# מחלקות של שפות:

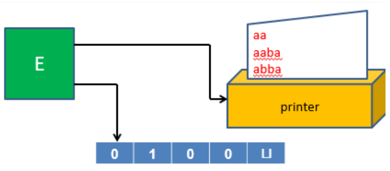
**1.16 + 1.40+ 1.54– שפה רגולרית:** אם אוטומט סופי (אסד או אסלד)/ביטוי רגולרי מזהה אותה (זה אמם)
**סגירות שפות רגולריות לפעולות:** חיתוך, איחוד (משפט 1.25), שרשור (משפט 1.26), משלים, היפוך, כוכב (הגדרה ב1.23 – שרשור של מילים בשפה, הוכחה ב1.49)
**2.20 – שפה חסרת הקשר** – אם אוטומט מחסנית מזהה אותה (אמם) לדוגמה: $\{a^nb^n | n>=0\}$
2.32 – כל שפה רגולרית היא חסרת הקשר
**סגירות שפות חסרות הקשר לפעולות:** היפוך, חיתוך עם שפה רגו'(אוטומט מכפלה), איחוד, שרשור, כוכב (**לא** למשלים וחיתוך)
**1.39** – כל אסלד אפשר להפוך לאסד
**למת הניפוח**: אם לאוטומט סופי יש $k$ מצבים, ובשפה שהוא מקבל יש מילה $x$ כך ש: $|x|\geq k$ הרי שיש חזרה למצב שכבר היינו בו, וחזרה זו ניתנת לשיכפול אינסוף פעמים – ולכן – יש בשפה אינסוף מילים.
כלומר – שפה היא סופית אםם כל המילים בשפה הן מאורך קטן מ $k$, כש – $k$ מספר המצבים באוטומט.
**Regular ⊆ Context free ⊆ Decidable ⊆ Turing recogniable (4.10)**

# תזת צ'רץ' טיורינג:

**מכונת טיורינג M:** $M=(Q,\Sigma,\Gamma,\delta,q_0,\ q_{accept},\ q_{reject})$ כאשר $\delta: Q\times\Gamma\longrightarrow Q\times\Gamma\times\{L,R\}$
**תיאור מילולי של מ"ט:** M="על הקלט w... אם ... קבל, אחרת דחה" (דוג' בעמ 174)
**תיאור פורמלי / גרפי של מ"ט:** חייב להופיע - $q_0, q_{accept}, q_{reject}$ , מעברים: לדוגמא $0\rightarrow\sqcup,R$ (עמ 172)
**קונפיגורציות:** (**uqv** - מתאר קונפיגורציה בה המצב הנוכחי הוא q, הסרט מכיל מחרוזת uv והראש נמצא על התו הראשון של v).
לדוגמא: $011q_80111$
מ"ט M **מקבלת** קלט w אם קיים רצף קונפיגורציות המקיים :
1. קונפיגורציה $C_1$ קונפיגורציה התחלתית של M על w.
2. קונפיגורציה $C_i$ מניבה (yields) קונפיגורציה $C_{i+1}$ לכל $1\le i<k$
3. קונפיגורציה $C_k$ היא קונפיגורציה מקבלת.

**השפה הניתנת לזיהוי ע"י M:** אוסף המחרוזות שמכונת טיורינג M מקבלת היא השפה של M, או השפה הניתנת לזיהוי ע"י M ומסמנים אותה L(M)
**3.5 - שפה מזהה טיורינג** : שפה היא ניתנת לזיהוי ע"י מכונת טיורינג או שפה מזהה טיורינג (turing recognizable) אם קיימת מכונת טיורינג המזהה אותה (כלומר שאוסף המחרוזות שמכונת הטיורינג מקבלת היא השפה).
- המכונה מקבלת כל מילה ששייכת ל- L **ולא מקבלת** (דוחה/נתקעת) כל מילה שלא שייכת ל- L
**מכונת טיורינג מכריעה (decide) שפה מסויימת:** אם היא מ"ט כריעה שגם מזהה את השפה (כלומר תמיד עוצרת על כן או לא לכל קלט)
**3.6 - שפה כריעה**: שפה "כריעה טיורינג" או כריעה (decidable) אם קיימת מכונת טיורינג המכריעה אותה.
- המכונה מקבלת כל מילה ששייכת ל- L דוחה כל מילה שלא שייכת ל- L. אין אפשרות לריצה ללא עצירה

**גרסאות של מכונת טיורינג:**
**מ"ט בעלת מס' סרטים:** $M=(Q,\Sigma,\Gamma,\delta,q_0,\ q_{accept},\ q_{reject})$ כאשר $\delta: Q\times\Gamma^k\longrightarrow Q\times\Gamma^k\times\{L,R,S\}^k$
**3.13** – מ"ט **בעלת מס' סרטים** ניתנת להדמיה ע"י מ"ט עם סרט אחד
**3.15** שפה L היא **מזהה טיורינג** אם"ם קיימת מ"ט בעלת סרטים מרובים המזהה את השפה.
**מ"ט לא דטרמניסטית:** $N=(Q,\Sigma,\Gamma,\delta,q_0,\ q_{accept},\ q_{reject})$ כאשר $\delta: Q\times\Gamma\longrightarrow\mathcal{P}(Q\times\Gamma\times\{L,R\})$
**3.16** לכל **מ"ט לא דטרמניסטית** יש מ"ט דטרמיניסטית שקולה לה.
**3.18** שפה L היא **מזהה טיורינג** אם"ם קיימת מ"ט ל"ד המזהה אותה
**3.19** שפה היא **כריעה** אם"ם קיימת מ"ט ל"ד שמכריעה אותה (???אמ"מ עץ הקונפיגרוציות המתקבל במכונה זו הוא סופי??)
**מונה:** $E=(Q,\Sigma,\Gamma,\delta,q_0,\ q_{print},\ q_{halt})$ כאשר $\delta: Q\times\Gamma\longrightarrow Q\times\Gamma\times\{L,R\}\times(\Sigma\cup\{\varepsilon\})$




מבנה מונה: סרט עבודה + סרט הדפסה **לכתיבה בלבד** אין מצבים דוחים ומקבלים אלא מצב הדפסה – $q_{print}$ ומצב עצירה – $q_{halt}$ שמציין שסיימנו
**3.21** שפה היא **מזהה טיורינג** אם"ם קיים **מונה** שמונה(מדפיס) אותו
משפט – שפה L היא **כריעה** אמ"מ קיים **מונה** שמדפיס אותה לפי **הסדר הסטנדרטי**
(בהוכחה יש להבחין בין L סופית לבין L אינסופית באחד הכיוונים)

**מאמת:** מ"ט שמקבלת מילת קלט **w** + אימות שייכות של w לשפה = **c**. כלומר **הקלט** הוא: w#c
המאמת בודק האם האימות c באמת מוכיח שייכות של w לשפה.
לדוגמא: עבור מאמת V למספרים פריקים (לא ראשוניים) c יכול להיות 2 מספרים ו-V יכפול אותם ויבדוק האם מכפלתם היא w.
אם כן – **יקבל**, אחרת – **ידחה**.
**קבלה** פירושה ש w שייכת לשפה ואילו **דחייה** אומרת רק ש c לא מוכיח ש w שייכת לשפה.
משפט – לשפה L יש מאמת ⟺ L מזהה טיורינג (הוכחה – מ"ט לא דטרמינסטית N, c יהיה מסלול בעץ החישוב על N)

צביקה פרלמוטר, 2017 ©

# כריעות:

**סגירות בשפות ניתנות לזיהוי והכרעה:**

משפטים:

- כל שפה **רגולרית** היא **כריעה** – יש להראות שאפשר ליצור מאס"ד מכונת טיורינג מכריעה
- כל שפה **חסרת הקשר** היא **כריעה** - לכל שפה חסרת הקשר יש דקדוק בצורת הנורמלית של חומסקי
- כל שפה **כריעה** היא **מזוהה טיורינג** – נובע מיידית מההגדרה

מחלקת השפות **הכריעות** סגורה ל: איחוד, חיתוך, **משלים**, שרשור, איטרציה (הוכחה – בחלק מומלץ ל"ד)
מחלקת השפות **מזוהות טיורינג** סגורה ל: איחוד, חיתוך, שרשור, איטרציה (הוכחה – בחלק מומלץ ל"ד)

**4.22** L **כריעה** ⟸ L **מזוהה טיורינג** וגם $\bar{L}$ **מזוהה טיורינג**

**דוגמאות לשפות:**

א"ד

**$A_{DFA}$ – כריעה**: "על הקלט <A,w> : חקה ריצה של A על w, אם קיבלה קלה אם דחתה דחה." (4.1)
**$A_{NFA}$ – כריעה**. מעבר מאס"ד לאס"ד (אוטומט חזקה) ואז כמו $A_{DFA}$. (4.2)
**$A_{REX}$ – כריעה** . מעבר מביטוי רגולרי לאס"ד ע"י משפט קלין (4.3)
**$E_{DFA}$ –כריעה**. "על הקלט <A>, A אס"ד: סמן את המצב התחלתי של A, עבור כל המצבים היוצאים ממצבים מסומנים סמן אתם (חזור עד שלא מסומנים יותר). בדוק האם הגעת למצב מקבל – אם כן קבל אחרת דחה" (4.4)
**$EQ_{DFA}$-כריעה**. הרעיון – שימוש באוטומט להפרש סימטרי של L(A) ו L(B) (4.5)

דקדוק ח"ה

**$A_{CFG}$ – כריעה**. הרעיון – נעבור לצורה הנורמלית של חומסקי ואז צעדי הגזירה ידועים: אם w≠ε אז מס' צעדי הגזירה הוא 2|w|-1 ואם w=ε אז מספר צעדי הגזירה הוא 1. לכן נבנה את כל הגזירות. אם יש גזירה שיוצרת את w – נקבל אחרת נדחה.
**$E_{CFG}$ – כריעה**. "על הקלט <G>, סמן את כל הטרמינלים של G, חזור עד שלא סומן משתנה חדש : סמן את כל משתנה A כל שיש ב G כלל שכתוב $A\text{->}X_1X_2...X_n$ וכל $X_i$ כבר מסומן. בדוק האם המשתנה ההתחלתי מסומן – לא, קבל. כן, דחה.
**$EQ_{CFG}$ – לא כריעה**

מ"ט

**$A_{TM}$ – מזוהה**: נחכה ריצתה ע"י מ"ט אוניברסלית U. נשים לב שמ"ט זו מזהה את M אך לא מכריעה אותה.
משפט: **$A_{TM}$ אינה כריעה!**

**הוכחה:** נניח בשלילה ש $A_{TM}$ כריעה 🡸 יש לה מכונה H מכריעה: $H(<M,w>) = \begin{cases} \textbf{accept} \text{ if } M \text{ accepts } w \\ \textbf{reject} \text{ if } M \text{ dosent accept } w \end{cases}$

כעת נבנה מכונה D בעזרת H: $D(<M>) = \begin{cases} \textbf{accept} \text{ if } M \text{ dosent accept } <M> \\ \textbf{reject} \text{ if } M \text{ accepts } <M> \end{cases}$

- כלומר אם H מקבלת <M,<M>> אז D דוחה את <M> ואם H דוחה את <M.<M>> אז D מקבלת

כעת נבדוק מה יקרה אם נריץ את D על התיאור של עצמי: $D(<D>) = \begin{cases} \textbf{accept} \text{ if } D \text{ dosent accept } <D> \\ \textbf{reject} \text{ if } D \text{ accepts } <D> \end{cases}$

**קיבלנו סתירה! מסקנה: $A_{TM}$ לא כריע**

| | $\langle M_1\rangle$ | $\langle M_2\rangle$ | $\langle M_3\rangle$ | ... | $\langle D\rangle$ |
|---|---|---|---|---|---|
| $M_1$ | $\boxed{\text{accept}}$ | reject | accept | | |
| $M_2$ | accept | $\boxed{\text{accept}}$ | accept | | |
| $M_3$ | reject | reject | $\boxed{\text{reject}}$ | | |
| $M_4$ | accept | accept | reject | | |
| ⋮ | | ⋮ | | ⋱ | |
| $D$ | reject | reject | accept | | $\boxed{???}$ |
| ⋮ | | ⋮ | | ... | |

**הוכחה בשיטת האלכסון:** D הופכת את הערכים באלכסון ובכך מבטיחה להיות שונה מכל מכונה קיימת (היא שונה מ $M_k$ במה שהיא מחזירה על $<M_k>$). אבל אין מכונה ששונה מכל המכונות ולכן לא קיימת D כזו.

מסקנה ממשפט 4.22 - $\overline{A_{TM}}$ **איננה מזוהה טיורינג.**

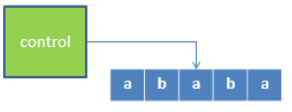
**$HALT_{TM}$** – בעיית העצירה, שפת כל הקלטים <M,w> שמכונת טיורינג עוצרת עליהם – מזוהה טיורינג ואיננה כריעה (הוכחה בשיטת האלכסון / ברדוקציה מ $A_{TM}$)

# רדוקציות:

**<u>רדוקציות:</u>** שיטה אלגוריתמית להעברת בעיה נתונה A לבעיה אחרת B שבעזרתה ניתן לפתור את הבעיה המקורית.
**<u>מסקנה:</u>** אם יש **רדוקציה מ- A ל- B**, ו- B כריעה ← A כריעה ("בעיה A אינה קשה יותר מבעיה B ").
אם הוכחנו ש A איננה כריעה, והצלחנו למצוא רדוקציה מ- A ל- B ← אנחנו יכולים להסיק ש B איננה כריעה

**שיטה א' – רדוקציית טיורינג**
<u>דוגמא</u>: **רדוקציה מ $HALT_{TM}$ ל $A_{TM}$** : נראה כי ניתן בהינתן מ"ט R המכריעה את $HALT_{TM}$ לבנות מ"ט S המכריעה את $A_{TM}$ :
S="על הקלט <M,w> כאשר M מ"ט ו w מחרוזת: הרץ את מ"ט R על הקלט <M,w> אם דחתה, דחה. אם קיבלה – בצע סימולציה על w עד לעצירה. אם קיבלה, קבל. אם דחתה, דחה".
**רדוקציה מ $A_{TM}$ ל $HALT_{TM}$**: R="על הקלט <M,w> כאשר M מ"ט ו w מחרוזת: בנה מ"ט K הזהה למ"ט M פרט לכך ש $q_{reject}$ וה $q_{accept}$ יתחלפו. הרץ את S על <K,w>. אם קיבלה – קבל, אם דחתה – דחה".

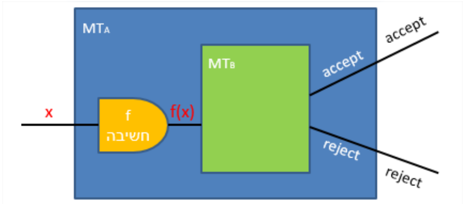
**שיטה ב' – היסטוריית חישוב**
**<u>היסטוריה חישובית:</u>** היסטוריית חישוב של מכונת טיורינג M על קלט w היא רשימת קונפיגורציות ש – M עוברת בזמן העיבוד של w עד שהיא עוצרת ומקבלת את w או דוחה אותו. אם M איננה עוצרת על w, אז היסטוריית החישוב של M על w היא אינסופית.
**<u>5.5</u>** – **<u>היסטוריה חישובית מקבלת:</u>** תהי M מ"ט ו- w מחרוזת קלט. הסטוריה חישובית מקבלת היא רצף קונפיגורציות : $C_1, C_2,\ldots ,C_m$ כך ש $C_1$ היא קונפיגורציה התחלתית של M על w, $C_m$ היא קונפיגורציה מקבלת של M, וכל $C_i$ היא קונפיגורציה עוקבת הנגזרת מהקונפיגורציה $C_{i-1}$ לפי החוקים של M.
**<u>הסטוריה חישובית דוחה</u>** של M על w מוגדרת באופן דומה, אלא ש $C_m$: היא קונפיגורציה דוחה.

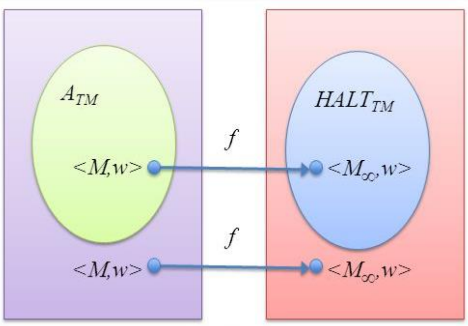



**<u>5.6</u>** – **<u>אוטומט חסום ליניארית LBA:</u>** אוטומט חסום ליניארית הוא מ"ט מוגבלת, בה אסור לראש לנוע על הסרט מעבר לחלק בו מופיע הקלט. אם המכונה מנסה לגרום לראש לנוע מעבר לאחד הקצוות של הקלט – הראש נותר במקום שבו היה (חסרונות – הגדלת א"ב הסרט, הגדלה של מס' המצבים החדש). **כל שפה חופשית הקשר כריעה ע"י LBA**
**<u>למה 5.8</u>** אם M היא LBA בעלת q מצבים ו – g סמלים בא"ב הסרט – יש בדיוק $qng^n$ קונפיגורציות שונות של M, לסרט באורך n.
$A_{LBA}$ כריעה ואילו $E_{LBA}$ אינה כריעה

**<u>משפט Rice</u>**
תהי A קבוצת התוכניות (=שפת קידודי מ"ט) המקיימת:
1. לכל שתי תוכניות P,Q המקיימות L(P)=L(Q), בהכרח מתקיים $P \in A \Leftrightarrow Q \in A$ – כלומר מדובר בתכונה של השפה ולא של מבנה מ"ט (התכנית). אם יש 2 מ"ט מזהות אז שתיהן שייכות / לא שייכות ל-A ביחד.
2. קיימת תכנית $P \in A$ וכמו כן קיימת תוכנית $Q \notin A$ - כלומר התכונה לא טריוואלית. לא אף/כל מ"ט מקיימת .
אזי **A איננה כריעה.**

דוגמא לתכונה של מבנה ולא של השפה (סותר תנאי 1): A={<M>| <M> דוחה את הקלט M}
דוגמא לטריוואלית (סותר תנאי 2): B={<M>|השפה) L(M) ניתנת לזיהוי} - כי מכילה כל <M>
דוגמאות לשפות שמקיימות תנאי רייס:
- A={<M>| 3 מקבלת את הקלט M}
- B={<M>| השפה) L(M) היא ח"ה}
- C= {<M>| x! מחשבת את הפונקציה M}
- D={<M> |L(M)|=7}




**שיטה ג' – רדוקציית מיפוי**
**<u>פונקציה חשיבה:</u>** פונקציה $f: \Sigma^* \to \Sigma^*$ היא פונקציה חשיבה אם יש מ"ט M, כך שלכל קלט w, כאשר M עוצרת, על הסרט שלה מופיע f(w)
**<u>רדוקציית מיפוי:</u>** שפה A ניתנת ל"רדוקציית מיפוי" לשפה B (כותבים זאת) :
($A \le_M B$) אם קיימת פונקציה חשיבה כך שלכל w מתקיים : $w \in A \Leftrightarrow f(w) \in B$ . ל- f נקרא רדוקציה מ- A ל- B.
**<u>5.22</u>** אם **קיימת רדוקציית מיפוי** מ- A ל- B ($A \le_M B$) ו – B היא **שפה כריעה** – אזי גם **A שפה כריעה**
**<u>5.23</u>** אם קיימת רדוקציית מיפוי מ- A ל- B ו – A היא **שפה שאיננה כריעה** – אזי גם B שפה שאיננה כריעה

$$A \le_M B \Leftrightarrow \bar{A} \le_M \bar{B}$$

*משפטים 5.22 ו 5.23 נכונים גם אם A ו B **מזהות טיורינג**

**דוגמאות לרדוקציות מיפוי**

| **נסמן** $L_5^+ = \{< M > \| \|L(M)\| \geq 5\}$<br>**הראו ש** $A_{TM} \leq_m L_{5^+}$ | **נסמן** $L_5 = \{< M > \| \|L(M)\| = 5\}$<br>**הראו** $\overline{A_{TM}} \leq_m L_5$ |
|---|---|
| נראה ע"י מ"ט F שמחשבת את הרדוקציה f:<br><br>F="על הקלט <M,w> כאשר M מ"ט ו-w מחרוזת:<br>בנה את מ"ט M' הבאה:<br>M' = "על הקלט x: הרץ (בעזרת מ"ט U) את M על w. אם קיבלה, קבל. אם דחתה, דחה."<br>החזר את <M'>."<br><br>מתקיים: <M,w>∈ATM ↔ <M'>∈L5+.<br>**מסקנה -** L5+ אינה כריעה. | מ"ט F הבאה מחשבת את הרדוקציה f:<br><br>F= "על הקלט <M,w> כאשר M מ"ט ו-w מחרוזת:<br>1. בנה את מ"ט M' הבאה:<br>M' = "על הקלט x:<br>1. אם x∌{1,2,3,4,5}, קבל.<br>2. הרץ (בעזרת מ"ט U) את M על w.<br>אם קיבלה, קבל. אם דחתה, בצע לולאה אינסופית."<br>2. החזר את <M'>."<br>מתקיים <M,w>∈ATM ↔ <M'>∈L5 |
| מצד שני נשים לב ש $L_{5^+}$ **ניתנת לזיהוי**:<br>נציע מ"ט S שמזהה אותה.<br>S="על הקלט <M> כאשר M מ"ט:<br>1. נחש (באופן לא דטרמיניסטי) 5 קלטים: x1...x5<br>2. הרץ (בעזרת מ"ט U) את M על 5 הקלטים.<br>3. אם כולם התקבלו, קבל. אם אחד לפחות נדחה, דחה." | |

**דוגמא 3** - נסמן $ALL_{TM} = \{< M > \|L(M) = \Sigma *\}$ הראו $\overline{A_{TM}} \leq_m ALL_{TM}$ והסיקו כי $ALL_{TM}$ אינה ניתנת לזיהוי.

מ"ט F הבאה מחשבת את הרדוקציה f:
F="על הקלט <M,w> כאשר M מ"ט ו-w מחרוזת:
1. בנה את מ"ט M' הבאה:
M' = "על הקלט x:
1. הרץ (בעזרת מ"ט U) את M על w |X| צעדים.
2. אם קיבלה (בתוך |x| צעדים) דחה. אחרת, קבל."
2. החזר את <M'>."

וודאו כי מתקיים: <M,w>∈ATM ↔ <M'>∈ALLTM

צביקה פרלמוטר, 2017 ©

# סיבוכיות זמן

**חסם פולינומי:** $c > 0 \quad f(n) = n^c$ **חסם אקספוננציאלי:** $c > 0 \quad f(n) = 2^{cn}$ **מתקיים:** $2^{O(\lg n)} = n^c$
**אלגוריתם לא סביר:** בעל סיבוכיות זמן אקספוננציאלית
**מודלים חישוביים סבירים:** כל המודלים החישוביים הסבירים – שקולים זה לזה פולינומיאלית.

**7.7** תהי t הפונקציה $t: N \to R^+$, נגדיר את מחלקת סיבוכיות הזמן: $TIME(t(n))$: אוסף השפות הניתנות להכרעה ע"י מ"ט שרצה בזמן $O(t(n))$.

## סיבוכיות זמן במודלים שונים של מ"ט

**7.47** (בעיה) – שפה שניתנת להכרעה בזמן $o(n\lg n)$ (כלומר נמוך ולא שווה ל nlgn) **במכונה עם סרט אחד** היא **רגולרית**
**7.8** תהי $t(n)$ פונקציה, כך ש: $t(n) \geq n$. אזי לכל מ"ט **בעלת סרטים מרובים** הרצה בזמן $t(n)$, קיימת מ"ט שקולה, בעלת סרט יחיד, הרצה לכל היותר בזמן $O(t^2(n))$
**7.9 זמן ריצה של מ"ט ל"ד:** מוגדר לפי מסלול החישוב הארוך ביותר שלה על מילה. כלומר מסתכלים על כל מסלולי החישוב, גם כאלה שמסתיימים בדחייה והמכונה הל"ד חייבת להגיע ל $q_{accept}$ או $q_{reject}$ בתוך מגבלת הזמן.
**7.11** תהי $t(n)$ פונקציה כך ש: $t(n) \geq n$. לכל **מ"ט ל"ד** בעלת סרט יחיד הרצה בסיבוכיות זמן: $t(n)$, יש מ"ט **דטרמיניסטית** בעלת סרט יחיד – שקולה שרצה בסיבוכיות זמן: $2^{O(t(n))}$.

## המחלקות P ו NP

**המחלקה P :** $P = \cup_k TIME(n^k)$ – **L ניתנת להכרעה ע"י מ"ט דטרמיניסטית בזמן פולינומיאלי.**
**המחלקה P סגורה ל-** איחוד, חיתוך, משלים, שרשור, איטרציה (הוכחה בעזרת תכנות דינמי)
- אם $L \in P$ אז $\overline{L} \in P$

**דוגמאות לשפות ב P:**

- $PATH = \{\langle G,s,t\rangle \mid G$ is a **directed** graph that has a **directed** path from s to t$\}$ (7.14)
- $RELPRIME = \{\langle x,y\rangle \mid x$ and y are relatively prime$\}$ (7.15-כל זוגות המספרים הזרים שהמחלק הגדול ביותר שלהם הוא 1 )
- $ALL_{DFA} = \{\langle G\rangle \mid G$ is a DFA and $L(G) = \sum^*\}$ (תרגיל 7.10)
- $EVEN_{DFA} = \{< M > \mid M$ is a DFA, $|w|$ is even for all $w \in L(M)\}$
- COMPOSITES={x|x=pq, for integers p,q>1} שפת המספרים הפריקים
- 2SAT={<Ø>| Ø is a satisfiable 2cnf-formula} (מהמדריך-2CNF וספיקה)
- Regular ⊆ Context free ⊆ P **(Theorem 7.16)**
- $EQ_{DFA}$, 2-COLOR, $INFINIT_{REG}$ – תרגיל במצגת

**7.18** – **מאמת**: מאמת הוא מכונה, שיחד עם מילת קלט w מקבל כקלט אימות (הוכחה) c לשייכות של w לשפה.
L(V)={w| V accepts w#c for some string c} 🡸 ניתן להגדיר את NP כמחלקת השפות בעלי מאמת פולינומיאלי בגדול של w.
מדידת סיבוכיות הזמן של המאמת היא במונחי אורך הקלט w. (לדוגמא, מאמת לבעיית הפריקות הוא המחלק של 2 המספרים).

**המחלקה NP :** $NP = \{L \mid$ **L ניתנת לאימות בזמן פולינומיאלי**$\}$ נשים לב ש- $P \subseteq NP$ כי כל מכונה דט' היא גם ל"ד
או **הגדרה שקולה** $NP = \{L \mid$ **L ניתנת להכרעה ע"י מ"ט ל"ד בזמן פולינומיאלי**$\}$ (7.20)
**7.21** - {L|L is a language decided by a O(t(n)) time nondeterministic Turing machine} = NTIME(t(n))
**7.22** $NP = \cup_k NTIME(n^k)$ ו- $CoNP = \{\overline{L} \mid L\epsilon NP\}$ , **הערה** – לא ידוע אם NP=CoNP או NP≠CoNP
**המחלקה NP סגורה ל-** איחוד, חיתוך, שרשור, איטרציה (הוכחה בעזרת תכנות דינמי), לא ידוע אם סגורה למשלים
**דוגמאות לשפות שהן ב coNP אך המשלימה שלהן כנראה לא ב NP:** $\overline{CLIQUE}, \overline{HAMPATH}, \overline{IS}$
**7.26** $EXPTIME = \cup_k TIME\left(2^{n^k}\right)$

$P \subseteq NP \cap coNP$ **P ⊆ NP ⊆ EXPTIME (298)**

## רדוקציה בזמן פולינומיאלי

*בשונה מרדו' מיפוי שעשינו בין שפות מזהות-טיורינג כאן כל השפות הן כריעות 🡸 בין כולן יש רדו' מיפוי. נרצה להגביל בזמן
**7.29 - רדוקציה בזמן פולינומיאלי-** היא פונקציה ניתנת לחישובית בזמן פולינומיאלי (בגודל הקלט) : $f: \Sigma_A^* \to \Sigma_B^*$ שמקיימת:
$w \in A \Rightarrow f(w) \in B$ וגם $w \notin A \Rightarrow f(w) \notin B$ מסמנים : $A \leq_p B$
*ניתן להוכיח ש: אם $A \in P$ ו B לא טריוואלית כלשהי (כלומר לא $\emptyset, \Sigma^*$) אז $A \leq_P B$
*אם A כריעה ו B לא כריעה אז בטוח $A \nleq_p B$
**7.31** אם $A \leq_p B$ אז $B \in P \Rightarrow A \in P$ וכן $A \notin P \Rightarrow B \notin P$ ניתן להראות גם ש $B \in NP \Rightarrow A \in NP$ וכן $\overline{A} \leq_P \overline{B}$

**שקילות פולינומיאלית:** אם $B \leq_P A$ וגם $A \leq_P B$ נאמר ששתי השפות **שקולות פולינומיאלית** ונסמן : $A \equiv_P B$

*ניתן להוכיח כי זה מתקיים בפרט בין כל 2 שפות ב P או בין כל 2 שפות ב NPC!

**NP- שלמה:** שפה L היא NP שלמה (NPC) אם היא **מקיימת שני תנאים** :

**1.** $L \in NP$

**2.** לכל $L' \in NP$ מתקיים $L' \leq_P L$ (שפה המקיימת רק תנאי זה = NP-קשה).

**7.36** אם $L \in NPC$ ו- $A \in NP$ , וקיימת רדוקציה $L \leq_P A$ אז $A \in NPC$

**בעיית הספיקות של נוסחאות בוליאניות (SAT):** בהינתן נוסחא בוליאנית בת n משתנים, יש לקבוע האם קיימת לה הצבת ערכים מתאימה (אחת מבין $2^n$ ההצבות האפשריות) המספקת אותה, כלומר הגורמת לה לקבל את הערך 1

**נוסחת-cnf** – היא נוסחא בוליאנית המורכבת ממספר פסוקיות המקושרות ע"י "וגם" לדוגמא: $\Phi = (x_1 \vee x_2 \vee x_3) \wedge (x_3 \vee x_5 \vee x_4) \ldots$

* m = מס' הפסוקיות, n = מס' הליטרלים בכל פסוקית

*אם Φ ספיק אז יש לו ליטרל אחד לפחות בכל פסוקית שערכו True.

**נוסחת 3cnf** - נוסחת cnf שבה כל הפסוקיות מורכבות מ- 3 ליטרלים

**משפט קוק-לוין:** $SAT \in NPC$

**כדי להוכיח ששפה שייכת ל NPC יש צורך להראות רדוקציה פולינומיאלית לשפה אחרת שנמצאת ב NPC:**

**1.** $L \in NP$

**2.** $SAT \leq_P L$ (במקום SAT אפשר כל שפה אחרת שהוכחנו ב NPC) – בהינתן מופע של SAT נבנה מופע של L :

א. נגדיר פונקציה f הממפה מופע של L למופע מתאים של SAT.

ב. הרדוקציה תקפה - נראה כי $x \in A$ אמ"מ $f(x) \in B$

ג. נוכיח שזמן הריצה של f פולינומיאלי בגודל הקלט

**דוגמאות לרדוקציות לבעיות NPC (חלק 2 של ההוכחה בלבד):**

**בעיות בלוגיקה**

$SAT \leq_P 3SAT$

$3SAT=\{\Phi|$ לפסוק Φ בצורה 3-CNF (כלומר כל פסוקית מכילה בדיוק 3 ליטרלים) קיימת השמה מספקת $\}$

בהינתן פסוק Φ בצורת CNF מופע של SAT נבנה (פונקציית הרדוקציה) פסוק Φ' בצורת 3CNF מופע של 3SAT:

עבור כל פסוקית ב-Φ : אם m=3 נשאיר כפי שהיא,

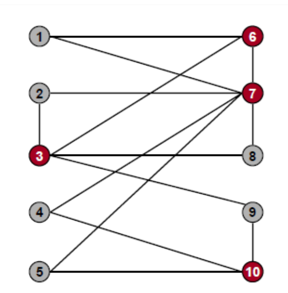
אם m<3 נכפיל את אחד הליטרלים

אם m>3 אז לכל פסוקית נבנה פסוקיות מתאימות:

טענה: $\Phi \in SAT \leftrightarrow \Phi' \in 3SAT$ (ע"מ 310)

$$C_j = x_1 \vee \overline{x_3} \vee \overline{x_4} \vee x_5 \vee x_6 \vee \overline{x_9} \Rightarrow \begin{array}{l} C'_{j1} = x_1 \vee x_1 \vee y_1 \\ C'_{j2} = \overline{y_1} \vee \overline{x_3} \vee y_2 \\ C'_{j3} = \overline{y_2} \vee \overline{x_4} \vee y_3 \\ C'_{j4} = \overline{y_3} \vee x_5 \vee y_4 \\ C'_{j5} = \overline{y_4} \vee x_6 \vee y_5 \\ C'_{j6} = \overline{y_5} \vee \overline{x_9} \vee \overline{x_9} \end{array}$$

**בעיות על גרפים**

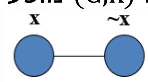



1) $3SAT \leq_P VC$

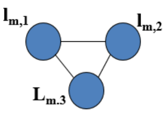
Vertex Cover – בעיית **כיסוי הקודקודים את הקשתות** כלומר לכל $(x,y) \in E$ מתקיים $x \in U$ או $y \in U$. לדוגמא: בגרף משמאל קיים כיסוי בגודל 4 אך לא קיים בגודל 3.

$VC=\{(G,k)|\ |U| \leq k$ ,U כיסוי קודקודים, G גרף לא מכוון, קיים ל-G $\}$

בהנתן Φ מופע של 3SAT בצורת 3-CNF נבנה (G,k) מופע של VC

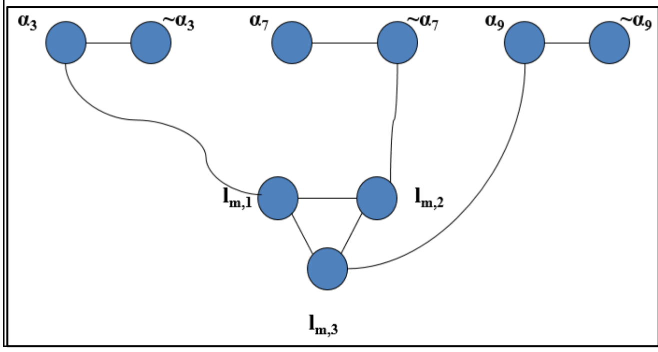



עבור כל פסוק אטומי x של Φ נוסיף לגרף:




עבור כל פסוקית $C_m=(l_1 \vee l_2 \vee l_3)$ נוסיף לגרף:

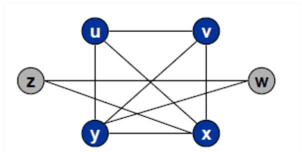
כאשר צמתי הפסוקית מחוברים בקשתות לצמתי הליטרלים המתאימים. כמו כן נקבע $k=n+2m$ כאשר n-מספר הפסוקים האטומים ו-m מספר הפסוקיות.



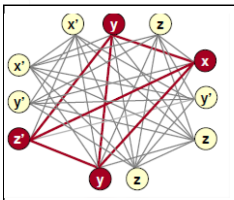

צביקה פרלמוטר, 2017 ©

לדוגמא עבור הפסוקית: $C_m=(\alpha_3 \vee \sim\alpha_7 \vee \alpha_9)$ הגרף יראה כמשמאל. (ע"מ 313 בספר)
טענה: $\Phi\in 3SAT \leftrightarrow (G,k)\in VC$
**צד ראשון** - בהינתן ביטוי ספיק נראה כי קיים כיסוי קודקודים מתאים: מבין הזוגות של הליטרל והיפוכו -ניקח ל VC את הצמתים אלה שערכם TRUE
ומבין המשולשים (הפסוקיות) –נבחר צומת אחד שמחובר לצומת של ליטרל TRUE ונוסיף ל VC את 2 הצמתים המחוברים אליו. קיבלנו סה"כ $k=n+2m$ צמתים, ואלה מכסים את כל הקשתות כי כל משתנה מכוסה וכל פסוקית מכוסה.
**צד שני** – בהינתן כיסוי קודקודים VC של G בעל k צמתים נראה כיצד ניתן לבנות לביטוי ספיק השמה מספקת:
על מנת לכסות את כל המשתנים והפסוקיות, על כיסוי הקודקודים VC להכיל צומת אחד לכל משתנה ו-2 צמתים לכל פסוקית.
ניקח את כל הצמתים של הליטראלים שנמצאים ב VC ונשים בהם TRUE, ובשאר הליטראלים FALSE. השמה זו מספקת מכוון שכל פסוקית מכוסה ומקושרת לליטרל TRUE אחד לפחות ולכן ההשמה הנ"ל מספקת את הפסוק כולו.

2) $3SAT\le_P$**Clique**



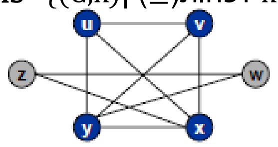

**Clique**$=\{(G,k)|$ G גרף לא מכוון וקיימת בו קלילה בגודל k לפחות$(\le)\}$
לדוגמא: בגרף משמאל קיימת קליקה בגודל 4 אך לא קיימת בגודל 5.
הוכחה בעמוד 302 בספר (7.24 + 7.32+7.43)
להלן תיאור כללי של הרדוקציה:



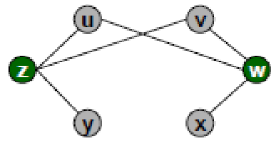

בהינתן Φ מופע של 3SAT בצורת 3-CNF נבנה (G,k) מופע של Clique. הרעיון – נבנה גרף שיתאים קודקוד לכל ליטרל
בפסוק המקורי. נחלק את הגרף לקבוצות בשם triples כך שיתאימו לליטרלים של כל פסוקית.
נמתח קשתות בין כל הקודקודים פרט ל 2 מקרים: ליטרל והיפוכו (לדוגמא לא נמתח בין $x_1$ ל $\overline{x_1}$) או קודקודים שנמצאים באותו ה triple. k = מספר ה triples. ציור בעמוד 303 בספר.
$C=(x'+y+z)(x+y'+z)(y+z)(x'+y'+z')$

3) $Clique\le_P$**IS**

**IS**$=\{(G,k)|$ G גרף לא מכוון וקיימת בו קבוצה ב"ת בגודל k לפחות$(\le)\}$

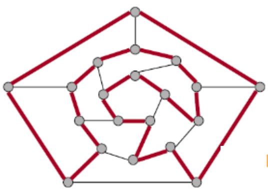



הוכחה בעמ' 91 (4.10) במדריך – הרעיון = מעבר לגרף המשלים
4) $IS\equiv_P$**VC** – הוכחה בעמ' 91 במדריך (4.10.3)

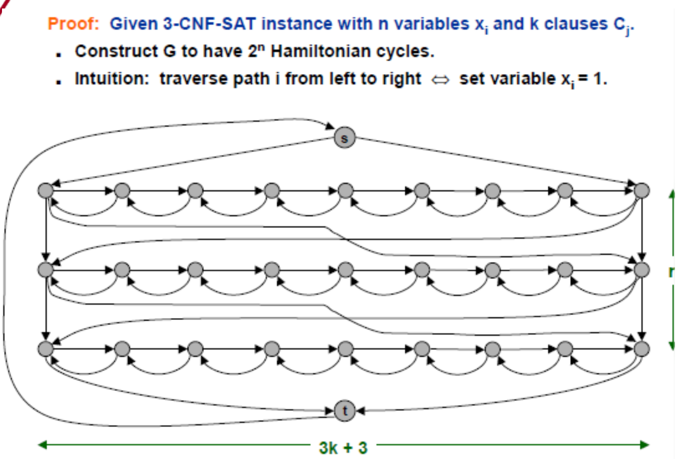


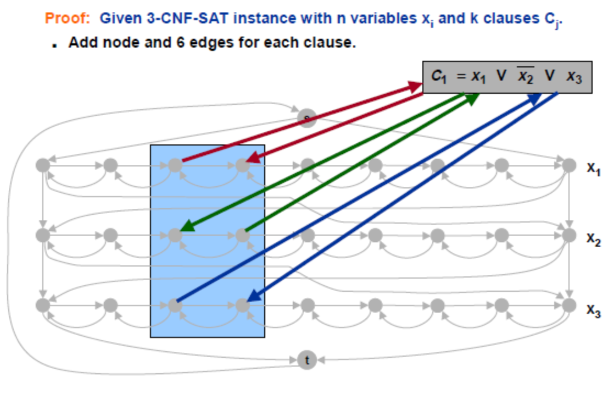

- Given an undirected graph G = (V, E), its complement is G' = (V, E'), where E' = { (v, w) : (v, w) ∉ E}.
- G has a clique of size k if and only if G' has a vertex cover of size |V| - k.

5) $3SAT\le_P$**HAMPATH**

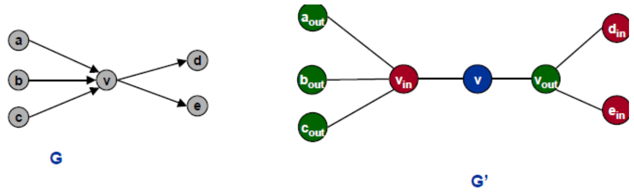


**__מסלול המילטוני בגרף מכוון:__** הוא מסלול פשוט ומכוון, העובר בכל הקדקודים בדיוק פעם אחת.
**HAMPATH**$=\{(G,s,t)|$ G גרף **מכוון** וקיים בו מסלול המילטוני שמתחיל ב s ומסתיים ב t$\}$
**UHAMPATH**$=\{(G,s,t)|$ G גרף **לא מכוון** וקיים בו מסלול המילטוני שמתחיל ב s ומסתיים ב t$\}$
הוכחה בעמוד 314 בספר (7.46)
הרעיון: כיוון -ניתן לנוע שמאלה או ימינה.




6) ניתן להראות ש $HAMPATH\le_P$**UHAMPATH** – עמוד 319 בספר(7.55)
רעיון ההוכחה – ניצור מגרף מכוון G בעל n קשתות, גרף לא מכוון G' בעל 3n קשתות. לכל v יהיה: $v_{in}$-v-$v_{out}$




7) $HAM_{cycle} \leq_P$ **TSP**

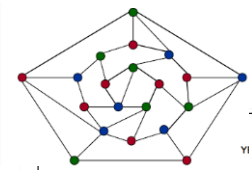
בעיית הסוכן הנוסע – בגרף **מלא** G עם משקלים האם קיים מעגל המילטוני במשקל כולל לכל היותר k?

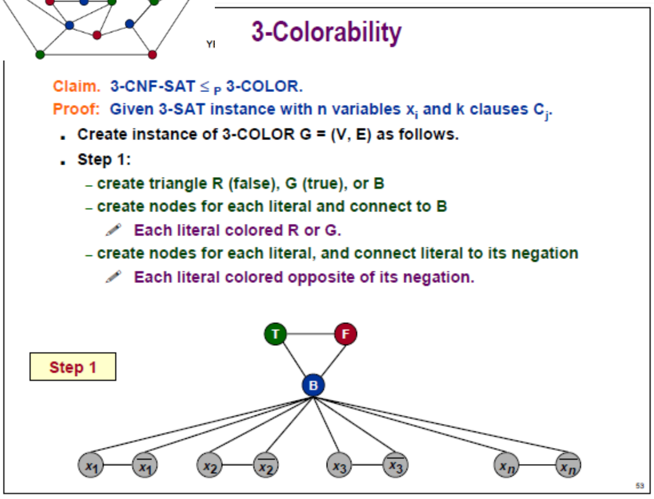
TSP={(G,c,k)| בגרף מלא G עם פונקציית משקל c קיים מעגל המילטוני במשקל של לכל היותר k }

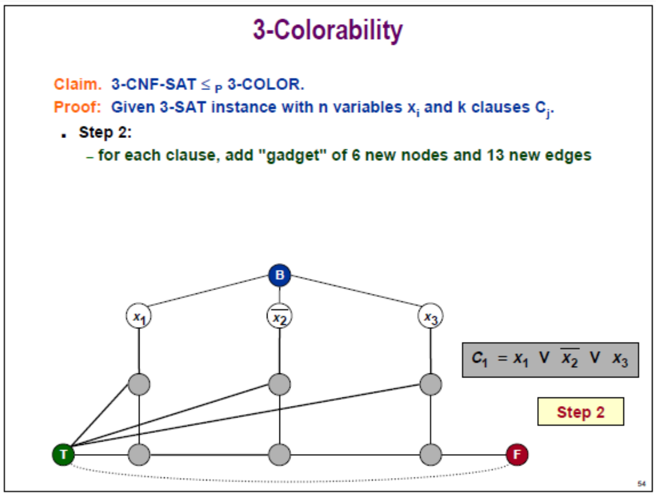
| |
|---|
| בהינתן גרף G=(V,E) מופע של $HAM_{cycle}$ נבנה גרף משוקלל G' וערך k מופע של TSP: G'=(V,E'), E'=VXV |
| עם פונקציית משקל: $C(u,v) = \begin{cases} 0 & (u,v) \in E \\ 1 & (u,v) \notin E \end{cases}$ כמו כן נקבע k=0. טענה: $G \in HAM_{cycle} \leftrightarrow (G',c,k) \in TSP$ |

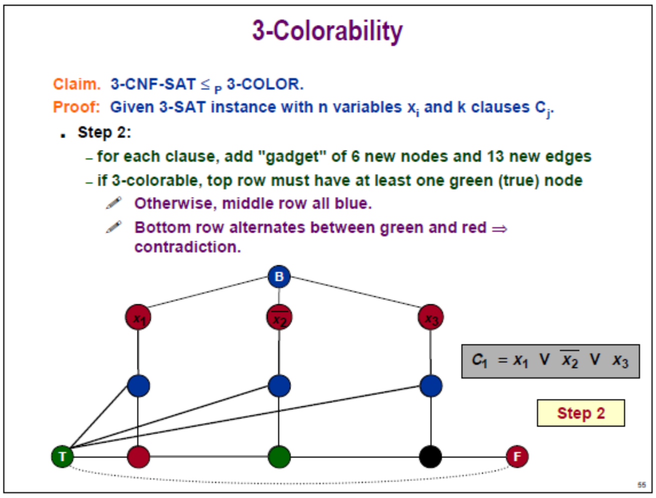
## בעיות צביעה בגרף

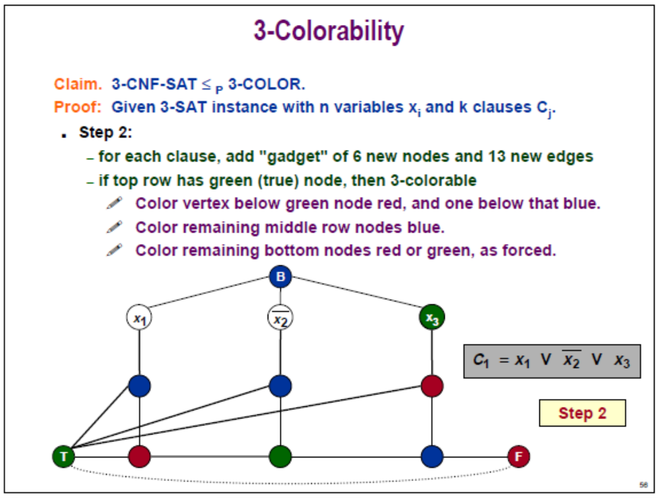
1) 3SAT$\leq_P$**3-COLOR**






## בעיות על קבוצות

1) 3SAT$\leq_P$**SUBSET-SUM**

מקבלים קבוצה $S=\{x_1,\ldots,x_n\}$ של מספרים טבעיים ומספר טבעי t.

SUBSET-SUM={(S,t)| קיימת תת קבוצה של S שסכום המספרים בה הוא בדיוק t }

הוכחה בעמוד 320 בספר (7.25+7.56)

**Example:** X = {1, 4, 16, 64, 256, 1040, 1041, 1093, 1284, 1344}, t = 3754.

- **YES:** S = {1, 16, 64, 256, 1040, 1093, 1284}.

2) VC$\leq_P$**Set-Cover**

**Ex.** U = {1, 2, 3, . . . , 12}, k = 3.

- $S_1$ = {1, 2, 3, 4, 5, 6} $S_2$ = {5, 6, 8, 9}
- $S_3$ = {1, 4, 7, 10} $S_4$ = {2, 5, 7, 8, 11}
- $S_5$ = {3, 6, 9, 12} $S_6$ = {10, 11}

**YES:** $S_3$, $S_4$, $S_5$.

נתונה קבוצה S , משפחה של תתי קבוצות שלה $\{S_1,S_2,\dots,S_m\}(1\le i\le m)$ ומספר טבעי k. הבעיה-האם ניתן לבחור לכל היותר k תתי קבוצות מתוך המשפחה הנתונה כך שאיחודם הוא S?
דוגמא – יש מכללה עם n סטודנטים ו m כיתות (לאו דווקא זרות) ויש להעביר הודעה לכל הסטודנטים במספר כיתות מינימלי.
SC={(S,(Si),k)| קיים ל-S כיסוי בקבוצות (Si) בגודל k לכל היותר}
הוכחה בתרגיל בעמ' 92 (4.11) במדריך .

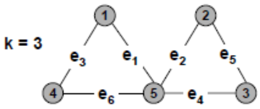
## Subset Sum

Treat as base k+1 integer

Claim. VERTEX-COVER $\le_P$ SUBSET-SUM.

Proof. Given instance G, k of VERTEX-COVER, create following instance of SUBSET-SUM.

k = 3




| | e1 | e2 | e3 | e4 | e5 | e6 |
|---|---|---|---|---|---|---|
| v1 | 1 | 0 | 1 | 0 | 0 | 0 |
| v2 | 0 | 1 | 0 | 0 | 1 | 0 |
| v3 | 0 | 0 | 0 | 1 | 1 | 0 |
| v4 | 0 | 0 | 1 | 0 | 0 | 1 |
| v5 | 1 | 1 | 0 | 1 | 0 | 1 |

Node-arc incidence matrix

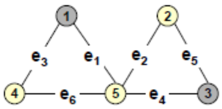
| | | e1 | e2 | e3 | e4 | e5 | e6 | decimal |
|---|---|---|---|---|---|---|---|---|
| x1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 5,184 |
| x2 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 4,356 |
| x3 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 4,116 |
| x4 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 4,161 |
| x5 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 5,393 |
| y1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1,024 |
| y2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 256 |
| y3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 64 |
| y4 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 16 |
| y5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 4 |
| y6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| t | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 15,018 |
| | k | | | | | | | |

## Subset Sum

Claim. G has vertex cover of size k if and only if there is a subset S that sums to exactly t.

Proof. $\Rightarrow$

- Suppose G has a vertex cover C of size k.
- Let S = C $\cup$ { $y_j$ : $|e_j \cap C| = 1$ }
  - most significant bits add up to k
  - remaining bits add up to 2



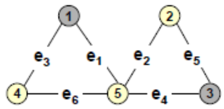

| | | e1 | e2 | e3 | e4 | e5 | e6 | decimal |
|---|---|---|---|---|---|---|---|---|
| x1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 5,184 |
| x2 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 4,356 |
| x3 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 4,116 |
| x4 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 4,161 |
| x5 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 5,393 |
| y1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1,024 |
| y2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 256 |
| y3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 64 |
| y4 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 16 |
| y5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 4 |
| y6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| t | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 15,018 |
| | k | | | | | | | |

## Subset Sum

Claim. G has vertex cover of size k if and only if there is a subset S that sums to exactly t.

Proof. $\Leftarrow$

- Suppose subset S sums to t.
- Let C = S $\cap$ {$x_1, \dots, x_n$}.
  - each edge has three 1's, so no carries possible
  - |C| = k
  - at least one $x_i$ must contribute to sum for $e_j$




| | | e1 | e2 | e3 | e4 | e5 | e6 | decimal |
|---|---|---|---|---|---|---|---|---|
| x1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 5,184 |
| x2 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 4,356 |
| x3 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 4,116 |
| x4 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 4,161 |
| x5 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 5,393 |
| y1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1,024 |
| y2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 256 |
| y3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 64 |
| y4 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 16 |
| y5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 4 |
| y6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| t | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 15,018 |
| | k | | | | | | | |

3) SUBSET-SUM$\le_P$**PAR**
PAR={(S)| קיימת חלוקה של איברי S ל 2 קבוצות זרות ושוות סכום }
רדוקציה: $A=\sum_{i-1}^{k} x_i$ כאשר $f(x_1,..x_s,k)=x_1,\dots,x_s,B=2A-k,C=A+k$
אבחנה: סכום האיברים הוא 4A בשל החלוקה B,C לא יימצאו ביחד (כי סכומם ביחד הוא 3A). הקבוצה I שסכומה הוא $k$, ביחד עם B תהיה בסכום של 2A ולכן זוהי חלוקה. בניה מבטיחה שאם לקלט הרדוקציה יש פתרון ב SS אז לפלט יש פתרון ב PART.
בצד השני- בהינתן פתרון ל PART, נתבונן ב "צד" שמכיל את B, נבחר בתור פתרון ל SS את כל ה $x_i$ ים מאותו צד. מאותו שיקול ברור כי זהו פתרון מתאים ל SS (סכומו הוא $k$).
4) PAR$\le_P$**BIN-PACKING**
קלט: k תאים בגודל B כל אחד ועצמים שגדלהם $x_1,\dots,x_s$
פלט: האם ניתן להכניס את העצמים ל k התאים כך שבכל תא סכום העצמים אינו גדול מ B
הוכחה – $f(x1,\dots x_s)=\left(x1,\dots,x_s,k=2,B=\frac{1}{2}\sum_{i=1}^{s}x_i\right)$, מחלקים ל 2 תאים כאשר בכל תא יש בדיוק חצי מהסכום.
5) SUBSET-SUM$\le_P$**KNAP-SACK**
נתונה קבוצת חפצים $A=(a_1,\dots,a_n)$ ולכל חפץ יש ערך U ונפח V. לגגב תרמיל בנפח W , האם ניתן לקחת חפצים בשווי סה"כ שגדול מ k? כלומר לבחור קבוצה B מתוך A כך ש $\sum_{j\in B} v_i \le W$ וגם $\sum_{j\in B} u_i \ge k$
רדוקציה: עבור הקלט $C=(c_1,\dots,c_n),T$ לבעיית SUBSET-SUM נבנה מופע ל KNAP-SCACK כך: $A=(a_1,\dots,a_n)$, $U=V=C$ ואז $W,k=T$

צביקה פרלמוטר, 2017 ©

# סיבוכיות מקום:

**הגדרה - סיבוכיות מקום (8.1):** תהי M מכונת טיורינג **דטרמיניסטית** שעוצרת על כל קלט. סיבוכיות המקום של M היא הפונקציה $f: N \rightarrow N$ כך ש$f(n)$ הוא המספר המקסימאלי של "תאי סרט" ש M "סורקת", לכל קלט שהוא באורך n . נאמר ש- M רצה ב"מקום" ש$f(n)$.

תהי M מכונת טיורינג **לא דטרמיניסטית** שבה כל ענפי החישוב **עוצרים** על כל קלט. סיבוכיות המקום של M, $f(n)$ היא: "המספר המקסימאלי של תאי סרט ש M "סורקת" על כל ענף שהוא של החישוב שלה, לכל קלט שהוא באורך n."

**8.2 מחלקות סיבוכיות מקום-** $f: N \rightarrow R^+$

$$\mathbf{SPACE}(f(n)) = \{\, L \mid \text{היא שפה המוכרעת ע\"י \textbf{מט\"ד} שרצה בסיבוכיות מקום} f(n) \,\}$$

$$\mathbf{NSPACE}(f(n)) = \{\, L \mid \text{היא שפה המוכרעת ע\"י \textbf{מטל\"ד} שרצה בסיבוכיות מקום} f(n) \,\}$$

$$\mathbf{PSPACE} = \text{מחלקת השפות שניתנות להכרעה במקום פולינומיאלי ע\"י מכונת טיורינג דטרמיניסטית} = \bigcup_k \mathrm{SPACE}(n^k)$$

$$\mathbf{NPSPACE} = \text{מחלקת השפות שניתנות להכרעה במקום פולינומיאלי ע\"י מכונת טיורינג ל\"ד} = \bigcup_k \mathrm{SPACE}(n^k)$$

$$\mathbf{EXPSPACE} = \bigcup_k \mathrm{SPACE}(2^{n^k})$$

**לדוגמא** $SAT \epsilon SPACE(n) \subseteq PSAPCE$ כי יש צורך רק לשמור את מצב המשתנים השונים ולבדוק על הפסוק עד לקבלת השמת אמת, כלומר סה"כ m משתנים, שהוא לכל היותר גודל הקלט- n.

**שאלה – האם מכאן נובע ש** $NP \subseteq SPACE(n)$? **לא.** הרדוקציה של שפה L ב NP ל SAT לא מחזירה בהכרח פלט שהוא ליניארי בגודל הקלט

$\overline{ALL_{NFA}} \epsilon NSPACE(n)$*– כלומר שפת כל האסל"דים שבהם קיימת מילה שלא מתקבלת. הרעיון : ננחש מחרוזת שה NFA דוחה ונשתמש במקום ליניארי כדי לעקוב אחר המצבים שה NFA יכול להיות בהם בזמן מסויים. (תיאור המכונה עמ' 333)

*הערה חשובה – אסור לתרגם את האס"ד לאסל"ד כי זה יקח מקום אקספוננציאלי. אז מה נעשה? ראינו שאם יש מילה שלא מתקבלת מספיק לבדוק עד גודל האוטומט. אם יש m מצבים באס"ד יש צורך ב $2^m$ בדיקות כלומר הזמן אספוננציאלי. אבל מה לגבי המקום? צריך רק מונה (ימנה) עד $2^m$ ← אם נשמור בבינארי, סה"כ נדרש m מקום) ולשמור את המצבים שאליהם הגענו וצריך לראות האם מתישהו הגענו ממילה מסוימת לקבוצת מצבים שכולם לא מקבלים אז המילה לא מתקבלת.

**הוכחה ש** $m\text{-PATH} \in SPACE(n)$**:**

"על הקלט <G,s,m> כאשר G גרף ל"מ, s צומת ו-m מספר טבעי:

1. אם m=0, קבל.
2. סמן את s.
3. לכל צומת v שכן של s שאינו מסומן, בצע:
   3.1 הפעל את האלג' m-PATH על הקלט <G,v,m-1>
   3.2 אם האלג' קיבל, קבל.
4. הסר את הסימון מ-s
5. דחה"

## מסקנות ממטלות בקביעת סיבוכיות מקום:

*מונה -ניתן לממש בינארי ב O(lgn)

*כדי לעבור על תת קבוצות של קבוצה בגודל k, ניתן לעבור בסדר לקסיקוגרפי על כל המילים באורך k מעל {0,1}, $x_i=1$ אם שייך לקבוצה ולהפך

*המקום הדרוש לשמירת סכום של m מספרים איננו גדול מהמקום הדרוש ל m מספרים

*בניית אוטומט מכפלה – גודלו חסום ע"י ריבוע גודל הקלט כי מס' מצביו יהיה מס' מצבי A*מס' מצבי B – לכן $SPACE(n^2)$

**משפט Savitch (8.5):** מ"ט דטרמיניסטית – לא משתמשת בכמות זיכרון רבה יותר (משמעותית) ממטל"ד לאותה שפה. מה שמטל"ד מכריעה ב- f(n) מקום, מ"ט דטרמיניסטית מכריעה ב $f^2(n)$ מקום לכן: NPSPACE=PSPACE

$$\mathbf{NSPACE}(\mathbf{f}(\mathbf{n})) \subseteq \mathbf{SPACE}\left(\mathbf{f}^2(\mathbf{n})\right) \quad \text{where } f(n) \geq \log n$$

**מסקנה חשובה:** אם רוצים להראות ששפה שייכת ל$\mathrm{SPACE}(n^2)$ אפשר גם להראות שהיא שייכת לNSPACE(n)

**לדוגמא** מכיון שהראנו שמתקיים $\overline{ALL_{NFA}} \epsilon NSPACE(n)$ הרי ש $\overline{ALL_{NFA}} \epsilon SPACE(n^2)$ לפי Savitch. (8.4)

בנוסף מתקיים גם $ALL_{NFA} \epsilon SPACE(n^2) \subseteq PSAPCE$ כי כשמדובר על הכרעה במכונת טיורינג דטרמיניסטית אז אם הראנו מגבלות זמן ומקום לשפה מסויימת, אז יש גם אותם המגבלות לשפה המשלימה כי בסה"כ צריך להחליף מצב accept ב reject ולהפך.

**להלן ההיררכיה המתקבלת:**

$$\mathbf{P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME\ (337)}$$

- באופן לא פורמלי – **סיבוכיות זמן תמיד גדולה מסיבוכיות מקום** (זיכרון אפשר למחזר, זמן לא)
- **הקשר בין P לPSPACE:** אלגוריתם שרץ בזמן פולינומיאלי, לא יכול להשתמש ביותר ממקום פולינומיאלי. או באופן פורמאלי יותר : אלגוריתם הפועל בזמן t(n) יכול להשתמש לכל היותר בt(n) מקום - יכול לבקר בתא אחד לכל היותר בכל צעד של חישוב.

**הקשר בין NP לPSPACE** : אלגוריתם לא דטרמיניסטי שרץ בזמן פולינומיאלי, לא יכול להשתמש ביותר ממקום פולינומיאלי

**כמה זמן צורך אלגוריתם הצורך $f(n)$ מקום**: מ"ט M הצורכת $f(n)$ **מקום** לצורך חישוב, צורכת לכל היותר $2^{O(f(n))}$ **זמן** לצורך החישוב.

**PSPACE-שלמות**: שפה B שייכת לקבוצה **PSPACE-שלמות** אם היא מקיימת את שני התנאים :

1. B שייך ל- PSPACE
2. כל שפה A השייכת ל- PSPACE ניתנת לרדוקציה **בזמן** פולינומיאלי ל- B. **(PSPACE-קשה)**

**לדוגמא** $\boldsymbol{TQBF} = \{\langle\phi\rangle \mid \phi \text{ is a true fully quantified Boolean formula}\}\ \boldsymbol{\epsilon PSPACEC}$ (8.9)

**סיבוכיות מקום תת לינארית** (המחלקות L ו NL) – אפשר לקרוא את כל הקלט, אבל אין מקום לשמור את כולו

**מ"ט חדשה לצורך סיבוכיות מקום תת לינארית -** נגדיר מ"ט חדשה המכילה שני סרטים:

1. **סרט הקלט** - לקריאה בלבד
2. **סרט עבודה** – בגודל $O(\log n)$

בסרט הראשון הראש הקורא יכול להמצא רק על חלק הסרט בו כתוב הקלט. לכן צריך לזהות את הקצה הימני והשמאלי. לבי סרט העבודה – עליו הראש הקורא/כותב יכול לנוע וגם לכתוב. **רק התאים שנסרקו בסרט העבודה משמשים לחישוב סיבוכיות המקום של המכונה.**

- ניתן לחשוב על אלגוריתם כזה, המשתמש בסיבוכיות מקום תת לינארית, כעל אלגוריתם המבצע מניפולציה לקלט, מבלי לשמור את כולו בזיכרון.

מחלקת השפות הניתנות להכרעה בסיבוכיות מקום לוגריתמית (בגודל הקלט) ע"י מ"ט דטרמ'= **L=SPACE (log n)**

מחלקת השפות הניתנות להכרעה בסיבוכיות מקום לוגריתמית (בגודל הקלט) ע"י מ"ט לא דטר'= **NL=NSPACE (log n)**

**לדוגמא: (1** $\left\{\boldsymbol{0^k1^k}\right\} \in L$ - דרך א' - נשתמש בתחזוק של 2 מונים בבסיס בינארי ונשווה את 2 המונים.

דרך ב' – לבצע זגזג ואת המקום לשמור בסרט העבודה באופן בינארי (8.18)

**(2** $\boldsymbol{PATH} \in NL$ – בעיית האם קיים מסלול מכוון מ- s ל t. פתרון – נמספר את הצמתים. נחזיק כל פעם צומת וננחש את הצומת הבאה. אורך המסלול המקסימלי יכול n ולכן נתחזק מונה ונעצור כשנעבור n צמתים. כלומר סה"כ צריך לשמור מונה + צומת = log n (8.19)

**8.20 – קונפיגורציה של M על w:** תהי *M* מ"ט בעלת סרט קלט read-only נפרד, ו *w*– מחרוזת קלט. **קונפיגורציה של *M* על *w***, היא "תמונה" של: המצב, סרט העבודה ומיקום שני הראשים של שני הסרטים (סרט הקלט וסרט העבודה).מחרוזת הקלט, *w*, אינה חלק מהקונפיגורציה של *M* על *w*.

- אם *M* היא מ"ט שרצה במקום ***f*(*n*)** ו *w* הוא קלט באורך *n*, אזי **מספר הקונפיגורציות** של *M* על *w* הוא $\boldsymbol{2^{O(f(n))}}$

**רדוקציה במקום לוגריתמי:**

זוהי רדוקצית מיפוי שהיא עדינה יותר מרדוקציה פולינומיאלית. כדי להגדיר היטב רדוקצית מקום לוגריתמי נגדיר :

- **מתמר מקום לוגריתמי** : מכונת טיורינג בעלת 3 סרטים:
  1. סרט קלט – לקריאה בלבד
  2. סרט פלט – לכתיבה בלבד
  3. סרט עבודה – לקריאה וכתיבה -יכול להכיל $O(\log n)$ תווים בלבד (כאשר *n* אורך הקלט)
- **כדי להוכיח ש** $\boldsymbol{A \le_L B}$ צריך בהינתן קלט של A לבנות קלט של B + צריך להוכיח את נכונות הרדוקציה + צריך להוכיח שהבנייה מתבצעת במקום לוגריתמי

**8.22 – NL שלמות:** שפה B היא ב NL – שלמות אם :

1. $B \in NL$
2. כל $A \in NL$ ניתנת לרדוקציית מקום לוגריתמי ל- B (**הערה:** מספיק להראות $PATH \le_L B$)

**8.23** אם $\boldsymbol{A \le_L B}$ ו $B \in L$ אז $\boldsymbol{A \in L}$

*אם נוכיח עבור שפה NLC כלשהי שהיא מוכלת ב L אז L=NL **(8.24)**

**לדוגמא:** $\boldsymbol{PATH} \in \boldsymbol{NLC}$ **(8.25)**

**טיפים ממטלות (בנושא רדו'):**

*אם בניית אוטומט לא תלויה בקלט w, אז ניתן לבנות אותו פעם אחת ולתמיד

*רדוקציה שדורשת רק מעבר על הקלט והעתקתו ניתנת למימוש ע"י מונה ואז זה ידרוש מקום לוגוריתמי

*מעבר על גרף – ניתן לבצע ע"י 2 מונים על זוגות סדורים של צמתים (s,t) בגרף

**להלן ההירכיה המתקבלת (8.27 ,8.26):**

$$\mathbf{L \subseteq NL = coNL \subseteq P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME}$$

# משפטי היררכיה:

**9.1 - פונקציה הניתנת לבנייה במגבלת מקום עצמית:** נאמר שפונקציה $f$:N→N **ניתנת לבניה במגבלת מקום עצמית** אם היא לפחות O(logn) וניתן לחשב את המיפוי של המחרוזת $1^n$ לייצוג הבינארי של $f(n)$ בסיבוכיות מקום של O($f(n)$). ($f(n)=\Omega$ (logn))

**דוגמא** – חישוב $n^2$: קלט – n, ייצוג המחרוזת – n פעמים 1, ממירים את n לייצוג הבינארי שלה (מניה של ה-1 ים), אורך המונה: $\log_2(n)$. עכשיו מכפילים אותו בעצמו לחישוב $n^2$ ואורך התוצאה: $\log_2(n^2)=2\log_2(n)$ והסיבוכיות קטנה מ$O(n^2)$

**9.3 – משפט היררכיית המקום:** לכל פונקציה $f$ הניתנת לבניה במגבלת מקום עצמית $f$:N→N, קיימת שפה $A$ כך ש: A ניתנת להכרעה בסיבוכיות מקום O(f(n)) אך אינה ניתנת להכרעה בסיבוכיות מקום o(f(n))

**הוכחה: רעיון –** נראה קיומה של שפה A הניתנת להכרעה בסיבוכיות מקום O(f(n)) אך לא בסיבוכיות מקום o(f(n)) (עבור f(n) הניתנת לבנייה במגבלת מקום עצמית). נתאר את A בעזרת מ"ט D המכריעה אותה בסיבוכיות מקום O(f(n)) ושונה מכל מ"ט M המכריעה שפה בסיבוכיות מקום o(f(n))

**משפט היררכיית המקום (הצעה ראשונית)**

D" = על הקלט w (|w|=n):
1. אם w אינה תיאור של מ"ט M, דחה.
2. אחרת (w=<M>) התחל להרץ את M על <M>
   2.1 אם יש חריגה מ-f(n) תאי מקום, דחה.
   2.2 אחרת אם M דחתה, קבל. אם M קיבלה, דחה. "

**תוצאה (שיטת האלכסון)**: D שונה מכל מ"ט M המכריעה במקום o(f(n)) ביחס לקלט <M>.

**בעיות**: 1.התייחסות ללולאה אינסופית בהרצת M על <M> 2.עבור ערכי n קטנים יתקיים f(n)<g(n)

**משפט היררכיית המקום (הצעה מתוקנת)**

D" = על הקלט w (|w|=n):
1. אם w אינה מהצורה $\langle M\rangle 10\ldots0^k$, דחה.
2. אחרת התחל להריץ את M על $\langle M\rangle 10\ldots0^k$
   2.1 אם יש חריגה מ-f(n) תאי מקום, דחה.
   2.2 אם יש חריגה מ-$2^{f(n)}$ צעדים, דחה.
   2.3 אחרת אם M דחתה, קבל. אם M קיבלה, דחה. "

**9.4** לכל שתי פונקציות $f_1$,$f_2$:N→N, כך ש: $f_1(n)=o(f_2(n))$ ו- $f_2$ ניתנת לבניה במגבלת מקום עצמית, מתקיים: $SPACE(f_1(n)) \subset SPACE(f_2(n))$

**מסקנות מ9.4:**

- לכל מספר טבעי $c$, ניתן להראות שהפונקציה $n^c$ ניתנת לבניה במגבלת מקום עצמית.
- לכן, לכל שני מספרים טבעיים $c_1 < c_2$ מתקיים : $\mathbf{SPACE(n^{c_1}) \subset SPACE(n^{c_2})}$
- ניתן להראות גם שלכל מספר רציונאלי $c$, הפונקציה $n^c$ ניתנת לבניה במגבלת מקום עצמית.
- גם לכל שני מספרים רציונאליים $0 \le c_1 < c_2$ ההכלה מתקיימת.
- (9.6) $\mathbf{NL \subsetneq PSPACE}$
- (9.7) $\mathbf{PSPACE \subsetneq EXPSAPCE}$

**9.5** בין כל 2 מספרים ממשיים, נמצאים תמיד לפחות 2 מספרים רציונליים כך ש $\varepsilon_1 < c_1 < c_2 < \varepsilon_2$ ולכן **לכל 2 מספרים ממשיים** $0 \le \varepsilon_1 < \varepsilon_2$ מתקיים $SPACE(n^{\varepsilon_1}) \subset SPACE(n^{\varepsilon_2})$

**9.8 פונקצית זמן הניתנת לבניה במגבלת זמן עצמית:** פונקציה $t$:N→N, כך ש $t(n)$ היא לפחות O($n$log$n$) תקרא : **פונקציה הניתנת לבניה במגבלת זמן עצמית** אם הפונקציה הממפה את המחרוזת $1^n$ לייצוג הבינארי של $t(n)$ ניתנת לחישוב בזמן $t(n)$. כלומר : אם קיימת מ"ט M כך שבהנתן הקלט $1^n$ היא עוצרת עם היצוג הבינארי של $t(n)$ על הסרט תוך O($t(n)$) זמן.

**9.10 – היררכיית הזמן:** לכל פונקציה הניתנת לבניה במגבלת זמן עצמית $t$:N→N, קיימת שפה $A$ שניתנת להכרעה ב- O($t(n)$) זמן אך לא ניתנת להכרעה בזמן $o(\frac{t(n)}{\log(t(n))})$

**9.11 משפט היררכיית הזמן**: לכל שתי פונקציות $t_1$,$t_2$:N→N, כך ש: $t_1(n)= o(\frac{t_2(n)}{\log(t_2(n))})$ ו- $t_2$ ניתנת לבניה במגבלת זמן עצמית, מתקיים: $TIME(t_1(n)) \subsetneq TIME(t_2(n))$.

**9.12** לכל 2 מספרים ממשיים $0 \le \varepsilon_1 < \varepsilon_2$ מתקיים $TIME(n^{\varepsilon_1}) \subset TIME(n^{\varepsilon_2})$

- (9.13) $\mathbf{P \subsetneq EXPTIME}$

# נושאים מתקדמים – אלגוריתמי קירוב

**אלגוריתם קירוב**: לא מבטיח הגעה לפתרון הכי טוב, אך כן מתחייב ליחס קירוב $\rho$ כלשהו.

**יחס קירוב**: אלגוריתם קרוב $A$ הוא בעל יחס קרוב $\rho$ $(\rho \geq 1)$ אם עבור כל קלט היחס בין העלות C של הפתרון שמפיק האלגוריתם A לעלות הפתרון האופטימלי *C מקיים:

1. עבור בעיות **מקסימום** - $\frac{C^*}{C} \leq \rho$
2. עבור בעיות **מינימום** – $\frac{C}{C^*} \leq \rho$

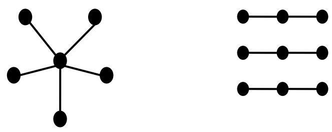
***APROX-VERTEX_COVER(G)***

1. *$C \leftarrow \emptyset$*
2. *$E' \leftarrow E$*
3. ***while** $E' \neq \emptyset$*
4. *    **do** let (u,v) be an arbitary edge of E'*
5. *        $C \leftarrow C \cup \{u,v\}$*
6. *        remove from E' every edge incident on either u or v*
7. ***return** C*

**MIN-VERTEX-COVER - כיסוי מינימאלי בקודקודים** ( ***minimal vertex cover***) הוא קבוצה $C$ של קודקודים ב $G$ המהווה כיסוי בקודקודים של $G$, ואין לה קודקודים הניתנים להסרה. **בעיה זו שייכת ל *NP***

להלן אלגוריתם קירוב לבעיה זו שמתחייב ל**יחס קירוב $\rho = 2$**:

הוכחה ליחס הקירוב: נסמן ב-$A$ את קבוצת הקשתות *(u,v)* שנבחרה בשורה 4. קבוצה זו אינה מכילה קשתות שלהן צמתים משותפים, לפיכך $|C|=2|A|$. עתה, צמתי הפתרון האופטימלי $C^*$ מכסים את קשתות $A$ ולכן מכילים לפחות צומת אחד לכל קשת ומכאן $|A| \leq |C^*|$. ובסך הכל נקבל $|C| \leq 2|C^*|$



דוגמאות לגרפים כך שבכל הרצה יתקבל יחס קירוב 2:

**הערה:** אם היינו בוחרים כל פעם **באחד מצמתי הקשת**, כלומר בשורה 5 היה כתוב $C \leftarrow C \cup \{u\}$ אז לכל ערך קבוע $k$ קיים גרף שאלגוריתם עשוי שלא לספק $k$-קירוב. הוכחה: נתבונן בגרף "כוכב" המכיל $k+2$ צמתים ו- $k+1$ קשתות המחברות את אחד הצמתים $(v)$ לשאר הצמתים. הפתרון האופטימלי יכיל צומת יחיד - $v$. ואילו האלגוריתם הנתון עשוי לבחור את שאר $k+1$ הצמתים.

**בעיית האריזה בקופסאות זהות (*Bin Packing*) :** למלא קופסאות בעלות נפח זהה, בחפצים שונים, ששונים בנפחם, כך שכל החפצים יאוחסנו בקופסאות, ונשתמש בכמה שפחות קופסאות.

- מניחים שיש לנו אינסוף קופסאות. נפח קופסא מוגדר להיות 1. החפצים מוגדרים ע"י נפחם ביחס לנפח קופסא
- קלט: קבוצה של $n$ מספרים $A = \{a_1, \dots, a_n\}$ כאשר $0 \leq a_i \leq 1$ (הנפחים ביחס לקופסא)
- פלט: חלוקה של $A$ ל-$k$ תת קבוצות זרות ($k$ קופסאות) כך שנפח החפצים בקופסא לא חורג מנפחה + ***k* מינימלי**

***גרסת הקירוב = first fit*** נשים את הראשון ב $K_1$, כעת את השני ננסה להכניס גם ל $K_1$ – אם לא נצליח נעבור ל $K_2$, כעת את השלישי ננסה ל $K_1, K_2, K_3$ וכן הלאה.

***אלגוריתם זה מבטיח יחס קירוב 2!*** הרעיון – לא יכול להיות שיש 2 תיבות שיחס גדול מ 0.5 כי אם כן היינו מכניסים אותם אחת לשניה. נסמן $S = \sum_{i=1}^{n} s_i$. נעגל למעלה את הסכום [S] ≥ *C (מספר התיבות הנדרשות בפתרון האופטימלי) האלגוריתם החמדני מותיר לכל היותר תיבה אחת מלאה עד פחות מחציה. [2S] ≥ C (מספר התיבות הנדרשות ע"י האלגוריתם החמדני). *C ≤ 2C

***APROX-Makespan-Scheduling($(S_i)$)***

1. Order the jobs arbitrarily.
2. Until the job list is empty, move the next job in the list to the end of the shortest machine queue.

**בעיית תזמון משימות (*Makespan Scheduling*) :** נתונה קבוצה של n משימות שצריכות להתבצע ב m מכונות ואורך כל משימה i הוא $s_i$. פלט: תזמון המשימות במכונות, שימזער את זמן הסיום של כל המשימות.

הוכחה שזה *NPC* – ברדוקציה מבעיית *TSP* – *m=1*, הסוכן הנוסע = מכונה, הערים = העבודות שצריכות להתבצע.

**גרסת הקירוב = *Graham's List Scheduling***: הרעיון – נשבץ כל פעם במכונה שנכון לעכשיו תסיים הכי מוקדם

***אלגוריתם זה מבטיח יחס קירוב 2!*** נסמן ב-i את המשימה אשר הסתיימה אחרונה. וב-$t_i$ את זמן התחלתה.

$$ALG = s_i + t_i \leq s_i + \frac{\left(\sum_{j=1}^{n} s_j\right) - s_i}{m} = \frac{1}{m}\sum_{j=1}^{n} s_j + (1 - 1/m)s_i$$

מכיוון שמתקיים: $s_i \leq OPT$ וכן $\frac{1}{m}\sum_{j=1}^{n} s_j \leq OPT$

נקבל: $ALG \leq (2 - 1/m)OPT$

צביקה פרלמוטר, 2017 ©

**בעיית כיסוי הקבוצות (*Set Cover*) :**

**רעיון אלגוריתם הקירוב** – חמדני. נבחר את הכי גדול שלא כוסה בכל שלב.

יחס הקירוב המתקבל – תלוי בקלט, בגודל הקבוצה הגדולה ביותר.

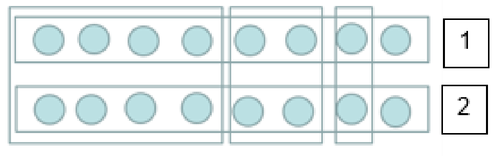
ניתן לחשב ע"י הטור ההרמוני עם n=$|\widehat{S}|$ כאשר $\hat{S}$ היא הקבוצה הגדולה ביותר ב S ו $H(n) = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$

שהוא סדר גודל של ln(n).

***APROX-SET-COVER(S,(S_i))***

```
U←S
C←∅
while U≠∅
    select an Si that maximize |Si∩U|
        U←U-Si
        C←C∪{Si}
return C
```

***תרגיל:*** נניח שנרצה לכסות 16 איברים כאשר 2 תתי הקבוצות הם:




**א)** מהם יחסי הקירוב האפשריים ע"י אלגוריתם הקירוב? ניתן לראות בקלות כי הפתרון האופטימלי הוא 2. אלגוריתם הקירוב יכול גם לתת פתרון 2 – ואז יחס הקירוב הוא 1. ייתכן גם שהפתרונות של אלגוריתם הקירוב יהיו 2,3,4,5 ואז יחס הקירוב הוא 1-2.5

**ב)** במקרה הכללי שבו $n=2^k$ הפתרון הטוב ביותר הוא 2 (יחס קירוב 1) והגרוע ביותר יבחרו הקבוצות לפי הסדר הבא: $\frac{n}{2}, \frac{n}{4}, \frac{n}{8}, \ldots, 1, 1 = O(\lg n)$

**בעיית תרמיל הגב (*Knapsack problem*)**

נתונה קבוצת חפצים $A = (a_1, \ldots, a_n)$ ולכל חפץ יש ערך U ונפח V. לגנב תרמיל בנפח W , האם ניתן לקחת חפצים בשווי סה"כ שגדול מ k?

לבעיה זו בגרסת השברים – יש פתרון פולינומיאלי ע"י ערך סגולי, בגרסת השלמים – יש פתרון ע"י תכנות דינאמי

אך מצד שני אם מדובר על מספרים לא שלמים ולא שברים – NPC ברדוקציה מ PAR.

**__אלגוריתם קירוב:__** נראה כי שילוב של 2 אלגוריתמים בעלי יחס קירוב לא טוב, יכול להוביל ליחס קירוב טוב:

__אלגוריתם א__: בחר פריט בודד שערכו מקסימלי. – לבדו לא טוב, יכול להשיג גרוע מ k

__אלגוריתם ב__: (בדומה לגרסה השברית) מיין את הפריטים לפי הערך הסגולי והוסף פריטים לפי הסדר כל עוד לא עברנו את המכסה . – לבדו לא טוב, יכול להשיג גרוע מ k

__אלגוריתם ג__: הרץ את אלגוריתם א' וב' ובחר את הטוב מבין השניים. **לאלגוריתם זה יחס קירוב 2.**

**__בעיית הסוכן הנוסע:__** בהינתן קבוצת ערים ומחיר הנסיעה בין כל שתי ערים, בעיית הסוכן הנוסע (***traveling salesman problem***) היא למצוא את הדרך הזולה ביותר לבקר בכל הערים ולחזור לעיר המוצא.

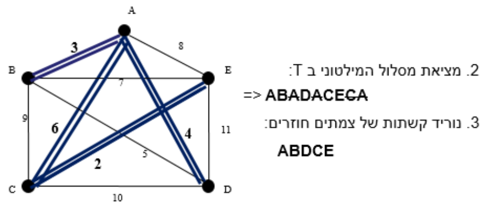
- __קלט__: גרף לא מכוון מלא *G=(V,E)* עם מחירים אי שליליים על הקשתות.
- __פלט__: מעגל המילטוני ב *G* בעל עלות מינימאלית
- **__לבעיה זו אין אלגוריתם קירוב בקירוב כלשהו!__** – רעיון ההוכחה = אם היה אלג' קירוב אז היינו מצליחים לפתור את בעיית *HAMPATH* ולהוכיח ש *P=NP*

**__בעיית הסוכן הנוסע המטרית__**: מקרה מיוחד של בעיית הסוכן הנוסע, שבו מחירי הקשתות מקיימים את אי שוויון המשולש: לכל זוג קודקודים $u, v \epsilon V$ מוגדר מחיר הקשת המחברת אותם : $c(u,v) \geq 0$ ולכל $u, v, w \epsilon V$ מתקיים : $c(u,w) \leq c(u,v) + c(v,w)$

במקרה זה ניתן לקבל **יחס קירוב 2** ואפילו **1.5**!

**__להלן הוכחה ליחס קירוב 2:__**

1. מצא עץ פורש מינימלי T (ע"י האלגוריתם של פרים/קרוסקל 🡸 ליניארי) והכפל את קשתותיו
2. מצא מעגל אויילר EC בגרף T – מסלול מעגלי שעובר בכל הקשת בדיוק פעם אחת (מוכר מבעיית הגשרים). ידוע שאם (ורק אם) כל הצמתים הם מדרגה זוגית אז קיים מעגל אויילר.
3. **3.** קצר את EC למעגל המילטון A על ידי דילוג על קשתות המובילות לצמתים שבקרנו בהם.

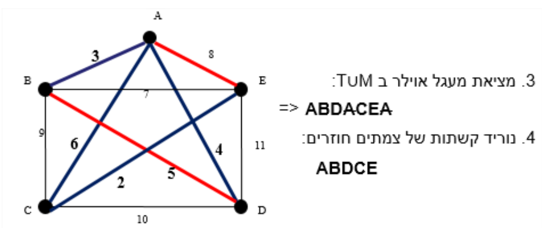


1. מציאת עפ"מ והכפלת הקשתות

שיעור הקירוב הוא: $2 \cdot OPT \leq_{\text{מעגל המילטון חסר קשת הוא עץ פורש}} 2w(T) = w(EC) \leq_{\text{אי שיוויון המשולש}} A$

**__להלן אלגוריתם עם יחס קירוב 1.5:__** (Christofides)

1. מצא עץ פורש מינימלי T
2. מצא זיווג מושלם מינימלי M בין הצמתים שדרגתם אי-זוגית ב-T
3. מצא מעגל אויילר EC בגרף TUM
4. קצר את EC למעגל המילטון A על ידי דילוג על קשתות המובילות לצמתים שבקרנו בהם.



1. מציאת עפ"מ והכפלת הקשתות
2. מציאת זיווג מושלם מינימלי M

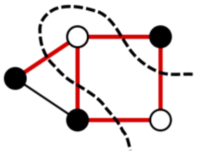
**<u>הוכחת שיעור הקירוב:</u>** ראשית נוכיח כי $w(M)\le 0.5\cdot OPT$ נקצר את OPT למעגל C העובר רק דרך הצמתים מדרגות אי-זוגיות ב-T. מתוך אי שוויון המשולש מתקיים: $w(C)\le OPT$ נצבע לסירוגין את קשתות C באדום וירוק ונקבל שני זיווגים $C_1, C_2$ ומתקיים: $w(C)=w(C_1)+w(C_2)\ge 2w(M)$ עתה נקבל: $A\le w(EC)=w(T)+w(M)\le 1.5\cdot OPT$

**<u>תרגיל:</u>** נתון גרף עם משקלים. כיצד ניתן בזמן פולינומי להופכו לגרף המקיים את אי-שוויון המשולש, כך שהמעגל ההמילטוני הזול ביותר יישאר כזה? תשובה: להוסיף לכל הקשתות את משקל הקשת היקרה ביותר

***<u>חתך בגרף:</u>*** *קבוצה של קשתות, C⊆E, עבורה קיימת קבוצת קודקודים S לא ריקה, $S\subsetneq V$, כך ש: C היא קבוצת כל הקשתות בגרף המחברות קודקוד מ S לקודקוד ב V\S )*

***<u>גודל של חתך</u>*** *: מספר הקשתות בחתך.*

***<u>בעיית החתך המקסימאלי MAX-CUT</u>:*** בעיה זו שואלת, בהינתן גרף לא מכוון *G=(V,E)* ש למצוא את החתך בעל הגודל המקסימאלי.



# <u>נושאים מתקדמים – אלגוריתמים הסתברותיים</u>

**10.3 – <u>מכונת טיורינג הסתברותית:</u>** מכונת טיורינג הסתברותית, *M*, היא סוג של מטל"ד בה כל צעד לא דטרמיניסטי נקרא *צעד **הטלת מטבע*** ויש לו שתי תוצאות חוקיות לצעד הבא .

- לכל ענף של החישוב, *b*, של מכונת טיורינג הסתברותית *M*, על קלט *w*, כאשר מספר הטלות המטבע לאורך חישוב הענף *b*, הוא *k* נגדיר את ההסתברות של הענף *b* כך : $\mathbf{Pr}[\boldsymbol{b}]=2^{-k}$.
- נגדיר את ההסתברות שמ"ט הסתברותית *M*, תעצור במצב מקבל על קלט *w* להיות סכום ההסתברויות של הענפים המקבלים של החישוב ההסתברותי של *M*: $\mathbf{Pr}[\boldsymbol{w}\text{ מקבלת את }\boldsymbol{M}]=1-\Pr[w\text{ דוחה את }M]=\sum_{b\text{ ענף מקבל}}\Pr[b]$
- **M מ"ט הסתברותית מזהה את השפה A עם הסתברות ε לשגיאה** אם : ההסתברות לקבלת תוצאה שגויה, ע"י הרצת *M* על *w*, קטנה מ-$\frac{1}{2}$. $0\le\boldsymbol{\varepsilon}<\frac{1}{2}$ ובאופן פורמאלי:

  עבור *w∈A*: $\mathbf{Pr}[\boldsymbol{w}\text{ מקבלת את }\boldsymbol{M}]\ge 1-\boldsymbol{\varepsilon}$

  ועבור *w∉A*: $\mathbf{Pr}[\boldsymbol{w}\text{ דוחה את }\boldsymbol{M}]\ge 1-\boldsymbol{\varepsilon}$
- מדידת הזמן או המקום של מ"ט הסתברותיות תהיה כמו למלט"ד- לפי הענף המייצג את זמן/מקום החישוב של **המקרה הגרוע** ביותר לכל קלט

**10.5 – <u>משפט ההגברה:</u>** יהי 0<ε<½. לכל פולינום ***poly(n)*** ולכל מ"ט הסתברותית *$M_1$*, שרצה בזמן פולינומיאלי, ובהסתברות ***ε*** לשגיאה, יש מ"ט הסתברותית *$M_2$*, מקבילה שרצה בזמן פולינומאלי ובהסתברות $\boldsymbol{2^{-poly(n)}}$ לשגיאה.

## <u>אלגוריתם אקראי לבדיקת ראשוניות של מספר</u>

- האלגוריתם הנאיבי לבדיקת ראשוניות ע"י חיפוש מחלקים של מספר הוא אקספוננציאלי בגודל הקלט

נראה כעת כיצד ניתן לשפר אותו ע"י שימוש באלגוריתם הסתברותי ב *O(n)*. לפני כן נציג מספר הגדרות:

**מספרים שווים מודולו *p*:** לכל מספר *p>1* : נאמר ששני מספרים **שווים מודולו *p***, אם ההפרש ביניהם הוא *p* בדיוק. כותבים זאת : $x\equiv y(mod\ p)$

- כל מספר שלם *x*, שווה מודולו *p* לאיבר בקבוצה : $Z_p=\{0,\dots,p-1\}$ למען הנוחות, נגדיר : $Z_p^+=\{1,\dots,p-1\}$

**<u>10.6 משפט פרמה הקטן:</u>** אם *p* ראשוני אז לכל *a* טבעי שקטן ממנו מתקיים(מבחן פרמה): $a^p\equiv a(mod\ p)$ . אם p אינו ראשוני אז קיים $a$ עבורו $a^p\not\equiv a(mod\ p)$ . דוגמה : p=5 ראשוני ומחלקיו $0^5=0,\ 1^5=1,\ 2^5=32\equiv 2,\ 3^5=243\equiv 3,\ 4^5=1024\equiv 4$

לעומת זאת עבור *p=6*: $2^6=64\equiv 4\not\equiv 2$

<u>מבחן הראשוניות של מילר ורבין:</u> עבור מספר p נגריל מספר a בין 1 לבין p-1 ונבדוק אם הוא עובר את מבחן פרמה.
אם לא נכריז "פריק" (100%) ודאות) ונקרא ל a "עד" לפריקותו של p. אחרת **נמשיך לבדוק כך עוד k פעמים**.
אם עדיין לא הגענו לעד נכריז "ראשוני" בשיעור $2^{-k}$.

## <u>מחלקות אלגוריתמיות הסתברויות</u>

**10.10 – <u>המחלקה RP</u> :** מחלקת השפות שמזוהות ע"י מ"ט הסתברותית שרצה בזמן פולונומיאלי כך ש-קלט ששייך לשפה יתקבל בהסתברות של ½ לפחות, קלט שאינו שייך לשפה יידחה בהסתברות 1.
לדוגמא: COMPOSITES∈RP

**המחלקה coRP :** מחלקת השפות שיש להן אלגוריתם הסתברותי בעל זמן ריצה פולינומיאלי המקבל כל קלט בשפה בהסתברות 1, ודוחה כל קלט שאינו בשפה בהסתברות ½ לפחות.
לדוגמה: PRIMES∈coRP

**10.4 – *BPP*** - מחלקת השפות המזוהות ע"י מ"ט הסתברותית שרצה בזמן פולינומיאלי בהסתברות 1/3 לשגיאה
לדוגמה: PRIMES∈BPP (10.9) וגם $EQ_{RDBP}$∈BPP (10.13)

טענות שמוכחות במדריך (עמ' 141): 1. **RP⊆BPP** 2. **coRP⊆BPP** 3. **P⊆RP⊆NP**

**תרגילים:**

1. **הוכח / הפרך** - אם תמצא שפה B במחלקה RP, כך שהמשלימה שלה לא שייכת ל-RP אז P≠NP

**תשובה:** נכון! מאחר ומתקיים P⊆RP⊆NP לפיכך אם P=NP אז גם P=RP ומאחר ש-P סגורה להשלמה גם RP היא כזו ולכן לא יכולה להימצא שפה B כפי שנתון..

2. נגדיר את המחלקה RL בדומה למחלקה RP כאשר זו מכילה את השפות המוכרעות ע"י מ"ט הסתברותית העושה שימוש בסרט עבודה במקום לוגריתמי. **הוכיחו:** RL⊆SPACE($\log^2 n$)

**פתרון:** RL⊆NL⊆SPACE($\log^2 n$)

3. נגדיר את השפה הבאה: $DROP-MIDDLE = \{w|\ w = uv, |u| = |v|, \exists\sigma \in \Sigma, u\sigma v \in D\}$

הוכיחו: אם $D \in RP$ אז $DROP-MIDDLE \in RP$

**פתרון:** תהי M מ"ט הסתברותית המכריעה את D בזמן פולי' עם הסתברות לשגיאה חד-כיוונית ½.
נתאר מ"ט M′ המכריעה את DROP-MIDDLE:
"עבור קלט w:
1. אם |w| אי-זוגי, דחה.
2. נסמן w=uv, |u|=|v|.
3. עבור כל σ∈∑, בצע:
   3.1 בדוק האם M מקבלת את uσv אם כן, קבל.
4. דחה"

סיבוכיות זמן: פולינומיאלית (בדקו) נכונות: אם w∈DROP-MIDDLE ההסתברות לדחיה קטנה מ- $1/2^k$ כאשר k הוא מספר האותיות σ כך ש- uσv∈D אם w∉DROP-MIDDLE אם |w| אי-זוגי w נדחית בהתחלה. אחרת w אינה מתקבלת באף איטרציה ולפיכך נדחית.

# רדוקציה עצמית

**רדוקציה עצמית:** היא רדוקציה מבעיית החיפוש / **אופטימיזציה** לבעיית ה**הכרעה**. כלומר, בהינתן קופסה שחורה A שמסוגלת לקבוע האם קיים פתרון לבעיה או לא, נבנה אלגוריתם העושה בו שימוש ובונה את הפתרון בזמן פולינומי.
למה זה טוב? אם קיימת רדוקציה עצמית לבעיה מסוימת, הדבר מבטיח לנו שאם קיים **אלגוריתם פולינומי לבעיית ההכרעה**, אזי קיים גם **אלגוריתם פולינומי לבעיית החיפוש**.

**דוגמא להוכחה:** כתיבת אלג' לבעיית האופטימיזציה שעושה שימוש בבעיית ההכרעה מס' פול" של פעמים