הרצאה 3 – יום ראשון 27.7

מכונת טיורינג אוניברסלית – *Universal Turing Machine (UTM)*

מ"ט אוניברסלית U מקבלת בתור קלט "תיאור" של מ"ט אחרת M, ומחרוזת x שהיא הקלט של M, וממדלת את הריצה של M.

- U עוצרת אמ"מ M עוצרת.
- התשובה המוחזרת מ-U זהה לתשובה המוחזרת מ-M.
- בצורה יעילה (בערך).

תזת *Church-Turing* אומרת שזה אפשרי. כדי לעשות את זה, נצטרך לפרמל את התיאור של מ"ט ע"י מחרוזות, ואת הרעיון של "ריצה".

קידודים – *Encodings*

כל אובייקט סופי אפשר לתאר ע"י מחרוזת ביטים $\langle a\rangle$, או ע"י מחרוזת מא"ב סופי (אם זה יותר נוח).

כלומר מספרים שלמים, רציונליים, וקטורים ומטריצות מעל השלמים, גרפים וכו. ממשיים אי אפשר – זה לא מחרוזת סופית.

הקידוד צריך להיות "סביר":

- אפשר (וקל) להבין את המאפיינים הפשוטים של האובייקט.
- אפשר לבדוק את הנכונות (ומהר).
- אפשר למפות כל ייצוגים לא נכונים למצב דחייה.

יכול להיות יותר מייצוג אחד לכל אובייקט.

דוגמאות

דוגמה 1: נקודד גרף $G \coloneqq (V,E)$. נשתמש בא"ב $\{0,1,\#\}$. נקודד את המספרים השלמים בבינארי, ונגדיר:

$$n \coloneqq |V|, \qquad m \coloneqq |E|, \qquad V=\{1,2,\dots,n\}, \qquad (v_i,u_i)\in E$$

$$\langle G\rangle \coloneqq n\#m\#v_1\#u_1\#v_2\#u_2\#\dots\#v_m\#u_m$$

קידוד מכונת טיורינג

א"ב הסרט	קידוד – מחרוזת אונארית באורך $\lvert\Gamma\rvert$
$blank$	$000\dots000$
$start$	$111\dots111$
סימנים אחרים	אפס, ואז אפסים (לפי אינדקס הסימן) ואז אחדות
מצבים	**קידוד – מחרוזת אונארית באורך $\lvert Q\rvert$**
q_S	$111\dots111$
q_Y	$000\dots001$
q_N	$000\dots000$
מצבים אחרים	אפסים (לפי אינדקס המצב) ואז אחדות
כיוונים (של ראש הקריאה / כתיבה)	**קידוד – מחרוזת אונארית באורך 2**
L	11
R	01
S	00
פקודה	**קידוד**
$(q_1,c_1)\to(q_2,c_2,d)$	$\langle q_1\rangle\#\langle c_1\rangle\#\langle q_2\rangle\#\langle c_2\rangle\#\langle d\rangle$

אז קידוד של תכנית: $1^{|\Gamma|}\#1^{|Q|}\#1$, ואז קידוד של כל הפקודות.

כדי לקודד מ"ט עם k סרטים, הקידוד של כל c,d הוא:

$$\langle c\rangle \coloneqq \langle c[1]\rangle\langle c[2]\rangle\dots\langle c[k]\rangle, \qquad \langle d\rangle \coloneqq \langle d[1]\rangle\langle d[2]\rangle\dots\langle d[k]\rangle$$

פשוט מתאר את הכתיבה ותזוזה של כל הסרטים, לפי הסדר.

UTM יכול לממש כל TM סופי

משפט: יהי $\langle M,x\rangle \coloneqq \langle M\rangle\#\langle x\rangle$. אזי, קיימת מ"ט U כך ש:

לכל א"ב סופי Σ, ולכל מ"ט M על Σ, ולכל מחרוזת $x\in\Sigma^*$, מתקיים:

$U(\langle M,x\rangle)$ עוצר אמ"מ $M(x)$ עוצר.

מספר הצעדים ש-U מבצע הוא לכל היותר ריבועי במספר הצעדים ש-M מבצע (כאשר הגודל של M הוא קבוע, בעניין הזה).

אם U היא מ"ט עם 2 סרטים, אז מספר הצעדים ש-U מבצע הוא לכל היותר לינארי במספר הצעדים ש-M מבצע.

הוכחה ע"י בנייה: נבנה מ"ט עם 2 סרטים:

- סרט 1 – הקידוד של M, לקריאה בלבד. $1^{|\Gamma|}\#1^{|Q|}\#1^k\#\dots instructions\dots\#x$.
- סרט 2 – הסרט של המידול עצמו.

בהתחלה, בסרט 2 יש $\#1^{|Q|}\#x$.

ה-TM סורק את המצב (ה-$|Q|$ מקומות אחרי ה-# הראשון) ואת התו הנוכחי (נמצא אחרי ה-# השני) ומוצא את הצירוף הזה בסרט 1 (בחלק של הפקודות).

הוא קורא את הפקודה, וכותב את מה שצריך על סרט 2 (המצב החדש והתו) ומזיז את הראש.

רק $|\Gamma|+|Q|+2$ מקומות של סרט 2 מושפעים בכל צעד כזה – שזה מספר קבוע (בהינתן M).

אז זמן הריצה של U לינארי בגודל התוכנית (הפקודות) של M.

כשיש מצב עצירה (של M) על סרט 2, U נכנס למצב העצירה המתאים ומסיים.

ה"מסלול" של הקונפיגורציות שמתואר בסרט 2 מתאר באופן מדויק את התהליך של M.

אם $M(x)=1$ אז נקבל $U(\langle M,x\rangle)=1$. אם $M(x)=0$ אז נקבל $U(\langle M,x\rangle)=0$. אם $M(x)=\infty$ אז נקבל $U(\langle M,x\rangle)=\infty$.

צריך גם שלב של בדיקת תקינות הקלט עבור $\langle M,x\rangle$.

R, RE, ומחלקות שפות

שפה של מ"ט – תזכורת:

נאמר שמ"ט M מקבלת שפה L אם: $\forall x\in L,\ M(x)=1$.

נאמר ש-L היא השפה של מ"ט M ונרשום $L=L(M)\coloneqq\{x\in\Sigma^*\mid M(x)=1\}$.

המשלים של שפה L זה קבוצת כל המילים מעל אותו א"ב שלא ב-L. $\bar{L}=\overline{L(M)}\coloneqq\{x\in\Sigma^*\mid M(x)=0\vee M(x)=\infty\}$.

$$\bar{L}=\Sigma^*\setminus L,\qquad \overline{L(M)}=\Sigma^*\setminus L(M)$$

אם קיים DTM כזה עבור L, נאמר ש-L כריעה למחצה, או *Recursively Enumerable*.

$$RE\coloneqq\{L\mid L\ is\ Recursively\ Enumerable\}$$

נאמר ש-M מכריעה שפה L אם M תמיד עוצרת (לכל x מעל הא"ב) ומתקיים:

$$x\in L\Leftrightarrow M(x)=1,\qquad x\notin L\Leftrightarrow M(x)=0$$

אם קיים DTM כזה עבור L, נאמר ש-L כריעה, או רקורסיבית.

$$R\coloneqq\{L\mid L\ is\ recursive\}$$

דוגמאות של שפות כריעות למחצה:

- שפות טריוויאליות, $\emptyset, \Sigma^*$.
- שפות רגולריות (כי אפשר למדל *DFA* ע"י *TM*).
- שפות חסרות הקשר (כי אפשר למדל *PDA* ע"י *TM*).
- שפת כל המספרים הראשוניים.
- שפת כל הגרפים הקשירים.
- שפת כל הגרפים ההמילטוניים.

תכונות של R

<u>טענה</u>: R סגורה תחת משלים, כלומר $L \in R \implies \bar{L} \in R$.

<u>הוכחה</u>: תהי $L \in R$. כלומר קיים M שמקבל כל $x \in L$ ודוחה כל $y \notin L$.

נבנה מ"ט M' שזהה ל-M למעט זה שנחליף את q_Y, q_N. אז M' תקבל כל $y \notin L$ ותדחה כל $x \in L$.

<u>טענה</u>: R סגורה תחת איחוד, כלומר $L_1, L_2 \in R \implies L_1 \cup L_2 \in R$.

<u>הוכחה</u>: יהיו $L_1, L_2 \in R$. כלומר קיימים M_1, M_2 מתאימים עבורן.

נבנה מ"ט M שמריצה את $M_1(x), M_2(x)$ ומקבלת את x אם אחד מהם מגיע למצב מקבל.

<u>טענה</u>: R סגורה תחת חיתוך, כלומר $L_1, L_2 \in R \implies L_1 \cap L_2 \in R$.

<u>הוכחה</u>: יהיו $L_1, L_2 \in R$. כלומר קיימים M_1, M_2 מתאימים עבורן.

נבנה מ"ט M שמריצה את $M_1(x), M_2(x)$ ומקבלת את x אם שניהם מגיעים למצב מקבל.

<u>טענה</u>: R סגורה תחת שרשור, כלומר $L_1, L_2 \in R \implies L_1 \circ L_2 \in R$.

<u>הוכחה</u>: יהיו $L_1, L_2 \in R$. כלומר קיימים M_1, M_2 מתאימים עבורן. בהינתן מילה w, אנחנו רוצים לבדוק אם קיימים $x \in L_1, y \in L_2$ כך ש- $w = xy$.

נבנה מ"ט M: לכל $i \in [|w| + 1]$, נבדוק אם $M_1(w[:i])$ מקבלת. אם כן, נבדוק אם $M_2(w[i:])$ מקבלת. אם כן, סיימנו.

אם בדקנו את כל האפשרויות של $M_1(w[:i])$ ואף מילה לא התקבלה, נדחה.

שפות כריעות למחצה – *Recursively Enumerable*

מה הכוונה ב-*enumerable*? התרגום של *enumerate* הוא למנות (כלומר לספור). שפה שהיא *enumerable* הכוונה שיש *enumerator* (מונה) עבורה.

מונה הוא מ"ט עם "מדפסת" – פשוט סרט שעליו רק כותבים ומתקדמים ימינה. בין מחרוזות שמדפיסים, מדפיסים #.

מצב התחלה q_S, עם קלט ריק. התוכנה עובדת עד שהמכונה עוצרת (אולי עד אינסוף).

במהלך הריצה, המ"ט מדפיסה מחרוזות (יש פקודות מיוחדות לזה).

כל החרוזות (עד כדי חזרות) שמודפסות עד שהמכונה עוצרת, מהוות את השפה L. (עדיין יכול להיות $|L| = \infty$).

תרגיל: נבנה מונה עבור השפה $L := \{0^n 1^n\}$.

נשתמש בשני סרטים: הסרט הראשון סופר את ה-n, והסרט השני זה הפלט. נרוץ:

1. נתחיל מההתחלה של סרט 1, במצב התחלה. נכתוב 0 בפלט ונעבור למצב 0.
2. במצב 0, כל עוד קוראים x, נכתוב 0 בפלט ונזוז ימינה.
3. אם הגענו לסוף של סרט1, נכתוב 1 בסוף ונזוז שמאלה, נכתוב 1 בפלט, ונעבור למצב 1.
4. במצב 1, כל עוד קוראים x, נזוז שמאלה ונכתוב 1 בפלט.
5. כשמגיעים להתחלה של סרט1, נכתוב # בפלט ונעבור למצב *hash*.
6. במצב *hash*, עוברים ימינה בסרט1 וכותבים 0 בפלט, וחוזרים למצב 0.

הפלט יהיה: ... #000111#0011#01.

אם רוצים להחשיב את המילה הריקה (כלומר $\varepsilon = 0^0 1^0$) אז נוסיף מצב התחלה ספציפי שכותב #, ואז עובר למצב התחלה ה"רגיל".

<u>משפט</u>: שפה היא כריעה-למחצה אמ"מ יש מונה עבורה.

<u>הוכחה</u> – כיוון ראשון: יהי מונה E עבור שפה L. נבנה מ"ט M שמקבלת את L כך:

בהינתן קלט x, נריץ את E ונקבל את x אם x מודפסת בתור מחרוזת של E.

אם x מודפסת, היא תתקבל. אם E עוצרת ו-x לא הודפסה, היא לא מתקבלת. אם E לא עוצרת, אז x לא מתקבלת.

מכיוון שהראינו בעבר שמ"ט הן מודולריות, מכונה שהיא שני מ"ט היא בעצמה מ"ט.

כיוון שני – תהי מ"ט M שמקבלת את L. נבנה מונה E עבור L:

- שלב 0: E מריצה 0 צעדים של M על קלט ריק, ומדפיסה כל מחרוזת מתקבלת.
- שלב 1: E מריצה צעד אחד של M על כל הקלטים באורך לכל היותר 1, ומדפיסה כל מחרוזת מתקבלת.
- שלב 2: E מריצה 2 צעדים של M על כל הקלטים באורך לכל היותר 2, ומדפיסה כל מחרוזת מתקבלת.
- וכו...

כל מילה $x \in L$ באורך n שמקבלת ע"י M אחרי m צעדים, תודפס ע"י E בשלב: $\max(n,m)$.

כל שלב הוא סופי כי חסמנו את מספר הצעדים – כל שלב יהיה לכל היותר $n \cdot |\Sigma|^n$ צעדים.

כל מילה $x \notin L$ לא מתקבלת ע"י M ולכן לא תודפס ע"י E.

תכונות של RE

<u>טענה</u>: R סגורה תחת איחוד, כלומר $L_1, L_2 \in RE \Longrightarrow L_1 \cup L_2 \in RE$.

<u>הוכחה</u>: יהיו $L_1, L_2 \in RE$. כלומר קיימים M_1, M_2 מתאימים עבורן.

נבנה מ"ט M שתריץ את $M_1(x), M_2(x)$ כך:

בשלב ה-i, נריץ i צעדים של $M_1(x), M_2(x)$ ונחזיר 1 אם אחד מהם הגיע למצב מקבל. אם שניהם הגיעו למצב דוחה, נחזיר 0.

אם M_1 קיבלה את x אחרי n צעדים, או M_2 קיבלה את x אחרי m צעדים, אז M תקבל את x בשלב $\min(n,m)$.

אם M_1 דחתה את x אחרי n צעדים, ו-M_2 דחתה את x אחרי m צעדים, אז M תדחה את x בשלב $\max(n,m)$.

אם שניהם לא עוצרים, או אחת דוחה והשנייה לא עוצרת, אז M לא עוצרת.

בסה"כ, M מקבלת את $L_1 \cup L_2$ אז $L_1 \cup L_2 \in RE$.

<u>טענה</u>: R סגורה תחת חיתוך, כלומר $L_1, L_2 \in RE \Longrightarrow L_1 \cap L_2 \in RE$.

<u>הוכחה</u>: יהיו $L_1, L_2 \in RE$. כלומר קיימים M_1, M_2 מתאימים עבורן.

נבנה מ"ט M שתריץ את $M_1(x), M_2(x)$ כך:

בשלב ה-i, נריץ i צעדים של $M_1(x), M_2(x)$ עד ששניהם מגיעים למצב מקבל, או עד שאחד מהם דוחה.

(אם אחד הגיע למצב מקבל אחרי n צעדים, נמשיך להריץ אותו וזה לא משנה, כי גם בצעד ה- $n+1$ המצב יהיה מקבל).

אם M_1 קיבלה את x אחרי n צעדים, ו- M_2 קיבלה את x אחרי m צעדים, אז M תקבל את x בשלב $\max(n,m)$.

אם M_1 דחתה את x אחרי n צעדים, ו-M_2 דחתה את x אחרי m צעדים, אז M תדחה את x בשלב $\min(n,m)$.

אם אחד מהם לא עוצר, אז M לא עוצרת.

בסה"כ, M מקבלת את $L_1 \cap L_2$ אז $L_1 \cap L_2 \in RE$.

<u>טענה</u>: R סגורה תחת שרשור, כלומר $L_1, L_2 \in RE \Longrightarrow L_1 \circ L_2 \in RE$.

<u>הוכחה</u>: יהיו $L_1, L_2 \in RE$. כלומר קיימים M_1, M_2 מתאימים עבורן.

נבנה מ"ט M: בשלב ה-i, נבצע:

לכל $j \in [|x| + 1]$, נריץ i צעדים של $M_1(x[:j])$. אם הוא מקבל, נריץ i צעדים של $M_2(x[j:])$. אם הוא מקבל, נחזיר 1.

אם לכל j, $M_1(x[:j])$ דוחה, נדחה. אם לכל j שעבורו $M_1(x[:j])$ מקבל, $M_2(x[j:])$ דוחה, נדחה.

אם אין חלוקה של x כך ששניהם עוצרים, אז M לא עוצרת.

בסה"כ, M מקבלת את $L_1 \circ L_2$ אז $L_1 \circ L_2 \in RE$.