חישוביות

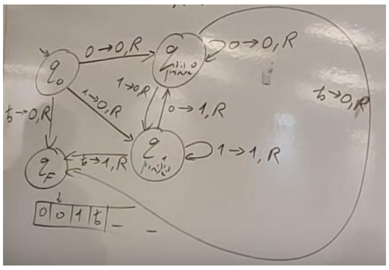
מהן המגבלות של המחשב?
האם קיימות בעיות שמחשב לא מסוגל לחשב?
אלגוריתם: אלגוריתם A הינו תוכנית מחשב המקבלת קלט x ומחזירה פלט $A(x)$ כך $x, A(x) \in \Sigma^*$.
כמות **הפונקציות** היא $\aleph \geq |\Sigma^*|^{|\Sigma^*|}$ (פלט בחזקת קלט)
כמות **האלגורתמים** היא $\aleph_0$ (Σ^*)
ולכן יש פונקציות **שלא** ניתנות לחישוב (שיקולי ספירה)

שפה: קבוצה (לא דווקא סופית) של מילים מעל הא"ב Σ

מכונת טיורינג: (מודל שקול בכוחו למחשב החזק ביותר ופשוט מתמטית)
טיורינג היא מודל אשר מורכב **מהשבעייה** הבאה: $(\boldsymbol{Q}, \boldsymbol{F}, q_0, \boldsymbol{\delta}, b, \Gamma, \Sigma)$
$\boldsymbol{Q}$ היא קבוצת מצבים **סופית**
$\boldsymbol{F} \subseteq Q$ קבוצת מצבי עצירה
Σ א"ב הקלט – קבוצה סופית
Γ א"ב הסרט – קבוצה סופית כאשר $b \in \Gamma$ ו- $\Sigma \subset \Gamma$
b תו רווח
ישנו סרט זיכרון **אינסופי** מצד ימין לקריאה/כתיבה כאשר יש ראש/קורא כותב אשר "רואה" בכל רגע נתון תא **אחד** בסרט הזיכרון מתוך א"ב הסרט.
פונקציית המעברים δ $\delta: (Q \backslash F \times \Gamma) \to (Q \times \Gamma \times (L, S, R))$ - נקרא גם צעד חישוב
כלומר ממצב **לא** מקבל ומראש קורא/כותב
עוברים למצב חדש (יכול להיות מקבל), כותבים לאותו מקום שהראש קורא/כותב היה וזזים ימינה או שמאלה או נשארים במקום.

מצב ההתחלתי של מ"ט הוא שהקלט כתוב על סרטון הזיכרון החל מהתא השמאלי ביותר ולאחר מכן יש רק b. המצב ההתחלתי הוא q_0 וראש קורא כותב מצביע בסרט על התא השמאלי ביותר.

מצבי עצירה הינם מצבים שאם מגיעים אליו החישוב מסתיים והמכונה מוציאה את הפלט שלה
דוגמא למ"ט המחשב את הפונקצייה $f: \{0,1\}^* \to \{0,1\}^*$ המוגדרת ע"י $f(x) = 0 \cdot x$



הסבר: בהתחלה עבור כל תו כותבים 0 ואז עוברים למצב שזוכר איזה תו נמצא על הראש קורא כותב, $q_{remember\ \sigma}$,
לאחר מכן ממצב $q_{remember\ \sigma}$ אם רואים $\sigma_2 \in \Sigma$ אז עוברים למצב $q_{remember\ \sigma_2}$ וכותבים σ, עבור כל מצב שזוכר נעבור למצב מקבל אם רואים את התו b

שפה של מ"ט: עבור מ"ט M נסמן ב-$L(M)$ את שפת המכונה המוגדרת ע"י $L(M) \coloneqq \{\omega \in \Sigma^* |\ M(\omega) = 1\}$ כאשר ההגדרה שהיא שמכונת טיורינג פולטת רק 0 או 1 בלי תלות במה שכתוב על הסרט, לחלופין אפשר להסתכל על זה ששפה של המכונה M היא כל המילים שהמכונה עוצרת ומקבלת אותם (יכול להיות שלא תקבל או לא תעצור עליהם)

<u>סוגי מכונות טיורינג:</u>

1. מכונה המחשב פונקצייה – מקבל קלט x ומוציאה פלט y כאשר פלט מוגדר להיות כל מה שכתוב על הסרט הזיכרון מהמיקום השמאלי ביותר ועד מיקום הראש **<u>לא כולל</u>** בזמן שהמכונה נכנס למצב עצירה $q \in F$.
2. **מכונת טיורינג המכריעה שפות** – למכונה זו יש 2 מצבי עצירה q_{acc} ו- q_{rej} (קבלה ודחייה)

נאמר שמ"ט M **<u>מקבלת</u>** שפה L אם $L = L(M)$

לכל $w \in \Sigma^*$

$w \in L \Longrightarrow M(w) = 1$

$w \notin L \Longrightarrow M(w) = \{\text{לא מקבלת, לא עוצרת}\}$

נאמר שמ"ט M **<u>מכריעה</u>** שפה L אם $L = L(M)$ ודוחה לכל $w \notin L$

לכל $w \in \Sigma^*$

$w \in L \Leftrightarrow M(w) = 1$

אם $w \notin L$ אז $M(w)$ **עוצרת** אבל לא מקבלת.

הערה: שפה L היא **כריעה** אם קיימת מ"ט M אשר מכריעה אותה.

הערה: אם שפה מכריעה שפה אז היא בהכרח מקבלת גם.

הערה: $L(M) = \{x \mid M \; accepts \; x\}$

3. מכונה טיורינג אי"ד – יכולה לנחשב ויש לה מספר מסלולי חישוב (נבנים בצורה של עץ) היא שקולה בכוחה למכונה הרגילה אבל **<u>אינה שקולה</u>** בזמן הריצה שלה.

היא מוגדרת כמו המודל הרגיל פרט לפונקציית המעברים שלה:

$\delta: \; Q \times \Gamma \rightarrow (Q \times \Gamma \times \{\text{L}, \text{S}, \text{R}\})^2$

המכונה "מנחשת" את אחת מ-2 האופיציות הנתונות לה.

כאשר $L(N) \coloneqq \{x \in \Sigma^* \mid \exists \; path \; that \; excepts \; x\}$

<u>משפט:</u> לכל מ"ט אי"ד M קיימת מ"ט דטרמנסטית M' כך ש-$L(M) = L(M')$

<u>הוכחה:</u> הרעיון הוא שמכונה M' תסרוק את עץ החישוב ותבדוק אם יש ענף מקבל, היא תסרוק זאת בעזרת אלגוריתם BFS מאורך 1, ואז אורך 2 וכן הלאה.

דוגמא למ"ט אי"ד ל- $L \coloneqq \{< M > \mid L(M) \neq \emptyset\}$

כאשר M על $< M >$:

נחש מילה x, הרץ את M על x וענה כמוה.

דוגמא למ"ט **המכריעה** את השפה $L \coloneqq \{a^n b^n c^n \mid n \in \mathbb{N}^+\}$

על קלט $w \in \Sigma^*$:

1. בדוק ש-w מכילה רצף של a ואז b ואז c
2. עבור כל תו a ב-w:

2.1 מחק את a ועבור על הסרט עד למציאת ה-b הראשון. אם לא מצאת – דחה. אחרת, מחק את ה-b הראשון ועבור על הסרט עד למציאת ה-c הראשון, אם לא מצאת – דחה. אחרת, מחר את ה-c הראשון שמצאת וחזור לתחילת הסרט

3. אם מחקנו את כל התווים – קבל, אחרת אם מצאת a חזור ל-2, אחרת דחה.

<u>התזה של צרץ טיורינג</u> – כל מודל כללי וסביר **שקול** למכונת טיורינג (אין הוכחה ולכן זה תזה)

כללי – מודל חזק לפחות כמו המודל הרגיל

סביר – לכל אובייקט בתיאור יש תיאור סופי.

<u>שקילות מודלים:</u>

טענה: מ"ט k סרטית שקולה למ"ט עם סרט אחד.

הוכחה: <u>סרט אחד $\Leftarrow k$ סרטים</u>

בהינתן מ"ט עם סרט אחד נגדיר מ"ט עם k סרטים שקולה כאשר בסרט 1 נבצע את אותם פעולות כמו המכונה הרגיה ובשאר הסרטים לא נשתמש (מקרה פרטי)

<u>סרט אחד $\Rightarrow k$ סרטים</u>

הוכחת הכיוון השני מתבצעת לפי הרעיון הבא, כאשר לשם פשטות נניח כי $k = 2$. בהנתן מכונה עם שני סרטים M, נבנה מכונה M^* בעלת סרט יחיד אשר תבצע סימולציה של הרצת שני הסרטים על גבי סרט יחיד. לשם כך, נצטרך אלפבית עבודה גדול במיוחד אשר בו כל אות מסמלת את שני התווים שנמצאים על הסרט של המכונה M. למשל, נניח שבמכונה הדו-סרטית הסרט הראשון מכיל abc והשני 01 (ושאר התאים ריקים), אזי הסרט של M^* יכיל $(a, 0)(b, 1)(c, \sqcup)$ כאשר כל זוג בתוך סוגריים מהווה "אות יחידה" באלפבית של M^*. בעיה נוספת עמה יש להתמודד, היא שב-M יש שני ראשים שיכולים להיות בשני מקומות שונים.. פתרון לכך מתבצע על-ידי "סימון" מיקום הראשים במכונה הדו-סרטית, על-ידי הוספת תג. למשל, אם בדוגמא לעיל הראש הראשון נמצא בתא הראשון (על ה"a"), והראש השני נמצא בתא השלישי, הריק, אזי במכונה M^* תוכן הסרט יהיה $(a', 0)(b, 1)(c, \sqcup')$. לסיכום, אם Γ הוא האלפבית של M אזי האלפבית של M^* נתון על-ידי

$$\Gamma^* = \Gamma \cup [(\Gamma \cup \Gamma') \times (\Gamma \cup \Gamma')]$$

כאשר $\Gamma' = \{a' \mid a \in \Gamma\}$.

נכתב ע"י צבי מינץ zvimints@gmail.com

הגדרה: מודלים A,B יקראו **שקולים** אם לכל שפה L
יש ל-L מודל מכריע A **אם"ם** יש ל-L מודל מכריע B
ההוכחה מתבצעת באופן הבא:
"לכל שפה L, אם ל-L יש מודל מ"ט ללא <תכונה A> אז יש ל-L מ"ט <תכונה B> כי ..."

קונגפרוראציה (תיאור רגעי $snapshot$):
השלשה $[q,w,i]$ כאשר $q\in Q$ הינו המצב הנוכחי, $w\in\Gamma^*$ הינו תוכן הסרט ו-i מיקום הראש קורא כותב.
נשים לב כי אם המכונה לא עוברת את התא ה-k אז כמות הקונפגוראציות השונות היא

$$|Q|\cdot \overbrace{|\Gamma|^k}^{for\ each\ cell\ |\Gamma|\ and\ there\ at\ most\ k\ cells}\cdot k$$

ולכן אם עברנו את כמות הצעדים הזו, יש **בהכרח** לולאה.
הקונפגוראציה ההתחלתית הינה $(q_0,x,1)$ כאשר x הינו הקלט.
הגדרה נוספת לשפה של מ"ט M:

$$L(M)=\{\,w\in\Sigma^*\mid C_0,C_1,\dots,C_k \text{ קיימת סדרת קונגפוראציות עוקבות}\}$$

עבור קלט w כאשר C_0 הינה קונג'התחלתית ו- C_k הינה קונג' מקבלת (שהמצב הוא מצב מקבל).
כאשר C_i,C_j יקראו קונפיגוראציות **עוקבות** אם קיים מעבר **אחד** המתחיל בC_i ומעביר אותנו ל-C_j
הערה: אם קיימת $C_i=C_j$ עבור $i\neq j$ אז **קיימת לולאה** והיות והמחשב דטרמיניסטי.

מ"ט אוניברסלית U - מ"ט אוניברסלית U מקבלת קלט $<M,x>$ (קידוד של מ"ט ומילה) ומריצה את M על x ועונה כמוה או לא תעצור כמו M.
איך U עובדת? ל-U יש 2 סרטים, סרט הקלט וסרט העבודה, בתחילת העבודה היא תכתוב על הסרט השני את הקונפגוראציה ההתחלתית של M על x
$C_0=(q_0,w,1)$
וכל פעם תסתכל בסרט הקלט מה לעשות. (בקצרה)

קידוד של מכונת טיורינג הינו $<M>:=1^{|Q|}01^{|\Gamma|}01^{|\Sigma|}0<\delta(1,1)>0\dots0<\delta(|Q|,|\Gamma|)00$
כאשר $\delta(p,\alpha)=(q,\beta,c\in\{1,0,2\})\Rightarrow<\delta(p,\alpha)>:=1^p01^\beta01^c$
ולכן אם נתונה מ"ט M ניתן לדבר על $<M>$ ולאמר ש-U (מ"ט אונברסאלית) מסמלץ את ריצה M על קלט x.

מחלקות חישוביות:

$RE:=\{L\subseteq\Sigma^*: L \text{ קיימת מ"ט המקבלת את}\}$
(מקבל כל מילה בשפה אבל יכולה **לא לעצור** על קלט שלא שייך לשפה)

$R:=\{L\subseteq\Sigma^*: L \text{ קיימת מ"ט המכריעה את}\}$
(מקבלת או דוחה **כל** מילה)

$coRE:=\{L\subseteq\Sigma^*|\bar{L}\in RE\}$
(**מקבלת/עוצרת** כל מילה **שלא** שייכת לשפה אבל יכולה לא לעצור על קלט ששייך לשפה)

תכונות סגור של המחלקות R ו-RE
משפט: המחלקה RE סגורה תחת **איחוד** $\cup$, **חיתוך** $\cap$, **שרשור** $\cdot$ **וכוכב קליני** $*$.
הוכחה: סגירות לחיתוך: ביהנתן $L_1,L_2\in RE$ המתקבלות ע"י מ"ט M_1,M_2 נקבל את $L_1\cap L_2$ ע"י מ"ט דו סרטית M אשר בהינתן קלט x תבצע:
תעתיק את x לסרט נוסף ותריץ את M_1 על x בסרט הראשון

אם M_1 דחתה, אזי M תדחה, אם M_1 קיבלה אז M תריץ את M_2 על x בסרט השני ותענה כמוה.
סגירות לאיחוד: נבנה מ"ט אי"ד ל-$L_1 \cup L_2$ כך שבהינתן קלט w המכונה באופן אי דטרמנסטי תבחר אם לעבור למצב התחלתי של M_1 או M_2 ולאחר מכן תריץ את מכונה M_i כאשר $i \in \{1,2\}$ ותענה כמוה.
ניתן גם להריץ 2 מכונות במקביל (צעד צעד - **לא טורי!**) ואם אחת מקבלת אז לעצור ולקבל.
כאשר הרצה במקביל ע"י 2 סרטים – אחד לכל מכונה
סגירות לשרשור: **חסר** (עמוד 56 בספר אוטומטים של בן גוריון)
סגירות לכוכב קליני: **חסר** (עמוד 56 בספר אוטומטים של בן גוריון)

משפט: המחלקה R סגורה תחת **איחוד, חיתוך, משלים, שרשור, הפרש סימטרי וכוכב קליני.**
סגירות למשלים: בהינתן $L \in R$ תהא M מ"ט המכריעה את L. נבנה $\bar{L}$ ע"י כך שנהפוך את המצב המקבל את M למצב דוחה ולהפך.
סגירות לשרשור: רעיון ההוכחה: בהינתן מילה w נעבור על כל החלקות האפשריות של uv ונריץ את M_1 על u ו-M_2 על v ונקבל אם באחת ההרצאות שתיהן קבלו, אחרת דחה.

משפט: $R = RE \cap coRE$
הוכחה: צד ראשון טרוואלי, צד שני בגלל שכל מילה $x \in L$ או $x \in \bar{L}$ אז ניתן להריץ את 2 המכונות במקביל (קיימת מכונה M_1 שמקבלת את $x \in L$ ומכונה M_2 שמקבלת את $x \in \bar{L}$) ולכן בהכרח אחת מהן תעצור קודם, ונחזיר את הפלט שהתקבל.

תזכורת: מכונה M **שמכריעה** את השפה L:
$\forall x \in L$ אזי M עוצרת על q_{acc}
$\forall x \notin L$ אזי M עוצרת על q_{rej}
מכונה M **שמקבלת** את השפה L:
$\forall x \in L$ אזי M עוצרת על q_{acc}
$\forall x \notin L$ אזי M עוצרת על q_{rej} או לא עוצרת (נכנסת ללולאה אינסופית)

רדוקציה בין שפות
הגדרה: יהיו $L_1, L_2 \subseteq \Sigma^*$ נאמר ש- $L_1 \leq L_2$ אם קיימת פונקצייה $f: \Sigma^* \to \Sigma^*$ המקיימת:
1. f **ניתנת לחישוב**
פונקציית f ניתנת לחישוב אם יש מ"ט M אשר בהינתן קלט x המכונה כותבת על הסרט את $f(x)$ ומקבלת (מ"ט לחישוב פונקציות)
דוגמא לפונקציה **שאינה** חשיבה: $f(<M><w>) = \begin{cases} 1, & w \in L(M) \\ 0, & w \notin L(M) \end{cases}$
2. f **מלאה**
f נקראת מלאה אם היא מוגדרת לכל קלט, כלומר לכל קלט יש פלט
3. f **תקפה**
כלומר $\forall x \in \Sigma^*$ עבור $x \in L_1 \Leftrightarrow f(x) \in L_2$

משמעות: L_1 **בדרגת קושי פחותה או שווה** ל-L_2 כלומר L_1 קלה יותר במובן של אם ניתן לקבל (להכריע) את L_2 אז ניתן גם לקבל (להכריע) את L_1

משפט הרדוקציה:
אם $L_1 \leq L_2$ אזי:
1. אם $L_2 \in R$ אזי גם $L_1 \in R$
2. אם $L_2 \in RE$ אזי גם $L_1 \in RE$
3. אם $L_2 \in coRE$ אזי גם $L_1 \in coRE$

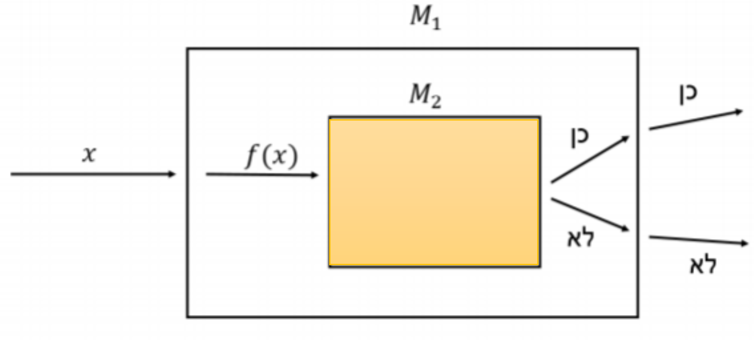
4. אם $L_1 \notin R$ אזי גם $L_2 \notin R$
5. אם $L_1 \notin RE$ אזי גם $L_2 \notin RE$

6. אם $L_1 \notin coRE$ אזי גם $L_2 \notin coRE$

הרעיון של רדוקציה בן שפות הינו:

בשביל להראות ש-$L \notin R$ נניח בשלילה כי $L \in R$ ולכן קיימת מכונה M_2 אשר מכריעה את L.
כדי להגיע לסתירה, נתאר מ"ט M_1 המכריעה שפה $L_{\notin R}$ וזו סתירה לכך ש-$L_{\notin R} \notin R$
כלומר בהינתן קלט x ל-M_1 נשנה אותו מעט וניצור קלט חדש $y = f(x)$ ל-M_2 כך שיתקיים:
$$x \in L_{\notin R} = L(M_1) \Leftrightarrow f(x) \in L = L(M_1)$$

בצורה זו, לאחר השינוי בקלט נוכל להריץ את M_2 ולענות כמוה.



<u>ולכן:</u>

M_1 על קלט x :

1. הרץ את M_f על x ושמור את הפלט ב-y
2. הרץ את M_2 על y וענה כמוהו

<u>לסיכום:</u>

כדי להראות ש-$L \in R$ נתאר מ"ט **המכריעה** את L או שנראה רדוקציה $L \leq L'$ כך ש-$L' \in R$
כדי להראות ש $L \in RE$ נראה מ"ט **המקבלת** את L או שנראה רדוקציה $L \leq L'$ כך ש-$L' \in RE$
כדי להראות ש $L \notin R$ אז נראה רדוקציה $L' \leq L$ כך ש $L' \notin R$ או לפי משפט רייס (בהמשך)
כדי להראות ש $L \notin RE$ אז נראה רדוקציה $L' \leq L$ כך ש $L' \notin RE$ או לפי משפט רייס (בהמשך)
כדי להראות שייכות מומלץ להראות מכונה ולא רדוקציה.

$$L_u \coloneqq \{< M, x >: x \text{ מקבלת את } M\} \in RE$$

הוכחה: נתאר מכונה M_u שמסמלצת את M על קלט $< M, x >$
בהינתן קלט $< M, x >$ ל- M_u
נריץ את M על x ונענה כמוה

נכונות:

$< M, x > \in L_u$ אם"ם M מקבלת את x ולכן $L(M_u) = L_u$

נראה כי $L_u \notin R$

נניח בשלילה כי $L_u \in R$ ולכן קיימת מ"ט M_u אשר מכריעה את L_u
נראה רדוקציה $L_d \leq L_u$ ולכן נקבל כי $L_d \in R$ לפי משפט הרדוקציה, אך $L_d \in RE \backslash R$ ולכן נגיע לסתירה.
נגדיר מכונה M_d אשר מכריעה את L_d:
M_d על קלט $< M >$ כאשר M היא מ"ט:
הרץ את M_u על $< M, < M \gg$ וענה כמוה.
M_d ניתנת לחישוב ונכונות טרוואלית.
מסקנה: $L_u \in RE \backslash R$

$$HP \coloneqq \{< M, x >: x \text{ עוצרת על } M\} \in RE$$

הוכחה: נבנה מ"ט דומה ל-M_u רק עם שינוי קטן. אם M מגיעה למצב דוחה, במקום לדחות המכונה שלנו תקבל. לכן, המכונה תקבל קלט $< M, x >$ ואם M מקבלת או דוחה את x אז M עוצרת על x ולכן נקבל, אחרת, נכנס ללולאה אינסופית.

נראה כי $HP \notin R$
הוכחה:
נראה רדוקציה מ-L_u, אשר אינה ב-R ולכן ממשפט הרודקציה ינבע כי $L \notin R$.
$f(< M, w >) =< M', w' >$
נרצה להראות ש- M **מקבלת** את $w \Leftrightarrow M'$ **עוצרת** על w'
כאשר M' על קלט x : (קלט אחר)
הרץ את M על w
אם M דחתה – היכנס ללולאה אינסופית
אם M קיבלה – קבל
$w' = \epsilon$
נשים לב כי f היא פונקציה חשיבה היות והיא פולטת קידוד של מכונה והראנו בשיעור שפליטה של קידוד מכונה הוא חשיב, בנוסף היא מלאה כי לכל קידוד יש פלט (קידוד חדש)
בנוסף, f תקופה היות ו- M' עוצרת על w אם"ם M מקבלת את w
אבל $L_u \notin R$ ולכן הגענו לסתירה.

מסקנה: $HP \in RE \backslash R$

טריק לרדוקציה מ-$\overline{HP}$
נגדיר $HP_{ALL} \coloneqq \{< M >: M \text{ halts on all inputs}\}$
נראה רדוקציה $\overline{HP} \leq HP_{ALL}$ ולכן ממשפט הרדוקציה $HP_{ALL} \notin RE$
נגדיר את הרדוקציה f ע"י
$f(< M, w >) =< M' >$
כאשר M' על קלט x :
הרץ את M על w למשך $|x|$ צעדים
אם M **עצרה** את w אזי אכנס ללולאה אינסופית
אם M **לא עצרה** – קבל

f תקפה: נשים לב כי M **לא עוצרת** על w אם"ם M' **עוצרת על כל קלט**
(אם M עוצרת על w אז היא עושה זאת לאחר k צעדים, ומכאן עבור כל קלט באורך קטן מk M לא תספיק לעצור ולכן M' תקבל, אבל לכל קלט גדול יותר אז היא תספיק לעצור ולכן M' תכנס ללולאה אינסופית ולכן היא לא עוצרת על כל קלט)

$$L_d \coloneqq \{< M > \mid < M > \in L(M)\} \in RE$$

שפת האלכסון – שפת כל המכונות M אשר מקבלת את הקלט של עצמן $< M >$
הוכחה:
נראה מ"ט M_d אשר מקבלת את L_d באופן הבא:
M_d על קלט $< M >$ אשר M היא מ"ט :
הרץ את M על $< M >$ וענה כמוה
הפונקציה ניתנת לחישוב (ראינו איך לבנות מ"ט אונברסאלית)
נשים לב כי $< M > \in L_d$ אזי M מקבלת את $< M >$ ולכן M_d גם תקבל
$< M > \notin L_d$ אזי M לא מקבלת או לא עוצרת את $< M >$ ולכן M_d תתנהג כמוהה

נוכיח כי $L_d \notin R$: (שיטת האלכסון)
נניח בשלילה כי L_d כריעה ולכן קיימת מ"ט M_d כך ש- $L(M_d) = L_d$
נגדיר את המכונה D הבאה:
D על קלט $< M >$ כאשר M היא מ"ט :
הרץ את M_d על $< M >$ ועונה הפוך.
D היא מ"ט כריעה לפני ההנחה ש-M_d היא מ"ט כריעה.
נבדוק האם $< D >$ שייך לשפה של L_d

1. $L_d \ni< D >$ **ולכן** M_d מקבלת את D **ולכן** D על קלט $< D >$ דוחה ולכן D לא מקבלת את הקידוד של עצמה ולכן $< D >\notin L_d$ – סתירה.

2. $< D >\notin L_d$ **ולכן** M_d לא מקבלת את D **ולכן** D על קלט $< D >$ מקבלת ולכן $< D >\in L_d$ – סתירה

טענה: $\overline{L_d} \notin RE$ כי אם היא כן הייתה אז נקבל כי $L_d \in R$
מסקנה: אם מצאנו שפה אשר נמצאת ב-$RE \backslash R$ אז המשלימה שלה לא ב-RE אלא ב-$coRE$

דוגמא נתונה השפה $L_{dd} \coloneqq \{< M > : M\ accepts < M >\ \boldsymbol{not}\ in\ even\ number\ of\ steps\}$
הוכיחו כי $L_{dd} \notin RE$
נוכיח באמצעות שיטת האלכסון
נניח בשלילה כי $L_{dd} \in RE$ ולכן קיימת מ"ט $M_{L_{dd}}$ אשר מקבלת את L_{dd}
לפי סעיף א' (שלכל מכונה יש מכונה אחרת שמקבלת במספר צעדים זוגי)
קיימת $M'_{L_{dd}}$ המקבלת את L_{dd} ומקבלת כל מילה בשפה במס' צעדים זוגי
$< M'_{L_{dd}} > \in L_{dd}$ ולכן $M_{L_{dd}}$ עונה כן על $< M'_{L_{dd}} >$
ולכן $M'_{L_{dd}}$ עונה כן על $< M'_{L_{dd}} >$ (מכונה שקולה)
לפי הגדרת $M'_{L_{dd}}$ קיבלה את $< M'_{L_{dd}} >$ במספר צעדים זוגי
ולכן $< M'_{L_{dd}} >\notin L_{dd}$
$< M'_{L_{dd}} > \notin L_{dd}$ ולכן $M_{L_{dd}}$ לא מקבלת את $< M'_{L_{dd}} >$
ולכן $M'_{L_{dd}}$ לא מקבלת את $< M'_{L_{dd}} >$ (מכונה שקולה)
ולכן $< M'_{L_{dd}} >\in L_{dd}$

$$L_\emptyset \coloneqq \{< M >: L(M) = \emptyset\} \in coRE$$

1. נראה כי
$$\overline{L_\emptyset} = L_{\overline{\emptyset}} \coloneqq \{< M > | L(M) \neq \emptyset\} \in RE$$
ולכן $L_\emptyset \in coRE$
הוכחה: נראה מ"ט $M_{\overline{\emptyset}}$:
יהא $x_1, x_2, \ldots$ הסדר הלקסקוגרפי של כל המילים מעל Σ^*
$M_{\overline{\emptyset}}$ על קלט $< M >$ כאשר M היא מ"ט:
עבור i החל מ-1 עד ∞ בצע:
עבור j מ-1 עד i בצע:
הרץ את M על x_j למשך i צעדים ובדוק אם M קיבלה
קבל אם M קיבלה.
נכונות: אם $< M >\in L_{\overline{\emptyset}}$ אזי קיימת $w \in \Sigma^*$ כך ש-M מקבלת את w כי השפה $L(M)$ אינה ריקה)
ולכן בהכרח **בהרצה מבוקרת** נמצא את w ונקבל.
אם $< M >\notin L_{\overline{\emptyset}}$ אזי השפה של M היא ריקה ולכן המכונה תמשיך לרוץ עד אינסוף.

2. נראה כי $L_\emptyset \notin RE$ ולכן $L_\emptyset \in coRE \backslash RE$
הוכחה: נראה ע"י רדוקציה $\overline{L_u} \leq L_\emptyset$ ולכן בגלל ש$\overline{L_u} \notin RE$ נקבל ממשפט הרדוקציה כי $L_\emptyset \notin RE$
נגדיר $f(< M, w >) =< M' >$
כך ש-M לא מקבל את w אם"ם $L(M') = \emptyset$
M' על קלט x :
הרץ את M על w וענה כמוהו
(ניתן גם לעשות מ-$\overline{L_d} \coloneqq \{< M >: < M >\notin L(M)\} \notin RE$) ע"י $f(< M >) =< M' >$ כאשר M′ש
על קלט x היא תריץ את M על $< M' >$ ותענה כמוה)

$$L_{\Sigma^*} \coloneqq \{< M >: L(M) = \Sigma^*\} \in \overline{RE \cup coRE}$$

הוכחה:
תזכורת: $L_u \coloneqq \{< M, x >: M \text{ מקבלת את } x\} \in RE \notin coRE$

נראה רדוקציה $L_u \leq L_{\Sigma^*}$ ולכן $L_{\Sigma^*} \notin coRE$

$$f(<M, x>) = <M'>$$

כך ש-M' על קלט w:

הרץ את M על x

אם M קבלה את x – קבל

נשים לב כי $f(<M,x>) = <M'> \in L_{\Sigma^*} \Leftrightarrow <M,x> \in L_u$

נראה רדוקציה $\overline{L_u} \leq L_{\Sigma^*}$ ולכן $L_{\Sigma^*} \notin RE$

$$f(<M, x>) = <M'>$$

כך ש-M' על קלט w:

הרץ את M על x למשך $|w|$ צעדים

אם M דחתה או לא סיימה את החישוב בזמן הזה – קבל

אחרת - דחה

נשים לב כי $f(<M,x>) = <M'> \in L_{\Sigma^*} \Leftrightarrow <M,x> \in \overline{L_u}$

היות ואם M לא עוצרת על x אזי $L(M') = \Sigma^*$

אם M דוחה את x אזי $L(M') = \Sigma^*$

אם M מקבלת את x אזי היא תקבל רק חלק מהמילים ולכן $L(M') \neq \Sigma^*$

$$L_{EQ} \coloneqq \{<M_1, M_2> : L(M_1) = L(M_2)\} \in \overline{RE \cup coRE}$$

נראה רדוקציה מ- L_{Σ^*}

נגדיר:

$$f(<M>) = <M_1, M_2>$$

כאשר M_1 על קלט x :

קבל

$$M_2 = M$$

$$L_{\leq 3} \coloneqq \{<M>: |L(M) \leq 3\} \in coRE \backslash R$$

$$L_{=3} \coloneqq \{<M>: |L(M) = 3\} \in \overline{RE \cup coRE}$$

1. נראה רדוקציה $L_u \leq L_{=3}$ ולכן לפי משפט הרדוקציה $L_{=3} \notin coRE$
2. נראה רדוקציה $\overline{L_u} \leq L_{=3}$ ולכן לפי משפט הרדוקציה $L_{=3} \notin RE$

$L_u \leq L_{=3}$ .1

$$f(<M, w>) = <M'>$$

M' על קלט x:

אם $x = 0$ או $x = 0^2$ או $x = 0^3$

הרץ את M על w וענה כמוה.

אחרת, דחה.

***f* מלאה וניתנת לחישוב:** היות והיא פולטת קידוד של מכונה והראנו בשיעור שפליטה של קידוד מכונה הוא חשיב, בנוסף היא מלאה כי לכל קידוד יש פלט (קידוד חדש) ובנוסף היא בודקת תנאי ומריצה מכונה

***f* תקפה:** נשים לב כי M מקבלת את w אם"ם $|L(M')| = 3$

אם M מקבלת את w אזי M מקבלת את $0, 0^2, 0^3$ ולכן $|L(M')| = 3$ ולכן $<M'> \in L_{=3}$

אם M לא מקבלת את w אזי M לא מקבלת כלום ולכן $|L(M')| = 0 \neq 3$ ולכן $<M'> \notin L_{=3}$

$\overline{L_u} \leq L_{=3}$ .2

$$f(<M, w>) = <M'>$$

M' על קלט x:

אם $x = 0$ או $x = 0^2$ או $x = 0^3$

קבל

אחרת
הרץ את M על w וענה כמוה

f **תקפה:** נשים לב כי M **לא מקבלת** את w אם"ם $|L(M')| = 3$

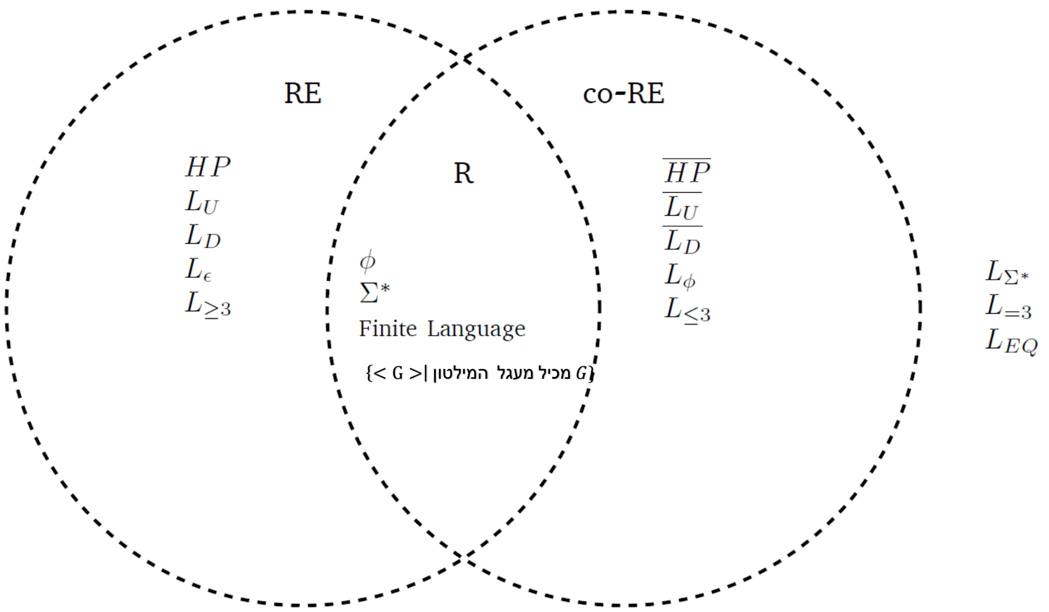
$L_{\geq 3} := \{< M >: |L(M) \geq 3\} \in RE \backslash R$

נראה כי $L \in RE$ ע"י מ"ט $M_{\geq 3}$ אשר מזהה את $L_{\geq 3}$ באופן הבא:
יהא Σ א"ב מעל M ויהא $x_1, x_2, \dots$ הסדר הלקסקוגרפי של כל המילים מעל Σ^*
$M_{\geq 3}$ על קלט $< M >$ כאשר $< M >$ זהו קידוד של מ"ט:
1. לכל i החל מ-1:
   1.1 לכל j החל מ-1 עד i:
      1.1.1 סמלץ את M על x_j למשך i צעדים **ובדוק** אם M קבלה.
      1.1.2 $M_{\geq 3}$ מחזיקה **מונה** של כמות המחרוזות שהתקבלו ע"י M
      1.1.3 **קבל** אם כמות המחרוזות שהתקבלו ע"י M היא 3

הרצה מבוקרת

נכונות: אם $L_{\geq 3} \ni < M >$ אזי קיימות 3 מילים כך ש-M תעצור ותקבל אותן ולכן נמצא את המילים הנ"ל עבור i, j מספיק גדולים.
אולם אם $L_{\geq 3} \not\ni < M >$ אזי לעולם M לא תעצור ותקבל 3 מילים ולכן M תמשיך לנצח
נוכיח ש- $L \notin R$ בעזרת משפט רייס.

נקבל את הגרף הבא:



<u>תכונות של רדוקציות:</u>
1. לכל שפה L מתקיים $L \leq L$
   הוכחה: בהינתן שפה L נגדיר את הרדוקציה $f(x) = x$
   f ניתנת לחישוב – נזיז את הראש קורא כותב עד הסוף ונעצור
   היא מלאה – טרוואלי
   תקיפות: $f(x) = x \in L \Leftrightarrow x \in L$
2. לכל L_1, L_2 שפות מתקיים:
   אם $L_1 \leq L_2$ אזי $\overline{L_1} \leq \overline{L_2}$
   (זו אותה רדוקציה)
   לכל $x \in \Sigma^*$ מתקיים כי $x \in L_1 \Leftrightarrow f(x) \in L_2$

אשר שקול לוגית ל- $f(x) \notin L_2 \Leftrightarrow x \notin L_1$
אשר שקול לוגית ל- $f(x) \in \overline{L_2} \Leftrightarrow x \in \overline{L_1}$
3. טרנזטביות

<u>פורמט כתיבה:</u>
נראה רדוקציה $L_1 \leq L_2$
ומכאן נקבל לפי משפט הרדוקציה כי $< solution >$
נגדיר
$f(< input\ to\ L_1 >) =< input\ to\ L_2 >$
כאשר $< input\ to\ L_2 >$ על קלט $< can\ be\ in\ input\ to\ L_1\ or\ not >$:
$< body\ of\ the\ reduction >$
f מלאה וניתנת לחישוב: היות והיא פולטת קידוד של מכונה והראנו בשיעור שפליטה של קידוד מכונה הוא חשיב, בנוסף היא מלאה כי לכל קידוד יש פלט (קידוד חדש)

f תקפה:
נשים לב כי $f(< input\ to\ L_1 >) =< input\ to\ L_2 > \in L_2 \Leftrightarrow < input\ to\ L_1 > \in L_1$
היות ו:
אם $< input\ to\ L_1 > \in L_1$ ולכן נקבל כי $< input\ to\ L_2 > \in L_2$
אם $< input\ to\ L_1 > \notin L_1$ ולכן נקבל כי $< input\ to\ L_2 > \notin L_2$

משפט: אם $L \in R$ כך ש $L \neq \Sigma^*, \emptyset$ - אזי לכל $L' \in R$ מתקיים כי $L' \leq L$
הוכחה: מכיוון ש-$L' \in R$ קיימת מ"ט M' אשר מכריעה את L' ומכוון ש $L \neq \Sigma^*, \emptyset$ אזי קיימת מילה $w_1 \in L, w_2 \notin L$ ולכן נגדיר את הרדוקציה $f(x) = \begin{cases} w_1, & M'(x) = 1 \\ w_2, & M'(x) = 0 \end{cases}$
דוגמא עבור השפה $L_1 = \{x\ | x\ is\ odd\}$ ו- $L_2 = \{0\}$ אזי נקבל כי $f(x) = \begin{cases} 0\ , & if\ x\ is\ odd \\ 00\ , & else \end{cases}$

<u>תכונות ומשפט רייס</u>
תכונה S היא תת קבוצה של שפות ב RE
בהינתן **תכונה** S נגדיר את שפת כל המכונות אשר מקיימות את תכונה S:
$L_s \coloneqq \{< M > : L(M) \in S\}$
תכונה S היא **<u>טרוואלית</u>** אם $S = \emptyset$ או $S = RE$

דוגמא: $S = \{L \in RE |\ \text{שפה סופית}\ L\}$
אזי $L_s = \{< M >|\ L(M) \in S\}$

<u>משפט רייס</u>: אם S תכונה לא טרוואלית אז $L_s \notin R$
בנוסף אם $S \notin \emptyset$ אזי גם $L_s \notin coRE$
אם $\emptyset \in S$ אזי גם $L_s \notin RE$
הערה: אם S תכונה טרוואלית אזי $L_s \in R$
<u>הוכחה:</u>
תהי S תכונה לא טרוואלית של שפות
נוכיח ש: $L_S \notin R$
נחלק למקרים:
1. S $\emptyset \notin$
מכיוון ש-S אינה טרוואלית קיימת $L \in RE$ כך ש-$L \in S$
מכיוון ש-$L \in RE$ אזי קיימת מ"ט M_L אשר מקבלת את L
נראה רדוקציה $HP \leq L_s$ ולכן $L_s \notin R$ לפי משפט הרדוקציה
ע"י $f(< M, w >) = < M' >$
כאשר M' על קלט x:

הרץ את M על w
הרץ את M_L על x וענה כמוה
(אם M לא עוצרת על w אזי $L(M') = \emptyset$ ולפי ההנחה $\emptyset \notin S$ ולכן $< M' > \notin L_S$

2. $\emptyset \in \boldsymbol{S}$
אזי נגדיר את התכונה $\bar{S} = RE \backslash S$
ו-$\bar{S} \notin \emptyset$ בנוסף S אינה טרוואלית כי $S \neq RE, S \neq \emptyset$
נשים לב כי $L_{\bar{S}} = \overline{L_S}$ כי
$L_{\bar{S}} = \{< M > | L(M) \notin RE \backslash S\} = \{< M > | L(M) \notin S\} = \overline{\{< M > | L(M) \in S\}} = \overline{L_S}$
לפי מקרה 1: $L_{\bar{S}} \notin R$ ולכן $\overline{L_S} \notin R$ ולכן $L_S \notin R$ (כי אחרת מסגירות למשלים נקבל כי $\overline{L_S} \in R$)
מכאן קבלנו כי אם $\emptyset \in S$ אזי קיימת רדוקציה $HP \leq L_{\bar{S}}$ ולכן מתכונת הרדוקציה $\overline{HP} \leq L_S$
ומכאן ממשפט רייס נובע כי $L_S \notin RE$

דוגמה:

$$L_{\geq 3} \coloneqq \{< M > \mid L(M) \geq 3 \} \notin R$$

נראה כי $L \notin R$ ע"י משפט רייס.
נגדיר את התכונה $S \coloneqq \{L \in RE \,||L| \geq 3\}$ ונשים לב כי $L_s = \{< M > | L \in S\} = L$
נוכיח כי S אינה טרוואלית כי:
- $S \neq \emptyset$ ו- $\Sigma^* \in S$
- $S \neq RE$ כי $\emptyset \notin S$

מכאן הראנו כי קיימות 2 שפות כך שאחת ב-S ואחת לא ולכן S לא טרוואלית ולכן לפי משפט רייס $L_S = L_{\geq 3} \notin R$.
נשים לב כי $\emptyset \notin S$ ולכן גם $L_{\geq 3} \notin coRE$

עוד דוגמא $L \coloneqq \{< M > | L(M) \subseteq \{0^n 1^n | n \in \mathbb{N}\}$
נוכיח באמצעות רייס כי $L \notin RE$
נגדיר את התכונה $S = \{L \in RE \,| L \subseteq \{0^n 1^n | n \in \mathbb{N}\}$
נשים לב כי $L = L_S$
נוכיח כי S אינה טרוואלית כי
$S \neq RE$ היות ו- $\{0\} \notin S$
$S \neq \emptyset$ כי $\{01\} \in S$
ולכן ממשפט רייס $L_S \notin R$
נשים לב כי $\emptyset \in S$
כי $\emptyset \subseteq \{0^n 1^n | n \in \mathbb{N}\}$
ולכן לפי הכללה משפט רייס, $L_S \notin RE$

<u>דגשים לשימוש במשפט רייס:</u>
1. רק עבור שפות מהצורה $L = \{< M > | \ldots.\}$
   כלומר רק מכונה אחת.
2. יש לשים לב שהתכונה מדברת על שפות
   דוגמא לשפות שלא ניתן להשתמש ברייס עבורם
   1. $L = \{< M > | M \text{ halts on } \epsilon\}$
   2. $L = \{< M > | M \text{ reject all } \omega \in \Sigma^*\}$

   זוהי תכונה של המכונה, יכול להיות שהיא לא תעצור, אבל היא ספציפית דוחה.

על מנת להוכיח שהתכונה לא מתאימה לרייס צריך להראות דוגמא לשתי מכונות עם אותה שפה כך שאחת מקיימת אותה ואחת לא.

<u>שלמות שפות</u>

נאמר ששפה L היא RE-**קשה** אם לכל $L' \in RE$ מתקיים $L' \leq L$
נאמר ששפה L היא RE-**שלמה** אם L היא RE-**קשה** וגם $L \in RE$

נכתב ע"י צבי מינץ zvimints@gmail.com

דוגמא להוכחה כי L_u היא RE-שלמה

1. הוכחנו כי $L_u \in RE$
2. תהיה $L' \in RE$ שפה כלשהי, לכן קיימת מ"ט $M_{L'}$ אשר מקבלת את L'
נראה $L' \leq L_u$ ע"י הפונקציה $f: \Sigma^* \to \Sigma^*$
$f(x) =< M_{L'}, x >$

נשים לב כי f מלאה וניתנת לחישוב וf תקפה כי אם $x \in L' \Leftrightarrow M_{L'}$ מקבלת את x
$\Updownarrow$
$< M_{L'}, x > \in L_u$

משפט: אם L שפה RE-שלמה ו $L \leq L'$ עבור $L' \in RE$ אזי L' היא גם RE-שלמה
משפט: אם L שפה RE-שלמה אזי $\bar{L}$ שפה $coRE$-שלמה

סיבוכיות

בתוך הבעיות הניתנות לחישוב, מהן הבעיות הקשות יותר ומהן הקלות יותר?

סיבוכיות הזמן (זמן ריצה) של מ״ט M היא פונקציה חלקית $t_M: \Sigma^* \to \mathbb{N}$
אם M עוצרת על x אז $t_M(x)$ הוא מספר צעדי החישוב עד לעצירה (אם M לא עוצרת על x אזי $t_M(x)$ לא מוגדרת)

חסם זמן ריצה עבור מ"ט M היא פונקצייה $T: N \to N$ המקיימת לכל $x \in \Sigma^*$
$t_M(|x|) \leq T(|x|)$

מ"ט M תקרא **פולינומית** אם קיים פולינום $p(n) = O(n^c)$ כך שזמן הריצה של M מהווה חסם עבור, כלומר $T(n) = p(n)$
דוגמא n^2 יעיל 2^n לא יעיל

מ״ט **א״ד** N תקרא **פולינומית** אם קיים פולינום $p(n)$ כך ש-N עוצרת על x תוך $p(|x|)$ צעדים **בכל** מסלוליה

עבור פונקצייה $T(n)$ נגדיר:
$DTIME(T(n)) \coloneqq \{L \in R \mid$ קיימת מ"ט M המכריעה את L בזמן $O(T(n))\}$
$NTIME(T(n)) \coloneqq \{L \in R \mid$ קיימת מ"ט א"ד N המכריעה את L בזמן $O(T(n))\}$

מחלקות

$$P \coloneqq \{L \in R \mid exists\ \textbf{polynomial deterministic}\ TM\ for\ L\} = \cup_{c=0}^{\infty} DTIME(n^c)$$

$$NP \coloneqq \{L \in R \mid exists\ \textbf{polynomial non-deterministic}\ TM\ for\ L\} = \cup_{c=0}^{\infty} NTIME(n^c)$$
$$coNP \coloneqq \{L \mid \bar{L} \in NP\}$$

אבחנה: $P \subseteq NP$ כי כל מ"ט דטרמינסטית היא מקרה פרטי של מ"ט א"ד
$P \subseteq_? NP$ - לא ידוע (בעיית מליון הדולר)

הגדרה שקולה למחלקה NP
NP היא קבוצת כל השפות שיש עבורן מ"ט דטרמינסטית שיודעת, בהינתן עד המעיד על השייכות של המילה לשפה, לוודא שאכן העד הזה הוא נכון.

שפה $L \subseteq \Sigma^*$ שייכת ל-NP אם:

נכתב ע"י צבי מינץ zvimints@gmail.com

קיים עבורה אלגוריתם וידוא $V_L(x,y)$ וקיימים 2 פולינומים $p,q:\mathbb{N}\rightarrow\mathbb{N}$ כך ש:

1. אם $x\in L$ אזי קיים y (**עד**) כך ש- $|y|\le p(|x|)$ כך ש- $V_L(x,y)=1$
2. אם $x\notin L$ אזי לא קיים y (**עד**) כך ש- $V_L(x,y)=1$
3. עבור כל $x\in\Sigma^*$ ו- y **עד** אזי $V_L(x,y)$ רץ ב$q(|x|)$ זמן

דוגמאות:

$3\text{-}Col \coloneqq \{ <G> \mid G \ is\ 3\ color \}$

טענה: $3\text{-}COL\in NP$

הוכחה: נגדיר את מ"ט הא"ד הפולינומית N:
N על קלט $<G>$ כאשר G הינו גרף לא מכוון:
עבור כל קודקוד בגרף נחש צבע מתאים מהצבעים $\{1,2,3\}$
בדוק האם לא קשת בין כל 2 קודקדים צבועים באותו צבע, במידה ואין – קבל.
אחרת, דחה.

נכונות: אם $<G>\in 3COL$ אזי קיימת צביעה חוקית $C:V\rightarrow\{1,2,3\}$ כך שלכל $(u,v)\in E$ מתקיים $c(u)\neq c(v)$ ולכן N תנחש השמה זאת ולאחר בדיקה תקבל
אם $<G>\notin 3COL$ אזי לא קיימת צביעה חוקית, אזי לא נעבור את שלב הבדיקה והמכונה תדחה.

זמן ריצה: לעבור על כל קודקודים בגרף ולבחור צבע מתאים זה $O(|<V>|)$
לבדוק אם בכל כל 2 קודקדים בגרף אם הם צבועים באותו צבע יכול להעשות ב $O(|<V^2>|)$
סה"כ פולינום בגודל הקלט

הוכחה בעזרת עד ואלגוריתם וידוא:
על מנת להראות שהבעיה היא ב-NP, העד שלנו לוקח גרף $G=(V,E)$ וצביעה $c:V\rightarrow\{1,2,3\}$ ובודקת ב-$O(n^2)$ אם c היא צביעה חוקית ע"י בדיקה כל אחת מקודקודים בקצוות של כל צלע האם הם צבועים בצבעים שונים
ניתן לראות כי גם העד וגם האלגוריתם וידוא רצים בפולינום בגודל הקלט.

טענה: $3\text{-}COL\notin P$

האלגוריתם הדטרמינסטי הידוע עובר על כל האפשריות הצביעה של הקודקודים ב-3 צבעים רץ בסיבוכיות $3^{|V|}$ (אקספוננציאלי)

טענה: $2\text{-}COL\in P$

נבנה מכונה דטרמניסטית פולינומית שבודקת האם הגרף הוא דו צדדי $O(|E|+|V|)$

טענה: $L=\{<A,n>:$ מורכבת ממספרים שלמים, וקיימת תת קבוצה של A שסכומה n $A\}\in NP$

הוכחה: נראה מ"ט א"ד:
N על קלט $<A,n>$ כאשר A היא קבוצה של מספרים שלמים ו-n הוא מספר:
נחשב תת קבוצה $S\subseteq A$
אם הסכום של איברי S שווה ל-n – קבל
אחרת, דחה.

נכונות: $<A,n>\in L$ אם"ם קיימת $S\subseteq A$ כך שסכום האיברים של S שווה ל-n
אם"ם קיים ניחוש של S שכזאת
אם"ם N מקבלת

זמן ריצה: ניחוש תת הקבוצה זה $O(|A<A>|)$ – מעבר על A והחלטה אם לקחת איבר
סכימה – $O(|<A>|)$
השוואה – $O(|<A>|)$
סה"כ פולינומי באורך הקלט

טענה: $L=\{<G>\mid there\ in\ G\ clique\ of\ size\ k\}\in P$

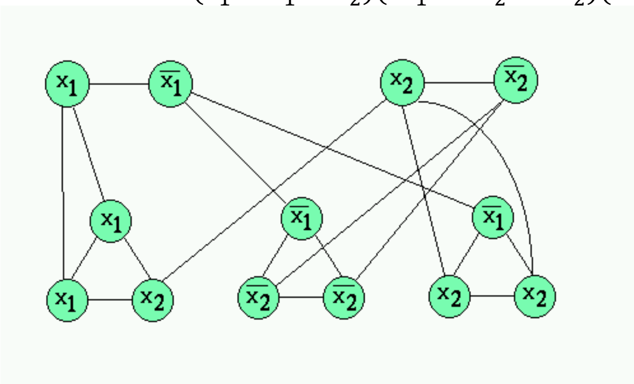
הוכחה: נראה מ"ט דטרמינסטית המחשבת את L:
עבור על כל k הקודקודים $(v_1,\dots v_k)$
בדוק האם כולם מחבורים – אם כן, קבל
דחה

זמן ריצה: כמות הk הזוגות היא $\binom{|V|}{k}$ אשר זהו $O(|V|^k)$
ולכן פולינומי בגודל הקלט

טענה: $Vertex\ Cover \in NPC$

1. $\boldsymbol{VC \in NP}$
עד: קבוצת קודקודים בגודל k המהווה את הכיסוי
אלגוריתם וידוא: עבור על כל אחת מהצלעות ובדוק אם אחת מהקצוות נמצא בקבוצה
2. $\boldsymbol{VC \in NPH}$
נראה רדוקציה $3SAT \leq_p VC$
רעיון: נמיר פסוקית $\varphi := (x_1 \vee x_2 \vee x_3) \wedge (x_1' \vee x_2' \vee x_3)$
לגרף כאשר עבור כל ליטרל x_i נבנה לו קודקוד x_i בגרף ו-x_i' ונחבר בינהם בצלע (1)
לאחר מכן נבנה גאדט K_3 עבור כל אחת מהפסוקיות
לאחר מכן נחבר כל קודקוד x_i בגאדט המתאים לפסוקית לקודקודים הנוספים ב(1).

נקבל: $(x_1 \vee x_1 \vee x_2)(!x_1 \vee !x_2 \vee !x_2)(!x_1 \vee x_2 \vee x_2)$



נשים לב כי יש השמה מספקת ל-$\boldsymbol{\varphi}$ אם"ם יש כיסוי תקין בגודל $\boldsymbol{k = 2 \cdot m + n}$
כאשר $\boldsymbol{m}$ זה כמות הפסוקיות ו- $\boldsymbol{n}$ כמות המשתנים

דוגמא לשפה שלא ידוע אם היא ב-NP:
$$\overline{3COL} := \{<G>|\ G\ is\ not\ 3\ color\}$$

בעיות חיפוש: רוצים את הפתרון ולא מספיק לומר כן או לא
בעיות הכרעה: החלטה האם המילה כן בשפה או לא

רדוקציה פולינומית
רדוקציה פולינומית f משפה A לשפה B היא פונקציה המקיימת:
1. קיים אלג' פולינומי שבהינתן x מחזיר $f(x)$
2. $x \in A \Leftrightarrow f(x) \in B$

סימון: $A \leq_p B$

משפט הרדוקציה
לכל $L_1, L_2 \subseteq \Sigma^*$ שפות מתקיים כי אם $L_1 \leq_p L_2$ אזי:
1. אם $L_2 \in P$ אזי $L_1 \in P$
2. אם $L_2 \in NP$ אזי $L_1 \in NP$
3. אם $L_2 \in coNP$ אזי $L_1 \in coNP$

הגדרה: שפה ***B*** היא ***NP***-קשה אם לכל $B \subseteq \Sigma^*$ מתקיים $A \leq_p B$
הגדרה: שפה ***B*** היא ***NP***-שלמה אם:
$B \in NP$
B היא ***NP***-קשה

קורס **חישוביות**
מרצה: גב' סבתו יעל
נכתב ע"י צבי מינץ zvimints@gmail.com
מבוסס על הרצאות של גיל לוי תשע"ח

SAT: $SAT \coloneqq \{<\varphi> \mid$ φ היא נוסחה לוגית בצורה CNF כך שיש ל-φ השמה מספקת $\}$

נוסחה לוגית בצורה CNF :

$$\underbrace{\varphi}_{\text{נוסחת } CNF} \coloneqq \underbrace{(x_1 \vee \dots . x_3)}_{\text{פסוקית}} \wedge \left(\underbrace{x_2}_{\text{ליטרל}} \vee \dots \vee \overline{x_n} \right) \wedge \dots$$

השמה מספקת: עבור נוסחא φ עם משתנים $x_1, \dots, x_n$ השמה ל-φ היא הצבת ערכים (0 או 1) בכל אחת ממשתני φ כך שהערך הכולל של הנוסחא הוא $True$ (1)

אם יש לנו פסוק CNF שבו כל פסוקית מכילה בדיוק k ליטרלים, נאמר שהפסוק φ הינו $k - CNF$

טענה: $SAT \in NP$
הוכחה: נגדיר את מ"ט הא"ד הפולינומית N:
N על קלט $<\varphi>$ כך ש-φ היא נוסחת לוגית בצורת CNF:
1. נחש סדרה של n ביטים כאשר n = למספר המשתנים השונים ב-φ
כל ביט מייצג השמה למשתנה המתאים.
2. הצב את הניחוש ב-φ ואם יצא ערך אמת – קבל, אחרת – דחה.

נכונות: אם $<\varphi> \in SAT$ אזי קיימת השמה מספקת ל-φ ולכן N תנחשב את סדרת הביטים המייצגת את ההשמה הנ"ל ותציב אותה ותקבל
אחרת, לא קיימת השמה מספקת ולכן N תדחה

<u>משפט קוק-לוין (1971)</u>: SAT היא NP-שלמה.

Note: We could prove directly that 3COL $\leq_p$ 100COL as following: Clearly that 100COL $\in$ NP. the reduction will construct a new graph G' such that G' is union of the original graph, and K_{97}, each vertex v in G will be connect the every vertex in K_{97}.
Its pretty easy to see that $G \in$ 3COL if and only if $G' \in$ 100COL, and clearly that the reduction is polynomial. hence, 100COL is NP-complete problem.

<u>רדוקציות ידועות:</u>

SAT $\leq_p$ 3SAT

נראה כעת רדוקציה פולינומית מ-SAT אל 3SAT. יחד עם מה שכבר ראינו, זה מוכיח כי 3SAT היא NP-שלמה.

מספיק להראות איך לתרגם פסוקית אחת $C = (u_1 \vee u_2 \vee \dots \vee u_n)$ לפסוק 3CNF; עבור פסוק כללי אפשר להפעיל את התהליך על כל פסוקית לחוד.

אם $C = (u_1)$ אז מתרגמים אותה לפסוקית $(u_1 \vee u_1 \vee u_1)$.

אם $C = (u_1 \vee u_2)$ אז מתרגמים אותה לפסוקית $(u_1 \vee u_1 \vee u_2)$

אם $C = (u_1 \vee u_2 \vee u_3)$ משאירים אותה כמות שהיא.

עד כה היה קל לראות שכל השמה שמספקת את הפסוקית המקורית, מספקת גם את החדשה ולהפך.

עבור פסוקית $C = (u_1 \vee u_2 \vee \dots \vee u_n)$ כך ש-$n \geq 4$ הפתרון מורכב יותר ודורש שימוש במשתני עזר שנסמן כ-$y_1, y_2, \dots, y_{n-3}$.

נחליף את הפסוק בסדרת הפסוקיות הבאה:

$$(u_1 \vee u_2 \vee y_1) \wedge (\overline{y_1} \vee u_3 \vee y_2) \wedge \dots \wedge (\overline{y_{n-3}} \vee u_{n-1} \vee u_n)$$

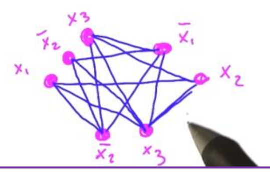
$$\leq_p 3SAT \leq_p VC$$

$$SAT \leq_p 3COL \leq_p k_{\geq 3} - COL$$

$$\leq_p Clique \leq_p IS$$

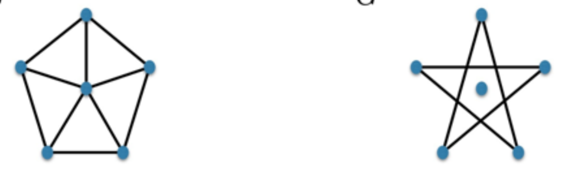
SAT ——> CLIQUE

$(x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2) \wedge (\overline{x_2} \vee x_3)$



Clique $\leq_p$ IS
(G, k) *has a clique of size* $k \Leftrightarrow (G', k)$ *has IS of size* k

G $\qquad$ G'



דוגמאות מקורס אלגורתמים 2 למצטיינים חלק סיבוכיות

$VC \leq_p IS$

Lemma

I is an independent set of $G(V, E)$, if and only if $V - I$ is a vertex cover of G.

Proof

$\Rightarrow$ If I is an independent set, then there is no edge with both endpoints in I. Hence $V - I$ touches every edge.

$\Leftarrow$ If $V - I$ touches every edge, then each edge has at least one endpoint in $V - I$. Hence I is an independent set. $\square$

DNF $\in P$

Proof. To satisfy a DNF formula, all we need is to satisfy one clause and the disjunction of all clauses is satisfied. Traverse the clauses. If a clauses contains a literal and its negation $x_i \wedge \neg x_i$, then obviously it cannot besatisfied. Otherwise, it can be satisfied (assign F to all variables that appear with ¬ sign, assign T to all variables without ¬ sign, assign arbitrary values to all variables that do not appear in the clause at all). Going one clause after another and checking if some clause does not contain both a variable and its negation is clearly poly-time (in fact, linear time).

1SAT $\in P$

Proof. Clearly, we can solve a given φ in $O(k)$ linear time, when k being the number of clauses. we just need to assign values *true* to the propositions that are positive and *false* otherwise. Ofcourse we we have a literal and his negation s.t $x_i \wedge \neg x_i$ the formula is unsatisfiable.