חישוביות ומבוא לסיבוכיות

דורון מור- arch7486730@gmail.com

שני שוב- abhauc97@gmail.com

ניר סון- nir16832@gmail.com

מנהלות

שעות קבלה יפורסמו במודל.

ציון הקורס ייבנה מ 85% ציון מבחן ו- 15% ציון מטלות (מגן). יש לעבור את הבחינה בשביל לשקלל את המטלות. יהיו 2-3 מטלות במהלך הקורס.

(הסילבוס במודל הוא עבור סמסטר רגיל. פרט לשמות הסגל וסדר הנושאים, הכל זהה).

יהיו לנו שני חלקים עיקריים לקורס:

1. <u>חישוביות:</u> בעיות שניתן לפתור על ידי מחשב.

2. <u>סיבוכיות:</u> נחלק את עולם הבעיות שלנו לכמה מחלקות לפי סיבוכיות זמן הריצה שלהן. (אפשר לקרוא לחלק הזה "יעילות")

מודל חישובי

- מודל חישובי הוא מכונה תאורטית עם רכיבים מסוימים. למשל: גודל זיכרון, א"ב, סרט קלט.
- כשנדבר על מודלים חישוביים נדבר על אלגוריתם כללי לפתרון הבעיה.
- באוטומטים ראיתם מודלים חישוביים כמו אס"ד, ואוטומט מחסנית.
- המודל החישובי העיקרי שלנו בקורס יהיה מכונת טיורינג- עליו נרחיב בהמשך.

פסוקים

- בשקפים הבאים יהיו כמה דוגמות לגבי "פסוק".
- לא מדובר בפסוקים מהחומש או מהתנ"ך.

- פסוק שנעסוק בו הינו פונקציה בוליאנית בצורת CNF (המוכרת ממערכות ספרתיות כצורת POS).
לדוגמה:

$$(x_1 \vee \overline{x_2} \vee x_4) \wedge (x_2 \vee \overline{x_3} \vee x_4)$$

- "השמה מספקת" שנפגוש בשקפים הבאים – הצבת ערכי True, False למשתנים כך שהפסוק מקבל ערך True.

בעיה

אפשר להסתכל על שני סוגים של בעיות- בעיית קיימות ובעיית חיפוש.
בקיימות אנחנו רק רוצים לדעת אם יש פתרון, בחיפוש אנחנו רוצים לדעת- אם יש פתרון, מהו.

דוגמות:
- בהינתן גרף האם יש בו מסלול המילטוני? אם יש בו אז מהו?
- בהינתן פסוק האם הוא ספיק? אם כן אז מהי ההשמה המספקת שלו?
- בהינתן לוח סודוקו $n^2 \times n^2$, האם הוא פתיר? מהו הפתרון?

אנו נעסוק בעיקר בבעיות הקיימות.
נפגוש בבעיות האלו בהמשך וכמובן נגדיר במדויק את כל המונחים.

פתירות של בעיה

- פתירות של בעיה היא ביחס למודל חישובי כלשהו.
- כשנאמר שבעיה כלשהי לא פתירה על ידי מודל חישובי כלשהו, הכוונה שלנו היא שאין פתרון כללי לכל הבעיה, אין הכוונה שאין אלגוריתם היודע לפתור מקרים פרטיים.

  למשל, בהינתן פסוק כלשהו, נרצה לדעת אם הוא ספיק או לא.
  אנחנו לא מחפשים לדעת עבור פסוק ספציפי אם יש פתרון, אלא אלגוריתם שבהינתן כל פסוק שהוא יקבל, ידע לענות אם קיימת השמה מספקת.

פתירות של בעיה

- גם בעיות פתירות ניתן לסווג לבעיות 'קלות' ובעיות 'קשות'.
- נסווג אותן לפי זמן הריצה המינימלי הנדרש כדי לפתור אותן, כאשר זמן ריצה הוא פונקציה של אורך הקלט. למשל עבור פסוק, זמן הריצה יחושב עבור n משתניו.

- נאמר שבעיות 'קלות' הן בעיות שיש אלגוריתם שיודע לפתור את הבעיה בזמן ריצה פולינומי.
- נאמר שבעיות 'קשות' אם כל אלגוריתם צריך זמן ריצה לפחות אקספוננציאלי בגודל הקלט.

כדי להראות שבעיה היא קלה- מספיק להראות קיימות של אלגוריתם בעל זמן ריצה פולינומי (ולהוכיח את זמן הריצה)

כדי להראות שבעיה היא קשה יש להוכיח שלא קיים אלגוריתם פולינומי שיפתור אותה.

ייצוג בעיה על ידי שפה

מכונות טיורינג יעבדו על גרפים, פסוקים, פולינומים ועוד, כאשר המכונה תקבל קלט קידוד של האובייקט עליו היא עובדת.

אובייקט יסומן לרוב ב-X כאשר הקידוד שלו יסומן כ-$< X >$.

כשנרצה לעסוק בבעיה כלשהי, לדוגמה בעיית קשירות של גרף, המכונה תקבל קידודי גרפים שונים והיא תצטרך לפעול עליהם.

לדוגמה, מכונת טיורינג שתבדוק אם גרף G כלשהו קשיר, תקבל כקלט את קידוד הגרף, $< G >$.

מודל החישוב – מכונת טיורינג

טז

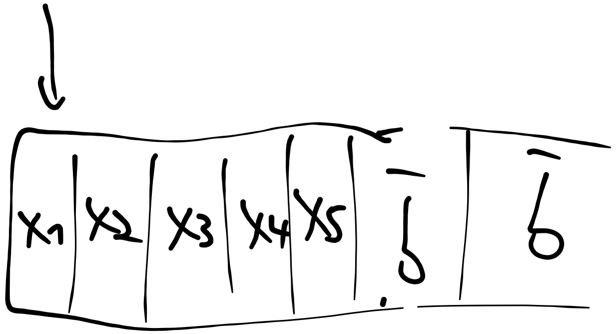
למה שנדבר בכלל על מכונה שטיורינג המציא בשנות ה-40?

- כי נוכיח בהמשך שהיא שקולה למחשב. כלומר, כל מה שניתן לבצע ביעילות במ"ט, נוכל לבצע ביעילות גם במחשב.
- כי יהיה יותר נח לדבר עליה מאשר על מחשב.
- כי האלגוריתמים שנדבר עליהם כלליים, ולא רוצים להוריד אותם לרמת "שפת תכנות" כזו או אחרת.
- יש המודל יחסית פשוט (בטח בהשוואה למחשב), ויהיה קל להוכיח עליו טענות.
- כי יש למ"ט זיכרון אינסופי, שזה כבר נשמע טוב ביחס לתוכנות מחשב גדולות ☺
- כי המרצים רוצים לאמלל לנו את החיים ולא מוכנים שנתכנת.

הקדמה למכונת טיורינג

x $\{R, L, S\}$

TM



blank

- **מכונת טיורינג תסומן** M.
- **בעלת סרט חד צדדי אינסופי.**
- **קלט עבור מכונת טיורינג**, הוא מילה $x \in \Sigma^*$ המקודדת על הסרט.
- **ראש קורא כותב**, מצביע על תא בסרט, קורא את התו שנמצא בו וכותב עליו במידת הצורך. לא הולך יותר שמאלה מהתא הכי שמאלי.
- קבוצת **מצבי בקרה** (אי אפשר להיפטר מהם).

רכיבי מכונת טיורינג

- מכונת טיורינג היא שביעיה: $M = (Q, q_0, F, \Gamma, \Sigma, \bar{b}, \delta)$ כאשר:
- Q היא קבוצת מצבים סופית.
- q_0 הינו המצב ההתחלתי.
- $F \subseteq Q$ הינה קבוצת המצבים הסופיים.
- $\Sigma \subseteq \Gamma$ הינה א"ב הקלט. (בדר"כ בקורס $\Sigma = \{0,1\}$)
- Γ הינה א"ב עבודה. (בדר"כ בקורס $\Gamma = \{0,1,\bar{b}\}$)
- $\bar{b}$, נקרא "בלנק", סימון של תא ריק. (לפי הגדרה $\bar{b} \notin \Sigma$)
- δ היא פונקציית המעברים. $\delta: (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \{R,L,S\}$

פונקציית המעברים

$$\delta: \underbrace{(Q \setminus F)}_{\text{מצב נוכחי}} \times \underbrace{\Gamma}_{\text{אות הנקראת ע"י הראש}} \to \underbrace{Q}_{\text{המצב הבא}} \times \underbrace{\Gamma}_{\text{מה לכתוב בתא הנוכחי}} \times \underbrace{\{R, L, S\}}_{\text{האם ולאן להזיז את הראש}}$$

תכונות מכונת טיורינג

- בהגיעה למצב סופי היא עוצרת, היא עשויה גם לא להגיע לכזה לעולם,
  היא עשויה גם לא להגיע לכזה לעולם,
  היא עשויה גם לא להגיע לכזה לעולם,
  היא עשויה גם לא להגיע לכזה לעולם...
- פלט המכונה הוא מילה $y \in \Sigma^*$ והיא כל התווים על הסרט שמשמאל לראש.
- אם M לא עצרה על x, אזי הפלט שלה לא מוגדר. זה מאפשר חישוב פונקציות "לא מלאות".
- ישנן מ"ט המחשבות פונקציות וישנן המזהות שפות.
  כשאנחנו במכונה המחשבת פונקציה- נתמקד בפלט המכונה.
  כשאנחנו במכונה המזהה שפה- נתמקד במצב בו סיימנו (מקבל או דוחה).
- נסמן פונקציה של מ"ט M בתור $f_M(x)$.

דוגמת בנייה למכונה המזהה שפה

$L = \{x \in \{0,1\}^* \mid w = x1\}$

השפה: $x1 \in \{0,1\}^*$

פתרון: $M = (Q, q_0, F, \Gamma, \Sigma, \bar{b}, \delta)$

$Q \setminus F \quad \Gamma$	0	1	$\bar{b}$
q_0			
q_1			

דוגמת בנייה למכונה המזהה שפה

<u>השפה</u>: $x1 \in \{0,1\}^*$

<u>פתרון</u>: $M = (Q, q_0, F, \Gamma, \Sigma, \bar{b}, \delta)$

$Q \setminus F$ Γ	0	1	$\bar{b}$
q_0	$(q_0, 0, R)$		
q_1	$(q_0, 0, R)$		

דוגמת בנייה למכונה המזהה שפה

<u>השפה</u>: $x1 \in \{0,1\}^*$

<u>פתרון</u>: $M = (Q, q_0, F, \Gamma, \Sigma, \bar{b}, \delta)$

$\delta : (Q \setminus F) \times \Gamma \to Q \times \Gamma \times D$

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

$Q = F \cup \{q_0, q_1\}$

$Q \setminus F$ \ Γ	0	1	$\bar{b}$
q_0	$(q_0, 0, R)$	$(q_1, 1, R)$	$(q_{rej}, 0, S)$
q_1	$(q_0, 0, R)$	$(q_1, 1, R)$	$(q_{acc}, 1, S)$

$x1 \in \{0,1\}^*$

פתרון אחר: נרוץ עד $\bar{b}$ הראשון, נחזור
אחד שמאלה, ולפי האות הראשונה שם
נקבל/נדחה.

	0	1	$\bar{b}$
q_0	$q_0, 0, R$	$q_0, 1, R$	$q_1, \bar{b}, L$
q_1	$q_{rej}, 0, S$	$q_{acc}, 1, S$	$\emptyset$

דוגמת בנייה למכונה המזהה שפה

<u>השפה</u>: $x1 \in \{0,1\}^*$

<u>פתרון</u>: $M = (Q, q_0, F, \Gamma, \Sigma, \bar{b}, \delta)$

$Q \setminus F$ Γ	0	1	$\bar{b}$
q_0	$(q_0, 0, R)$	$(q_1, 1, R)$	$(q_{rej}, 0, R)$
q_1	$(q_0, 0, R)$	$(q_1, 1, R)$	$(q_{acc}, 1, R)$

דוגמת בנייה למכונה המחשבת פונקציה

$$f(x) = 0 \cdot x$$

$f(x) = 1 \cdot x$

$M = (\quad q_1 \quad)$

רעיון הבנייה: נרשום 0 בהתחלה, ובכל צעד נזכור האם לכתוב 0 או 1.

פורמלית:

$$M = (Q = \{q_0, q_1, q_{end}\}, q_0, F = \{q_{end}\}, \Gamma = \{0,1,\bar{b}\}, \Sigma = \{0,1\}, \bar{b}, \delta)$$

	$Q \backslash F$ $\quad \Gamma$	0	1	$\bar{b}$
צריך לכתוב 0	q_0	$q_0, 0, R$	$q_1, 0, R$	$q_{end}, 0, R$
צריך לכתוב 1	q_1	$q_0, 1, R$	$q_1, 1, R$	$q_{end}, 1, R$

דוגמת בנייה למכונה המחשבת פונקציה

$$f(x) = 0 \cdot x$$

רעיון הבנייה: נרשום 0 בהתחלה, ובכל צעד נזכור האם לכתוב 0 או 1.

פורמלית:

$$M = (Q = \{q_0, q_1, q_{end}\}, q_0, F = \{q_{end}\}, \Gamma = \{0,1,\bar{b}\}, \Sigma = \{0,1\}, \bar{b}, \delta)$$

$Q \backslash F$ Γ	0	1	$\bar{b}$
q_0	$(q_0, 0, R)$		
q_1	$(q_0, 0, R)$		

דוגמת בנייה למכונה המחשבת פונקציה

$$f(x) = 0 \cdot x$$

רעיון הבנייה: נרשום 0 בהתחלה, ובכל צעד נזכור האם לכתוב 0 או 1.

פורמלית:

$$M = (Q = \{q_0, q_1, q_{end}\}, q_0, F = \{q_{end}\}, \Gamma = \{0,1,\bar{b}\}, \Sigma = \{0,1\}, \bar{b}, \delta)$$

$Q \setminus F$ Γ	0	1	$\bar{b}$
q_0	$(q_0, 0, R)$	$(q_1, 0, R)$	
q_1	$(q_0, 0, R)$	$(q_1, 1, R)$	

דוגמת בנייה למכונה המחשבת פונקציה

$$f(x) = 0 \cdot x$$

רעיון הבנייה: נרשום 0 בהתחלה, ובכל צעד נזכור האם לכתוב 0 או 1.

פורמלית:

$$M = (Q = \{q_0, q_1, q_{end}\}, q_0, F = \{q_{end}\}, \Gamma = \{0,1,\bar{b}\}, \Sigma = \{0,1\}, \bar{b}, \delta)$$

$Q \setminus F$ Γ	0	1	$\bar{b}$
q_0	$(q_0, 0, R)$	$(q_1, 0, R)$	$(q_{end}, 0, R)$
q_1	$(q_0, 0, R)$	$(q_1, 1, R)$	$(q_{end}, 1, R)$

הפסקה

קונפיגורציות

קונפיגורציה היא מעיין תמונת מצב במהלך ריצת המכונה- לפני או אחרי השלמה של פעולת חישוב ב-δ .

קונפיגורציה היא שלשה $c = (\alpha,q,i)$ כאשר:

- $\alpha \in \Gamma^*$ תוכן הסרט. (עד הבלנק הראשון שהחל ממנו הכל בלנקים, לא כולל)
- $q \in Q$ המצב הנוכחי.
- $i \in \mathbb{N}$ מיקום הראש הקורא כותב.

ניתן לתאר ריצת מכונת טיורינג ע"י סדרת קונפיגורציות.

(דומה ברעיון למה שפגשנו באוטומטים כ "תיאור רגעי" של א"מ)

קונפיגורציות

ההגדרה של "תוכן הסרט" קצת מורכבת. בקונפיגורציה הראשונה מדובר על הקלט, אבל עם התקדמות המכונה יתכן ויהיה כתוב שם תוכן שונה לחלוטין.

נסמן T – מספר הצעדים ש M ביצעה עד כה על הקלט x. אזי, התא הימני ביותר בו ביקרנו בסרט, t, מקיים $t \leq T$. נקבע ש α יכיל את $\max\{t,|x|\}$ התאים הראשונים של הסרט.

קונפיגורציות

כאשר x היא המילה אותה קיבלנו כקלט,

- קונפיגורציה התחלתית: $(x,\ q_0,\ \ \ \ 1)$

כאשר u הוא תוכן הסרט כעת ו- i הוא מיקום הראש כעת.

- קונפיגורציה מקבלת: $(u,\ q_{accept},\ i)$
- קונפיגורציה דוחה: $(u,\ q_{reject},\ i)$
- קונפיגורציה עוצרת: $(u,\ q_{accept},\ i)$

או $(u,\ q_{reject},\ i)$

(לפעמים נקרא לזה קונפיגורציה סופית)

עוצרת: $q \in F$

(u, q, i)

צעד חישוב

הגדרה: צעד חישוב במ"ט המתחיל ב $c=(\alpha,q,i)$ יבוצע כך:

אם $\delta(q,\alpha_i)=(p,b,d)$ אזי: q יוחלף ב p (עברנו מצב); α_i יוחלף ב b, ומיקום הראש יוחלף לפי d.

שימו לב כי אם $d=L$ אזי מעדכנים את המיקום לפי $\max\{1,i-1\}$ (לא נופלים מצד שמאל של הסרט).

קונפיגורציות

ניתן לתאר ריצת מכונת טיורינג ע"י סדרת קונפיגורציות.

הגדרה: חישוב של M על קלט x הינה סדרת קונפיגורציות $c_0, c_1, \ldots$ המקיימת:
- c_0 קונפיגורציה התחלתית
- $\forall i > 0 : c_{i-1} \vdash_M c_i$
- אם קיימת קונפיגורציה סופית $c = (\alpha, q, i)$ אז היא האחרונה בסדרה, והפלט של M על x מוגדר להיות $\alpha[1 \ldots i-1]$.
- אם הסדרה לא מכילה קונפיגורציה סופית, אזי הסדרה היא אינסופית, והפלט של M על x לא מוגדר. במקרה זה נסמן כי הפלט $f_M(x) = \perp$.

דוגמת הרצה + קונפיגורציות

נניח שאנחנו במכונה המחשבת את הפונקציה $f(x) = 0 \cdot x$,

כאשר $x = 1001$

$Q \backslash \Gamma$	0	1	$\sqcup$
q_0	$(q_0, 0, R)$	$(q_1, 0, R)$	$(q_{end}, 0, R)$
q_1	$(q_0, 1, R)$	$(q_1, 1, R)$	$(q_{end}, 1, R)$

$C_0 = (1001, q_0, 1)$

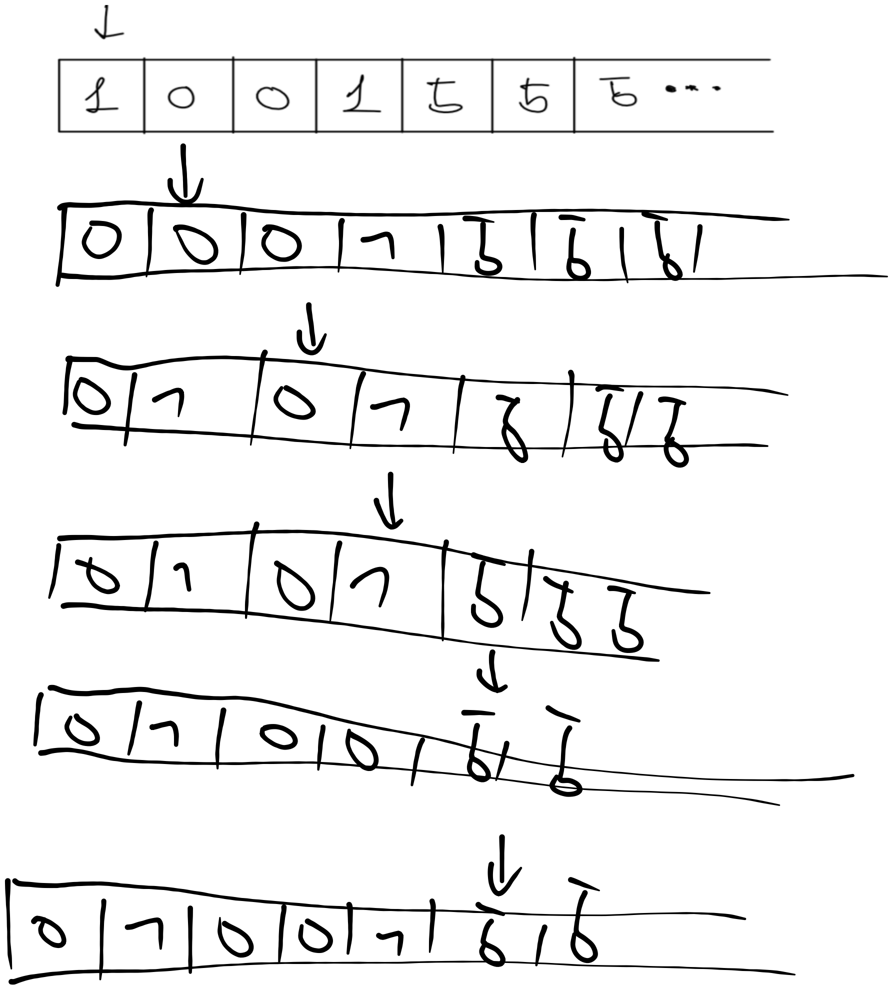
$C_1 = (0001, q_1, 2)$

$C_2 = (0101, q_0, 3)$

$C_3 = (0101, q_0, 4)$

$C_4 = (0100, q_1, 5)$

$C_5 = (01001, q_{end}, 6)$



דוגמת הרצה + קונפיגורציות

נניח שאנחנו במכונה המחשבת את הפונקציה $f(x) = 0 \cdot x$,

כאשר $x = 1001$

$Q \setminus \Gamma$	0	1	$\sqcup$
q_0	$(q_0, 0, R)$	$(q_1, 0, R)$	$(q_{end}, 0, R)$
q_1	$(q_0, 0, R)$	$(q_1, 1, R)$	$(q_{end}, 1, R)$

$C_0 = (1001, q_0, 1)$

1	0	0	1	$\sqcup$	$\sqcup$	$\sqcup \cdots$

$C_1 = (0001, q_1, 2)$

0	0	0	1	$\sqcup$	$\sqcup$	$\sqcup \cdots$

$C_2 = (0101, q_0, 3)$

0	1	0	1	$\sqcup$	$\sqcup$	$\sqcup \cdots$

$C_3 = (0101, q_0, 4)$

0	1	0	1	$\sqcup$	$\sqcup$	$\sqcup \cdots$

$C_4 = (0100, q_1, 5)$

0	1	0	0	$\sqcup$	$\sqcup$	$\sqcup \cdots$

$C_5 = (01001, q_{end}, 6)$

0	1	0	0	1	$\sqcup$	$\sqcup \cdots$

הוכחת נכונות הבנייה

כיצד נוכיח שהמכונה שבנינו עובדת?

ההוכחה הפורמלית מתייחסת לקלט x באורך n. הטענה החזקה שנוכיח היא שהחישוב של M על x הוא רצף קונפיגורציות מהצורה הבאה:

$$c_0, c_1, \dots, c_{n+1}$$

כאשר $c_0 = (q_0, x, 1)$,

$$\forall i \in [n]: c_i = \left(q_{x_i}, 0x_i \dots x_{i-1}x_{i+1} \dots x_n, i+1\right)$$

ולבסוף $c_{n+1} = (q_{end}, 0x, n+2)$.

ההוכחה לא קשה, אך לא נעשה זאת. בד"כ אפילו לא ננסח את הטענה הזו.

הפסקה?

שקילות מודלים

לכל **מודל** A של מכונת חישוב, נגדיר:

F_A = קבוצת כל הפונקציות הניתנות לחישוב ע"י מכונה כלשהי מהמודל A.

אז: בהינתן שני מודלים שונים A,B נאמר שהם שקולים אם"ם $F_A = F_B$.

מסקנה: כדי להוכיח ששני מודלים שקולים, צריך להוכיח שהקבוצות של הפונקציות המתאימות לשני המודלים שוות.

(ע"י הכלה דו כיוונית)

לדוגמה: $DFA \equiv NFA \equiv \epsilon - NFA$, אבל DFA אינו שקול ל PDA, ולא ל TM.

הוכחה (לחלק השני): תהי

$$Primality = \{p \in \{0,1\}^* | p\ is\ binary\ representation\ of\ a\ prime\ number\}$$

מ"ט מכריע את השפה (לבדוק חלוקה עד השורש) אבל השפה אינה ח"ה.

שקילות מודלים – מכונות שקולות

הגדרה: שתי מכונות טיורינג M_1,M_2 יקראו שקולות אם לכל קלט $x \in \Sigma^*$ התוצאה של שתי המכונות זהה.

- אם $M_1(x)$ מוגדר, אז גם $M_2(x)$ מוגדר, ושווה ל $M_1(x)$.
- אם $M_1(x)$ לא מוגדר, אז גם $M_2(x)$ לא מוגדר.

כלומר, $f_{M_1}(x) = f_{M_2}(x)$

- דוגמה: מ"ט זריזה, שבה מותר גם לזוז שני צעדים לכל כיוון ולא רק אחד. (רעיון ההוכחה: כיוון אחד בדיחה, בכיוון השני נוסיף מצבי q_L ו-q_R שבהם במסע ϵ מבצעים את התזוזה הנוספת).

תזת צ'רץ'-טיורינג

<u>התזה</u>: כל מודל כללי וסביר שקול למכונת טיורינג

- כללי = הכוונה למודל חזק לפחות כמו מכונת טיורינג.
- סביר = הכוונה למודל שלא מכיל רכיבים אינסופיים.

<u>שאלה</u>: אבל למכונת טיורינג יש רכיב אינסופי – הסרט הוא אינסופי.

<u>אבחנה</u>: המשמעות מבחינתנו של הסרט האינסופי, היא רק שבמהלך הריצה של מכונת טיורינג לא יקרה מצב שחסר מקום אחסון. בפועל כל כמות של תווים שנוכל לכתוב על גבי הסרט היא תמיד <u>סופית</u>.
כלומר, אם נתייחס לאורך המילה כקבוע תמיד נשתמש באורך סרט סופי אבל לא חסום.

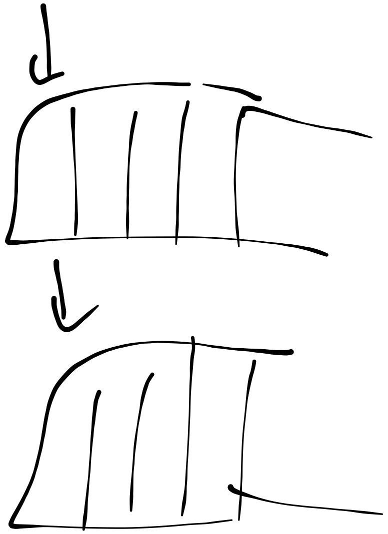
שינויים אפשריים

- מספר **קבוע** של סרטים אינסופיים.
- הסרט יכול להיות אינסופי לשני הכיוונים.
- שינויים קלים בקבוצת הצעדים האפשריים. (כך שניתן לבצע את הצעדים המקוריים, ע"י שילוב כלשהו של הצעדים החדשים).
- הוספת מספר **קבוע** של ראשים קוראים כותבים.
- באופן כללי- שינויים שלא כוללים רכיב אינסופי.

$f(x) = x \cdot x^R$

מכונה מרובת סרטים

הגדרנו מודל יחיד של מכונת טיורינג. מודל נוסף הוא מכונה בעלת k סרטים (עבור $k \geq 1$).



- לכל סרט יש ראש קורא-כותב משלו
- בתחילת הריצה הקלט רשום בסרט מספר 1, ושאר הסרטים ריקים (בלנק בכל הריבועים)
- $\delta: Q' \times \Gamma^k \to Q \times \Gamma^k \times \{L,R,S\}^k$
- הפלט מוגדר להיות תוכן הסרט הראשון בזמן העצירה.

בתרגול יוכח שמודל בעל שני סרטים שקול למודל הרגיל שהגדרנו.

מודל אינסופי

מכונת טיורינג עם קבוצת מצבים אינסופית:

נגדיר מודל M^* הזהה למכונת טיורינג הרגילה, רק שקבוצת המצבים יכולה להיות אינסופית.

טענה: המודל איננו שקול למכונת טיורינג.

הסבר: כל פונקציה הינה בעצם רשימה אינסופית של מחרוזות. אזי, בהינתן פונקציה כלשהי, ניתן לשמור את כל המחרוזות המתאימות באינסוף המצבים הנתונים לנו.

דוגמה למודל אינסופי

$$Q = \{q_\varepsilon, q_0, q_1, q_{00}, q_{01}, q_{10}, q_{11}, q_{000}, q_{001}, ..\}$$

כל מצב כזה "זוכר" את הפלט המתאים לו, כאשר נקרא את המילה נוכל לעבור בין המצבים לפי המילה שקראנו. ולבסוף, כאשר נגיע לרווח הראשון, נדע שסיימנו לקרוא את המילה ונחזיר את הפלט שהמצב שהגענו אליו זוכר. (נחזור לתחילת הסרט ונכתוב את הפלט המתאים)

- תרגיל: השלימו את הפרטים.

מכונת טיורינג זריזה

מכונה זו מוגדרת בדומה למכונת טיורינג רגילה, רק שאפשר לבצע בה גם שני צעדים ימינה ושני צעדים שמאלה.

כלומר, פונקציית המעברים נראית כך:

$$\delta : (Q \setminus F) \times \Gamma \to Q \times \Gamma \times \{LL, L, S, R, RR\}$$

<u>טענה</u>: מודל מכונת טיורינג זריזה שקול למודל מכונת טיורינג רגילה (זו שהגדרנו).

<u>רעיון ההוכחה</u>: כפי שהסברנו, על מנת להוכיח שקילות מודלים, ניקח מכונה הפועלת במודל אחד ונראה מכונה שקולה לה במודל השני, ולהיפך.

כיוון ראשון:

תהי M מ"ט רגילה. נבנה מ"ט זריזה 'M שתחשב את אותה הפונקציה ש-M מחשבת.

נשים לב ש-M מתאימה למודל של מ"ט זריזה, פשוט לא משתמשת באפשרויות RR ו-LL, על כן נוכל לבחור M'=M.

כיוון שני:

תהי M מ"ט זריזה, נבנה מ"ט 'M מהמודל הרגיל, שתחשב את אותה הפונקציה ש-M מחשבת.

$$M' = (Q', {q_0}', F', \Gamma', \Sigma', \bar{b}', \delta')$$

את רוב הרכיבים אין לנו צורך לשנות, לכן

$$q_0' \coloneqq q_0 \qquad F' \coloneqq F \qquad \Gamma' \coloneqq \Gamma \qquad \Sigma' \coloneqq \Sigma \qquad \bar{b}' \coloneqq \bar{b}$$

המשך 1 כיוון שני:

המוטיבציה שלנו היא 'לזכור' כשהמכונה M רוצה לזוז שני צעדים באותו כיוון, ותיישם זאת.

נגדיר אם כך:

$$Q' := \mathrm{Q} \cup Q_L \cup Q_R$$
$$Q_L := \{q_{iL} \mid q_i \in Q\}$$
$$Q_R := \{q_{iR} \mid q_i \in Q\}$$

עבור δ' בכל פעם ש-δ משתמשת ב- L, R, S נגדיר

$$\delta'(q,a) := \delta(q,a)$$

המשך 2 כיוון שני:

כאשר δ משתמשת ב-RR או LL,

$$\delta(q,a) = (q_i, b_i, LL) \;\Rightarrow\; \delta'(q_i,a) = (q_{iL}, b_i, L)$$

$$\delta(q,a) = (q_i, b_i, RR) \;\Rightarrow\; \delta'(q_i,a) = (q_{iR}, b_i, R)$$

וכך 'נזכור' הזזה נוספת שיש לנו להשלים.

ההשלמה תעשה כך: לכל q_i ולכל a נבצע:

$$\delta'(q_{iL}, a) = (q_i, a, L)$$

$$\delta'(q_{iR}, a) = (q_i, a, R)$$

תרגיל לבית

תהי מ"ט בעלת כיווני הצעדים הבאים:

$$\{S, Right, Reset\}$$

כאשר $Reset$ הינו לחזור לתא מספר 1.

האם מ"ט זו שקולה למכונה רגילה?

הכוונה: כדי להפריך, מספיקה שפה אחת שניתן להכריע בעזרת אחת מהן ולא בעזרת השנייה. כדי להוכיח, יש להראות איך מבצעים $Left$ במודל החדש.