# חלק 1 - חישוביות

## א) מספר תוכניות מחשב

לכל תוכנת מחשב יש קלט ופלט המתאים לקלט. לכן ניתן לומר שכל תוכנת מחשב מחשבת פונקציה $F:\Sigma^* \to \Sigma^*$ כאשר $\Sigma$ היא קבוצת כל התווים שהמחשב יכול להשתמש בהם.

כמה פונקציות כאלו קיימות? נשים לב ש-$|\Sigma^*| = \aleph_0$ (כי ניתן לסדר את $\Sigma^*$ בסידור לקסיקוגרפי). לכל קלט x יכולים להיות $\aleph_0$ אפשרויות ל-$F(x)$. לכן מספר הפונקציות הכולל הוא: $\aleph_1 = 2^{\aleph_0} \le \aleph_0^{\aleph_0}$.

לעומת זאת, כל תוכנת מחשב מורכבת ממספר סופי של תווים, שכן כל תוכנה היא מחרוזת תווים.

לכן מספר תוכניות המחשב הוא לכל היותר $\aleph_0 = |\Sigma^*|$.

אם נסתכל על כל פונקציה כעל בעיה לפתור נובע משיקולי הספירה לעיל שכמעט לכל הבעיות אין בכלל תוכנת מחשב שפותרת אותן. בהמשך הפרק נרצה ללמוד כיצד לקבוע האם בעיה היא פתירה או לא.

אמנם לשם כך נבדוק מודל חישובי השקול למחשב המודרני שבאמצעותו נבחן בעיות אלו והאם המודל יכול לפתור אותן. מודל חישובי זה יהיה מכונת טיורינג.

## ב) מכונת טיורינג דטרמיניסטית

תיאור:

- סרט אינסופי (לימין) של תאים. לכל תא יש אינדקס, נניח בקורס שמתחיל מ-1, ובכל תא יש אות מ-$\Gamma$.
  הסרט מוגדר אינסופי כדי שלא יקרה מצב שחסר מקום אחסון, אמנם בפועל הוא אינו אינסופי שכן ניתן "לכתוב" עליו, וכל מה שניתן לכתוב הוא סופי.
- ראש קורא/כותב שיכול לזוז על גבי הסרט ימינה, שמאלה או להישאר במקום $(R,S,L)$. מסומן ע"י חץ קטן בסרט.

הגדרה: מכונת טיורינג $M$ מורכבת משביעייה $M=(Q,q_0,F,\Gamma,\Sigma,b,\delta)$.

1) $Q$ - קבוצת מצבים סופית.

2) $q_0$ - מצב התחלתי שבו המכונה נמצאת בתחילת כל חישוב.

3) $F\subseteq Q$ - קבוצת מצבים מקבלים.

4) $\Gamma$ - א"ב עבודה. אלו האותיות שניתן לכתוב בסרט. בקורס זה בדר"כ $\Gamma=\{0,1,b\}$.

5) $\Sigma\subseteq\Gamma$ - א"ב הקלט. האותיות שמהם מורכב הקלט x למכונה. בקורס זה בדר"כ $\Sigma=\{0,1\}$. $\Sigma$ לא מכיל $\varepsilon$.

6) $b\in\Gamma$ - אות מיוחדת הנקראת "בלנק" ומשמשת כדי לסמן תא ריק. כל תא בסרט שאינו קלט או פלט מסומן ב-$b$.

7) $\delta$ - פונקציית המעברים. מוגדרת כך $\delta:(Q\setminus F)\times\Gamma \to Q\times\Gamma\times\{R,L,S\}$. $\delta$ מקבלת מצב לא סופי, האות בסרט שעליו נמצא הראש קורא, ומחזירה מצב לעבור אליו, מה לכתוב בסרט במקום הנוכחי ולאן להזיז את הראש קורא לאחר הכתיבה. כשמגיעים למצב סופי עוצרת. במכונת טיורינג אין חובה לקרוא את כל הקלט ולא אפשר לזוז אין שרודים.

(2)

## מהלך החישוב:

בתחילת החישוב הקלט x רשום על גבי התאים הכי שמאליים בסרט וכל שאר התאים הם ␣. הראש הקורא נמצא בתא הראשון ומצב המכונה הוא $q_0$.

כל צעד חישוב הוא לפי פונקציית המעברים δ. כיוון שהמכונה דטרמיניסטית δ מוגדרת עבור כל מצב אפשרי של $\Gamma \times (Q \setminus F)$. כאשר המכונה מגיעה ל- $q\in F$ היא עוצרת ומחזירה פלט. במקרה שנמצאים בתא הראשון ופונקציית המעברים הורתה לראש הקורא ללכת שמאלה הראש הקורא ישאר במקום.

## פלט:

הפלט של המכונה מוגדר להיות המילה שכתובה משמאל לראש הקורא כאשר המכונה מגיעה למצב מקבל.*

| 0 | 1 | 1 | 0 |
|---|---|---|---|

$F(x) = 011$

כאשר המכונה עוצרת והראש הקורא נמצא בתא הראשון בסרט הפלט הוא ε (מילה ריקה).

כאמור לכל מכונת טיורינג M ניתן לתאר את הקלט והפלט המתאים לכל קלט כפונקציה $F_M:\Sigma^* \to \Gamma^*$.

לכל $x\in\Sigma^*$ נגדיר את הפלט של מכונת טיורינג M $F_M(x)$ או $M(x)$. חשוב לציין שלא בהכרח שמכונת טיורינג תעצור על קלט x, אלא יכול להיות שעבור קלט מסוים המכונה תיכנס ללולאה אינסופית ולא תעצור לעולם.

במקרה כזה נאמר שהפונקציה של המכונה אינה "מלאה", כלומר אינה מוגדרת לכל קלט.

## קונפיגורציה (snap shot):

היא שלשה $(\alpha, q, i)$ המתארת מצב כלשהו של מכונת טיורינג. זהו המידע המינימלי הנדרש לשחזר מצב של מכונה.

α - תוכן הסרט עד הבלנק הראשון שממנו לצד ימין כל התאים מכילים בלנקים, לא כולל. α תמיד מחרוזת סופית.

$|\alpha| \le \max(|x|, t)$ - כאשר t - מס' הצעדים עד הקונפיגורציה הנתונה.

q - מצב נוכחי $q\in Q$.

i - האינדקס של התא בסרט שעליו מצביע הראש קורא.

ניתן לתאר ריצה של מכונת טיורינג כסדרה של קונפיגורציות $C_0, C_1, \dots, C_n$, כאשר $C_n$ מצב סופי כלשהו.

## ג) שקילות

נלמד שני סוגים של שקילות: שקילות מכונות ושקילות מודלים.

**שקילות מכונות** - שתי מכונות טיורינג $M_1$ ו- $M_2$ יקראו שקולות אם לכל קלט $x\in\Sigma^*$ התוצאה של שתי המכונות זהה. כלומר אם $M_1(x)$ מוגדר אזי גם $M_2(x)$ מוגדר ושווה ל-$M_1(x)$. ואם $M_1(x)$ לא מוגדר גם $M_2(x)$ לא מוגדר.

ראשית, צריך להראות $M_1(x)=M_2(x)$ לכל $x\in\Sigma^*$ באינדוקציה על $|x|$. אמנם זה לא מספיק המספר לכל. במכונות אפשריות כי לא ניתן להוכיח בהכלה דו־כיוונית.

**שקילות מודלים** - לכל מודל A של מכונת חישוב נגדיר $F_A$ קבוצת כל הפונקציות הניתנות לחישוב על ידי מכונה כלשהי ממודל A. נאמר כי שני מודלים A ו-B שקולים אם"ם $F_A=F_B$. דרך ההוכחה היא באמצעות הכלה דו-כיוונית, אם $F_A\subseteq F_B$ וגם $F_B\subseteq F_A$ אזי $F_A=F_B$*. לשם כך מספיק להראות שמכל מכונה במודל A ניתן לבנות מכונה שקולה במודל B, וכן להיפך. אם מודל A מוכל ב-B אזי יש כיוון טריוויאלי שכן לכל מכונה $q\in A$ גם קיימת $q\in B$.

* $x\in F_A$ אז $x\in F_B$.
$x\in F_B$ אז $x\in F_A$.

## תזת צ'רץ' - טיורינג:

תזת צ'רץ' טיורינג אומרת שכל מודל "כללי" וסביר" שקול למכונת טיורינג.

כללי - הכוונה למודל החזק לפחות כמו מכונת טיורינג.

סביר - הכוונה למודל שלא מכיל רכיבים אינסופיים.

דוגמאות למודלים השקולים למכונת טיורינג:

- מודל עם מספר קבוע של סרטים אינסופיים (הוכחה בעבודה 1 שאלה 2.ב).
- הוספת מספר קבוע של ראשים קוראים כותבים כך שמכל סרט קוראים וכותבים מספר מקומות במקביל.
- הוספת סוגי צעדים עם שינויים קלים לקבוצת הצעדים האפשריים (במצגת 1 שקופית 22-25 הוכחה למודל עם הצעדים RRLL).
- באופן כללי שינויים למכונת טיורינג שאינם כוללים רכיבים אינסופיים.

דוגמאות למודלים שאינם שקולים למכונת טיורינג:

- מכונת טיורינג עם קבוצת מצבים אינסופית (הוכחה מצגת 1 שקופית 26).
- מכונת טיורינג עם א"ב עבודה $\Gamma$ אינסופית (הוכחה בעבודה 1 שאלה 2.א).
- באופן כללי - שינויים למכונת טיורינג הכוללים רכיב אינסופי.

## 3) מכונת טיורינג אוניברסלית

קידוד מכונת טיורינג $\langle M\rangle$ - ניתן לקודד כל מכונת טיורינג למחרוזת. לשם כך צריך לקודד את הרכיבים $\delta, \Sigma, \Gamma, Q$, שכן שאר הרכיבים זהים בכל מכונת טיורינג. הקידוד יכול להיות על בסיס בינארי או אונארי.*

*בבסיס אונארי: $1^3 0 1^4 0 1^5 0$.

דוגמה לקידוד מכונת טיורינג $M=(Q,q_0,F,\Gamma,\Sigma,b,\delta)$:

כל אחד מהמצבים והאותיות יקבל מספר שאותו נקודד. נשתמש כאן בקידוד על בסיס אונארי כאשר 0 מפריד בין הרכיבים.

$Q=\{1,2,\dots,|Q|\}$

$\Gamma=\{1,2,\dots,|\Gamma|\}$

$\Sigma=\{1,2,\dots,|\Sigma|\}\subsetneq\Gamma$

$b=|\Gamma|$

$(L,R,S)=(1,2,3)$

$\delta(q,a)=\delta(p,b,d) \Rightarrow \langle\delta(q,a)\rangle = 1^p 0 1^b 0 1^d 0$

$\langle M\rangle = 1^{|Q|} 0 1^{|\Gamma|} 0 1^{|\Sigma|} 0 \langle\delta(1,1)\rangle 0 \langle\delta(1,2)\rangle 0 \dots \langle\delta(|Q|,|\Gamma|)\rangle 0$

הערה - יש הנחה שנהוג להשתמש בה שכל מחרוזת נתונה היא קידוד של מכונת טיורינג כלשהי.

מכונת טיורינג אוניברסלית - זוהי מכונת טיורינג שהקלט שלה הוא קידוד של מכונת טיורינג ומחרוזת $x$ $\langle M,x\rangle$ ומה שהיא עושה הוא להריץ את $x$ כקלט על $M$ ומחזירה את הפלט שלה. נסמן מכונת טיורינג אוניברסלית ב-$U$. $U(\langle M,x\rangle)=M(x)$ לכל $x\in\Sigma^*$ ולכל מכונת טיורינג $M$.

## ה) מכונת טיורינג לקבלת שפות

מכונת טיורינג לקבלת שפות היא מכונת טיורינג רגילה שמקיימת $F=\{q_{acc},q_{rej}\}$, כלומר יש לה שני מצבים בהם היא עוצרת. מצב אחד "מקבל" את הקלט ומצב שני דוחה את הקלט. במכונות אלו אין חשיבות לפלט.

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

- אם המכונה M בריצה על x עוצרת ב-$q_{acc}$ נאמר ש-M "מקבלת" את x.
- אם המכונה M בריצה על x נעצרת ב-$q_{rej}$ נאמר ש-M "דוחה" את x.
- אם המכונה M בריצה על x לא עוצרת נאמר ש-M לא מקבלת את x וגם לא דוחה.

"השפה של המכונה M" היא כל המילים $x\in\Sigma^*$ אשר המכונה M מקבלת. שפה זו מסומנת ב-$L(M)$.
השפה המשלימה $\overline{L(M)}=\Sigma^*\setminus L(M)$ מכילה את כל המילים ש-M דוחה או לא עוצרת בהן. לכל מכונה שפה אחת.

### קבלת שפות מול הכרעת שפות

בהינתן שפה $L\subseteq\Sigma^*$ נאמר כי

- מכונה M מקבלת את השפה L אם $\forall x\in\Sigma^*$ $x\in L$ אם ורק אם M מקבלת את x.
- מכונה M מכריעה את השפה L אם $\forall x\in\Sigma^*$ $x\in L$ אם ורק אם M מקבלת את x וגם M עוצרת תמיד.

כלומר לכל מילה M מכריעה אם מתקבלת או נדחית ולעולם לא נכנסת ללולאה אינסופית.

### קבלת שפות מול חישוב פונקציות

עד כה עסקנו במכונות שמחשבות פונקציות $f:\Sigma^*\to\Gamma^*$ וכעת אנו מדברים על מכונות שמקבלות שפות שניתן להסתכל עליהן כפונקציה $F:\Sigma^*\to\{0,1\}$. לכאורה נראה שיש כאן צמצום בכח החישוב, אך למעשה המודלים שקולים שכן כל פונקציה F ניתן לתאר כשפה של מחרוזות מהצורה $\langle x,F(x)\rangle$. אם $\langle x,F(x)\rangle$ ש"ך למכונה M אזי במכונה זו $F(x)$ הוא הפלט של x.

### בניית מכונה לקבלת שפה והוכחת נכונות

בהינתן שפה L שנרצה לבנות לה מכונת טיורינג שמקבלת אותה צריך לתאר מכונת טיורינג עם שם ייחודי למכונה וקלט, הגדרה מה המכונה מבצעת בכל מצב, ותיאור פסאודו-קוד מתי המכונה מקבלת או דוחה.

כדי להוכיח $L(M)=L$ צ"ל: $x\in L(M) \iff x\in L$ $\forall x\in\Sigma^*$.

(1) $x\in L \Rightarrow \dots \Rightarrow M(x)=1 \Rightarrow x\in L(M)$

(2) $x\notin L(M)$:
- מכונה מכריעה: $M(x)=0 \Rightarrow \dots \Rightarrow x\notin L(M)$
- מכונה מקבלת: $M(x)=0$ או לא עוצרת $\Rightarrow \dots \Rightarrow x\notin L(M)$

## ו) מילון הגדרות

- מכונת טיורינג - תוכנית מחשב או אלגוריתם.
- מכונת טיורינג אוניברסלית - מחשב.
- פונקציה - בעיה. x מופע של הבעיה ו-$f(x)$ הפתרון.
- שפה - בעיית הכרעה.

## ג) המחלקות $R$, $RE$, ו-$coRE$

ניתן לחלק את כל השפות ב-$\Sigma^*$ למחלקות.

### המחלקה $RE$

קבוצת השפות שקיימת עבורן מכונת טיורינג שמקבלת אותן. כדי להראות ששפה $L$ שייכת ל-$RE$ מספיק להראות שקיימת מכונת טיורינג שמקבלת אותה. כלומר עבור כל קלט בשפה "המכונה" מקבלת ועבור קלט שאינו בשפה המכונת טיורינג דוחה או לא עוצרת. בהמשך נלמד כיצד להוכיח ששפה אינה ב-$RE$.

המחלקה $RE$ סגורה לאיחוד, חיתוך, שרשור, איטרציה, רברס. אך אינה סגורה למשלים. כלומר אם $L \in RE$ אזי $\bar{L}$ לא בהכרח שייכת ל-$RE$. הוכחה לכך שסגורה לאיחוד בשקף 15 במצגת 2.

### המחלקה $R$

קבוצת השפות שקיימת עבורן מכונת טיורינג המכריעה אותן. כדי להראות ששפה $L$ שייכת ל-$R$ יש להראות שקיימת מכונת טיורינג שמכריעה את השפה. בהמשך נלמד כיצד להוכיח ששפה אינה ב-$R$.

המחלקה $R$ סגורה לכל התכונות ש-$RE$ סגורה עליהן כולל משלים (הוכחה לכך שסגורה לאיחוד בשקף 17 במצגת 2).

דוגמאות לשפות ב-$R$: $\Sigma^*$, $\phi$ (השפה הריקה), שפות רגולריות (אס"ד הוא מקרה פרטי של מכונת טיורינג), שפות שאנו מכירים אלגוריתם, אפילו אינו פולינומי, שפותר אותן.

### המחלקה $coRE$

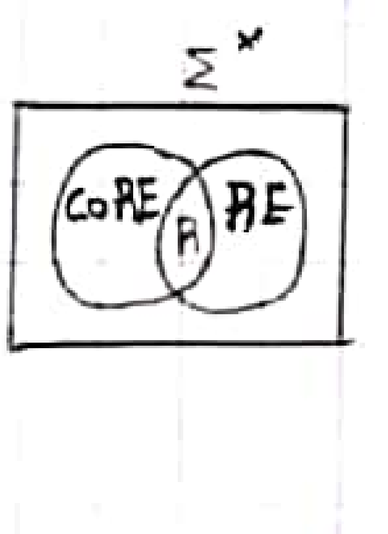
קבוצת השפות אשר לשפה המשלימה שלהם יש מכונת טיורינג מקבלת. $coRE=\{L\subseteq\Sigma^* \mid \bar{L}\in RE\}$. הגדרה שקולה היא קבוצת כל השפות שיש עבורן מכונת טיורינג אשר לכל מילה שאינה בשפה המכונה עוצרת ודוחה ולכל מילה שכן בשפה המכונה או עוצרת ומקבלת או לא עוצרת. כתוצאה מכך, כאשר מגדירים $L\in coRE$ על ידי מכונה $M$ שמקיימת את התנאים לעיל, לא בהכרח $L=L(M)$, שכן $M$ לא בהכרח עוצרת לכל $x \in L$.

* הערה - כל שפה שאינה באחת משלושת המחלקות נאמר שהיא שייכת ל-$\overline{coRE \cup RE}$.

### הקשר בין המחלקות

$R\subseteq RE$ - מכיוון שמכונת טיורינג המכריעה שפה בהכרח גם מקבלת אותה.

$R\subseteq coRE$ - מכונת טיורינג המכריעה שפה גם עוצרת על כל קלט שאינו בשפה.




משפט: $R = RE\cap coRE$. נוכיח באמצעות הכלה דו כיוונית.

הוכחה: כיוון 1 - תהי $L\in R$. כיוון ש-$R\subseteq RE$ וכיוון ש-$R\subseteq coRE$, $L\in coRE$, לכן $L\in RE\cap coRE$.

כיוון 2 - תהי $L\in RE\cap coRE$. כיוון ש-$L\in RE$ קיימת לה מכונה $M_1$ המצפה הקלט השייך ל-$L$ ומקבלת אותו. וכיוון ש-$L\in coRE$ קיימת מכונה $M_2$ המצפה הקלט לא בשפה ודוחה אותו. לכן נבנה מכונה $M^*$ המכריעה את $L$ כך שעל כל קלט $x$ מריצה במקביל* את $M_1$ ו-$M_2$ על $x$ אם $M_1$ קיבלה $M^*$ מקבלת ואם $M_2$ דחתה $M^*$ גם דוחה. הוכחת נכונות טריוויאלית.

* הרצה מקבילית - ר' עוד בסעיף הבא.

7

## ח) הרצה מבוקרת

כאשר מכונת טיורינג $M$ מסמלצת הרצת שתי מכונות טיורינג $M_1,M_2$, והפלט של $M$ תלוי בפלט של $M_1$ ו-$M_2$ (כמו לדוגמה המכונה שבנינו בסעיף קודם). דרך אחת לעשות זאת היא להריץ את $M_1$ ולאחר מכן את $M_2$, אמנם לא יכולה להיות בעיה ש-$M_1$ לעולם לא תעצור. הפתרון לבעיה זו הוא הרצה מבוקרת שבה מריצים את שתי המכונות במקביל. יש כמה דרכים כיצד לעשות זאת, למשל הצעד הזוגי להריץ את $M_1$ והצעד האי-זוגי להריץ את $M_2$, או הרצה על שני סרטים (ראינו ששקול למ"ט) כאשר כל סרט מריץ מכונה אחרת.

הרצה מבוקרת חכמה - כאשר רוצים שמכונת טיורינג תעבור על כל המילים ב-$\Sigma^*$ ותחפש מילה שמקיימת תכונה מסוימת. אם נעבור על כל המילים מילה אחר מילה נוכל להיכנס ללולאה אינסופית והחיפוש יתקע. דרך נוספת היא להגביל את מספר הצעדים בכל מילה, אמנם זה יכול להיות שנדחה מילה למרות שהיא המילה שחיפשנו. הפתרון לכך הוא הרצה מבוקרת חכמה, שבה משתמשים בסידור לקסיקוגרפי של $\Sigma^*$: $w_1,w_2,\ldots$. בשיטה זו באיטרציה הראשונה מתחילים ב-$w_1$ ומריצים אותה למשך צעד אחד. בכל איטרציה $i$ מוסיפים מילה $w_i$ ומריצים את כל המילים $w_1,\ldots,w_i$ אחת אחרי השנייה למשך $i$ צעדים.

פסאודו-קוד: • לכל $i=1,2,3,\ldots$

• לכל $j=1,2,\ldots,i$

• הרץ את המכונה $M$ על המילה $w_j$ למשך $i$ צעדים.

## ט) השפה $L_D$

כדי להוכיח ששפה אינה ב-$R$ יש להוכיח שכל מכונת טיורינג לא מכריעה אותה. זוהי כמובן הוכחה מאוד מסובכת, בסעיף הבא נמצא שיטה יותר קלה לעשות זאת, אמנם היא דורשת שפה שכבר ידוע שאינה ב-$R$ כדי להשתמש בה. השפה הזו היא $L_D=\{\langle M\rangle \mid M \text{ accepts } \langle M\rangle\}$. בסעיף זה נוכיח בדרך השלילה ש-$L_D\notin R$*.

הוכחה: נניח בשלילה ש-$L_D\in R$. אזי קיימת ל-$L_D$ מכונה $M_D$ המכריעה אותה. נבנה מכונה חדשה $A_D$ שעל כל קלט $\langle M\rangle$ מריצה את $M_D$ על $\langle M\rangle$ והופכת את הפלט, כלומר אם $M$ מקבלת $\langle M\rangle$ $A_D$ תדחה ואם $M$ דוחה את $\langle M\rangle$ $A_D$ תקבל. בנוסף, כיוון שיש מספר בן מנייה של מכונות טיורינג* ניתן לסדר אותן בסידור כלשהו. נבנה טבלה אינסופית של כל הקידודים האפשריים של מכונות טיורינג כך שבכל תא $(j,i)$ יהיה 1 אם המכונה $M_i$ מקבלת את הקלט $\langle M_j\rangle$, אחרת 0.* המכונה $M_D$ יודעת למלא את האלכסון בטבלה זו. כיוון ש-$A_D$ גם היא מכונת טיורינג היא צריכה להופיע בטבלה באינדקס $i$ כלשהו, $A_D=M_i$. נשקול את האפשרויות לערכים בתא $(i,i)$:

אם $(i,i)=1$ אזי $M_D(\langle A_D\rangle)$ מקבלת. לפי הגדרת $L_D$ $A_D(\langle A_D\rangle)$ מקבלת. אמנם כך $A_D$ צריכה לענות הפוך מ-$M_D$ לאותו קלט. סתירה!

אם $(i,i)=0$ אזי $M_D(\langle A_D\rangle)$ דוחה. לפי הגדרת $L_D$ $A_D(\langle A_D\rangle)$ דוחה. אמנם $A_D$ צריכה לענות הפוך מ-$M_D$ לאותו קלט. סתירה!

בכל מקרה הגענו לסתירה, לכן $L_D\notin R$. מ.ש.ל.

*אלכסון (באנגלית Diagonal) ...לכסון). בגלל שיטת ...כחה. ההוכחה כאן דומה למשפט קנטור ...וכיח שבין $[0,1]$ יש עוצמ... ...בת מנייה של מספרים.

*מכונה היא מחרוזת סופית.

| | $\langle M_1\rangle$ | $\langle M_2\rangle$ | $\langle M_3\rangle$ | $\langle M_4\rangle$ | $\ldots$ |
|---|---|---|---|---|---|
| $M_1$ | 0 | 1 | 1 | 0 | |
| $M_2$ | 0 | 1 | 0 | 1 | |
| $M_3$ | 0 | 0 | 1 | 0 | |
| $M_4$ | 1 | 0 | 1 | 0 | |
| $\vdots$ | | | $\vdots$ | | $\ddots$ |

## ד) רדוקציות

יהיו שני שפות $L_1,L_2\subseteq\Sigma^*$, נאמר ש-$L_1$ ניתנת לרדוקציה ל-$L_2$, בסימון $L_1\le L_2$, אם קיימת פונקציה $f:\Sigma^*\to\Sigma^*$ המקיימת שלושה תנאים:

(1) הפונקציה מלאה, כלומר מוגדרת לכל קלט.

(2) ניתנת לחישוב - קיימת מכונת טיורינג שיכולה לחשב את $f$.

(3) תקפה - לכל $x\in\Sigma^*$ מתקיים: $x\in L_1 \iff f(x)\in L_2$.

הוכחת שני התנאים הראשונים הינה בד"כ די טריוויאלית ומספיק משפט אחד לכל תנאי כדי להוכיח אותם. התנאי השלישי הוא מרכז הוכחת נכונות הרדוקציה.

קיום פונקציית הרדוקציה בעצם אומר לנו שאם יכולנו להכריע את השפה $L_2$ אזי גם יכולנו להכריע את $L_1$, שכן כל קלט של $L_1$ ניתן להעביר לקלט של $L_2$ באמצעות פונקציית הרדוקציה ואז להכריע אם הקלט מתקבל או נדחה באמצעות המכונה המכריעה של $L_2$. במילים אחרות, $L_2$ קשה לפחות כמו $L_1$.

**משפט הרדוקציה:**

אם $L_1\le L_2$ אז:

- אם $L_2\in R$ אז גם $L_1\in R$
- אם $L_2\in RE$ אז גם $L_1\in RE$
- אם $L_2\in coRE$ אז גם $L_1\in coRE$

**משפט הרדוקציה ניסוח שקול:**

אם $L_1\le L_2$ אז:

- אם $L_1\notin R$ אז גם $L_2\notin R$.
- אם $L_1\notin RE$ אז גם $L_2\notin RE$.
- אם $L_1\notin coRE$ אז גם $L_2\notin coRE$.

באמצעות משפט הרדוקציה הרגיל והשקול ניתן להוכיח ששפה שייכת או לא שייכת למחלקה מסוימת, אומנם אנו לרוב נשתמש ברדוקציה כדי להוכיח ששפה אינה שייכת למחלקה כלשהי, וזאת מכיוון שכדי להוכיח ששפה שייכת למחלקה יש להראות קיום מכונת טיורינג מתאימה שזה בד"כ יותר קל להוכיח מרדוקציה.

תכונות: משפט הרדוקציה הוא רפלקסיבי, אנטי סימטרי וטרנזיטיבי. בנוסף אם $L_1\le L_2$ אזי גם $\overline{L_1}\le\overline{L_2}$.

**דוגמאות:**

(1) עבור השפה $HP=\{\langle M,x\rangle \mid$ M עוצרת על x$\}$ מתקיים $L_D\le HP$ וכתוצאה מכך $HP\notin R$. Halting Problem

$L_D=\{\langle M\rangle \mid$ M מקבלת $\langle M\rangle\}$

הוכחה: נגדיר את פונקציית הרדוקציה $f(\langle M\rangle)=\langle M'\rangle\langle M\rangle$, כאשר $M'$ על קלט $y$ מריצה את $M$ על $y$, אם $M$ קיבלה גם $M'$ מקבלת ואם $M$ דחתה $M'$ נכנסת ללולאה אינסופית. הפונקציה מלאה שכן מכל קידוד מכונה $\langle M\rangle$ ניתן לבנות את $M'$. והפונקציה גם ניתנת לחישוב מאחר שכל מכונת טיורינג ניתן לקודד כמחרוזת.

נוכיח תקפות: $\langle M\rangle\in L_D \Rightarrow M$ מקבלת את $\langle M\rangle \Rightarrow M'$ מקבלת את $\langle M\rangle \Rightarrow \langle M'\rangle\langle M\rangle\in HP$.

$\langle M\rangle\notin L_D \Rightarrow M$ לא מקבלת את $\langle M\rangle \Rightarrow M'$ נכנסת ללולאה על $\langle M\rangle \Rightarrow \langle M'\rangle\langle M\rangle\notin HP$.

* בדרך כלל כדי להוכיח ששפה $L$ לא שייכת ל-$R$ הכי קל הוא להוכיח רדוקציה $HP\le L$.

נ' ידוע ש- $\overline{HP}\notin RE$ ו- $HP\notin coRE$.

(2) עבור השפה $L_\infty=\{\langle M\rangle \mid |L(M)|=\infty\}$ מתקיים $HP\le L_\infty$ וגם $\overline{HP}\le L_\infty$ ולכן $L_\infty\in\overline{RE\cup coRE}$.

הוכחת $HP\le L_\infty$: נגדיר את פונקציית הרדוקציה $F(\langle M,x\rangle)=\langle M_x\rangle$ כאשר $M_x$ על קלט y מריצה את M על x (מתעלמת מ-y). אם השלב הראשון יסתיים מקבלת. הפונקציה מלאה וניתנת לחישוב מפני שמקבל קידוד של מכונה וקלט ניתן לבנות את $M_x$ וכל מכונת טיורינג ניתנת לקידוד כמחרוזת. נוכיח תקפות:

$\langle M,x\rangle\in HP \Leftarrow M$ עוצרת על $x \Leftarrow M_x$ מקבלת כל קלט $\Leftarrow$ יש $\infty$ קלטים $\Leftarrow \langle M_x\rangle\in L_\infty$.

$\langle M,x\rangle\notin HP \Leftarrow M$ לא עוצרת על $x \Leftarrow M_x$ לא מקבלת אף מילה $\Leftarrow \langle M_x\rangle\notin L_\infty$.

הוכחת $\overline{HP}\le L_\infty$: נגדיר את פונקציית הרדוקציה $F(\langle M,x\rangle)=\langle M'_x\rangle$ כאשר $M'_x$ על קלט y מריצה את M על x למשך $|y|$ צעדים. אם M לא עצרה בשלב 1 $M'_x$ מקבלת אחרת $M'_x$ דוחה. גם כאן כמובן הפונקציה מלאה וניתנת לחישוב, נוכיח תקפות.

$\langle M,x\rangle\in\overline{HP} \Leftarrow M$ לא עוצרת על $x \Leftarrow$ לכל y $M'_x$ לא תעצור ולכן תקבל $\Leftarrow M'_x$ מקבלת $\infty$ מילים $\Leftarrow \langle M'_x\rangle\in L_\infty$.

$\langle M,x\rangle\notin\overline{HP} \Leftarrow M$ עוצרת על x לאחר i צעדים $\Leftarrow M'_x$ תקבל כל y עבורו $|y|\le i \Leftarrow |L(M'_x)|=2^0+2^1+\dots+2^i$

$\Leftarrow |L(M'_x)|$ הוא מספר סופי $\Leftarrow \langle M'_x\rangle\notin L_\infty$.

$L_{\Sigma^*}=\{\langle M\rangle \mid L(M)=\Sigma^*\}$, $L_{\Sigma^*}\in\overline{RE\cup coRE}$

(3) עבור השפה $L_{eq}=\{\langle M_1\rangle\langle M_2\rangle \mid L(M_1)=L(M_2)\}$ מתקיים $L_{\Sigma^*}\le L_{eq}$ ולכן $L_{eq}\in\overline{RE\cup coRE}$.

הוכחה: נגדיר את פונקציית הרדוקציה $F(\langle M\rangle)=\langle M\rangle\langle M_{stam}\rangle$ כאשר $M_{stam}$ זו מכונה שמקבלת כל קלט. הפונקציה מלאה וניתנת לחישוב מאחר ורק מוסיפים מחרוזת קבועה בסוף הקלט לפונקציה. נוכיח תקפות.

$\langle M\rangle\in L_{\Sigma^*} \Leftarrow M$ מקבלת $\Sigma^* \Leftarrow M_{stam}$ מקבלת $\Sigma^* \Leftarrow \langle M\rangle\langle M_{stam}\rangle\in L_{eq}$.

$\langle M\rangle\in L_{\Sigma^*} \Leftarrow M$ לא מקבלת $\Sigma^* \Leftarrow M_{stam}$ מקבלת $\Sigma^* \Leftarrow \langle M\rangle\langle M_{stam}\rangle\notin L_{eq}$.

## י"א) משפט רייס

זהו משפט שבעזרתו ניתן לסווג שפות מהצורה $L=\{\langle M\rangle \mid L(M)$ מקיימת תכונה S כלשהי$\}$ אל המחלקות שבהן היא שייכת. תכונה בהקשר של משפט רייס היא קבוצה של שפות ב-RE שיש להן מכנה משותף כלשהו.

השפה $L_S$: בהינתן תכונה $S\subseteq RE$ נגדיר את השפה $L_S=\{\langle M\rangle \mid L(M)\in S\}$. כלומר לכל תכונה S יש שפה $L_S$ שהיא שפת קידודי כל המכונות כך ששפת המכונה שייכת ל-S. $L_S$ למעשה היא השפה אותה אנו מעוניינים להכריע באיזה מחלקה היא. לדוגמא, עבור $S=\{L\subseteq\Sigma^* \mid |L|\ge 2\}$ מתקיים: $L_S=\{\langle M\rangle \mid |L(M)|\ge 2\}$.

דוגמאות נוספות בשקופית בהרצאה 4.

משפט רייס מחלק בין תכונות טריוויאליות לתכונות לא טריוויאליות: תכונות טריוויאליות הן תכונות S כך ש $S=RE$ או $S=\emptyset$, ואילו תכונות לא טריוויאליות הן כל שאר התכונות $RE\ne S\ne\emptyset$. כלומר יש לפחות שפה אחת ב-RE שאינה בתכונה ויש לפחות שפה אחת ב-RE שכן בתכונה.

חשוב לשים לב שהתכונה היא על השפה של המכונה ולא של המכונה עצמה. אחרת, משפט רייס לא עובד. אמנם גם התכונה של המכונה שקולה לתכונה על שפת המכונה, כן ניתן להשתמש במשפט רייס.

<u>משפט ר"ס:</u> לכל תכונה לא טריוויאלית $\emptyset \neq S \subsetneq RE$ מתקיים $L_S \notin R$

משפט ר"ס המורחב: אם בנוסף $\varphi \notin S$ אזי גם $L_S \notin coRE$.

אך אם $\varphi \in S$ אזי גם $L_S \notin RE$.

אם S כן טריוויאלית: אם $S = RE$ אזי $L_S = \Sigma^*$ ולכן $L_S \in R$.

אם $S = \emptyset$ אזי $L_S = \varphi$ ולכן $L_S \in R$.

כלומר כאשר S טריוויאלית בכל מקרה $L_S \in R$ וכאשר S לא טריוויאלית בכל מקרה $L_S \notin R$.

$\varphi$ - השפה הריקה
$\emptyset$ - קבוצה ריקה של שפות.

<u>דוגמאות:</u>

3. דוגמאות שקופית 17 מצגת 11.

(1) $L_1 = \{\langle M\rangle \mid |L(M)| \le 3\}$ - כיוון ש-S אינה טריוויאלית $L_1 \notin R$. בנוסף $\varphi \in S$ ולכן $L_1 \notin coRE$. צריך רדוקציה לקבוע אם שייך RE.

(2) $L_2 = \{\langle M\rangle \mid$ M דוחה את המילה $\varepsilon\}$ - תכונה של המכונה ולא של שפת המכונה ולכן לא ניתן להשתמש במשפט ר"ס.

(3) $L_3 = \{\langle M\rangle \mid L(M) \subseteq \Sigma^*\}$ - זוהי תכונה טריוויאלית ולכן $L_3 \in R$.

ב) <u>Bounded Halting Problem - BHP</u>

השפה BHP מוגדרת כך: $BHP = \{\langle M,x\rangle \mid$ M עוצרת על x ולא נכנסת לתא ה-$f(|x|)$ של הסרט שלה$\}$.

כלומר BHP היא שפת כל המכונות M והקלטים x כך ש-M עוצרת על x תוך שהיא מוגבלת במספר התאים בסרט בהם היא יכולה להשתמש. מספר התאים הוא נתון ע"י פונקציה של $|x|$. נוכיח כי $BHP \in R$.

כדי להוכיח זאת נבנה מכונת טיורינג אוניברסלית U שתכריע את BHP. מכונה זו תבדוק בכל צעד אם M נכנסת לתא ה-$f(|x|)$ במידה וכן תדחה, ולאחר מכן תבדוק אם M עצרה - אם כן תקבל.

אמנם כיצד נדע אם M נכנסת ללולאה בתוך $f(|x|)-1$ התאים הראשונים?

נשים לב שכיוון שמספר התאים מוגבל כך גם מספר הקונפיגורציות האפשריות. לכן נחשב חסם לכמות זו, אותה נסמן ב-T, במידה ו-M תמשיך ביותר מ-T צעדים בהכרח הייתה באותה קונפיגורציה פעמיים מה שאומר שבוודאי היא נכנסה ללולאה. החסם פה הוא: $T = |\Gamma|^{f(|x|)-1} \cdot |Q| \cdot \overbrace{(f(|x|)-1)}^{\text{(מיקום של הראש)}}$.

קונפיגורציה: $(\alpha, q, i)$
$\alpha \le |\Gamma|^{f(|x|)-1}$
$q \le |Q|$
$i \le f(|x|)-1$

U על קלט $\langle M,x\rangle$:

- הרץ את M על x למשך $T+1$ צעדים
- בכל שלב: אם M נכנסה לתא ה-$f(|x|)$ - דחה.
  אם M עצרה על x - קבל.
- אם הסתיימו $T+1$ הצעדים ללא עצירה - דחה.

$\langle M,x\rangle \in BHP \Leftarrow$ M עוצרת על x בלי לעבור את תא ה-$f(|x|)$ בתוך T צעדים לכל היותר $\Leftarrow \langle M,x\rangle \in U$.

$\langle M,x\rangle \notin BHP \Leftarrow$ או ש-M נכנסת לתא ה-$f(|x|)$ או שנכנסה ללולאה ועשתה יותר מ-$T+1$ צעדים $\Leftarrow \langle M,x\rangle \notin U$.

## יג) שפות שלמות

שפה שלמה $L$ במחלקה כלשהי, נניח $RE$ אך נכון גם עבור $R$ ו- $coRE$ היא שפה המקיימת:

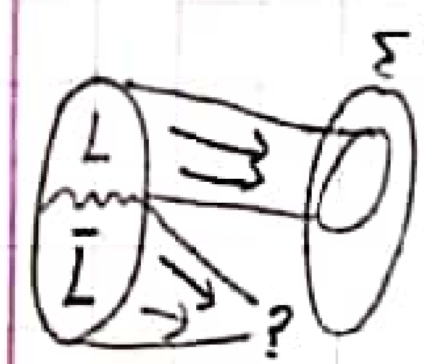
- $L \in RE$.
- $L' \leq L$ לכל $L' \in RE$.

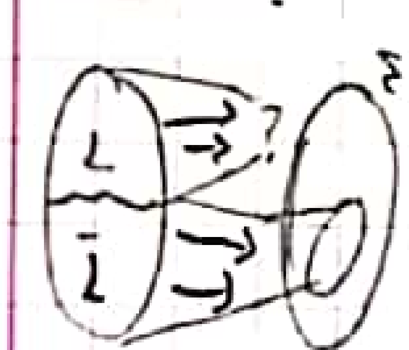
במידה ו- $L$ מקיימת שני תנאים אלו נסמן $L \in RE\text{-complete}$ כאשר $RE\text{-complete}$ היא קבוצת כל השפות השלמות ב- $RE$. שפה שלמה היא השפה הכי "קשה" מבחינה חישובית במחלקה שלה.

<u>משפט:</u> אם נתון כי $L^* \in RE\text{-complete}$ ונתונה שפה נוספת $L$ המקיימת $L \in RE$ וגם $L^* \leq L$, אזי גם $L \in RE\text{-complete}$. נכונות הטענה נובעת מתכונת הטרנזיטיביות של הרדוקציה.

- במחלקה $RE$ השפות $L_u = \{<M,x> \mid x \in L(M)\}$ ו- $HP = \{<M,x> : M \text{ stops on } x\}$ הן שפות שלמות ב- $RE$.
- במחלקה $R$ כל השפות הן שלמות חוץ מ- $\Sigma^*$ ו- $\varphi$. $R\text{-complete} = R \setminus \{\Sigma^*, \varphi\}$.

הוכחה: $\Sigma^*$ ו- $\varphi$ לא יכולות להיות שלמות שכן לא ניתן לעשות רדוקציה אליהן מאף שפה. זאת מפני שבכל שפה אחרת יש מילים שבשפה ומילים שאינן בשפה, ולא ניתן למפות את שתי הקבוצות לאחת מהשפות $\Sigma^*$ ו- $\varphi$. כעת נוכיח שכל שפה אחרת ב- $R$ שלמה. בהינתן שתי שפות $L_1 \in R$ ו- $L_2 \in R \setminus \{\Sigma^*, \varphi\}$ נוכיח $L_1 \leq L_2$. כיוון ש- $L_2$ לא טריוויאלית נבחר שני מילים $x \in L_2$ ו- $y \notin L_2$. פונקציית הרדוקציה לכל $w \in \Sigma^*$ היא: $f(w) = \begin{cases} x & w \in L_1 \\ y & w \notin L_1 \end{cases}$. הפונקציה ניתנת לחישוב שכן $L_1 \in R$ ולכן יש מכונת טיורינג $M_1$ שיכולה לבדוק האם $w \in L_1$. הפונקציה כמובן גם תהיה טריוויאלית.

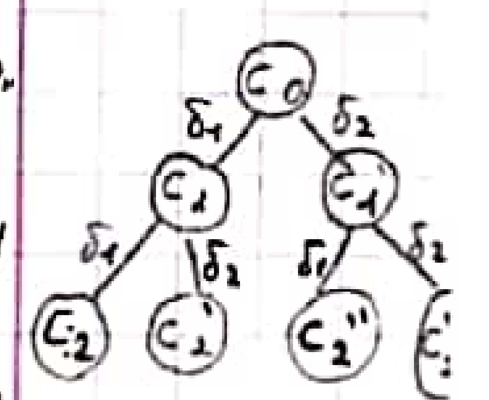



- במחלקה $RE \cup coRE$ לא יכולה להיות שפה שלמה, שכן אם כן אז נקבל סתירה (מצגת 5 שקופית 19).

## יד) מכונת טיורינג אי-דטרמיניסטית

כמו מכונת טיורינג דטרמיניסטית אלא שקיימות בה שתי פונקציות מעברים $\delta_1$ ו- $\delta_2$. ובכל צעד המכונה יכולה לבחור מבין שתי האפשרויות איזה מעבר לעשות. ניתן לראות את כל מסלולי החישוב האפשריים של מכונת טיורינג על מילה כלשהי כעץ בינארי, בעוד במכונה הדטרמיניסטית יש רק מסלול חישוב אחד. נאמר שמכונת טיורינג א"ד מקבלת מילה אם יש לפחות מסלול חישוב אחד מקבל. נשים לב שאם בהרצה כלשהי המכונה דחתה אין זה אומר שהמילה לא שייכת למכונה שכן במסלול אחר כן יכולה לקבל.



<u>מכונת טיורינג א"ד מקבלת שפה $L$:</u> עבור כל $x \in L$ יש לפחות מסלול אחד מקבל, ועבור $x \notin L$ כל המסלולים או דוחים או נכנסים ללולאה.

<u>מכונת טיורינג א"ד מכריעה שפה $L$:</u> עבור כל $x \in L$ יש לפחות מסלול אחד מקבל, ועבור $x \notin L$ כל המסלולים דוחים.

* חלוקת קבוצות השפות שפתן מכונה א"ד מקבלת ומכריעה מקבילה לחלוקה של $R$ ו- $RE$ במכונה דטרמיניסטית.

## שקילות המודלים

כיוון 1 - ממכונה דטרמיניסטית $M$ ניתן לבנות מכונת טיורינג א"ד $M'$ שקולה על ידי הגדרת $\delta_2=\delta_1=\delta$ כך שכל תוצאה ב-$M$ תהיה גם ב-$M'$. ניתן לראות הוכחה מלאה במצגת 6 שקופיות 10-13.

כיוון 2 - ממכונה א"ד $N$ ניתן לבנות מכונה דטרמיניסטית $N'$. $N'$ תסמלץ את ריצת $N$ על כל קלט $x\in\Sigma^*$ בכך שתבנה את עץ החיפוש של $N$ ותסריק אותו לפי שכבות באמצעות BFS. אם $N'$ מוצאת מצב מקבל תעצור ותקבל את $x$, אם עוברת על כל העץ ולא מוצאת מצב מקבל - תדחה, אם העץ אינסופי $N'$ תיכנס ללולאה. הסריקה של BFS לפי שכבות הכרחית כדי למנוע מצב שנתקעים בענף אינסופי לפני שיש מצב מקבל.

$x\in L(N) \Rightarrow$ יש מסלול מקבל בעץ החישוב בעומק $t \Rightarrow$ הסריקה של $N'$ בעומק $t$ ימצא מסלול זה $\Rightarrow x\in L(N')$.

$x\notin L(N) \Rightarrow$ כל המסלולים דוחים או אינסופיים $\Rightarrow N'$ לא תמצא מצב מקבל ולכן תדחה או תיכנס ללולאה $\Rightarrow x\notin L(N')$.

## שימושי המודל האי-דטרמיניסטי

במכונת טיורינג דטרמיניסטית כאשר רצינו למצוא מילה מסויימת מבין כל המילים ב-$\Sigma^*$ היינו צריכים לעשות הרצה מבוקרת על $\Sigma^*$ כדי למצוא אותה. במכונת טיורינג א"ד אין צורך לעשות הרצה מבוקרת במידה וזה שכן יכולה "לנחש" עבורי אותה מילה. כיצד היא עושה זאת? ניתן להסתכל על עץ החישוב כעל עץ בינארי שבו כל פנייה שמאלה רושמת 0 ופנייה ימינה 1, בפועל המכונה יכולה לעצור לאחר כל צעד. ולכן עץ החישוב בעצם מכיל את $\Sigma^*$, וכיוון שאם עבור מסלול אחד יש מצב מקבל המכונה מקבלת, ניתן להניח שהמכונה הא"ד "מנחשת" את אותה מילה. בהוכחות מספיק לרשום שהמכונה מנחשת מילה $x$.

דוגמא: נוכיח שהשפה הבאה שייכת ל-RE. $L=\{\langle M\rangle \mid \exists x\in\Sigma^*: f_M(x)=x\}\in RE$. שפה כל מכונות שבפונקציה שמחשבות יש נק' שבת. מכונה $N$ על קלט $\langle M\rangle$:

- מנחשת מילה $x$.
- מסמלצת $M$ על $x$.
- אם פלטה $x$ - קבל, אחרת - דחה.

$\langle M\rangle\in L \Rightarrow$ יש מילה $x$ עבורה $f_M(x)=x \Rightarrow N$ תנחש נכון $\Rightarrow \langle M\rangle\in L(N)$.

$\langle M\rangle\notin L \Rightarrow$ אין מילה $x$ מתאימה $\Rightarrow$ ב-$N$ כל המסלולים דוחים או אינסופיים $\Rightarrow \langle M\rangle\notin L(N)$.