הרצאה 6: שפות מתקבלות ושפות מוכרעות ע"י מ"ט

מרצה: אורי שטמר

Based on previous iterations of this course, given by Nir Bitansky, Rotem Oshman, Iftach Haitner, and Omer Paneth.

כפי שראינו בהרצאה הקודמת, מ"ט תופסים אלגוריתמים כלליים. נרצה כעת להבין מהן המגבלות של אלג' כלליים.

<u>תזכורת:</u> בהינתן מ"ט M סימנו:
$L(M) =$ אוסף כל המילים ש- M מקבלת

מי לא נמצא ב- $L(M)$? מילים ש- M דוחה ומילים ש- M לא-עוצרת עליהם (אם יש כאלה...)

הגדרה:

$$RE = \left\{ L \subseteq \Sigma^* : \begin{array}{c} M \text{ קיימת מ"ט} \\ L \text{ המקבלת את} \\ L = L(M) \text{ כלומר} \end{array} \right\}$$

השם RE פירושו Recursively Enumerable (ניתן למניה רקורסיבית). מקור המילה "Enumerable" הוא שלשפות ב- RE יש אלג' המבצע אנומרציה של כל המילים בשפה, כפי שתיכף נראה... המונח "Recursive" מגיע מהקשר היסטורי: לפני פיתוח מכונות טיורינג, מושג החישוביות התבסס על פונקציות רקורסיביות שניתן לחשב אותן במספר סופי של צעדים לפי כללים מסוימים. זה היה מודל תאורטי לחישוב שהשתמשו בו בשנים 1930-1936, ע"י Gödel, Kleene, Church, עד שב 1936 הוצגו מ"ט.

הגדרה:
נאמר שמ"ט M <u>מכריעה</u> שפה L אם מתקיים:
(א) $L = L(M)$
(ב) לכל $w \in \Sigma^*$ מתקיים ש- M עוצרת על w

סימון:

$$R = \left\{ L \subseteq \Sigma^* : \begin{array}{c} M \text{ קיימת מ"ט} \\ L \text{ המכריעה את} \end{array} \right\}$$

שפות ב- R נקראות "שפות כריעות" או "שפות רקורסיביות". שפות ב- RE נקראות "כריעות למחצה" או "ניתנות למניה רקורסיבית" או "ניתנות לקבלה".

שימו לב: $R \subseteq RE$

כעת נראה מהו מקור השם "אנומרציה" ב- RE:

הגדרה:
מ"ט E הינה <u>מונה עבור שפה</u> $L \subseteq \Sigma^*$ אם:

- ל- E סרט פלט לכתיבה חד-פעמית (תא אשר המכונה מאתחלת לא משתנה)
- א"ב פלט $\Sigma \cup \{\$\}$ כאשר בה"כ $\$ \notin \Sigma$
- בריצה של E על הקלט הריק:
  - לכל $x \in L$, המחרוזת $\$x\$$ כתובה על סרט הפלט לאחר מספר סופי של צעדים
  - לכל $x \notin L$, המחזורת $\$x\$$ לעולם לא כתובה בפלט

טענה: $L \in RE$ אם"ם קיים עבורה מונה

<u>הוכחה:</u> כיוון ראשון – בהינתן מונה E עבור L נראה כי $L \in RE$

נבנה מ"ט M המקבלת את L באופן הבא:
$M(x)$ מריצה את המונה E ומקבלת אם וכאשר $\$x\$$ מופיע על סרט הפלט.

כיוון שני – בהינתן מ"ט M המקבלת את L נבנה מונה עבור L

יהי $\sigma_1, \sigma_2, \ldots$ סידור לקסיקוגרפי של Σ^*

רעיון טבעי אבל שגוי: לכל i נריץ $M(\sigma_i)$ ונוסיף את σ_i לרשימה אם M מקבלת. מה הבעיה כאן?

נבנה מונה E באופן הבא:

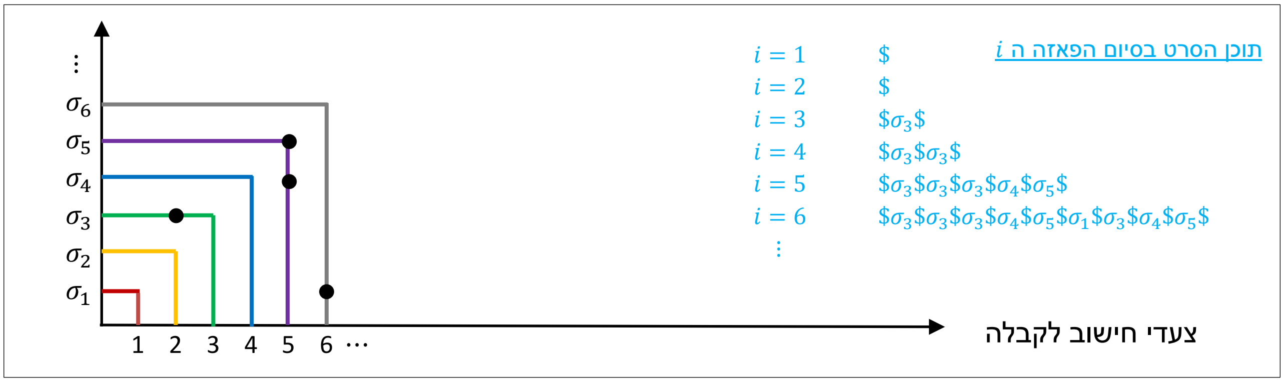
- כותב $ בתא הפלט הראשון לאתחול הרשימה.
- לכל $i \geq 1$:
  - מריץ את M על כל אחד מהקלטים $\sigma_1, \ldots, \sigma_i$ במשך i צעדים
  - בכל פעם שמתקבלת מילה σ_j בתהליך, מוסיפים את $\sigma_j\$$ לסרט הפלט

מתקיים:

- אם $\$\sigma_j\$$ נכתב לסרט אז אחת הריצות של M קיבלה את σ_j ולכן $\sigma_j \in L(M)$
- אם $\sigma_j \in L(M)$ ומתקבלת תוך t צעדים של M אזי באיטרציה $i = \max\{j, t\}$ נקבל ש- $\$\sigma_j\$$ תכתב לסרט הפלט

שאלה: למה השתמשנו דווקא בסדר לקסיקוגרפי? האם כל סדר יעבוד?
תשובה: לא חייבים דווקא סדר לקסיקוגרפי, הוא פשוט נח. טכנית לא כל סדר יעבוד כאן. אנחנו צריכים סדר עם שתי התכונות הבאות: (1) בהינתן איבר קל לנו למצוא את העוקב שלו בסדר (2) לכל איבר $\sigma_j \in \Sigma^*$ יש מספר סופי של איברים קטנים ממנו בסדר. לא כל סדר מלא על Σ^* בהכרח יקיים את שתי התכונות האלה...

שאלה: האם המונה שבנינו פולט את המילים בשפה לפי סדר לקסיקוגרפי? לא! בציור:



שימו לב שהמופע הראשון של σ_1 מתרחש אחרי ש $\sigma_3, \sigma_4, \sigma_5$ כבר הופיעו.
האם ניתן לתקן את המונה מונה כך שיכתוב את המילים לפי סדר לקסיקוגרפי?

טענה: $L \in R$ אם"ם קיים עבורה מונה לקסיקוגרפי, כלומר מונה המבטיח כי לכל $\sigma_i < \sigma_j \in L$ מתקיים כי המופע הראשון של σ_i בסרט הפלט מתרחש לפני המופע הראשון של σ_j.

הסבר:
- אם $L \in R$ אז "הרעיון הטבעי אבל שגוי" שתיארנו עובד ולקסיקוגרפי.
- אם קיים מונה לקסיקוגרפי, נוכל להשתמש בו כדי לקבוע האם $\sigma_i \in L$ באופן הבא:
  - נריץ את המונה
  - אם מופיעה σ_i אז נעצור ונקבל
  - אם מופיעה σ_j עבור $j > i$ אז נעצור ונדחה

שאלה: מה אם L סופית? אז הכל טריוויאלי... ברור שיש מ"ט. יש אפילו אוטומט ☺

הגדרה:

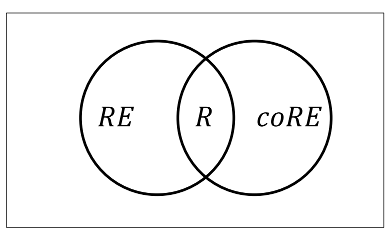
$$coRE = \{L \subseteq \Sigma^* : \overline{L} \in RE\}$$

כלומר $coRE$ זה אוסף כל השפות שהמשלים שלהם ב- RE. כלומר אוסף כל השפות L כך שקיימת מ"ט המקבלת את $\overline{L} = \Sigma^* \setminus L$.

שאלה: האם $coRE$ זה המשלים של RE, כלומר האם $coRE = \Sigma^* \setminus RE$?
תשובה: לא!! למשל $\emptyset, \Sigma^* \in RE \cap coRE$

טענה: $R = RE \cap coRE$

בציור:



<u>הוכחה:</u> כיוון ראשון: $\boldsymbol{R \subseteq RE \cap coRE}$

תהי $L \in R$. אזי גם $\overline{L} \in R$ (מדוע?)
כמו כן, $R \subseteq RE$. לכן $L, \overline{L} \in RE$. לכן גם $L, \overline{L} \in coRE$. לכן $L \in RE \cap coRE$.

כיוון שני: $\boldsymbol{R \supseteq RE \cap coRE}$

תהי $L \in RE \cap coRE$ ותהיינה בהתאם $M, \overline{M}$ מ"ט המקבלות את $L, \overline{L}$. נבנה מ"ט M' המכריעה את L באופן הבא: בהינתן קלט x, המ"ט M' תריץ במקביל את M ואת $\overline{M}$ על x. נקבל אם M מקבלת ונדחה אם $\overline{M}$ מקבלת.

מה הכוונה ב- "להריץ במקביל"?
כל פעם נבצע צעד אחד של M על x ואז נבצע צעד אחד של $\overline{M}$ על x. יש כמה דרכים לממש את זה, אולי הדרך הכי קלה היא לחשוב על מכונה עם 2 סרטים: על סרט אחד נריץ את M ועל השני את $\overline{M}$ וכך נבצע את הריצות שלהם במקביל.
הנקודה כאן היא שאנחנו לא יכולים "להריץ על הסוף את הראשונה" ורק אח"כ להריץ את השניה כי יכול להיות שהראשונה שנריץ דווקא לא עוצרת...
למה אחת מהן בהכרח עוצרת ומקבלת? אנחנו יודעים שהקלט הוא או בשפה (ואז M תקבל אותו) או שהוא לא בשפה (ואז $\overline{M}$ תקבל אותו)

מ"ט אוניברסליות וקידוד של מ"ט

עד עכשיו השתכנענו שלכל תוכנית תוכנית בפייתון (למשל) ניתן לבנות מכונת טיורינג שתסמלץ אותה "ותעשה את אותו דבר". זה טוב, אבל יש משהו שהאמירה הזאת עדיין לא תופסת: בחיים האמיתיים אנחנו מבדילים בין "המחשב" לבין "התוכנית" שרצה עליו, כאשר אותו מחשב מסוגל להריץ כל תוכנה שנרצה. מה האנלוג של זה במכונות טיורינג?

עכשיו נראה שאנחנו יכולים לבנות מכונת טיורינג אחת (נקרא לה "מכונה אוניברסלית") שתדע לקבל כקלט קידוד של מ"ט אחרת M וקלט x ולסמלץ את הריצה של M(x).

שימו לב: בהרצאות הקודמות ראינו כמה דוגמאות למקרים בהם בהינתן מכונה M בנינו מכונה M’ מתאימה אשר סימלצה את M. עכשיו אנחנו רוצים לבנות מכונה <u>אחת</u> U שתדע לסמלץ <u>כל</u> מכונה אחרת M (כאשר נותנים ל- U כקלט את הקידוד של M).

בשביל לפרמל את הדבר הזה נצטרך להסכים על דרך ספציפית לקודד תיאור של מכונות טיורינג.

הגדרה: <u>קידוד</u> (בינארי) של מ"ט הוא מיפוי חח"ע שלכל מ"ט M מתאים מחרוזת $\langle M \rangle \in \{0,1\}^*$
(כאן חח"ע היא עד כדי שינוי שמות איברי המצבים, אלפאבית הקלט, ואלפאבית העבודה)

סימון: עבור מ"ט M וקלט x עבורה נסמן ב- $\langle M, x \rangle$ קידוד בינארי של M ושל הקלט x

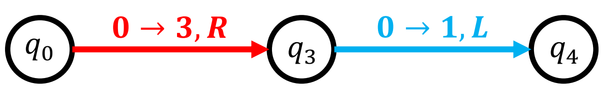
יש הרבה דרכים דיי טבעיות לקודד מכונות טיורינג, אבל זה לא ממש ישנה לצורכי הקורס שלנו ולכן לא נתעכב על זה יותר מדי. הדבר היחיד שנדרוש בהמשך הוא שניתן לקבוע ביעילות האם מחרוזת היא קידוד חוקי של מ"ט.

לשם פשטות נראה עכשיו קידוד "כמעט בינארי" שיהיה קל לתיאור.

תהי M מ"ט עם מצבים $Q=\{q_0,q_1,\dots,q_n\}$ כאשר בה"כ q_0,q_1,q_2 מצבים תחילי/מקבל/דוחה בהתאמה, ועם אלפאבית עבודה $\Gamma=\{\sqcup,0,1,2,\dots,m\}$ כאשר 0,1 הם תווי הקלט. לשם פשטות נניח כי $\Sigma=\{0,1\}$.

- נקודד מצב $q_k\in Q$ על ידי המחרוזת qw כאשר w הוא הייצוג הבינארי של k. למשל נייצג את q_3 על ידי $q011$.
- נקודד תו $k\in\Gamma$ על ידי המחרוזת aw כאשר w הוא הייצוג הבינארי של k.

כעת כדי לתאר/לקודד את המכונה M הנתונה, כל מה שאנחנו צריכים לעשות זה לתאר/לקודד את פונק' המעברים של המכונה. נוכל לרשום זאת ע"י רשימת חמישיות כאשר כל חמישייה מתארת את אחד המעברים של פונק' המעברים. למשל, אם המכונה שלנו מכילה את המעברים הבאים:



אז הקידוד שלנו עבור M יכיל את שתי החמישיות הבאות:

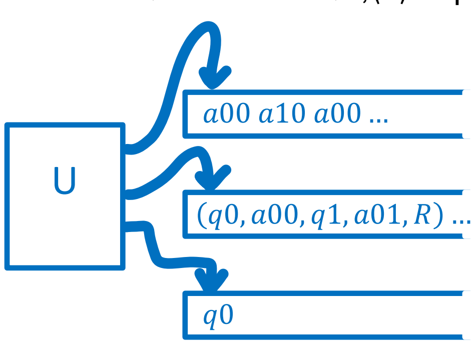
$$\langle M\rangle = (\underbrace{q000}_{\text{מצב נוכחי}},\underbrace{a00}_{\text{תו נוכחי}},\underbrace{q011}_{\text{המצב הבא}},\underbrace{a11}_{\text{התו הבא}},\underbrace{R}_{\text{כיוון}}),(q011,a00,q100,a01,L)$$

משפט: קיימת מ"ט U אשר בהינתן קידוד $\langle M,x\rangle$ של מ"ט וקלט, מקבלת/דוחה/לא-עוצרת אם $M(x)$ מקבלת/דוחה/לא-עוצרת.

הערה 1: בהינתן קלט שאינו קידוד חוקי, U דוחה.
הערה 2: מספר המצבים וגודל האלפאבית של U קבוע (קטן מ 100).

<u>אינטואיציה לבניית U:</u>

- למכונה יהיו 3 סרטים. בתחילת הריצה על סרט 1 מופיע קידוד של הקלט $\langle x\rangle$, על סרט 2 מופיע הקידוד של המ"ט אותה עלינו לסמלץ $\langle M\rangle$, ועל סרט 3 מופיע הקידוד של המצב ההתחלתי, כלומר מחרוזת מהצורה $q000$.
- במהלך הריצה, סרט 1 יסמלץ את הסרט של M בריצה על x (כולל מיקום הראש), וסרט 3 יחזיק את המצב הנוכחי של M.
- בכל שלב בסימולציה, נקרא את קידוד התו עליו מצביע הראש בסרט 1 ואת קידוד המצב הכתוב בסרט 3, ונחפש בסרט 2 חמישייה המתאימה לתו+מצב האלה. אם לא מצאנו אז נעצור ונדחה. אחרת נבצע את השינויים המתאימים בסרטים 1+2.



שאלה: איך בונים מ"ט שמבצעת חיפוש של מחרוזת x בתוך מחרוזת y?
תשובה: אפשר לעשות את זה בקלות ע"י מ"ט עם שני סרטים כאשר על סרט 1 נכתוב את y ועל הסרט השני את x. בתחילת הריצה שני הראשים בהתחלה. בכל "פאזה" של הריצה נתקדם במקביל ימינה לאורך x ו- y ונשווה את התווים. אם הצלחנו אז נסיים. אם y נגמר אז נדחה. אחרת נרוץ חזרה שמאלה עד סוף x (ובמקביל נרוץ שמאלה גם עם y, אבל נעשה צעד אחד פחות שמאלה, כדי שבפאזה הבאה נתחיל נתקדם תו אחד בתוך y)

האם יש בעיות שתוכניות מחשב לא יכולות לפתור?

טענה: קיימת $L \in \{0,1\}^*$ כך ש- $L \notin RE \cup coRE$. בפרט L אינה כריעה (כלומר אינה ב R).

ראינו משהו דומה לזה בהקשר של מעגלים. זה נובע מכך שיש המון שפות ומעט תוכניות.

הוכחה: שיקולי ספירה:

- מהי עוצמת אוסף כל השפות הבינאריות? יש $\aleph_0$ מחרוזות בינאריות ומספר תתי הקבוצות שלהם הוא $\aleph = 2^{\aleph_0}$. כלומר
$$|\{L : L \subseteq \{0,1\}^*\}| = \left|2^{\{0,1\}^*}\right| = \aleph$$
- מה לעומת זאת עוצמת כל השפות ב- $RE \cup coRE$? כל שפה כזאת יש מ"ט שמקבלת אותה או את המשלים שלה. לכן

$$|RE \cup coRE| = |\{L : L \in RE \cup coRE\}| \lesssim \left|\{M : \text{מ"ט } M\}\right| \leq \left|\{\langle M\rangle \in \{0,1\}^* \text{ קידוד מ"ט}\}\right| = \aleph_0$$

מ.ש.ל.

ההוכחה הנ"ל "לא קונסטרוקטיבית" – לא נותנת בעיה מפורשת שלא ניתן להכריע, אלא רק מראה קיום. כעת נראה שישנן בעיות מפורשות (ובמידה מסוימת טבעיות) שאינן ניתנות להכרעה. בפרט, בהינתן תכנית, לא בהכרח ניתן לקבוע כיצד היא מתנהגת על קלט נתון.

הגדרה:
$$ACC = \left\{\langle M,x\rangle : \begin{array}{c} \text{קידוד של } \langle M,x\rangle \\ \text{מ"ט } M \text{ וקלט } x \\ \text{כך ש } M(x) \text{ מקבלת} \end{array}\right\}$$

לעיתים נקרא לבעיית ההכרעה של ACC "בעיית הקבלה"

טענה: $ACC \in RE$

הוכחה: מתקיים $ACC = L(U)$ עבור המכונה האוניברסלית U. מ.ש.ל.

האם $L \in R$? עבור $\langle M, x\rangle$ שרירותיים אפילו לא ברור איך לקבוע אם M עוצרת על x..

דוגמה: האם המ"ט M הבאה מקבלת את הקלט הריק?

$M(x)$: מתעלמת מהקלט x. עבור $i = 1,2,3, \ldots$ בצע: אם <u>לא</u> קיימים ראשוניים p_1, p_2 עבורם $2i = p_1 + p_2$ אז עצור וקבל. אחרת המשך ל $i + 1$.

לא ידוע האם המכונה הזאת עוצרת. למעשה זאת בעיה פתוחה במתמטיקה (מ 1742) האם כל מספר זוגי אפשר להציג כסכום של שני ראשוניים (נקראת השערת גולדבך). אם היינו יכולים להכריע ביעילות את ACC זה היה מאפשר לנו לבדוק את השערת גולדבך. כעת נוכיח שאלג' כזה למעשה לא קיים.

טענה: $ACC \notin R$

<u>הוכחה:</u>

נגדיר את השפה
$$F = \{\langle M\rangle : \langle M\rangle \notin L(M)\}$$

כלומר זאת שפת כל קידודי המ"ט אשר אינן מקבלות את הקידוד של עצמן

<u>טענה א:</u> לא קיימת מ"ט M כך ש- $L(M) = F$. כלומר לא קיימת מ"ט המקבלת את F.
כלומר $F \notin RE$.

<u>מדוע?</u>
נניח בשלילה שקיימת מ"ט M המקבלת את השפה F, כלומר $L(M) = F$.

מה קורה כאשר מריצים את המ"ט M על הקידוד של עצמה (כלומר על $\langle M\rangle$) ?

- אם היא לא מקבלת, אזי $\langle M\rangle \notin L(M)$, מה שלפי הגדרת F אומר ש $\langle M\rangle \in F$. כלומר $\langle M\rangle \in F \setminus L(M)$ ולכן $L(M) \neq F$.
- אם היא כן מקבלת, אזי $\langle M\rangle \in L(M)$, מה שלפי הגדרת F אומר ש $\langle M\rangle \notin F$. כלומר $\langle M\rangle \in L(M) \setminus F$ ולכן $L(M) \neq F$.

כלומר בכל מקרה $L(M) \neq F$.
סתירה.

אז F היא שפה קשה במובן הזה שלא רק שאי אפשר להכריע אותה, אלא אפילו אי אפשר לקבל אותה.
עכשיו נראה שאם אפשר להכריע את ACC אז אפשר להכריע את F מה לא יתכן.

<u>טענה ב:</u> אם קיימת מ"ט M_A המכריעה את ACC אז קיימת מ"ט M_F המכריעה את F (מה שלא יתכן...)

<u>מדוע?</u> נבנה M_F אשר בהינתן קלט $\langle M\rangle$ מריצה את $M_A(\langle M, \langle M\rangle\rangle)$ ועונה ההפך ממנה. מתקיים:

$M_F(\langle M\rangle)$ דוחה

אם"ם

$M_A(\langle M,\langle M\rangle\rangle)$ מקבלת

אם"ם

$M(\langle M\rangle)$ מקבלת

אם"ם

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

מ.ש.ל.

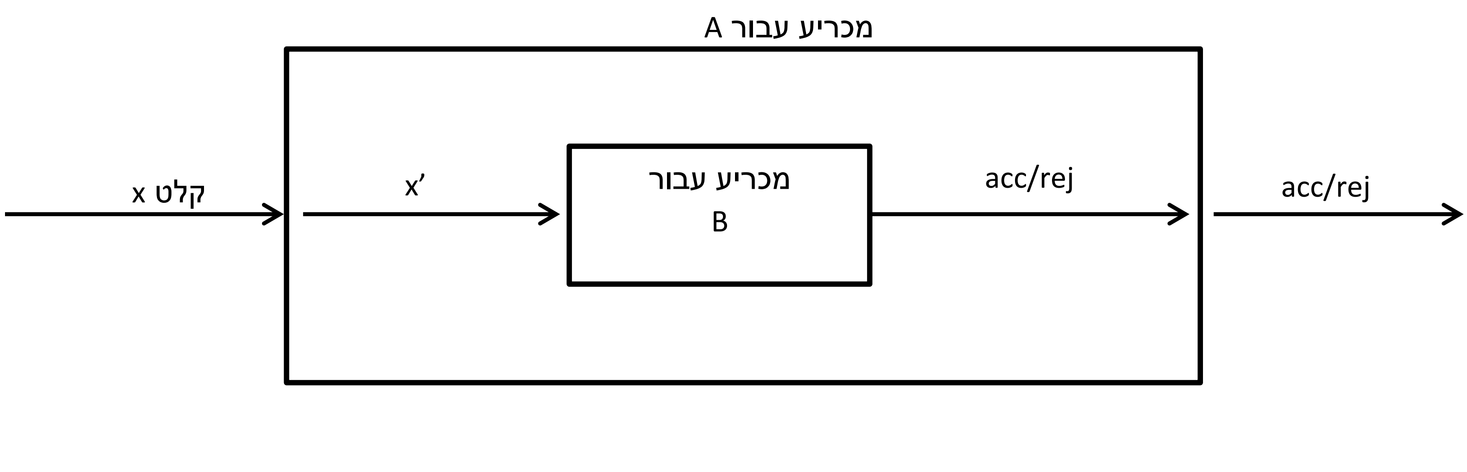
שאלה: איפה ההוכחה של טענה ב הנ"ל נשברת אם M_A רק <u>מקבלת</u> את ACC ולא <u>מכריעה</u> אותה?
תשובה: האם"ם הראשון לא יהיה נכון יותר מכיוון שאם M_A לא עוצרת אז גם M_F לא תעצור...

טכניקת ההוכחה הזאת של "להריץ קוד על עצמו" נקראת לפעמים <u>לכסון</u> כי אפשר להציג אותה בצורה חצי-גרפית ע"י טבלה והסתכלות על האלכסון של הטבלה הזאת (מזכיר את שיטת ההוכחה של קנטור לכך ש $\aleph_0 \neq \aleph$)

בעצם מה שהראינו כאן בטענה ב זה <u>טיעון רדוקציה</u>: הראינו רדוקציה מבעיית ההכרעה של F לבעיית ההכרעה של ACC

הגדרה (חצי פורמלית): קיימת <u>רדוקציה</u> מבעיית הכרעה A לבעיית הכרעה B אם ניתן להשתמש בכל מכריע עבור B על מנת להכריע את A.

<u>סימון</u>: $A \leq B$. פרשנות: "B לא פחות קשה להכרעה מ- A". בפרט, זה גורר שאם A לא ניתנת להכרעה אז גם B לא ניתנת להכרעה.



<u>הערה בנוגע לאי-יוניפורמיות:</u>

ראינו כי ACC לא ניתנת להכרעה ע"י מ"ט. האם ACC ניתנת להכרעה ע"י משפחת מעגלים?

כלומר האם קיימת משפחה אינסופית של מעגלים (מעגל לכל אורך קלט) אשר בהינתן קידוד $\langle M,x\rangle$ כך ש- M מקבלת את x המעגל מחזיר 1 (="כן") ואם M לא מקבלת את x (דוחה, או לא-עוצרת, או קידוד לא תקין) אז המעגל מחזיר 0 (="לא")

כן! ראינו שלכל שפה יש משפחת מעגלים שמכריעה אותה.

בפרט מעגלים יכולים לפתור בעיות לא כריעות.

דוגמה לשפה נוספת:

הגדרה: $$HALT=\left\{\langle M,x\rangle : \begin{array}{c}\text{קידוד של } \langle M,x\rangle \\ \text{מ"ט } M \text{ וקלט } x \\ \text{כך ש } M(x) \text{ עוצרת}\end{array}\right\}$$

כלומר זאת הגדרה דומה ל- ACC רק שהחלפנו את הדרישה ש $M(x)$ מקבלת בדרישה שהיא עוצרת

טענה: $HALT\in RE\setminus R$

<u>הוכחה:</u>

את העובדה כי $HALT\in RE$ אפשר להראות בעזרת המ"ט האוניברסלית, בדומה למה שעשינו עבור ACC (בדקו!).

נראה כי $HALT\notin R$ ע"י כך שנראה רדוקציה $ACC\le HALT$

תהי H מ"ט המכריעה את $HALT$. נשתמש ב- H על מנת לתאר מ"ט A המכריעה את ACC, בסתירה לכך ש- $ACC\notin R$

<u>A בהינתן קלט $\langle M,x\rangle$:</u> - תריץ $H(\langle M,x\rangle)$. אם H דוחה אז נדחה - תריץ $M(x)$ עד לעצירה ותקבל אם"ם M מקבלת

מתקיים:

- אם $\langle M,x\rangle\in ACC$ אז בפרט $\langle M,x\rangle\in HALT$ ולכן A תגיע לשורה השניה ותקבל
- אם $\langle M,x\rangle\notin ACC$ נשקול שלוש אפשרויות:
  1. $\langle M,x\rangle$ לא קידוד חוקי. במקרה זה נדחה בשורה הראשונה
  2. $M(x)$ דוחה. במקרה זה נגיע לשורה השניה ונדחה
  3. $M(x)$ אינה עוצרת. במקרה זה נדחה בשורה הראשונה.

דוגמה נוספת:

הגדרה:

$$EMPTY = \left\{ \langle M \rangle : \begin{matrix} \text{קידוד של מ"ט } M \ \langle M \rangle \\ L(M) = \emptyset \text{ כך ש} \end{matrix} \right\}$$

טענה: $EMPTY \notin R$

נקודה למחשבה: לבנות מ"ט M כך ש- $L(M) = \emptyset$ זה קל לנו מאוד. אבל הטענה הזאת אומרת שבהינתן $\langle M \rangle$ קשה לנו לומר האם $L(M) = \emptyset$. כמה קשה? בשיעור הבא נראה שגם $EMPTY \notin RE$.
הערה: כן מתקיים ש- $EMPTY \in coRE$. נסו לחשוב מדוע. פתרון:
$EMPTY \in coRE$ אם"ם $\overline{EMPTY} \in RE$ $\approx \{\langle M \rangle : L(M) \neq \emptyset\}$. כלומר זאת שפת כל הקידודים $\langle M \rangle$ כך שיש משהו ש M מקבלת (שווה בערך כי צריך להתייחס גם לקידודים לא-חוקיים). נוכל לקבל את $\overline{EMPTY}$ בצורה דומה לבניית המונה: לכל פאזה i נריץ למשך i צעדים כ"א מ- $M(\sigma_1), \dots, M(\sigma_i)$. אם מתישהו M מקבלת אז נקבל.

<u>הוכחה: נראה רדורציה $ACC \leq EMPTY$:</u>
ברדוקציה שתיכף נראה, ניעזר בהגדרה הבאה:
לכל מ"ט M וקלט עבורה $x \in \Sigma^*$ נגדיר מ"ט M_x באופן הבא:

> M_x בהינתן קלט $y \in \Sigma^*$:
> ----------------------------------
> - נתעלם מ- y
> - נריץ את $M(x)$ ונקבל אם"ם M מקבלת

נשים לב שמתקיים:
((א)) אם $L(M)$ $x \in$ אז $L(M_x) = \Sigma^* \neq \emptyset$
((ב)) אם $L(M)$ $x \notin$ אז $L(M_x) = \emptyset$

כעת, בהינתן מכריע E עבור $EMPTY$ נבנה מכריע A עבור ACC באופן הבא:

> A בהינתן קלט $\langle M, x \rangle$:
> ----------------------------------
> - אם $\langle M, x \rangle$ לא קידוד חוקי, אז נדחה
> - נחשב את הקידוד $\langle M_x \rangle$ ונריץ $E(\langle M_x \rangle)$
> - אם E מקבלת אז נדחה ואם E דוחה אז נקבל.

מתקיים:
- אם $\langle M, x \rangle \in ACC$ אז $x \in L(M)$ ולכן לפי ((א)) $L(M_x) \neq \emptyset$ ולכן E דוחה ו- A יקבל
- אם $\langle M, x \rangle \notin ACC$ אז $x \notin L(M)$ ולכן לפי ((ב)) $L(M_x) = \emptyset$ ולכן E מקבלת ו- A דוחה

הערה חשובה: A הנ"ל לא מריץ בעצמו את המ"ט M שהוא מקבל כקלט. הוא רק "מתקן" את הקידוד שלה ומריץ את E על הקידוד המתוקן. אם A היה מריץ בעצמו את M זה היה יוצר לנו בעיה שאם M לא עוצרת אז גם A לא עוצר...