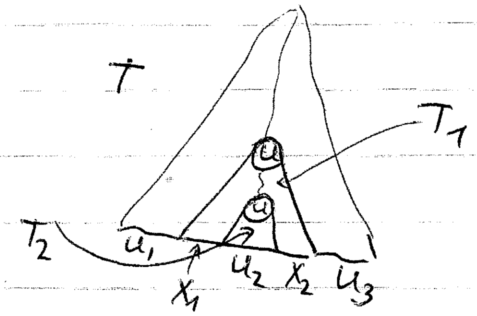
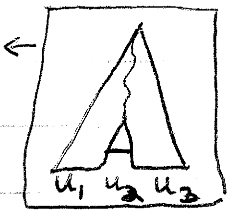
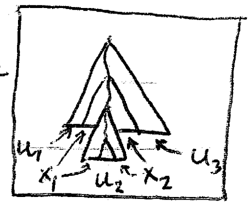
הוכחה:

נתבונן בעץ גזירה T של y ב-$\tilde{\tilde{G}}$. כיוון שהדרגה של $T \geq r$, ומספר העלים $> r^{n+1}$, אורך המסלול הארוך ביותר מהשורש של T לעלה הוא לפחות $n+2$, ולכן מכיל לפחות שני משתנים זהים. נתבונן במסלול זה, ויהי A משתנה כזה, שהקדקד הלפני-אחרון שלו הוא הנמוך ביותר מכל המשתנים הכפולים.

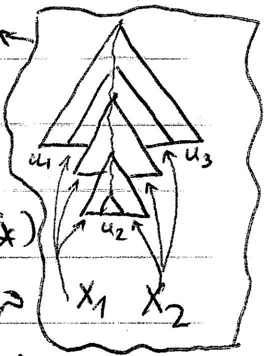




אם נציב במקום T_1 את T_2, נקבל עץ הגוזר מילה $u_1u_2u_3$.



אם נציב במקום T_2 את T_1, נקבל עץ הגוזר מילה $u_1x_1^2u_2x_2^2u_3$.



אם נעשה את ההצבה הנ"ל k פעמים (רקורסיבית) נקבל עץ הגוזר מילה $u_1x_1^ku_2x_2^ku_3$. כל המילים הללו נגזרות בצורה חוקית, ולכן (*) מתקיים.

(*) מתקיים כיוון שאין ב-$\tilde{\tilde{G}}$ גזירות "ריקות" ובמרווח בין A הלפני אחרון, וה-A האחרון חייב להיגזר קטע (או קטעים) לא ריקים.

(**) כיוון שבעץ T_1 אף משתנה לא מופיע פעמיים, גובה העץ הוא לכל היותר $n+1$. כיוון שהדרגה חסומה ע"י r, האורך של המחרוזת $x_1u_2x_2$ הנגזרת מ-A הלפני-אחרון $\leq r^{n+1}$.

מ.ש.ל.

דוגמאות לשימוש בלמת השאיבה/ניפוח

הסכמה, באופן כללי, היא כזאת. תהי L השפה שעבורה אני מנסה להוכיח כי היא לא ח"ה. נניח בשלילה שהיא כן ח"ה. אזי למת השאיבה חלה עליה עם פרמטר N מסוים (שאין אנו יודעים עליו דבר). נבחר $w \in L$, שאורכה עולה על N, ויש לה תכונה (כאן כל החכמה בשימוש בלמה!) שכל ניפוח (או שאיבה, או מספיק ניפוחים) בהכרח "יצור" מילה $w' \notin L$. זאת הסתירה הרצויה.

דוגמא א': $L=\{0^i1^i2^i\}_{i=0}^{\infty}$ איננה ח"ה. * (קלה)

נפעיל את הסכמה הכללית, ונבחר $w=0^k1^k2^k$, $k>N$. למת הניפוח חלה על w. היכן יכולים ליפול קטעי ניפוח? אם יש קטע ניפוח הגולש מעבר לבלוקים, נעשה ניפוח ונקבל מילה לא מהסוג $0^*1^*2^*$, שהיא ודאי לא בשפה. אחרת, כיון שיש 3 בלוקים ולכל היותר 2 קטעי ניפוח, ק"ים בלוק המכיל קטע ניפוח, ובלוק הלא מכיל קטע כזה. נעשה ניפוח, ונקבל מילה עם בלוקים באורך שונה, $\notin L$. מ.ש.ל.

דוגמא ב': *

$L=\{ww\} \subseteq \{0,1\}^*$ איננה ח"ה.

נבחר $L \ni w=0^k1^k0^k1^k$, כאשר $k>N$. האבחנה העיקרית לגבי קטעי הניפוח היא שהם חייבים ליפול בבלוקים סמוכים ע"פ (**). נבצע שאיבה. כתוצאה מכך נקבל מחרוזת מסוג $0^{k_1}1^{k_2}0^{k_3}1^{k_4}$ כך ששניים מה-k_i-ים הם ה-k הישן, ומבין שני האחרים, לפחות אחד קטן ממש. בהכרח יתקיים $k_1 \neq k_3$ או $k_2 \neq k_4$. אך במקרה זה יש בהכרח $\notin L$! מ.ש.ל.

(37)

<u>דוגמא ג'</u> (*) $L=\left\{w\#w' \;\middle|\; \begin{matrix} w,w'\in\{a,b\}^* \\ |w|=|w'| \\ w\neq w' \end{matrix}\right\}$ אינה ח"ה.

קשה יותר

נבחר מחרוזת $\overset{L}{\in}w = \underbrace{0}_{n!+n}\underbrace{1}_{n}\#\underbrace{0}_{n}\underbrace{1}_{n!+n}$

כאשר $n>N$.

קטעי הניפוח חייבים ליפול משני צידי #, ואורכם חייב להיות זהה, נגיד k. (אם זה לא המקרה, נעשה ניפוח, ונקבל מילה לא ב-L, כיוון ש $|w|\neq|w'|$).
כיוון ש $k<n$ (לפי תנאי (**)) חייבים קטעי הניפוח ליפול בבלוקים <u>הסמוכים</u> ל-#. לכן, אחרי t ניפוחים נקבל מחרוזת

$$w_t = 0^{n!+n}1^{n+t\cdot k}\#0^{n+t\cdot k}1^{n!+n}.$$

אבל, n! מתחלק ב-k! לכן, אם נבחר $t_0=n!/k$, נקבל כי

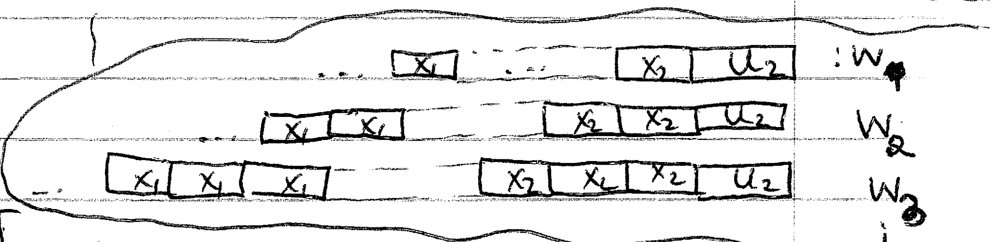
$$w_{t_0} = 0^{n!+n}1^{n!+n}\#0^{n!+n}1^{n!+n}\notin L!$$

סתירה. מ.ש.ל.

<u>דוגמא ד'</u> (*) $L=\left\{\begin{matrix}\text{ייצוגים עשרוניים} \\ \text{של } 3^i\end{matrix}\right\}_{i=0}^{\infty}$ אינה ח"ה.

כ"א $L=\{1,3,9,27,81,\ldots\}\subseteq\{0,..9\}^*$

נניח כי L כן ח"ה, ונבחר $w\in L$, $|w|>N$. נבנה סדרה של ניפוחים: $w_1=w, w_2, w_3, \ldots$



נשים לב כי:

(א) $|w_{i+1}|-|w_i| = |x_1|+|x_2| < N$.
לכן, היחס w_{i+1}/w_i <u>כמספרים עשרוניים</u> הינו חסום.

(ב) <u>כמספרים עשרוניים</u>, $w_{i+1}-w_i$ מסתיים ב-$|u_2|+i\cdot|x_2|$ אפסים ולכן מתחלק ב-$10^{|u_2|+i|x_2|}$.

נניח בשלילה שה-w_i-ים הם אכן ייצוגים של חזקות של 3. אזי, כמספרים

$$w_{i+1}-w_i = 3^{r_{i+1}}-3^{r_i} = 3^{r_i}\underbrace{(3^{r_{i+1}-r_i}-1)}_{\text{חסום ע"פ א'}}$$

לכן החזקה המקס' של 10 המחלקת את $w_{i+1}-w_i$ חייבת להיות <u>חסומה</u>.
אך ע"פ ב' היא לא חסומה - סתירה!

הערות מסיימות על שפות ח"ה ואוטומטים

* אם L ח"ה, $\bar{L}$ לא חייבת להיות כנ"ל. אם A_L דטרמיניסטי, $\bar{L}$ גם תהיה חסרת הקשר, אך, בניגוד לאוטומטים סופיים, יש שפות ח"ה שעבורן החוסר דט' הינו הכרחי.

דוגמא:

$$L = \{ww' \mid |w| = |w'|,\ w \neq w'\}$$
ח"ה (ראינו)

$$\bar{L} = \{ww\} \cup \{u \mid |u| \text{ אי-זוגי}\}$$

$\bar{L}$ לא ח"ה $\Leftarrow$ אכן,
$$\bar{L} \cap \underbrace{\{u \mid |u| \text{ זוגי}\}}_{\text{רגולרי}} = \{ww\}$$
לא ח"ה, ראינו!

בפרט, ה אוטומט של L חייב להיות לא דט'.

* אם L_1, L_2 ח"ה, אין זה אומר דבר על $L_1 \cap L_2$. אכן, לא ניתן להכליל L_1, L_2 בו-זמנית: צריך 2 מחסניות.

דוגמא:
$$\underbrace{\{a^n b^n c^*\}_{n=1}^{\infty} \cap \{a^* b^n c^n\}_{n=1}^{\infty}}_{\text{ח"ה}} = \underbrace{\{a^n b^n c^n\}_{n=1}^{\infty}}_{\text{איננה ח"ה}}$$

* בהינתן דקדוק G קל לבדוק האם G יוצר שפה ריקה או לא. קצת קשה יותר לבדוק האם G יוצרת מילה נתונה w. אפשר לעשות זאת, לדוגמא, כך:

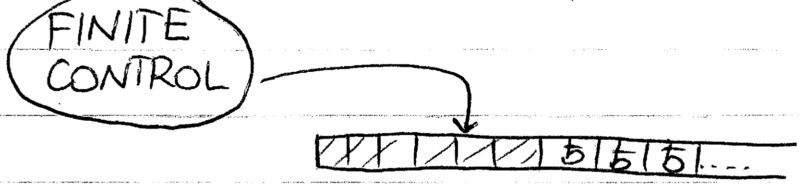
$$\underset{\text{האוטומט המתאים}}{G \to A_G} \to \underset{\text{אוטומט החיתוך: } A_w \text{ האוטומט הסופי המקבל את } \{w\}}{A_G \cap A_w} \to \underset{\text{מעבר לדקדוק המתאים}}{G'} \dashrightarrow \text{האם יוצר שפה ריקה?}$$

לעומת זאת, לקבוע האם G יוצר את Σ^*, לא אפשרי...

פרק ג

מכונת טיורינג

הגדרה של מכונת טיורינג דומה להגדרת המודלים אשר ראינו קודם:



- ראש קורא/כותב (עם מס' סופי של מצבים), אשר בכל שלב יכול לזוז משבצת אחת שמאלה או ימינה;
- סרט (עם התחלה ובלי סוף), אליו מותר לכתוב, ולקרוא ממנו. א"ב הסרט, כרגיל, סופי. פרט לחלק המכיל קלט, בתחילת העבודה הסרט מלא בסימני "␢" (blank).
- מוסכם כי הראש נמצא בתחילת הריצה במשבצת הראשונה, אשר כתוב "␢"; הריצה מסתיימת אם הראש מגיע למצב מקבל, או מצב דוחה (שאר המצבים אינם מקבלים/דוחים).

פורמלית:

פונקצית המעבר

$\Sigma \subseteq \Gamma$, Q , $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L,R\}$

Σ – א"ב הקלט

Γ – א"ב הסרט, מכיל "␢"

Q – מצבי הראש, מכיל q_r, q_a (q_r – דוחה, q_a – מקבל)

q_0: מצב התחלה

Q (בתחום) – מצב ישן של הראש

Γ (בתחום) – האות שהראש קורא על הסרט

Q (בטווח) – מצב חדש

Γ (בטווח) – האות שראש כותב על הסרט

$\{L,R\}$ – כיוון התזוזה של הראש

40

למרות הדימיון החיצוני למודלים אחרים שלמדנו, מכונת טיורינג שונה מהם באוניברסליות שלה. בפרט, הכוח החישובי שלה זהה לזה של מחשב, ואין זה קשה לתרגם תוכנית מחשב (נגיד, בשפת C) למכונת טיורינג, אשר על קלט זהה תתן אותה תוצאה. כיוון שבפרק הזה הדגש יהיה על מה ניתן ומה לא ניתן לחשב, נדלג על תוכנית בעזרת מ"ט (מכונת טיורינג), ונשאיר את הדוגמאות לתירגול. הגישה שלנו:

תוכנית בשפת C (או בכל שפה אחרת) ⟷ TM

מערכת ההפעלה: תוכנית אשר יודעת לפענח ולהריץ תוכניות אחרות. ⟷ UTM

מכונת טיורינג אוניברסלית, אשר יודעת לפענח קידוד של מכונת טיורינג, ולעשות לה סימולציה

UTM היא אחת, מסוימת, וקידודה ... (על תוכנית בה?)

הא"ב של UTM הוא $\{0,1\}^*$; הקלט שהיא מקבלת הוא צוג מחרוזת בינארית $\langle M\rangle, I$ המקודדת תאור של מ"ט M, וקלט I שלה.

תשוב: הקידוד תמיד יהיה בינארי, למרות שהא"ב של M יכול להיות מכל גודל. כך אכן קורה במחשב אמיתי: למרות שתוכנית כתובה ב"אנגלית" והגדירה המין משתנים, המחשב יקבל אותה, בסופו של דבר, כמחרוזת של 0-ים ו 1-ים.

לפשטות אנו מניחים כי ⟨M⟩ מקודדת כמו עם קידוד בינארי, שקולה למכונה המקורית

תגלו בו תפתחו קידוד אוניברסלי של TM

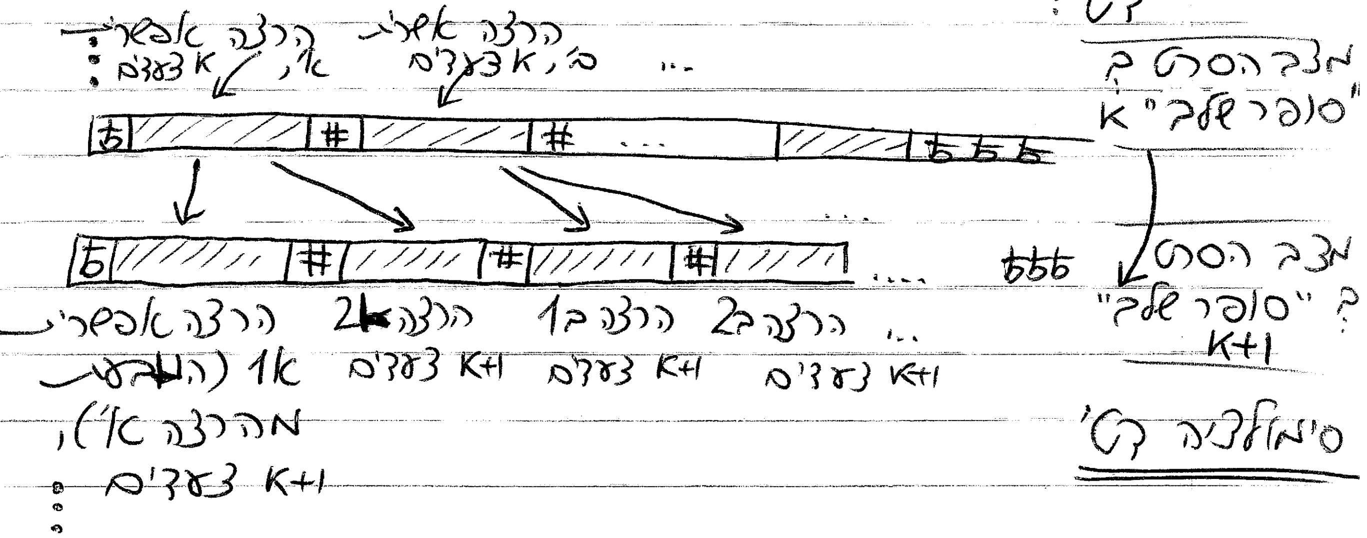
תשוב: תוכנית M → תאור תפקיד של מודל חישובי
→ ⟨M⟩, סתם מחרוזת של 0-ים ו 1-ים.

שתי המשמעויות הללו של תוכנית בנויות הפרק הזה.

לפני שנמשיך נעיר מס' הערות על מכונות טיורינג:

* יכולנו לדבר על מכונות עם הרבה סרטים, ובפרט סרט INPUT נפרד מהסוג READ ONLY (כפי שהיה דומה). קל להראות כי המודל שהגדרנו יכול לעשות סימולציה למודל הנ"ל; נשלם בהאטה מסוימת במהירות החישוב, אך התוצאה הסופית תהיה זהה.

* כנ"ל, מכונת טיורינג ~~לא~~ דט' שקולה מבחינת כוח החישוב שלה למכונת טיורינג דטרמיניסטית. הסימולציה, בגדול, תעבוד כך: נריץ במקביל את כל ההרצות האפשריות (עד שלב k) של המכונה הלא דט':



"הרצה" היא מצב הסרט (החלק המשמעותי, בלי "⊔"), כ"א מה שכתוב על הסרט + מיקום ומצב הראש, המסומן במשבצת בה נמצא הראש.

המכונה הדט' תריץ במקביל את כל ההרצות האפשריות של המכונה הלא דט' (נספר לכל k סופי!); אם באחד ההרצות המכונה הלא דט' תקבל את הקלט, המכונה הדט' המבצעת סימולציה גם תקבל.

שפות ומכונות טיורינג

שפה L:

* ניתנת להכרעה/ זיהוי = רקורסיבית
אם ק"מת מ"ט M_L המכריעה/מזהה אותה.

** ניתנת לקבלה = נל"ר
אם ק"מת מ"ט M_L המקבלת אותה. להזכיר, להבדיל מהכרעה, M_L המקבלת את L אמורה להחזיר "כן" רק אם $w \in L$; אם $w \notin L$ יכולה להחזיר "לא" או לא להחזיר כלום, ולא לעצור לעולם.

*** לא ניתנת לקבלה - כל שאר השפות.

האם יש בכלל שפות אשר לא ניתנות לקבלה?
ועוד איך! הדבר נובע משיקולי עוצמה; נצטמצם לשפות בינאריות: $\Sigma=\{0,1\}$

שפה ← תת קבוצה של Σ^*;	TM ← $\langle M\rangle$ מילה ב-Σ^*;
כל השפות ← כל התת-קבוצות של Σ^*, 2^{Σ^*}.	כל TM ← $\{\langle M\rangle\}$ תת קבוצה אחת של Σ^*.

כדי להוכיח כי ק"מת שפה שלה לא מתאימה אף מ"ט, מספיק להוכיח כי לא ק"מת פונקציה חח"ע f,

$$f: 2^{\Sigma^*} \longrightarrow \Sigma^*$$

אך הדבר נכון לכל קבוצה A, ובפרט ל-Σ^*!

* **משפט**: לכל קבוצה K, לא קיימת פונקציה חח"ע f,
$$f:2^K \to K.$$
צורה שקולה: לא קיימת פונקציה על φ,
$$\varphi: K \to 2^K.$$

* **הוכחה**: (צורה שקולה)
נניח בשלילה שכן קיימת φ כנ"ל. נגדיר קבוצה $K' \subseteq K$ באופן הבא:
$$K' = \{x \in K \mid x \notin \varphi(x)\}.$$
כיוון ש-φ על, קיים $y \in K$ כך ש-$\varphi(y) = K'$.
האם $y \in \varphi(y)$?

a. אם $y \in \varphi(y) = K'$ $\xRightarrow[K']{\text{לפי הגדרת}}$ $y \notin K'$ סתירה

b. אם $y \notin \varphi(y) = K'$ $\xRightarrow[K']{\text{לפי הגדרת}}$ $y \in \varphi(y)$ סתירה.

כיוון שח"ב להתקיים a. או b., קבלנו סתירה להנחה שלנו.
מ.ש.ל.

* **שים לב**: ההוכחה הינה קיומית ולא קונסטרוקטיבית, ולא נותנה לנו שום דוגמא לשפה אשר אינה ניתנת לקבלה!

(13) המטרה הבאה שלנו היא לבנות שפה מפורשת הניתנת לקבלה, אך לא ניתנת לזיהוי. נבנה גם שפה מפורשת אשר לא ניתנת לקבלה.

<u>הגדרה:</u> שפה $A_{TM} \subseteq \{0,1\}^* \times \{0,1\}^*$ מוגדרת ?

$$A_{TM} = \{(\langle M \rangle, I) \mid M(I) = accept\}$$

<u>משפט:</u> A_{TM} ניתנת לקבלה, אך לא ניתנת לזיהוי.

הוכחה:

(*) A_{TM} מתקבלת ע"י UTM כפי שהגדרנו אותה.

אך,

$$UTM(\langle M \rangle, I) = accept \iff M(I) = accept.$$

(*) נראה כי לא ד"מ מ"ט המכריעה את A_{TM}.

נתבונן בטבלה הבאה, המהווה תוצאת ריצה של כל מ"ט על כל המחרוזות הבינאריות:

כל Σ^* לפי הסדר הלקסיקוגרפי

	w_1 ε	w_2 0	w_3 1	w_4 00	w_4 01	w_5 10	w_6 11	w_7 000	w_8 001	..
M_1	$\boxed{0}$	1	0	1	1	0	...			
M_2	0	$\boxed{0}$	0	1	0	0	...			
M_3	1	1	$\boxed{1}$	1	1	0	...			
M_4	1	0	0	$\boxed{0}$	0	1	....			
$\vdots$										

(האלכסון)

כל מ"ט (עם $\Sigma=\{0,1\}^*$) הממוספרות בסדר הלקסי' של $\langle M_i \rangle$

כאשר במקום ה M_i, w_j רשום 1 אם $M_i(w_j)=accept$, 0 אחרת!

(הטבלה מוגדרת היטב, למרות שאין אנו יכולים לחשב אותה.)

נגדיר שפת האנטי-אלכסון, D, כ:

$D = \{w_i \in \Sigma^* \mid M_i(w_i) \neq accept\}$.

שם לב כי השורות של הטבלה הנ"ל מתארות את השפות המתקבלות ע"י TM המתאימות
השורה של M_i ← שפה L, המתקבלת ע"י M_i.
השפה D שבנינו, באופן בנוי, שונה מכל השפות הנ"ל. כיוון בטבלה מופיעות כל מ"ט, אנו מגיעים למסקנה כי
(א) השפה D אינה מתקבלת ע"י אף מכונת טיורינג.

נניח עתה בשלילה כי ק"ימת מ"ט M_A המכריעה את A_{TM}. נראה כי במקרה זה ק"ימת מכונת טיורינג המקבלת (ואפילו מכריעה) את D. הסתירה שנקבל (!) תראה כי אין כזאת.

נבנה M_D בעזרת M_A:

1. בהנתן w, נמצא תחילה אינדקס i, כך ש $w = w_i$. הדבר ניתן לעשות, לדוגמא, ע"י מעבר על המחרוזות בסדר לקסי', תוך כדי העלאת COUNTER, עד אשר נגיע ל w.

יש כמובן דרכים הרבה יותר מהירות לעשות את זה!

2. נמצא את $\langle M_i \rangle$ (שוב, לדוגמא, ע"י מעבר על המחרוזות בסדר לקסי', ...).
3. נפעיל את M_A על קלט $\langle M_i \rangle, \langle M_i \rangle$. אם התשובה "לא" נחזיר "כן", ולהפך.

אכן בנינו M_D, בסתירה לכך שזה לא אפשרי.
מסקנה: M_A לא ק"ימת.
מ.ש.ל.

* הוכחה שניה לאי-כריעות של A_{TM}:

תהי

$$L = \{\langle M\rangle \mid M(\langle M\rangle) \neq accept\}$$

אם A_{TM} הייתה כריעה, L, כמובן, גם הייתה כריעה, והיתה קיימת מ"ט M_L המכריעה אותה. אך קיום M_L כזאת הינו פרדוקסלי, בדומה ל"ספר הכפרי":

* בכפר מסוים, הספר המקומי מגלח רק את אלה אשר אינם מגלחים את עצמם.

מה יעשה הספר עם עצמו? אם הוא מגלח את עצמו, הוא לא אמור לגלח את עצמו, ואם הוא לא מגלח את עצמו, הוא אמור לגלח את עצמו...

$\Rightarrow$ ספר כזה לא ק"ים.

כך גם המצב עם M_L: הרי הוא מקבלת רק את אלה הדוחים את הקידוד של עצמם... אם M_L מקבלת את $\langle M_L\rangle$, היא אמורה לדחות את $\langle M_L\rangle$, ואם M_L לא מקבלת את $\langle M_L\rangle$, היא אמורה לקבל את $\langle M_L\rangle$. לכן יש סתירה בעצם הגדרת M_L, ואין היא ק"מת כלל. לכן, כיוון ש: קיום M_A $\Leftarrow$ קיום M_L, אנו מסיקים כי M_A גם לא ק"מת. מ.ש.ל.

שים לב: L מוגדרת היטב, ואין שום בעיה איתה.

הערה: נראה כי $\overline{A_{TM}}$ לא רק לא ניתנת להכרעה, אך אפילו לא ניתנת לקבלה! נשים לב כי

$$\overline{A_{TM}} = \{ \text{זבל} \} \cup \{(\langle M\rangle, I) \mid M(I) \neq accept\}$$

מחרוזות שלא מקודדות דבר

החלק המעניין

כוון ש"הזבל" ניתן להכריע (יש לנו דרך לפענח מחרוזות ולהגיע למסקנה שהן אינן קידוד חוקי) $\overline{A_{TM}}$ לא ניתנת לקבלה $\Leftrightarrow$ החלק העיקרי שלה לא ניתן לקבלה

* <u>טענה</u>: אם L, $\overline{L}$ ניתנות לקבלה, אזי L ניתנת להכרעה.

הוכחה: נבנה מכונה M המכריעה באופן הבא. תהיינה M_L, $M_{\overline{L}}$ מכונות המקבלות L ו-$\overline{L}$ בהתאמה. בהינתן קלט w, נריץ את M_L ו-$M_{\overline{L}}$ <u>במקביל</u> על w. (כיוון שהמודל שלנו סדרתי, הכוונה ל"במקביל" היא צעד של M_L (עם שמירה) — צעד של $M_{\overline{L}}$ — וכך הלאה). אם $w \in L$, M_L תעצור ותקבל אותו. אם $w \notin L$, $M_{\overline{L}}$ תעצור ותקבל אותו. בכל מקרה, נקבל תשובה נכונה. מ.ש.ל.

החשיבות של הריצה במקביל היא שאם, לדוגמא, $w \notin L$ M_L עלולה לא לעצור

⊛ מסקנה: $\overline{A_{TM}}$ איננה ניתנת לקבלה.

אכן, אם $\overline{A_{TM}}$ הייתה ניתנת לקבלה, אזי, כיוון ש-A_{TM} ניתנת לקבלה, היינו מקבלים כי A_{TM} ניתנת להכרעה. אך אנו יודעים כי הדבר אינו כך. מ.ש.ל.

עתה, כאשר הכרנו כי ד"מ בעיה קונקרטית אשר איננה כריעה, נוכל בעזרתה להראות שהרבה בעיות אחרות הן גם כאלה. שיטת הפעולה תהיה כדלהלן.
נניח כי הכרנו כי A איננה כריעה; אם נראה כי B "יותר קשה" מ-A, המסקנה המיידית תהיה כי B גם כזאת.
"יותר קשה" אומר כי בהינתן מכונה M_B המכריעה את B (שאם היא ד"מ או לא) נוכל לבנות מכונה M_A המכריעה את A.

בעיית העצירה היא המעניינת בבעיות הנ"ל:

$$HALT_{TM} = \{ (\langle M \rangle, w) \mid M \text{ עוצרת על } w \}.$$

* משפט: בעיית העצירה ניתנת לזיהוי, אך לא ניתנת להכרעה.

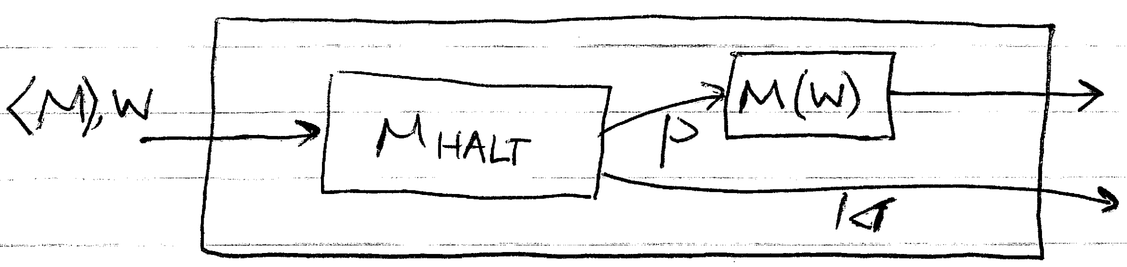
הוכחה: ניתנת לזיהוי כיוון ש $UTM(\langle M \rangle, w)$ עוצרת אם ורק אם M עוצרת על w. לכן, כל מה שעלינו לעשות זה להריץ את UTM על $(\langle M \rangle, w)$, ולקבל אם היא עוצרת.

לא ניתנת להכרעה:

נניח בשלילה כי קיימת מ"ט M_{HALT} המכריעה את $HALT_{TM}$. אזי המכונה הבאה מכריעה את A_{TM}:

> על קלט $\langle M \rangle, w$
> 1. הרץ את M_{HALT} על $\langle M \rangle, w$;
> 2. אם התשובה היא "כן" הרץ את הסימולציה של M על w והחזר את התוצאה;
> אחרת, החזר "לא".

סכמטית,



M_{ATM} (המכריע של A_{TM})

המסקנה הוסקה: אם יש מכריע ל $HALT_{TM}$, אזי יש גם מכריע ל A_{TM}, בסתירה למה שאנו יודעים. ☒

כמסקנה נקבל משפט Gödel.

המשפט הזה טוען כי קיימים משפטים מתמטיים נכונים, אשר אין להם הוכחה פורמלית (המתחילה מאקסיומות, משתמשת בחוקי גזירה לוגיים, ולבסוף מסיקה את הטענה).

לא נכנס לדיון כללי העדין בהגדרת מתי ראוי לקרוא למשפט נכון, אלא נתבונן במשפטים מהסוג הבא:

"בהנתן $\langle M_0\rangle$ (קידוד של מ"ט מסוימת) ו-w_0 (מחרוזת בינארית מסוימת), M_0 לא עוצרת על w_0."

ברור (?) שניתן לייחס לטענה מסוג זה ערך true/false, ויש עבור M_0, w_0 מסוימים היא אכן נכונה.

משפט: יש משפטים נכונים מהסוג הנ"ל, אשר אין להם הוכחה פורמלית.

הוכחה: נתבונן בשפה

$$PROVABLE = \{(\langle M\rangle, w) \mid \text{יש הוכחה פורמלית לכך ש-}M\text{ לא עוצרת על }w\}$$

(*) $PROVABLE \subseteq NON\text{-}HALT = \{(\langle M\rangle, w) \mid M \text{ לא עוצרת על } w\}$

לא ניתן להוכיח דבר לא נכון

(**) PROVABLE קבילה: פשוט נעבור על כל הנימוקים הפורמליים, עד אשר נגיע לאחד המוכיח את הטענה.

(***) NON-HALT איננה קבילה. אכן, לפי מה שידוע שכבר ראינו, $\overline{HALT}$ איננה קבילה. אך

$$\overline{HALT} = \{\text{קידוד לקוי}\} \cup NON\text{-}HALT.$$

כיון שבדיקת קידוד לקוי ניתן לעשות באופן יעיל (לכתוב תוכנית שתמיד עוצרת ועונה "כן"/"לא"), NON-HALT איננה קבילה.

שים לב: A,B קבילות $\Downarrow$ $A \cup B$ קבילה

(****) לכן NON-HALT מכילה זוגות $\langle M\rangle, w$ כך ש-M איננו עוצר על w, אך אין שום הוכחה פורמלית לכך. מ.ש.ל.

עבור לבעיה הבאה:

תהי A תכונה של שפות (כגון, L רגולרית, L ניתנת לקבלה, L מכילה מחרוזת אחת פי 5 ארוכה מהאחרת, (שונה) L לא רגולרית). יתר על כן, נניח כי A לא טריביאלית במובן הבא: קיימת מ"ט M^+ כך של-L_{M^+} יש תכונה A, וקיימת מ"ט M^- כך של-L_{M^-} אין תכונה A.

* **משפט:** הבעיה

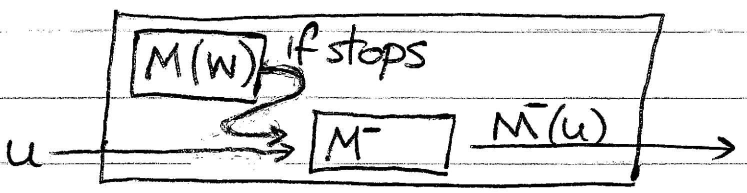
$$PROPERTY_A = \{\langle M\rangle \mid \text{יש תכונה } A \text{ ל-} L_M\}$$

אינה כריעה.

הוכחה: נטפל בנפרד בשני מקרים: **א** לשפה $\varnothing$ יש תכונה A **ב** לשפה $\varnothing$ אין תכונה A.

בכל מקרה, נניח בשלילה כי קיימת מ"ט מכונה M_A המכריעה את $PROPERTY_A$.

א. נתבונן במכונה $N_{M,w}$ הבאה:



ברור כי

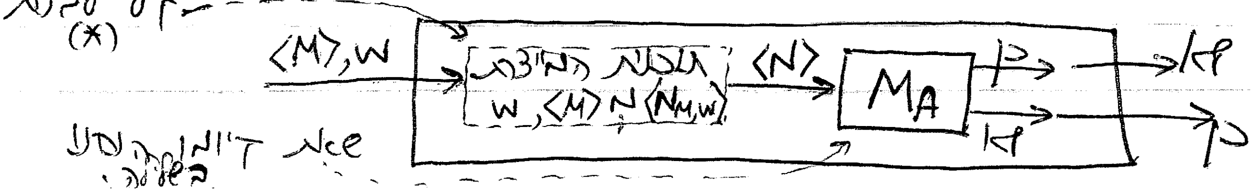
(קל!) (*) ניתן לבנות $\langle N\rangle$ בהינתן $\langle M\rangle, w$;

(**) ל-L_N יש תכונה A אם"ם M לא עוצרת על w.

אכן, אם M עוצרת על w, $L_N=\varnothing$ (מקיים את A)

ואם M עוצרת על w, $L_N=L_{M^-}$ (לא מקיים A).

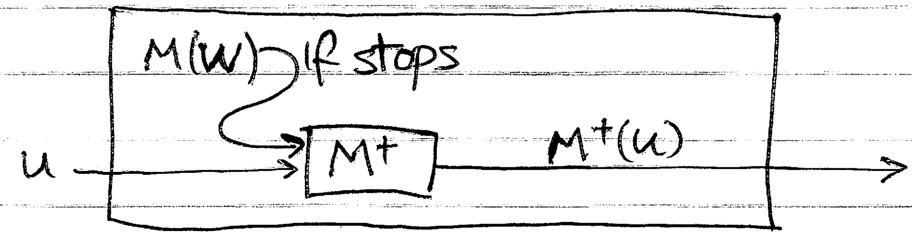
לכן, המכונה הבאה מכריעה את HALT:



סתירה

(63)

ב. במקרה זה $N_{M,w}$ תפעל כך:



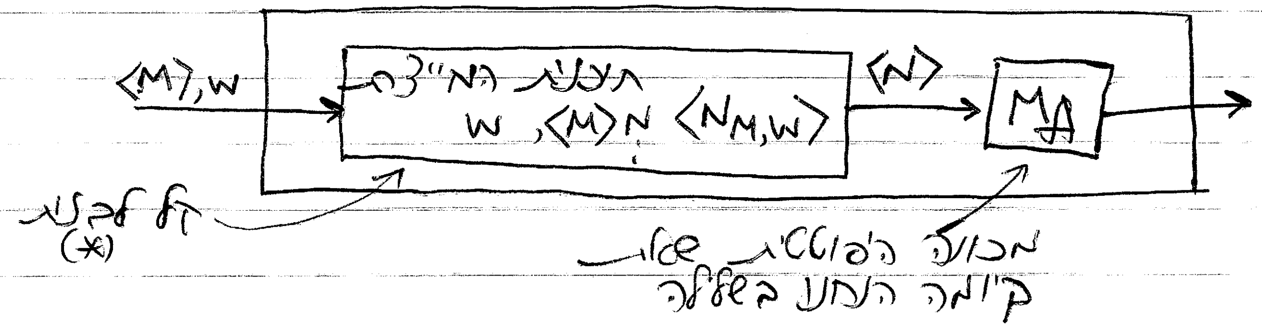
כמו קודם,

(*) ניתן לבנות $\langle N\rangle$ בהינתן $\langle M\rangle, w$;

(**) השפה של N הינה בעלת תכונה A אמ"ם M(w) עוצרת

אכן, אם היא עוצרת, $L_N = L_{M^+}$, יש תכונה; אם היא לא עוצרת, $L_N = \emptyset$, אין תכונה.

לכן המכונה הבאה מכריעה את HALT:



סתירה.

המטרה הבאה שלנו היא להראות כי בעיות הלא ניתנות להכרעה צצות לא רק בקונטקסט של מ"ט (המודל האולטימטיבי, ולכן קשה להכריע), אלא כבר בקונטקסט של שפות ח"ה (דקדוקים ח"ה = אוטומטי מחסנית). בפרט, בהינתן שני דקדוקים ח"ה G_1, G_2 נראה כי הבעיות הבאות הן בלתי כריעות:

(*) האם $L_{G_1} = \Sigma_1^*$?

(**) האם $L_{G_1} \cap L_{G_2}$ ריק?

(50) נמשיך עם כמה הגדרות.

תהי M מכונת טיורינג. ויהי w קלט שלה. נניח בה"כ כי $Q \cap \Gamma$ זרים. לכל רגע $i=0,1,2,\dots$ נתאים מחרוזת $s_i \in (Q\cup\Gamma)^*$ המתארת את מצב הסרט ברגע i (עד המקום שם מתחילים רווחים). במקום בו נמצא הראש, נרשום את מצב הראש.

כך, לדוגמה, $s_0 = q_0 \underbrace{w_1 w_2 \dots w_n}_{w}$.

נניח כי M עוצרת על w; נתבונן במחרוזת חדשה

$$s = s_0 \underset{\text{סימן הפרדה}}{\square} \overset{\text{reverse}}{s_1^R} \square s_2 \square s_3^R \dots \square \underset{\text{כאן עצר החישוב}}{s_k^R} \in (Q\cup\Gamma\cup\{\square\})^*.$$

*אבחנה: בהינתן שתי מחרוזות סמוכות, $v\square u^R$ (או $v^R\square u$) קיים אוטומט מחסנית הבודק כי אם ברגע מסוים מצב הסרט הוא v, ברגע הבא הוא חייב להיות u. כל מה שצריך לעשות זה להכניס v למחסנית, להשוות את v עם u עד אשר פוגשים שוני בלשון, לוודא שבמקום הזה הראש זז כפי שהוא אומר (בהתאם להגדרת M), ולבסוף, לוודא שהחל מהמקום הזה המחרוזות שוב זהות.

$(\{\square\}\cup\Gamma\cup Q)^* \supseteq L = \{u_1 \square u_2^R \square \dots u_t^R \mid$ כל המעברים הם M-חוקיים$\}$ ← מוגדר ע"י $\langle M\rangle, w$

—״— $\supseteq L_1 = \{u_1 \square u_2^R \square \dots u_t^R \mid$ כל המעברים האי-זוגיים הם חוקיים$\}$

—״— $\supseteq L_2 = \{u_1 \square u_2^R \square \dots u_t^R \mid$ כל המעברים הזוגיים הם חוקיים$\}$

(51)

*** מתדיים:**

1. $L = L_1 \cap L_2$ (ברור).

2. L_1, L_2 הן ח"ה. אכן, לדוגמא עבור L_1, אוטומט המחסנית יבדוק את u_1 vs. u_2, אחרי זה את u_3 vs. u_4, וכך הלאה.

כפי שתארנו קודם

3. $\overline{L}$ היא ח"ה. אכן, במקרה זה אוטומט המחסנית ינחש את הזוג השגוי (תוך כדי ספירת הזוגות, כדי לדעת האם זה $u\Box v^R$ או $v\Box u^R$), יבדוק אותו, וימצא את השגיאה.

*** אבחנה אחרונה:**

נגדיר שתי שפות רגולריות מעל $(Q\cup\Gamma\cup\{\Box\})^*$ באופן הבא:

$R_1 = \{$ כל המחרוזות כך שלפני $\Box$ הראשון עומד $q_0w_1\dots w_n$ $\}$

$R_2 = \{$ כל המחרוזות כך שאחרי ה-$\Box$ האחרון מופיע סימן בודד של Q, והוא מצב עוצר $\}$

לאור הדיון הקודם, נסכם את המסקנות שלנו:

1. $R_1 \cap L_1 \cap L_2 \cap R_2$ היא שפה המכילה את תיאור החישוב של M על w אם הוא עוצר, וריקה אחרת.

2. $\overline{L} \cup \overline{R_1} \cup \overline{R_2}$ היא שפה ח"ה השווה ל-Σ^* אם M לא עוצרת על w, ושווה ל-$\Sigma^* \setminus \{$תיאור החישוב של M על $w\}$ אחרת.

מחרוזת בודדת

3. בהנתן ⟨M⟩, w ניתן לבנות באופן יעיל את הדקדוקים של $L\cap R_1$, $L_2\cap R_2$, $\overline{L\cup R_1\cup R_2}$.
הבניה מתחילה מבניית האוטומט (כפי שאמרנו) בונה אוטומט שני הלוקח בחשבון את חיתוך/איחוד עם השפה הרגולרית, ולבסוף יוצרת דקדוק מתאים.

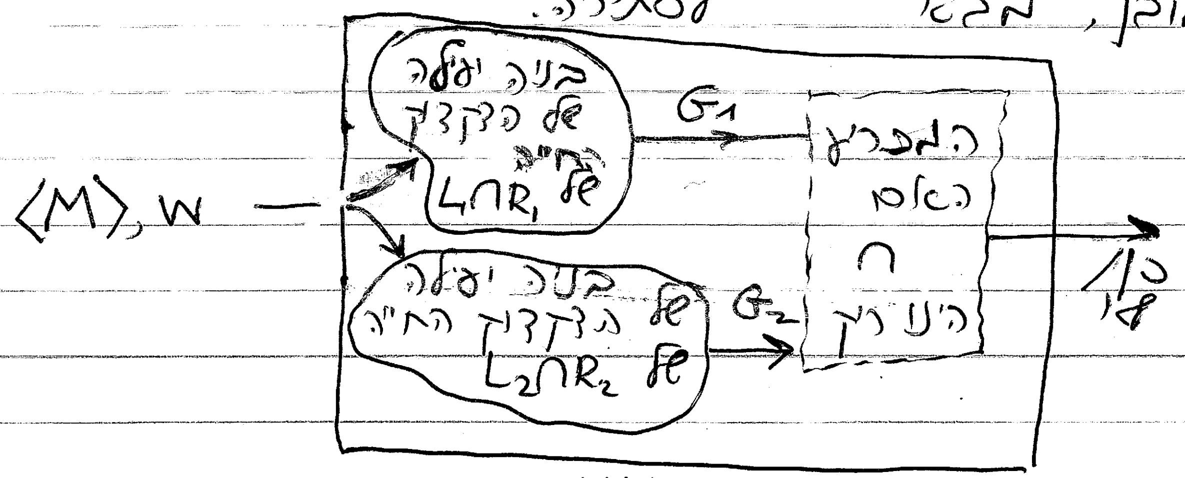
משפט

א. בהנתן שני דקדוקים חסרי הקשר G_1, G_2, לא ניתן להכריע האם $L_{G_1}\cap L_{G_2}=\emptyset$.

ב. בהנתן דקדוק חסר הקשר G, לא ניתן להכריע האם $L_G=\Sigma^*$.

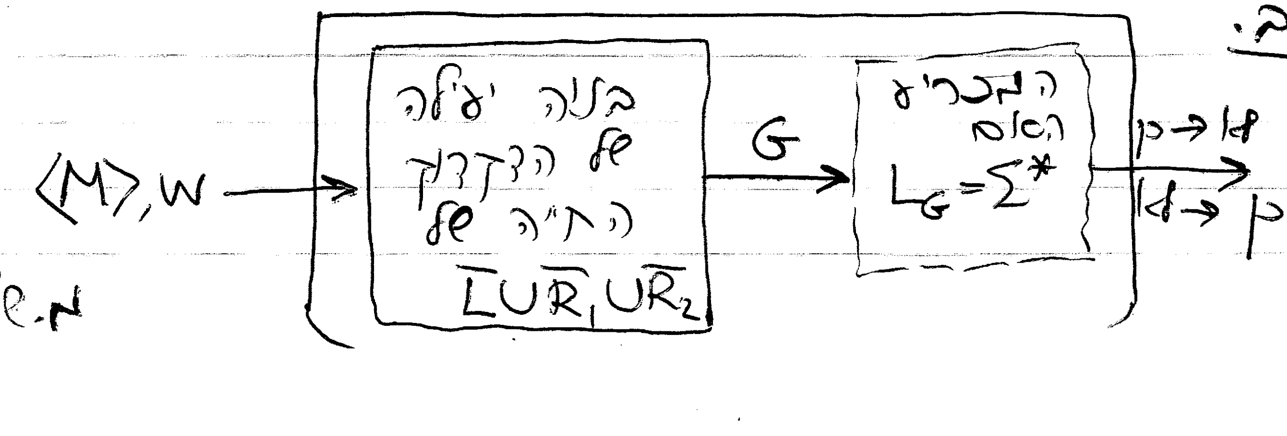
הוכחה: נראה כי בשני המקרים, לו היה ניתן להכריע את הבעיה, היה ניתן להכריע את בעית העצירה. זה, כמובן, מביא לסתירה.

א.



NOT HALT

ב.



מ.ש.ל.

נס"ם את הפרק הזה עם דוגמא נוספת של בעיה מעניינת אשר לא ניתנת להכרעה.

תהי w מחרוזת בינארית מסוימת. נגדיר $C(w)$ כאורך התוכנית הקצרה ביותר בשפת C אשר על קלט ε מדפיסה את w ועוצרת.

$C(w)$ קשורה ל-$K(w)$, סיבוכיות תיאור שלו, אך לא זהה לה.

ברור כי עבור w כל w, $C(w) \leq |w| + c_0$ (מדוע?)

קבוע אוניברסלי

ואכן, עבור מחרוזת מקרית, $C(w) \sim |w|$. אבל עבור w בעלת מבנה, $C(w)$ באופן טיפוסי קטנה בהרבה מ-$|w|$. כך, לדוגמא,

$C(\underbrace{0^{2^{2^{\cdot^{\cdot^{2}}}}}}_{n \text{ פעמים}}) \leq \log_2 n + c_1$ (מדוע?)

משפט: לא ק"מת תוכנית P בשפת C (=מ"ט) אשר בהינתן w יודעת לחשב את $C(w)$.

<u>הוכחה:</u> "הגרעין" של ההוכחה הוא פרדוקס הבא:

"המספר הקטן ביותר שלא ניתן לתאר בפחות מעשרה מלים."

לכאורה, המשפט הנ"ל מגדיר מספר כלשהו, אך אם ההגדרה הזאת היתה תקינה, המספר הנ"ל היה ניתן לתיאור בתשע מלים.

נניח בשלילה כי ק"מת תוכנית P כפי שמופיע בטענת המשפט, ואורכה k. תהי x_n המחרוזת הבינארית הקצרה ביותר כך ש $n < C(x_n)$.

אנו נראה כי הנחה שלנו מביאה למסקנה

כי ק"מ תוכנית Q באורך $\overset{\text{קבוע}}{c}+k+\log_2 n$, (53)
אשר על קלט ε מדפיסה x_n ועוצרת. עבור n מספיק
גדול, זה מביא לסתירה (עם הגדרה של x_n).
בפסאודו-קוד, Q נראית כך:

```
Q:
{
   c←0;   s←ε;
   while (c≤n)                    (* n: בבינארי *)
   {
      s← next(s);                 (* בסדר הלקסיקוגרפי *)
      c← P(s)
   }
   print(s).
}
```

ברור כי האורך של Q הוא $\log_2 n + k + c$

n דיוק של   P אורך   Δ השאר

מ.ש.ל.