חישוביות ומבוא לסיבוכיות

מצגת 2- המחלקות R,RE

קידודים

- קלט למכונת טיורינג הוא מחרוזת של סימנים.
- נרצה אלגוריתמים שיעבדו על מכונות טיורינג, גרפים, מטריצות, פולינומים ועוד.
- לכן נרצה לבחור קידוד לאובייקטים.
- בדר"כ נבחין בין אובייקט X לבין הקידוד שלו <X>.

מכונת טיורינג אוניברסלית

- נרצה לתאר מכונת טיורינג כללית, שמקבלת קידוד של מכונה M וקלט x עבור M, ותענה מה יהיה הפלט של M על x.
- נרצה שיתקיים:

$$\underbrace{U(< M,x >)}_{\text{הקלט}} = \underbrace{M(x)}_{\text{הפלט}}$$

- כדי להשתכנע שזה אפשרי, נצטרך להראות שניתן לקודד מכונת טיורינג למחרוזת.

קידוד של מכונת טיורינג

- תהי $M = (Q,\Sigma,\Gamma,\delta,q_0,q_{accept},q_{reject},\bar{b})$
- נניח בה"כ ש-M היא מהצורה:
$Q = \{q_1, q_2, \dots, q_m\}, \Sigma = \{a_1, a_2, \dots, a_k\}, \Gamma = \{b_1, b_2, \dots, b_s\}$
כאשר $q_0,q_{accept},q_{reject}$ הם q_1,q_2,q_3 , $\bar{b}$ $a_1,a_2,\dots,a_k,$ הם $b_1,b_2,\dots,b_{k+1}$
- נקודד את L להיות 1, את R להיות 2 ואת S להיות 3.
- לכל $\delta(q,a) = (p,b,LRS)$ נקודד
$<\delta(q,a)> = 1^p01^b123$
- מעברים שונים יופרדו ע"י 00.
- קידוד אפשרי ל-M יהיה אם כן כך:
$< M \geq 1^m0\,1^{|\Gamma|}01^{|\Sigma|}0 < \delta(q_1,b_1) > 00 \dots \delta(q_m,b_s) > 00$

מכונת טיורינג אוניברסלית

- ראינו שניתן לקודד כל מכונת טיורינג.
- צריך עדיין להוכיח שקיימת מכונה אוניברסלית שיכולה לשחזר את הפעולה של כל מכונה שנתונה לה בתוך קידוד. איך זה יעבוד?
  - U תשמור את הקונפיגורציה ההתחלתית של M בסרט נוסף (או בסוף הקלט של $<M>$)
  - U גם תשמור סימן מיוחד שמציין את מיקום הראש הקורא
  - בכל שלב, U תעדכן את הקונפיגורציה הנוכחית של M לפי ההוראות שנתונות בקידוד של M.
  - כאשר U תזהה מצב סופי של M, גם U תעצור ותחזיר את הפלט של M.

מכונת טיורינג אוניברסלית

- נניח אם כן כי קיימת מכונת טיורינג, הנקראת מכונת טיורינג אוניברסלית, שעל קלט $< M,w >$ יכולה לבדוק ש-$< M >$ הוא קידוד של מכונת טיורינג ויכולה לחקות את ריצת M על w.

מכונת טיורינג אוניברסלית היא מחשב מודרני

מונח	מקבילה בעולם שלנו
מכונת טיורינג אוניברסלית	מחשב
מכונת טיורינג רגילה	אלגוריתם ספציפי במחשב.

הערה- בהמשך הקורס נוכל להשתמש בביטוי "לסמלץ" בלי לתאר לפרטי פרטים את התהליך. (U תסמלץ את ריצת M על הקלט שלה).

עוד נקודה בקידוד

- שימו לב שיתכן ו $\Sigma_U \neq \Sigma_M$ ואז הקלט של M מורכב מתווים שונים!
- לכן, בד"כ נקודד גם את הקלט x ב Σ_U, כלומר, הקלט יהיה בד"כ $< M >< x >$, ולא $x < M >$ כמו שהיינו מצפים.

- ואם הקידוד לא תקין?
- אם קידוד המכונה $< M >$ לא תקין, המכונה תקודד M_{STAM} שעוברת מיידית למצב q_{accept}.

כיצד המכונה U תרוץ?

- נבנה מ"ט הממשת את U. להלן סקיצה של מימוש מסוים, וזה יהיה "המכונה האוניברסלית" שעוד נפגוש המון בקורס.
- M_U על קלט $< M,x >$:
  - אם $< x >$ לא חוקי אזי נבצע לולאה אינסופית (ניתקע במצב מסוים שאינו סופי)
  - נכתוב על הסרט את הקונפיגורציה $< c >$, $< M >$ כאשר $< c >$ היא c_0 של M. (יהיה נח להשתמש במכונה מנויית Netflix)
  - כל עוד c אינה קונפיגורציה סופית, נחשב את $Next(< M > , < c >)$ ונכתוב אותה בתור הקונפיגורציה הנוכחית (עדכון של c הקודמת).
  - אם c קונפיגורציה סופית, נפלוט את קידוד הפלט המיוצג על ידי c.
- המכונה M_U מבצעת סימולציה צעד-צעד של ריצת M על x.

שקף "הפסקה" לעיכול הקידוד המוזר

- לקום בבקשה, למחוא כף פעמיים, ונמשיך הלאה.

מכונת טיורינג לקבלת שפות - תזכורת

- תזכורת: שפה L הינה תת קבוצה סופית או אינסופית של Σ^*
- הגדרה: מכונת טיורינג לקבלת שפות הינה מכונת טיורינג רגילה שמקיימת

$$F = \{q_{acc}, q_{rej}\}$$

- q_{acc} הינו מצב שמשמעותו היא שהמכונה **מקבלת** את הקלט שלה.
- q_{rej} הינו מצב שמשמעותו היא שהמכונה **דוחה** את הקלט שלה.

- <u>הערה</u> – גם למכונה לקבלת שפות יש פלט כמו שהוגדר במודל המקורי, אבל לרוב לא נתייחס אליו.

שפה של מכונה

בהינתן מכונה M, וקלט כלשהו $x \in \Sigma^*$:

- אם המכונה M בריצתה על x נעצרת על מצב q_{acc}, נאמר ש-M **מקבלת** את x.
  נסמן: $M(x) = 1$.
- אם המכונה M בריצתה על x נעצרת על מצב q_{rej}, נאמר ש-M **דוחה** את x.
  נסמן: $M(x) = 0$.
- אם המכונה לא עצרה – הפלט לא מוגדר.

<u>הגדרה</u>: השפה $L(M)$ הינה קבוצת כל המילים אשר המכונה M מקבלת.

נשים לב <u>שבשפה המשלימה</u> $\overline{L(M)} = \Sigma^* \setminus L(M)$, נמצאות כל המילים ש-M דוחה **וגם** כל המילים שעליהן M איננה עוצרת.

קבלת שפות למול **הכרעת** שפות

בהינתן שפה $L \subseteq \Sigma^*$,

הגדרה: נאמר שמכונה M_L **מקבלת (מזהה)** את השפה L אם מתקיים:

$x \in L \leftrightarrow M(x) = 1$.

הגדרה: נאמר שמכונה M_L **מכריעה** את השפה L אם מתקיים:

$x \in L \leftrightarrow M(x) = 1$,

ובנוסף, M_L עוצרת תמיד. (במילים אחרות: לכל L $x \notin$ מתקיים $M(x) = 0$)

הוכחת נכונות של שיוויון מהצורה $L(M) = L$

בהינתן שפה L, שבנינו לה מכונה M, נצטרך להוכיח $L(M) = L$ (כלומר, ששפת המכונה היא אכן השפה L).

- הטענה שאותה מוכיחים: $\forall x \in \Sigma^*: \left(x \in L \leftrightarrow x \in L(M)\right)$
- מבנה ההוכחה עבור מכונה **מקבלת**
- $x \in L \rightarrow \cdots \rightarrow M(x) = 1 \rightarrow x \in L(M)$
- $x \notin L \rightarrow \cdots \rightarrow (M(x) = 0\ or\ M(x)\ is\ not\ defined) \rightarrow x \notin L(M).$
- מבנה ההוכחה עבור מכונה **מכריעה**
- $x \in L \rightarrow \cdots \rightarrow M(x) = 1 \rightarrow x \in L(M)$
- $x \notin L \rightarrow \cdots \rightarrow M(x) = 0 \rightarrow x \notin L(M).$

המחלקות R ו-RE

- קבוצת השפות שקיימות עבורן מכונת טיורינג **המכריעה** אותן הינה המחלקה R:

$$R = \{L \subseteq \Sigma^* |\ there\ exists\ a\ turing\ mechine\ M\ such\ that\ L(M) = L\ and\ M\ halts\ on\ every\ input.\}$$

(R = recursive)

- קבוצת השפות שקיימת עבורן מכונת טיורינג **שמקבלת (מזהה)** אותן:

$$RE = \{L \subseteq \Sigma^* |\ there\ exits\ a\ turing\ mechine\ M\ such\ that\ L(M) = L\}$$

(RE = recursively enumerable)

דוגמאות לשפות ב - R:

- השפות הטריוויאליות: Σ^* , $\emptyset$
- השפות הרגולריות. אוטומט סופי דטרמיניסטי הוא מקרה פרטי של מכונת טיורינג (עם ראש קורא כותב שעובר פעם אחת בדיוק על הקלט)
- שפות של בעיות מוכרות שידוע לנו על אלגוריתם שפותר אותן (תמיד):

כמו:

- $L = \{x|x\ is\ a\ prime\ number\}$
- $L = \{< G >|G\ is\ a\ connected\ graph\}$
- $L = \{< G >|G\ is\ a\ graph\ that\ has\ a\ hamilton\ path\}$

- ועוד

תכונות של המחלקות R, RE

- $R \subseteq RE$:

מ"ט המכריעה שפה, בהכרח גם מקבלת אותה.

- <u>R סגורה למשלים</u>: אם $L \in R$ אז $\bar{L} \in \mathrm{R}$

<u>הוכחה</u>: תהי $L \in R$, לפי הגדרת R, יש מכונה M המכריעה את השפה L.

נבנה מכונה $\bar{M}$ שמכריעה את השפה $\bar{L}$:

$\bar{M}$ זהה ל-M למעט החלפה בין המצב המקבל והמצב הדוחה.

תכונות של המחלקות R, RE

המשך ההוכחה (נכונות):

$$x \in \bar{L} \to x \notin L \to M\ rejects\ x \to \bar{M}\ accepts\ x$$
$$x \notin \bar{L} \to x \in L \to M\ accepts\ x \to \bar{M}\ rejects\ x$$

קיבלנו ששפת המכונה $L(\bar{M})$ שווה ל-$\bar{L}$.

בנוסף, $\bar{M}$ עוצרת תמיד (כי M עוצרת תמיד)

מכאן שקיימת מכונה מכריעה ל-$\bar{L}$. לכן $\bar{L} \in R$. מ.ש.ל.

תכונות של R,RE - המשך

טענה: תהיינה $L_1,L_2 \in R$ אז $L_1 \cup L_2 \in \mathrm{R}$. (במילים אחרות: **R סגורה לאיחוד**)

הוכחה: תהיינה L_1,L_2 שייכות ל-R, ותהיינה M_1,M_2 מ"ט המכריעות את L_1,L_2 בהתאמה.

נגדיר מכונה חדשה M' על קלט x:

1. מריצה את M_1 על x ואם M_1 קיבלה, אזי M' עוצרת ומקבלת.
2. אחרת, מריצה את M_2 על x ועונה כמוה.

הוכחת נכונות (השלימו):

תכונות של R,RE - המשך

- **טענה:** תהיינה $L_1,L_2 \in RE$ אז $L_1 \cup L_2 \in RE$ (כלומר: **RE סגורה לאיחוד**)

הוכחה:

נסיון ראשון - הוכחה זהה להוכחה עבור R.

הוכחה: תהיינה L_1,L_2 שייכות ל-RE, ותהיינה M_1,M_2 מ"ט המקבלות את L_1,L_2 בהתאמה.

נגדיר מכונה חדשה M' על קלט x:

1. מריצה את M_1 על x ואם M_1 קיבלה, M' עוצרת ומקבלת.
2. אחרת, מריצה את M_2 על x ועונה כמוה.

מה הבעיה?

תכונות של R,RE - המשך

אבל: זה בעייתי כי יכול להיות שהמכונות לא עוצרות.....

פתרון: הרצה מבוקרת כמו שראינו שניתן לסמלץ מכונה בהינתן קידוד עבורה. ניתן גם לסמלץ שתי מכונות במקביל.

למשל: בכל צעד זוגי, להריץ את המכונה הראשונה ובכל צעד אי זוגי להריץ את המכונה השנייה.

נוכל להשתמש ב-2 סרטים – אחד עבור ריצת המכונה הראשונה ואחד עבור ריצת המכונה השנייה.

סגורה לאיחוד - הוכחה RE

- טענה: תהיינה $L_1,L_2 \in RE$ אז $L_1 \cup L_2 \in RE$

(רעיון ההוכחה)

נסמלץ את שתי המכונות במקביל.

אם אחת מהן קיבלה נקבל.

אם שתיהן דחו נדחה.

אחרת – המכונה שלנו פשוט תמשיך לרוץ לנצח (וזה בסדר...)

(השלימו את הוכחת הנכונות)

סגירויות של המחלקות R, RE

	R	RE
משלים		
איחוד		
חיתוך		
שרשור		
איטרציה		
היפוך		

- כל אחת מהטענות הללו דורשת הוכחה. (לחלקן צריך הרצה מבוקרת...)

סגירויות של המחלקות R, RE

	R	RE
משלים	V	X
איחוד	V	V
חיתוך	V	V
שרשור	V	V
איטרציה	V	V
רופיה	V	V

- כל אחת מהטענות הללו דורשת הוכחה. (לחלקן צריך הרצה מבוקרת...)

הפסקה

המחלקה co-RE

<u>הגדרה</u>:

$$coRE = \{L | \bar{L} \in \mathrm{RE}\}$$

כלומר, קבוצת כל השפות אשר לשפה המשלימה שלהן יש מכונה מקבלת.

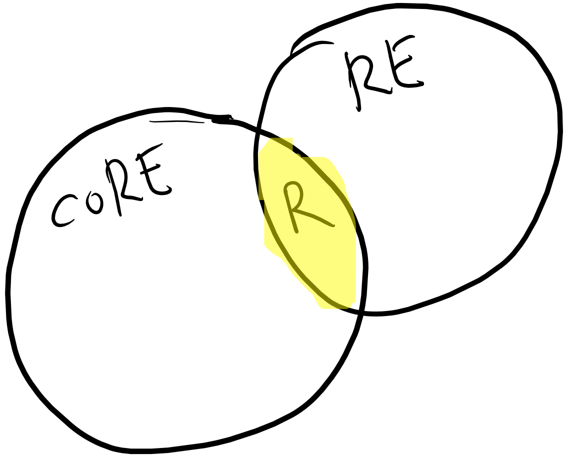
<u>הגדרה שקולה</u>:

coRE היא קבוצת השפות שיש עבורן מכונה M כך ש:

- אם המכונה עוצרת, היא עונה נכון.
- המכונה חייבת לעצור על כל קלט ש**לא** בשפה.

<u>אבחנה</u>: כשמגדירים שפה $L \in coRE$ ע"י מכונה M כנ"ל, לא בהכרח מתקיים $L = L(M)$.

(שאלה- מתי זה כן מתקיים ?)



תכונות של המחלקה coRE

- $\boldsymbol{R \subseteq coRE}$:

מ"ט המכריעה שפה, בהכרח עוצרת על כל קלט שאיננו בשפה.

- **טענה:** $R = RE \cap coRE$

הוכחה: (הכלה דו כיוונית)

$\underline{\Rightarrow}$: ראינו ש- $\underline{R \subseteq RE}$ וגם ש- $\underline{R \subseteq coRE}$ מכאן ש- $\underline{R \subseteq RE \cap coRE}$. מ.ש.ל.

דרך נוספת להוכיח את הכיוון הנ"ל:

תהי $L \in R$, משום ש- $R \subseteq RE$ נובע כי $L \in RE$

מצד שני R סגורה למשלים, לכן גם $\bar{L} \in R$ מכאן שמתקיים $\bar{L} \in RE$.

לכן, לפי הגדרת coRE: $L \in coRE$

$R = RE \cap coRE$

המשך ההוכחה:

←: תהי $L \in RE \cap coRE$ (צ"ל $L \in R$)

כלומר: $L \in RE$, לכן קיימת לה מכונה M_1 ה**מקבלת** את L.

וגם $L \in coRE$, לכן קיימת לה מכונה M_2 המזהה קלטים שלא ב-L ודוחה אותם.

(זאת בעצם המכונה ה**מקבלת** של השפה $\bar{L}$)

(שימו לב, אי אפשר להניח שמדובר באותה מכונה)

$R = RE \cap coRE$

- טענה: $R = RE \cap coRE$

המשך ההוכחה:

נתאר מכונה חדשה M^*:

M^* על קלט X:

1. מריצה את M_1 על X, אם קיבלה – M^* גם מקבלת
2. מריצה את M_2 על X, אם דחתה – M^* גם דוחה

האם זה כיוון טוב?

$R = RE \cap coRE$

- טענה: $R = RE \cap coRE$

המשך ההוכחה: שיפור של הבנייה, (כי צריך הרצה מבוקרת...):

נתאר מכונה חדשה M^*:

M^* על קלט X:

1. מריצה **במקביל** את M_1 על X ואת M_2 על X.
2. בכל שלב, אם M_1 קיבלה – M^* גם מקבלת. ואם M_2 דחתה – M^* גם דוחה.

האם זה כיוון טוב?

$R = RE \cap coRE$

- הוכחת נכונות:

$$x \in L \ \rightarrow M_1\ accepts\ X \ \rightarrow M^*\ accepts\ X \ \rightarrow M^*(X) = 1$$

$$x \notin L \ \rightarrow M_2\ rejects\ X \ \rightarrow M^*\ rejects\ X \ \rightarrow M(X) = 0$$

השפה $L_u = \{< M,x > | M \text{ accepts } x\}$

- האם L_u שייכת ל-RE? האם שייכת ל-R?

<u>טענה</u>: $\boldsymbol{L_u \in RE}$

<u>הוכחה</u>: נתאר מכונת טיורינג M_u המקבלת את השפה L_u:

M_u על קלט $< M,x >$:

- M_u תסמלץ את ריצת M על הקלט x, ותענה כמוה.

(ראינו שניתן "לסמלץ" ריצה של מכונה שנתונה לנו כקידוד.)

נכונות:

אם $< M,x > \in L_u$ אז M מקבלת את הקלט x, מכאן ש-M_u מקבלת את הקלט $< M,x >$

ואזי $< M,x > \in L(M_u)$

אם $< M,x > \notin L_u$ אז M לא מקבלת את הקלט x, מכאן ש M_u לא מקבלת את הקלט $< M,x >$

ואזי $< M,x > \notin L(M_u)$

השפה $L_u = \{< M,x > | M \text{ accepts } x\}$

הערה – לצורך השלמות של האלגוריתם, צריך שנתייחס למקרה שבו הקידוד של המכונה איננו תקין.

יש שתי גישות עיקריות:

- יחסית קל לבדוק בתחילת האלגוריתם אם הקלט תקין, לכן אפשר להניח שהקלט תקין.
- לכל קידוד של מכונה שאיננו תקין, נתייחס בתור המכונה M_{stam} שהיא מכונה שתמיד עוצרת על מצב מקבל.

כעת, נחזור לשאלה מהשקף הקודם - האם L_u שייכת ל-R?

תשובה: $\boldsymbol{L_u \notin R}$

כדי להוכיח זאת אנחנו בעצם צריכים להוכיח ש**כל** אלגוריתם בעולם, לא יכול להכריע את השפה הזו.

אז - המשך יבוא...

השפה $L_D = \{< M > | M \ accepts < M >\}$

- האם L_D שייכת ל-RE? האם שייכת ל-R?

<u>טענה</u>: $\boldsymbol{L_D \in RE}$

<u>הוכחה</u>: (ממש כמו ההוכחה עבור השפה L_u)

נתאר מכונת טיורינג M_D המקבלת את השפה L_D:

M_D על קלט $< M >$:

- M_D תסמלץ את ריצת M על הקלט $< M >$, ותענה כמוה.

<u>נכונות</u>:

$$< M > \in L_D \rightarrow M_D(< M >) = 1$$
$$< M > \notin L_D \rightarrow M_D(< M >) = 0 \ or \ undefined$$

שיטת הליכסון של קנטור

קנטור הוכיח שבקטע $[0,1]$ יש כמות לא בת מניה של מספרים.
מבנה ההוכחה:

- נניח בשלילה שיש מספר בן מניה של מספרים בין 0 לאחת
- לשם הנוחות ניקח תת קבוצה A, של מספרים בין אפס לאחת, שהיצוג שלהם מכיל רק 2 תווים. (למשל, ניקח את 0,1)
- נסדר את המספרים הללו בסידור כלשהו (מובטח שקיים סידור כזה מכיוון שהנחנו שמדובר בקבוצת בת מניה)

- נבנה מספר חדש c ע"י זה שניקח את הספרות שבאלכסון ונחליף אותן, אחת יהפוך לאפס, ואפס יהפוך לאחת.
- c שייך לקבוצה של המספרים A, לכן קיים אינדקס i כך שהמספר שלנו הוא בדיוק האיבר n_i.
- אבל, לפי הבנייה של c, הערך של c בתו ה-i שונה מהערך של n_i בתו ה-i.
- סתירה.

	a1	a2	a2	a4	.....			.
n1	0	0	0	0	1	1		
n2	1	1	1	0	0	0		
n3			1					
				0				
.....					0			
.....						0		
							1	
								0

השפה $L_D = \{< M > | M \ accepts < M >\}$

- האם L_D האם שייכת ל-R?

<u>טענה</u>: $\boldsymbol{L_D \notin R}$

<u>הוכחה</u>: (מבנה ההוכחה – ניצור **פרדוקס**: נניח בשלילה ונראה שזה גורר סתירה).

נניח בשלילה ש-$L_D \in R$

אזי קיימת לה מכונה M_D המכריעה אותה.

נבנה מכונה חדשה A_D שפועלת כך:

A_D על קלט $< M >$:

- מריצה את M_D על הקלט $< M >$ ועונה **הפוך**.

מכיוון שיש מספר בן מניה של מכונות טיורינג, ניתן לסדר את כל המכונות בסידור כלשהו

(למשל סידור לקסיקוגרפי של מחרוזות הקידוד)

אז נבנה טבלה אינסופית של כל הזוגות האפשריים של מכונות טיורינג, כך שבתא ה-i,j נאחסן את התשובה לשאלה: האם המכונה M_i מקבלת את הקלט $< M_j >$?

	<M1>	<M2>	<M3>	<M4>	.....			
<M1>	0	0	0	0	1	1		
<M2>	1	1	1	0	0	0		
<M3>			1					
<M4>				0				
.....					0			
.....						0		
							1	
								0

המשך ההוכחה- $L_D \notin R$

- המכונה M_D יודעת למלא את התאים של אלכסון הטבלה, לכן המכונה A_D גם כן יודעת למלא את תאי האלכסון של הטבלה (ממלאת בדיוק את ההפך).
- המכונה A_D היא גם מכונת טיורינג, ולכן היא נמצאת במקום כלשהו בטבלה, כלומר קיים אינדקס i כך ש-$A_D = M_i$.
- אם כך מה הערך בתא ה-i,i בטבלה?

המשך ההוכחה- $L_D \notin R$

- מה הערך בתא ה-i,i בטבלה? כלומר מה הערך של $M_i(\langle M_i \rangle) = A_D(\langle M_i \rangle)$?

- אם A_D מקבלת על הקלט $\langle A_D \rangle$- כלומר בתא ה-i,i צריך להיות ערך 1
אז בהרצה של A_D על הקלט $\langle A_D \rangle$, המכונה M_D תענה 1, ואז A_D תפלוט 0.
-סתירה

- אם A_D לא מקבלת על הקלט $\langle A_D \rangle$ - כלומר בתא ה-i,i צריך להיות ערך 0
אז בהרצה של A_D על הקלט $\langle A_D \rangle$, המכונה M_D תענה 0, ואז A_D תפלוט 1.
-סתירה

מ.ש.ל

השפה $HP = \{< M,x > | M\ halts\ on\ x\}$

- האם HP שייכת ל-RE? האם שייכת ל-R?

<u>טענה</u>: $\boldsymbol{HP \in RE}$

<u>הוכחה</u>: נתאר מכונת טיורינג M_{HP} המקבלת את השפה HP:

M_{HP} על קלט $< M,x >$: (אם הקידוד לא חוקי – עוצרת ודוחה)

- M_{HP} תסמלץ את ריצת M על הקלט x, אם עצרה – קבל (כלומר, עבור ל q_{accept})

נכונות:

אם $< M,x > \in HP$ אז M עוצרת על הקלט x, מכאן ש-M_{HP} מקבלת את הקלט שלה $< M,x >$, ואזי $< M,x > \in L(M_{HP})$

אם $< M,x > \notin HP$ אז M לא עוצרת על הקלט x, מכאן ש-M_{HP} לא מקבלת את הקלט $< M,x >$ (בפרט, היא לא עוצרת) ואזי $< M,x > \notin L(M_{HP})$. מ.ש.ל.

בעיית העצירה

האם HP שייכת ל-R?

תשובה: $\boldsymbol{HP \notin R}$

<u>רעיון ההוכחה</u>:

- נניח בשלילה שיש מכונה M_{HP} שמכריעה את HP,
- נשתמש ב-M_{HP} כדי להכריע את L_D
- נקבל סתירה

בעיית העצירה

טענה: $\boldsymbol{HP \notin R}$

<u>ההוכחה</u>:

נניח בשלילה $HP \in R$, כלומר יש מכונה M_{HP} שמכריעה את HP

כעת, בהינתן קידוד של מכונה כלשהי <M>, שנרצה **להכריע** האם שייכת לשפה L_D או לא:

נבנה מכונה M' אשר זהה ל-M, רק שבכל פעם ש-M דוחה, M' נכנסת ללולאה אינסופית:

M' <u>על קלט y</u>:

- מסמלצת את ריצת M על y, אם קיבלה- מקבלת.
- אם דחתה – M' נכנסת ללולאה אינסופית.

בעיית העצירה

כעת נשלח אל המכונה M_{HP} את הקידוד $\ll M', <M \gg$.

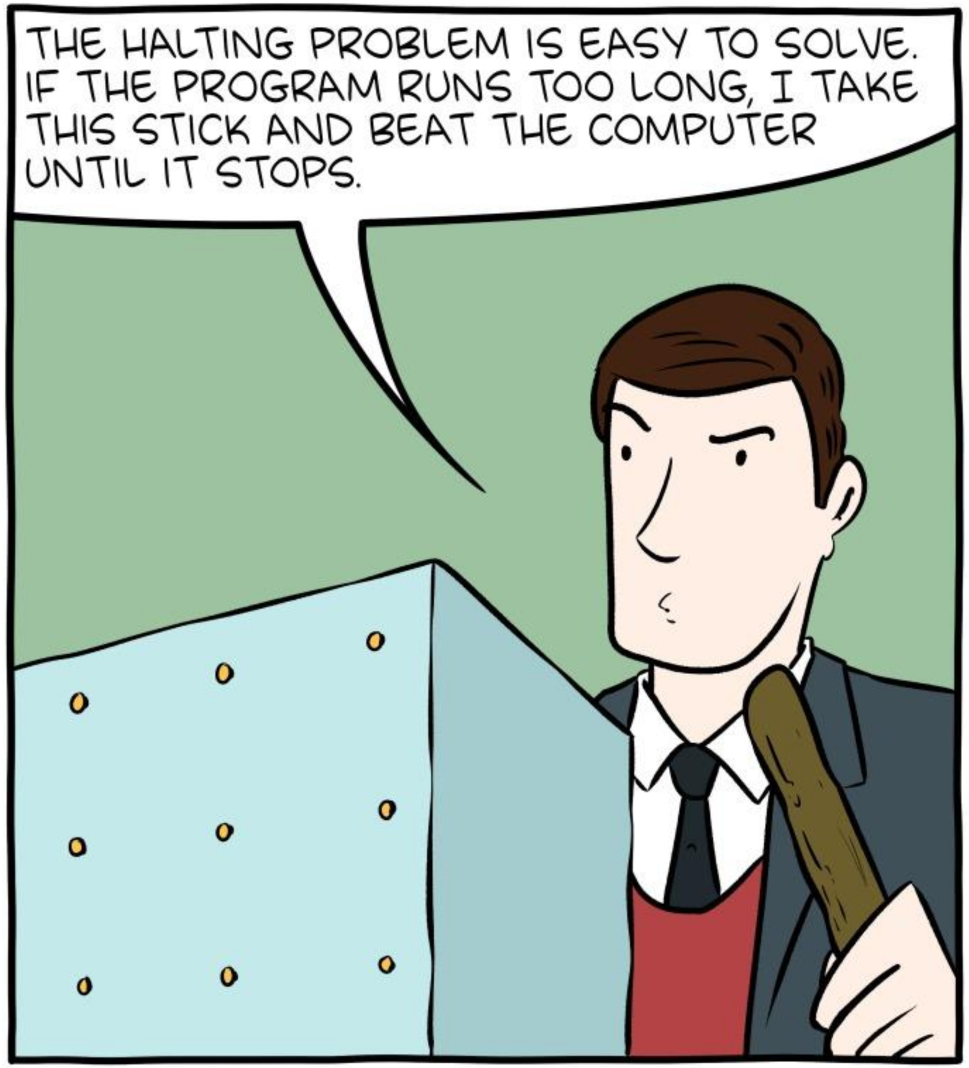
אם M_{HP} תענה שהקלט בשפה HP, זה אומר ש- M' עוצרת על $<M>$, ולפי בניית M' נסיק ש-M **מקבלת** את $<M>$, ונסיק מכך כי $<M> \in L_D$.

- אם M_{HP} תענה שהקלט לא בשפה HP, זה אומר ש-M' לא עוצרת על $<M>$, ולפי בניית M' נסיק ש-M **לא מקבלת** את $<M>$ (דוחה או נכנסת ללולאה אינסופית), ונסיק מכך כי $<M> \notin L_D$.

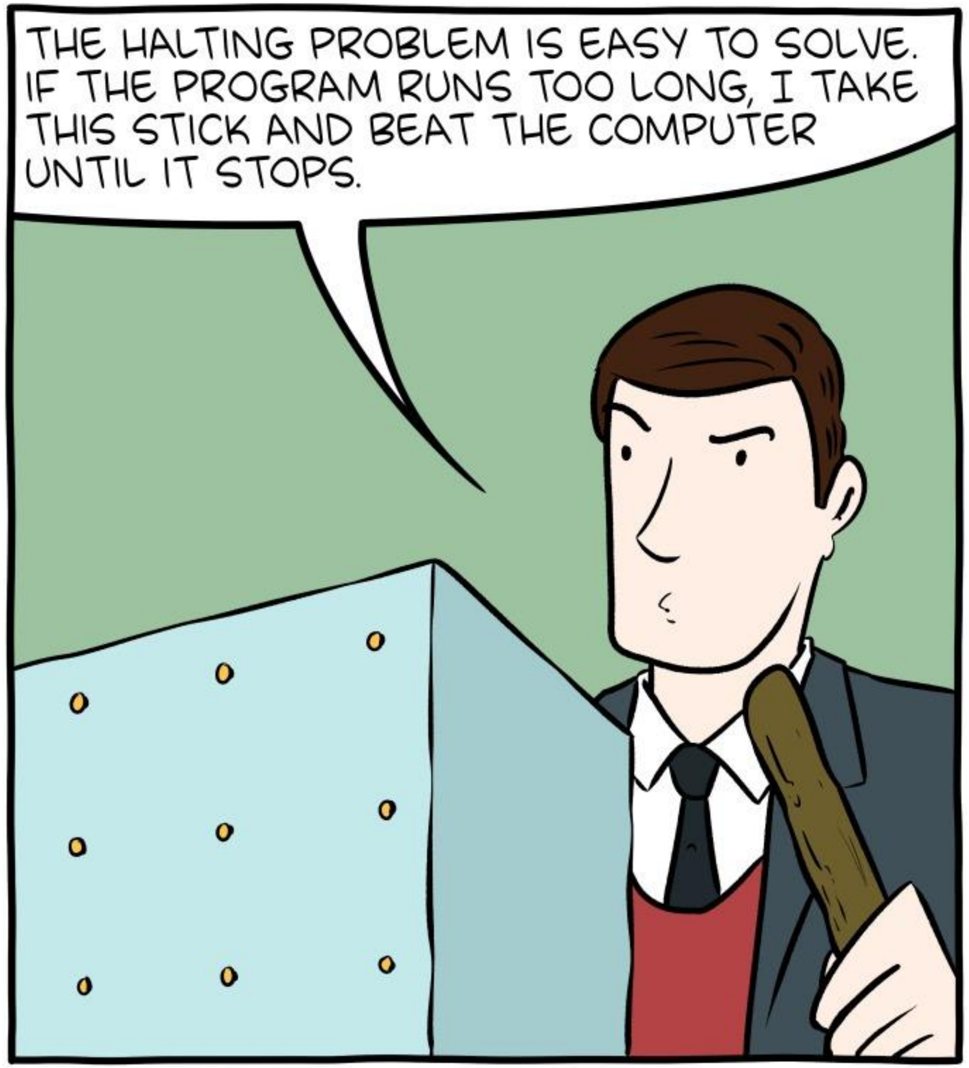
קיבלנו אלגוריתם **שמכריע** את השפה L_D.

סתירה! מ.ש.ל

הערה - בשיעור הבא נחקור את המבנה של ההוכחה הנ"ל ונקבל כלי חזק להוכחה ששפות מסוימות **אינן שייכות למחלקה R.**



What if Alan Turing had been an engineer?



What if Alan Turing had been an engineer?