן: $P(t)=P(t-1)\cdot P$, נוכל להגדיר מעבר של m צעדים:

$$P(t+m)=P(t)\cdot P^m$$

ההתפלגות בזמן $t+m$ היא פשוט ההתפלגות בזמן t כפול המטריצה P^m.

באופן כללי: אם $\mu_0\coloneqq\left(\mu_0(i)\right)_{i\in[n]}$ היא ההתפלגות ההתחלתית, ומתקיים:

$$\mu_n\coloneqq(\mu_n(i)\coloneqq\mathbb{P}[X_n=i\mid X_0\sim\mu_0])_{i\in[n]}$$

אז:

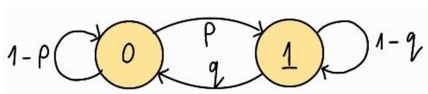
$$\mu_n=\mu_0\cdot P^n$$

דוגמאות לשרשראות מרקוב הומוגניות

התגעגעתם לאוטומטים?

<u>דוגמה 1</u>

$$P=\begin{bmatrix}1-p & p\\ q & 1-q\end{bmatrix}$$



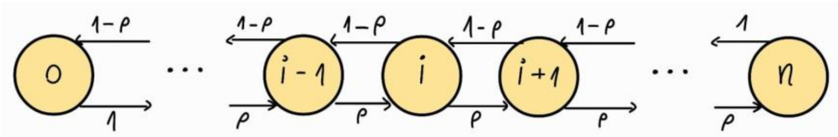
<u>דוגמה 2</u> – הילוך שיכור עם גבולות מחזירים $(reflecting\ boundaries)$

הילוך שיכור על מצבים $\{0,1,\dots,n\}$. בכל זמן עוברים ימינה בהסתברות p, ושמאלה בהסתברות $1-p$.

בגבולות $\{0,n\}$ תמיד עוברים למצב הסמוך בהסתברות 1. מטריצת המעברים P נתונה ע"י:

$$\forall 0<i<n, \qquad P_{i,i+1}=p, \qquad P_{i,i-1}=1-p, \qquad P_{0,1}=1, \qquad P_{n,n-1}=1$$

בכל שאר המצבים, $P_{ij}=0$.



אפשר להכליל את ההגדרה:

לכל $1\le i\le n$ נבחר שלושה מספרים a_i,b_i,c_i כך ש: $a_i+b_i+c_i=1$.

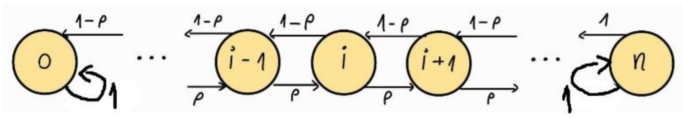
זה ההסתברויות ללכת ימינה, להישאר במקום, וללכת שמאלה.

מטריצת המעברים המתארת הילוך "חוזר" תוגדר:

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & \cdots & 0 & 0 & 0 \\ 0 & a_3 & b_3 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$$

<u>דוגמה 3</u> – הילוך שיכור עם גבולות סופגים $(absorbing\ boundaries)$

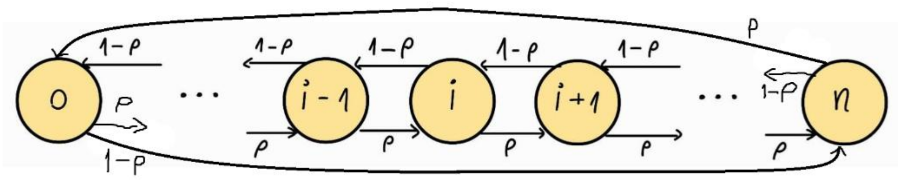
ההבדל היחיד הוא הגבולות. אם הגענו לגבול, נישאר בו:



$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & \cdots & 0 & 0 & 0 \\ 0 & a_3 & b_3 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}$$

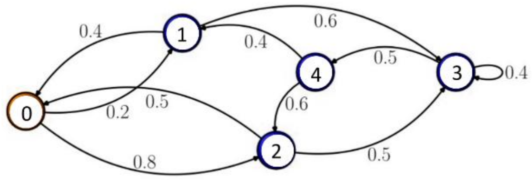
<u>דוגמה 4</u> – הילוך שיכור מעגלי $(cyclic)$

כל קצה מחובר גם לקצה השני:



$$\begin{bmatrix} b_1 & c_1 & 0 & \cdots & 0 & 0 & a_1 \\ a_2 & b_2 & c_2 & \cdots & 0 & 0 & 0 \\ 0 & a_3 & b_3 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1} & b_{n-1} & c_{n-1} \\ c_n & 0 & 0 & \cdots & 0 & a_n & b_n \end{bmatrix}$$

דוגמה 5 – סתם עוד דוגמה



$$P = \begin{bmatrix} 0 & 0.2 & 0.8 & 0 & 0 \\ 0.4 & 0 & 0 & 0.6 & 0 \\ 0.5 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.4 & 0.6 \\ 0 & 0.4 & 0.6 & 0 & 0 \end{bmatrix}$$

אלגוריתם אקראי ל- *2-CNF-SAT*

הבעיה נמצאת ב-P. ראינו אלגוריתם דטרמיניסטי. נרצה עכשיו לתאר אלגוריתם אקראי פולינומי:

קלט: נוסחת *2-CNF* φ עם n משתנים, ופרמטר m. בה"כ, בכל פסוקית יש 2 משתנים שונים.

1. תהי a השמה אקראית למשתנים של φ. נאתחל $i \coloneqq 1$.
2. כל עוד $i \leq 2mn^2$, נבצע:
   a. נבחר פסוקית אקראית C, שלא מסופקת תחת a.
   b. נבחר באופן מקרי ואחיד משתנה של C ונחליף את הערך שהוא מקבל.
   c. אם ההשמה המתקבלת מספקת, **נחזיר** ש-φ ספיקה עם ההשמה הנוכחית. ונסיים.
   d. $i \coloneqq i + 1$.
3. נחזיר ש-φ לא ספיקה.

התוצאה העיקרית שלנו, נקרא לה **למה א:**

תהי φ נוסחת *2-CNF*, ויהי $m \in \mathbb{N}$. נריץ את האלגוריתם לעיל עם φ, m. אזי:

1) אם φ לא ספיקה, אז בהסתברות 1 האלגוריתם מחזיר תשובה נכונה.
2) אם φ כן ספיקה, אז בהסתברות לפחות $1 - 2^{-m}$ האלגוריתם מוצא השמה מספקת.

הוכחה: (1) טריוויאלי – אף השמה לא תהיה מספקת, אז נחזיר שלא אחרי $2mn^2$ איטרציות.

כדי להוכיח את (2), ניעזר ב**למה ב:**

תהי φ נוסחת *2-CNF* ספיקה עם n משתנים.

אזי, מספר האיטרציות שנצטרך עד שנמצא השמה מספקת (בתוחלת) הוא לכל היותר n^2.

(נתייחס כאילו מריצים את האלגוריתם עם $m = \infty$, כך שהוא לא יעצור עד שנמצא השמה מספקת).

נניח בנתיים שלמה ב' נכונה, ונוכיח את למה א'. ואז בסוף נוכיח את למה ב' ונסיים.

נחלק את האיטרציות לקבוצות (*batches*) של $2n^2$ איטרציות כל אחת.

לפי למה ב, לא משנה מה הייתה ההשמה הראשונית, נמצא השמה מספקת תוך n^2 איטרציות (בתוחלת).

נסמן עבור קבוצה נתונה את מספר האיטרציות הנדרש כדי למצוא השמה מספקת – X. אזי, לפי אי"ש מרקוב נקבל:

$$\mathbb{P}[X > 2n^2] \leq \frac{\mathbb{E}[X]}{2n^2} = \frac{1}{2}$$

אז ההסתברות ש-m קבוצות נכשלות הוא לכל היותר 2^{-m}, כנדרש.

הוכחת למה ב'

תהי S השמה מספקת כלשהי עבור φ. נמצא חסם עליון על מספר האיטרציות שצריך כדי למצוא דווקא את S.

עבור איטרציה i, נסמן:

A_i – ההשמה בסוף האיטרציה. X_i – מספר המשתנים שבהם S ו-A_i **מסכימים**.

האלגוריתם בוודאי עוצר כאשר $X_i = n$, כי זה אומר ש S ו-A_i מסכימים על כל המשתנים, כלומר $A_i = S$.

הוא יכול לעצור לפני אם במקרה נגיע להשמה מספקת (יכולה להיות יותר מאחת).

נרצה להעריך את הזמן המצופה עד ש $X_i = n$.

בניית תהליך סטוכסטי (לא מרקוב)

אם $X_i = 0$ ב-i כלשהו, אזי $X_{i+1} = 1$.

כי אם בשלב i לא מסכימים באף משתנה (ולפי האלגוריתם מחליפים את ההשמה של משתנה אחד אקראי), אז בשלב $i+1$ תהיה הסכמה במשתנה הזה.

$$\mathbb{P}[X_{i+1} = 1 \mid X_i = 0] = 1$$

נסמן C את הפסוקית הלא-מסופקת שהאלגוריתם בוחר באיטרציה ה-i.

מכיוון ש-S מספקת ו-C לא מסופקת תחת A_i, מתקיים ש- S ו-A_i לא מסכימות בלפחות משתנה אחד של C.

אם S ו-A_i לא מסכימות בשני משתנים של C, אז כל שינוי יגרום ליותר הסכמה. בסה"כ, לכל $1 \leq j \leq n-1$:

$$\mathbb{P}[X_{i+1} = j+1 \mid X_i = j] \geq 0.5, \qquad \mathbb{P}[X_{i+1} = j-1 \mid X_i = j] \leq 0.5$$

הבעיה שלנו היא שהתהליך המתואר לא מרקובי. ההסתברות ש-X_i יגדל תלויה במספר המשתנים ב-C, וכל משתנה יכול להופיע ביותר מפסוקית אחת.

בניית תהליך מרקובי

אז נייצר גרסה יותר "פסימית" של התהליך, שכן תהיה מרקובית:

$$Y_0 \coloneqq X_0, \qquad \mathbb{P}[Y_{i+1} = 1 \mid Y_i = 0] = 1, \qquad \mathbb{P}[Y_{i+1} = j+1 \mid Y_i = j] = 0.5, \qquad \mathbb{P}[Y_{i+1} = j-1 \mid Y_i = j] = 0.5$$

תוחלת הזמן שייקח ל-X להגיע ל-n, היא לכל היותר הזמן שייקח ל-Y.

עבור $j \in \{0, \dots, n\}$ נגדיר את H_j להיות מספר הצעדים הנדרש ל- $Y_0, Y_1, \dots$ להגיע ל-n אם מתחילים מ-j.

אנחנו רוצים להראות ש: $\mathbb{E}[H_0] \leq n^2$. מספיק להוכיח ש:

$$(*) \quad \forall j \geq 0, \qquad \mathbb{E}[H_j] = \mathbb{E}[H_{j+1}] + 2j + 1$$

למה? כי אם נעשה את זה, אז:

$$\mathbb{E}[H_0] = \mathbb{E}[H_1] + 1 = (\mathbb{E}[H_2] + 2 \cdot 1 + 1) + 1 = \cdots = \sum_{i=0}^{n-1} (2i+1) = n^2$$

כדי להוכיח את $(*)$, נוכיח את מערכת המשוואות $(**)$:

$$(**) \quad \mathbb{E}[H_n] = 0, \qquad \mathbb{E}[H_j] = \frac{\mathbb{E}[H_{j-1}]}{2} + \frac{\mathbb{E}[H_{j+1}]}{2} + 1 \;\; j \notin \{0, n\}, \qquad \mathbb{E}[H_0] = \mathbb{E}[H_1] + 1$$

ואז באינדוקציה נוכיח את $(*)$.

הוכחת $(**)$

טריוויאלי ש $\mathbb{E}[H_n] = 0$. ולפי הגדרה, $\mathbb{E}[H_0] = \mathbb{E}[H_{0+1}] + 2 \cdot 0 + 1 = \mathbb{E}[H_1] + 1$.

נקבע $0 < j < n$ כלשהו. לפי חוק התוחלת השלמה:

$$\mathbb{E}[H_j] = \underbrace{\mathbb{P}[H_j = H_{j-1} + 1]}_{0.5} \cdot \mathbb{E}[H_j \mid H_j = H_{j-1} + 1] + \underbrace{\mathbb{P}[H_j = H_{j+1} + 1]}_{0.5} \cdot \mathbb{E}[H_j \mid H_j = H_{j+1} + 1]$$

$$= \mathbb{E}\left[\frac{1}{2}(H_{j-1} + 1) + \frac{1}{2}(H_{j+1} + 1)\right] = \frac{\mathbb{E}[H_{j-1}]}{2} + \frac{\mathbb{E}[H_{j+1}]}{2}$$

נוכיח ש $(**) \Longleftarrow (*)$

<u>בסיס</u>: עבור $j = 0$, כבר הראינו ש $\mathbb{E}[H_0] = \mathbb{E}[H_1] + 1$.

<u>צעד</u>: לפי $(**)$, נעביר אגפים ומתקיים:

$$\mathbb{E}[H_{j+1}] = 2\mathbb{E}[H_j] - \mathbb{E}[H_{j-1}] - 2$$

לפי הנ"א:

$$\mathbb{E}[H_{j-1}] = \mathbb{E}[H_j] + 2(j-1) + 1$$

כלומר נוכל לכתוב:

$$\mathbb{E}[H_{j+1}] = 2\mathbb{E}[H_j] - \left(\mathbb{E}[H_j] + 2(j-1) + 1\right) - 2 = 2\mathbb{E}[H_j] - \mathbb{E}[H_j] - 2(j-1) - 1 - 2 = \mathbb{E}[H_j] - 2j - 1$$

כלומר:

$$\mathbb{E}[H_j] = \mathbb{E}[H_{j+1}] + 2j + 1$$

כנדרש.

סיכום

האלגוריתם רץ עד $2mn^2$ פעמים. בכל פעם בוחר פסוקית לא מסופקת ומחליף את ההשמה של אחד המשתנים שלה.

תיארנו את מספר הצעדים הנדרש כדי להגיע להשמה מספקת ספציפית S, כתלות במספר הצעדים שהיה נדרש בשלב הקודם. והראנו ש $\mathbb{E}[H_0] = n^2$.

חילקנו את התהליך לקבוצות של $2n^2$ איטרציות. לפי אי"ש מרקוב, ההסתברות שבקבוצה אחת לא נמצא השמה מספקת היא לכל היותר חצי.

אז עבור m קבוצות כאלה, ההסתברות שלא נמצא השמה מספקת היא לכל היותר 2^{-m}.