מרתון חישוביות - 2.12
המודל: מכונת טיורינג.
כמו אוטומט עם מצבים וסרט זיכרון אינסופי לצד ימין. (ניתן להתייחס לזה כאל מערך המתחיל בתא 0 בצד שמאל וממשיך עד אינסוף) ובכל רגע נתון יש ראש קורא/כותב מצביע על תא אחד מתוך הזיכרון (בהתחלה הוא מצביע על תא 0)
בכל צעד חישוב, המכונה קוראת את התו שעליו הראש קורא/כותב מצביע ובהתאם לתו ולמצב הנועחי שלה, היא עוברת למצב חדש, כותבת תו אחר (או אותו תו) במקום התו שאותו קראה וזזה עם הראש צעד אחד ימינה או שמאלה.

שימו לב: בשונה מאוטומטים, לא קוראים את המילה בהכרח תו אחרי תו. כמו כן, לאחר סיום קריאת המילה המכונה לא עוצרת.
הדרך היחידה שהמכונה תעצור היא אם היא תיכנס למצב עצירה - מצב שהוא כמו return של פונקציה.

פורמאלית: $M = (Q, \Sigma, \Gamma, q_0, Ƀ, \delta, F)$
Q - קבוצת מצבים סופית.
Σ - א"ב שממנו הקלט מורכב - סופי.
Γ - התווים שניתן לכתוב בסרט הזיכרון (כולל את Σ ואת Ƀ) - סופי.
q_0 - מצב התחלתי.
Ƀ - תו ריק שכתוב בכל המקומות בסרט אחרי תווי הקלט.
δ - המעברים - מקבלת מצב ותו שאותו הראש קורא - עוברת למצב חדש, כותבת תו אחר במקום וזזה עם הראש ימינה או שמאלה.
F - מצבי עצירה.

מכונת טיורינג היא בכוח חישוב של מחשב.
נתייחס תמיד למ"ט כמו פונקציה אחת שמישהו כתב.

אם המכונה לא מגיעה למצבי עצירה אז היא לא עוצרת.

קונפיגורציה: תיאור מצב נוכחי של המכונה: (q, i, w) כאשר:
q - מצב נוכחי.
i - מיקום הראש בזיכרון.
w - תוכן כל הזיכרון (ללא הזנב האינסופי של התווים הריקים)

מכיוון שתמיד ניתן לרוץ ימינה ולכתוב עוד ועוד תווים אז כמות הקונפיגורציות השונות היא אינסופית.
אם המכונה חוזרת על קונפיגורציה שכבר הייתה בה - היא תקועה בלולאה אינסופית.

המסקנה: אם מגבילים את המכונה לא לעבור תא כלשהו אז ניתן לזהות מתי יש לולאה אינסופית.

שקילות מודלים:
- מ"ט רגילה (כמו שתיארנו כרגע) שקולה למספר מודלים אחרים:
  1. מ"ט עם מספר סרטים סופי.
  2. מ"ט א"ד: מכונה שלכל צעד חישוב יש מספר אפשרויות והמכונה בוחרת באיזה מסלול חישוב ללכת.

שפה של מכונה:

מכונה שיש לה 2 מצבי עצירה: q_{accept}, q_{reject} מקבלת חלק מהמילים ודוחה חלק מהמילים ואולי יש מילים שהיא לא עוצרת עליהן - לא מקבלת ולא דוחה.

$L(M)$ - היא שפה המכילה את כל המילים שהמכונה מקבלת.

כלומר, מה שהמכונה דוחה או לא עוצרת לא נמצא בשפה של המכונה.

דוגמא: עבור המכונה הבאה:

M: על קלט x:

- אם $|x|$ זוגי - קבל.
- אחרת, אם $x = 000$ - קבל.
- אחרת - דחה.

אזי: $L(M) = \{x: |x| \text{ זוגי}\} \cup \{000\}$.

פונקציה של מכונה:

מכונה שיש לה מצב עצירה: q_{halt}.

כאשר המכונה מגיעה למצב עצירה - הפלט שלה הוא כל מה שכתוב בסרט הזיכרון ממיקום 0 עד מיקום הראש (לא כולל)

$f_M(x)$ - היא התוצאה של המכונה על הקלט x.

אם המכונה לא עוצרת על x אז הפלט: $f_M(x)$ לא מוגדר.

פונקציה שלא מוגדרת על כל הקלטים נקראת **פונקציה לא מלאה.**

פונקציה שמחזירה תוצאה לכל $x \in \Sigma^*$ נקראת **פונקציה מלאה.**

דוגמא: עבור המכונה הבאה:

M: על קלט x:

- צעד צעד אחד ימינה.
- עצור.

אזי: $f_M(x)$ מחזירה את התו השמאלי ביותר של x.

שפות כריעות:

שפה L כריעה היא שפה שיש מ"ט <u>שמקבלת</u> כל $x \in L$ <u>ודוחה</u> כל $x \notin L$. (בפרט עוצרת על כל קלט)

במילים אחרות - המכונה M מכריעה את L ומקיימת: $L(M) = L$.

מכונה **מכריעה** היא מכונה שעוצרת על כל קלט.

קבוצת כל השפות הכריעות: R.

- כל השפות הסופיות.
- כל השפות הרגולריות וחסרות ההקשר.
- כל שפה שניתן לבנות לה תוכנית מחשב (פונקציה בוליאנית) שתמיד עוצרת.

סגירות של השפות הכריעות:
השפות הכריעות סגורות ל: איחוד, חיתוך, הפרש, משלים, שרשור, חזקה, היפוך, איטרציה.

כמובן שאין סגירות להכלה. כלומר אם שפה אחת מוכלת בשנייה ויודעים משהו על אחת מהן, זה לא אומר כלום על השנייה.

שפות מזוהות טיורינג:
שפה L מזוהה טיורינג היא שפה שיש מ"ט שמקבלת כל $x \in L$ **ולא מקבלת** (דוחה או לא עוצרת) כל $x \notin L$.
במילים אחרות - המכונה M מזהה את L ומקיימת: $L(M) = L$.

מכונה **מזהה** היא מכונה שעוצרת על כל קלט שהוא כן בשפה, ועבור קלט שהוא לא בשפה היא עלולה לא לעצור.

קבוצת כל השפות המזוהות: RE.
- מכילה את R.
- שפות נוספות ש**קל לוודא** שקלט אכן נמצא בשפה.

סגירות של השפות המזוהות טיורינג:
השפות המזוהות סגורות ל: איחוד, חיתוך, שרשור, איטרציה, היפוך, חזקה אבל לא **משלים (ולכן גם לא הפרש)**.

השפות המשלימות למזוהות טיורינג:
שפה L תהיה משלימה של מזוהה טיורינג אם יש מ"ט שלא דוחה (מקבלת או לא עוצרת) כל $x \in L$ ודוחה כל $x \notin L$.

הערה: ישנן שפות L משלימות למזוהות טיורינג שאין להן מ"ט M כך ש: $L(M) = L$.

קבוצת כל השפות המשלימות למזוהות: $coRE$.

כדי להוכיח ששפה שייכת ל $coRE$ עדיף להוכיח שהמשלימה שלה שייכת ל RE.

תכונות של $R, RE, coRE$:
- $R \subseteq RE$ וגם $R \subseteq coRE$.
- $R = RE \cap coRE$. כלומר אם יש שפה שהיא גם ב RE וגם ב $coRE$ אז היא ב R.
- אם $L \in RE$ אז $\bar{L} \in coRE$.

פונקציות ניתנות לחישוב
פונקציה שניתנת לחישוב היא פונקציה **שקיימת** מ"ט שמחזירה בדיוק את התוצאות של הפונקציה לכל קלט.
אם לא קיימת מ"ט כזו אז הפונקציה **לא ניתנת לחישוב.**

דוגמא:
הפונקציה: $f(x)$ - מחזירה את ......$x \cdot x \cdot x \cdot$ אינסוף פעמים.
הפונקציה היא מלאה (לכל x היא מחזירה תוצאה - את עצמו משורשר אינסוף פעמים - סדרה אינסופית) אבל לא ניתנת לחישוב.
קיימות גם פונקציות שלא מחזירות משהו אינסופי ועדיין לא ניתנות לחישוב.

הוכחת קיום של שפות שלא ניתנות לחישוב באמצעות עוצמות:

כמות השפות השונות בעולם היא: $\aleph = |P(\Sigma^*)|$.

כמות מ"ט השונות בעולם היא: $\aleph_0 = |\Sigma^*|$.

לכל מ"ט מותאמת שפה אחת ויחידה ולכן יש יותר שפות ממ"ט ומכאן בהכרח שיש שפות שאין להן מ"ט בכלל.

עוד עוצמות:

$|R| = \aleph_0$

$|RE| = \aleph_0$

$|coRE| = |RE| = \aleph_0$

לכן רוב השפות הן לא ב R, לא ב RE וגם לא ב coRE.

הסיבה ש $\aleph_0 = |\Sigma^*|$ הוא כי ניתן להגדיר סדר לקסיקוגרפי (מילוני) על כלל המילים בעולם:

0: ε.

1: a

2: b

3: c

....

27: aa

28: ab

.....

ניתן להתייחס לקלט של מ"ט כקלט בינארי: $\{0, 1\}$ כי תמיד ניתן לקודד כל דבר אחר לבינארי.

מספר: לפי הייצוג הבינארי.

תו: לפי הערך ה ASCII

מחרוזת: מערך של ערכים ASCII.

קידוד אובייקטים:

לכל אובייקט יש קידוד מוסכם בעולם (אין צורך לדעת אותו במסגרת הקורס אלא רק להבין שהוא קיים)

סימון: הקידוד של אובייקט A מסומן ב: $< A >$.

כאשר מדברים על שפה של אובייקטים, מתייחסים גם לקידודים לא תקינים שבדרך כלל מייצגים כולם אובייקט ריק (לכל אובייקט יש את ברירת המחדל שלו)

גם למ"ט M יש קידוד: $< M >$.

בהרה מקומות, קידוד לא תקין של מ"ט מייצג את ברירת המחדל - מ"ט שלא עוצרת על אף קלט.

כלומר,מעתה נדבר על שפות כאוסף של מחרוזות המייצגות קידודים ולא סתם מילים.

לדוגמא:

$L = \{< M >: M\ is\ TM\ and\ M\ has\ exactly\ 3\ states\}$.

הערה: $L \in R$. ניתן לעבור על הקידוד ולבדוק זאת מתוכו.

תרגיל:

תהי השפה הבאה:

$L = \{< M, x >: M\ accepts\ x\ in\ at\ most\ 10\ steps\}$.

הוכיחו ש $L \in R$.

פתרון: נראה מ"ט M_L: על קלט $< M, x >$ כאשר M - מ"ט, x - מילה:

- סמלץ את ריצת M על x למשך 10 צעדים.
- אם במהלך 10 הצעדים, M קיבלה את x - קבל.
- אחרת - דחה.

M_L תמיד עוצרת.
נכונות:
אם $< M, x >\in L$ אז M מקבלת את x תוך 10 צעדים ולכן הבדיקה של הסמלוץ תסתיים בקבלה ולכן נקבל בשורה 2.
אם $< M, x >\notin L$ אז M לא מקבלת את x תוך 10 צעדים ולכן נדחה.

שפות לא כריעות:
שפות השייכות ל RE ולא ל R ולכן גם לא ל coRE.
$L_u = A_{TM} = \{< M, x >: M\ is\ TM\ and\ M\ accepts\ x\} \in RE\backslash R$.
$HP = \{< M, x >: M\ is\ TM\ and\ M\ halts\ on\ x\} \in RE\backslash R$.

שפות השייכות ל coRE ולא ל R ולכן גם לא ל RE.
$\bar{L_u} \in coRE\backslash R$.
$\bar{HP} \in coRE\backslash R$.

שפות שלא שייכות לאף מחלקה מהמחלקות לעיל:
$L_\infty = \{< M >: M\ is\ TM\ and\ |L(M)| = \infty\} \notin RE$.
$L_\infty \notin coRE$.

אינטואיציה מתי שפה שייכת ל R,RE,coRE:
שואלים:
אם נניח שהקלט כן בשפה (מקיים את התנאים), כיצד ניתן לוודא זאת (להוכיח לעצמנו)? לא צריך משהו חכם - לרוב זה משהו נאיבי - לא דורשים יעילות.

- אם צריך להריץ מ"ט כלשהי ללא נתונים הרצה מלאה - החשש הוא שלא נעצור ולכן לא נוכל לוודא.
- אם צריך לעבור על כל המילים בעולם - לא ניתן לעשות זאת כי יש אינסוף מילים.

אם הצלחנו לוודא כאשר הנחנו שהקלט כן בשפה אז השפה שייכת ל RE.

לאחר מכן נשאל:
אם נניח שהקלט לא בשפה (לא מקיים את התנאים, כיצד ניתן לוודא זאת?
אם הצלחנו לוודא אז השפה ב coRE.

אם הצלחנו לוודא ב 2 המקרים - השפה ב R.

ישנם מקרים חריגים שבהם התנאי הוא טריוויאלי - תמיד מתקיים או תמיד לא מתקיים - במקרה כזה השפה שייכת ל R.
ישנם מקרים חריגים שבהם השפה מראש היא סופית, גם אם התנאי מורכב - גם במקרה כזה, השפה ב R.

תרגילים: עבור כל אחת מהשפות, קבעו היכן היא נמצאת:
$L_1 = \{< M_1, M_2 >: \exists x \in L(M_1) \cap L(M_2)\}$.

מקבלים זוג מכונות וצריכים לבדוק האם יש מילה בחיתוך של השפות של 2 המכונות- כלומר ש 2 המכונות מקבלות אותה.

אם נתון לנו שהקלט בשפה אז נוכל לעבור על כל המילים בסדר לקסיקוגרפי (ככה לא נפספס אף מילה ונגיע בפרט למילה המובטחת) ועל כל מילה נריץ את 2 המכונות כמות צעדים שהולכת וגדלה (תמיד מבוקרת).
כלומר: בשלב ה i נריץ את 2 המכונות על כל אחת מ i המילים הראשונות בסדר לקסיקוגרפי למשך i צעדים.

מ"ט מזהה:על קלט $< M_1, M_2 >$:

- עבור i מ 1 עד אינסוף בצע:
  - עבור כל מילה w מתוך i המילים הראשונות בסדר לקסיקוגרפי של Σ^*:
    - הרץ את M_1 על w למשך i צעדים.
    - הרץ את M_2 על w למשך i צעדים.
    - אם שתיהן עצרו וקיבלו בזמן המוקצב - קבל.

נכונות:
אם $< M_1, M_2 >\in L_1$ אז קיימת מילה שמתקבלת ב 2 המכונות ולכן בהרצה המבוקרת אנו נמצא בין היתר את המילה הזו עבור i מספיק גדול ונריץ את 2 המכונות מספיק זמן עליה עבור i מספיק גדול כך ששתיהן יעצרו ויקבלו בזמן המוקצב.
אם $< M_1, M_2 >\notin L_1$ אזי לא קיימת מילה שמתקבלת ב 2 המכונות ולכן לא יהיה מצב שבו 2 המכונות עוצרות ומקבלות את אותה המילה ולכן נמשיך לרוץ בלולאה אינסופית ולא נקבל.

לכן $L_1 \in RE$.
כאשר הקלט לא בשפה אזי צריך לעבור על כל המילים בעולם, ולהריץ מלא את 2 המכונות כדי לראות האם הן דוחות וייתכן גם שלא עוצרות ולכן: $L_1 \notin coRE$ ולכן: $L_1 \notin R$.

$L_2 = \{< M > |\forall x: x \in L(M) \rightarrow x^R \in L(M)\}$.
אינטואיציה: כדי לוודא שאכן התנאי מתקיים צריך לעבור על כל המילים בעולם, לבדוק אחת אחת האם היא מתקבלת ואם כן אז צריך גם לבדוק שההפוכה שלה מתקבלת. לכן: $L_2 \notin RE$.

אינטואיציה: כדי לוודא שיש מילה שמתקבלת במכונה אבל ההפוכה שלה לא מתקבלת במכונה צריך למצוא את המילה (זה ניתן לעשות ע"י הרצה מבוקרת על המילים בסדר לקסיקוגרפי) אבל לא יהיה ניתן לוודא שהמילה ההפוכה לא מתקבלת כיוון שייתכן והמכונה לא עוצרת עליה ואז נצטרך לחכות לנצח. לכן:
$L_2 \notin coRE$.

כאשר רוצים להוכיח שייכות ל R,RE,coRE פשוט מראים מ"ט מתאימה.

שיטה להוכחת אי שייכות - רדוקציה
הרעיון של רדוקציה: שימוש בפתרון קיים לטובת בעיה אחרת. או שימוש בפתרון לבעיה קשה יותר כדי לפתור בעיה קלה יותר.

סימון: $L_1 \le L_2$ -

L_1 ניתנת לרדוקציה ל L_2.

כלומר L_2 קשה יותר ולכן אם אנו יודעים לפתור את L_2 אז כל שכן שנדע לפתור את L_1 וגם להיפך אם לא יודעים לפתור את L_1 אז כל שכן שלא נדע לפתור את L_2.

הרדוקציה עצמה היא פונקציה מ Σ^* ל Σ^* המקבלת קלט של L_1 ומחזירה קלט של L_2 כך ש:

הרדוקציה מלאה - כלומר היא מוגדרת על כל קלט.

הרדוקציה ניתנת לחישוב - יש מ"ט שמחשבת את הפונקציה הזו. (המרת קלט לקלט בלי לפתור את הבעיות)

הרדוקציה מקיימת תקפות - אם $x \in L_1$ אז $f(x) \in L_2$ ואם $x \notin L_1$ אז $f(x) \notin L_2$. כלומר הפונקציה צריכה לשלוח מילה בשפה הראשונה למילה בשפה השנייה ולשלוח מילה שאינה בשפה הראשונה למילה שאינה בשפה השנייה.

הערה: הרדוקציה אינה פונקציה חח"ע ועל (כמעט תמיד היא לא חח"ע ועל)

תכונות של רדוקציה:

1. $L \le L$.
2. אם $L_1 \le L_2$ וגם $L_2 \le L_3$ אז $L_1 \le L_3$.
3. אם $L_1 \le L_2$ אז $\bar{L_1} \le \bar{L_2}$.
4. משפט הרדוקציה: אם $L_1 \le L_2$ אז:
   a. אם $L_1 \notin R$ אז $L_2 \notin R$. (בכיוון ההפוך: אם $L_2 \in R$ אז $L_1 \in R$)
   b. אם $L_1 \notin RE$ אז $L_2 \notin RE$. (בכיוון ההפוך: אם $L_2 \in RE$ אז $L_1 \in RE$)
   c. אם $L_1 \notin coRE$ אז $L_2 \notin coRE$. (בכיוון ההפוך: אם $L_2 \in coRE$ אז $L_1 \in coRE$)

תרגילים: הוכיחו אי שייכות עבור השפות הבאות:

1. $L_1 = \{< M >: |L(M)| > 3\}$.

פתרון: $L_1 \in RE$.

מ"ט מזהה: M_1: על קלט $< M >$:

- עבור i מ 1 עד אינסוף בצע:
  - $c = 0$
  - עבור כל w מ i המילים הראשונות בסדר לקסיקוגרפי של Σ^*:
    - הרץ את M על w למשך i צעדים, אם קיבלה: c ++.
    - אם $c = 4$ - קבל.

נכונות: ...

$L_1 \notin R$. נראה רדוקציה: $HP \le L_1$. ע"י: $f(< M, x >) =< M' >$ כאשר:

M': על קלט w:

- הרץ את M על x. // אם M לא עצרה אז אנו לא מקבלים שום דבר
- קבל. // מקבלים הכל

הרדוקציה מלאה וניתנת לחישוב - כתיבת קידוד של מכונה חוקית שמריצה מכונה על מילה ומקבלת.
הרדוקציה תקפה -

אם $< M, x >\in HP$ אז M עוצרת על x ולכן M' לכל קלט w תקבל ולכן: $L(M') = \Sigma^*$ ובפרט היא מקבלת יותר מ 3 מילים. ולכן: $< M' >\in L_1$.

אם $< M, x >\notin HP$ אז M לא עוצרת על x ולכן M' לכל קלט w נתקעת בשורה הראשונה ולכן: $L(M') = \Phi$ ובפרט היא לא מקבלת יותר מ 3 מילים (היא לא מקבלת אף מילה). ולכן: $< M' >\notin L_1$.

2. $L_2 = \{< M, n > | \exists x_1, x_2,..., x_n\ s.t.\ x_i\ is\ proper\ prefix\ of\ x_{i+1}\ and\ x_i \in L(M)\}$.

מקבלים מכונה M ומספר n ובודקים האם ישנן n מילים שאחת היא רישא ממש של הבאה בתור וכולן מתקבלות במכונה.

המחשה לרישא ממש:

$x_1 = 010$

$x_2 = 0101101$

$x_3 = 010110111$

$L_2 \notin R$. נראה רדוקציה: $HP \leq L_2$ ע"י: $f(< M, x >) =< M', 2 >$ כאשר:

M': על קלט w:

- הרץ את M על x. // אם M לא עצרה אז אנו לא מקבלים שום דבר
- קבל. // מקבלים הכל

הרדוקציה מלאה וניתנת לחישוב - כתיבת קידוד של מכונה חוקית שמריצה מכונה על מילה ומקבלת.
הרדוקציה תקפה -

אם $< M, x >\in HP$ אז M עוצרת על x ולכן M' לכל קלט w תקבל ולכן: $L(M') = \Sigma^*$ ובפרט היא מקבלת את $0,\ 00$.

אם $< M, x >\notin HP$ אז M לא עוצרת על x ולכן M' לכל קלט w נתקעת בשורה הראשונה ולכן: $L(M') = \Phi$ ובפרט אין 2 מילים שמתקבלות במכונה.

3. $L_3 = \{< M >: L(M)\ contains\ infinite\ words\ of\ odd\ length\}$.

מקבלים כקלט מכונה ובודקים האם היא מקבלת אינסוף מילים באורך אי זוגי.

פתרון:

נראה רדוקציה: $L_\infty \leq L_3$ ע"י: $f(< M >) =< M' >$ כאשר:

M': על קלט w:

- אם w באורך זוגי - דחה.
- אם w באורך אי זוגי: סמן: $w = \sigma x$.
- הרץ את M על x ועל w במקביל (צעד כאן וצעד כאן) וקבל אם היא קיבלה את לפחות אחת מהן. ואם היא דחתה את שתיהן - דחה.

ε

$0, 1$

$00, 01, 10, 11$

$000, 001,...., 111$

נראה רדוקציה: $\bar{HP} \leq L_3$ ע"י: $f(< M, x >) =< M' >$ כאשר:

M': על קלט w:

- הרץ את M על x למשך $|w|$ צעדים.
- אם בזמן המוקצב M עצרה - דחה.
- אחרת - קבל.

משפט רייס - שיטה נוספת להוכיח ששפה אינה ב R,RE

אם S היא תכונה סמנטית לא טריוויאלית של שפות אזי: $L_S = \{< M >: L(M) \in S\} \notin R$.

הכוונה: אם בשפה הנתונה מתקיימים הדברים הבאים:

1. הקלט הוא מכונה אחת בלבד.
2. התנאי של השפה מתייחס אך ורק לשפה של המכונה או למילים שהמכונה מקבלת אבל לא להתנהגות של המכונה כגון: עצירה, זמן חישוב, מספר מצבים, תאי זיכרון….

אז ניתן להשתמש במשפט רייס ולומר שהשפה לא ב R.