מודלים חישוביים 21.4.2025

הרצאה 5: מכונות טיורינג

Based on previous iterations of this course, given by Nir Bitansky, Rotem Oshman, Iftach Haitner, and Omer Paneth.

מרצה: אורי שטמר

תזכורת: <u>מכונת מטיורינג</u> (מ"ט) היא שביעייה $(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ כאשר:
- Q קבוצת מצבים סופית $q_0, q_a, q_r \in Q$
- Σ אלפאבית קלט
- Γ אלפאבית סרט כך ש- $\Gamma \subseteq \Sigma$, $\sqcup \in \Gamma \setminus \Sigma$
- δ פונקציית מעברים כך ש- $\delta: (Q \setminus \{q_a, a_r\}) \times \Gamma \to Q \times \Gamma \times \{L, R\}$
- q_0 מצב תחילי
- q_a מצב מקבל
- q_r מצב דוחה, $q_a \neq q_r$

<u>**דוגמה:**</u> $L = \left\{a^{2^n} : n \geq 0\right\}$

ראינו תיאור לא פורמלי של מ"ט עבור השפה. כעת נראה תיאור פורמלי של מ"ט מתאימה.

$$\Gamma = \{a, \cancel{a}, \sqcup, a_0\}, \quad \Sigma = \{a\}, \quad Q = \{q_0, q_1, q_2, q_3, q_4, q_a, q_r\}$$

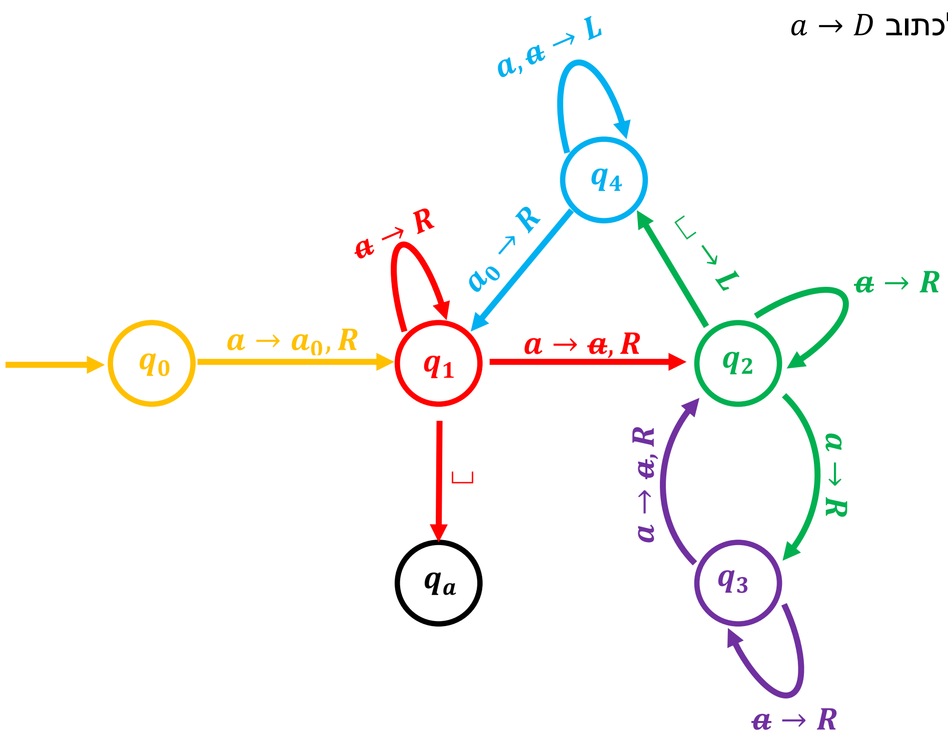
כעת נתאר את פונק' המעברים בעזרת תרשים

<u>**סימון:**</u> עבור $\delta(q, a) = (q', a', D)$ נסמן $q \xrightarrow{a \to a', D} q'$

אם $q' = q_r$ ניתן להשמיט את הקשת.

אם $a = a'$ ניתן להשמיט a' ולכתוב $a \to D$

פונק' המעברים δ:



q_0	אתחול: נסמן את ה- a הראשון ב- a_0
q_1	ניסיון סיום: נבדוק האם נותרו a למעט a_0. אם לא, נקבל. אם כן, נתחיל בתהליך מחיקת מחצית מה-a-ים הנותרים
q_2, q_3	נעים על הקלט ימינה ולסירוגין מדלגים על a או מחוקים אותו. q_2 מדלג ו- q_3 מוחק • אם q_2 מגיע לסוף הקלט (לתו $\sqcup$) אז סיימנו למחוק מחצית מה-a-ים ונעבור ל- q_4 דרכו ננוע שמאלה. • אם q_3 מגיע לסוף הקלט אז דוחים כי ז"א שנכנסנו ל- q_1 עם מס' אי-זוגי גדול מ-1 של a-ים.
q_4	נעים שמאלה עד שנגיע ל- a_0 ואז חזרה ימינה ול- q_1

q_0						
a	a	a	a	$\sqcup$	$\sqcup$	

	q_1					
a_0	a	a	a	$\sqcup$	$\sqcup$	

		q_2				
a_0	$\cancel{a}$	a	a	$\sqcup$	$\sqcup$	

			q_3			
a_0	$\cancel{a}$	a	a	$\sqcup$	$\sqcup$	

				q_2		
a_0	$\cancel{a}$	a	$\cancel{a}$	$\sqcup$	$\sqcup$	

			q_4			
a_0	$\cancel{a}$	a	$\cancel{a}$	$\sqcup$	$\sqcup$	

		q_4				
a_0	$\cancel{a}$	a	$\cancel{a}$	$\sqcup$	$\sqcup$	

	q_4					
a_0	$\cancel{a}$	a	$\cancel{a}$	$\sqcup$	$\sqcup$	

q_4						
a_0	$\cancel{a}$	a	$\cancel{a}$	$\sqcup$	$\sqcup$	

	q_1					
a_0	$\cancel{a}$	a	$\cancel{a}$	$\sqcup$	$\sqcup$	

		q_1				
a_0	$\cancel{a}$	a	$\cancel{a}$	$\sqcup$	$\sqcup$	

			q_2			
a_0	$\cancel{a}$	$\cancel{a}$	$\cancel{a}$	$\sqcup$	$\sqcup$	

				q_2		
a_0	$\cancel{a}$	$\cancel{a}$	$\cancel{a}$	$\sqcup$	$\sqcup$	

			q_4			
a_0	$\cancel{a}$	$\cancel{a}$	$\cancel{a}$	$\sqcup$	$\sqcup$	

		q_4				
a_0	$\cancel{a}$	$\cancel{a}$	$\cancel{a}$	$\sqcup$	$\sqcup$	

	q_4					
a_0	$\cancel{a}$	$\cancel{a}$	$\cancel{a}$	$\sqcup$	$\sqcup$	

q_4						
a_0	$\cancel{a}$	$\cancel{a}$	$\cancel{a}$	$\sqcup$	$\sqcup$	

	q_1					
a_0	$\cancel{a}$	$\cancel{a}$	$\cancel{a}$	$\sqcup$	$\sqcup$	

		q_1				
a_0	$\cancel{a}$	$\cancel{a}$	$\cancel{a}$	$\sqcup$	$\sqcup$	

			q_1			
a_0	$\cancel{a}$	$\cancel{a}$	$\cancel{a}$	$\sqcup$	$\sqcup$	

				q_1		
a_0	$\cancel{a}$	$\cancel{a}$	$\cancel{a}$	$\sqcup$	$\sqcup$	

q_a

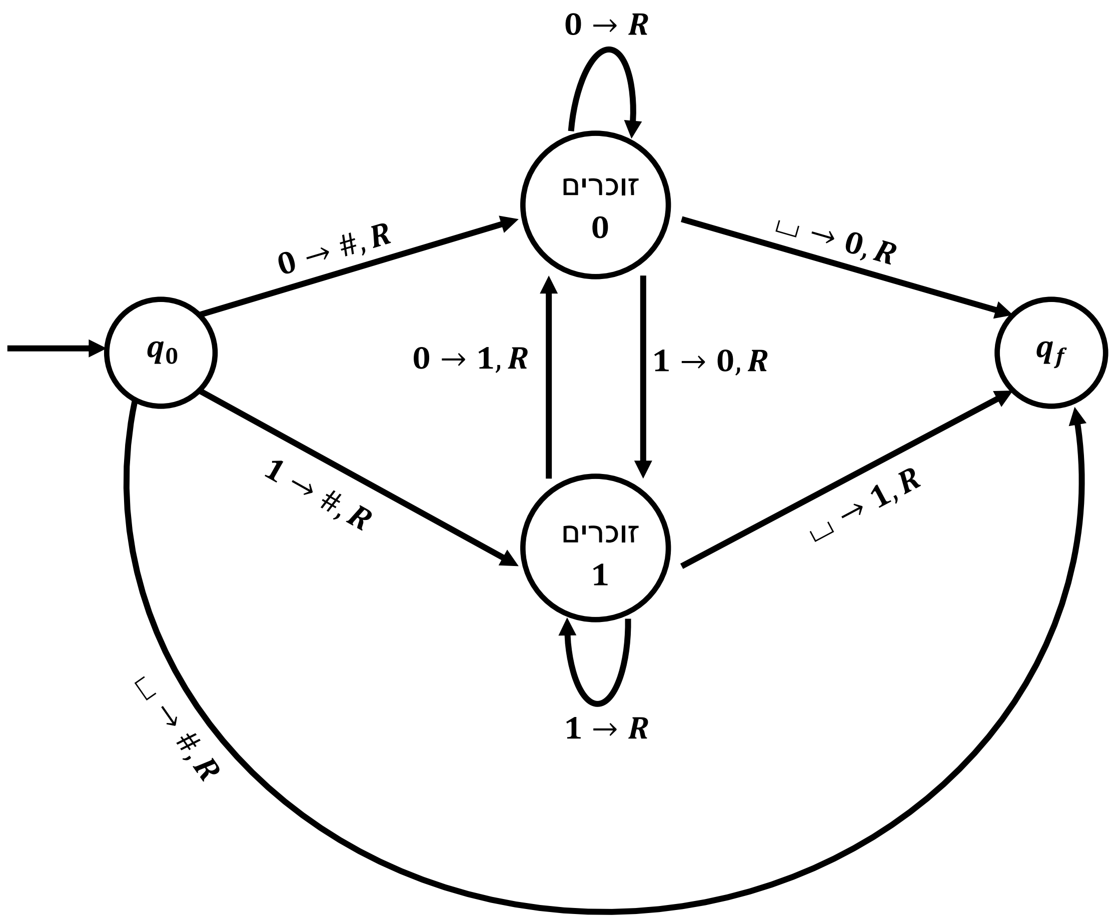
בואו נראה דוגמה נוספת למ"ט (בקרוב כשנחשוב על אלג' נשכח מהתיאור הקונקרטי של פונק' המעברים ונחשוב במושגים אליהם אנו רגילים)

<u>דוגמה נוספת:</u> מ"ט המקבלת קלט x ומסיטה אותו "שיפט אחד ימינה" תוך שבתא השמאלי ביותר מוסיפה תו מיוחד #

הערה: בעתיד נשתמש במ"ט כזו או דומה לה בתוך תת-פרוצדורה במ"ט גדולות ולכן אין משמעות לקבלה/דחיה של מילה

הרעיון: נעבור על הקלט משמאל לימין עד שנגיע ל ⊔ הראשון. בכל צעד נחליף את התו הנוכחי בתו שזכרנו ומכאן נזכור את זה שהוחלף כעת.

$\boldsymbol{\Sigma = \{0,1\}}$



<u>דוגמה נוספת:</u> מונה לקסיקוגרפי. נניח $\Sigma = \{0,1\}$. סידור לקסיקוגרפי על Σ^* הוא:

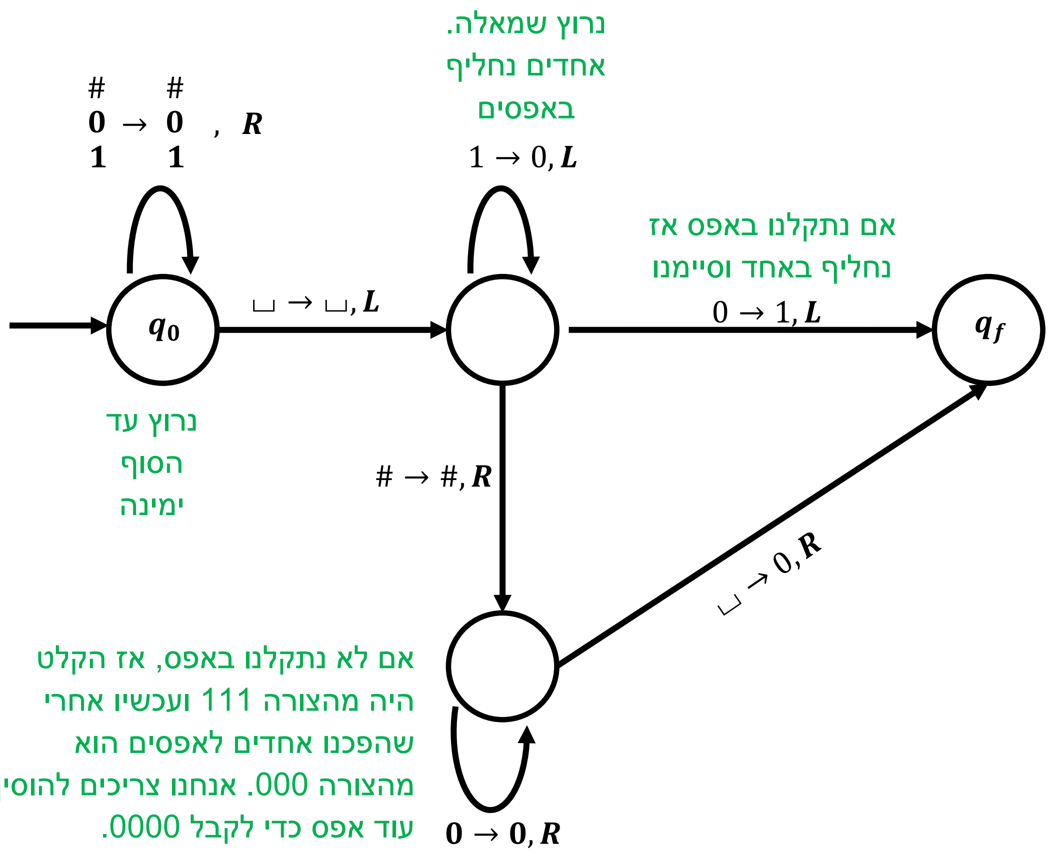
$\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots$

(כלומר – סידור קודם לפי מס' האותיות ואח"כ לפי האותיות עצמם)

<u>קלט:</u> $\#x$ עבור $x \in \Sigma^*$

<u>פלט:</u> $\#x'$ כאשר x' הוא העוקב של x בסידור הלקסיקוגרפי

<u>הרעיון:</u> נרוץ מימין לשמאל. כל עוד רואים 1 הופכים ל- 0. כשמגיעים ל- 0 הופכים אותו ל- 1 ועוצרים. אם הגענו ל # אזי לא היו אפסים בדרך ולכן צריך להוסיף ספרה.



<u>הגדרת ריצה של מ"ט באופן פורמלי: קונפיגורציות</u>

קונפיגורציה זה תיאור של "תמונת מצב גלובלית" ברגע מסוים במהלך הריצה. זה כולל מצב פנימי, תוכן סרט העבודה, ומיקום הראש.

בהגדרות הבאות $M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ מ"ט ונניח בה"כ $Q \cap \Gamma = \emptyset$.

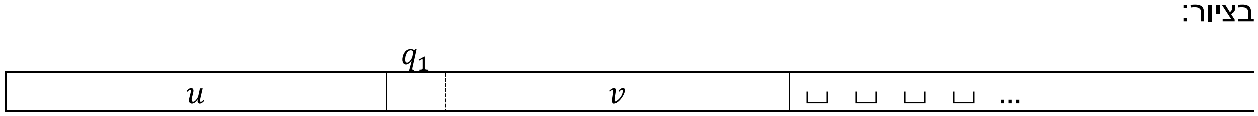
הגדרה: <u>קונפיגורציה</u> של M היא מחרוזת $c \in \Gamma^* Q \Gamma^*$.

<u>פרשנות:</u> עבור $q \in Q$, $u, v \in \Gamma^*$ נפרש את uqv באופן הבא:

- תוכן הסרט הנוכחי הוא $uv\sqcup^\infty$
- המצב הפנימי הוא q
- הראש נמצא מעל התו הראשון של v

<u>הערה:</u> נתייחס ל- c ול- $c\sqcup$ כאותה קונפיגורציה.

בציור:



הגדרה: תהי c קונפיגורציה.

- c <u>התחלתית</u> אם"ם $c = q_0 v$ עבור $v \in \Sigma^*$
- c <u>מקבלת</u> אם"ם $c = uq_a v$ עבור $u, v \in \Gamma^*$
- c <u>דוחה</u> אם"ם $c = uq_r v$ עבור $u, v \in \Gamma^*$

בריצה של מ"ט, המכונה עוברת מקונפ' c לקונפ' הבאה ע"י הפעלה של פונק' המעברים ועדכון c בהתאם.

הגדרה: קונפיגורציה c <u>עוברת-δ</u> ל- c' אם"ם מתקיים אחד מהתנאים הבאים עבור איזשהם $a, b \in \Gamma,\ \ u, v \in \Gamma^*,\ \ q, q' \in Q$:

$$c = uaqbv \quad , \quad \delta(q,b) = (q', b', L) \quad , \quad c' = uq'ab'v \qquad (1)$$
$$c = qbv \quad , \quad \delta(q,b) = (q', b', L) \quad , \quad c' = q'b'v \qquad (2)$$
$$c = uqbv \quad , \quad \delta(q,b) = (q', b', R) \quad , \quad c' = ub'q'v \qquad (3)$$

* אם $c = uq$ אז נפרש את ההגדרה לפי הקונפ' השקולה $uq\sqcup$

* אם δ ברורה מההקשר אז פשוט נאמר "עוברת" במקום "עוברת-δ"

<u>שימו לב:</u> הקונפ' משתנה רק באיזור הראש (u, v לא משתנים)

הגדרה: M <u>מקבלת/דוחה</u> קלט $x \in \Sigma^*$ אם קיימת סדרת קונפ' $c_0, \dots, c_t$ כך ש-

(1) $c_0 = q_0 x$
(2) c_{i-1} עוברת ל- c_i לכל $i \in [t]$
(3) c_t קונפ' מקבלת/דוחה

<u>סימון:</u> $L(M) =$ אוסף המילים ש- M מקבלת

<u>שימו לב:</u> יתכנו קלטים עליהם M אינה עוצרת (לא מקבלת ולא דוחה)!

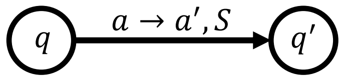
מודלים שקולים

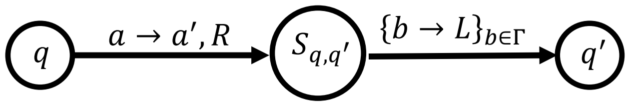
כפי שכבר אמרנו, מטרה עיקרית בקורס שלנו היא להבין את נושא החישוב. בשביל זה אנחנו צריכים להגדיר מודל חישוב ואז להבין את היכולות שלו. עברנו דרך המודל של מעגלים, אוטומטים, מ"ט. אבל מה שבאמת הייתה המוטיבציה שלנו בתחילת הקורס זה להבין את היכולות של חישוב נגיד בפייתון או ב C++. מ"ט זה מודל שיהיה לנו מאוד נח לנתח אבל הוא לא בדיוק פייתון. מה שנעשה עכשיו זה להשתכנע שמ"ט בעצם כן תופס את כל היכולות של פייתון. איך? אנחנו בכמה שלבים נעשה שינויים למודל של מ"ט ונטען שהמודל עם השינויים שקול למודל הקודם, עד שנקבל מודל שדומה לאסמבלי.

<u>חימום:</u> נניח שהיינו משנים את המודל כך שהראש יכול להישאר במקום.

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

האם מוסיף כח? לא... ניתן לחקות עמידה במקום ע"י שני מעברים: R ואז L.
בציור:



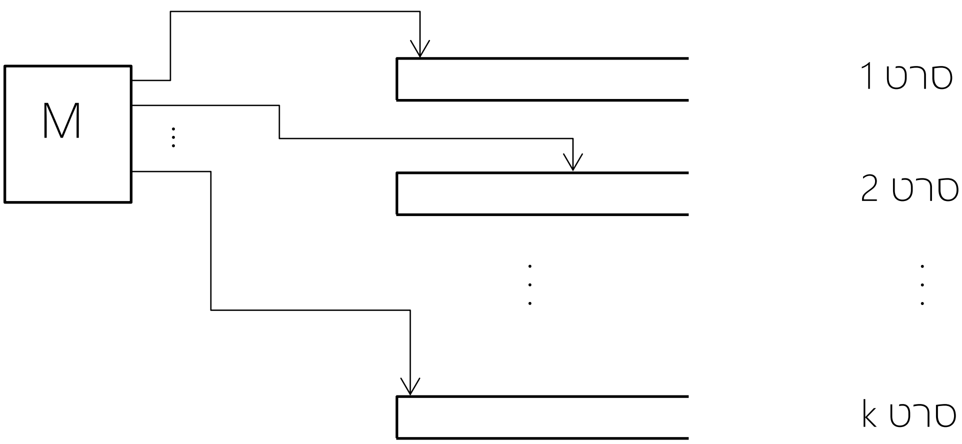


הגדרה (חצי פורמלית): נאמר כי שני <u>מודלים שקולים</u> אם כל אחד מהם יכול לסמלץ את השני. כלומר לכל מכונה M במודל אחד קיימת מכונה M' במודל השני שמקבלת/דוחה/לא-עוצרת בדיוק על אותם קלטים.

נדון כעת בשני מודלים השקולים למ"ט (באופן חצי פורמלי)

<u>מכונה רב-סרטית</u>

הכללה של המודל של מ"ט כפי שראינו אותו. עכשיו יש לנו גישה ל- k סרטים בעזרת k ראשים קוראים/כותבים:



- **בתחילת החישוב** הקלט נמצא על הסרט הראשון והשאר ריקים ($\sqcup^{\infty}$), ובכל סרט הראש מופיע בקצה השמאלי.
- **בכל צעד חישוב** המכונה קוראת k-איה של תווים (תו מתחת לכל ראש) ועפ"י ה-k-איה הנ"ל והמצב הפנימי מחליטה:
  (1) איזה תו לרשום תחת כל ראש
  (2) לאן להזיז כל ראש
  (3) לאיזה מצב פנימי לעבור
- **כלומר פונקציית המעברים** נראית כך: $\delta: Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R\}^k$

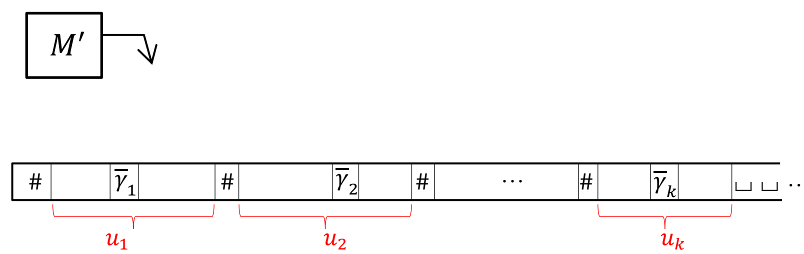
- **קונפ' k-סרטית** היא מחרוזת מהצורה $c = c_1\$c_2\$ \dots \$c_k$ כאשר כל c_i היא קונפ' חד-סרטית.
בפרט, קונפ' התחלתית נראית כך $q_0 x \sqcup \$ q_0 \sqcup \$ \dots \$ q_0 \sqcup$
(שימו לב, אנחנו המוסכמה היא שאנחנו רושמים את אותו מצב k פעמים. זה הדרך שלנו לסמן איפה נמצא הראש הקורא/כותב בכל סרט)

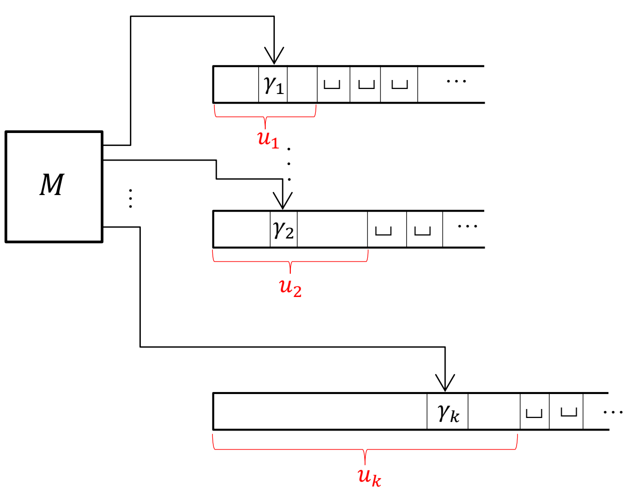
טענה: לכל קבוע $k \in \mathbb{N}$, מ"ט k-סרטית שקולה למ"ט חד-סרטית.

__**רעיון הוכחת הכיוון המעניין:**__ בהינתן מ"ט k-סרטית M נבנה מ"ט חד סרטית M' שמסמלצת אותה.
למה אנחנו מתכוונים "מסמלצת"? מי שבפועל תרוץ זאת תהייה M' (המכונה החד סרטית). אנחנו נדאג שבכל שלב בריצה של M', אם נרים את מכסה המנוע שלה, נראה תמונת מצב שמתאימה ל M. ואז, על כל צעד ש M הייתה רוצה לעשות, אנחנו נעשה __סדרת__ פעולות ונגיע לתמונת המצב הבאה שהייתה מתאימה ל M אחרי הצעד האחד הזה. לצורך כך:

- נשמור את תכני k הסרטים של M באופן משורשר (על הסרט היחיד של M'), מופרדים ב #
- מיקומי הראשים ייוצגו ע"י סימון bar, למשל $\bar{a}$

בציור:





__**סימולטור חד-סרטי M' למ"ט k-סרטית M:**__ **שלב האתחול:** בהינתן קלט $x = x_1 \dots x_n$, נרשום על הסרט $\#\overline{x_1}x_2 \dots x_n\#\bar{\sqcup}\#\bar{\sqcup}\# \dots \#\bar{\sqcup}\sqcup\sqcup \dots$ **כדי לסמלץ כל צעד של M:** 1. נסרוק את הסרט על מנת לקרוא את k התווים מתחת לראשים (שמופיעים עם bar). נזכור מידע זה בעזרת המצב של M' (ניתן לזכור מידע זה כי הוא בגודל קבוע – יש $\|\Gamma\|^k$ אפשרויות) 2. נסרוק פעם שניה את הסרט על מנת לעדכן את התווים תחת הראשים הווירטואליים ואת מיקום הראשים הווירטואליים (מיקום סמני bar) בהתאם לפונק' המעברים של M. __הערה:__ אם ראש וירטואלי כלשהו אמור לזוז ימינה למקום שבו יש #, כלומר למקום "לא מאותחל" בסרט הווירטואלי (בסרט המקורי מגיעים ל- $\sqcup$ מסדרת ה- $\sqcup$ האינסופית), אזי M' תסיט את כל סיפת הסרט תא אחד ימינה תוך שבמקום הנוכחי נכתוב $\sqcup$.

<u>מהי סיבוכיות הסימולציה?</u> כלומר, אם M מבצעת T_x פעולות חישוב על קלט x, כמה פעולות חישוב תבצע M' על x?

לשם פשטות נניח כי $T_x \geq |x|$ ולכן נוכל להתעלם משלב האתחול שלוקח $|x|+k$ צעדים.

כל צעד בסימולציה (כלומר צעד ששקול לפעולה יחידה של M) לוקח לכל היותר k מעברים על הסרט כולו. ומה אורך הסרט (החלק הפעיל)? הוא לכל היותר k פעמים אורך הסרט הארוך ביותר בשלב המקביל בחישוב M על x שהוא חסום ע"י T_x. לכן, מס' הפעולות ש- M' מבצעת בצעד סימולציה יחיד הוא לכל היותר

$$\underbrace{k}_{\substack{\text{מספר הסריקות}\\\text{של הסרט}}} \cdot \underbrace{k \cdot T_x}_{\substack{\text{אורך מקסימלי}\\\text{של הסרט}\\\text{בכל שלב}}}$$

בסה"כ זמן הריצה של M' על x הוא

$$O\left(\underbrace{T_x}_{\substack{\text{צעדים}\\\text{של } M}} \cdot \underbrace{k^2 T_x}_{\substack{\text{מחיר כל}\\\text{צעד}}}\right) \underbrace{=}_{k \text{ קבוע}} O(T_x^2)$$

<u>מכונת RAM (Random Access Machine/Memory), דיון לא פורמלי</u>

זה מודל שתופס בצורה יותר טובה את מה שבאמת קורה במחשבים שלנו. במקום לעבוד עם אלפבית סופי כמו מ"ט, במודל ה RAM נעבוד עם ערכים שלמים אי-שליליים (לא חסומים). במודל הזה יש לנו:

- "רגיסטרים" (בהם נחזיק ערכים מספריים עליהם נרצה לבצע פעולות בסיסיות)

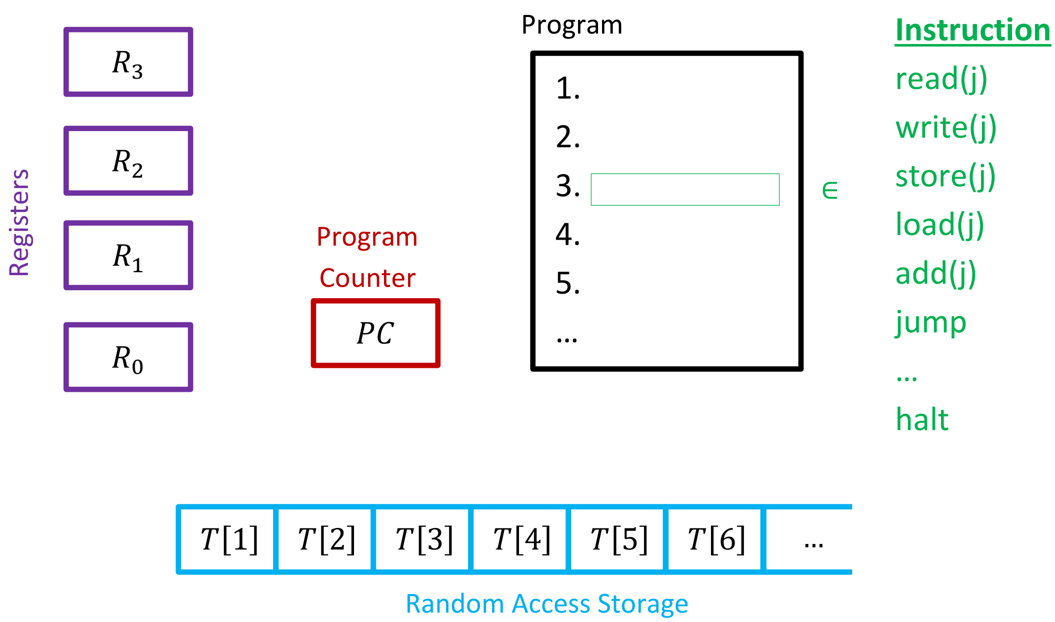
- "מערך" זיכרון אינסופי, אנלוגי לסרט הזיכרון האינסופי במ"ט. ישנם 2 הבדלים עיקריים בין הזיכרון הזה לבין הסרט של מ"ט:
  1. כל תא במערך הזה מחזיק <u>מספר</u>
  2. יש לנו גישה ישירה לכל תא במערך הזה, ללא צורך לסרוק את הזיכרון בצורה סדרתית כמו במ"ט

- בנוסף לזיכרון ולרגיסטרים, מכונה במודל ה RAM גם מכילה "תוכנית לביצוע" (=סדרה של פקודות). כדי שנדע "איפה אנחנו" בקריאת התוכנית הזאת, אחד הרגיסטרים במכונה הוא רגיסטר יעודי לשם כך, נקרא "מונה תוכנית" (program counter, PC). המונה הזה זוכר את האינדקס של הפקודה הבאה שעלינו לבצע.

- כל פקודה בתוכנית הזאת היא אחת מתוך אוסף סופי של פקודות אפשריות אשר מאוד דומות לפקודות באסמבלי. למשל:

Instruction	Operand	הסבר
Read	j	$R_0 \leftarrow T[j]$ (הערה: לרגיסטר R_0 יש סטטוס מיוחד והוא משתתף בכמעט כל פקודה)
write	j	$T[j] \leftarrow R_0$
store	j	$R_j \leftarrow R_0$
load	j	$R_0 \leftarrow R_j$
add	j	$R_0 \leftarrow R_0 + R_j$
jump		$PC \leftarrow R_0$ (יש כמה סוגי jump אפשריים)
...	...	...
halt		פקודת סיום ריצה. נחליט על קבלה/דחיה לפי הערך של R_0

בציור:



<u>בצורה פורמלית:</u>

- מכונה במודל RAM מוגדרת ע"י זוג (k, Π) כאשר:
  - $k \in \mathbb{N}$ מציין את מספר הרגיסטרים
  - $\Pi = (\pi_1, \pi_2, \dots, \pi_p)$ היא סדרה של פקודות

- קונפיגורציה במודל זה מוגדרת ע"י:
  1. הערך של מונה התוכנית PC
  2. ערכים ברגיסטרים $R_0, R_1, \dots, R_k \in \mathbb{N} \cup \{0\}$
  3. ערכים במערך הזיכרון $T : \mathbb{N} \rightarrow (\mathbb{N} \cup \{0\})$

  בתחילת הריצה כל המערך מאותחל לאפסים. שימו לב שבכל רגע נתון במהלך הריצה, ישנם מספר סופי של תאים המכילים ערכים ששונים מאפס.

טענה: מכונות טיורינג שקולות ל- RAM.

אנחנו לא נוכיח את זה, אלא רק נסביר זאת בצורה אינטואיטיבית.

הכיוון הקל הוא לממש מ"ט ע"י RAM:

- את הסרט נוכל לסמלץ ע"י המערך הזיכרון
- בכל רגע נזכור את מיקום הראש באחד הרגיסטרים, נניח ב R_1
- את פונקציית המעברים נממש בעזרת "תוכנית פקודות" מתאימה

הכיוון השני: בהינתן מכונת RAM נבנה מ"ט רב-סרטית מתאימה

- יהיו לנו $k+3$ סרטים:
  1. יהיה לנו סרט אחד לכל רגיסטר אשר יכיל את הערך באותו רגיסטר
  2. יהיה לנו סרט נוסף שישמש כזיכרון זמני, למשל כדי לזכור את המספר 271 בפקודה $R_0 \leftarrow R_0 + 271$
  3. יהיו לנו סרט נוסף שעליו יהיה כתוב הקלט בתחילת הריצה (נניח שעל אותו סרט גם נכתוב את הפלט, אם צריך, בסיום הריצה)
  4. וסרט נוסף אשר יסמלץ את מערך הזיכרון של מכונת ה RAM במהלך הריצה. אנחנו צריכים לזכור גם את תוכן התאים הלא ריקים ממערך הזיכרון וגם את הכתובות שלהם, ולכן יהיה נח לתחזק את זה כרשימת זוגות מהצורה $(m, T[m])$.

- את "התוכנית" של מכונת ה RAM נסמלץ בעזרת המצבים ופ' המעברים של המ"ט שלנו. כדי לעשות את זה אנחנו צריכים תת-שגרה (או תת-מ"ט) עבור כל פקודה בתוכנית. למשל, כדי לסמלץ פקודה אשר מערבת קריאה ממערך הזיכרון בתא m, נסרוק את הסרט המתאים ונמצא את הזוג עם האינדקס m, ואז נקרא או נכתוב את הערך המתאים בזוג הזה. (אם לא מצאנו זוג עם אינדקס מתאים, אז הערך שחיפשנו לא אותחל עדיין ולכן ערכו אפס). אחרי שסיימנו לטפל בפקודה הנוכחית, נעבור למצב שמתאים לתחילת הפקודה הבאה (זה אנלוגי להגדלת ה PC).

המטרה שלנו עם השקילות הזאת בין מ"ט ל *RAM* היא לתת גמישות מחשבתית: כשאנחנו רוצים להראות שאלגוריתמים "רגילים" יכולים לעשות משהו, יהיה לנו נח לחשוב על פייתון או מודל *RAM* אחר. כשנרצה להראות שאלגוריתמים "לא יכולים" לעשות משהו, יהיה לנו נח לחשוב על מ"ט, כי זה מודל פשוט ולכן יותר קל להראות מגבלות.

התזה של צ'רץ' וטיורינג

ישנם מודלי חישוב רבים, כולל שפות עיליות כמו *Python, Java, C*. כל המודלים הללו הוכחו כשקולים למ"ט! (נקראים גם טיורינג שלמים).

התזה של אלונזו צ'רץ' ואלן טיורינג (לא פורמלי): כל מודל חישוב "סביר" ניתן לסמלץ ע"י מ"ט.

אלונזו צ'רץ' ואלן טיורינג פרסמו את התזה הזאת במקביל בסביבות שנת 1936, צ'רץ' בארה"ב וטיורינג באנגליה. התזה של צ'רץ' הייתה מבוססת על תחשיב-λ, במקום מ"ט, אבל זה שקול. (צ'רץ' היה המנחה של טיורינג לדוקטורט)

מ"ט לא-דטרמיניסטיות (מטל"ד)

בדומה לאסל"ד, במטל"ד בכל צעד יתכנו כמה מעברים אפשריים.

הגדרה: מ"ט לא-דטרמיניסטית (מטל"ד) היא שביעייה $(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ כאשר:
- Q קבוצת מצבים סופית $q_0, q_a, q_r \in Q$
- Σ אלפאבית קלט
- Γ אלפאבית סרט כך ש- $\Sigma \subseteq \Gamma$, $\sqcup \in \Gamma \setminus \Sigma$
- δ פונקציית מעברים כך ש- $\delta: (Q \setminus \{q_a, a_r\}) \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L,R\}}$
- q_0 מצב תחילי
- q_a מצב מקבל
- q_r מצב דוחה, $q_a \neq q_r$

כדי לדבר על ריצה של מטל"ד נשתמש שוב במושג הקונפיגורציה. קונפ' של מטל"ד מוגדרת באופן דומה למקרה הדטר'. בשונה, כל קונפ' c עשויה לעבור למס' קונפ' c' ע"י עדכון c בהתאם לכל אחד מהמעברים האפשריים.

דוגמה: אם $\delta(q,b) = \{(q', b', L), (q'', b'', R)\}$ אז מהקונפיגורציה $c = uaqbv$ אפשר לעבור לקונפיגורציה $c' = uq'ab'v$ או לחילופין לעבור לקונפיגורציה $c'' = uab''q''v$.

הגדרה: עבור מטל"ד $(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ נאמר כי קונפיגורציה c היא <u>δ-עוברת</u> לקונפיגורציה c' אם"ם מתקיים (לפחות) אחד מהתנאים הבאים עבור איזשהם $a, b, b' \in \Gamma, \quad u, v \in \Gamma^*, \quad q, q' \in Q$:

$$c = uaqbv \quad , \quad (q', b', L) \in \delta(q, b) \quad , \quad c' = uq'ab'v \qquad (1)$$
$$c = qbv \quad , \quad (q', b', L) \in \delta(q, b) \quad , \quad c' = q'b'v \qquad (2)$$
$$c = uqbv \quad , \quad (q', b', R) \in \delta(q, b) \quad , \quad c' = ub'q'v \qquad (3)$$

* אם $c = uq$ אז נפרש את ההגדרה לפי הקונפ' השקולה $uq\sqcup$
* אם δ ברורה מההקשר אז פשוט נאמר "עוברת" במקום "δ-עוברת"

הגדרה: תהי $N = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ מטל"ד ויהי $x \in \Sigma^*$.
<u>עץ החישוב</u> של N על x, שנסמנו $T_{N,x}$, הוא עץ מושרש שבו כל צומת הוא קונפיגורציה כאשר:
- השורש הוא הקונפ' ההתחלתית $q_0 x \sqcup$
- לכל צומת c, ילדיו הם הקונפ' אליהן הקונפ' c עוברת.
- צומת הוא עלה אם"ם הוא קונפ' מקבלת/דוחה או שאינו עובר לאף קונפ', כלומר $\delta(q,b) = \emptyset$

הגדרה:

- N <u>מקבלת</u> את x אם"ם ב- $T_{N,x}$ יש עלה מקבל
- N <u>דוחה</u> אם"ם $T_{N,x}$ סופי וללא עלים מקבלים
- אחרת ($T_{N,x}$ אינסופי ואין עלים מקבלים), N <u>אינה עוצרת</u>.

זה מסיים את הגדרת המודל של מטל"ד

<u>**טענה:**</u> המודלים של מ"ט ומטל"ד שקולים.

מ"ט היא בבירור מקרה פרטי של מטל"ד. בכיוון השני נרצה להראות שכל מטל"ד N אפשר לסמלץ ע"י מ"ט M.

<u>הרעיון הבסיסי:</u> עבור קלט x, המ"ט M תסרוק את עץ החישוב $T_{N,x}$ ותחפש עלה מקבל. אם העץ סופי ללא עלים מקבלים אז נדחה (אחרת סורקים לנצח).

<u>בעיה:</u> ב- $T_{N,x}$ עשויים להיות מסלולים אינסופיים (גם עבור מילה $x \in L(N)$).

<u>פתרון:</u> נבצע חיפוש BFS ולא DFS. כלומר M תבצע סימולציה של כל החישובים של N על x באורך 1, אח"כ באורך 2, אח"כ באורך 3,...

<u>שאלה:</u> האם לא יתכן שברמה ה-k-ית בעץ ישנם אינסוף קודקודים?

<u>תשובה:</u> לא יתכן. לכל קונפ' (כלומר קודקוד בעץ), מספר הקונפיגורציות העוקבות האפשרי חסום ע"י

$$C_N = \underbrace{|Q|}_{\text{מס האפשרויות למצב הבא}} \cdot \underbrace{|\Gamma|}_{\text{מס האפשרויות לתו שיכתב}} \cdot \underbrace{2}_{\text{מס האפשרויות לתנועת הראש}}$$

לכן ברמה ה-k-ית בעץ יש לכל היותר $(C_N)^k$ קודקודים, שזה מספר סופי

<u>**בניית M בהינתן N:**</u>

<u>הנחה מפשטת:</u> נניח שלכל $\gamma \in \Gamma$ ולכל $q \in Q \setminus \{q_a, q_r\}$ מתקיים $|\delta(q,\gamma)| = C_N$

(בה"כ כי נוכל להוסיף מעברים למצב q_r)

כעת, בהינתן γ, q נניח סדר (לקסיקוגרפי) על המעברים האפשריים, כלומר על השלשות:

$$\delta(q,\gamma) = \{(q^1, \gamma^1, D^1), \dots, (q^{C_N}, \gamma^{C_N}, D^{C_N})\}$$

כך בהינתן העץ $T_{N,x}$ הגדרנו סידור על הבנים של כל קודקוד uqv בעץ.

<u>הערה:</u> לפי ההנחה המפשטת שלנו, לכל קודקוד יש או C_N בנים או 0 בנים (אם הוא עלה).

עתה, כל מחרוזת $\alpha \in \{1,2,\dots,C_N\}^k$ מגדירה מסלול בעץ $T_{N,x}$ באורך k (=חישוב של M על x ב- k צעדים).

נבנה את M בשני שלבים:

(1) נבנה מ"ט L שתשמש את M כתת-פרוצדורה אשר מקבלת x ומחרוזת בחירות α ומסמלצת את ריצת N על x עפ"י α

(2) נבנה M שתחזיק מונה בבסיס C_N שישמש כקלט (יחד עם x) ל- L.

<u>בניית L:</u>

L היא מ"ט דטר' שמצביה הם בדיוק המצבים של N.

L תשתמש בשלושה סרטים: על סרט הקלט תהיה רשומה המילה x. על הסרט השני תהיה רשומה מחרוזת בחירות α (באורך סופי). בסרט השלישי נשתמש בשביל לבדוק האם עץ החישוב הוא סופי וכבר סרקנו את כולו.

<u>ריצת L:</u>

- בכל שלב בחישוב, L קוראת בסרט 1 תו נוכחי γ וקוראת בסרט 2 תו "בחירות" c. בהינתן שהמצב הנוכחי הוא q, מבצעת את המעבר ה- c בסידור של $\delta(q,\gamma)$.
  כלומר, אם המעבר ה- c הוא (q',γ',D) אז:
  - בסרט הראשון, החלף את γ ב- γ' והזז את הראש לכיוון D
  - בסרט השני, הזז את הראש תא אחד ימינה
  - נעבור למצב q'.
- אם לא ניתן לבצע את המעבר, כלומר נתקלנו בעלה (כלומר $\delta(q,\gamma)=\emptyset$ או ש- $q\in\{q_a,q_r\}$) אז **נעצור ונדחה/נקבל** בהתאם
- אם בסרט 2 מגיעים לתו ריק אז נכתוב על הסרט השלישי "<u>לא סופי</u>" **ונעצור ונדחה** (אחרת נמשיך בסימולציה). כלומר, סיימנו לסמלץ את הענף הרצוי ולא הגענו לעלה אז נסמן למכונה שקראה לנו שהענף שבדקנו עכשיו לא נגמר וצריך להמשיך אותו.

<u>הערה:</u> L מבצעת סימולציה של חישוב <u>אפשרי</u> באורך $k=|\alpha|$ של N על x ולכן אם L דוחה את x,α זה <u>לא</u> אומר ש- $x\notin L(N)$.

<u>בניית M</u>

M תהיה מ"ט עם 4 סרטים:

סרט 1 יהיה סרט הקלט שאף פעם לא נשנה את תוכנו (עליו יופיע x)

סרטים 2,3,4 יהיו הסרטים שישמשו גם את L:

- בכל הרצה של L, המ"ט M תכתוב מחדש את x על סרט 2.
- בסרט 3 נתחזק מונה לקסיקוגרפי בבסיס C_N שישמש כמחרוזת הבחירות של L.
- בעזרת סרט 4 נסמן לעצמנו האם עץ החישוב הוא סופי וכבר סרקנו את כולו.

<u>פעולת M על קלט x:</u>
<u>אתחול:</u> אתחל את סרט 3 ל- "1" ואתחל את סרט 4 ל- "אולי סופי".
<u>בצע לנצח:</u>

(א) מחק את סרט 2 והעתק את x לסרט 2
(ב) הרץ את L על סרטים 2,3,4. אם L קיבלה, **עצור וקבל**
(ג) אם המונה בסרט 3 הוא מהצורה $C_N \ldots C_N$ אז:
- אם בסרט 4 רשום "אולי סופי" אז **עצור ודחה** (המשמעות של זה היא שאף אחת מהריצות של L לא שיתנה את תוכן הסרט הזה ל- "לא סופי", ולכן כל הענפים שסרקנו היו סופיים)
- אחרת רשום על סרט 4 "אולי סופי"

(ד) הגדל את המונה בסרט 3 ב- 1 וחזור ל- (א)