מכונת טיורינג לא דטרמיניסטית

מכונות לא דטרמיניסטיות

- למדנו על מודלים לא דטרמיניסטיים
  - אוטומט סופי לא דטרמיניסטי שקול בכוחו לאוטומט סופי דטרמיניסטי
  - אוטומט-מחסנית לא דטרמיניסטי חזק יותר מאוטומט-מחסנית דטרמיניסטי

מכונות לא דטרמיניסטיות

- למדנו על מודלים לא דטרמיניסטיים
  - אוטומט סופי לא דטרמיניסטי שקול בכוחו לאוטומט סופי דטרמיניסטי
  - אוטומט-מחסנית לא דטרמיניסטי חזק יותר מאוטומט-מחסנית דטרמיניסטי
- כעת נגדיר מהי מכונת טיורינג לא דטרמיניסטית
  - במצב נתון בו מכונה נמצאת במצב q, והראש הקורא-כותב שלה קורא את הסמל a, היא יכולה לבצע אחד מכמה צעדים אפשריים.

מכונת טיורינג לא דטרמיניסטית

- פונקצית המעברים של מכונה לא דטרמיניסטית:
  $$\delta: Q' \times \Gamma \rightarrow P(Q \times \Gamma \times \{L, S, R\})$$
- **הערה**: לפי ההגדרה הפורמלית, אפשרית גם קבוצה ריקה של צעדים אפשריים, אבל אנחנו נניח שתמיד יש לפחות צעד אפשרי אחד
  - שיכול להיות כניסה ל-q_{reject} אם התכוונו לסיים בדחייה
  בשקף הבא נבין למה דווקא דחייה היא ברירת המחדל.

שפה של מכונת טיורינג לא דטרמיניסטית

- נאמר שמכונת טיורינג לא דטרמיניסטית N **מקבלת** מילה x אם בריצה של N על x **קיימת** סדרת צעדים חוקית המסתיימת במצב q_{acc}.

שפה של מכונת טיורינג לא דטרמיניסטית

- נאמר שמכונת טיורינג לא דטרמיניסטית N **מקבלת** מילה x אם בריצה של N על x **קיימת** סדרת צעדים חוקית המסתיימת במצב q_{acc}.
- השפה של מכונת טיורינג לא דטרמיניסטית N מוגדרת בדומה לשפה של מכונת טיורינג דטרמיניסטית:

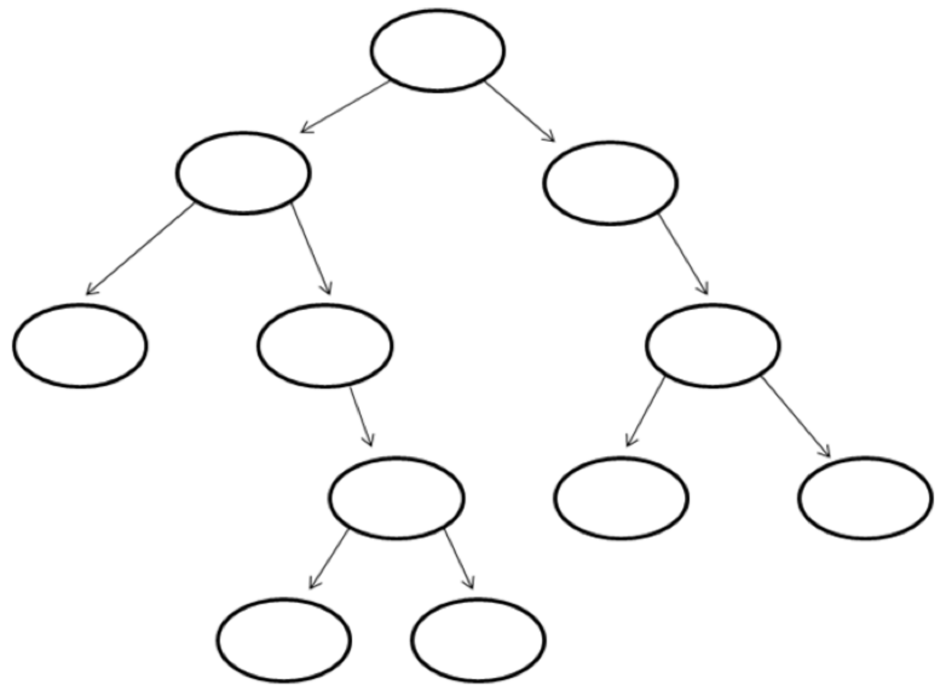
$$L(N) = \{x \in \Sigma^* | N \ accepts \ x\}$$

תזכורת חשובה

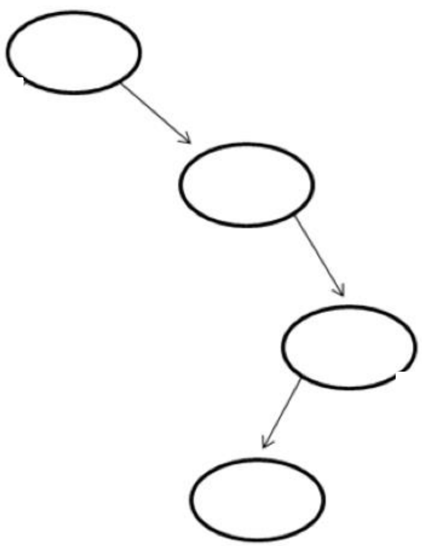
- **דגש**: הכוח של מודלים לא דטרמיניסטיים נובע מן ההגדרה שמילה מתקבלת אם יש לה מסלול חישוב מקבל אחד לפחות (מסלול חישוב שמסתיים במצב המקבל)
  - זה תקף גם אם מסלולי חישוב רבים (ייתכן שאינסוף מסלולי חישוב) אינם מסתיימים במצב מקבל.
- **שאלה לדיון קצר**: מתי מילה לא מתקבלת על-ידי מכונה לא דטרמיניסטית?

מסלול לעומת עץ

- עץ חישוב :



מסלול חישוב :



מסלול לעומת עץ

- אפשר להסתכל על חישוב של מכונת טיורינג דטרמיניסטית כעל מסלול במרחב הקונפיגורציות האפשריות:
  - מתחילים בקונפיגורציה ההתחלתית.
  - לכל קונפיגורציה ייתכנו מספר קונפיגורציות אפשריות עוקבות, אך 'תיבחר' קונפיגורציה עוקבת אחת לכל היותר.

מסלול לעומת עץ

- אפשר להסתכל על חישוב של מכונת טיורינג דטרמיניסטית כעל מסלול במרחב הקונפיגורציות האפשריות:
  - מתחילים בקונפיגורציה ההתחלתית.
  - לכל קונפיגורציה יש קונפיגורציה עוקבת אחת לכל היותר.
- חישוב של מכונה לא דטרמיניסטית, לעומת זאת, הוא עץ במרחב הקונפיגורציות:
  - תיתכן יותר מקונפיגורציה עוקבת אחת לכל קונפיגורציה

אי סימטריה של מכונות לא דטרמיניסטיות

אבחנה חשובה: אם הרצנו מכונה לא דטרמיניסטית על קלט x יש כמה אפשרויות:

- המכונה קיבלה.

- אפשר להסיק ש-x שייך לשפת המכונה.

אי סימטריה של מכונות לא דטרמיניסטיות

אבחנה חשובה: אם הרצנו מכונה לא דטרמיניסטית על קלט x יש כמה אפשרויות:

- המכונה קיבלה.

- אפשר להסיק ש-x שייך לשפת המכונה.

- המכונה דחתה.

- אי אפשר לדעת אם x שייך לשפה או לא. (למה?)

אי סימטריה של מכונות לא דטרמיניסטיות

אבחנה חשובה : אם הרצנו מכונה לא דטרמיניסטית על קלט x יש כמה אפשרויות :

- המכונה קיבלה.

- אפשר להסיק ש-x שייך לשפת המכונה.

- המכונה דחתה.

- אי אפשר לדעת אם x שייך לשפה או לא. (למה?)

- המכונה לא עצרה.

-אנחנו אפילו לא יודעים אם אנחנו בלולאה או לא...

דוגמה

כדוגמה פשוטה, נסתכל על השפה

$$L = \{w \in \{0,1\}^* | \\ w\ starts\ and\ ends\ in\ the\ same\ bit\}$$

במודל הדטרמיניסטי, $L \ni 1001$ ו-$1000 \notin L$.
כנ"ל במודל האי-דטרמניסטי.

אבל, כדי לגלות כי $1000 \notin L$, נצטרך לנסות את כל מסלולי החישוב האפשריים של המכונה, ורק כשכולם דוחים נוכל לדעת כי $1000 \notin L$.

הערות

- מודל שקול למכונת טיורינג לא דטרמיניסטית – אי דטרמיניזם מוגבל:

$$\delta: Q' \times \ \Gamma \to (Q \times \Gamma \times \{L,S,R\})^2$$

- נח לחשוב על עץ ההרצות הזה בתור עץ בינארי ☺

הערות

- מודל שקול למכונת טיורינג לא דטרמיניסטית – אי דטרמיניזם מוגבל:

$$\delta: Q' \times \ \Gamma \to (Q \times \Gamma \times \{L,S,R\})^2$$

- נח לחשוב על עץ ההרצות הזה בתור עץ בינארי ☺
- שימו לב שכאשר מריצים מכונה בהרצה בודדת, המכונה רצה על מסלול חישוב **יחיד** המוגדר לפי הבחירות שלה בכל הצמתים. היא לא רצה במקביל בכל העץ.

הכרעת שפות ע"י מכונה לא דטרמיניסטית

<u>הגדרה</u>: מכונת טיורינג אי דטרמיניסטית N **מקבלת** שפה L אם:

- **לכל** $x \in L$ **קיים** מסלול מקבל בעץ החישוב של N על x.
- **לכל** $x \notin L$ **לא קיים** מסלול מקבל בעץ החישוב של N על x (יכולים להיות מסלולים אינסופיים).

הכרעת שפות ע"י מכונה לא דטרמיניסטית

<u>הגדרה</u>: מכונת טיורינג אי דטרמיניסטית N **מקבלת** שפה L אם:

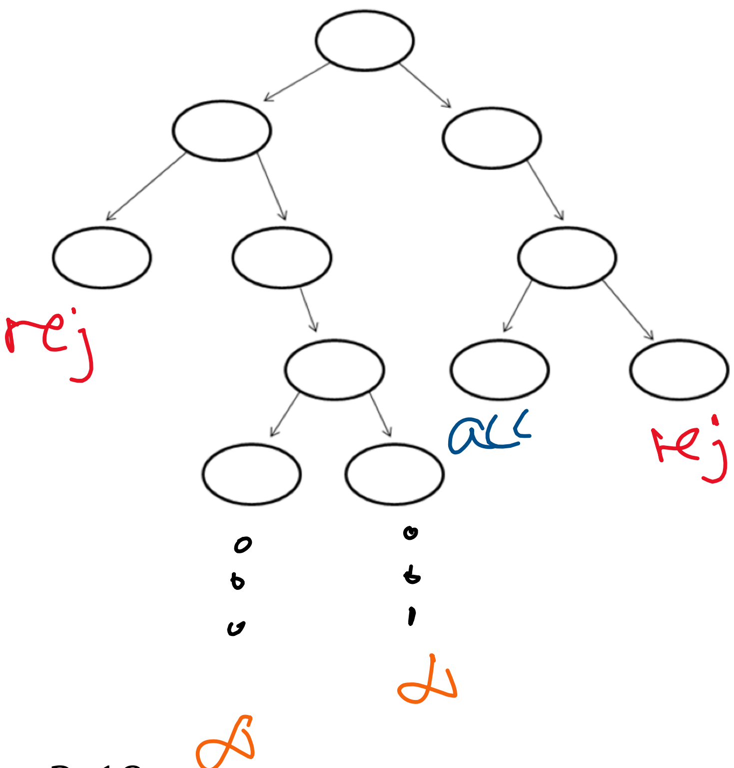
- **לכל** $x \in L$ **קיים** מסלול מקבל בעץ החישוב של N על x.
- **לכל** $x \notin L$ **לא קיים** מסלול מקבל בעץ החישוב של N על x (יכולים להיות מסלולים אינסופיים).

<u>הגדרה</u>: מכונת טיורינג אי דטרמיניסטית N **מכריעה** שפה L אם:

- **לכל** $x \in L$ **קיים** מסלול מקבל בעץ החישוב של N על x.
- בנוסף, לכל קלט, עץ החישוב של N הינו סופי.
- (במילים אחרות, לכל קלט $x \notin L$ כל המסלולים בעץ החישוב של N על x דוחים.)

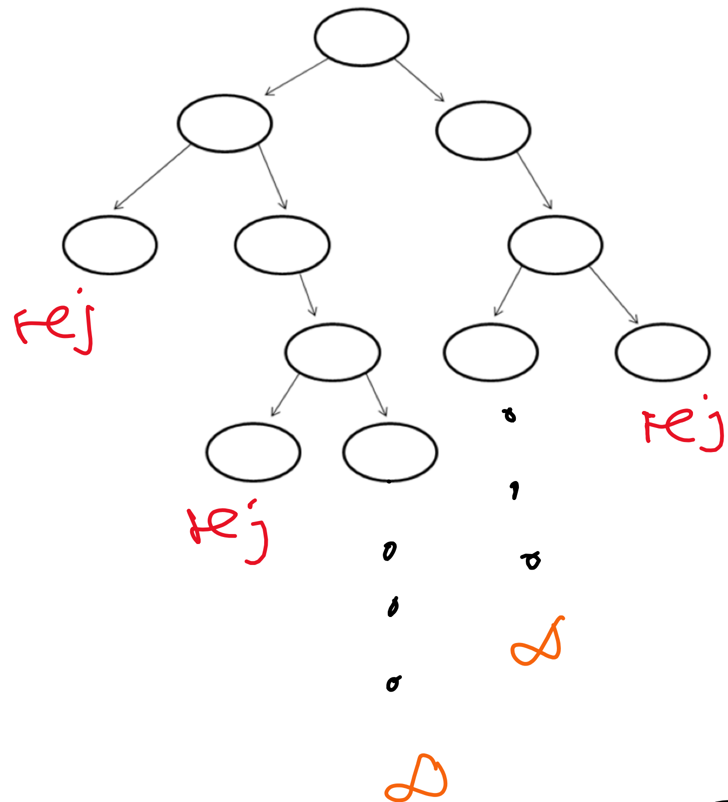
דוגמות לעצי חישוב במכונה מקבלת

$x \in L(M)$



לפחות מסלול 1 מקבל

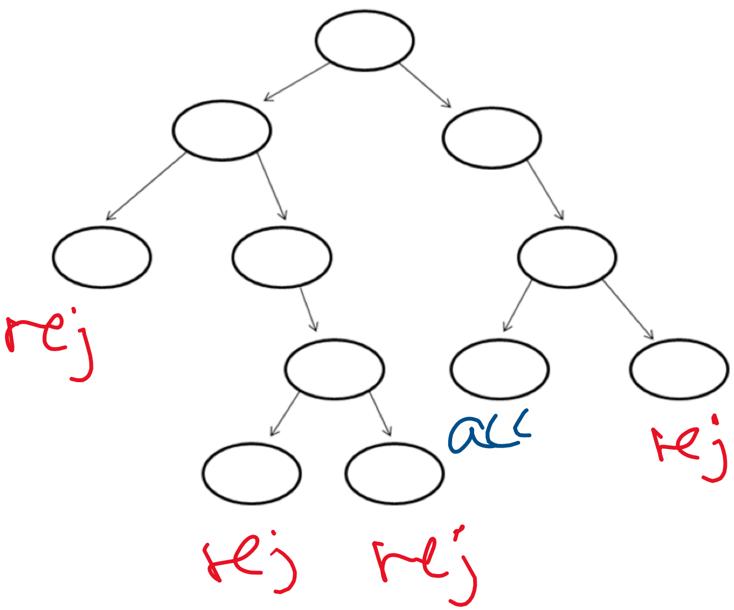
$x \notin L(M)$



כל מסלול לא מקבל

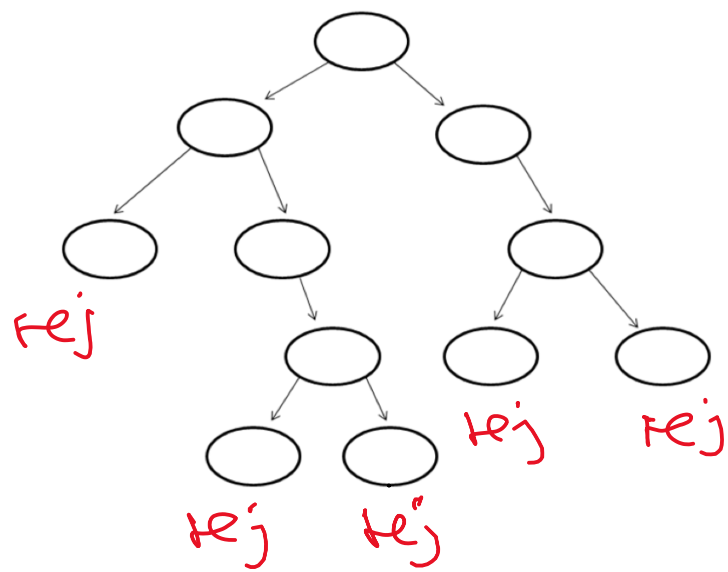
דוגמות לעצי חישוב במכונה מכריעה

$x \in L(M)$



לפחות מסלול 1 מקבל
היתר דוחים.

$x \notin L(M)$



∀ מסלול לא מקבל
∀ מסלול דוחה

מבנה הוכחת נכונות של שפת מכונה לא דטרמיניסטית

- מבנה ההוכחה:
- $L \ni x$ ← קיים מסלול מקבל בעץ החישוב של המכונה N על הקלט x.

עבור מכונה **מקבלת**:

- $x \notin L$ ← כל מסלולי החישוב בעץ החישוב של המכונה N על הקלט x דוחים או אינסופיים.

עבור מכונה **מכריעה**:

- $x \notin L$ ← כל מסלולי החישוב בעץ החישוב של המכונה N על הקלט x דוחים.

דוגמה

עבור מ״ט א״ד M ומילה x נגדיר :

$$L_4 = \{(M,x)\ |M\ has\ at\ least\ 4\ different\ accepting\ paths\ when\ running\ on\ x\}$$

בנו מ״ט U שתקבל את L_4

בניית U

בניית U

<u>U על קלט (<M>,x)</u> :

1. בחר 4 מחרוזות אקראיות.
2. וודא שהן שונות זו מזו.
3. סמלץ את M על x בהרצה מבוקרת לפי המחרוזות האקראיות.
4. אם ב-4 דרכים המילה התקבל, קבל.

משפט שקילות

- **משפט** : מכונת טיורינג לא דטרמיניסטית שקולה בכוח החישוב שלה למכונה דטרמיניסטית.

משפט שקילות

- **משפט** : מכונת טיורינג לא דטרמיניסטית שקולה בכוח החישוב שלה למכונה דטרמיניסטית.
- **הוכחה** : כיוון אחד פשוט - מכונה דטרמיניסטית היא מקרה פרטי של מכונה לא דטרמיניסטית. (כל האפשרויות הנוספות- זהות לאפשרות המקורית)

משפט שקילות

- **משפט** : מכונת טיורינג לא דטרמיניסטית שקולה בכוח החישוב שלה למכונה דטרמיניסטית.
- **הוכחה** : כיוון אחד פשוט - מכונה דטרמיניסטית היא מקרה פרטי של מכונה לא דטרמיניסטית. (כל האפשרויות הנוספות- זהות לאפשרות המקורית)
- להוכחת הכיוון השני, נסביר כיצד אפשר לבנות, לכל מכונה לא דטרמיניסטית N,
מכונה דטרמיניסטית D שתהיה שקולה ל-N

– D תעקוב אחרי כל מסלולי החישוב של המכונה N

הרעיון של הסימולציה

- **הרעיון**:
  - D – תנסה את כל ענפי החישוב של N על מילת הקלט w.
  - אם נמצא מסלול חישוב שמסתיים במצב המקבל (של N), D תקבל (תיכנס למצב המקבל שלה). –
  - אם כל מסלולי החישוב של N מסתיימים במצב הדוחה (של N), D תדחה (תיכנס למצב הדוחה שלה). –
  - אם יש ל-N מסלולי חישוב לא עוצרים, ואין לה מסלול חישוב מקבל, D לא תעצור. –

חיפוש לרוחב ולא לעומק

- חישוב של N על w הוא עץ במרחב הקונפיגורציות
  - כל צומת בעץ הזה הוא קונפיגורציה.
  - שורש העץ הוא הקונפיגורציה ההתחלתית של N על w
  - כל ענף בעץ הזה הוא חישוב חלקי אפשרי של N על w.

חיפוש לרוחב ולא לעומק

- חישוב של N על w הוא עץ במרחב הקונפיגורציות
  - כל צומת בעץ הזה הוא קונפיגורציה.
  - שורש העץ הוא הקונפיגורציה ההתחלתית של N על w
  - כל ענף בעץ הזה הוא חישוב חלקי אפשרי של N על w.
- **תרגיל**: הסבירו למה המכונה הדטרמיניסטית D לא תנסה לבצע חיפוש לעומק בעץ הקונפיגורציות?

חיפוש לרוחב ולא לעומק

- חישוב של N על w הוא עץ במרחב הקונפיגורציות
  - כל צומת בעץ הזה הוא קונפיגורציה
  - שורש העץ הוא הקונפיגורציה ההתחלתית של N על w
  - כל ענף בעץ הזה הוא חישוב חלקי אפשרי של N על w.
- **תרגיל**: הסבירו למה המכונה הדטרמיניסטית D לא תנסה לבצע חיפוש לעומק בעץ הקונפיגורציות?
- מיד נראה שחיפוש לרוחב בעץ הזה כן יהיה טוב

המכונה הדטרמיניסטית

- למכונה D יהיו שלושה סרטים
  - סרט הקלט שעליו רשומה מילת הקלט
    - זהו סרט לקריאה בלבד (לא משנים בו דבר)
  - סרט הסימולציה (לחיקוי מסלול חישוב חלקי של N)
  - סרט הכתובות שיציין את מסלול הבחירות הלא דטרמיניסטיות של החישוב החלקי הנוכחי

סרט הכתובות

- לכל צומת פנימי בעץ החישוב יש לכל היותר *2* בנים
  - לכן מילה מעל האלפבית $\{0,1\}$ יכולה לציין מסלול חישוב חלקי מהשורש אל צומת בעץ
    - זוהי ה"כתובת" של הצומת הזה בעץ

סרט הכתובות

- לכל צומת פנימי בעץ החישוב יש לכל היותר *2* בנים
  - לכן מילה מעל האלפבית {0,1} יכולה לציין מסלול חישוב חלקי מהשורש אל צומת בעץ
    - זוהי ה"כתובת" של הצומת הזה בעץ
    - למשל, ה"כתובת" 10010 אומרת:

בצעד החישוב הראשון בוחרים באפשרות הלא דטרמיניסטית השנייה של הצעד הזה (עוברים לבן השני)

בצעד החישוב השני בוחרים באפשרות הראשונה (הבן הראשון)

בצעד החישוב השלישי בוחרים באפשרות הראשונה (הבן הראשון)

בצעד החישוב הרביעי בוחרים באפשרות השנייה (הבן השני)

בצעד החישוב החמישי בוחרים באפשרות הראשונה (הבן הראשון)

כתובות לא חוקיות

- ייתכנו כתובות לא חוקיות
  - מאחר ולקונפיגורציה ההתחלתית יש רק שתי קונפיגורציות עוקבות (לשורש העץ יש שני בנים), אז כל כתובת שמתחילה ב-2 ומעלה איננה חוקית
- המכונה D תתייחס לכתובת לא חוקית כאל ענף חישוב שהסתיים בדחייה

הסימולציה - אתחול

- בתחילת הריצה של D
  – מילת הקלט רשומה על סרט 1 שהוא סרט הקלט
  – סרט 2 (סרט הסימולציה) וסרט 3 (סרט הכתובות) ריקים (רווחים בכל הריבועים)

הסימולציה - המשך

- שלב בסימולציה:
  - מעתיקים את תוכן סרט 1 (הקלט) לסרט 2
  - מבצעים את הצעדים של N על סרט 2 לפי הבחירות הלא דטרמיניסטיות הרשומות בסרט 3 (לפי הכתובת)
  - אם N הגיעה למצב המקבל שלה, מקבלים
    - D נכנסת למצב המקבל שלה, והחישוב מסתיים
  - אחרת, עוברים לבצע סימולציה של ענף החישוב הבא

הסימולציה - הסברים

- מה פירוש "אחרת"? מה יכול לקרות?
  - D סיימה לבצע את הצעדים שהיו רשומים בכתובת ולא הגיעה למצב של עצירה.
  - יתכן כי N נכנסה למצב הדוחה שלה
    - באיזשהו צעד בענף החישוב הנוכחי
  - או הכתובת שבסרט השלישי לא חוקית
- איך עוברים לענף החישוב הבא?
  - מוחקים את תוכן סרט הסימולציה
  - עוברים לכתובת הבאה בסרט הכתובות
  - מתחילים את הסימולציה של ענף החישוב הבא

מסקנה מן השקילות

- **מסקנה** : כדי להוכיח ששפה L היא ב-RE, אפשר לתאר מכונה לא דטרמיניסטית שמקבלת את L
  - משום שלכל מכונה כזו אפשר לבנות מכונה דטרמיניסטית שתקבל את L
- באותו אופן כדי להוכיח ששפה L היא ב-R, אפשר לתאר מכונה לא דטרמיניסטית שמכריעה את L
  - עץ החישוב במקרה זה יהיה **סופי לכל קלט**

הערה על זמן הריצה

- המכונה D שתיארנו לא יעילה מבחינת זמן הריצה
  - יש בה חזרות על חלקים של חישובים

הערה על זמן הריצה

- המכונה D שתיארנו לא יעילה מבחינת זמן הריצה
  - יש בה חזרות על חלקים של חישובים
- אפשר לייעל את פעולתה של D, אך עדיין זמן הריצה של כל סימולציה מוכרת של מכונה לא דטרמיניסטית על-ידי מכונה דטרמיניסטית הוא לפחות מעריכי (אקספוננציאלי) בזמן הריצה של המכונה הלא דטרמיניסטית
  - זמן הריצה של מכונה לא דטרמיניסטית מכריעה N על מילה w מוגדר כמספר הצעדים במסלול החישוב הארוך ביותר של N על w.

שימושי המודל הלא דטרמיניסטי

- לכאורה, אפשר לשאול, למה זה טוב? זה הרי מודל מאוד לא מציאותי ולא כ״כ שימושי - מספר המסלולים הדוחים יכול להיות גדול.

שימושי המודל הלא דטרמיניסטי

- לכאורה, אפשר לשאול, למה זה טוב? זה הרי מודל מאוד לא מציאותי ולא כ"כ שימושי - מספר המסלולים הדוחים יכול להיות גדול.
- אחד היתרונות של מכונה לא דטרמיניסטית הוא שהיא יכולה **לנחש מחרוזות**. לא נראה באופן פורמלי, מספיק שנאמר שהמכונה מנחשת מחרוזת כלשהי.

  בעזרת ניחוש אפשר לחסוך הרצה מבוקרת.
- נראה דוגמא למכונה כזו:

דוגמה

- לכל מט״ד M נגדיר את הפונקציה המוזרה הבאה:

$f_M(\Sigma^*) \to \Gamma^*$:

לכל $x \in \Sigma^*$ נרשום $f_M(x)$= הפלט של M על x.

- היות ו-M לא בהכרח עוצרת לכל קלט, הפלט $f_M(x)$ לא בהכרח מוגדר לכל קלט.

דוגמא לשימוש במכונה לא דטרמיניסטית

$$L = \{< M > | M \; is \; a \; function \; mechine, \\ and \; \exists x \in \Sigma^* : f_M(x) = x\}$$

<u>במילים</u> : M היא מכונת טיורינג לחישוב פונקציה (לאו דווקא מכונה לקבלת שפות), וקיימת לפונקציה של המכונה נקודת שבת. (כלומר קיימת מילה x כך ש- $f(x) = x$)

דוגמא לשימוש במכונה לא דטרמיניסטית

$$L = \{< M > | M\ is\ a\ function\ mechine, \\ and\ \exists x \in \Sigma^* : f_M(x) = x\}$$

<u>במילים</u>: M היא מכונת טיורינג לחישוב פונקציה (לאו דווקא מכונה לקבלת שפות), וקיימת לפונקציה של המכונה נקודת שבת. (כלומר קיימת מילה x כך ש- $f(x) = x$)

- כדי להוכיח שייכות של השפה הזו ל-RE במודל הדטרמיניסטי היינו צריכים לבצע הרצה מבוקרת חזקה של כל המילים בסיגמא כוכבית.
- לעומת זאת, בעזרת ניחוש, אפשר לחסוך את ההרצה המבוקרת.

דוגמא לשימוש במכונה לא דטרמיניסטית

$$L = \{< M > | \exists x \in \Sigma^* : f_M(x) = x\} \in RE$$

הוכחה:

נתאר מכונת טיורינג לא דטרמיניסטית N המקבלת את L :

דוגמא לשימוש במכונה לא דטרמיניסטית

$$L = \{< M > | \exists x \in \Sigma^* : f_M(x) = x\} \in RE$$

הוכחה:

נתאר מכונת טיורינג לא דטרמיניסטית N המקבלת את L:

N על קלט $< M >$:

1. מנחשת מילה x.
2. מסמלצת את ריצת M על x.
3. אם M פלטה x – קבל, אם פלטה מחרוזת שונה מ- x -דחה.

דוגמא לשימוש במכונה לא דטרמיניסטית

נוכיח את נכונות המכונה N, כלומר נוכיח כי $\boldsymbol{L(N)=L}$:

דוגמא לשימוש במכונה לא דטרמיניסטית

נוכיח את נכונות המכונה N, כלומר נוכיח כי ***L(N)=L***:

- אם $\boldsymbol{L}$ $\boldsymbol{< M >\in}$ אזי קיימת מילה $x_0 \in \Sigma^*$ כך שהפלט של M בריצתה על x_0 הוא בדיוק x_0. במקרה כזה בעץ החישוב של N על $< M >$, קיים מסלול שבו הניחוש של N הוא בדיוק המילה x_0, מכאן שהמסלול המדובר הוא **מסלול מקבל** ולכן
$$< M >\in L(N)$$

דוגמא לשימוש במכונה לא דטרמיניסטית

נוכיח את נכונות המכונה N, כלומר נוכיח כי ***L(N)=L***:

- אם $\boldsymbol{<M>\in L}$ אזי קיימת מילה $x_0\in\Sigma^*$ כך שהפלט של M בריצתה על x_0 הוא בדיוק x_0. במקרה כזה בעץ החישוב של N על $<M>$, קיים מסלול שבו הניחוש של N הוא בדיוק המילה x_0, מכאן שהמסלול המדובר הוא **מסלול מקבל** ולכן
$<M>\in L(N)$
- אם $\boldsymbol{<M>\notin L}$ אזי **לכל** $x\in\Sigma^*$ מתקיים $f_M(x)\neq x$.

לכן בעץ החישוב של N על $<M>$, לכל בחירה של מילה $x\in\Sigma^*$, המכונה M פולטת לא את x, לכן כל המסלולים של N דוחים או אינסופיים, לכן $<\mathrm{M}>\notin L(N)$.