גירסה 1.04 – 14.7.2010

תורת החישוביות – חלק 1

ניר אדר

הבהרה: מסמך זה מסתמך בין היתר על הקורס "תורת החישוביות" בטכניון, אך איננו חומר רשמי של הקורס, אלא סיכום אישי בלבד. רשימת המקורות נמצאת בסוף המסמך. אתר הקורס "תורת החישוביות": http://webcourse.cs.technion.ac.il/236343

תוכן עניינים

1. מבוא

1.1. בעיות לא פתירות והנושאים במסמך

<u>בעיות לא פתירות</u>:

- בהינתן שתי תוכניות מחשב, האם הן שקולות? (בהינתן קלט זהה מפיקות פלט זהה)
- בהנתן תוכנית M וקלט X, האם ניתן לבדוק האם M עוצרת על X.
- בהנתן תוכנית M, האם ניתן לבדוק אם היא עוצרת לכל קלט?

הגדרה: **תוצאות שליליות נסיבתיות** הינו אוסף בעיות המקיים ש: או לכל הבעיות בקבוצה קיים פתרון יעיל, או שלאף אחת מהן אין פתרון כזה.

על מנת לעסוק בנושאים אלו נגדיר מודל, שהוא מכונת טיורינג. מסמך זה והמסמך בנושא חישוביות יציגו אספקטים שונים לגבי מכונות טיורינג והשימושים שלהן.

<u>הנושאים בהם יעסוק מסמך זה</u>:

במהלך קריאת המסמך קל ללכת לאיבוד בין המוני הפרטים, ולאבד את התמונה הגדולה– מה אנחנו מנסים להראות, לאן אנחנו הולכים. הנושאים בהם עוסק מסמך זה:

- הצגת מכונת טיורינג כמודל חישוב כללי.
- התזה של צ'רץ: כל מודל "סביר" ו-"כללי" של חישוב שקול למ"ט (נציג דוגמאות למספר מודלים ספציפיים).
- נציג את נושא המכונה האוניברסילית.
- מה ניתן לחשב על ידי מכונות טיורינג ומה לא ניתן לחשב.
- הרדוקציה ושימושיה.

1.2. מכונת טיורינג

מ"ט היא השביעיה $M=\left(Q,q_0,F,\Gamma,\flat,\Sigma,\delta\right)$ כאשר:

Q: קבוצה סופית של **מצבים**.

$q_0\in Q$: **מצב התחלתי**.

$F\subseteq Q$: קבוצת **מצבים סופיים**.

Σ: קבוצה סופית של אותיות המכונה **א"ב הקלט**.

Γ: קבוצה סופית של אותיות המכונה **א"ב עבודה**. מתקיים כי $\Sigma\subseteq\Gamma$.

$\flat$: **סימן רווח** המקיים $\flat\in\Gamma\setminus\Sigma$.

δ: **פונקצית המעברים**: $\delta:\left(\left(Q\setminus F\right)\times\Gamma\right)\to\left(Q\times\Gamma\times\left\{L,R,S\right\}\right)$. $\left\{L,R,S\right\}$ הם סימנים המייצגים את כיוון תנועת הראש הקורא (שמאלה, ימינה או עמידה במקום). דוגמא:

$\delta\left(q_1,0\right)=\left(q_2,1,R\right)$ משמעותו: "אם במצב q_1 רואים 0, אז עוברים למצב q_2, כותבים 1 במקום עליו מצביע הראש, ומזיזים את הראש תא אחד ימינה".

תיאור המכונה: למכונה סרט חצי-אינסופי לכוון ימין, ראש קורא/כותב שיכול בכל צעד חישוב לקרוא אות אחת, לכתוב אות אחת במקומה, ולנוע צעד אחד באחד הכיוונים, או להשאר במקום.

המכונה מתחילה תמיד את החישוב על קלט $x\in\Sigma^*$ במצב הבקרה q_0 כאשר הראש מצביע לתא השמאלי ביותר בסרט. בקצה השמאלי של הסרט רשומה מילת קלט x, ואחריה אינסוף $\flat$-ים.

הריצה מתבצעת בצעדי חישוב על פי פונקצית המעברים δ. בכל צעד חישוב המכונה מבצעת פעולה התלויה במצב הבקרה בו היא נמצאת ובתוכן התא שעליו מצביע הראש.

במקרה בו הראש מצביע על התא השמאלי ביותר בקלט, ובפונקצית המעברים מסומן לנוע שמאלה, הראש יישאר בצעד הבא במקומו. הריצה על הקלט x מסתיימת כאשר המכונה נכנסת למצב סופי (מצב השייך לקבוצה F). במקרה הנ"ל נאמר כי המכונה **עוצרת על** x**.** לא מובטח כי חישוב המכונה יעצור לכל קלט!

בכל מקרה שהמכונה עוצרת, **הפלט** מוגדר כמחרוזת שמשמאל לראש בעת העצירה (לא כולל את התו עליו מצביע הראש).

קונפיגורציה רגעית של מכונת טיורינג היא שלישיה $c \in \Gamma^* \times Q \times \mathbb{N}$. השלישיה $c=(\alpha,q,i)$ מתארת את מצב המכונה ברגע מסויים במהלך החישוב שלה: α מתאר את תוכן הסרט, q מתאר את מצב הבקרה, ו-i מציין את מיקום הראש (אינדקס התא עליו מצביע הראש). המחרוזת α מתארת את תוכן הסרט מהקצה השמאלי עד המקום האחרון שאינו $\flat$.

הקונפיגורציה ההתחלתית של מכונת טיורינג על קלט $x \in \Sigma^*$ היא $(x,q_0,0)$.

קונפיגורציה סופית של מ"ט היא קונפיגורציה $(\beta\gamma,q,i)$ כאשר $\beta,\gamma \in \Gamma^*, q \in F$ וגם $|\beta| = i$.

במקרה זה נאמר כי β הוא **הפלט** המתאים לקונפיגורציה סופית זו.

צעד חישוב של מ"ט הוא מעבר מקונפיגורציה רגעית אחת לקונפיגורציה רגעית שניה, לפי פונקצית המעברים δ. תהי מכונה M הנמצאת בקונפיגורציה $c=(\alpha,q,i)$. נניח כי $\alpha_i = a$ וכי $\delta(q,a)=(p,b,d \in \{L,R,S\})$, אז המכונה מבצעת את הצעדים הבאים:

1. כותבת את האות b במקום האות a.
2. משנה את מצב הבקרה מ-q ל-p.
3. מזיזה את הראש בהתאם לאות: S=לא מזיזה, R=למקום $i+1$, L=למקום $i-1$. (אם $i=1$ לא זזים).

קונפיגורציה עוקבת: הקונפיגורציה העוקבת c' של הקונפיגורציה הנוכחית c תוגדר באופן יחיד על ידי <u>ביצוע צעד חישוב</u> על הקונפיגורציה הנוכחית. נסמן: $c \overset{1}{\underset{M}{\vdash}} c'$.

חישוב פונקציות: בהינתן מ"ט M, הפונקציה f_M שהמכונה M מחשבת היא פונקציה (חלקית או מלאה) מ-Σ^* ל-Γ^*, המוגדרת רק על קלטים שעליהם החישוב של M עוצר. ערך הפונקציה f_M על קלטים כאלו הוא **הפלט** שמכונה מוציאה בחישוב, כלומר לכל קלט $x \in \Sigma^*$, אם M עוצרת על x עם פלט y, אזי $f_M(x)=y$, ואחרת f_M אינה מוגדרת על המחרוזת x.

<u>הערה</u>: כאשר מגדירים פונקצית מעברים δ, גם אם יש מצבים אליהם לא נגיע, חובה להגדיר אותם (באופן שרירותי כלשהו כרצוננו) מכיוון ש-δ היא לפי הגדרתה פונקציה מלאה.

2. שקילות של מודלים

<u>הגדרה</u>: **מודל** של חישוב הוא אוסף של אובייקטים שלכל אחד מהם מתאימה פונקציה f שהוא מחשב.

שני מודלים יקראו **שקולים** אם אוסף הפונקציות שהם מחשבים זהה.

2.1. דוגמאות

<u>הגדרה</u>: **מכונת טיורינג חסרת מנוחה** היא מכונה שהגדרתה זהה לזו של מכונת טיורינג, מלבד פונקצית המעברים שתוגדר כך: $\delta:(Q\setminus F)\times\Gamma\to\left(Q\times\Gamma\times\{L,R\}\right)$.

כלומר, מכונת טיורינג חסרת מנוח תזוז ימינה או שמאלה בכל שלב, ולא ייתכן שהיא תשאר במקומה.

<u>טענה</u>: מודל מ"ט חסרת מנוחה שקול למודל מ"ט.

סקיצת הוכחה: ברור שקבוצת המ"ט חסרות המנוחה היא חלקית לקבוצת המ"ט.

נראה כי כל מ"ט ניתנת לייצוג כמ"ט חסרת מנוחה כך: תהי M מ"ט. עבור כל צעד מהצורה $\delta_1(q,a)=(p,b,L)$ או מהצורה $\delta_1(q,a)=(p,b,R)$, יתקיים $\delta_2(q,a)=\delta_1(q,a)$. עבור צעדים מהצורה $\delta_1(q,a)=(p,b,S)$ נגדיר שני מצבים: $\delta_2(q,a)=(p',b,R)$ וכן $\forall c\in\Gamma$ נגדיר $\delta_2(p',c)=(p,c,L)$. כלומר, המכונה תעבור על ידי מעבר ימינה למצב עזר אותו אנחנו מגדירים, ומיד לאחר מכן תבצע צעד שמאלה מבלי לשנות את הקלט.

ניתן להוכיח באינדוקציה כי בנייה זו אכן שקולה למ"ט.

<u>הגדרה</u>: מודל מ"ט עם k סרטים הינו מודל הזהה למ"ט, מלבד פונקצית המעברים שתוגדר $\delta:(Q\setminus F)\times\Gamma^k\to Q\times\Gamma^k\times\{L,R,S\}^k$. מכונה זו הינה בעלת k ראשים קוראים-כותבים.

<u>טענה</u>: לכל $k\geq 1$, מודל מ"ט עם k סרטים שקול למודל מ"ט רגיל.

מודלים שאינם שקולים למכונת טיורינג

- מודלים חלשים יותר: אוטומט סופי, אוטומט מחסנית.
- מודלים עם אלמנטים אינסופיים – למשל מ"ט עם ∞ מצבים.

התזה של Church: כל מודל כללי וסביר של מ"ט שקול בכוחו לכוחה של מ"ט.
כללי: מודל חזק לפחות כמו מ"ט. סביר: מודל שבו לכל אובייקט תיאור סופי.

2.2. שיטת ההוכחה של שקילות

השיטה: בהנתן מ"ט M ממודל A, נבנה מ"ט M' ממודל B המחשבת את אותה הפונקציה, ולהיפך.
הערות:

- אסור להסתמך בבעיה על תכונות ספציפיות של M. הבנייה צריכה להיות כללית לכל M מהמודל.
- קיימים מודלים חזקים יותר מהמודל הסטנדרטי של מ"ט וקיימים מודלים חלשים יותר.
- בד"כ הבניה שנבנה תממש סימולציה צעד אחד צעד.
- **דגש חשוב**: אנחנו בודקים האם המודלים מקבלים את אותה קבוצת שפות. אנחנו לא בודקים האם קבוצת המכונות שקיימות בקבוצה אחת שווה לקבוצת המכונות שבקבוצה השניה.

1.2.2. תרגיל

הוכח כי לכל מ"ט M קיימת מ"ט M' המחשבת את אותה פונקציה, אשר אינה מנסה ליפול מהסרט (על ידי נסיון ללכת שמאלה כאשר הראש נמצא בתא השמאלי ביותר בסרט).

פתרון:
נגדים בניה של מכונה אשר לא מנסה ליפול מהסרט:
נגדיר Σ', Γ' בצורה הבאה: שעבור כל אות σ ב-Σ או ב-Γ נגדיר אות מקבילה σ'.

האלגוריתם:
<u>אתחול:</u> המכונה תחליף את האות הראשונה σ באות המקבילה σ'.
<u>ריצה:</u> במהלך החישוב, בכל פעם שהמכונה נתקלת באות השייכת ל-Γ', היא תדע שהיא נמצאת בתא הראשון בסרט, ולכן היא תפעל על פי הכללים הבאים:

1. אם במקור M הייתה אמורה ללכת שמאלה, אז M' תישאר באותו המקום (ולא תפנה שמאלה).
2. אם M אמורה להחליף את האות σ_1 באות σ_2 אז M' תחליף את σ_1' באות σ_2'.

<u>סיום:</u>

אם M הגיעה למצב מקבל אז M' צריכה לסמן את המקום ב-$, לנוע שמאלה עד התחלת הסרט, שם היא תמצא אות השייכת ל-Γ'. את אות זו היא תחליף אותה באות המתאימה מ-Γ, תנוע חזרה ימינה עד לסימן ה-$ ותעצור.

בהינתן $M=(Q,q_0,F,\Gamma,\flat,\Sigma,\delta)$, נגדיר את M' כך:

$$M'=\left(Q\cup\{q_{02},q_f,q_R,q_L\},q_{02},\{q_f\},\Gamma\cup\Gamma'\cup\{\$\},\Sigma\cup\Sigma',b,\delta'\right)$$

פונקצית המעברים δ':

- מעבר התחלתי: $\delta'(q_{02},\sigma\in\Gamma)=(q_0,\sigma',S)$
- $\delta'(q\in Q,\sigma\in\Gamma)=\delta(q,\sigma)$ - אם לא נמצאים על האות הראשונה, מבצעים את מה שביצעה המכונה המקורית.
- אם $\delta(q\in Q\setminus F,a)=(p,b,X)$ כך ש $X\in\{L,S\}$ אז

$\delta'(q,a'\in\Gamma')=(p,b'\in\Gamma',S)$ - אם נמצאים באות הראשונה, והתנועה המקורית הייתה שמאלה או עצירה, אז נשארים במקום. בכל מקרה, האות שכותבים שייכת ל-Γ'.

- אם $\delta(q,a)=(p,b,R)$ אז $\delta'(q,a'\in\Gamma')=(p,b'\in\Gamma',R)$ - אם נמצאים באות הראשונה, והתנועה המקורית הייתה ימינה, אז נעים ימינה, אבל האות שכותבים היא מ-Γ'.
- $\delta'(q\in F,\sigma\in\Gamma)=(q_L,\$,R)$ - אם המכונה הקודמת עצרה, ולא על התו הראשון, אז מחליפים את התו ב-$ ועוברים למצב שממנו מתחילים לנוע שמאלה.
- $\delta'(q_L,\sigma\in\Gamma)=(q_L,\sigma,L)$ - נעים שמאלה בלי לשנות כלום.
- $\delta'(q_L,\sigma'\in\Gamma')=(q_R,\sigma\in\Gamma,R)$ - מחליפים את האות מ-Γ' לאות מ-Γ ועוברים למצב שממנו מתחילים לנוע ימינה.
- $\delta'(q_R,\sigma\in\Gamma)=(q_R,\sigma,R)$ - נעים ימינה בלי לשנות כלום.
- $\delta'(q_R,\$)=(q_F,\$,S)$ - עצירה.
- $\delta'(q\in F,\sigma'\in\Gamma')=(q_f,\$,S)$ - אם החישוב המקורי נעצר על האות הראשונה, אז עוצרים באותו המקום, ומחזירים כפלט מילה ריקה.

2.3. מודל ה-RAM

נציג כעת מודל נוסף, מודל ה-RAM. נטען, אך לא נוכיח, כי מודל זה שקול למכונת טיורינג.

1. זיכרון אינסופי (בן מניה) הממוען על ידי הכתובות $0,\pm1,\pm2,...$. כל תא יכול להכיל כל מספר שלם ביצוג בינארי.
2. מספר סופי של רגיסטרים $r_1,r_2,...,r_n$. כל רגיסטר יכול להכיל כל מספר שלם ביצוג בינארי.
   מוגדרים שלושה רגיסטרים מיוחדים:
   r_1 - משמש כ-PC (Program Counter).
   r_2 - רגיסטר הקלט.
   r_3 - רגיסטר הפלט.
3. <u>סט הפקודות</u>: סט הפקודות במודל מכיל מספר סופי של פקודות "אסמבלר" רגילות.**דוגמא** (!) לסט פקודות אפשרי:
   - $add(r_i,r_j)$ - $r_i=r_i+r_j$.
   - $\mathrm{load}\ r_i \leftarrow (r_j)+$ - הכנס ל-r_i את תוכן התא המוצבע על ידי r_j ואחר כך הוסף אחד לרגיסטר r_j.
   - $\mathrm{store}\ r_i \rightarrow (r_j)$ - כתוב את תוכן r_i בתא המוצבע על ידי r_j.
   - $\mathrm{inc}\ r_i$ - הגדל את r_i ב-1.
   - $\mathrm{dec}\ r_i$ - הקטן את r_i ב-1.
   - $\mathrm{bgt}\ r_i ? r_j$ - אם r_i גדול מאפס, אזי $\mathrm{PC} \leftarrow r_j$.
   - halt - עצירת החישוב.
4. <u>צעד חישוב</u>:
   א. קרא את המספר שבכתובת המוצבעת על ידי ה-PC.
   ב. $\mathrm{PC} \leftarrow \mathrm{PC}+1$.
   ג. פענח את הפקודה שנקראה ובצע אותה.
   ד. חזור ל-א'.

5. <u>מהלך החישוב:</u>

א. מצב התחלתי:

- קוד התוכנית שתבוצע נמצא בזיכרון החל מכתובת 0. תוכן תאי הזיכרון האחרים הינו 0.
- r_2 מכיל את הקלט וכל שאר הרגיסטרים מאותחלים בערך 0.

ב. ביצוע צעדי חישוב עד שמתבצעת פקודת halt.

ג. אם המכונה עוצרת הפלט הינו התוכן של r_3 ברגע העצירה.

מעט כיוון על ההוכחה שמודל ה-RAM שקול למכונת טיורינג.

כיוון אחד: מדובר למעשה בשפת אסמבלר בסיסית – ניתן לבנות בקלות יחסית שפת תכנות שתממש מכונת טיורינג, וכך מוכיחים את כיוון זה.

כיוון שני: נשתמש במכונת טיורינג בעלת k סרטים ליצוג מודל ה-ram. יהיה סרט אחד עבור כל רגיסטר, שם ישמר ערכו. יהיה סרט שיהווה את זיכרון ה-RAM, וכן עוד שני רגיסטרים לכתובת וערך שבתוכה לצורך ביצוע חישובים. על ידי בניה מתאימה נראה גם את הכיוון הזה. כיוון זה הוא הקשה יותר להוכחה.

3. מכונה אוניברסלית

על מנת לייצג מ"ט באופן נוח, נרצה לקודד אותה בצורת מחרוזת מעל הא"ב $\{0,1\}$.

דרישות מקידוד של מכונת טיורינג:

- יהיה ניתן לשחזר את המכונה בהינתן הקידוד שלה.
- יהיה ניתן "להריץ" את המכונה בהינתן הקידוד שלה.

תהי $M=(Q,q_0,F,\Gamma,\flat,\Sigma,\delta)$. נציע את הקידוד הבא (קיימים עוד קידודים, הבחירה היא שרירותית).

$Q=\{1,2,3,\ldots,|Q|\},q_0=1,F=\{2,3\},\Sigma=\{1,2,\ldots,|\Sigma|\},\Gamma=\{1,2,\ldots,|\Gamma|\},\flat=|\Gamma|$

הכיוונים יקודדו: $\{L,R,S\}=\{1,2,3\}$.

קידוד של מעברים: יהי אחד המעברים, $\delta(q,a)=(p,b,d)$. נקודד את המעבר על ידי המחרוזת הבינארית הבאה: $1^q01^a01^p01^b01^d$.

קידוד של המכונה M למחרוזת בינארית יהיה: $1^{|Q|}01^{|\Gamma|}01^{|\Sigma|}00\langle\delta_1\rangle00\langle\delta_2\rangle00\ldots\langle\delta_m\rangle000$

כאשר כל $\langle\delta_i\rangle$ הוא קידוד של אחד המעברים, וכן $m=(|Q|-2)\cdot|\Gamma|$.

עבור מ"ט M, נסמן את המחרוזת הבינארית המתאימה לה ב-$\langle M\rangle$. בהינתן מחרוזת בינארית, קל לבדוק האם היא מהצורה $\langle M\rangle$. נגדיר כל מחרוזת בינארית שאיננה ניתנת לפירוש לפי הקונבנציה שהגדרנו, כקידוד של מכונה אותה נכנה בשם M_{stam}, שעוצרת מייד על כל קלט במצב סופי 2.

קידוד של קלט x מעל א"ב Σ הוא המחרוזת $\langle x\rangle=1^{x_1}01^{x_2}0\ldots01^{x_n}$.

קידוד של קונפיגורציה $c=(\alpha,q,i)$: $\langle c\rangle=1^{\alpha_1}01^{\alpha_2}0\ldots01^{\alpha_{i-1}}001^{\alpha_i}0\ldots01^{\alpha_m}00$

כלומר, רצף של שני אפסים הוא זה שמסמן את מיקום הראש.

נגדיר את **הפונקציה NEXT**:

$$NEXT\left(\langle M\rangle,\langle c\rangle\right)=\begin{cases}\langle c'\rangle, c\overset{1}{\underset{M}{\vdash}}c' & \text{the input is valid and also } c\notin F\\ \text{Undefined} & \text{Else}\end{cases}$$

הפונקציה מקבלת קידוד של מכונת טיורינג ושל קונפיגורציה רגעית, ומחזירה את הקידוד של הקונפיגורציה הרגעית העוקבת, אם קיימת כזו.

קלט של מספר ארגומנטים: עד כה דיברנו על מילת קלט אחת w עבור מכונת טיורינג. ניתן להרחיב את הגדרת הקלט למספר ארגומנטים כלשהו. נבצע זאת על ידי הוספת הסימן "," לשפה, ושימוש בו כדי להפריד בין מילים שונות.

<u>טענה</u>: הפונקציה NEXT ניתנת לחישוב. כלומר, קיימת מ"ט שעל מילת קלט $\langle m\rangle,\langle c\rangle$ תוציא את הפלט $NEXT\left(\langle m\rangle,\langle c\rangle\right)$.

הוכחה: נציג בניה של מכונה:

א. בדוק את חוקיות הקלט. בכל מקרה של אי חוקיות נכנס ללולאה אינסופית.

ב. אם $\langle c\rangle$ הינה סופית, נכנס ללולאה אינסופית.

ג. אם $\langle M\rangle$ מתאים ל- M_{stam}: אם $\langle c\rangle$ היא הקונפיגורציה ההתחלתית, אזי $\langle c'\rangle$ היא קונפיגורציה סופית מתאימה. אחרת אם $\langle c\rangle$ לא תחילית, נכנס ללולאה אינסופית (מצב לא חוקי).

ד. נניח ש- c חוקית ומתאימה ל- M, אזי נחפש את המצב בו נמצאת המכונה והאותה אותה היא קוראת על ידי חיפוש המחרוזת 00 בקלט $\langle c\rangle$. $...001^q01^a0...$. נחפש בקידוד $\langle M\rangle$ מה עושה המכונה בסיטואציה כזו (כלומר מהו $\delta\left(q,a\right)$) על ידי חיפוש המחרוזת 001^q01^a0 ב- $\langle M\rangle$. נוציא מתוך $\langle M\rangle$ את p,b,d : $1^p01^b01^d$ ונשנה את c בהתאם. נקבל $\langle c'\rangle$ ונוציא אותו כפלט.

<u>הגדרה</u>: **הפונקציה האוניברסלית**:

$$U\left(\langle M\rangle,\langle x\rangle\right)=\begin{cases}\langle f_M\left(x\right)\rangle & \text{If the input is valid and also } f_M\left(x\right) \text{ is defined}\\ \text{Undefined} & \text{Else}\end{cases}$$

כלומר בהינתן קידוד של מכונה וקידוד של קלט, הפונקציה האוניברסלית מחזירה את קידוד הפלט של המכונה, על אותה מילה.

<u>טענה</u>: הפונקציה האוניברסלית U ניתנת לחישוב ע"י מכונת טיורינג.

הוכחה: על קלט $\langle m\rangle,\langle x\rangle$:

א. בדיקת חוקיות של הקלט.

ב. אם $M=M_{stam}$ נוציא פלט ε.

ג. נבנה את החישוב של M על x.

a. $\langle c_0\rangle=00\underset{q_0}{\underbrace{1}}0\langle x\rangle$

b. כל עוד c הנוכחית איננה סופית: $\langle c\rangle=NEXT\left(\langle M\rangle,\langle C\rangle\right)$.

c. אם מגיעים לקונפיגורציה סופית, נוציא כפלט את תוכן $\langle c\rangle$ עד מיקום 00.

מתקיים: אם הקלט לא חוקי או אם M לא עוצרת על הקלט x, המכונה לא עוצרת. אם הקלט חוקי, ו- M עוצרת על הקלט x תוך t צעדים, אזי נמצא את הפלט על ידי t הפעולות של NEXT.

<u>הגדרה</u>: כל מ"ט שמחשבת את הפונקציה U נקראת **מכונת טיורינג אוניברסלית**.

<u>טענה</u>: קיימת מ"ט אוניברסלית.

4. בעיות הכרעה

<u>הגדרה</u>: **שפה** L מעל א"ב Σ היא קבוצה $L \subseteq \Sigma^*$.

<u>הגדרה</u>: **מ"ט לזיהוי שפות** או בשם נוסף, **מ"ט לקבלת שפות**, היא מ"ט רגילה שבה: מוגדר $F=\{q_A,q_R\}$ (A מסמל accept ו-R מסמן reject). נאמר שמ"ט M כנ"ל **מקבלת** קלט X, אם M עוצרת עבור הקלט X במצב q_A, ונאמר שמ"ט כנ"ל **דוחה** קלט X, אם המכונה M עוצרת עבור הקלט X במצב q_R.

<u>אבחנה</u>: ייתכן שהחישוב של M על קלט X אינו מסתיים, ולכן "לא מקבלת" אינו זהה ל"דוחה" וכן "לא דוחה" אינו זהה ל"מקבלת".

<u>הגדרה</u>: **השפה של מכונה** תוגדר כך: L(M) = {x|x מקבלת את M}.

<u>הגדרה</u>: אם M עוצרת לכל קלט, אומרים ש-M **מכריעה** את השפה L(M). כלומר, שלכל $x \in L$ המכונה M עוצרת ב-q_A ולכל $x \notin L$ המכונה עוצרת ב-q_R.

דוגמאות לשפות שניתנות להכרעה:

- Σ^*: מ"ט שבצעד הראשון, ללא תלות בקלט, עוברת ל-q_A, היא מ"ט המכריעה את Σ^*.
- Φ: מ"ט שבצעד הראשון, ללא תלות בקלט, עוברת ל-q_R, היא מ"ט המכריעה את Φ.

הגדרה: נקבע את הא"ב שלנו להיות $\Sigma=\{0,1\}$.

נגדיר את קבוצות השפות הבאות (**מחלקות השפות**):

- R={ L| ניתנת להכרעה L }.
- RE = {L| קיימת מ"ט שמקבלת את L }.
- $\text{CO-RE}=\{L \mid \overline{L} \in RE\}$ כאשר $\overline{L} \triangleq \Sigma^* \setminus L$. כלומר, קבוצת כל השפות עבורן קיימת מ"ט M שלכל קלט שאינו בשפה – היא דוחה, ולכל קלט שבשפה – היא מקבלת או לא עוצרת.

שפה L השייכת ל-R נקראת **שפה רקורסיבית**. שפה L השייכת ל-RE מכונה שפה **הניתנת למניה רקורסיבית**.

אבחנות: $\mathrm{R}=\mathrm{RE}\bigcap \text{CO-RE},\ \mathrm{R}\subseteq \text{CO-RE},\ \mathrm{R}\subseteq \mathrm{RE}$

תכונות של המחלקות:

1. R סגורה למשלים: $L\in \mathrm{R}\Rightarrow \overline{L}\in \mathrm{R}$ (נשים לב כי RE לא סגורה למשלים!).
2. R סגורה לאיחוד: $L_1,L_2\in \mathrm{R}\Rightarrow L_1\bigcup L_2\in \mathrm{R}$.
3. R ו-RE סגורות לחיתוך $L_1,L_2\in \mathrm{R}\Rightarrow L_1\bigcap L_2\in \mathrm{R}$.
4. R ו-RE סגורות לשרשור.

מבחר הוכחות:

R סגורה למשלים – תהי $L\in \mathrm{R}$, אזי קיימת מ"ט M המכריעה את L. נבנה מ"ט $\overline{M}$ עבור השפה $\overline{L}$ בצורה הבאה: $\overline{M}$ זהה ל-M, למעט: $\overline{q_R}=q_A,\ \overline{q_A}=q_R$. מתקיים כי $\overline{M}$ עוצרת תמיד, וכן כי $L\left(\overline{M}\right)=\overline{L}$. מכאן $\overline{M}$ מכריעה את $\overline{L}$, ולכן $\overline{L}\in \mathrm{R}$.

R סגורה לאיחוד – רעיון ההוכחה הוא לבנות מ"ט M בעלת 2 ראשים המכריעה את $L_1\bigcup L_2$. נעתיק את X מילת הקלט לסרט הראשון, ונפעיל עליה את M_1. אם M_1 מקבלת, נקבל. אחרת, נעתיק את X לסרט השני ונפעיל עליו את M_2. אם M_2 מקבלת, נקבל, אחרת – נדחה. מתקיים כי M עוצרת תמיד, וכי M מקבלת קלט X אמ"מ $X\in L_1\bigcup L_2$, ולכן $L_1\bigcup L_2\in \mathrm{R}$.

<u>טענה</u>

RE סגורה לאיחוד

רעיון ההוכחה דומה להוכחה לגבי R.

נשים לב שלא ניתן לבצע בדיוק את אותה הוכחה, מכיוון שחישוב מכונה לא מסתיים תמיד. לפיכך נעבוד עם מכונה בעלת שני סרטים, שתעשה צעד אחד על סרט 1 כמו M_1 ולאחריו צעד על סרט 2 כמו M_2. במידה ואחת המכונות עוצרת, אזי גם המכונה בעלת שני הסרטים תעצור.

ניתן להוכיח כי השפה אותה מקבלת מכונה זו היא איחוד שתי השפות.

4.1. דוגמאות לשפות במחלקות

שפות השייכות ל-R: Σ^*, $\varnothing$, שפות רגולריות.

שפות השייכות ל-RE:

- כל שפה שב-R.
- שפה של כל אוטומט.
- **שפת העצירה HP**, המוגדרת כך: $HP = \{\langle M\rangle,\langle x\rangle \mid M \text{ stops on input } x\}$
- **השפה האוניברסלית:** $L_u = \{\langle M\rangle,\langle x\rangle \mid M \text{ accepts on the input } x\}$
- **שפת האלכסון:** $L_D = \{\langle M\rangle \mid \langle M\rangle \in L(M)\}$. שפה זו היא שפת כל המכונות <u>המקבלות</u> את הייצוג של עצמן.

תרגיל: לאיזו מחלקה שייכת השפה הבאה: $L_1 = \{\langle M\rangle \mid L(M) \in RE\}$?

תשובה: מכיוון שלכל M כלשהי ל-$L(M)$ קיימת מכונה המקבלת כל מילה בשפה, אזי מתקיים כי לכל M, $L(M) \in RE$. כלומר L_1 הינה אוסף ייצוגי כל מכונות טיורינג. עבור כל מחרוזת, ניתן להכריע האם היא מייצג מכונת טיורינג. לפיכך $L_1 \in R$.

4.2. המחלקה של שפת העצירה

טענה: שפת העצירה, HP, איננה שייכת ל-CO-RE.

הוכחה: ראשית נאמר כי שפת העצירה איננה שייכת ל-R. ניתן לראות זאת על ידי התרגיל המחשבתי הבא: נניח בשלילה ששפת העצירה כן שייכת ל-R, אזי קיימת מכונה M_{HP} המכריעה אותה. כלומר, בהינתן קידוד של מכונה וקלט, המכונה אומרת האם אותה מכונה עוצרת על הקלט או לא.

נגדיר מכונת טיורינג המתנהגת באופן הבא:

$$M'(w) = \begin{cases} \text{if } f_{M_{HP}}(w,w) \text{ accepts} & \text{infinite loop} \\ \text{else} & \text{stop} \end{cases}$$

כעת נשאל, האם M' עוצרת כאשר היא מקבלת את הקידוד של עצמה, כלומר, האם $M'(M')$ עוצרת.

אם כן, אז לפי הפונקציה שהגדרנו היא החזירה "לא", כלומר $M'(M')$ לא עוצרת.

אם היא לא עוצרת, $f_{M_{HP}}(M',M')$ החזירה "כן", כלומר $M'(M')$ כן עוצרת.

קיבלנו סתירה ולכן לא קיימת מכונת טיורינג כנ"ל.

כעת אנחנו יודעים ש-HP לא שייכת ל-R. נראה כי היא שייכת ל-RE. ניתן לראות זאת על ידי בניית מכונה פשוטה M: המכונה מריצה את המכונה האוניברסלית עם המכונה והקלט הנתונים. אם היא עוצרת, המכונה מקבלת. מכונה זו אכן עומדת בדרישות: עבור כל זוג השייך ל-HP, לפי הגדרה המכונה האוניברסלית תעצור בסופו של דבר, ולכן M תעצור.

HP איננה שייכת ל-CO-RE. אם היא היתה שייכת ל-CO-RE הרי שהיא היתה גם שייכת לחיתוך בין CO-RE ל-RE, כלומר היא היתה שייכת ל-R, בסתירה לכך שראינו שהיא איננה ב-R.

4.3. תרגיל חשוב

תרגיל: הוכח כי השפה הבאה הינה ב-RE: $L=\{\langle M\rangle \mid L(M)\neq\phi\}$

חשיבות התרגיל הינה בשיטת הפתרון, החוזרת על עצמה בתרגילים רבים.

פתרון: נראה כי השפה הינה ב-RE על ידי בניית מכונה M' המקבלת אותה.

המכונה M':

באיטרציה ה-i של המכונה, היא:

- מוסיפה על הסרט את המחרוזת ה-i בסדר הלקסיקוגרפי של Σ^*.
- לכל קלט שכבר כתוב על המכונה, היא מפעילה עליו את M עוד i צעדים.
- אם אחד הקלטים מגיע למצב מקבל – נעצור ונקבל.

חשוב לציין: במידה וקיים קלט עבורו M עוצרת – נגיע אליו בזמן סופי (על ידי מניה על כל הקלטים מובטח שנגיע לכל קלט סופי שהמכונה מקבלת). בנוסף, כל איטרציה של M' מורכבת ממספר סופי של צעדים.

5. רדוקציות

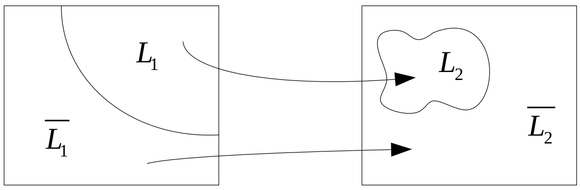
5.1. הגדרה

תהיינה $L_1, L_2 \in \Sigma^*$ שפות. פונקציה $f : \Sigma^* \to \Sigma^*$ נקראת **רדוקציה** מ- L_1 ל- L_2 אם היא מקיימת:

1. f מלאה (מוגדרת על כל קלט $x \in \Sigma^*$). (נשים לב ש- f אינה מוגדרת רק עבור המילים השייכות ל- L_1!).
2. f ניתנת לחישוב (קיימת מ"ט שמחשבת אותה).
3. תקפות: $\forall x, x \in L_1 \Leftrightarrow f(x) \in L_2$

אם קיימת f כנ"ל נאמר כי L_1 ניתנת לרדוקציה ל- L_2, ונסמן: $L_1 \le L_2$.

נבהיר בציור את מושג התקפות, על מנת לעשות אותו אינטואיטיבי יותר. יהיו L_1, L_2 שפות, ו- $\overline{L_1}, \overline{L_2}$ השפות המשלימות להן ($L \cup \overline{L} = \Sigma^*$). אזי:



הרדוקציה f מעבירה איברים שהיו בשפה L_1 אל איברים ב- L_2 ומעבירה איברים מ- $\overline{L_1}$ אל $\overline{L_2}$.

דוגמאות:

- $L_D \le L_u$: נראה פונקציה f המקיימת את הגדרת הרדוקציה: $f(\langle M \rangle) \triangleq (\langle M \rangle, \langle M \rangle)$.
- $L_u \le HP$. ניקח את הפונקציה $f(\langle M \rangle, \langle x \rangle) \triangleq (\langle M' \rangle, \langle x \rangle)$, כאשר M' היא מ"ט המוגדרת באופן הבא: M' זהה ל- M למעט אם M נכנסת ל- q_R, ואז M' נכנסת ללולאה אינסופית, כלומר אם $\delta(q,a) = (q_R, \ldots)$ אזי $\delta'(q,a) = (q,a,S)$.

5.2. משפט הרדוקציה

תהינה L_1, L_2 שפות כלשהן כך שמתקיים $L_1 \le L_2$, אזי:

1. אם $L_2 \in R \Leftarrow L_1 \in R$ (באופן שקול: אם $L_1 \notin \mathrm{R} \Leftarrow L_2 \notin \mathrm{R}$)
2. אם $L_2 \in RE \Leftarrow L_1 \in RE$ (באופן שקול: אם $L_1 \notin \mathrm{RE} \Leftarrow L_2 \notin \mathrm{RE}$)
3. אם $L_2 \in \text{CO-RE} \Leftarrow L_1 \in \text{CO-RE}$ (באופן שקול: אם $L_1 \notin \text{CO-RE} \Leftarrow L_2 \notin \text{CO-RE}$).

הערה: לרוב נשתמש במשפטים אלו כדי להוכיח כי שפות <u>אינן</u> ב- $\mathrm{R}, \mathrm{RE}, \text{CO-RE}$.

הוכחת 1.

$L_1 \le L_2 \Leftarrow$ קיימת פונקציה f המהווה רדוקציה מ- L_1 ל- L_2 ובפרט קיימת מ"ט M_f המחשבת אותה (ניתנת לחישוב). בנוסף מכיוון ש- f מלאה, M_f עוצרת תמיד.

$L_2 \in R \Leftarrow$ קיימת M_2 שעוצרת תמיד ו- $L(M_2) = L_2$. נבנה מ"ט M_1 עבור L_1 על קלט x:

1. המכונה משתמשת ב- M_f על מנת לחשב את $f(x)$.
2. המכונה מריצה את M_2 על $f(x)$. אם M_2 מקבלת/דוחה, M_1 עושה כמוה.

מתקיים:

- M_1 עוצרת תמיד (כי M_f, M_2 עוצרות תמיד).
- $x \in L_1 \Leftrightarrow f(x) \in L_2$ (עקב תקפות f) ומכאן מנכונות M_2 נובע כי גם M_1 עוצרת ומקבלת על x.
- $x \notin L_1 \Leftarrow f(x) \notin L_2 \Leftarrow M_2$ דוחה את $f(x)$ $M_1 \Leftarrow$ דוחה את x.

5.3. תכונות של רדוקציות

1. **כל שפה היא רדוקציה לעצמה**: לכל שפה L מתקיים $L \le L$. (פונקצית הזהות).
2. בהינתן 2 שפות L_1, L_2 כך שמתקיים $L_1 \le L_2$ אזי מתקיים $\overline{L_1} \le \overline{L_2}$ (אותה רדוקציה).
3. **טרנזיטיביות של רדוקציות**: יהיו L_1, L_2, L_3 כך שמתקיים: $L_1 \le L_2$ (f היא פונקצית הרדוקציה) וכן $L_2 \le L_3$ (g היא פונקציה הרדוקציה), אזי מתקיים: $L_1 \le L_3$, כאשר מתקיים כי $h(x) \triangleq g\left(f\left(x\right)\right)$ היא פונקציה הרדוקציה.
4. **היחס אינו סימטרי** – קיימות L_1, L_2 כך שמתקיים $L_1 \le L_2$ וכן $L_2 \not\le L_1$, לדוגמא: $L_1 \in R$ וגם $L_2 = HP$.
5. לכל $L \ne \phi$ מתקיים $L \not\le \phi$.

5.4. שימוש ברדוקציה – הוכחה ששפה אינה ב-RE / CO-RE

כאשר נתונה שפה L וברצוננו להראות כי L איננה ב-CO-RE, נוכל להשתמש ברדוקציה.

כלל אצבע: את רוב השפות שאינן ב-CO-RE ניתן להוכיח שאינן ב-CO-RE על ידי רדוקציה מ-HP.
על מנת להוכיח ששפה איננה ב-RE, נראה רדוקציה מ-$\overline{\mathrm{HP}}$.

דוגמא: נביט בשפה הבאה: $L = \left\{\left\langle M\right\rangle : f_M\left(\varepsilon\right) = \varepsilon\right\}$. האם שפה זו שייכת ל-R? האם היא שייכת ל-RE?

פתרון: נגדיר את הפונקציה f הבאה: $f\left(\left\langle M\right\rangle, x\right) = \left\langle M_x\right\rangle$.

נגדיר את M_x כך:

א. המכונה מתנהגת באופן שרירותי עבור כל קלט שאינו ε.

ב. עבור קלט שהוא ε, המכונה מריצה את M על x. כאשר M עוצרת על x, המכונה תפלוט ε.

טענה: הפונקציה f הינה רדוקציה $HP \le L$:

- הפונקציה מלאה: עבור כל קלט, מוגדרת מהי התוצאה של הפונקציה.
- הפונקציה ניתנת לחישוב: ע"י פעולת קומפילציה פשוטה ניתן ליצור פונקציה המתנהגת כמתואר.
- תקפות:

א. ניקח איבר $(\langle M\rangle, x)$ השייך ל-HP. נבנה את $\langle M_x\rangle$ המתאימה לו. מכיוון שהאיבר שייך ל-HP, אז כאשר נריץ את $\langle M_x\rangle$ עם ε היא תעצור ותפלוט ε.

ב. ניקח איבר $(\langle M\rangle, x)$ שאינו שייך ל-HP. נבנה את $\langle M_x\rangle$ המתאימה לו. מכיוון שהאיבר אינו שייך ל-HP, אז כאשר נריץ את $\langle M_x\rangle$ עם ε היא לעולם לא תעצור.

ניתן לראות כי אכן מתקיימת התקפות.

הראנו $HP \le L$. מכיוון ש-HP איננה שייכת ל-CO-RE, ולפי משפט הרדוקציה, נקבל כי L אינה שייכת אף היא ל-CO-RE.

נסכם את מה שעשינו:

1. רצינו להראות כי שפה L אינה ב-CO-RE.
2. הראנו רדוקציה $HP \le L$.
3. מ- $HP \le L, HP \notin \text{CO-RE}$ הגענו למסקנה: $L \notin \text{CO-RE}$.

הערה: ניתן להשתמש גם בטרנזיטיביות להוכחה דומה.

נניח כי עבור שפה L מתקיים כי $L_2 \le L$ וגם $HP \le L_2$, אזי מהטרנזיטיביות נובע כי $HP \le L$ ושוב נוכל להגיע למסקנה כי $L \notin \text{CO-RE}$.

6. שפות שאינן כריעות

6.1. קיום שפות שאינן כריעות

טענה: קיימות שפות שאינן ב-R (וכן נטען גם טענה חזקה יותר: קיימות שפות שאינן ב-RE).

הוכחה: נטען כי מתקיים:

$$\left|\text{RE}\right| \le \left|\text{Tuiring Machines Set}\right| \le \left|\left\{w \mid w \in \{0,1\}^* \wedge |w| \in \mathbb{N}\right\}\right| < \left|\left\{L \mid \forall w \in L, w \in \{0,1\}^*\right\}\right|$$

המעבר הראשון יבוצע על ידי התאמת מכונת טיורינג לכל שפה שב-RE (על פי הגדרה, קיים כזה).

המעבר השני יבוצע על ידי התאמת $\langle M \rangle$ לכל M. המעבר השלישי נכון מכיוון ש-$|\mathbb{N}| < |\mathbb{R}|$.

טענה: $\overline{L_D} \notin R$.

מסקנה: $L_D \notin \text{R} \underset{L_D \le HP}{\overset{L_D \le L_u}{\Longleftarrow}} L_u, \text{HP} \notin \text{R} \Leftarrow \overline{L_u}, \overline{\text{HP}} \notin \text{R}$.

טענה: $\overline{L_D} \notin \text{RE}$.

הוכחה: נניח בשלילה כי $\text{RE} \ni \overline{L_D} = \{\langle M \rangle \mid \langle M \rangle \notin L(M)\}$, אזי קיימת מכונת טיורינג M_D כך שמתקיים $L(M_D) = \overline{L_D}$.

מכאן: קיים $\langle M_D \rangle$. נרצה לבדוק האם $\langle M_D \rangle \in \overline{L_D}$.

נניח שכן, אזי $\langle M_D \rangle \in \overline{L_D}$, ומכאן לפי הגדרת $\overline{L_D}$ מתקיים $\langle M_D \rangle \notin L(M_D)$, כלומר $\langle M_D \rangle \notin \overline{L_D}$, בסתירה להנחה. בכיוון השני, אם נניח כי $\langle M_D \rangle \notin \overline{L_D}$ אזי $\langle M_D \rangle \in L(M_D)$, כלומר $\langle M_D \rangle \in \overline{L_D}$, שוב בסתירה להנחה. מכאן שהטענה נכונה. מ.ש.ל.

הגדרה: שפה L נקראת **RE שלמה** אם היא מקיימת:

1. $L \in \text{RE}$.
2. $\forall L' \in \text{RE}, L' \le L$.

במילים: שפה היא RE-שלמה אם היא שייכת ל-RE וכן ניתן לעשות אליה רדוקציה מכל שפה אחרת שב-RE.

<u>טענה</u>: L_u היא RE שלמה (ולכן גם HP היא RE שלמה).

6.2. משפט Rice

<u>הגדרה</u>: **תכונה** של שפות RE היא קבוצת שפות $S \subseteq \mathrm{RE}$. תכונה תיקרא **תכונה טריויאלית** אם מתקיים $S = \Phi$ או $S = \mathrm{RE}$.

<u>סימון</u>: עבור תכונה S, נגדיר את הסימון הבא: $L_S \triangleq \{\langle M\rangle \mid L(M) \in S\}$.

L_s היא שפה המכילה את קידודי כל מכונות טיורינג שהשפות שהן מקבלות שייכות ל-S.

<u>אבחנה</u>: אם S טריויאלית אז $L_S \in \mathrm{R}$.

משפט Rice: לכל תכונה לא טריויאלית S נובע כי $L_S \notin R$.

דוגמאות לשימושים במשפט:

1. נגדיר את השפה $L_\varepsilon = \{\langle M\rangle \mid \varepsilon \in L(M)\}$. L_ε היא שפה המתאימה לתכונה $L_{S_1} = L_\varepsilon$.
   S_1 איננה תכונה טריויאלית: קיימות שפות ב-RE המקיימת את S_1: $\Sigma^*, \{\varepsilon\}, ...$, וכמו כן קיימות שפות ב-RE שאינן מקיימות את S_1: $\phi, \{1111\}, ...$.
   מכאן לפי משפט Rice: $L_\varepsilon = L_{S_1} \notin R$.
2. $L_2 = \{\langle M\rangle \mid L(M) \text{ is final}\} \notin \mathrm{R}$.
3. $L_{\Sigma^*} = \{\langle M\rangle \mid L(M) = \Sigma^*\} \notin R$.

משפט Rice 2: תהי S תכונה לא טריויאלית המקיימת $\phi \in S$, אזי מתקיים: $L_S \notin \mathrm{RE}$.

כיצד משתמשים במשפט Rice כדי להוכיח ששפה אינה ב-R?

נגדיר תכונה S ונוכיח שמתקיים:

1. קיימת $L' \in RE$ כך ש-$L' \notin S$ (ומכאן $S \neq RE$).
2. קיימת $L'' \in RE$ כך ש-$L'' \in S$ (ומכאן $S \neq \phi$).
3. נראה כי $L_S = L$.

נסיק כי $L \notin R$.

מתי לא נשתמש במשפט Rice?

1. קיימות שפות שקל להוכיח בעזרת רדוקציה, אך קשה להתאימן למשפט Rice.

   דוגמא: $L_A = \left\{\left(\langle M_1\rangle,\langle M_2\rangle\right) \mid \left|L(M_1)\bigcap L(M_2)\right| = 3\right\}$.

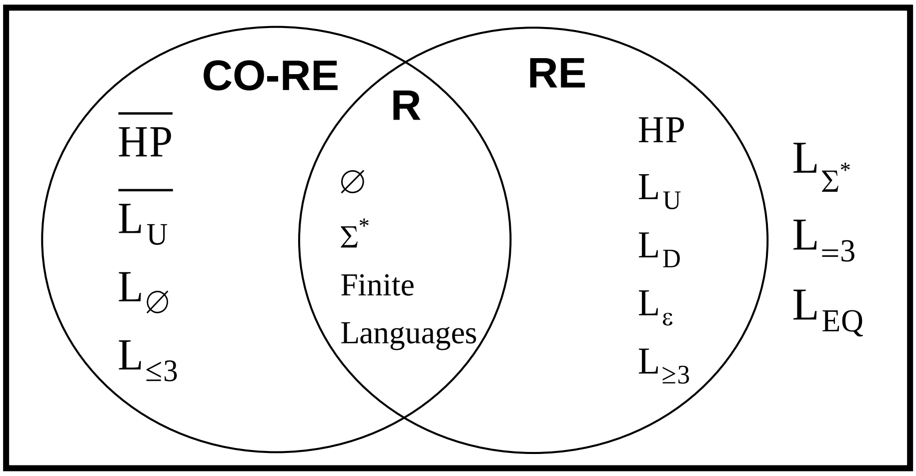
   קל למצוא רדוקציה מ-$L_{=3} \le L_A$, לדוגמא: $f\left(\langle M\rangle\right) = \left(\langle M\rangle,\langle M\rangle\right)$ אולם בעייתי להשתמש עבור שפה זו במשפט Rice.
2. כאשר התכונה היא תכונה של מכונה ולא של שפה. למשל: שפת קידודי כל המכונות העוצרות על הפלט הריק.

   לכל שפה ב-RE שאינה מכילה את ε יש מ"ט שמקבלת אותה וגם עוצרת על ε ויש מ"ט שמקבלת אותה ולא עוצרת על ε ולכן התכונה תלויה במכונה ולא בשפה.
3. כאשר התכונה טריויאלית. לדוגמא: $L = \{\langle M\rangle \mid \underbrace{L(M)}_{\in RE} = \underbrace{\overline{HP}}_{\notin RE}\} = \phi$

6.3. תמונת העולם

עמוד זה לקוח מסיכום שנכתב על ידי Psibeast בנושא חישוביות.

<u>מרחב השפות</u>: (כולן מעל א"ב {0,1})



<u>ריכוז שפות</u>:

בעיית העצירה/שפת העצירה (Halting Problem): HP={<M>,<X> | M עוצרת על X}

השפה האוניברסלית: L_u={<M>,<X> | M מקבלת את X}

שפת האלכסון - כל המכונות המקבלות את הקידוד של עצמן: L_D={<M> | M מקבלת בריצתה את המחרוזת <M>}

כל המכונות המקבלות כל קלט: $L_{\Sigma^*}=\{<M>\mid L(M)=\Sigma^*\}$

כל המכונות אשר מקבלות את ε: $L_\varepsilon=\{<M>\mid \varepsilon\in L(M)\}$

כל המכונות אשר לא מקבלות אף קלט: $L_{\varnothing}=\{<M>\mid L(M)=\varnothing\}$

כל המכונות המקבלות אותם קלטים: $L_{EQ}=\{(<M_1>,<M_2>)\mid L(M_1)=L(M_2)\}$

כל המכונות המקבלות 3 או פחות קלטים: $L_{\le 3}=\{<M>:\mid L(M)\mid\le 3\}$

כל המכונות המקבלות בדיוק 3 קלטים: $L_{=3}=\{<M>:\mid L(M)\mid=3\}$

כל המכונות המקבלות 3 קלטים או יותר: $L_{\ge 3}=\{<M>:\mid L(M)\mid\ge 3\}$

1.3.6. תרגיל 1

הוכח כי $L_{=3}$ אינה ב-RE או ב-CO-RE.

פתרון: נוכיח את הטענה בשני שלבים:

א. נראה רדוקציה $HP \le L_{=3}$ ונסיק ש-$L_{=3}$ אינה ב-CO-RE.

ב. נראה רדוקציה $\overline{HP} \le L_{=3}$ ונסיק ש-$L_{=3}$ אינה ב-RE.

א. נגדיר את הפונקציה הבאה: $f\left(\langle M\rangle, x\right)=\langle M_x\rangle$. פעולת M_x תוגדר באופן הבא:

לכל קלט w, המכונה תריץ ראשית את M על x. כאשר M עוצרת, היא תעבור למצב מקבל אם w הינו 1 או 11 או 100.

פונקציה זו הינה רדוקציה:

1. **פונקציה מלאה**: לכל $\langle M\rangle$ ולכל x מוגדר פלט הפונקציה.
2. **פונקציה ניתנת לחישוב**: ניתן לבנות מכונה שתריץ את M על x, וכן ניתן בפשטות להרכיב עליה מכונה שתקבל עבור הקלטים המבוקשים.
3. **תקפות**: נחלק לשני מקרים:

א. אם $\left(\langle m\rangle, x\right)$ אינו שייך ל-HP, אזי M_x הינה מכונה שלא תעצור, והשפה המתאימה לה היא השפה הריקה, ומכאן $\langle M_x\rangle \notin L_{=3}$.

ב. אם $\left(\langle m\rangle, x\right)$ שייך ל-HP, אזי M_x היא מכונה שמקבלת עבור אחד משלושת הקלטים 1,11,100. מתקיים כי $\langle M_x\rangle \in L_{=3}$.

ניתן לראות כי אכן התקפות מתקיימת.

הראנו רדוקציה $HP \le L_{=3}$ ומכאן $L_{=3}$ אינה ב-CO-RE.

ב. נגדיר את הפונקציה הבאה: $f\left(\langle M\rangle, x\right)=\langle M_x\rangle$.

פעולת M_x תוגדר באופן הבא: עבור קלט w: אם $w=1$ או $w=10$ או $w=110$ אזי המכונה עוצרת ומקבלת. אחרת המכונה מריצה את M על x. כאשר M עוצרת, המכונה מקבלת את w.

נטען שפונקציה זו הינה רדוקציה:

1. **פונקציה מלאה**: לכל $\langle M\rangle$ ולכל x מוגדר פלט הפונקציה.
2. **פונקציה ניתנת לחישוב**: ניתן לממש פונקציה כזו על ידי פעולת קומפילציה פשוטה.
3. **תקפות**: לא נוכיח את התקפות באופן פורמלי, אך נביט בשפה המוגדרת על ידי M_x:

השפה $L\left(M_x\right)$ הינה $\{1,10,110\}$ אם $\left(\langle m\rangle, x\right)\in\overline{HP}$. אחרת, השפה תהיה Σ^*.

הראנו רדוקציה $\overline{HP}\le L_{=3}$ ומכאן $L_{=3}$ אינה ב-RE.

2.3.6. תרגיל 2

הוכח כי $L_{\Sigma^*}=\left\{\langle M\rangle \mid L\left(M\right)=\Sigma^*\right\}$ אינה ב-RE ואינה ב-CO-RE.

פתרון: נפתור תרגיל זה בצורה דומה לתרגיל הקודם, על ידי רדוקציה מ-HP. עם זאת, הרדוקציות יוגדרו באופן שונה מעט בתרגיל זה.

א. נגדיר את הפונקציה הבאה: $f\left(\langle M\rangle, x\right)=\langle M_x\rangle$. פעולת M_x על קלט w תוגדר באופן הבא: M_x מריצה את x על M. כאשר M עוצרת, המכונה מקבלת את המילה w.

1. **פונקציה מלאה**: לכל $\langle M\rangle$ ולכל x מוגדר פלט הפונקציה.
2. **פונקציה ניתנת לחישוב**: ניתן לממש פונקציה כזו על ידי פעולת קומפילציה פשוטה.

3. **תקפות**: עבור $(\langle M\rangle, x)\in HP$ מתקיים כי המכונה תקבל כל מילה w, ולכן קידוד המכונה שייך ל- L_{Σ^*}. עבור $(\langle M\rangle, x)\notin HP$ מתקיים כי המכונה אינה עוצרת ולכן קידודה אינו שייך ל- L_{Σ^*}.

הראנו רדוקציה $HP \le L_{\Sigma^*}$ ומכאן L_{Σ^*} אינה ב-CO-RE.

ב. נגדיר את הפונקציה הבאה: $f(\langle M\rangle, x)=\langle M_x\rangle$. פעולת M_x על קלט w תוגדר באופן הבא: M_x מריצה את x על M למשך $|w|$ צעדים. אם M לא עצרה על x, M_x מקבלת את w. אם M עצרה על x, אז M_x דוחה את w או נכנסת ללולאה אינסופית.

1. **פונקציה מלאה**: לכל $\langle M\rangle$ ולכל x מוגדר פלט הפונקציה.
2. **פונקציה ניתנת לחישוב**: ניתן לממש פונקציה כזו על ידי פעולת קומפילציה פשוטה.
3. **תקפות**: אם M עוצרת עבור קלט x, אזי קיימת מילה w כך ש- M_x תדחה, ונקבל מכונה שאינה ב- L_{Σ^*}.

הראנו רדוקציה $\overline{HP} \le L_{\Sigma^*}$ ומכאן L_{Σ^*} אינה ב-RE.

7. חישוב פונקציות

7.1. הגדרות בסיסיות

<u>הגדרה</u>: פונקציה $f:\Sigma^* \to \Sigma^*$ היא **ניתנת לחישוב** אם קיימת מ"ט M כך ש- $f_M = f$.

<u>הגדרה</u>: פונקציה "קלה" היא פונקציה ניתנת לחישוב. פונקציה "קשה" היא פונקציה שלא ניתנת לחישוב.

<u>הגדרה</u>: בהינתן פונקציה f , נגדיר: $L_f \triangleq \{(x,y) \mid y = f(x)\}$.

<u>דוגמאות לפונקציות הניתנות לחישוב</u>:

- פונקצית הזהות.
- פונקציה המוציאה ε תמיד.
- פונקציות רבות נוספות.

<u>דוגמא לפונקציה שאינה ניתנת לחישוב</u>:

- פונקציה המקבלת קידוד של מכונה וקלט x ומחזירה 1 אם המכונה עוצרת על הקלט, ו-0 אם לא.

<u>תרגיל</u>: לאיזו מחלקה שייכת השפה הבאה: $L = \{\langle M\rangle \mid$ אינה ניתנת לחישוב $f_M\}$

<u>פתרון</u>: אם f_M אינה ניתנת לחישוב, אז אין מ"ט המתאימה לה, ולכן L היא השפה הריקה, ומכאן כפי שכבר ראינו $L \in R$.

<u>משפט</u>: תהי פונקציה f , אזי:

1. f ניתנת לחישוב $\Leftrightarrow L_f \in \mathrm{RE}$.
2. f היא פונקציה מלאה $\Rightarrow$ (f ניתנת לחישוב $\Leftrightarrow L_f \in R$)

החלק הראשון של המשפט אומר לנו בעצם כי <u>השפה של כל מכונה שייכת ל-RE</u>.

7.2. סיבוכיות קולמגורוב

<u>הגדרה</u>: למען הפשטות נגדיר $\Sigma=\{0,1\}$ וכן $\Gamma=\{0,1,\flat\}$. בהינתן מחרוזת $x\in\Sigma^*$, נגדיר את סיבוכיות קולמגורוב של x, $k(x)$, בתור מספר המצבים הקטן ביותר של מ"ט שמוציאה פלט x על ε (המילה הריקה). פונקציה זו הינה $k:\{0,1\}^*\to\mathbb{N}$.

<u>אבחנה 1</u>: $k(x)\le|x|+1$, וכן $k(x)$ היא פונקציה מלאה. בהינתן $x=x_1,x_2...,x_n$ קיימת M_x עם המצבים $q_1,q_2,...,q_{n+1}$ כשהמצב q_i מדפיס את x_i ועובר למצב הבא.

<u>אבחנה 2</u>: לכל $n\in\mathbb{N}$ קיים x כך שמתקיים $k(x)>n$ - כלומר $k(x)$ הינה פונקציה לא חסומה.

הוכחה: יהי $n\in\mathbb{N}$ כלשהו. מספר המכונות בעלות לכל היותר n מצבים הוא $(9k)^{3k}$ מכונות שונות לכל היותר. כלומר: $\delta:Q\times\Gamma\to Q\times\Gamma\times\{L,R,S\}$. כל מכונה M מועמדת להיות מ"ט מינימלית רק עבור x בודד שהוא הפלט שלה על ε. מספר המחרוזות עבורן $k(x)\le n$ סופי. כיוון שקיימות ∞ מחרוזות, הטענה נובעת.

<u>משפט</u>: הפונקציה $k(x)$ איננה ניתנת לחישוב.

<u>הוכחה</u>: נניח בדרך השלילה כי $k(x)$ ניתנת לחישוב $\Leftarrow$ קיימת M_k המקבלת מחרוזת x ופולטת את $k(x)$. נבנה מכונה M_1 כך ש: M_1 מפרשת את הקלט כמספר טבעי n ופולטת y כך ש-y הוא המחרוזת הראשונה בסדר לקסיקוגרפי המקיימת $k(y)>n$.

אבחנה: M_1 תמיד עוצרת.

יהי t מספר המצבים של M_1. ניקח מספר טבעי N כך שמתקיים: $2^N+N+10\ge t$.

נבנה את המכונה M_2 על קלט ε:

1. M_2 כותבת על הסרט את 2^N בייצוג בינארי.
2. לאחר מכן המכונה חוזרת אל התחלת הסרט, ומריצה על הסרט את M_1.
3. כאשר M_1 פולטת תוצאה, M_2 פולטת כמוה ועוצרת.

נתבונן בפלט של M_2 על ε. נסמן את פלט זה ב- z. כעת:

א. מצד אחד, מתקיים כי $k(z) > 2^N$, עקב האופן בו בנינו את M_1.

ב. מצד שני:

- M_2 בתגובה ל- ε פולטת את z.
- מספר המצבים של M_2 הוא $t + N + 10$:

- t הוא מספר המצבים של M_1.

- N אלו מצבים נוספים על מנת שהמכונה תהיה מסוגלת להדפיס את 2^N.

- 10 מצבים נוספים הינם מצבי עזר לצורך ההדפסה שהמכונה משתמשת בהן (יכולנו לבחור גם מספר אחר.

משני כיוונים אלו אנחנו מקבלים כי: $k(z) \le t + N + 10 < 2^N$ וקיבלנו סתירה.

מכאן אנחנו מסיקים: הפונקציה $k(x)$ איננה ניתנת לחישוב.

8. בעיות חיפוש

הגדרה: בהינתן יחס דו מקומי $S \subseteq \Sigma^* \times \Sigma^*$ אומרים שמכונת טיורינג M פותרת את בעיית החיפוש של S אם לכל x: אם קיים y כך ש-$(x, y) \in S$, אז M עוצרת עם y כזה כפלט. אם לא קיים y כנ"ל, אז M לא עוצרת.

נשים לב שקיימת התאמה בין יחס לשפה. $L_S = \{(x, y) \mid (x, y) \in S\}$.

בעיית הזיהוי של $S \equiv$ בעיית הזיהוי של L_s, כלומר התשובה לשאלה האם $L_s \in RE$.

לכל יחס S מתקיים כי זיהוי$\Leftarrow$חיפוש אולם חיפוש $\not\Leftarrow$ זיהוי.

חיפוש: בהינתן x מצא y מתאים. זיהוי: נתונים (x, y), האם הם שייכים ליחס S?

נביא דוגמא לצורך המחשת ההבדל בין חיפוש לזיהוי:

יהי S היחס הבא: $S = \{(\langle M_1 \rangle, \langle M_2 \rangle) \mid L(M_1) = L(M_2)\}$.

בעיית הזיהוי במקרה זה היא בעצם למצוא מכונה המקבלת את L_{EQ}, אבל כפי שכבר ראינו, $L_{EQ} \notin RE$.

לעומת זו, בעית חיפוש במקרה זה היא קלה: בהינתן מכונה M_1, נוכל בקלות למצוא M_2 שמתאים לה – נשכפל פשוט את M_1, וברור שהמכונה המשוכפלת תהיה בעלת אותה שפה.

9. רשימת מקורות

- הרצאות "תורת החישוביות" בטכניון. http://webcourse.cs.technion.ac.il/236343
- תרגולים בקורס "תורת החישוביות" בטכניון.
- Sipser, Michael', Introduction to the Theory of Computation 2nd edition (2004). Course Technology