תורת החישוביות 236343

15 ביולי 2010

תוכן עניינים

0.1 סקירת הקורס

שאלות מרכזיות

- אילו **בעיות** ניתן לפתור על ידי מחשב?
- אילו בעיות ניתנות לפתרון **יעיל** על ידי מחשב?

דגש - מה **לא ניתן** לעשות.

דוגמאות לבעיות לא פתירות

נדגיש שבעיות "לא פתירות" בהקשר זה אינן בעיות שאנחנו לא יודעים לפתור היום, אלא בעיות שאנחנו **יודעים** ש**לא ניתן לפתור** אותן.

- נתונות שתי תכניות מחשב, האם הן שקולות?
- [דוגמא מאוטומטים] נתונים שני דקדוקים חסרי הקשר, האם הם שקולים?
- בהנתן תכנית מחשב, האם היא עוצרת לכל קלט?
- [בעיית העצירה] בהנתן תכנית מחשב, וקלט לתכנית, האם התכנית עוצרת על הקלט הזה?
- בהנתן אוסף (סופי) של מטריצות ריבועיות, האם קיימת מכפלה שנותנת מטריצת אפסים?

הוכחה: [1]שבעיית העצירה לא ניתנת לפתרון
כלומר - נוכיח שלא קיימת תכנית, A, שעל כל קלט (M,x) (כאשר M תכנה, ו-x קלט ל-M) תקיים -

- A תמיד עוצרת
- $M(x)$ עוצרת $\Leftarrow A(M,x)=Yes$
- $M(x)$ לא עוצרת $\Leftarrow A(M,x)=No$

נניח בשלילה שקיימת A כזו, ונתאר תכנית חדשה $B(w)$ -

- אם $A(w,w)=Yes$ בצע לולאה אינסופית
- אחרת - עצור

נשים לב כי B מוגדרת היטב (חוקית). נרצה לדעת האם $B(B)$ עוצרת?

- אם $B(B)$ לא עוצרת $\Leftarrow$ מהבנייה נובע $A(B,B)=Yes$, אבל מנכונות A נקבל $B(B)$עוצרת - וזו סתירה.
- אם $B(B)$ עוצרת $\Leftarrow$ מהבנייה נובע $A(B,B)=No$, אבל מנכונות A נקבל $B(B)$ לא עוצרת - וזו סתירה.

כלומר - לא ייתכן שקיימת A כנ"ל.

■

[1]לא באמת הוכחה פורמלית

מהלך חלק א' של הקורס -

- הגדרת מודל החישוב
  - פורמלית
  - פשוטה
  - חזקה
- סווג בעיות (פתירה, לא פתירה)
  - דוגמאות חשובות
  - טכניקות הוכחה

חלק ב' של הקורס -

מבין הבעיות הפתירות, אילו בעיות ניתנות לפתרון יעיל? לא ניתנות לפתרון יעיל?
דוגמאות לבעיות ש"לא ניתנות לפתרון יעיל" -

- בהנתן גרף G, האם קיים מסלול ב-G שעובר בכל צומת בדיוק פעם אחת? (מסלול המילטוני)
- בהנתן נוסחא על n משתנים, האם ניתן לספק אותה?
- סודוקו $n \times n$, איך (או האם ניתן) לפתור?

עובדות -

- לכל הבעיות הנ"ל קיים פתרון לא יעיל
- לא ידוע האם באמת אין פתרון יעיל, אבל -
  - כולן שייכות למחלקת בעיות אחת ($NP - complete$) או שכולן פתירות ביעילות, או שאף אחת מהן לא פתירה ביעילות.

במהלך חלק ב' -

- הגדרת "יעילות"
- סיווג בעיות -
  - דוגמאות חשובות
  - טכניקות הוכחה
- מה בכל זאת ניתן לעשות?

חלק I

חלק א' של הקורס

1 מכונת טיורינג

הגדרה 1.1 מכונת טיורינג (מ"ט)
M היא שביעיה - $M = (Q, q_0, F, \Gamma, \Sigma, \flat, \delta)$ המקיימת -

- Q היא הקבוצה הסופית שאבריה נקראים מצבים
  - $q_0 \in Q$ נקרא המצב התחילי
  - $F \subseteq Q$ אבריה נקראים מצבים סופיים
  - Γ היא קבוצה סופית שנקראת א"ב העבודה
  - $\Sigma \subsetneq \Gamma$ נקראת א"ב הקלט
  - $\flat \in \Gamma \backslash \Sigma$ נקרא רווח או בלנק
  - $\delta : (Q \backslash F) \times \Gamma \mapsto Q \times \Gamma \times \{L, R, S\}$ נקראת פונקציית המעברים

הגדרה 1.2 קונפיגורציה (צילום מצב *snapshot*)
קונפיגורצית של מ"ט M היא שלשה $C = (\alpha, q, i)$ כאשר -

- $q \in Q$ נקרא המצב הנוכחי
  - $i \in \mathbb{N}$ נקרא מיקום הראש
  - $\alpha \in \Gamma^*$ נקרא תוכן הסרט

הקונפיגורציה התחילית של M על $x = \Sigma^*$ -

$$C = (x, q_0, 1)$$

קונפיגורציה סופית היא כל קונפיגורציה שבה - $q \in F$.

הערה 1.3 (תוספות להגדרה)

- T מספר הצעדים עד כה, t התא הימני ביותר בו ביקרנו עד כה, ומתקיים -

$$t \leq T$$

  - <u>מוסכמה</u> - אורך של α בהגדרת קונפיגורציה -

$$|\alpha| = \max\left\{t, {}^{\text{input}}_{\text{length}}\right\}$$

הגדרה 1.4 צעד חישוב
אם הקונפיגורציה הנוכחית היא (α, q, i), ואם - $\delta(q, \alpha_i) = (p, b, d)$ אזי -

- המצב q יוחלף במצב p

- האות α_i תוחלף באות b
- מיקום הראש ישתנה -

$$i+1 \Leftarrow d = R \text{ -}$$
$$i \Leftarrow d = S \text{ -}$$
$$\max\{1, i-1\} \Leftarrow d = L \text{ -}$$

מסקנה 1.5 לכל קונפיגורציה לא סופית C קיימת קונפיגורציה עוקבת יחידה C' ומסמנים -

$$C \vdash C'$$

הגדרה 1.6 החישוב של מ"ט M על קלט x הוא סדרת קונפיגורציות - $C_0, C_1, C_2 \ldots$ המקיימת -

- לכל $i > 0$ מתקיים $C_{i-1} \vdash C_i$
  - אם הסדרה סופית, אז הקונפיגורציה האחרונה בסדרה היא קונפיגורציה סופית

הגדרה 1.7 הפונקציה שמכונת טיורינג M מחשבת מסומנת ב-$f_M : \Sigma^* \mapsto \Gamma^*$ ומוגדרת באופן הבא -

- אם החישוב של M על x מסתיים ו-$C = (\alpha, q, i)$ היא הקונפיגורציה הסופית אזי -

$$f_M(x) = \alpha_1 \alpha_2 \ldots \alpha_{i-1}$$

- אם החישוב של M על x לא מסתיים אז $f_M(x)$ לא מוגדר!

דוגמא[2]

נבנה מ"ט M שמחשבת את הפונק' $f(x) = 0x$ כאשר $x \in \{0,1\}^*$.
יש יותר מדרך אחת לבנות את המכונה, נציע דרך שמבצעת את המשימה במעבר אחד - כשבכל פעם "נזכור" את התו הבא שצריך לרשום. נגדיר -

$$M = (Q, q_0, F, \Gamma, \Sigma, \flat, \delta)$$

כאשר -

$$Q = \{q_0, q_1, q_2\},\ F = \{q_2\},\ \Sigma = \{0,1\},\ \Gamma = \{0,1,\flat\}$$

המשמעות של המצבים -

- q_0 מצב שזוכר 0 לכתיבה בצעד הבא
- q_1 מצב שזוכר 1 לכתיבה בצעד הבא

פונקציית המעברים δ תתואר ע"י טבלה -

$Q\setminus F$ \ Γ	0	1	$\flat$
q_0	$(q_0, 0, R)$	$(q_1, 0, R)$	$(q_2, 0, S)$
q_1	$(q_0, 1, R)$	$(q_1, 1, R)$	$(q_2, 0, S)$

[2] הרצאה שניה - 9/3/10

טענת נכונות - החישוב של M על x הוא סדרה של $n+2$ קונפיגורציות -

$$\begin{aligned} C_0 &= (x, q_0, 1) \\ C_i &= (0x_1 \dots x_{i-1} x_{i+1} x_{i+2} \dots x_n, q_{x_i}, i+1) \\ C_{n+1} &= (0x, q_2, n+2) \end{aligned}$$

ההוכחה באינדוקציה.

דוגמא - מ"ט לחישוב $f(x) = xx$

האלגוריתם -

1. זהה את האות הבאה לכתיבה, a, וזכור אותה באמצעות המצב.
2. עבור ימינה עד ל-$\flat$ הראשון וכתוב שם a''.
   בפעם הראשונה עשה זאת תוך סימון כל האותיות שלא הועתקו ב-$'$.
3. חזור שמאלה על פני האותיות המסומנות על מציאת האות השמאלית ביותר שמסומנת ב-$'$, a'.
4. זכור את האות הזו באמצעות המצב, מחק ממנה את ה-$'$ ועבור ל-2.
5. אם אין אות כנ"ל - מחק את הסימונים משאר האותיות, מקם את הראש על ה-$\flat$ השמאלי ביותר - ועבור למצב הסופי.

תיאור פורמלי של המכונה -

$$M = (Q, q_0, F, \Gamma, \Sigma, \flat, \delta)$$

כאשר -

$$\begin{aligned} Q &= \{q_0, q_1, q_2, q_f\} \cup \{q_a : a \in \Sigma\} \\ F &= \{q_f\} \\ \Gamma &= \Sigma \cup \Sigma' \cup \Sigma'' \cup \{\flat\} \\ \Sigma' &= \{a' : a \in \Sigma\} \\ \Sigma'' &= \{a'' : a \in \Sigma\} \end{aligned}$$

תיאור המצבים -

- q_0 מצב תחילי וזיהוי האות הבאה להעתקה
- q_a שמירת האות a בזכרון ומעבר ימינה עד $\flat$. במעבר ראשון מתבצע סימון אותיות קלט ב-$'$.
- q_1 מעבר שמאלה על אותיות מתוך Σ', Σ''.
- q_2 החלפת אותיות מתוך Σ'' באותיות מתוך Σ (מחיקת ה-$''$).
- q_f מצב סופי

פונקציית המעברים -

	$a \in \Sigma$	$a' \in \Sigma''$	$a'' \in \Sigma''$	$\natural$
q_0	(q_a, a, R) גילוי אות להעתקה בפעם הראשונה	(q_a, a, R) גילוי אות להעתקה ומחיקת ה־′	(q_2, a'', S) גילוי שהכל כבר הועתק	$(q_f, \natural, S)$ טיפול במילה ריקה ε
$q_b : b \in \Sigma$	(q_b, a', R) סימון במעבר ראשון ימינה	(q_b, a', R) מעבר ימינה ע"פ אותיות ב־Σ'	(q_b, a'', R) מעבר ימינה ע"פ אותיות ב־Σ''	(q_1, b'', L) כתיבה מהזכרון
q_1	(q_0, a, R) הגענו לאות שכבר טופלה	(q_1, a', L) מעבר שמאלה ע"פ אותיות ב־Σ'	(q_1, a'', L) מעבר שמאלה ע"פ אותיות ב־Σ''	
q_2	-	-	(q_2, a, R) מחיקת ה־"	$(q_f, \natural, S)$ סיום

2 שקילות מודלים

הגדרה 2.1 פוזל

הוא אוסף של אובייקטים כך שלכל אובייקט מתאימה פונקציה שהוא מחשב.

ראינו את מכונת טיורינג בתור מודל, אבל אפשר לקחת למשל תוכנת מחשב בתור מודל, ניתן להתבונן על הגדרות קומבינטוריות וכו'... אנחנו נתעסק במודלים דומים למכונת טיורינג.

הגדרה 2.2 פוזלים שקולים

שני מודלים יקראו שקולים אם אוסף הפונקציות המחושבות על ידם זהה.

דוגמא - מודל מ"ט זריזה

מוגדר בדומה למ"ט רגילה, למעט:

$$\delta : (Q \backslash F) \times \Gamma \mapsto Q \times \Gamma \times \{LL, L, S, R, RR\}$$

כאשר LL, RR - לבצע שני צעדים ימינה או שמאלה - בהתאמה (תחת המגבלה הרגילה - לא ניתן ליפול "שמאלה" מקצה הסרט).

טענה 2.3 מודל מ"ט זריזה $\equiv$ מודל מ"ט

הוכחה: הוכחת שקילות, כרגיל, ע"י "הכלה דו כיוונית" -

1. בהנתן מ"ט רגילה M היא מקרה פרטי של מ"ט זריזה (מחשבת אותה הפונק').
2. בהנתן מ"ט זריזה M שמחשבת f_M נבנה מ"ט רגילה M' שמחשבת את אותה הפונק' באופן הבא -

$$\begin{aligned} Q' &= Q \cup Q_L \cup Q_R \\ Q_L &= \{q_L : q \in Q\} \\ Q_R &= \{q_R : q \in Q\} \end{aligned}$$

המשמעות של q_L - אתה נמצא במצב q, אבל צריך עוד לבצע צעד אחד שמאלה לפני כן. q_R - מוגדר באותו האופן. לכן -

$$\begin{aligned} \delta'(q_L, a) &= (q, a, L) \\ \delta'(q_R, a) &= (q, a, R) \\ \delta(q, a) &= (P, b, d) \end{aligned}$$

אם $d \in \{L, R, S\}$ אזי $\delta'(q, a) = (P, b, d)$.

אם $d = LL$ אזי $\delta'(q,a) = (P_L, b, L)$.
אם $d = RR$ אזי $\delta'(q,a) = (P_R, b, R)$.
טענת הנכונות

- לכל קלט x אם $(x, q_0, 1) \vdash_M^* (\alpha, q, i)$ אז - $(x, q_0, 1) \vdash_{M'}^* (\alpha, q, i)$ (לא בהכרח באותו מספר של צעדים).
  $\Leftarrow$ אם M עוצרת על x אז גם M' עוצרת על x ועם אותו פלט.
  ההוכחה - באינדוקציה.

■

2.1 מודל מכונת טיורינג דו[3] סרטית

תוגדר באופן דומה למכונת טיורינג רגילה - כאשר יש לה גישה לשני סרטים. בכל סרט ראש קורא־כותב, כאשר מיקום ראש אחד על סרט אחד לא תלוי במיקום הראש השני על הסרט השני.
באתחול -

- סרט אחד מכיל את הקלט ואחריו $\natural$, ראש במקום ה־1.
- הסרט השני מכיל רק $\natural$, וראש במקום ה־1.

הפלט - בסרט הראשון, משמאל לראש בזמן העצירה.
פונקציית המעברים -

$$\delta : (Q \backslash F) \times \Gamma^2 \mapsto Q \times \Gamma^2 \times \{L, R, S\}^2$$

מבוססת על שתי האותיות (משני הסרטים) ואומרת מה לכתוב בכל אחד מהם, לאיזה מצב לעבור, ולאן להזיז את כל אחד מהראשים.

דוגמא - מ"ט דו סרטית שמחשבת $f(x) = xx$

תאור מילולי -

- כתוב $ בתחילת סרט 2
- עבור על x בסרט 1, העתק אותו לסרט 2
- החזר את הראש בסרט 2 לתחילת x ($)
- העתק את x מסרט 2 להמשך סרט 1

טענה 2.4 מודל מ"ט דו (k) סרטית שקול למודל מ"ט הרגיל.

הוכחה: שני כיוונים -

- בהנתן מ"ט רגילה M שמחשבת f_M נבנה מ"ט דו סרטית M' שמחשבת את אותה פונק', ותוגדר באותו האופן פרט ל־δ' שתוגדר ע"י -
  אם $\delta(q,a) = (p,b,d)$ אז - $\delta'(q,a,\natural) = (p,b,\natural,d,s)$ (אף פעם אל תזיז את הראש בסרט השני, ותשאיר בו תמיד $\natural$).

[3] ניתן להגדיר באופן דומה מכונת טיורינג עם k סרטים

- בהנתן מ"ט דו סרטית M שמחשבת f_M נבנה מ"ט רגילה $M*$ שמחשבת את אותה הפונק'. ניתן כרגיל להגדיר במגוון דרכים, אנחנו נגדיר באופן הבא -

  אתחול (נעתיק את תוכן הסרט השני אחרי $\$_1$ בסוף הקלט בסרט הראשון)

	$a \in \Gamma \setminus \{\flat\}$	$\flat$	$\$_1$	$a' \in \Gamma'$
p_0	(p_1, a', R)	$(p_1, \flat', R)$		
p_1	(p_1, a, R)	$(p_2, \$_1, R)$		
p_2		$(p_3, \flat', R)$		
p_3		$(p_4, \$_2, L)$		
p_4				(p_5, a', L)
p_5	(p_5, a, L)		$(p_5, \$_1, L)$	(q_0, a', S)

  צעד -

  תתקיים האינווריאנטה הבאה - אם בשלב כלשהו M מכילה את α בסרט 1 ו־β בסרט 2 כאשר הראש הראשון במקום i והראש השני במקום j, אזי הסרט של M^* בתחילת סימולציה של הצעד הבא יכיל -

$$\alpha\$_1\beta\$_2$$

  עם סימון של α_i ו־β_j והראש של M^* מצביע על α_i.

  איסוף - מצא את 2 האותיות a, b ש־M רואה על 2 הסרטים ושמור אותן באמצעות המצב $q^{a,b}$ (q הוא המצב של M)

  החלטה - אם $\delta(q, a, b) = (p, t, u, d_1, d_2)$ אז $\delta^*\left(q^{a,b}, c\right) = (p_{t,u,d_1,d_2}, c, S)$ כלומר - עבור למצב שמקודד את כל מה שצריך לעשות, אבל אל תעשה כלום כרגע (ובלי תלות בערך התא הנוכחי).

  מימוש ההחלטה - ע"פ מיקום הראשים (קודם בסרט 2 ואז בסרט 1) בצע את השינויים המתבקשים.

  סיום - $F* = F$.

■

התזה של $Church$[4]

כל מודל כללי וסביר של חישוב שקול למ"ט

כללי מודל חזק לפחות כמו מ"ט

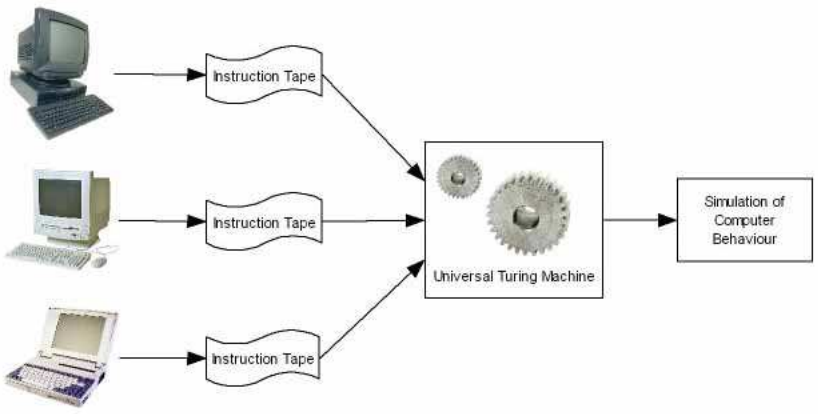
סביר לכל אובייקט במודל יש תיאור סופי

נשים לב שתזה אינה משפט, למה או משפט מתמטי דומה, זו השערה, או "הנחת עבודה". לא ניתן להתבסס עליה כשנרצה להוכיח שקילות מודלים - צריך להראות שקילות באופן פורמלי לכל מודל בנפרד.

2.2 מ"ט אוניברסלית

נרצה לבנות מ"ט שתוכל לבצע את המשימה של כל מ"ט אחרת, מ"ט כזו צריכה לקבל כקלט הן את המכונה M והן את הקלט x ולהחזיר כפלט את $f_m\langle x \rangle$.

[4] הרצאה שלישית - 16/3/10



איור 1: מכונת טיורינג אוניברסלית

קידודים

- A א"ב

בה"כ - $A = \{1, 2, 3, \ldots |A|\}$ ואם $x = x_1 x_2 \ldots x_n$ מחרוזת מעל A^* הקידוד של x יהיה -

$$\langle x \rangle \equiv 1^{x_1} 0 1^{x_2} 0 \ldots 1^{x_n} 0$$

לדוגמא -

$$x = 315 \;\Rightarrow\; \langle x \rangle = 111010111110$$

- קידוד של מ"ט - תהי $M = (Q, q_0, F, \Gamma, \Sigma, \natural, \delta)$

בה"כ $Q = \{1, 2, \ldots |Q|\}$, $q_0 = 1$, $F = \{2, 3\}$

$\Gamma = \{1, 2, \ldots |\Gamma|\}$

$\Sigma = \{1, 2, \ldots, |\Sigma|\} \subsetneq \Gamma$

$\natural = |\Gamma|$

$(L, R, S) \equiv (1, 2, 3)$

אם $\delta(q, a) = (p, b, d)$ נקודד -

$$\begin{array}{ccc} \langle\ \delta(q,a) & = & (p,b,d)\ \rangle \\ 1^q 0 1^a 0 & & 1^p 0 1^b 0 1^d 0 \end{array}$$

את המכונה עצמה נקודד -

$$\langle M \rangle = 1^{|Q|} 0 1^{|\Gamma|} 0 1^{|\Sigma|} 00 \langle \delta(1,1) \rangle 0 \langle \delta(1,2) \rangle 0 \ldots \langle \delta(|Q|, |\Gamma|) \rangle 00$$

הערה 2.5 בהנתן קידוד של מחרוזת או של מ"ט קל לבדוק את תקינותו ולשחזר את המחרוזת/מ"ט

הערה 2.6 (מוסכמה) כל מחרוזת בינארית שאיננה ניתנת לפירוש כקידוד של $\langle M \rangle$ נבין אותה כקידוד של מ"ט M_{STAM} שעוצרת מיד, במצב 3.

- קידוד של קונפיגורציה - תהי $C=(\alpha,q,i)$ ו־$\alpha=a_1a_2\ldots a_m$ אז -

$$\begin{aligned}\langle c\rangle &= \langle a_1a_2\ldots a_{i-1}\rangle \overbrace{0}^{\substack{2\text{ zeroes}\\ \text{with } a_{i-1}}} 1^q0\,\langle a_ia_{i+1}\ldots a_m\rangle \overbrace{00}^{\substack{3\text{ zeroes}\\ \text{with } a_m}}\\ \langle\varepsilon\rangle &= 0\end{aligned}$$

נגדיר פונקציה -

- אם $\langle c\rangle$ קונפיגורציה חוקית, מתאימה ל־$\langle M\rangle$, ולא <u>סופית</u>[5] (ו־c' היא הקונפיגורציה העוקבת) - $NEXT\left(\langle M\rangle,\langle c\rangle\right)=\langle M\rangle,\langle c'\rangle$
- אחרת - $NEXT\left(\langle M\rangle,\langle c\rangle\right)=0$

טענה 2.7 פונקציה $NEXT$ <u>ניתנת לחישוב</u>. כלומר קיימת M_{NEXT} שמחשבת אותה.

הוכחה: M_{NEXT} על קלט $\langle M\rangle,\langle c\rangle$:

- בדוק <u>תקינות</u> $\langle M\rangle,\langle c\rangle$ (ע"פ הגדרת $NEXT$) אם לא מתקיים, עצור עם פלט 0.
- אם תקין, מצא את q ואת האות הנוכחית a בתוך $\langle c\rangle$ (ע"י 00 בקידוד).
- מצא את $\delta\left(q,a\right)$ בתוך הקידוד של $\langle M\rangle$ וחלץ מההמשך את p,b,d, שנה את c ל־c' בהתאם.
- אם $\langle M\rangle$ מתאים למ"ט M_{STAM} עבור לקונפיגורציה הסופית של M_{STAM}.

■

הגדרה 2.8 הפונקציה האוניברסלית

$$U\left(\langle M\rangle,\langle x\rangle\right)=\begin{cases}\langle f_M\left(x\right)\rangle & \langle x\rangle \text{ is leagal together with}\langle M\rangle\\ & \text{and } \langle M\rangle\text{stops on } \langle x\rangle\\ \text{undefined} & \text{Otherwise}\end{cases}$$

טענה 2.9 הפונקציה האוניברסלית U ניתנת לחישוב.

הוכחה: נתאר M_U על קלט $\langle M\rangle,\langle x\rangle$:

- אם $\langle x\rangle$לא חוקי או לא מתאים ל־$\langle M\rangle$ - בצע לולאה אינסופית.
- כתוב על סרט II את $\langle M\rangle,\langle c\rangle$, כאשר $\langle c\rangle$ הקונפיגורציה ההתחלתית של M על x.
- כל עוד c <u>לא סופית</u> חשב $\langle M\rangle\langle c\rangle\leftarrow NEXT\left(\langle M\rangle,\langle c\rangle\right)$.
- אם c <u>סופית</u> הוצא את הפלט שלה.

■

הערה 2.10 נשים לב - אם M לא עוצרת על x גם M_U לא עוצרת על $\langle M\rangle,\langle x\rangle$

כל מ"ט שמחשבת את הפונק' האוניברסלית נקראת מ"ט אוניברסלית.

[5] מצב סופי

3 בעיות הכרעה

הגדרה 3.1 שפה L מעל א"ב Σ היא תת קבוצה (סופית או לא סופית) מתוך Σ^*.

הגדרה 3.2 מ"ט לזיהוי שפות היא מ"ט רגילה M שמקיימת $F = \{q_A, q_R\}$
אומרים שמ"ט M מקבלת את הקלט שלה x אם M עוצרת על x במצב q_A ($A - Accept$).
אומרים שמ"ט M דוחה את x אם M עוצרת על x במצב q_R ($R - Reject$).
ייתכן כמובן ש־M לא עוצרת על x.

הגדרה 3.3 השפה ש־M מקבלת, מסומנת ב־$L(M)$ -

$$L(M) \equiv \{x| \text{ M accepts x}\}$$

אומרים ש־M <u>מכריעה</u> את $L(M)$ אם בנוסף היא עוצרת לכל קלט.

3.1 דוגמאות לשפות ניתנות להכרעה

- Σ^* (לכל a $\delta(q,a) = (q_A, a, s)$)
- $\emptyset$ (באופן דומה)
- כל שפה רגולרית ניתנת להכרעה (אוטומט הוא מקרה פרטי של מ"ט שעוצרת תמיד)
- כל שפה סופית ניתנת להכרעה
- $a^n b^n$
- $L = \{G| \text{ G is connected-graph}\}$

<u>מוסכמה</u> $\Sigma = \{0,1\}$ ברירת המחדל

הגדרה 3.4

$$\begin{aligned} R &\equiv \{L \subseteq \Sigma^* | \text{ Exists M which } decides \text{L}\} \\ RE &\equiv \{L \subseteq \Sigma^* | \text{ Exists M which } accepts \text{L}\} \end{aligned}$$

3.2 תכונות

- $R \subseteq RE$
- R סגורה למשלים $\left(\underbrace{\overline{L}}_{\Sigma^* \setminus L} \in R \Leftarrow L \in R \right)$

הוכחה: לפי הגדרה - $L \in R$ אזי קיימת מ"ט M כל שלכל $x \in \Sigma^*$ -

$$\begin{aligned} M \text{ stops on } q_A &\Leftarrow x \in L \\ M \text{ stops on } q_R &\Leftarrow x \notin L \end{aligned}$$

בנייה - $\overline{M}$ זהה ל-M למעט החלפת תפקידים בין q_A, q_R -

$$\begin{aligned} \overline{M} \text{ stops on } q_R &\Leftarrow x \in L \\ \overline{M} \text{ stops on } q_A &\Leftarrow x \notin L \end{aligned}$$

כלומר -

$$L\left(\overline{M}\right) = \overline{L}$$

ובנוסף $\overline{M}$ עוצרת תמיד, לכן $\overline{L} \in R$. ■

- R סגורה תחת איחוד ($L_1, L_2 \in R$ אזי $L_1 \cup L_2 \in R$).

הוכחה: $L_1, L_2 \in R$ כלומר יש מכונות M_1, M_2 מתאימות.
נבנה M עבור $L_1 \cup L_2$, עם שני סרטים -

- העתק את x מסרט I לסרט II
- הרץ את M_1 על x (סרט I) אם עצרה ב-q_A - סיים ב-q_A.
- אחרת - הרץ את M_2 על x (סרט II). אם עצרה וקיבלה - קבל (q_A). אחרת - דחה (q_R)

קל לבדוק כי $L(M) = L_1 \cup L_2$.

■

תזכורת[6] - בהרצאה הקודמת ראינו את הגדרת המחלקות R ו-RE.
נגדיר כעת מחלקת שפות נוספת -

$$coRE \equiv \left\{L | \ \overline{L} \in RE\right\}$$

כלומר שפה L שייכת למחלקה $coRE$ אמ"ם קיימת מ"ט M כך ש-

- אם $x \notin L$ המכונה עוצרת ודוחה
- אם $x \in L$ המכונה עוצרת ומקבלת או לא עוצרת כלל

טענה 3.5

$$RE \ \cap \ coRE = R$$

בפרט -

$$L, \overline{L} \in RE \Longrightarrow L \in R$$

הוכחה: כיוון אחד - מהגדרה (אם $L \in R$ היא בהכרח ב-RE וב-coRE)
כיוון שני -
תהא $L \in RE \ \cap$ coRE -

- $L \in RE \Leftarrow$ קיימת מ"ט M_1 כך ש-
  - $x \in L \Leftarrow M_1$ עוצרת ומקבלת את x

[6] הרצאה רביעית - 23/3/10

- $x \notin L \Leftarrow M_1$ עוצרת ודוחה או לא עוצרת על x

- $L \in \text{CO-RE} \Leftarrow$ קיימת מ"ט M_2 כך ש־
  - $x \notin L \Leftarrow M_2$ עוצרת ודוחה את x
  - $x \in L \Leftarrow M_2$ עוצרת ומקבלת או לא עוצרת על x

נבנה מ"ט M שמכריעה את L. על קלט x ־

- הרץ במקביל את M_1 ו־M_2 על x. כאשר אחת מהן עוצרת קבל/דחה בהתאמה.

כדי להריץ במקביל נעתיק את הקלט x לסרט נוסף, ונבנה "אוטומט מכפלה", קבוצת המצבים תהיה $q_1 \times q_2$ (כאשר $q_1 \in Q_1,\ q_2 \in Q_2$) והמעברים וההחלפות יתבצעו לפי בנייה טרוויאלית...
כיוון שכאשר M_1, M_2 עוצרות הן במצב הנכון. נקבל כי ־

$$L(M) = L$$

ולכן $L \in R$ ■

3.3 דוגמאות לשפות שהן ב־RE

- כל שפה ב־R (אבל זה לא ממש מעניין)

3.3.1 שפת העצירה HP

$$HP \equiv \{\langle M\rangle, \langle x\rangle \mid M \text{ halts on x }\}$$

טענה 3.6 $HP \in RE$

הוכחה: נבנה M_{HP} שעל קלט $\langle M\rangle, \langle x\rangle$ מחשבת את סדרת הקונפיגורציות בחישוב של M על x (כמו בבנייה של מ"ט אוניברסלית). אם בשלב כלשהו נגיע לקונפיגורציה סופית ־ נעצור ונקבל.

נכונות

אם $\langle M\rangle, \langle x\rangle \in HP$ לפי הגדרת השפה M עוצרת על x, לכן לפי הבנייה של מ"ט אוניברסלית בשלב כלשהו נגיע לקונפיגורציה סופית, ולכן לפי הבנייה החדשה ־ נעצור ונקבל.
אם $\langle M\rangle, \langle x\rangle \notin HP$ לפי הגדרת השפה M לא עוצרת על x, לכן לפי הבנייה לא נגיע לקונפיגורציה סופית ־ ולכן לא נעצור.
■

3.3.2 השפה האוניברסלית L_u

$$L_u = \{\langle M\rangle, \langle x\rangle \mid \text{M accept x}\}$$

טענה 3.7 $L_u \in RE$

ההוכחה דומה מאד להוכחה עבור HP.

3.3.3 שפת האלכסון L_D

$$L_D \equiv \{\langle M\rangle \mid \text{M accept } \langle M\rangle\} = \{\langle M\rangle \mid \ \langle M\rangle \in L\left(M\right)\}$$

ההוכחה כי $L_D \in RE$ גם היא דומה (נריץ את M על $\langle M\rangle$ וכו'...)
המשותף לכל השפות האלו הוא שכרגע אנחנו לא יודעים להוכיח שהן ב־R, בהמשך הקורס נראה שהן אכן לא ב־R.

4 רדוקציה

רדוקציה היא פתרון לבעיה א' ע"י אלגוריתם שפותר את בעייה ב'

דוגמאות

- ראינו שניתן לבנות אלגוריתם למימוש L_u ניתן לבנות ממנו אלגוריתם למימוש L_D על ידי הזנת $\langle M\rangle$ הן כמכונת טיורינג והן כקלט.
- בקורס באלגוריתמים ראינו איך ניתן למצא שידוך מושלם בגרף דו צדדי ע"י אלגוריתם למציאת זרימת מקסימום.

למעשה אנחנו מבצעים רדוקציה כל הזמן...
בקורס בחישוביות נעשה שימוש מעט שונה ברדוקציה, אבל הרעיון הבסיסי זהה.

הגדרה 4.1 תהיינה $L_1, L_2 \subseteq \Sigma^*$ שתי שפות. אומרים שפונקציה $f : \Sigma^* \mapsto \Sigma^*$ היא רדוקציה מ־L_1 ל־L_2 אם -

- f היא פונקציה מלאה (מוגדרת לכל קלט)
- f ניתנת לחישוב (קיימת מ"ט M שבהנתן x מחשבת את $f\left(x\right)$ לכל x)
- (תקפות) $x \in L_1 \ \Leftrightarrow \ f\left(x\right) \in L_2$

אם קיימת f כנ"ל אומרים ש־L_1 <u>ניתנת לרדוקציה</u> ל־L_2 ומסמנים $L_1 \leq L_2$.

4.0.4 דוגמאות

- $L_D \leq L_u$

צריך להראות f המקיימת את ההגדרה -

$$f\left(\langle M\rangle\right) = \left(\langle M\rangle, \langle M\rangle\right)$$

נוודא ש־f מקיימת את ההגדרה -

- f מלאה (הגדרנו כי כל מחרוזת היא קידוד של מכונה, בפרט כל קידוד שלא מתאים לצורת הקידוד שלנו הוא מ"ט שעוצרת מיד).
- f ניתנת לחישוב (ראינו בצורה מפורשת מ"ט שמשכפלת את הקלט)
- תקפות -
  אם $\langle M\rangle \in L_D$, לפי הגדרת L_D $\Leftrightarrow$
  M מקבלת את $\langle M\rangle$ $\Leftrightarrow$
  $\left(\langle M\rangle, \langle M\rangle\right) \in L_2$ $\Leftrightarrow$
  $f\left(\langle M\rangle\right) \in L_2$

- $L_u \leq HP$

צריך להראות f כנדרש בהגדרה -

$$f\left(\langle M\rangle, \langle x\rangle\right) = \left(\langle A\rangle, \langle x\rangle\right)$$

כאשר A היא מ"ט זהה ל-M למעט השינוי הבא - אם M עוברת ל-q_R אז A נכנסת ללולאה אינסופית.
נשים לב שמתקיים -

- $\Leftrightarrow \langle M\rangle, \langle x\rangle \in L_u$
- $\Leftrightarrow M$ מקבלת את x
- $\Leftrightarrow$ מהבנייה, A עוצרת על x
- $\Leftrightarrow$ מהגדרת HP - $\left(\langle A\rangle, \langle x\rangle\right) \in HP$
- $f\left(\langle A\rangle, \langle x\rangle\right) \in HP$

f מלאה (למעשה היא לא מלאה, כי לא מוגדר מה קורה על קלט לא חוקי, אבל אפשר להגדיר אותה באופן מנוון גם שם)

f ניתנת לחישוב -

- אם $\delta_M(q,a) = (q_R, b, d)$ אז $\delta_A(q,a) = (q,a,S)$ (המכונה נשארת במקום לעבור למצב המסיים, זה יוצר לולאה אינסופית) כל מה שנותר לעשות הוא לעבור על הקידוד ולהחליף את המעברים המתאימים בפונקציית המעברים (אין צורך להפעיל את המכונה או כל דבר דומה).

4.1 תכונות של רדוקציות

- לכל L מתקיים $L \leq L$ (באמצעות פונקציית הזהות)
- טרנזיטיביות -

$$L_1 \leq L_3 \Leftarrow \begin{cases} L_1 \leq & L_2 \\ L_2 \leq & L_3 \end{cases}$$

הוכחה:

- $L_1 \leq L_2$ כלומר קיימת f כנדרש בהגדרה
- $L_2 \leq L_3$ כלומר קיימת g כנדרש בהגדרה

נבנה רדוקציה מ-L_1 ל-L_3 ע"י -

$$h(x) \equiv g\left(f(x)\right)$$

ברור ש-h מלאה וניתנת לחישוב. ומתכונות f, g ברור שהיא מקיימת את תנאי התקפות.

■

- אם $L_1 \leq L_2$ אז $\overline{L_1} \leq \overline{L_2}$

מסקנה 4.2 $\overline{L_D} \leq \overline{L_u} \leq \overline{HP}$

משפט 4.3 משפט הרדוקציה (נוסח א') -
אם $L_1 \leq L_2$ אזי -

1. $L_2 \in R \Rightarrow L_1 \in R$

2. $L_1 \in RE \Leftarrow L_2 \in RE$

3. $L_1 \in coRE \Leftarrow L_2 \in coRE$

(כלומר, ה"קושי" של L_1 תמיד קטן או שווה מה"קושי" של L_2).

הוכחה: $L_1 \leq L_2 \Leftarrow$ קיימת רדוקציה f כמובטח בהגדרה ובפרט מ"ט M_f המחשבת אותה.

1. $L_2 \in R \Leftarrow$ קיימת M_2 המכריעה אותה. נבנה M_1 עבור L_1, הפועלת באופן הבא על קלט x -
   - מחשבת $y = f(x)$ (ע"י M_f)
   - מפעילה את M_2 על y ומקבלת / דוחה כמוה

   מתקיים M_f עוצרת תמיד (f מלאה), M_2 עוצרת תמיד (נתון) לכן גם M_1 עוצרת תמיד.
   לפי התקפות $x \in L_1 \Leftrightarrow f(x) \in L_2 \Leftrightarrow M_2$ מקבלת את $f(x) \Leftrightarrow M_1$ מקבלת את x.

2. $L_2 \in RE \Leftarrow$ קיימת M_2 המקבלת אותה. נבנה M_1 עבור L_1 באמצעות אותה הבניה.
   ההוכחה דומה מאד (עד כדי זה ש-M_2 לא בהכרח עוצרת תמיד, במקרה כזה M_1 לא עוצרת - אבל זה לא מטריד אותנו כיוון שאז $x \notin L_1$).

3. ניתן להוכיח בדומה לסעיפים הקודמים, נוכיח בדרך אחרת.
   $\overline{L_2} \in RE \Leftarrow L_2 \in coRE$
   $\overline{L_1} \leq \overline{L_2} \Leftarrow L_1 \leq L_2$
   מתכונה שכבר הוכחנו נובע כי -

$$\overline{L_1} \in RE \implies L_1 \in coRE$$

■

הערה 4.4 לכן - $HP \in RE$
$L_u \in RE \Leftarrow L_u \leq HP$
$L_D \in RE \Leftarrow L_D \leq HP$

משפט 4.5 משפט הרדוקציה (נוסח 2)
אם $L_1 \leq L_2$ אזי -

1. $L_2 \notin R \Leftarrow L_1 \notin R$

2. $L_2 \notin RE \Leftarrow L_1 \notin RE$

3. $L_2 \notin coRE \Leftarrow L_1 \notin coRE$

טענה 4.6 (בשיעור הבא נוכיח כי) $L_D \notin R$.

מסקנה 4.7

1. $\overline{L_D} \notin R$ (סגירות למשלים)

2. $L_u, HP \notin R$ (משפט הרדוקציה, נוסח 2)

3. $\overline{L_u}, \overline{HP} \notin R$

4. $\overline{L_D}, \overline{L_u}, \overline{HP} \notin RE$ (אנחנו יודעים כי $L_D \in RE$, אילו גם $\overline{L_D} \in RE$ אז מטענה 3.5 נקבל כי $L_D \in R$)

טענה 4.8 קיימות שפות שאינן ב־RE (וכמובן אינן ב־R) [7]

הוכחה: משיקולי עוצמות

תזכורת

A, B קבוצות. מתקיים ־

$$|A| \leq |B|$$

אמ"מ קיימת פונקציה חח"ע־

$$f : A \mapsto B$$

בנוסף מתקיים ־

$$|\mathbb{N}| < |\mathbb{R}\,[0,1]|$$

עוצמת קבוצת כל הטבעיים קטנה מעצמת הממשיים בקטע $[0,1]$.

נשים לב שמתקיים ־

$$\begin{aligned} |RE| &\leq |\{\text{All Turing machine which accepts languages}\}| \leq \\ &\leq \left|\{0,1\}^*\right| \leq \\ &\leq |\mathbb{N}| < \\ &< |\mathbb{R}\,[0,1]| \leq \\ &\leq \left|\left\{\text{All languages above } \{0,1\}^*\right\}\right| \end{aligned}$$

בהנתן ממשי a בקטע $[0,1]$ יש לו ייצוג בינארי אינסופי ־

$$a = 0.a_1a_2a_3\ldots$$

נתאים לו את השפה ־

$$L_a = \{w_i | a_i = 1\}$$

כאשר w_i היא המחרוזת ה־i לפי סדר לקסיקוגרפי.

■

טענה 4.9 $\overline{L_D} \notin R$

הוכחה: נניח בשלילה ש־

$$\overline{L_D} \in R$$

אזי קיימת M_D כך ש־$L(M_D) = \overline{L_D}$, כלומר ־

$$\langle M_D \rangle \in \overline{L_D}$$

[7] הרצאה חמישית ־ 6/4/10

ומבחירת M_D זה מתקיים אמ"מ

$$\langle M_D \rangle \in L(M_D)$$

מהגדרת L_D, זה מתקיים אמ"מ -

$$\langle M_D \rangle \in L_D$$

וזה יתקיים אמ"מ -

$$\langle M_D \rangle \notin \overline{L_D}$$

וזו סתירה. לכן $\overline{L_D} \notin R$ ■

מסקנה 4.10 לא השתמשנו בשום שלב בעצירה, לכן ההוכחה תקפה גם עבור RE לכן -

$$\overline{L_D}, \overline{HP}, \overline{L_u} \notin RE$$

מסקנה 4.11 ולכן -

$$L_D, HP, L_u \in RE \backslash R$$

דוגמא

$$L = \{\langle M \rangle \,|\, M \text{ stops for all input } x\}$$

טענה 4.12

$$L \notin R$$

הוכחה: ע"פ משפט הרדוקציה מספיק לקחת שפה אחרת שאינה ב-R, ולהראות רדוקציה ממנה ל-L. נראה רדוקציה מ-HP ל-L.
נתאר פונקציה f מתאימה -

$$f(\langle M \rangle, \langle x \rangle) = \langle M_x \rangle$$

אם הקלט חוקי (כאשר אם הקלט לא חוקי, פלוט קידוד לא חוקי).

נרצה שיתקיים - M_x עוצרת לכל קלט $\Leftrightarrow M$ עוצרת על x

M_x תפעל על כל קלט w באופן הבא -

- הרץ את M על x (כלומר - מחק את w, כתוב את x, הרץ את M).

נשים לב שמתקיים -

- f מלאה

- f ניתנת לחישוב
- <u>תקפות</u> -

$$\langle M\rangle, \langle x\rangle \in HP$$

אמ"מ M עוצרת על x (הגדרת HP) ומהבניה -
זה מתקיים אמ"מ M_x עוצרת לכל קלט. זה מתקיים אמ"מ -

$$\langle M_x\rangle \in L$$

מהגדרת L, ולסיום (מהגדרת f) זה מתקיים אמ"מ -

$$f(\langle M\rangle, \langle x\rangle) \in L$$

■

דוגמא 2

$$L_{\Sigma^*} = \{\langle M\rangle \,|\, L(M) = \Sigma^*\}$$

טענה 4.13 $L_{\Sigma^*} \notin R$

הוכחה: מספיק להוכיח $L_u \leq L_{\Sigma^*}$. ניתן לחזור בדיוק על הרדוקציה הקודמת, ולהחליף כל מופע של HP ב-L_u וכל מופע של L ב-L_{Σ^*}. ■

דוגמא 3

$$L_{EQ} = \{\langle M_1\rangle, \langle M_2\rangle \,|\, L(M_1) = L(M_2)\}$$

טענה 4.14 $L_{EQ} \notin R$

הוכחה: ע"י רדוקציה $L_{\Sigma^*} \leq L_{EQ}$ -
נגדיר את הפונקציה f באופן הבא -

$$f(\langle M\rangle) = \langle M\rangle, \langle M_0\rangle$$

כאשר M_0 מ"ט כך ש-$L(M_0) = \Sigma^*$ (למשל - מכונה שעל כל קלט מיד עוברת למצב מקבל).
מתקיים -

- f מלאה
- f ניתנת לחישוב

- תקפות -

$$\Leftrightarrow \langle M\rangle \in L_{\Sigma^*}$$
$$\Leftrightarrow L(M) = \Sigma^*$$
$$\Leftrightarrow L(M) = L(M_0)$$
$$\Leftrightarrow \langle M\rangle, \langle M_0\rangle \in L_{EQ}$$
$$f(\langle M\rangle) \in L_{EQ}$$

■

4.2 תכונות של שפות

הגדרה 4.15 תכונה של שפות S היא $S \subseteq RE$

הגדרה 4.16 תכונה נקראת לא טרוויאלית אם $\emptyset \neq S \neq RE$
ונסמן -

$$L_S \triangleq \{\langle M\rangle | L(M) \in S\}$$

דוגמאות -

$$S_1 = \{L \in RE | \varepsilon \in L\}$$

$$S_2 = \{L | L \text{ is finite}\}$$

$$S_3 = \{\Sigma^*\}$$

$$L_{S_3} = L_{\Sigma^*}$$

אבחנה: אם S טרוויאלית $\Leftarrow L_S \in R$
בדיקה:
אם $S = \emptyset$

$$L_S = \{\langle M\rangle | L(M) \in \emptyset\} = \emptyset \in R$$

אם $S = RE$

$$L_S = \{\langle M\rangle | L(M) \in RE\} = \Sigma^* \in R$$

משפט 4.17 משפט $Rice$
לכל S לא טרוויאלית $L_S \notin R$

הוכחה:

<u>מקרה א' - $\emptyset \notin S$</u>

מספיק להראות $HP \leq L_S$.

S לא טרוויאלית, בפרט קיימת $L_0 \in S$ ו-$S \subseteq RE$ $\Leftarrow$ קיימת M_0 עבור L_0.

נראה את הרדוקציה -

$$f\left(\langle M\rangle, \langle x\rangle\right) = \langle M_x\rangle$$

התקפות המבוקשת -
$L\left(M_x\right) \in S \Leftrightarrow M$ עוצרת על x
$L\left(M_x\right) = L_0 \Rightarrow M$ עוצרת על x
$L\left(M_x\right) = \emptyset \Rightarrow M$ לא עוצרת על x

M_x תפעל על קלט w באופן הבא -

- הרץ את M על x
- הרץ את M_0 על w

<u>אבחנה</u>

$$L\left(M_x\right) = \begin{cases} \emptyset \notin S & M \text{ does not stop on } x \\ L_0 \in S & M \text{ stops on } x \end{cases}$$

ולכן מתקיים -

$\langle M\rangle, \langle x\rangle \in HP$ $\Leftrightarrow$ (מהגדרת HP)

M עוצרת על x $\Leftrightarrow$ (מהאבחנה)

$L\left(M_x\right) \in S$ $\Leftrightarrow$ (מהגדרת L_s)

$\langle M_x\rangle \in L_S$ $\Leftrightarrow$ (מהגדרת f)

$f\left(\langle M\rangle, \langle x\rangle\right) \in L_S$

והוכחנו את מקרה א'.

<u>מקרה ב' - $\emptyset \in S$</u>

<u>הוכחה 1</u>

S לא טרוויאלית $\Leftarrow$ קיימת $L_0 \in RE \backslash S$ וקיימת M_0 מתאימה

אותה רדוקציה בדיוק:
אם M עוצרת על x אז $L\left(M_x\right) = L_0 \notin S$
אם M לא עוצרת על x אז $L\left(M_x\right) = \emptyset \in S$

אותה רדוקציה בדיוק מהווה -

$$\overline{HP} \leq L_S$$

וכיוון ש-$\overline{HP} \notin R$ נקבל $L_S \notin R$.

<u>הוכחה 2</u>

נתבונן בתכונה המשלימה -

$$\overline{S} = RE \backslash S$$

$\overline{S}$ לא טרוויאלית, ובנוסף $\emptyset \notin \overline{S}$, ממקרה א' נקבל $L_{\overline{S}} \notin R$ אבל -

$$L_{\overline{S}} = \overline{L_S}$$

וכיוון ש-R סגורה למשלים נקבל כי - $L_S \notin R$.

■

דוגמאות

1.

$$L_\varepsilon = \{\langle M\rangle \mid \varepsilon \in L(M)\}$$

טענה 4.18 $L_\varepsilon \notin R$

הוכחה: התכונה המתאימה היא S_1. נשים לב ש־$L_{S_1} = L_\varepsilon$ ו־S_1 לא טרוויאלית ($\{\varepsilon\} \in S_1,\ \emptyset \notin S_1$). לכן ממשפט רייס $L_\varepsilon \notin R$. ■

2.

$$L_2 = \{\langle M\rangle \mid L(M) \text{ is finite}\}$$

על ידי S_2 ניתן להראות כי $L_2 \notin R$.

3.

$$L_{\Sigma^*} = L_{S_3} \notin R$$

4. דוגמא מעניינת נוספת ־

$$L_\emptyset = \{\langle M\rangle \mid L\langle M\rangle = \emptyset\}$$

והתכונה המתאימה ־ $S = \{\emptyset\}$

4.2.1 דרכים להוכחת טענות מהצורה $L \notin R$

- ישירה ($\overline{L_D}$)
- משפט הרדוקציה
- משפטים כלליים (כרגע ־ רייס)

4.2.2 דרכים להוכחת טענות מהצורה $L \notin RE$[8]

- ישירה ($\overline{L_D}$)
- משפט הרדוקציה
- $L \notin RE \Leftarrow \begin{cases} L & \notin R \\ \overline{L} & \in RE \end{cases}$
- משפטים כלליים (כרגע ־ רייס)

דוגמא $L_\varnothing = \{\langle M\rangle \,|\, L(M) = \varnothing\}$

טענה 4.19 $L_\varnothing \notin RE$

[8] הרצאה שישית

הוכחה: נראה כי מתקיים

1. $L_\varnothing \notin R$ (ממשפט רייס, עם $S = \{\varnothing\}$)
2. $\overline{L_\varnothing} \in RE$

ולכן נקבל $L_\varnothing \notin RE$.

נתאר מ"ט עבור $\overline{L_\varnothing}$ על קלט $\langle M \rangle$:

- עבור $i = 1, 2, 3 \ldots$ -
  - הרץ את M על i המילים הראשונות (לפי סדר לקסיקוגרפי), כל אחת למשך i צעדים. אם M קיבלה את אחת המילים - עצור וקבל.

אנליזה

- אם $L(M) = \varnothing$ - המכונה לעולם לא תעצור, ובפרט לא תקבל את M.
- אם $L(M) \neq \varnothing \Leftarrow$ קיימת $w \in L(M)$.

נסמן ב-j את מיקומה של w בסדר הלקסיקוגרפי, נסמן ב-k את מספר הצעדים הדרוש ל-M כדי לקבל את w. ונסמן - $t = \max\{j, k\}$.

- אם $M_\varnothing$ לא מגיעה לאיטרציה ה-t, כלומר היא עוצרת לפני כן (כי כל איטרציה היא סופית), ולכן עצרה וקיבלה.
- אחרת - $M_\varnothing$ כן מגיעה לאיטרציה ה-t. לכן w היא אחת המילים שתבדקנה, ו-M תרוץ $t \geq k$ צעדים. לכן M תקבל את w ב-$M_\varnothing$ תעצור ותקבל את M.

$$L_\varnothing \notin RE \;\Leftarrow\; L(M_\varnothing) = \overline{L_\varnothing} \Leftarrow$$

■

דוגמא $L_{\Sigma^*} = \{\langle M \rangle \mid L(M) = \Sigma^*\}$

טענה 4.20 $L_{\Sigma^*}, \overline{L_{\Sigma^*}} \notin RE$

הוכחה:

1. ראינו רדוקציה - $HP \leq L_{\Sigma^*} \Leftarrow \overline{HP} \leq \overline{L_{\Sigma^*}}$

$HP \leq L_{\Sigma^*}$ לכן קיימת $f(\langle M \rangle, \langle x \rangle) = \langle M_x \rangle$
M_x על קלט w -
- הרץ את M על x
- קבל
מתקיים - M עוצרת על $x \Leftrightarrow L(M_x) = \Sigma^*$

2. $\overline{HP} \leq L_{\Sigma^*} \equiv HP \leq \overline{L_{\Sigma^*}}$

נרצה פונקציה $f(\langle M \rangle, \langle x \rangle) = \langle M_x \rangle$, כאשר התקפות המבוקשת -
M לא עוצרת על $x \;\Leftrightarrow\; L(M_x) = \Sigma^*$
אופן פעולת M_x על קלט w -

- הרץ את M על x למשך $|w|$ צעדים
- אם M לא עצרה על x (בזמן הנתון) קבל. אחרת - דחה.

אבחנה -

- אם M עוצרת על x ב־k צעדים - $L(M_x) = \Sigma^{<k}$ (כל המילים מעל א"ב Σ שאורכן קטן מ־k)
- אם M לא עוצרת על x - $L(M_x) = \Sigma^*$

תקפות M לא עוצרת על x (מהגדרת HP) $\Leftrightarrow (\langle M\rangle, \langle x\rangle) \in \overline{HP}$

$\Leftrightarrow L(M_x) = \Sigma^*$ (אבחנה)

$\Leftrightarrow \langle M_x\rangle \in L_{\Sigma^*}$ (מהגדרת L_{Σ^*})

$\Leftrightarrow f(\langle M\rangle, \langle x\rangle) \in L_{\Sigma^*}$ (מהגדרת f)[9]

■

דוגמא $L_{eq} = \{\langle M_1\rangle, \langle M_2\rangle \,|\, L(M_1) = L(M_2)\}$

טענה 4.21 $L_{Eq}, \overline{L_{Eq}} \notin RE$

הוכחה: ראינו

$$\begin{array}{rcl} L_{EQ} \notin RE & \Leftarrow & L_{\Sigma^*} \leq L_{EQ} \\ & & \Downarrow \\ \overline{L_{EQ}} \notin RE & \Leftarrow & \overline{L_{\Sigma^*}} \leq \overline{L_{EQ}} \end{array}$$

■

משפט 4.22 <u>משפט *Rice* לשפות RE</u>
תהא S תכונה לא טרוויאלית המקיימת $\varnothing \in S$. אזי - $L_S \notin RE$ (תזכורת - $L_S = \{\langle M\rangle \mid L(M) \in S\}$)

הוכחה: בהוכחת משפט Rice מקרה ב' -

$$L_S \notin RE \Leftarrow \overline{HP} \leq L_S$$

(ממשפט הרדוקציה) ■

הערה 4.23 התנאי הנוסף הכרחי (למשל $\overline{L_\varnothing} \in RE$)

הערה 4.24 התנאי הנוסף איננו אפיון ($L_{\Sigma^*} \notin RE$)

הערה 4.25 קיים תנאי המהווה איפיון, לא נעבור על הגרסה הזו בהרצאה.

5 סוגים של בעיות חישוב

- בעיות הכרעה/שפות
- חישוב פונקציה
- בעיות חיפוש/יחסים

עד עכשיו דיברנו על בעיות הכרעה או שפות, כעת נעבור לדבר על פונקציות.

[9] צריך לטעון גם ש־f מלאה וניתנת לחישוב על מנת שההוכחה תהיה מלאה.

תזכורת פונקציה $f:\Sigma^*\to\Gamma^*$ (מלאה או חלקית) נקראת ניתנת לחישוב אם קיימת מ"ט M כך ש־$f=f_M$.

סימון $L_f\triangleq\{(x,y)\mid y=f(x)\}$

משפט 5.1 f ניתנת לחישוב $\Leftrightarrow L_f\in RE$

הערה 5.2 אם בנוסף f מלאה, אזי f ניתנת לחישוב $\Leftrightarrow L_f\in R$

הערה 5.3 קיימת f לא מלאה כך ש־$L_f\in R$
למשל - f שאינה מוגדרת לשום קלט $L_f=\varnothing$

הוכחה: שני כיוונים -

- כיוון אחד - f ניתנת לחישוב $\Leftarrow$ קיימת מ"ט M_f המחשבת אותה.
  נבנה M עבור L_f המוגדרת כך שעל קלט (x,y) היא מבצעת -
  - הרץ את M_f על x
  - נסמן ב־$f(x)$ את הפלט של הצעד הקודם (אם הצעד הקודם הסתיים)
  - אם $y=f(x)$ קבל.
  - אחרת - דחה.

  נכונות -
  - $(x,y)\in L_f\Leftarrow y=f(x)\Leftarrow M$ מקבלת
  - $\Leftarrow(x,y)\notin L_f$
    - $f(x)$ מוגדר אך שונה מ־$y\Leftarrow M$ דוחה
    - או - $f(x)$ לא מוגדר $\Leftarrow M$ לא עוצרת

  $\Leftarrow L_f\in RE$

- כיוון שני - $L_f\in RE$ ו־M מ"ט המקבלת אותה
  נבנה M_f לחישוב f (עבור קלט x נרצה לחפש $(x,f(x))\in L_f$).
  M_f תבצע על קלט x -
  - עבור $i=1,2,3\ldots$ -
    - הרץ את M על זוגות $(x,w_1),(x,w_2),\ldots(x,w_i)$ כל אחד במשך i צעדים. אם (x,w_j) כלשהו התקבל - עצור עם פלט w_j.

  נכונת -
  - אם $f(x)$ לא מוגדר $\Leftarrow M_f$ לא עוצרת
  - אחרת, $f(x)$ מוגדר $\Leftarrow$ קיים j כך ש־$f(x)=w_j$, וגם קיים k - מספר הצעדים הדרוש ל־M כדי לקבל את הזוג $(x,w_j)\Leftarrow$ באיטרציה ה־$t=\max\{k,j\}$ המכונה M_f בהכרח תעצור. [10]

■

דוגמא $f(\langle M\rangle)=\begin{cases}|L(M)| & |L(M)|<\infty\\ \text{undefined} & \text{Otherwise}\end{cases}$

[10] כיוון ש־f היא פונקציה וקיים רק $f(x)$ אחד (אם בכלל) - במקרה זה אין אופציה שהמכונה תעצור לפני האיטרציה ה־t, בניגוד להוכחה הקודמת בה השתמשנו בהרצה מבוקרת.

טענה 5.4 f לא ניתנת לחישוב

הוכחה: (ישירה)
אילו f ניתנת לחישוב $\Leftarrow L_{\varnothing} \in RE$ אם קיימת M_f המחשבת את f .
נתאר את פעולת $M_{\varnothing}$ על קלט $\langle M \rangle$ -

- הרץ את M_f על $\langle M \rangle$
- נסמן ב־y את הפלט. אם $y = 0$ קבל, אחרת - דחה.

נכונות

- $L(M) = \varnothing \Leftarrow f(\langle M \rangle) = 0 \Leftarrow M_{\varnothing}$ מקבלת.
- $L(M)$ סופית אך לא ריקה $\Leftarrow f(\langle M \rangle)$ מוגדר אבל לא אפס $\Leftarrow M_{\varnothing}$ דוחה.
- $L(M)$ אינסופית $\Leftarrow f(\langle M \rangle)$ לא מוגדר $\Leftarrow M_{\varnothing}$ לא עוצרת.

■

הוכחה: (דרך שנייה, ע"י המשפט)

$$L_f = \{\langle M \rangle, k \mid k = |L(M)| \ \ k \in \mathbb{N}\}$$

ע"י המשפט - מספיק להוכיח $L_f \notin RE$ לכן מספיק להוכיח $L_{\varnothing} \leq L_f$, וזאת ניתן לעשות ע"י

$$g(\langle M \rangle) = \langle M \rangle, 0$$

■

דוגמאות

- $f_1(\langle M \rangle, \langle x \rangle) = \begin{cases} 1 & M \text{ stops on } x \\ 0 & \text{Otherwise} \end{cases} \Leftarrow$ לא ניתנת לחישוב, אם כן - $HP \in R$
- $f_2(\langle M \rangle, \langle x \rangle) = \begin{cases} 1 & M \text{ stops on } x \\ \text{undefined} & \text{Otherwise} \end{cases} \Leftarrow$ ניתנת לחישוב, כמו ההוכחה ש - $HP \in RE$
- $f_3(\langle M \rangle, \langle x \rangle) = \begin{cases} \text{undefined} & M \text{ stops on } x \\ 0 & \text{Otherwise} \end{cases} \Leftarrow$ לא ניתנת לחישוב, אם כן - $\overline{HP} \in RE$

5.1 בעיות חיפוש/יחסים

דוגמא

$$S = \{(G,T) \,|G \text{ - graph}, T \text{ - spanning tree in } G\}$$

בעיית הזיהוי של S - נתון (G,T), האם הוא מקיים את היחס.
בעיית החיפוש - נתונן גרף G - מצא עץ פורש ב־G, אם קיים אחד.

הגדרה 5.5 בהנתן $S \subseteq \Sigma^* \times \Sigma^*$ אומרים ש<u>בעיית הזיהוי</u> של S <u>ניתנת לפתרון</u> אם $S \in RE$

הגדרה 5.6 בהנתן $S \subseteq \Sigma^* \times \Sigma^*$ אומרים ש<u>בעיית החיפוש</u> של S <u>ניתנת לפתרון</u> אם קיימת מ"ט M כך שלכל x -

- אם קיים y כך ש-$(x,y) \in S$ - M עוצרת עם y כזה כפלט (אם יש כמה y-ים, אפשר להחזיר כל אחד מהם)
- אם לא קיים y כזה, אם M לא עוצרת

מקרה פרטי בנהתן פונקציה f -

$$S_f = L_f = \{(x, f(x))\}$$

שאלות

- האם S ניתנת לזיהוי $\overset{?}{\Leftarrow}$ S ניתנת לחיפוש (נכון!)
- האם S ניתנת לחיפוש $\overset{?}{\Leftarrow}$ S ניתנת לזיהוי (לא נכון!, למשל $S_{EQ} = L_{EQ}$, בבעיית החיפוש אפשר להחזיר תמיד את אותה מכונה M_1, אבל אנחנו יודעים שבעיית הזיהוי היא לא ב-RE).

דוגמאות[11]

- $S_1 = L_u = \{(\langle M \rangle, \langle x \rangle) \mid M \text{ accepts } x\}$
  - ניתנת לזיהוי.
  - ניתנת לחיפוש (זיהוי גורר חיפוש. לחילופין - ניתן פשוט לעבור על כל הקלטים בהרצה מבוקרת עד שמוצאים אחד שמתקבל).
- $S_2 = \overline{L_u} = \{(\langle M \rangle, \langle x \rangle) \mid M \text{ does not accept } x\}$
  - לא ניתנת לזיהוי (ראינו)
  - האם היא ניתנת לחיפוש?

  אנחנו יודעים כי $\overline{L_{\Sigma^*}} \notin RE$, כלומר אוסף כל ה-$\langle M \rangle$ כך שיש קלט שאינו בשפה אינו ב-RE, וזה למעשה מה שאנחנו מתיימרים לפתור באמצעות בעיית החיפוש. כלומר - S_2 לא ניתנת לחיפוש.

5.2 דוגמא - סיבוכיות קולמגורוב

קולמגורוב שאל את השאלה, מדוע סדרה כמו זו -

000000000000000

לא נראית לנו אקראית במיוחד, בעוד שסדרה כזו -

001001001001001

נראית מעט יותר אקראית, והסדרה הזו -

011101100010010

נראית הכי אקראית מבין שלושתן. אינטואיטיבית זה נראה לנו פשוט, אבל איך ניתן להגדיר "אקראיות"?

[11] הרצאה שביעית 27/4/10

הגדרה 5.7 $\Sigma = \{0,1\}\ \Gamma = \{0,1,\text{ב}\}$
עבור מחרוזת $x \in \Sigma^*$, סיבוכיות קולמגורוב $k(x)$ היא מספר המצבים הקטן ביותר של מ"ט (שעל קלט ε) פולטת את x.

משפט 5.8 הפונקציה k לא ניתנת לחישוב

למה 5.9 (k פונקציה מלאה) $k(x) \leq |x| + 1$.

הוכחה: נגדיר מ"ט עם מצבים $q_1, q_2, q_3 \ldots q_{n+1}$ (כאשר $x = x_1 x_2 \ldots x_n$) ונגדיר -

$$\delta(q_i, \text{ב}) = (q_{i+1}, x_i, R)$$

כלומר המצב ה-i פשוט כותב את האות ה-i על הסרט ועובר למצב הבא. ■

למה 5.10 (k לא חסומה) לכל t טבעי קיים x כך ש-$k(x) > t$. **הוכחה:** מספר המכונות עם t מצבים הוא סופי[12]. לכל מכונה יש (לכל היותר) פלט יחיד על ε, לכן מספר המחרוזות עם $k(x) \leq t$ הוא סופי, ויש כמובן מספר אינסופי של מחרוזות. ■

הוכחה: למשפט 5.8 (משפט קולמגורוב)
נניח בשלילה כי k ניתנת לחישוב, ו-M_k מ"ט שמחשבת אותה (עוצרת תמיד). נגדיר -

1. M_1 על קלט t ופולטת x כך ש-$k(x) > t$:
   עבור על המחרוזת x ע"פ סדר לקסיקורגפי, חשב $k(x)$ ע"י M_k עד שתמצא x מתאים.
   מובטח ש-M_1 עוצרת ומוצאת x כזה.
   נסמן m - מספר המצבים של M_1, ונבחר n מספיק גדול כך ש-$2^n - n \geq m + 3$. ⋆
2. M_2 על קלט ε עובדת כך -
   - רושמת על הסרט 2^n (1 ואחריו n אפסים) (ניתן לבצע ב-$n+2$ מצבים)
   - מחזירה את הראש לתחילת הסרט (ניתן לבצע במצב אחד)
   - ממשיכה כמו M_1 (על קלט 2^n) (ממומשת ב-m מצבים)

   נסמן את הפלט ב-x.
   מספר המצבים של M_2 - $n + 3 + m$

מצד אחד - x הוא הפלט של M_1 על 2^n, לכן $k(x) > 2^n$
מצד שני - x הוא הפלט של M_2 על ε, לכן $k(x) \leq n + 3 + m \leq 2^n$ (בגלל ⋆)
קיבלנו סתירה להנחה שקיימת M_k ■

[12] מספר המכונות מקיים קטן או שווה ל-$(9t)^{3t}$ כי ל-δ יש $3t$ איברים בתחום ו-$9t$ בטווח

חלק II

חלק ב' של הקורס

נתבונן על הבעיות שניתן לפתור (ע"פ חלק א'). נרצה לזהות בתוכן את הבעיות שניתנות לפתרון יעיל (במונחים של **זמן**, או זכרון).

כמו שכבר ראינו (במבוא, מבנה וכו'... בכל פעם שדיברנו על סיבוכיות) מפתה אולי למדוד יעילות לפי הזמן שלוקח לנו להריץ תוכנה - אבל זה לא מדד טוב, כיוון שהוא תלוי מימוש, תלוי חומרה שעליה אנחנו רצים וכו'... לא ניתן להשתמש במדד כזה עבור הגדרה תיאורטית. אנחנו נשתמש במדד אחר - כמה צעדים דרושים למ"ט על מנת לבצע את החישוב.

הגדרה 5.11 סיבוכיות הזמן 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 (אחרת, אם M לא עוצרת על x, $t_M(x)$ לא מוגדרת).

הגדרה 5.12 חסם סיבוכיות עבור מ"ט M הוא פונקציה $T : \mathbb{N} \mapsto \mathbb{N}$ המקיימת לכל x -

$$t_M(x) \leq T(|x|)$$

נתעניין בשאלה - איזה חסם סיבוכיות יחשב כיעיל?

מה נרצה?

- "התאמה למציאות" - מה שאנחנו מגדירים כיעיל יחשב כיעיל גם ע"י ההגדרה, ולהפך.
- "נוחות מתמטית".
- חסם סיבוכיות "גדול מדי" (למשל - $O(2^{2^n})$, $O(2^n)$) הוא כמובן לא רצוי.
- חסם נמוך מדי לא אפשרי. דרוש לפחות $T(n) \geq n$ (כי, למשל, $O(log(n))$, $O(\sqrt{n})$ לא מאפשרים אפילו לעבור על כל הקלט).

6 חישוב יעיל

הגדרה 6.1 מ"ט M תקרא פולינומית או יעילה אם קיים פולינום $p(n) = O(n^c)$ שמהווה חסם סיבוכיות עבורה.

יתרונות ההגדרה

1. "התאמה למציאות" - כמעט כל אלגוריתם שמריצים ב"עולם האמיתי" מתאים להגדרה.
2. עמידות - מושג הפולינומיות לא רגיש למודל (מ"ט רגילה - שקולה למ"ט דו־סרטית, שקולה למכונה עם RAM וכו'...)
3. פולינומיות מקיימת תכונות סגור -
   דוגמא - f, g ניתנות לחישוב יעיל אז גם - $h(x) = f(g(x))$ ניתנת לחישוב יעיל.

   הוכחה: f ניתנת לחישוב יעיל $\Leftarrow$ קיימת M_f המחשבת אותה בזמן $O(n^c)$.
   g ניתנת לחישוב יעיל $\Leftarrow$ קיימת M_g המחשבת אותה בזמן $O(n^d)$.
   נבנה מ"ט M_h המחשבת את h באופן הבא -
   - חשב את $y = f(x)$ (ע"י M_f)
   - חשב את $z = g(y)$ (ע"י M_g) ופלוט אותו.

נכונות - ברור.

סיבוכיות של M_h -

- צעד 1 - $O\left(n^c\right)$
- צעד 2 - $O\left(|y|^d\right) = O\left(|n^c|^d\right) = O\left(n^{cd}\right)$

בסך הכל - $O\left(n^{cd}\right)$

■

חסרונות ההגדרה

1. התאמה לא מושלמת למציאות.

(א) במציאות נעדיף $n^{\log\log n}$ על פני n^{1000}, למרות שהראשון לא פולינומי והשני כן.

(ב) סימלפקס - אלגוריתם שממומש בהרבה תוכנות במציאות, ובמקרים מסויימים רץ בזמן אקספוננציאלי. אבל פרקטית כמעט ולא נתקלים בקלטים כאלו אלא רק בקלטים עליהם האלגוריתם יעיל.

הגדרה 6.2 <u>בעיות הניתנות לפתרון יעיל</u> -

$$P \triangleq \{L \subseteq \Sigma^* | \text{ exists polynomial TM for } L \}$$

$$POLY \triangleq \{f : \Sigma^* \to \Gamma^* | \text{ exists polynomial TM which computes } f \}$$

אבחנות

- $P \subseteq R$
- $f \in POLY \Leftarrow f$ מלאה, יתר על כן f "חסומה פולינומיאלית". כלומר קיים פולינום p כל ש-$|f(x)| \leq p(|x|)$.

6.1 קשר בין פונקציות לשפות

תזכורת

בהנתן פונקציה f מלאה הגדרנו $L_f = \{(x,y) | y = f(x)\}$ והוכחנו כי -

$L_f \in R \Leftrightarrow f$ ניתנת לחישוב

נרצה לדעת האם קיים קשר דומה בין פונקציה f הניתנת לחישוב יעיל לבין העובדה ש-$L_f \in P$.
$\Leftarrow$ בהנתן (x,y) חשב $f(x)$ והשווה ל-y (יעיל)
$\Rightarrow$ עוברים על כל ה-y-ים, בודקים האם $(x,y) \in L_f$, עד שמוצאים y מתאים.
יש בעיות -

- הכיוון השני לא בהכרח יעיל. מספר ה-yים שנצטרך לבדוק עלול להיות אקספוננציאלי באורך של x אפילו אם ידוע כי y חסום על ידי הגודל של x.
- פונקציה כמו $f(x) = 1^{2^{|x|}}$ - אין מספיק "זמן" כדי לכתוב את הפלט ביעילות, אפילו שהפונקציה "פשוטה" (עבור הכיוון הראשון הקלט הוא (x,y) ואז הבדיקה היא פולינומית בקלט, עבור הסעיף השני קלט הוא רק x, לכן ייצור הפלט לא יכול להתבצע בזמן פולינומי בקלט).
- אפילו אם f חסומה פולינומיאלית - לא ידוע (אם המשפט נכון?).

הגדרה 6.3 $L'_f = \{(x,y) \mid y \text{ is prefix of } f(x)\ \}$

משפט 6.4 $f \in POLY \Leftrightarrow L'_f \in P$ חסומה פולינומיאלית וגם f.

הוכחה: כיוון ראשון ($\Leftarrow$) טרוויאלי.
כיוון שני $\Rightarrow$ -
נתאר את פעולת M_f על קלט x -

- אתחול - $y = \varepsilon$
- איטרציה - עבור כל $a \in \Gamma$ בדוק האם $(x, ya) \in L'_f$ -
  - אם כן - $y \leftarrow ya$ ונתחיל איטרציה חדשה
  - אם לא - ננסה את האות הבאה
  - אם ניסינו את כולם - עצור עם פלט y

<u>מתקיים</u>

- באיטרציה ה-i מתקיים - $y = y_1 y_2 \dots y_{i-1}$ (ומכאן נובעת הנכונות)
- סיבוכיות -
  - מספר האיטרציות - פולינומי (כי f חסומה פולינומיאלית)
  - בכל איטרציה החישוב פולינומי

■

6.2 בעיות חיפוש[13]

הגדרה 6.5 יהא $S \subseteq \Sigma^* \times \Sigma^*$ יחס דו מקומי -

- אומרים ש<u>בעיית הזיהוי</u> של S ניתנת לפתרון יעיל אם $S \in P$
- אומרים ש<u>בעיית החיפוש</u> של S ניתנת לפתרון יעיל אם קיימת מ"ט פולינומית M כך שלכל $x \in \Sigma^*$ -
  - אם קיים y כך ש-$(x,y) \in S$ אז M עוצרת ב-q_A עם פלט y כזה
  - אחרת (לא קיים y כזה) M עוצרת ב-q_R (ואין חשיבות לפלט)

למה 6.6 חיפוש יעיל <u>לא גורר</u> זיהוי יעיל
לדוגמא - $L_{EQ} = \{(\langle M_1 \rangle, \langle M_2 \rangle) \mid L(M_1) = L(M_2)\}$

- בעיית הזיהוי קשה, ראינו כי $L_{EQ} \notin RE \Leftarrow L_{EQ} \notin P$
- בעיית החיפוש קלה - בהנתן $\langle M_1 \rangle$ פלוט את $\langle M_1 \rangle$ (דורש זמן לינארי $O(n)$).

6.2.1 האם זיהוי יעיל גורר חיפוש יעיל?

הגדרה 6.7 יחס S הוא חסום פולינומיאלית אם קיים פולינום P כך שלכל $(x,y) \in S$ מתקיים - $|y| \leq P(|x|)$.

לאור ההגדרה, ננסה לפתור ראשית בעיה פשוטה יותר -

[13] הרצאה שמינית 4/5/10

האם לכל יחס S חסום פולינומיאלי זיהוי יעיל גורר חיפוש יעיל?
לא ניתן לבדוק לכל y האם $(x,y) \in S$ כי מספר ה־y הוא בערך $2^{O(P(n))}$.
השאלה הזו היא השאלה הפתוחה המרכזית של מדעי המחשב (נוסח 1)

6.3 מ"ט אי דטרמיניסטית

הגדרה 6.8 מ"ט אי דטרמיניסטית (א"ד) מוגדרת כמו מ"ט רגילה (דטרמיניסטית) למעט הגדרת פונקציית המעברים שתוגדר -

$$\delta : (Q \backslash F) \times \Gamma \mapsto (Q \times \Gamma \times \{L, R, S\})^2$$

כאשר המשמעות היא שאם -

$$\delta(q,a) = \{(p_0, b_0, d_0), (p_1, b_1, d_1)\}$$

המכונה מבצעת אחד מהשניים - (p_0, b_0, d_0) או (p_1, b_1, d_1).

המשמעות היא שכעת במקום מסלול חישוב נקבל "עץ חישוב", שהמכונה יכולה להגיע לכל עלה שלו. לכן נשאלת השאלה - **מה בעצם המכונה עושה?**

הגדרה 6.9 אומרים שמ"ט א"ד M מקבלת את הקלט x אם קיים מסלול בעץ החישוב של M על x שמסתיים ב־q_A. ונסמן כרגיל -

$$L(M) = \{x |\ M \text{ accepts } x\}$$

טענה 6.10 מודל מ"ט א"ד שקול (במובן של שפות) למודל מ"ט דטרמיניסטית. (ההוכחה המלאה בתרגול)

רעיון ההוכחה -

- בהנתן מ"ט דטר' היא מקרה פרטי של מ"ט א"ד (אפשר להגדיר את שתי האפשרויות באופן זהה).
- בכיוון השני - בהנתן מ"ט א"ד M נראה שניתן לחשב פרדיקט $M(x,w)$ ששווה לאחד אמ"מ M מקבלת את x במסלול w. וברגע שיש פרדיקט כזה אפשר לחפש w כך ש־$M(x,w) = 1$ (אם קיים כזה - נמצא ונקבל...).

הגדרה 6.11 מ"ט א"ד M תקרא פולינומית (או - יעילה) אם קיים פולינום $P(n)$ כך ש־M עוצרת על x תוך $p(|x|)$ צעדים בכל מסלוליה.

(נשים לב שלמרות שהחסם על עומק העץ הוא פולינומי, גודל העץ עדיין אקספוננציאלי)

הגדרה 6.12 NP (הגדרה 1)

$$NP \triangleq \{L |\ \text{Exists Non-Deterministic TM for } L\ \}$$

השאלה הפתוחה המרכזית (נוסח 2) - האם $P = NP$?

הגדרה 6.13 NP (הגדרה 2)
שפה $L \in NP$ אם קיים יחס דו מקומי R_L המקיים -

1. R_L חסום פולינומיאלית
2. R_L ניתן לזיהוי יעיל (על ידי מכונה דטרמיניסטית)
3. $L = \{x |\ \exists y\ (x,y) \in R_L\}$

דוגמא השפה $L = \{x|\ x \text{ is not a prime numer (composite) }\}$ כלומר - L היא שפת המספרים הפריקים.

טענה 6.14 $L \in NP$ (לפי הגדרה 2)

הוכחה: צריך להראות יחס R_L מתאים, נגדיר -

$$R_L = \{(x,y) \mid 1 < y < x \ \wedge \ y|x \ \}$$

1. היחס חסום פולינומיאלית ($y < x$)
2. ניתן לזיהוי יעיל
3. מתוך הגדרת מספר פריק מתקיים - $L = \{x|\ \exists y \ \ (x,y) \in R_L\}$.

לכן $L \in NP$ ע"פ הגדרה 2.

■

טענה 6.15 $L \in NP$ (לפי הגדרה 1)

הוכחה: נציג מ"ט א"ד פולינומית עבור L -
M תפעל על x באופן הבא -

- "נחש" y באורך של x
- בדוק האם $1 < y < x$ והאם $y|x$
  - אם כן - קבל
  - אחרת - דחה

<u>נכונות</u> -

- אם x פריק $\Leftarrow$ (לפי הגדרה)
  קיים $1 < y < x$ כך ש-$y|x$ $\Leftarrow$ (מהבנייה)
  קיים מסלול שבו "מנחשים" את ה-y הזה $\Leftarrow$ (מהבנייה)
  מקבלים $\Leftarrow$
  קיים מסלול מקבל של M על x, ולכן לפי הגדרת $L(M)$ (הגדרה 6.9) $x \in L(M)$
- אם x לא פריק $\Leftarrow$ (לפי הגדרה)
  לא קיים y כך ש-$1 < y < x$ ו-$y|x$ $\Leftarrow$ (מהבנייה)
  לא קיים מסלול מקבל (כל מסלול דוחה) $\Leftarrow$
  $x \notin L(M)$

לכן - $L = L(M)$.

<u>מימוש ה"ניחוש"</u> (ע"י מ"ט א"ד דו-סרטית)

על הסרט הראשון יהיה הקלט x, על הסרט השני ניצור ניחוש y, לכל תו בקלט x נבחר על הסרט השני האם לכתוב 0 או 1, ונלך צעד אחד ימינה על שני הסרטים עד שניתקל ב-$\flat$ על הסרט הראשון.

פונקציית המעברים -

$$\begin{aligned}\delta(q_0, a, \flat) &= \{(q_0, a, 0, R, R), (q_0, a, 1, RR)\} \\ \delta(q_0, \flat, \flat) &= \{(q_1, \dots)\dots\}\end{aligned}$$

<u>סיבוכיות</u>
ניחוש - לינארי באורך של x
השוואה, וחילוק קל לבצע בזמן פולינומי (באורך של x, y אבל $|y| = |x|$)
בסך הכל - סיבוכיות פולינומית.

ולבסוף - $L \in NP$ לפי הגדרה 1.

■

טענה 6.16 $P \subseteq NP$ -
$L \in P \Leftarrow$ קיימת מ"ט פולינומית M עבור L

הוכחה: (הגדרה 1): M היא מקרה פרטי של מ"ט א"ד. ■

הוכחה: (הגדרה 2): נראה R_L מתאים -

$$R_L = \{(x, \varepsilon) \,|x \in L\}$$

היחס מקיים את שלושת הדרישות... ■

טענה 6.17 $NP \subseteq R$

הוכחה: (הגדרה 1): חזור על ההוכחה מטענת השקילות (טענה 6.10) אך הגבל את $|w| \leq p(|x|)$ (הפולינום עבור M א"ד). ■

הוכחה: (הגדרה 2): $L \in NP \Leftarrow$ קיים R_L המקיים את הדרישות (3-1) בהגדרה.
נבנה M שעוצרת תמיד עבור L. היא תפעל על x באופן הבא -

- עבור על כל ה-y באורך לכל היותר $p(|x|)$
- לכל y כזה בדוק האם $(x, y) \in R_L$ ע"י המכונה שעוצרת תמיד (מובטחת כיוון ש-R_L ניתן לזיהוי יעיל)

קיבלנו מ"ט שעוצרת תמיד, והנכונות נובעת מסעיף 3 בהגדרה ($L = \{x| \ \exists y \ \ (x, y) \in R_L\}$).

■

משפט 6.18 שתי ההגדרות של NP שקולות

הוכחה: שני כיוונים -

- $L \in NP$ לפי הגדרה 2 $\Leftarrow$ קיים R_L כנדרש בהגדרה.
  נראה מ"ט א"ד פולינומיאלית עבור L. M על x -
  - "נחש" y באורך לכל היותר $p(|x|)$ (p הפולינום המובטח מהיות R_L חסום פולי')
  - בדוק האם $(x, y) \in R_L$ (על ידי המכונה M_L מובטחת מהיות היחס ניתן לזיהוי יעיל)
    * אם כן - קבל.

$*$ אחרת - דחה.

<u>נכונות</u>

- אם $x \in L \Leftarrow$ (מהגדרת היחס R_L, ובפרט דרישות 1,3)
  קיים y באורך לכל היותר $p(|x|)$ כך ש-$(x,y) \in R_L \Leftarrow$ (לפי הבנייה)
  קיים מסלול שבו אנחנו מנחשים y כזה $\Leftarrow$ (מהבנייה)
  המכונה תקבל במסלול זה, ולכן $x \in L(M)$
- אם $x \notin L \Leftarrow$ (מהגדרת R_L)
  לא קיים y כנ"ל (בפרט באורך המתאים) $\Leftarrow$ (מהבנייה)
  M דוחה בכל המסלולים $\Leftarrow$
  $x \notin L(M)$

לכן - $L = L(M)$ כנדרש.

<u>סיבוכיות</u>

צעד 1 (ניחוש) - חישוב $p(|x|)$ וניחוש y באורך לכל היותר $p(|x|)$ דורש לכל היותר $O(p(|x|))$ צעדים.

צעד 2 (חישוב) - המכונה פולינומית (הפולינום q) לכן -

$$O(q(|y|+|x|)) = O(q(O(p(|x|))+|x|))$$

וזה עדיין פולינומי ב-x.

- נניח כי L מקיימת את הגדרה 1, ותהא M מ"ט א"ד פולינומית כך ש-p הפולינום שלה. נראה R_L כנדרש -

$$R_L = \{(x,y) \mid M(x,y) = 1 \ \wedge \ |y| \le p(|x|)\}$$

כאן $M(x,y)$ הוא תוצאת החישוב של מ"ט א"ד M על קלט x במסלול המתואר ע"י המחרוזת y.
לכן משמעות הסימון $M(x,y) = 1$ היא ש-M מקבלת את x במסלול y.

- היחס חסום פולינומית
- ניתן לזיהוי יעיל (סימולציה של y צעדים)
- העובדה ש-R_L הוא היחס המתאים נובעת ישירות מההגדרה -
  $x \in L \Leftrightarrow$ קיים מסלול y באורך $p(|x|)$ שבו M מקבלת את $x \Leftrightarrow$ קיים y כך ש-$(x,y) \in R_L$

■

משפט 6.19 שני הניסוחים של הבעיה הפתוחה המרכזית של מדעי המחשב - שקולים. כלומר - $P = NP \Leftrightarrow$ לכל יחס פולינומיאלי זיהוי יעיל גורר חיפוש יעיל.

הוכחה: [14]

- כיוון אחד - נניח כי לכל יחס חסום פולינומאלי זיהוי יעיל גורר חיפוש יעיל, נוכיח כי $P = NP$.
  מספיק להוכיח - $NP \subseteq P$
  תהא $L \in NP$ כלשהי, יהא R_L היחס המובטח (בהגדרה של NP)
  R_L ניתן לזיהוי יעיל, לכן על פי ההנחה, R_L ניתן לחיפוש יעיל ע"י מ"ט יעילה M_L (אבחנה - M_L היא מ"ט פולי' המזהה את L). מתקיים -

[14] הרצאה תשיעית 11/5/10

- $x \in L \Leftrightarrow$ (R_L מהגדרת)
- קיים y כך ש־$(x,y) \in R_L \Leftrightarrow$ (M_L מהגדרת)
- M_L מקבלת את x

לכן $L(M_L) = L$ ־ ולכן $L \in P$

- כיוון שני ־ ננית כי $P = NP$

יהא S יחס חסום פולינומאלית, ניתן לזיהוי יעיל. נוכיח כי S ניתן לחיפוש יעיל.
נגדיר יחס עזר ־

$$S' = \{((x,w),z) \mid (x,wz) \in S\}$$

- S' חסום פולי' (כי S כזה)
- S' ניתן לזיהוי יעיל
- ההבדל ־ בבעיית החיפוש

נגדיר שפה $L_{S'} = \{(x,w) \mid \exists z : ((x,w),z) \in S'\}$. קל לראות (לפי הגדרה) ש־$L_{s'} \in NP$.
מההנחה ־ $L_{s'} \in P$, כלומר ־ קיימת מ"ט M' דטרמיניסטית פולינומיאלית המזהה את $L_{S'}$.
נשתמש ב־M' לפתרון בעיית החיפוש של S, על x נבצע ־

- ע"י M' בדוק האם $(x,\varepsilon) \in L_{S'}$ (כלומר ־ האם יש פתרון שמרחיב את המחרוזת הריקה, כל פתרון הוא כזה לכן למעשה ־ "בדוק האם קיים פתרון")
  - אם לא ־ אין פתרון, עצור ב־q_R.
  - אם כן ־ $y \leftarrow \varepsilon$
- איטרציה ־ לכל $a \in \Gamma$ בדוק האם $(x,ya) \in L_{S'}$
  - אם כן ־ $y \leftarrow ya$
  - אם לא ־ בדוק את האות הבאה. אם בדקנו את כולן ־ עצור ב־q_A עם פלט y.

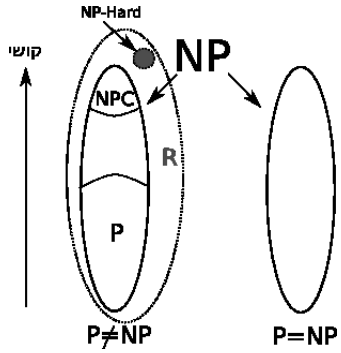
<u>אנליזה</u> (מקוצרת)

- נכונות
  - אם אין פתרון נגלה זאת.
  - אם יש פתרון, מתקיימת האינווריאנטה ־ בתחילת האיטרציה ה־i המחרוזת y היא רישא באורך $i-1$ של פתרון.
- סיבוכיות
  - מספר האיטרציות ־ פולינומי (S חסום פולי')
  - בכל איטרציה מפעילים $O(1)$ פעמים את M' שהיא מ"ט פולינומיאלית
  - סה"כ (מתכונות של פולינומים) ־ פולינומאילי.

■

7 בעיות "קשות" NPC

היחסים בין הקבוצות השונות -



כשדיברנו על רדוקציות הראנו כי -

$$L_2 \in R \Leftarrow \begin{cases} L_1 \leq L_2 \\ L_2 \in R \end{cases}$$

נרצה לדעת האם אותו דבר קורה גם עבור P, כלומר, האם -

$$L_2 \in P \overset{?}{\Leftarrow} \begin{cases} L_1 \leq L_2 \\ L_2 \in P \end{cases}$$

כשהוכחנו את משפט הרדוקציה השתמשנו בהרכבה של מכונות M_2 ו־M_f כדי לבנות מ"ט M_1 כנדרש. אבל! אפילו אם המכונה M_2 יעילה מאד אנחנו לא יכולים להבטיח יעילות של M_f ולכן ההוכחה שבה השתמשנו לא עובדת (בפרט - המשפט שלמעלה גם לא נכון, אבל נראה את זה בשלב מאוחר יותר).

הגדרה 7.1 פונקציה $f : \Sigma^* \mapsto \Sigma^*$ היא רזוקציה פולינומית מ־L_1 ל־L_2 ומסמנים $L_1 \leq_P L_2$ אם -

1. $f \in POLY$
2. תקפות $x \in L_1 \Leftrightarrow f(x) \in L_2$

תכונות בסיסיות

- משפט הרדוקציה -

$$L_1 \in P \Leftarrow \begin{cases} L_1 \leq_P L_2 \\ L_2 \in P \end{cases}$$

הסבר - אותה בניה + הוכחה של משפט הרדוקציה ($\Leftarrow$ נכונות). הסיבוכיות פולינומית כי $M_f + M_2$ מ"ט פולינומיות.

- אותו משפט תקף גם עבור NP במקום P
- טרנזיטיביות -

$$L_1 \leq_P L_3 \Leftarrow \begin{cases} L_1 \leq_P L_2 \\ L_2 \leq_P L_3 \end{cases}$$

הסבר - הרכבה.

הגדרה 7.2 שפה נקראת NP-שלמה אם -

1. $L \in NP$
2. לכל $L' \in NP$ מתקיים - $L' \leq_P L$

אוסף השפות הנ"ל מסומן ב-NPC.

טענה 7.3 תהא L שפה NP-שלמה. אזי $P = NP$ אמ"מ $L \in P$

הוכחה:

- כיוון אחד - $L \in NPC$ אזי (מהגדרה) $L \in NP$ אזי (מהנחה) $L \in P$
- כיוון שני - מספיק להוכיח $NP \subseteq P$

ניקח $L' \in NP$ מההנחה - $L \in P$, ומהיות $L \in NPC$ נקבל $L' \leq_P L$ אזי (ממשפט הרדוקציה) -

$$L' \in P \Leftarrow \begin{cases} L' \leq_P L \\ L \in P \end{cases}$$

■

7.1 דרכי הוכחה ל-NP-שלמות

1. ישירות (מהגדרה)
2. דרך עקיפה -

טענה 7.4 אם מתקיים -

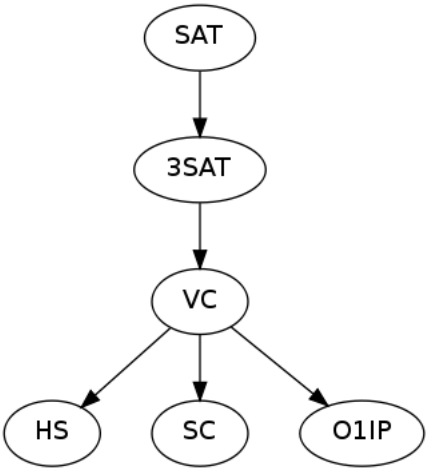
$$\begin{aligned} L_2 &\in NP \\ L_1 &\in NPC \\ L_1 &\leq_P L_2 \end{aligned}$$

אזי $L_2 \in NPC$

הוכחה: $L_2 \in NP$ (נתון)
לכל $L' \in NP$ מתקיים $L' \leq_P L_1 \leq_P L_2$ ולכן מטרנזיטיביות - $L' \leq_P L_2$. ■

7.2 לאן הולכים מכאן?

נראה דוגמאות לשפות NP שלמות -



7.2.1 כיסוי בצמתים Vertex Cover

הגדרה 7.5 נתון גרף $G = \{V, E\}$. קבוצה $B \subseteq V$ נקראת כיסוי בצמתים עבור G אם לכל $e = (a, b) \in E$ מתקיים - $a \in B$ או $b \in B$.

הערה 7.6 תמיד יש כיסוי בגודל $n = |V|$, וגם בגודל $n - 1$

הערה 7.7 הגרף המלא K_n דורש לפחות $n - 1$ צמתים

$$VC = \{(G, k) \mid exists\ B \text{ (vertex cover) such as } |B| = k\}$$

היחס המתאים - $S_{VC} = \{(G, k, B) \mid B \text{ is a vertex cover for } G \text{and } |B| = k\}$
היחס S_{VC} מקיים -

- חסום פולי'
- ניתן לזיהוי יעיל
- לבעיית החיפוש לא ידוע פתרון יעיל (יש פתרון אקספוננציאלי)

השפה המתאימה -

$$VC = \{(G, k) \mid \text{exists vertex cover } B \text{ for } G \text{such as } |B| = k\}$$

טענה 7.8 $VC \in NPC$ - הוכחה בהמשך.

בינתיים נציין כי $VC \in NP$ -

- ניתן להראות כי S_{VC} הוא היחס הנדרש לפי אחת ההגדרות של NP.
- דרך אחרת - "ננחש" B, נבדוק האם מתאימה, אם כן נקבל ואחרת נדחה.

7.2.2 בעיית הקבוצה המייצגת - Hitting Set (HS)

נתונים טבעיים $A_1, A_2, \ldots A_n \in [n]$ וקבוצות n, k
האם קיימת קבוצה $R \subseteq [n]$ בגודל k כך שלכל i מתקיים - $R \cap A_i \neq \emptyset$

$$HS \triangleq \{n, k, A_1, A_2, \ldots A_n \mid \text{exits } R \text{ such as...}\}$$

טענה 7.9 $HS \in NPC$

הוכחה: נחלק לשלושה חלקים -

1. $HS \in NP$ (מ"ט א"ד פולינומית תנחש את R ותבדוק)
2. $VC \in NPC$
3. צ"ל - $VC \leq_P HS$

נראה רדוקציה פולינומית כנדרש -

$$f(G, k) = n, k, A_1, A_2, \ldots A_m$$

כך ש-

- $n = |V|$
- $m = |E|$
- $e_i = (a, b) \Rightarrow A_i = \{a, b\}$

מתקיים - $f \in POLY$
תקפות -
$(G, k) \in VC$ (מהגדרה) $\Leftrightarrow$
קיימת $B = \{i_1, i_2, \ldots i_k\}$ המהווה כיסוי בצמתים עבור G (בנייה) $\Leftrightarrow$
קיימת $R = B = \{i_1, i_2, \ldots i_k\}$ המהווה קבוצה מייצגת עבור $A_1, A_2, \ldots A_m$ (הגדרת HS) $\Leftrightarrow$
$f(G, k) \in HS$ ■

7.2.3 בעיית הכיסוי בקבוצות Set Cover - SC[15]

נתונות קבוצות רבות, כל קבוצה "מכסה" חלק מסויים מהעולם. בהנתן מספר k רוצים לדעת האם קיימות k קבוצות כך שכל העולם מכוסה על ידי קבוצה אחת לפחות מתוך ה-k הנ"ל.
פורמלית -

- נתונים טבעיים n, k וקבוצות - $C_1, C_2, \ldots C_t \subseteq [n]$ (תתי קבוצות של הטבעיים $1, 2, \ldots n$)
- האם קיימות k קבוצות $C_{i_1}, C_{i_2}, \ldots C_{i_k}$ שמכסות את $[n]$?
- כשפה - $SC = \{n, k, C_1, C_2, \ldots C_t | \text{ exists k groups ... } \}$

[15] הרצאה עשירית 25/5/10

טענה 7.10 $SC \in NPC$

הוכחה: נשתמש ב"דרך העקיפה" -

- $SC \in NP$

כזכור - קיימות שתי הגדרות ל-NP -

1. מ"ט א"ד פולי', שבהנתן הקלט "תנחש" את האינדקסים המתאימים ותבדוק ש-$C_{i_1}, C_{i_2}, \ldots C_{i_k}$ מהווה כיסוי מתאים.
2. $R_{SC} = \{((n, k, C_1, C_2, \ldots C_t), (i_1, i_2, \ldots i_k)) \mid (C_{i_1}, C_{i_2}, \ldots C_{i_k}) \text{ is a set cover for } [n] \}$.

ובכל אחד מהמקרים ההוכחה הפורמלית פשוטה ולא נתעכב עליה...

- הוכחנו בעבר כי $VC \in NPC$
- נותר להראות רדוקציה פולינומית מ-VC אל SC -

נגדיר -

$$f(G, k) = (n, k, C_1, C_2, \ldots C_t)$$

כך ש-

- $n = |E|$
- $t = |V|$
- לכל צומת v_i נגדיר קבוצה C_i המכילה את כל המספרים j כך שקיימת קשת $e_j = (v_i, v_k)$ (כלומר - קשת e_j שנוגעת בצומת v_i)

מתקיים -

- $f \in POLY$
- תקפות -
  - $(G, k) \in VC \Leftrightarrow$ (מהגדרת VC)
  - קיימת $B = \{v_{i_1}, v_{i_2}, v_{i_k}\}$ המהווה כיסוי בצמתים ל-$G \Leftrightarrow$ (בנייה)
  - קיימות k קבוצות $C_{i_1}, C_{i_2}, \ldots C_{i_k}$ המהוות כיסוי ל-$[n] \Leftrightarrow$ (הגדרת f, SC)
  - $f(G, k) \in SC$

כלומר הראנו רדוקציה פולינומית כנדרש ו-$SC \in NPC$.

■

7.2.4 תכנות בשלמים $0, 1$ (**01 Integer Programing - $01IP$**)

נתונים -

- מטריצה $A \in \mathbb{Z}^{m \times n}$
- וקטור $b \in \mathbb{Z}^m$

האם קיים פתרון $x \in \{0, 1\}^n$ למערכת - $Ax \geq b$? (כלומר - כל כניסה בווקטור Ax גדולה או שווה מהכניסה המתאימה בווקטור b).

הערה 7.11 כרגיל, בהנתן פתרון קל לוודא שהוא אכן נכון, ובהנתן זמן אקפוננציאלי - קל לבדוק את כל האפשרויות.

טענה 7.12 $01IP = \{A, b \mid \text{exists solution } x...\} \in NPC$

הוכחה: כרגיל -

- $01IP \in NP$ (טרוויאלי)
- $VC \in NPC$
- נראה רדוקציה פולינומית - $VC \leq_p 01IP$ -

נגדיר פונקציה f -

$$f(G, k) = (A, b)$$

הרעיון - לכל צומת v_i בגרף נתאים משתנה x_i כך ש־$x_i = 1$ אומר ש־v_i בכיסוי ו־$v_i = 0$ אומר ש־v_i אינו בכיסוי. כלומר - $n = |V|$.

כך ש־

- $A^{(|E|+1)\times|V|}$ מוגדרת באופן הבא -
  - לכל קשת $e_j = (u, v)$ נגדיר אי שוויון $x_u + x_v \geq 1$, כלומר - $A_{ju} = A_{jv} = 1$ ו־$A_{jk} = 0$ לכל $k \neq u, v$.
  - את השורה האחרונה במטריצה נגדיר על ידי אי השוויון - $x_1 + x_2 + \ldots x_{|V|} \leq k$ כלומר - $A_{(|E|+1)i} = -1$ לכל i
- b (על פני הבניה לעיל - $b_i = 1$ לכל $1 \leq i \leq |E|$ ו־$b_{(|E|+1)} = -k$)

כעת קל להווכח כי בהנתן כיסוי מתאים B ההשמה המספקת היא - $i \in B \Leftrightarrow x_i = 1$ (קל לראות שכל האי שוויונים מתקיימים)

מצד שני - אם קיימת השמה x המקיימת את כל אי השוויונים נגדיר את B על ידי -

$$B = \{i \mid x_i = 1\}$$

וקל לראות ש־$|B| \leq k$, ואכן B כיסוי בצמתים (הקשת $e_j = (u, v)$ בהכרח מכוסה כי אי השוויון ה־j מתקיים אמ"מ $x_u + x_v \geq 1$, כלומר לפחות אחד מהצמתים u, v ב־B).

■

7.2.5 השפה 3SAT

נתון - פסוק φ בצורת $3CNF$

האם φ ספיק?

כלומר -

$$\varphi : C_1 \wedge C_2 \wedge \cdots C_m$$

כאשר כל C_i הוא פסוקית בת 3 ליטרלים -

$$C_i = l_{i_1} \vee l_{i_2} \vee l_{i_3}$$

כאשר ליטרל

$$l_{i_j} \in \{x_1, \neg x_1, x_2, \neg x_2, \ldots x_n, \neg x_n\}$$

טענה 7.13 $3SAT \leq_p VC$ (בהמשך נראה כי $3SAT \in NPC$ ונקבל את ההוכחה לכך שאכן $VC \in NPC$) **הוכחה:** נראה רדוקציה $f(\varphi) = (G, k)$

בניית הגרף -

- $2n$ צמתי ליטרלים, יסומנו $x_1, \overline{x_1}, x_2, \overline{x_2} \dots x_n, \overline{x_n}$, עם n קשתות, קשת בין כל x_i ו-$\overline{x_i}$.
- $3m$ צמתי פסוקיות, מסודרים כ-m משולשים ואם $C_i = l_{i_1} \wedge l_{i_2} \wedge l_{i_3}$ צמתי המשולש ה-i מסומנים $l_{i_1}, l_{i_2}, l_{i_3}$.
- לכל צומת מצמתי הפסוקיות - נחבר קשת לצומת הליטרל בעל אותו שם. למשל אם -

$$l_{i_j} = \overline{x_k}$$

נחבר קשת $e : \left(l_{i_j}, \overline{x_k}\right)$

בנוסף - $k = n + 2m$.

נכונות -

- $f \in POLY$
- תקפות -

– נניח כי $\varphi \in 3SAT$ ונראה כי $(G, k) \in VC$

אם $\varphi \in 3SAT$ אזי קיימת השמה מספקת α למשתנים, נשתמש ב-α על מנת לבנות כיסוי B לגרף -

* הוסף ל-B ל צומת ליטרל l כך ש-$\alpha(l) = TRUE$.
* השמה מספקת, לכן בכל פסוקית קיים לפחות ליטרל אחד l כך ש-$\alpha(l) = TRUE$, ניקח לכיסוי את שני צמתי הפסוקיות האחרים.

נראה ש-B כיסוי מתאים -

* $|B| = n + 2m = k$
* נוודא ש-B מכסה את הקשתות משלושת הסוגים -

1. לכל קשת מבין n קשתות הליטרלים, אחד מבין $x_i, \overline{x_i}$ מקבל $TRUE$, לכן בכיסוי, ולכן מכוסות.
2. קשתות ה"משולשים" - כל שני צמתים במשלוש הן כיסוי, ובחרנו שני צמתים מכל משולש.
3. קשתות מהסוג השלישי -

(א) אם צומת הפסוקית בכיסוי - אזי כמובן שהקשת מכוסה

(ב) אחרת - צומת הפסוקית אינה בכיסוי, כלומר הפסוקית מסופקת, אבל אז צומת הליטרל בכיסוי.

בכל מקרה - אחת מצמתי הקשת בכיסוי, ולכן הקשת מכוסה.

כלומר - B מהווה כיסוי, כלומר $(G, k) \in VC$

– כיוון שני - נניח כי $\varphi \in 3SAT$ ונוכיח כי $f(\varphi) = (G, k) \in VC$

$(G, k) \in VC$, כלומר קיימת קבוצה B בגודל לכל היותר k המהווה כיסוי.

נשים לב שאפילו ללא הקשתות מהסוג השלישי יש צורך בלפחות n צמתי ליטרלים ו-$2m$ צמתי פסוקיות על מנת לכסות את קשתות הליטרלים והמשולשים.

כיוון ש-$k = n + 2m$ נקבל כי זה בדיוק המצב - B מכיל בדיוק n צמתי ליטרלים ו-$2m$ צמתי פסוקיות (אחד מכל זוג - פסוק אטומי ושלילתו, ושניים מכל משולש).

נבנה השמה מספקת α -

$$\alpha(x_i) = TRUE \;\Leftrightarrow\; (v_{x_i} \in B) \wedge (v_{\overline{x_i}} \notin B)$$

מהטענות הנ"ל α מוגדרת היטב.

נראה כי כל פסוקית C_j מסופקת, כלומר שקיים ליטרל בה שמקבל $TRUE$ -

נתבונן על המשולש של הפסוקית C_j, שניים מהצמתים שלו הם בכיסוי, נתבונן על השלישי - יש קשת ממנו אל צומת הליטרל המתאים, הדרך היחידה של הכיסוי לכסות את הקשת הזו היא לבחור את צומת הליטרל הזה.

כלומר (על פי בניית α) ההשמה נותנת $TRUE$ לליטרל המתאים - ולכן הפסוקית C_j מסופקת ע"י α.

לסיכום - הראנו כי קיימת α שמספקת את φ, כלומר $\varphi \in 3SAT$.

ולכן - $f(\varphi) = (G,k) \in VC \Leftrightarrow \varphi \in 3SAT$ וההוכחה הושלמה.

■

7.2.6 דוגמא - Bounded Halting

שפת מ"ט האי דטרמיניסיות $\langle M \rangle$, הקלטים $\langle x \rangle$ והמחרוזות 1^t כך ש-למ"ט M יש מסלול באורך קטן או שווה ל-t המקבל את x.

טענה 7.14 $BH \in NPC$

הוכחה: כרגיל, שלושה חלקים -

- $BH \in NP$ - נתאר מ"ט א"ד פולי M_{BH} ש"מנחשת" מסלול w באורך t ומקבלת אמ"מ M מקבלת את x במסלול w. M_{BH} פולינומית כי אורך הקלט גדול או שווה ל-t (לכן t רשום בייצוג אונרי, ולא בינארי למשל).
- תהא $L \in NP$ כלשהי ונראה $L \leq_p BH$ -
  $L \in NP$ לכן (מהגדרה) קיימת מ"ט א"ד M_L ופולינום P_L עבורה.
  נתאר רדוקציה -

$$f_L(x) = \left(\langle M_L \rangle, \langle x \rangle, 1^{P_L(|\langle x \rangle|)} \right)$$

קל לראות שאכן $f \in POLY$ והתקפות מתקיימת באופן טרוויאלי.

■

7.2.7 השפה SAT[16]

SAT היא שפת הפסוקים φ הספיקים מסוג CNF.
כזכור -

$$\begin{aligned} \varphi &= C_1 \wedge C_2 \wedge \ldots C_N \\ C_i &= x_{i_1} \vee x_{i_2} \cdots \vee \overline{x_{i_k}} \vee \ldots x_{i_n} \end{aligned}$$

(פסוק CNF הוא "וגם" בין פסוקיות שמורכבות מ"או" של ליטרלים - שהם משתנה או שלילתו)

משפט 7.15 <u>משפט *Cook*</u> - $SAT \in NPC$

דוגמא - $VC \leq_P SAT$ נראה רדוקציה $f(G,k) = \varphi$

- משתנים - $|V| = n, \quad \{X_{i,r}\}_{i=1..n}^{r=1..k}$
  - המשמעות - $X_{i,r} = 1$ אמ"מ V_i הוא הצומת ה-r בכיסוי
- פסוקיות -

[16] הרצאה אחת-עשרה - 1/6/2010

- לכל קשת $e=(a,b)\in E$ פסוקית -

$$X_{a,1}\vee X_{a,2}\vee\ldots X_{a,k}\vee X_{b,1}\vee X_{b,2}\cdots\vee X_{b,k}$$

פסוקיות כאלו מבטיחות שפסוק ספיק מכסה את הגרף.
- לכל r ולכל $i<j$ -

$$\overline{X_{i,r}}\vee\overline{X_{j,r}}$$

פסוקיות כאלו מבטיחות שלא נבחר לתת "אמת" לשני "צמתים" עם אותו אינדקס r.

מתקיים -

- הבנייה פולינומית.
- תקפות -
  - $\varphi\in SAT\overset{?}{\Leftarrow}(G,k)\in VC$
    נתאר α -השמה מספקת עבור φ.
  - $(G,k)\in VC\Leftarrow$ קיימת קבוצה $B=\{i_1,i_2,\ldots i_k\}$ המהווה כיסוי, לכל i_j כזה נגדיר -

$$\alpha\left(X_{i_j,j}\right)=T$$

    ו-$\alpha\left(X_{i,r}\right)=F$ לכל משתנה אחר.
  - מצורת הבניה (וכיוון ש-B כיסוי), קל לוודא ש-α אכן מספקת את כל הפסוקיות ואכן $\alpha\left(\varphi\right)=T$ כלומר $\varphi\in SAT$.
  - כיוון שני, נניח כי - $\varphi=f\left(G,k\right)\in SAT$, כלומר קיימת השמה α המספקת אותה. בפרט, α מספקת את הפסוקיות מהסוג השני - כלומר לכל r קיים $X_{i_r,r}$ **אחד** לכל היותר כך ש-$\alpha\left(X_{i,r}\right)=T$. נגדיר -

$$B=\{i_1,i_2,\ldots i_k\}$$

    כלומר - $|B|\leq k$.
    ההשמה α מספקת גם את הפסוקיות מהסוג הראשון, לכן לכל קשת $e=(a,b)\in E$ נובע כי $a\in B$ או $b\in B$ (פשוט מהבנייה) ולכן B כיסוי כנדרש.

הוכחה: (למשפט Cook - משפט 7.15)

- $SAT\in NP$
- עובדה - לכל $f:\{0,1\}^n\mapsto\{0,1\}$ (פסוק בן n משתנים) ניתן לכתוב את f כ-CNF בן 2^n פסוקיות (או פחות). בפרט אם $n=O\left(1\right)$ אז גם גודל ה-CNF הוא $O\left(1\right)$.

צ"ל - לכל $L\in NP$ - $L\leq_p SAT$
$L\in NP$, לכן קיימת מ"ט א"ד פולי' מתאימה M והפולינום P המתאים.
נתאר רדוקציה h שתקיים תקפות -

$$w\in L\Leftrightarrow\overbrace{h\left(w\right)}^{\varphi}\in SAT$$

- $\varphi\in SAT\Leftrightarrow$ <u>קיימת</u> השמה α מספקת עבור φ
- מצד שני - $w\in L\Leftrightarrow$ <u>קיים</u> מסלול מקבל של M על $w\Leftrightarrow$ קיימת סדרת קונפיגורציות -

$$c_0,c_1\ldots c_t$$

שמהווה חישוב מקבל (כלומר c_0 קונפיגורציה התחלתית של M, c_t מקבלת, ו-c_i,c_{i+1} קונפיגורציות עוקבות)

נראה אנלוגיה בין ההשמות למסלולים. נשים לב שלפסוקים יש מבנה קשיח, אבל חישוב הוא גמיש למדי, נתבונן על טבלת חישוב -

	$0, 1, 2, \ldots t \triangleq P(\lvert w \rvert)$
c_1	$q_0 w_1 w_2 \ldots w_n$
c_2	$\vdots$
$c_{t \triangleq P(\lvert w \rvert)}$	

כדי לקבע את מבנה טבלת החישוב, נשתמש בקונבנציות הבאות -

- אם הקונפיגורציה קצרה מ־t נוסיף לה $\flat$ עד אורך t
- אם החישוב הסתיים לפני "זמן" t (מצב c_t) נשכפל את הקונפיגורציה הסופית עד ל־c_t

כעת יש לנו מבנה קבוע, מטריצה בגודל $(p(|w|)+1)^2$ כאשר כל אחד מהתאים שלה הוא $a \in \Gamma \cup Q$

משתני φ לכל $0 \leq i, j \leq t$ ולכל $a \in \Gamma \cup Q$ נגדיר משתנה $X_{i,j,a}$ כך ש־$X_{i,j,a} = T$ אמ"מ במקום ה־i, j בטבלה מופיע a.

בניית φ

$$\varphi = \varphi_0 \wedge \varphi_A \wedge \varphi_{comp} \wedge \varphi_{leagal}$$

נתאר כל אחד ממרכיבי φ (כל אחד מהם הוא CNF, ניתן לבניה בזמן פולינומי ב־$|w|$)

- φ_A [משמעות - נוסחה המקבלת את הערך T אמ"א בשורה האחרונה בטבלה מופיע המצב q_A - כלומר, חישוב מקבל]

$$\varphi_A \bigvee_{0 \leq j \leq t} X_{t,j,q_A}$$

- φ_0 [משמעות - נוסחה המקבלת את הערך T אמ"מ השורה הראשונה היא הקונפיגורציה ההתחלתית של M על w]

$$\varphi_0 = X_{0,0,q_0} \wedge X_{0,1,w_1} \wedge X_{0,2,w_2} \wedge \ldots X_{0,n,w_n} \wedge X_{0,n+1,\flat} \wedge \ldots X_{0,t,\flat}$$

- φ_{leagal} [משמעות - בכל כניסה i, j בטבלה יש a יחיד כך ש־$X_{i,j,a} = T$]

$$\varphi_{leagal} = \bigwedge_{0 \leq i,j \leq t} \left(\bigvee_{a \in \Gamma \cup Q} X_{i,j,a} \right) \wedge \bigwedge_{0 \leq i,j \leq t} \left(\bigwedge_{\substack{a \neq b \\ a,b \in \Gamma \cup Q}} \left(\overline{X_{i,j,a}} \vee \overline{X_{i,j,b}} \right) \right)$$

- φ_{comp} [משמעות - כל 2 שורות עוקבות מתארות צעד אפשרי של M]

  - אבחנות + מינוח -

  * כל שתי קונפיגורציות עוקבות שונות בלכל היותר 3 מקומות (המקום בו היה הראש, ואחד מימינו ומשמאלו)
  * לכן, נתבונן על תתי טבלאות בגודל 2×3 (2 שורות עוקבות, 3 עמודות עוקסות) ונאמר כי מלבן כזה הוא חוקי אם קיימות 2 קונפיגורציות עוקבות C, C' בהן הוא מופיע.
  * אוסף המלבנים החוקיים תלוי רק ב־M והוא קבוע! (לא תלוי בקלט)
  * טבלת חישוב היא חוקית אמ"מ כל המלבנים 2×3 שלה הם חוקיים (בהנחה ו־c_0 חוקית), הוכחה ניתן לעשות באינדוקציה.

– נכתוב

$$\varphi_{comp} = \bigwedge_{\substack{0\leq i\leq t-1\\0\leq j\leq t-2}} U_{comp}^{i,j}$$

כאשר המשמעות של $U_{comp}^{i,j}$ היא שהמלבן שפינתו השמאלית העליונה היא ב־i,j הוא חוקי. נותר להביע את $U_{comp}^{i,j}$ -

נשים לב כי הפסוקית תלויה ב־$|\Gamma \cup Q|^6 = O(1)$ משתנים, לכן ניתן להפוך אותה ל־CNF שגם הוא בגודל $O(1)$ ומתאר את $U_{comp}^{i,j}$. כיוון שאוסף המלבנים החוקיים לא תלוי ב־w אז יצור הנוסחה הנ"ל מתבצע בזמן פולינומי ($O(1)$)

מתקיים

- כל אחד ממרכיבי φ הוא CNF שניתן לבנות בזמן פולינומי לכן φ עצמה ובנייתה פולי'. כלומר - הרדוקציה פולי'
- תקפות - נניח $w \in L$ ונראה ש־$\varphi = h(w) \in SAT$
  - $w \in L \Rightarrow$ קיים חישוב מקבל של M על $w \Rightarrow$
  - קיימת <u>טבלת חישוב מקבלת</u>

נגדיר השמה α על פי הטבלה הנ"ל - $\alpha(X_{i,j,a}) = T$ אמ"מ במקום ה־i,j בטבלה מופיע a. מהבנייה α אכן מספקת את כל חלקי φ ולכן $\alpha(\varphi) = T$ כלומר $\varphi \in SAT$.

כיוון שני - נניח כי $h(w) = \varphi \in SAT$, כלומר קיימת α המספקת אותה. בפרט - היא מספקת את כל הפסוקיות של φ_{leagal} ולכן ניתן לבנות טבלת חישוב מתאימה. בנוסף - α מספקת את φ_0 - כלומר טבלת החישוב מתחילה בקונפיגורציה ההתחלתית של M על w, וכן את φ_A כלומר החישוב מסתיים במצב מקבל, ולבסוף גם את φ_{comp} - כלומר הטבלה מהווה חישוב חוקי.

ולכן M מקבלת את w בפרט - $w \in L$.

■

7.2.8 סכום תת הקבוצה[17] (Subset Sum)

נתונים - $x_1, x_2, \ldots x_s, k'$ טבעיים

האם קיימת $I \subseteq [s]$ כך ש־$\sum_{i\in I} x_i = k'$.

טענה 7.16 $SS \in NPC$

הוכחה: מספיק להראות $VC \leq_p SS$

בהנתן קלט (G,k) נבנה קלט ל־SS:

$$b_1 \ldots b_m, a_1, \ldots a_n, k'$$

כאשר $m = |E|$, $n = |V|$ ונגדיר -

- לכל $j \in [m]$ - $b_j = 10^{j-1}$
- לכל $i \in [n]$ - $a_i = 10^m + \sum_{j:v_i \in e_j} 10^{j-1}$
- $k' = k \cdot 10^m + \sum_{j=1}^{m} 2 \cdot 10^{j-1}$

[17] הרצאה שתיים־עשרה (8/6/10)

למה זה עובד? נתבונן על המספרים -

$$\begin{aligned} b_1 &= 0.....1 \\ b_2 &= 0...10 \\ b_3 &= 0...100 \\ &\vdots \\ b_m &= 1\overbrace{0...0}^{m-1} \end{aligned}$$

לעומתם ה־a־ים -

$$a_i = 1\overbrace{0...0110...010...0}^{m}$$

נשים לב כי לכל a_i יש 1 בספרה השמאלית ביותר (ספרות מימין לו) ו־1 בכל אחד מהמקומות j שבהם הצומת ה־i מופיע בקשת ה־j . ולסיכום -

$$k' = k\overbrace{22...2}^{m}$$

פולינומיות - ברור שמתקיימת.

תקפות - נשים לב שבכל אחת מ־m העמודות הימניות יש בדיוק 3 "אחדים" ולכן אין "נשא" (carry) בחיבור.
בהנתן פתרון לבעיית ה־SS שיצרנו נגדיר -

$$B = \{V_i \mid a_i \in I\}$$

ונראה כי B היא פתרון לבעיה VC -

- $|B| = k$, כיוון שה"ספרה" המובילה ב־k' היא k. אנחנו יודעים שאין נשא בחיבור, רק a_i־ים מכילים 1 בעמודה השמאלית בהכרח יש בדיוק k צמתים בכיסוי.
- B הוא כיסוי. לכל e_j ה־b_j המתאים תורם לכל היותר 1 לספרה ה־j של k' ולכן לפחות אחד מה־a_i־ים בכיסוי תורם 1 לעמודה זו - וזה קורה אמ"מ v_i הוא חלק מהקשת ה־j.

בכיוון השני, בהנתן כיסוי B עבור (G,k) נבנה I מתאים -

- ראשית, נחבר את כל ה־a_i־ים שמתאימים ל־$v_i \in B$, נקבל מספר מהצורה -

$$k\overbrace{2212112...12}^{m}$$

- לכל מקום j שעבורו קיבלנו 1 נוסיף את b_j המתאים, ואז הסכום שיתקבל יהיה -

$$k' = k\overbrace{22...2}^{m}$$

כנדרש.

■

7.2.9 בעיית החלוקה Partition

קלט - מספרים טבעיים $x_1, x_2 \dots x_s$
האם קיימת $I \subseteq [s]$ כך ש־$\sum_{i \in I} x_i = \sum_{j \notin I} x_j$ (כלומר - חלוקה לשתי קבוצות מספרים שסכומן זהה)?

טענה 7.17 $PART \in NPC$

הוכחה: נראה רדוקציה $SS \leq_p PART$
בהנתן $x_1, x_2, \dots x_s, k$ נבנה -

- $x_1, x_2, \dots x_s, B = (2A - k), C = A + k$ כאשר $A \triangleq \sum_i x_i$.

תקפות - נשים לב כי סכום המספרים הוא $4A$, וכן כי $B + C = 3A$, לכן בהכרח B, C בשני "צידי" החלוקה.
בהנתן פתרון I לבעיית SS -

- נבנה פתרון ל־SS -

$$\{x_i\}_{i \in I}, B$$

וברור כי הסכום הוא $2A$ וכך גם הסכום של המספרים שנותרו.

בנהתן 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 זה הוא פתרון מתאים.

■

7.2.10 אכסון בתאים Bin Partition

קלט - $x_1, x_2, \dots x_s$ טבעיים. מספר תאים k וגודל של תא B
האם ניתן לאכסן את כל האיברים בתאים הנתונים?

טענה 7.18 $BP \in NPC$ **הוכחה:** נראה רדוקציה $PART \leq_p BP$
בהנתן $x_1, \dots x_s$ נבנה קלט לבעיית החלוקה -

$$x_1, x_2 \dots x_s$$

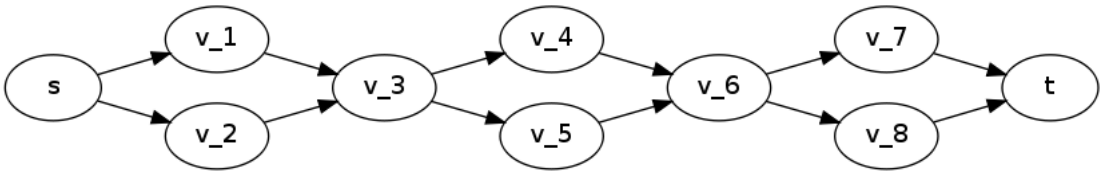
ונגדיר $k = 2$ ו־$B = \frac{1}{2} \sum x_i$.
ברור כי התקפות מתקיימת (לא נוכיח באופן מלא) ■

7.2.11 בעיית המסלול עם אורך ומחיר חסומים

נתון - גרף $G = (V, E)$ לא מכוון ופשוט, כך שלכל קשת נתון אורך $l(e)$ ומחיר $w(e)$, שני צמתים בגרף s, t, ושני מספרים W, L.
האם קיים מסלול מ־s ל־t שאורכו קטן או שווה ל־L ומחירו קטן או שווה ל־W?

טענה 7.19 $L \in NPC$

הוכחה: נראה רדוקציה $PART \leq_p L$
בהנתן $x_1, x_2 \dots x_n$ קלט ל־$Partition$, נבנה גרף מתאים מהצורה -



הגרף מורכב מסדרה של מעויינים, כאשר במעויין ה־i המסלול העליון מתאים לבחירת x_i, והמסלול התחתון לכך שלא נבחר את x_i. הדבר יתבצע על ידי הגדרת המשקלים על הקשתות העליונות כ־x_i והאורכים של הקשתות התחתונות כ־x_i. שוב לא נוכיח באופן מלא, אבל קל להשתכנע שמציאת מסלול באורך ומשקל המוגבלים ע"י $\frac{1}{2}\sum_i x_i$ מתאים לפתרון של בעיית החלוקה. ■

8 התמודדות עם בעיות NP־שלמות (ובעיות קשות אחרות)

דוגמא $VC \in NPC$ נגדיר - $f_{VC}(G) = \min_k \{\text{exists vertex cover of size k for } G\}$

טענה 8.1 $VC \in P \Leftrightarrow f_{VC} \in POLY$

הוכחה: כיוון אחד - בהנתן (G,k), נחשב $f_{VC}(G)$ ונקבל אמ"מ $k \geq f_{VC}(G)$
כיוון שני - בהנתן G, נבדוק עבור $k = 0, 1, 2 \dots$ האם $(G,k) \in VC$? אם כן - נעצור ונפלוט את k (הראשון שמקיים). ■

למעשה - אנחנו מתעניינים בבעיית החיפוש -

$$R_{VC} = \{((G,k), B) \mid B \text{ is a VC of size } k \text{ for } G\}$$

טענה 8.2 בעיית החיפוש של R_{VC} ניתנת לפתרון יעיל אמ"מ $VC \in P$

הוכחה: כיוון אחד - אם בעיית החיפוש של R_{VC} ניתנת לפתרון יעיל אותה מכונה מראה כי $VC \in P$
כיוון שני - $VC \in P$ אזי $P = NP$ לכן (ממשפט 6.19) לכל R חסום פולינומי וניתן לזיהוי יעיל הוא גם ניתן לחיפוש יעיל. בפרט ל־R_{VC}. ■

8.1 אלגוריתמי קירוב

נראה אלגוריתם 2־קירוב, כלומר A יעיל ולכל גרף G -

$$f_{VC} \leq A(G) \leq 2 f_{VC}(G)$$

יתר על כן - A מוצא כיסוי כזה.

תזכורת - אם $G = (V, E)$ גרף -

- שידוך היא קבוצה $M \subseteq E$ של קשתות זרות בצמתים.
- שידוך מקסימלי - הוא שידוך שאיננו ניתן להרחבה.
- שידוך מקסימום - הוא השידוך הגדול ביותר (לא כל שידוך מקסימלי הוא שידוך מקסימום!).

8.1.1 אלגוריתם יעיל למציאת שידוך מקסימלי

- $M = \emptyset$
- עבור על כל קשתות $e \in E$ בסדר כלשהו -
  - אם e זרה לכל קשתות M אז $M \leftarrow M \cup \{e\}$
- פלוט את M

מתקיים -

- האלגוריתם פולינומיאלי
- אינווריאנטה - לכל אורך הריצה M הוא שידוך
- לכל $e \notin M$ לא ניתן להוסיף את e ל-M (אחרת היינו מוסיפים אותה במקור). לכן M שידוך מקסימלי.

8.1.2 אלגוריתם A על קלט G

- מצא שידוך מקסימלי M על G
- פלוט B - קבוצת כל הצמתים שהוא אוסף הצמתים של קשתות אלו.

מתקיים

- האלגוריתם פולינומי
- B הוא כיסוי - ניקח קשת $e \in E$-
  - אם $e \in M$ ברור כי שני הצמתים של e ב-B
  - אם $e \notin M$ מקסימלי ולכן לא ניתן להוסיף אותה לשידוך, כלומר אחד הצמתים שלה הוא צומת של $e' \in M$ וצומת זה בכיסוי.
- נראה כי B לא "גדול מדי" -
  - נסמן ב-B^* כיסוי קטן ביותר עבור G. ברור כי $|B| \leq |B^*|$.
  - B^* מכסה את כל קשתות הגרף, בפרט את קשתות M. כיוון שקשתות M זרות בצמתים נקבל כי $|M| \geq |B^*|$ ומכן מתקיים -

$$|B| = 2\,|M| \leq 2\,|B^*|$$

כלומר - A הוא אכן אלגוריתם כנדרש. אלגוריתם פולינומי (יעיל) שמוצא כיסוי שהוא לכל היותר כפול בגודלו מהכיסוי האופטימלי.

8.1.3 דוגמא נוספת - בעיית האכסון BP

בעיית האופטימיזציה -

הקלט - $x_1, x_2, \ldots x_n$, וגודל תא B.

המטרה - למצא מספר מינימלי של תאים המאפשר אכסון.

נראה אלגוריתם 2-קירוב[18] לבעיה.

[18] העובדה שגם כאן וגם בבעיה הקודמת קיבלנו קירוב מסדר שתיים היא מקרית. לבעיות שונות קירובים מסדרים שונים, לעיתים גדולים ולעיתים קטנים יותר.

האלגוריתם -

- עבור $x_1, x_2, \ldots x_n$ -
  - אם יש מקום ל-x_i בתא קיים הכנס אותו לשם. אחרת - פתח תא חדש עבורו.

מתקיים -

- האלגוריתם פולינומי

הערה 8.3 אבחנה - בכל זמן במהלך הריצה קיים לכל היותר תא אחד שהוא פחות מחצי מלא. **הוכחה:** באינדוקציה -

- בהתחלה - ברור שנכון (אין תאים)
- צעד האינדוקציה -
  - אם $x_i > \frac{B}{2}$ - הוא לא יכול לקלקל את התכונה, כי גם אם נכניס אותו לתא חדש - התא החדש יהיה עכשיו יותר מחצי מלא.
  - אם $x_i \leq \frac{B}{2}$ - אם יש תא כנ"ל - הוא יוכנס אליו. אחרת - אין תא כזה, נפתח חדש ויהיה בדיוק תא אחד.

■

כעת נסמן ב-k את מספר התאים שהאלגוריתם מצא. וב-k^* את המספר האופטימלי.

- ברור כי $k^* < k$ (אופטימליות k^*)
- מצד שני, מתקיים -

$$\sum x_i \leq k^* B$$

בכל שלב יש לכל היותר תא שמלא עד כדי $\frac{B}{2}$, ולכן -

$$\sum x_i > (k-1)\frac{B}{2}$$

ולכן -

$$\begin{aligned} k-1 &\leq 2k^* \\ k &\leq 2k^* \end{aligned}$$

8.2 צמצום מרחב הקלט[19]

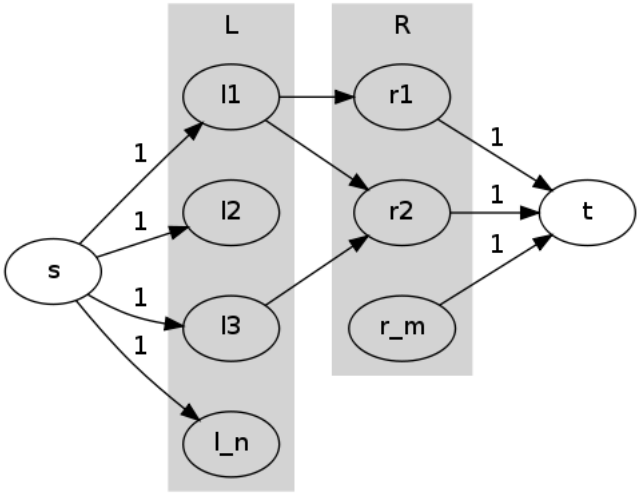
אפשר לחשוב על מרחבי קירוב בתור "הרחבת מרחב הפלט" - במקום לאפשר לכל קלט רק פלט מסויים, נרחיב את מרחב הפלטים שאנחנו מוכנים לקבל (לפלטים "קרובים") ובכך נהפוך את הבעיה לקלה יותר. באופן דומה - נוכל לצמצם את מספר הקלטים שאנחנו מוכנים לקבל, בשאיפה - נוציא מהבעיה את החלק ה"קשה" ונוכל לפתור את הבעיה המצומצמת בקלות.
לדוגמא - ראינו את $3SAT$, שהיא צמצום של "מרחב הקלט" של SAT. אבל במקרה הזה גם $3SAT \in NPC$, ולכן לא הפכנו את הבעיה לקלה יותר. אבל (לא הראנו בקורס) $2SAT \in P$, כלומר צמצום לשתי פסוקיות הופך את הבעיה ל"קלה".

8.2.1 דוגמא - כיסוי בצמתים (VC) בגרף דו צדדי

תזכורת - בגרף דו צדדי $G = (L, R, E)$. L, R שתי קבוצות צמתים כך שכל קשת $e \in E$ היא מצומת $r \in R$ לצומת $l \in L$.

[19] הרצאה שלוש-עשרה - 15/6/10

אלגוריתם פולינומי לבעיה (רדוקציה לזרימה) - בהנתן גרף דו צדדי כנ"ל G נבנה רשת זרימה N -



(נוסיף לגרף מקור s שיחובר לכל צמתי L בקשתות שקיבולן 1, בור t שכל צמתי R יחוברו אליו עם קיבול 1. את כל קשתות הגרף המקורי נגדיר לקיבול ∞)

תזכורת - חתך ברשת זרימה $\left(T,\overline{T}\right)$ הוא חלוקה זרה וממצה של V לשתי קבוצות כך ש-$s \in T$ ו-$t \in \overline{T}$.
קיבול של חתך - סכום קיבולי הקשתות מ-T ל-$\overline{T}$.
זרימת מקסימום בגרף = חתך מינימום.

- האלגוריתם פולינומי.

טענה 8.4

1. אם קיים כיסוי $B = (B_L, B_R)$ בגרף G בגודל k אזי קיים חתך $\left(T,\overline{T}\right)$ ב-N בקיבול k.
2. אם קיים חתך $\left(T,\overline{T}\right)$ ב-N שקיבולו **סופי** k אז קיים כיסוי $B = (B_R, B_L)$ שגודלו k .

מסקנה 8.5 גודל כיסוי מינימום ב-G שווה לגודל חתך מינימום ב-N (שווה לגודל זרימת מקסימום ב-N)

הוכחה: ■

1. בהנתן כיסוי $B = (B_L, B_R)$ נבנה חתך -

$$T = \{S\} \cup \overline{B_L} \cup B_R$$

ונחשב את קיבול החתך -[20]

- קיבול הקשתות מ-s ל-B_L - $|B_L|$
- קיבול הקשתות מ-$\overline{B_L}$ ל-$\overline{B_R}$ - 0 (אין קשתות כאלו, כי קשת כזו לא מכוסה על ידי הכיסוי)
- קיבול הקשתות מ-B_R ל-t - $|B_R|$

ולכן קיבול החתך הוא $|B|$.

[20] כאן $\overline{B_L}$ היק קבוצת כל הצמתים ב-L שאינם ב-B_L.

2. בהנתן חתך $\left(T,\overline{T}\right)$ שקיבולו k (סופי). נגדיר כיסוי -

$$\begin{aligned} B_R &= T \cap R \\ B_L &= \overline{T} \cap L \end{aligned}$$

כיוון שקיבול החתך הוא סופי - אין קשת מ-$\overline{B_L}$ ל-$\overline{B_R}$, ולכן כל הקשתות מכוסות בכיסוי. כיוון שקיבול החתך הוא k, ניתן לחזור על החשבון מהסעיף הקודם ונקבל כי $|B| = k$.

אנחנו מכירים אלגוריתמים פולינומיאליים למציאת זרימת מקסימום, ולכן יכולים לפתור את הבעיה בזמן פולינומיאלי.

8.3 גישות הסתברותיות

תחת השם של "גישות הסתברותיות" ניתן להכניס שני סוגי פתרונות -

- <u>סיבוכיות ממוצעת</u> - מניחים שקיים פילוג הסתברות D על הקלטים לבעיה, ואז מראים אלגוריתם ש**בממוצע** עובד בזמן פולינומיאלי (כיחס לפילוג הזה).
- <u>אלגוריתמים הסתברותיים</u> - אין פילוג על הקלטים, אנחנו מנתחים את המקרה הרע ביותר (מבחינת הקלט - הקלט הקשה ביותר). אך במקרה הזה האלגוריתם עצמו הוא הסתברותי.

8.3.1 סיבוכיות ממוצעת

הגישה הזו בעייתית, כיוון שאנחנו חייבים לדעת באופן מדוייק מה ההסתברות לקבל כל אחד מהקלטים האפשריים לבעיה. למעשה - לקבל את D בעצמו זו שאיפה כמעט לא פרקטית. שימוש בפילוג לא נכון, גם אם קרוב לפילוג האמיתי, יכול להשפיע בצורה קיצונית על האנליזה. בפרט - הנחה על פילוג אחיד היא בדרך כלל רחוקה מאד (מאד) מהמציאות.
גם אם ידוע לנו מה הוא D - החישוב המדוייק של התוחלת הוא (ברוב המוחלט של המקרים) קשה מאד.

8.3.2 אלגוריתמים הסתברותיים

הרעיון המרכזי הוא שעל אותו קלט, ריצות שונות יעבדו באופן שונה, כאשר צעדים מסויימים באלגוריתם יבוססו על החלטות רנדומליות. ונשאל - מה ההסתברות שהאלגוריתם יגיע לתשובה הנכונה? בפרט דורשים -

- לכל קלט x האלגוריתם מחשב נכון בהסתברות $0.5 < P \leq p\left(x\right)$.

הסיבה ש-P עצמו לא קריטי, היא שניתן להפעיל את האלגוריתם מספר גדול של פעמים ומחליטים על התוצאה הנכונה על פי כלל התוצאות, ההסתברות לשגיאה קטנה אקספוננציאלית במספר ההפעלות.

הגברה - עבור בעיות הכרעה, אם חוזרים על האלגוריתם $O\left(k\right)$ פעמים, ומחליטים ע"פ הרוב, השגיאה קטנה ל-$\frac{1}{2^k}$.
הערות

- שאלה פתוחה - האם קיים אלגוריתם הסתברותי יעיל לבעיה NPC כלשהי?
- במשך שנים הדוגמא לאלגוריתם הסתברותית "טוב יותר" מכל אלגוריתם לא הסתברותי אחר היא אלגוריתם מילר רבין לבדיקת ראשוניות. אבל, מרגע שהוכח אלגוריתם AKS שגם הוא פולינומי לבדיקת ראשוניות - אנו יודעים ש-$PRIMES \in P$ ולכן מילר-רבין הפך לדוגמא פחות מעניינית.

8.3.3 דוגמא - MAX-CUT

נתון גרף $G = (V, E)$ <u>לא מכוון</u> עם קיבולים $c\left(e\right) \geq 0$ לכל $e \in C$.
המטרה - למצא חתך גדול ביותר.

- עובדה - בעיית ההכרעה היא NPC

נראה אלגוריתם קירוב הסתברותי - שיוציא בכל פעם חתך אחר, עם קיבול שונה. זה יהיה אלגוריתם יעיל, שבכל גרף G פולט בכל ריצה חתך $\left(T,\overline{T}\right)$ בקיבול x אז מתקיים (לכל G) -

$$E\left(x\right)\geq\frac{OPT}{2}$$

כאשר OPT הוא גודל החתך המקסימלי (אופטימלי).

האלגוריתם בהנתן G נבנה חתך $\left(S,\overline{S}\right)$ -
לכל $a\in V$ בהסתברות $\frac{1}{2}$ הוסף את a ל-S, אחרת הוסף את a ל-$\overline{S}$.

מתקיים האלגוריתם מוצא חתך, ובזמן פולינומי.

- לכל קשת $e=(a,b)$, הסיכוי ש-e בחתך הוא $\frac{1}{2}$.
- נגדיר משתנה מקרי $\{0,1\}$ לכל e - $x_e=1$ אמ"מ e בחתך.
- נקבל כי $P\left(x_e=1\right)=\frac{1}{2}$ וכן $\mathbb{E}\left(x_e\right)=\frac{1}{2}$.
- נגדיר $X=\sum_e c\left(e\right)\cdot x_e$ (זה הוא קיבול החתך).
- ולכן $\mathbb{E}\left(X\right)=\frac{1}{2}\sum_e c\left(e\right)$ (כי תוחלת היא לינארית...)
- בפרט מתקיים כי עבור חתך המקסימום, גודל החתך הוא לכל היותר $\sum_e c\left(e\right)$ (אמ"מ כל הקשתות בחתך). לכן - תוחלת קיבול החתך שנותן האלגוריתם היא לכל הפחות חצי מקיבול החתך האופטימלי.

אי שוויון מרקוב -

$$P\left(x\geq\frac{1}{4}\sum_e c\left(e\right)\right)\geq\frac{1}{3}$$

הוכחה: נניח שלא. אזי -

$$\mathbb{E}\left(x\right)<\frac{1}{3}\sum_e c\left(e\right)+\frac{2}{3}\cdot\frac{1}{4}\sum_e c\left(e\right)=\frac{1}{2}\sum_e c\left(e\right)$$

וזו כמובן סתירה. ■

כעת ניתן להפעיל **הגברה** - נריץ את האלגוריתם k פעמים ונפלוט את החתך הגדול ביותר שפגשנו. יתקיים -

$$P\left(X\geq\frac{1}{4}\sum_e c\left(e\right)\right)\geq1-\left(\frac{2}{3}\right)^k$$

באותו האופן ניתן לבחור קבוע שונה מ-$\frac{1}{4}$ ולקבל קבוע שונה בצד ימין.

8.4 פונקציות קשות[21] לקירוב

8.4.1 דוגמא - #SAT

כאשר $\#SAT\left(\varphi\right)$ מוגדר להיות מספר ההשמות המספקות את φ.

טענה 8.6 אם $\#SAT\in POLY$ אזי $P=NP$ **הוכחה:** מההנחה - ניתן לקבל פתרון פולי' ל-SAT, בהנתן פסוק φ נחשב את $\#SAT\left(\varphi\right)$ - אם הוא אפס - הפסוק לא ספיק. אחרת - ספיק! ■

[21] כמובן - בהנחה ו-$P\neq NP$

נעיר כי הכיוון ההפוך אינו טרוויאלי (זו היא בעייה פתוחה). כלומר - האם מהעובדה שאנחנו יודעים האם קיימת השמה מספקת ניתן לגזור ביעילות מה הוא מספר ההשמות המספקות?

טענה 8.7 לכל $d \geq 1$. אם קיים אלגוריתם d-קירוב עבור $\#SAT$. כלומר - A יעיל המקיים לכל φ -

$$\frac{\#SAT(\varphi)}{d} \leq A(\varphi) \leq d \cdot \#SAT(\varphi)$$

אזי $P = NP$.

הוכחה: אם $\#SAT(\varphi) > 0$ כלומר, φ ספיק. אזי בשני אגפי אי השוויון נקבל מספר גדול מאפס, ולכן בהכרח גם $A(\varphi)$ יתן מספר גדול מאפס. אחרת - $\#SAT(\varphi) = 0$ - שני אגפי אי השוויון הם אפס, ולכן גם $A(\varphi) = 0$. בפרט - $A(\varphi)$ מספק לנו פתרון פולינומיאלי ל-SAT, וכיוון ש-$SAT \in NPC$ זה מוכיח כי $P = NP$. ■

מה עם קירוב חיבורי? ראינו כי קירוב כפלי לא מועיל, כי הבעיה היא להבדיל בין מצב שבו $\#SAT$ שווה לאפס למצב שבו הוא שונה מאפס, וכפל בקבוע לא מועיל כאן.

טענה 8.8 לכל $d \geq 0$, אם קיים אלגוריתם d-קרוב חיבורי יעיל עבור $\#SAT$, כלומר A יעיל המקיים לכל φ -

$$\#SAT(\varphi) - d \leq A(\varphi) \leq \#SAT(\varphi) + d$$

אזי $P = NP$.

הוכחה: נראה אלגוריתם פולי' B עבור SAT -

- $k \triangleq \lceil log(d) \rceil + 2$
- נגדיר - $\varphi' = \varphi \wedge ((y_1 \vee \overline{y_1}) \wedge (y_2 \vee \overline{y_2}) \vee \cdots (y_k \vee \overline{y_k}))$
- חשב את $A(\varphi')$ וקבל אמ"מ הערך המתקבל גדול או שווה ל-$2d$.

מתקיים -

- B פולי (כי A פולינומי ב-$|\varphi'|$ ו-$|\varphi'| = O(|\varphi|)$ - כזכור, d קבוע)
- $\#SAT(\varphi') = 2^k \#SAT(\varphi)$
- אזי - $\varphi \notin SAT$
  - $\varphi' \notin SAT$ ולכן - $\#SAT(\varphi) = 0$, ולכן $A(\varphi') \leq d$ ולכן B דוחה.
- אזי - $\varphi \in SAT$
  - $\#SAT(\varphi) \geq 0$ ולכן $\#SAT(\varphi') \geq 2^d = 4d$ ובפרט $A(\varphi') \geq 3d \geq 2d$ ולכן B מקבל.

■

9 מערכות הוכחה

באופן לא מאד פורמלי, מערכת הוכחה מורכבת משני צדדים - "מוכיח" ו-"מוודא".
לשם הפשטות, נדבר על טענות מהצורה $x \in L$.

שלמות אם הטענה נכונה $\Leftarrow$ קיימת הוכחה שהמוודא יקבל

נאותות אם הטענה לא נכונה $\Leftarrow$ לא חשוב מה יעשה המוכיח, המוודא "דוחה"

אנחנו נתעניין במערכות הוכחה "יעילות" - כאלו שהמוודא יכול לבדוק כי ההוכחה נכונה ביעילות (כלומר - פולינומי באורך הטענה x). מערכות כאלו הן NP.

- לכל שפה ב-NP קיימת מערכת הוכחה

הוכחה: (לכך שמערכות שהשפות שעבורן קיימת מערכת הוכחה יעילה היא NP)

- תהא $L \in NP$, כלומר קיים R_L מתאים. כעת הוכחה היא y כך ש-$(x,y) \in R_L$.
- תהא L שפה שקיימת עבורה מערכת הוכחה, ונוכיח $L \in NP$ ע"י R_L מתאים -

$$R_L = \{(x,y) \mid V(x,y) \text{ accetps}\}$$

כאשר V הוא המוודא.

■

10 תמונת העולם[22]

הכרנו את המחלקות P ו-NP כך ש-$P \subseteq NP$, והשאלה הגדולה היא האם ההכלה היא ממש (כלומר $P \neq NP$) או שיש שוויון ($P = NP$).
לפני כן ראינו גם את המחלקה R כך שמתקיים $NP \subset R$, וכן את RE ו-$coRE$ כך ש-

$$\begin{aligned} RE \cap coRE &= R \\ R &\subset RE \\ R &\subset CO - RE \end{aligned}$$

נשים לב שבמקרים האחרונים ההכלה היא ממש (כלומר, $R \neq RE$ וכו'...).
בתרגול ראינו גם את המחלקה $PSPACE$, מחלקת השפות הניתנות לזיהוי ע"י מ"ט בעלת זיכרון פולי' באורך הקלט. בפרט ראינו את הקשרים -

$$P \subseteq NP \subseteq PSPACE \subseteq R$$

גם כאן אנחנו לא יודעים האם ההכלה היא ממש או שיש שוויון? כלומר האם למשל $P = PSPACE$? (תשובה חיובית תגרור מיידית $P = NP$, תשובה שלילית לא תיתן מידע על היחס בין P ו-NP). אנו לא יודעים גם האם $NP = PSPACE$?
מחלקה נוספת היא EXP - אוסף השפות שניתנות לזיהוי ע"י מ"ט בזמן אקספוננציאלי באורך הקלט) -

$$EXP \triangleq \left\{ L \mid L \in DTIME\left(2^{n^c}\right),\ c \text{ is constant} \right\}$$

ומתקיים -

$$PSPACE \subseteq EXP \subseteq R$$

(על פי אותה הוכחה ש-$PSPACE \subseteq R$, "ספירת קונפיגורציות").

[22] הרצאה ארבע-עשרה - 22/6/10

10.1 המחלקה $coNP$

מוגדרת כאוסף כל השפות L כך ש־$\overline{L} \in NP$.

$$coNP \triangleq \{L \mid \overline{L} \in NP\}$$

לדוגמא - $\overline{SAT}$, אוסף כל פסוקי ה־CNF שאינם ספיקים.

$$\overline{SAT} \in coNP$$

10.1.1 שפה $coNP$ שלמה

היא שפה שמקיימת

- $L \in coNP$
- לכל $L' \in coNP$ קיימת רדוקציה פולינומית - $L' \leq_p L$

מסקנה 10.1 L היא NP שלמה $\Leftrightarrow$ $\overline{L}$ היא $coNP$ שלמה

הוכחה:

- $\overline{L} \in coNP \Leftarrow L \in NP \Leftarrow L \in NPC$
- $\Leftarrow L' \in coNP$ תהא
  $\Leftarrow \overline{L'} \in NP$
  $\Leftarrow \overline{L'} \leq_p L$
  $L' \leq_p \overline{L}$

■

למה 10.2 $P \subseteq coNP$

הוכחה: מ"ט דטרמיניסטית פולי' היא מקרה פרטי של מ"ט א"ד פולי' הדרושה כדי להיות ב־$coNP$.
לחילופין, $L \in P \Leftarrow \overline{L} \in P \Leftarrow \overline{L} \in NP \Leftarrow L \in coNP$. ■

למה 10.3 $coNP \subseteq PSPACE$ **הוכחה:** (ישירה) - מעבר כל עץ החישוב של מ"ט א"ד או בדיקת כל ה־y ביחס R_L המתאים. ■

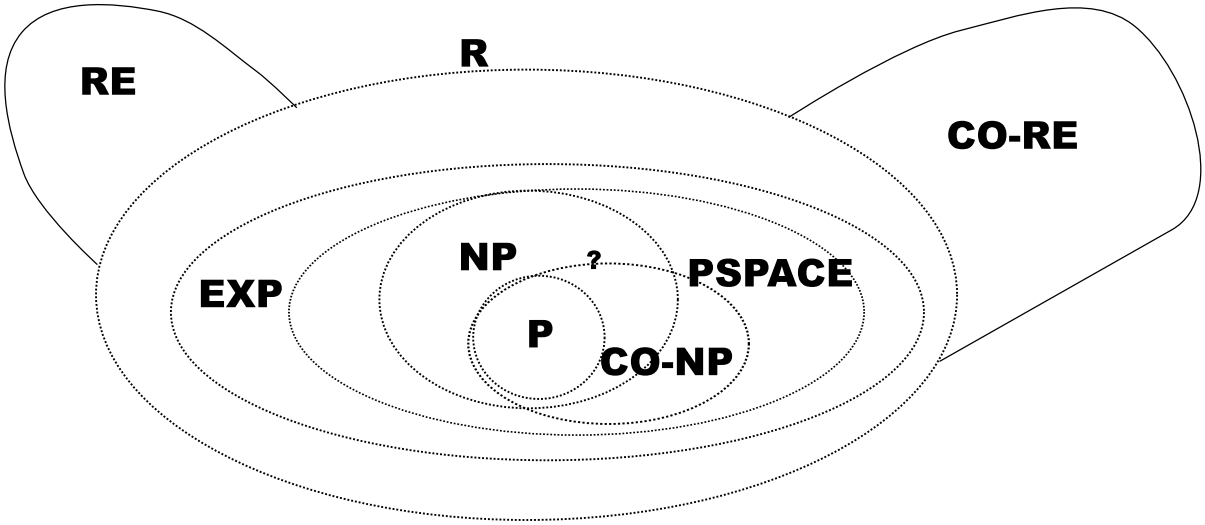
הוכחה: (שנייה) - $L \in coNP \Leftarrow \overline{L} \in NP \Leftarrow \overline{L} \in PSPACE \Leftarrow L \in PSPACE$ (בגלל סגירות למשלים של $PSPACE$). ■

הוכחה: (שלישית) - $\overline{SAT} \in PSPACE$ כיוון שהאלגוריתם הנאיבי שעובר על כל ההשמות אחת אחת צורך זיכרון פולינומיאלי (אמנם, הוא רץ זמן אקספוננציאלי, אבל לא אכפת לנו לצורך השייכות ל־$PSPACE$).
לכל שפה אחרת $L \in coNP$ קיימת רדוקציה פולינומית אל $\overline{SAT}$ (כיוון ש־$\overline{SAT} \in CO - NPC$) ולכן קיים אלגוריתם שמכריע אותה בזמן פולי' (מבצע את הרדוקציה ובודק באמצעות האלגוריתם על $\overline{SAT}$). ■

10.1.2 שאלות פתוחות

- האם $P = coNP$? (אם כן, בפרט $P = NP$)
- האם $NP = coNP$? (בניסוח שקול, האם $\overline{SAT} \in NP$?)

10.2 תמונה גרפית



כפי שכבר ראינו, עבור רוב ההכלות אנחנו לא יודעים האם מדובר בשוויון או בהכלה ממש. בכל זאת, יש כמה דברים שאנחנו יודעים לקבוע -

טענה 10.4 קיימת שפה $L \in R \backslash P$ **הוכחה:** נבנה מ"ט U שהשפה שלה $L = L(U)$ היא השפה המבוקשת.
U על קלט x תפעל באופן הבא -

- נפרש את x כ- $1^k 0 \langle M \rangle$, כלומר (נספור כמה 1ים מופיעים לפני ה-0 הראשון, המספר הזה יסומן כ-k, ושאר הקלט יפורש בתור מ"ט M, אם לא ניתן לפרש בצורה כזו - נדחה).
- הרץ את M על x למשך 2^k צעדים (מ"ט אוניברסלית + מונה) -
  - אם M עצרה במהלך 2^k הצעדים. עצור וקבל/דחה. להפך מ-M
  - אם M לא עצרה. דחה.

אבחנה

- $L \in R$ (מהבניה, תמיד עוצרת)
- נותר להוכיח ש-$L \notin P$.

נניח בשלילה כי $L \in P$, לכן קיימת מ"ט M הרצה זמן $p(n)$ ומקבלת את L.
נבחר k מספיק גדול, כזה ש-$2^k > p(k + 1 + |\langle M \rangle|)$ (ברור שקיים כזה, כי הפונקציה באגף שמאל אקספוננציאלית ובימין פולינומית ב-k, ו-$1 + |\langle M \rangle|$ הוא מספר קבוע).
נתבונן בריצת U על $x = 1^k 0 \langle M \rangle$. על פי האופן שבו בחרנו את k, בהכרח הסימולציה של M על x תסתיים בתוך פחות מ-2^k צעדים, ולכן U ו-M יחזירו תשובות הפוכות על x, ובפרט $L(U) \neq L(M)$. זו סתירה, ולכן לא קיימת מ"ט פולי' המקבלת את L, ומתקיים $P \subset R$ (הכלה ממש, $R \neq P$). ■

מסקנה 10.5 (מההוכחה) קיימת $L \in EXP \backslash P$.
ולכן לפחות אחת מההכלות - $P \subseteq NP \subseteq PSPACE \subseteq EXP$ היא הכלה ממש (ולא שוויון).

מסקנה 10.6 (מההוכחה) קיימת $L \in R \backslash EXP$ (ואז היא לא ב־$PSACE$ וכו'...)
באופן דומה מאד להוכחה, ניתן לבנות מכונה שעוצרת עבור 2^{2^k} (למשל) צעדים.

10.3 שפות $PSPACE$־שלמות

הגדרה 10.7 השפה L היא $PSPACE$ שלמה, אם מתקיים -

- $L \in PSPACE$
- לכל $L' \in PSPACE$ קיימת $L' \leq_p L$

טענה 10.8 אם L היא $PSPACE$ שלמה אז -

- $P = PSPACE \Leftrightarrow L \in P$
- $NP = PSPACE \Leftrightarrow L \in NP$

10.3.1 דוגמא ל־$L \in PSPACE\ Complete$

ניתן להתבונן על SAT בתור -

$$SAT = \{\psi |\ \psi = \exists x_1 \exists x_2 \ldots \exists x_n\ \varphi(x_1, x_2 \ldots x_n)\ \ \varphi \text{ is CNF, } \psi \text{ is TRUE}\}$$

ובהתאמה -

$$\overline{SAT} = \{\psi |\ \psi = \forall x_1 \forall x_2 \ldots \forall x_n\ \neg\varphi(x_1, x_2 \ldots x_n)\ \ \varphi \text{ is CNF, } \psi \text{ is TRUE}\}$$

נגדיר כעת את -

$$TQBF = \{\psi |\ \psi = Q_1 x_1 Q_2 x_2 \ldots Q_n x_n\ \varphi(x_1, x_2 \ldots x_n)\ \ \varphi \text{ is CNF, } \psi \text{ is TRUE}\}$$

כאשר $Q_i \in \{\forall, \exists\}$.

משפט 10.9 $TQBF$ היא $PSPACE$ שלמה. **הוכחה:** <לא תינתן הסמסטר> ■

11 קריפטוגרפיה

קריפטוגרפיה הוא התחום שעוסק בהצפנה, הזדהות, אימות וכו'...
אפשר לחשוב על מערכת הצפנה "מושלמת" -
נתבונן על S, R (שולח ומקבל) כאשר S רוצה לשלוח ל־R הודעה m בת n ביטים, ועל קו התקשורת בינהם יש מאזין שרוצה ליירט את ההודעה.
על מנת להבטיח את מעבר ההודעה בצורה בטוחה, S ו־R יבחרו מראש בצורה בטוחה מחרוזת אקראית k בת n ביטים.
כעת S יחשב את -

$$c = m \oplus k$$

וישלח את c על הרשת. R יקבל את c ויבצע -

$$m = c \oplus k$$

ויקבל את m המקורי.
לעומת זאת, מאזין שמצוטט לקו לא יכול לדעת שום דבר על m, כיוון שמבלי ידע על k (שכאמור נבחר באקראי) הידיעה של c לא נותנת לו שום אינפורמציה על m.

חסרונות

- k צריך להקבע מראש
- השימוש במפתח הוא חד פעמי! [23]
- יש צורך במפתח ארוך

גישה אחרת אם נוותר על הדרישה שהמאזין לא ידע כלום על ההודעה, ובמקום זה נדרוש שאם המאזין יירט את ההודעה הוא יצטרך לפתור בעיה קשה מאד.

11.1 פרוטוקול החלפת מפתחות של Diffie Hellman (1976)

הנחה חישובית - לא קיים אלגוריתם יעיל עבור הבעיה הבאה - בוחרים באקראי ראשוני p באורך n, ויוצר g עבור המספרים מ-1 עד $p-1$. ומספר $a \in \{1, \ldots p-1\}$. הקלט -

$$p, g, g^a \ (mod\ p)$$

והפלט הדרוש - a. ההנחה החישובית היא שהבעיה של מציאת a מתוך $p, g, g^a \ \ (mod\ p)$ היא קשה.

הפרוטוקול

S בוחר באקראי את p, g. בוחר באקראי a ושולח ל-R את $p, g, A = g^a \ (mod\ p)$
R בוחר באקראי b ושולח את $B = g^b \ \ (mod\ p)$.
המפתח המשותף - $k = g^{ab}$.

הוכחת נכונות

S יכול לחשב את המפתח ע"י - $k = B^a$.
R יכול לחשב את המפתח ע"י $k = A^b$.
המאזין לא יכול לחשב (ביעילות, אילו הוא היה יכול היינו מקבלים אלגוריתם שסותר את ההנחה החישובית) את a, b. מכאן "אינטואיטיבית" המאזין לא יודע מה הוא k.

[23] אחרת, המאזין יכול לבצע xor בין שתי ההודעות, ולקבל את $m_1 \oplus m_2$, כעת יש לו די הרבה מידע.