תרגיל בית 4
מגיש: אור דינר
ת.ז: 207035809

שאלה 1

הראו דח"ה עבור השפה $L = \{a^n b^m : 2m < n < 4m\}$.

נתבונן בדח"ה G עם החוקים הבאים:

1. $S \to aaaAb$
2. $A \to aaAb \mid aaaAb \mid aaaaAb \mid \varepsilon$

רעיון: נכניס את המילה בעלת האורך המינימלי שמתקבלת בשפה L, ולאחריה עבור כל b שמכניסים נכניס או 2 או 3 או 4 a. אופן ההכנסה שתואר מאפשר לכל מילה שמתקבלת על ידי הדח"ה להיות מהצורה של מילים מהשפה L.

נוכיח בעזרת הכלה דו כיוונית שמתקיים $L = L(G)$.

א. $L \supseteq L(G)$ תהי $w \in L(G)$, אז w היא שרשרת גזירות על ידי הנונטרמינלים S ו-A, כך שכמות ה-a בהתחלה היא: $n = 3 + 2i + 3j + 4k$ עבור $i, j, k \in \mathbb{N}$ (כולל 0), כאשר כמות ה-b היא בדיוק $m = 1 + i + j + k$ וזאת לפי חוקי הגזירה. נבדוק האם מתקיים:

$$\mathbf{2m < n < 4m}$$
$$\mathbf{2m < 3 + 2i + 3j + 4k < 4m}$$

1. $2m < 3 + 2i + 3j + 4k$
   $2 + 2i + 2j + 2k < 3 + 2i + 3j + 4k$
   $0 < 1 + j + 2k$ ← *מתקיים תמיד*

2. $3 + 2i + 3j + 4k < 4m$
   $3 + 2i + 3j + 4k < 4 + 4i + 4j + 4k$
   $0 < 1 + 2i + j + 2k$ ← *מתקיים תמיד*

קיבלנו שכמות ה-a תמיד בין 2 כמות ה-b ל-4 כמות ה-b כפי שדרוש לשפה L, לכן כל מילה בדח"ה G תהיה גם בשפה L.

ב. $L \subseteq L(G)$ תהי $w \in L$ אז $w = a^n b^m \ s.t\ 2m < n < 4m$, על מנת לקבל את המילה הזו באמצעות הדח"ה G, נגזור את S ואז את A למשך $(m-1)$ פעמים. נקבל מילה $\bar{w} = a^{3+p} b^m$ כך ש-$n = 3 + p = 3 + 2i + 3j + 4k$ והראינו כבר שהביטוי הזה נמצא בתחום של בין $2m$ ל-$4m$. לכן כל מילה ב-L היא גם ב-$L(G)$.

שאלה 2

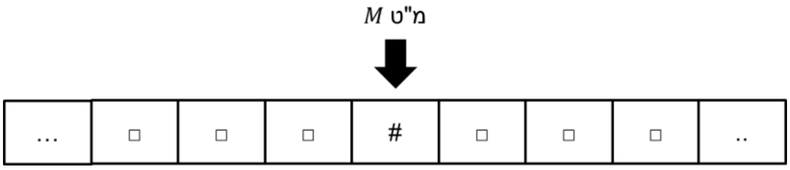
יהיו M מכונת טיורינג ו-P אוטומט מחסנית כפול.

סעיף א

נראה שאפשר לייצג א״מ בעל 2 מחסניות במ״ט באמצעות בנייה.

רעיון הבניה: נרצה שבכל זמן נתון יהיה לנו תו מיוחד על הסרט שמפריד בין תוכן המחסנית ״השמאלית״ לתוכן המחסנית ״הימנית״. כאשר המ״ט תהיה ריקה ועל מצב מקבל – המילה תתקבל, אחרת נמשיך להריץ את הא״מ.

1. אתחול המ״ט: נאתחל את המ״ט בכך שיהיה לנו סרט מלא ברווחים ובאמצע תו מיוחד #, כמו שתיארנו קודם – האיברים מימין ל-# הם תוכן המחסנית ״הימנית״, וכן על הצד השמאלי של הסרט.



באיור המ״ט מבטאת א״מ כפול כאשר 2 המחסניות ריקות.

2. בדיקת ראש המחסנית: כאשר נרצה לבדוק מה בראש המחסנית הימנית, נלך ימינה עד שנגיע לתו ריק, נלך לשמאלה צעד אחד והערך שבתא שהגענו אליו הוא ראש המחסנית הימנית. בצורה סימטרית מתבצעת בדיקת ראש המחסנית השמאלית.
3. הכנסת והוצאת איברים: בכל פעם שאנחנו עוברים על תו בא״מ אנחנו יכולים להכניס או להוציא איבר בהתאם למה שבראש המחסנית. עבור מעבר $\sigma, a_1 \to a_2, \beta_1 \to \beta_2$, נבדוק האם a_1 נמצא בצד הכי שמאלי, אם כן נמחק אותו ונכתוב במקומו a_2, ואז נחזור ימינה עד שנראה תו #. בצורה סימטרית הכנסת β_2 לצד הכי ימני.
4. תנאי קבלה: כשהמ״ט תכיל רק את # ונהיה במצב מקבל באוטומט שמבטא את המ״ט נקבל את המילה.

סעיף ב

תהי שפה $L = L(P)$ שפת הא״מ הכפולה, נראה שאפשר לייצגה בעזרת מכונת טיורינג. נזכיר כי מכונת טיורינג היא סרט עם ראש קורא שתומך בתזוזה ימינה/שמאלה קריאה וכתיבה. נוכיח שניתן לייצג מ״ט בעזרת P באמצעות בניה:

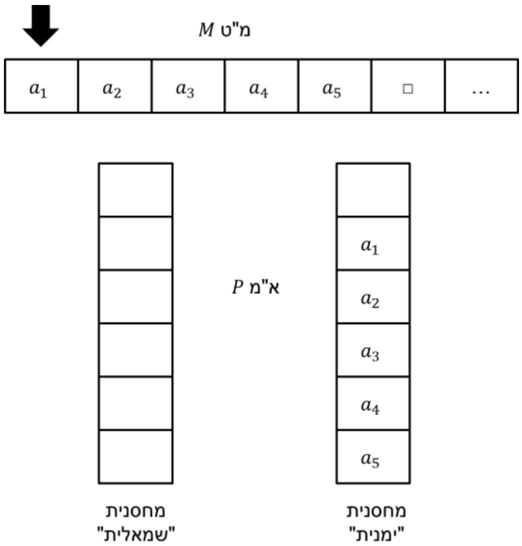
נגדיר:

$$\boldsymbol{P = (Q, \Sigma, \Gamma, \delta, q_0, F)}$$
$$Q = states$$
$$\Sigma = language\ alphabet$$
$$\Gamma = Pushdown\ alphabet$$
$$q_0 = starting\ state$$
$$F = accepting\ states$$
$$\delta: Q \times \Sigma \times \Gamma \times \Gamma \to P(Q \times \Gamma \times \Gamma)$$

נניח בה״כ ששפת 2 המחסניות זהה, כלומר האותיות שאפשר לדחוף לכל מחסנית הן מאותו א״ב.

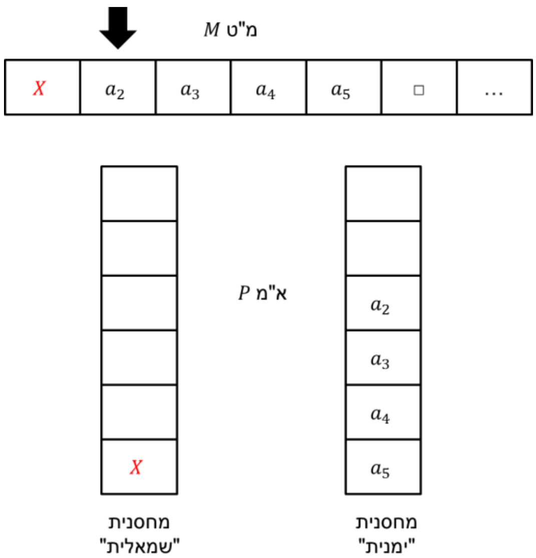
רעיון הבניה: מחסנית אחת מסמלצת את התווים שנמצאים מצד שמאל לראש הקורא, והמחסנית השנייה מסמלצת את התווים שנמצאים מצד ימין לראש הקורא.

1. אתחול המחסניות: רצים על המילה בעזרת הא״מ ומכניסים את כל האותיות של המילה למחסנית הראשונה (״השמאלית״) (שמסמלצת את מה שנמצא מצד שמאל לראש הקרוא). אחר כך מרוקנים את המחסנית הראשונה לתוך השנייה (״הימנית״) ומגיעים ״למצב אפס״ שבו כל מה שנמצא מצד ימין לראש הקורא במ״ט שאנו רוצים לסמלץ נמצא במחסנית ״הימנית״ ובהתחלה במחסנית ״השמאלית״ אין כלום.



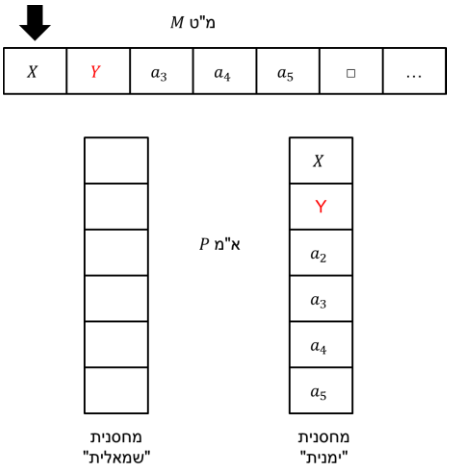
ניתן להגיע למצב הזה בעזרת מעבר על כל אותיות המילה והכנסתן למחסנית ״השמאלית״ ע״י המעבר $\varepsilon \to \varepsilon, a_n \to a_n, a_{n-1}$ עבור $w = a_1 a_2 \ldots a_n$, ואז כאשר הגענו למצב בו אנחנו בסוף המילה משתמשים במעבר $a_n \to \varepsilon, \varepsilon \to \varepsilon, a_n$ (רואה a_n במחסנית ״השמאלית״ ודוחף אותו למחסנית ״הימנית״) כדי למלא את המחסנית הימנית. לבסוף נקבל שהמחסניות מייצגות את המ״ט לפני תחילת פעולתה.

2. <u>תזוזה ימינה במ״ט</u>: כאשר נרצה לזוז ימינה במ״ט, באוטומט המחסנית אנחנו נעשה POP למה שנמצא בראש המחסנית ״הימנית״ ונדחוף לראש המחסנית ״השמאלית״ את מה שרצינו לכתוב באותו תא שזזנו ממנו ימינה (אם לא שינינו את ערך התא בסרט – נדחוף את ערך התא. אם שינינו אותו נדחוף את הערך החדש שנכתב למחסנית ״השמאלית״).



בדוגמא הזו הראש הקורא רואה a_1, זז ימינה וכותב במקומו X.

3. <u>תזוזה שמאלה במ״ט</u>: כאשר נרצה לזוז שמאלה במ״ט, באוטומט המחסנית אנחנו נעשה POP ו-$PUSH$ למה שאנחנו רוצים לכתוב לתוך המחסנית ״הימנית״, ונעשה POP ונדחוף למחסית ״הימנית״ את מה שנמצא בראש המחסנית ״השמאלית״ .



בדוגמא הזו הראש הקורא זז ימינה והחליף את a_2 ב-Y.

4. <u>תנאי קבלה</u>: בשימוש הנוכחי במחסניות הן לא תמיד יתרוקנו, לכן נגדיר שכאשר הא״מ הגיע למצב מקבל המילה מתקבלת. בגלל שאנחנו מכניסים את האותיות בצורה דטרמיניסטית לתוך המחסניות, נרצה לסמלץ מתי המ״ט תקבל את המילה, אבל אין אפשרות לדעת מתי המ״ט מקבלת רק על פי הסרט אלא על ידי המצבים שלה (לו היינו בונים לה אוטומט). ולכן ניתן להגדיר מצבים מקבלים כמון המצבים המקבלים במ״ט.

שאלה 3

$$ACCEPT = \{< M, w >: M\ accepts\ w\}$$
$$REJECT = \{< M, w >: M\ rejects\ w\}$$

סעיף א

$$ACCEPT \neq \overline{REJECT}$$

לפי הגדרת $REJECT$ אנחנו יכולים לומר שהמשלים שלה הוא השפה $\overline{REJECT} = \{< M, w >: M\ does\ not\ reject\ w\}$, כלומר, ב-$\overline{REJECT}$ יהיו מכונות הטיורינג שמקבלות **או** נכנסות ללולאה אינסופית עבור ה-w שלהן, והשפה $ACCEPT$ מהווה רק חלק מהגדרה – בכך שהיא מכילה מכונות טירינג שרק מקבלות את ה-w שלהן ולא מכסה את המקרה בו המ״ט תיכנס ללולאה אינסופית. ישנן מכונות טיורינג שייכנסו ללולאה אינסופית על ה-w שלהן ולא יהיו חלק מ-$ACCEPT$.

סעיף ב

$$ACCEPT \leq_m REJECT$$

נבנה מ״ט R שבהינתן $< M, w >$ קידוד של מ״ט מ-ACCEPT – רצה, עוצרת, ופולטת קידוד $< M', w' >$ כך שיתקיים:

$$< M, w > \in ACCEPT \iff < M', w' > \in REJECT$$

ומתקיים: $\boldsymbol{R(< M, w >) = < M', w' >}$

נגדיר:

$M'(x)$:

1. הרץ את M על w:
   1.1. אם קיבלה – דחה.
   1.2. אם דחתה או נכנסה ללולאה – קבל.

נוכיח את הנכונות של הבנייה:

- אם $< M, w > \in ACCEPT$, אז M מקבלת את w $\Leftarrow$ M' מריצה את $M(w)$ ודוחה תמיד, כלומר $L(M) = \emptyset$.
  ומשום ששפת המ״ט היא השפה הריקה בפרט היא תדחה כל מילה ולכן היא תהיה חלק מ-$REJECT$:
  $R(x) = < M', w' > \in REJECT$.
- אם $< M, w > \notin ACCEPT$, אז M לא מקבלת את w $\Leftarrow$ M' מריצה את $M(w)$ ומקבלת תמיד, כלומר $L(M) = \Sigma^*$. ומשום ששפת המ״ט היא השפה של כל המילים, היא תקבל כל מילה ולכן לא תהיה חלק מ-$REJECT$:
  $R(x) = < M', w' > \notin REJECT$.

סעיף ג

$$REJECT \leq_m ACCEPT$$

נבנה מ״ט R שבהינתן $< M, w >$ קידוד של מ״ט מ-REJECT – רצה, עוצרת, ופולטת קידוד $< M', w' >$ כך שיתקיים:

$$< M, w > \in REJECT \iff < M', w' > \in ACCEPT$$

ומתקיים: $\boldsymbol{R(< M, w >) = < M', w' >}$

נגדיר:

$M'(x)$:

1. הרץ את M על w:
   1.1. אם קיבלה – דחה.
   1.2. אם דחתה – קבל.

נוכיח את הנכונות של הבנייה:

- אם $< M, w > \in REJECT$, אז M דוחה את w $\Leftarrow$ M' מריצה את $M(w)$ ומקבלת תמיד, כלומר $L(M) = \Sigma^*$.
  ומשום ששפת המ״ט היא השפה של כל המילים בפרט היא תקבל כל מילה ולכן היא תהיה חלק מ-$ACCEPT$:
  $R(x) = < M', w' > \in ACCEPT$.
- אם $< M, w > \notin REJECT$, אז M מקבלת את w (או נכנסת ללולאה) $\Leftarrow$ M' מריצה את $M(w)$ ודוחה תמיד, כלומר $L(M) = \emptyset$. ומשום ששפת המ״ט היא השפה הריקה, היא תדחה כל מילה ולכן לא תהיה חלק מ-$REJECT$ עבור כל מילה:
  $R(x) = < M', w' > \notin REJECT$.

שאלה 4

הוכח שהשפה $L = \{< M, k >: |L(M)| > k \ , k \in \mathbb{N}\}$ אינה כריעה אך ניתנת לקבלה.

נוכיח את הטענה באמצעות רדוקציה מ-ACCEPT, כך ש: $ACCEPT \leq_m L$.

נבנה מ"ט R שבהינתן $< M, w >$ קידוד של מ"ט מ-ACCEPT – רצה, עוצרת, ופולטת קידוד $< M', k >$ כך שיתקיים:

$$< M, w >\in ACCEPT \ \Leftrightarrow < M', k >\in L$$

ומתקיים: $\boldsymbol{R(< M, w >) =< M', k >}$

נגדיר:

$M'(x)$:

Const a = <M>, b = w

1. הרץ את M על w והחזר מה שהיא מחזירה.

נוכיח את הנכונות של הבנייה:

- אם $< M, w >\in ACCEPT$, אז M מקבלת את $w \Leftarrow M'$ מריצה את $M(w)$ ומחזירה את מה ש-M מחזירה, כלומר מקבלת כל x. וזה אומר ששפת M' מכילה אינסוף מילים כי $L(M') = \Sigma^*$. לכן מתקיים: $R(x) =< M', k >\in L$.
- אם $< M, w >\notin ACCEPT$, אז M לא מקבלת את $w \Leftarrow M'$ מריצה את $M(w)$ ומחזירה את מה ש-M מחזירה, כלומר דוחה כל x. וזה אומר ששפת M' מכילה 0 מילים כי $L(M') = \emptyset$ ולכן מתקיים: $R(x) =< M', k >\notin L$ כי מ"ט ששפתם ריקה אינם נכללים ב-L.

הראינו רדוקציה נכונה מ-ACCEPT וזה אומר ש-L ניתנת לקבלה, ומשום שמתקיים עבור רדוקציות מיפוי עבור $L_1 \leq_m L_2$:

1. L_2 כריעה $\leftarrow L_1$ כריעה. במקרה שלנו אנחנו יודעים ש-ACCEPT לא כריעה ולכן L לא יכולה להיות כריעה (אחרת ACCEPT הייתה כריעה).

שאלה 5

הוכח שהשפות הבאות אינן ניתנות לקבלה.

סעיף א

$$L = \{< M, x >: words\ in\ M\ don't\ start\ with\ x\}$$

נוכיח את הטענה באמצעות רדוקציה מ-$NOT - ACCEPT$.

נבנה מ״ט R שבהינתן $< M, w >$ קידוד של מ״ט מ-$NOT - ACCEPT$ – רצה, עוצרת, ופולטת קידוד $< M', x >$ כך שיתקיים:

$$< M, w > \in NOT - ACCEPT \iff < M', x > \in L$$

ומתקיים: $\boldsymbol{R(< M, w >) =< M', x >}$

נגדיר:

$M'(x)$:

Const a = <M>, b = w

1. החזר את התוצאה ההפוכה של ההרצה של M על w. (אם M מקבלת את w – דחה, ולהיפך).

נוכיח את הנכונות של הבנייה:

- אם $< M, w > \in NOT - ACCEPT$, אז M לא מקבלת את $w \Leftarrow M'$ מריצה את $M(w)$ ודוחה תמיד. לכן לא תקבל אף מילה שמתחילה ב-x. ובכזה מצב מתקיים: $R(x) =< \boldsymbol{M', x} > \in L$ כי השפה של M' ריקה.
- אם $< M, w > \notin NOT - ACCEPT$, אז M מקבלת את $w \Leftarrow M'$ מקבלת את כל המילים תמיד אז מתקיים $L(M') = \Sigma^*$. לכן מתקיים: $R(x) =< M', k > \notin L$ כי M' תקבל גם מילים שמתחילות ב-x (מילים כאלה נמצאות ב-Σ^*).

סעיף ב

$$L = \{< M_1, M_2 >: L(M_1)\ has\ a\ finite\ amount\ of\ words\ that\ are\ not\ in\ L(M_2)\}$$

נמצא רדוקציה ל-L מ-$NOT - ACCEPT$.

נבנה מ״ט R שבהינתן $< M, w >$ קידוד של מ״ט מ-$NOT - ACCEPT$ – רצה, עוצרת, ופולטת קידוד $< M_1, M_2 >$ כך שיתקיים:

$$< M, w > \in NOT - ACCEPT \iff < M_1, M_2 > \in L$$

ומתקיים: $\boldsymbol{R(< M, w >) =< M_1, M_2 >}$

נגדיר:

M_1 –

1. אם הקלט הוא המילה הריקה ε – קבל, אחרת. (יכולנו לשים כל מילה אחרת באותה מידה)
2. הרץ את M על w ופעל כמוה.

M_2 – מכונת טיורינג שדוחה כל מילה. כלומר $L(M_2) = \emptyset$

נוכיח שהרדוקציה עובדת.

- אם $< M, w > \in NOT - ACCEPT$, אז M דוחה את $w \Leftarrow M_1$ מקבלת רק את הקלט שהוא המילה הריקה, עבור כל מילה שהיא לא ε נריץ את M על w ונפעל כמוה, ומשום ש-M דוחה את w לא נקבל אף מילה שאינה ε. כלומר השפה של M_1 היא $L(M_1) = \{\varepsilon\}$, כלומר $|L(M_1)| = 1$ לכן מתקיים גם $< M_1, M_2 > \in L$ כי יש בשפה של M_1 כמות קבועה של מילים שאין ב-M_2.
- אם $< M, w > \notin NOT - ACCEPT$, אז M מקבלת את $w \Leftarrow M_1$ מקבלת את המילה הריקה, וגם את כל המילים שאינן המילה הריקה לפי צעד 2, אז השפה של M_1 היא שפת כל המילים מעל הא״ב, כלומר יש כמות אינסופית של מילים ב-M_1 וכמות סופית (0) של מילים ב-M_2 לכן מתקיים גם $< M_1, M_2 > \notin L$.

שאלה 6

סעיף א

$$L = \{a^n b^m : n \neq 7m\}$$

השפה הינה ח״ה.

נשים לב ש- $\bar{L} = \{a^n b^m : n = 7m\}$, ובנוסף, $L \cup \bar{L} = \{a^n b^m\}$ נוכיח שהאיחוד הוא שפה חסרת הקשר ואז לפי סגירות של איחוד שפות ח״ה בגלל שהאיחוד ח״ה (הראינו בכיתה – אפשר לבנות דח״ה), אז כל קבוצה לחוד צריכה להיות ח״ה.

ניתן לתאר את $\bar{L}$ באמצעות הדח״ה הבא:

$$S \rightarrow aSbbbbbbb \mid \varepsilon$$

קל לראות שהדח״ה מתאר את $\bar{L}$ בדיוק, ובגלל שיש לה דח״ה היא ח״ה, ואז מתקיים $L \cup \bar{L} = \{a^n b^m\}$ חסרת הקשר וגם $\bar{L}$, אז בהכרח L ח״ה.

כעת נוכיח שאינה רגולרית באמצעות למת הניפוח לשפות רגולריות:

עבור כל w שמקיים ש- $|w| \geq n_0$, קיים n_0 כך שפירוק למילה $w = xyz$, ומתקיים:

1. $|xy| \leq n_0$
2. $y \neq \varepsilon$
3. ולכל $k > 0$ מתקיים $xy^k z \in L$

נבחן את המילה $w = a^{n_0} b^{14n_0}$, לפי התנאי 1 ו-2 y מכיל רק את האות a. כאשר ננפח את המילה באמצעות $k = \frac{n_0}{|y|} + 1$ נקבל שהמילה המנופחת תהיה $\bar{w} = a^{2n_0} b^{14n_0}$, אבל היא לא מקיימת את תנאי 3 ולא חלק מהשפה, ולכן היא אינה רגולרית.

סעיף ב

$$L = \{< M >: M\ stops\ for\ 010\}$$

השפה ניתנת לקבלה אבל לא כריעה, נוכיח את זה בעזרת רדוקציה מ-HALT.

נבנה מ״ט R שבהינתן $< M, w >$ קידוד של מ״ט מ-$HALT$ – רצה, עוצרת, ופולטת קידוד $< M' >$ כך שיתקיים:

$$< M, w > \in HALT \iff < M' > \in L$$

ומתקיים: $\boldsymbol{R(< M, w >) = < M' >}$

נגדיר:

$M'(x)$:

Const a = <M>

1. בדוק האם x מכיל 010.
   1.1. הרץ את M על w ופעל כמוה.
2. דחה.

נוכיח את הנכונות של הבנייה:

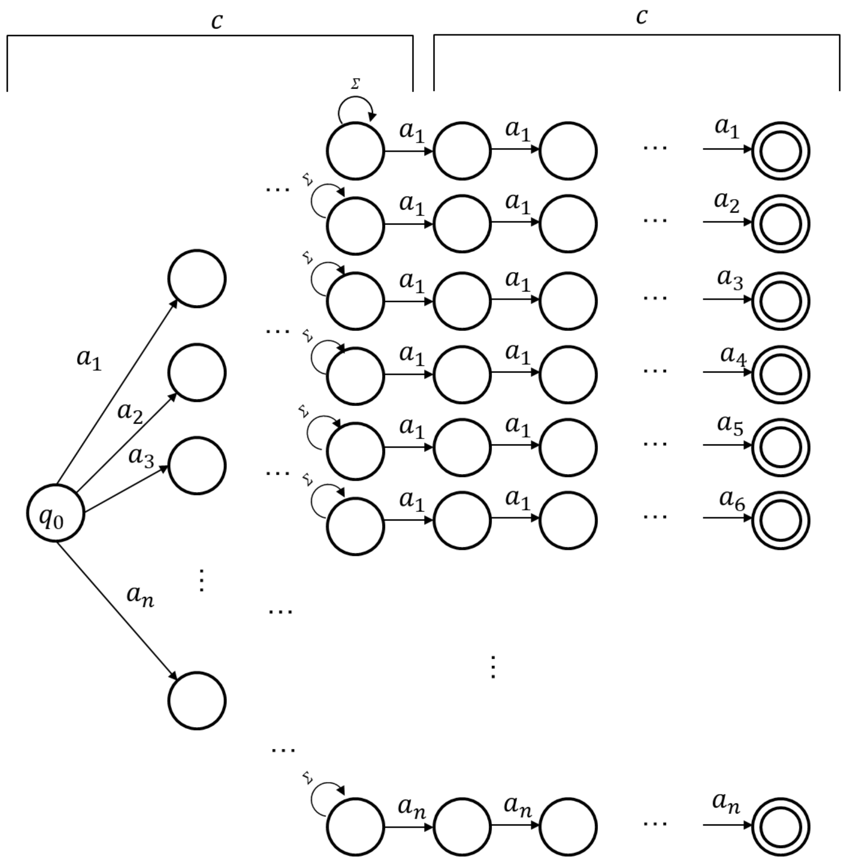
- אם $< M, w > \in HALT$, אז M עוצרת על $w \Leftarrow M'$ בודקת אם הקלט שלה מכיל 010, אם כן היא תריץ את M על w ותקבל, ואם לא היא תדחה. ובכזה מצב מתקיים: $R(x) = < M' > \in L$ כי השפה של M' היא כל מכונות הטיורינג שמקבלות מילה עם 010.
- אם $< M, w > \notin HALT$, אז M לא עוצרת על $w \Leftarrow M'$ בודקת אם הקלט שלה מכיל 010, אם כן היא תריץ את M על w ותיכנס ללולאה אינסופית (כי M לא עוצרת על w), ואם לא היא תדחה. ובכזה מצב מתקיים: $R(x) = < M' >$.
  הראינו רדוקציה מ-HALT לכן L אינה כריעה וניתנת לקבלה.

סעיף ג

$$L_c = \{w : w = uxu\ and\ u, x \in \Sigma^*\ and\ |u| = c\}$$

השפה רגולרית, נראה אסל״ד שמקבל אותה ונוכיח שהיא השפה בעזרת הכלה דו כיוונית.

<u>רעיון</u>: ניצור עץ של כל המילים האפשריות מאורך c כך ש-q_0 הוא ראש העץ, כל מסלול לעלה הוא החלק בקלט שהוא ה-u הראשון. לכל עלה נוציא חץ לעצמו עם כל האותיות האפשריות – וזה מייצג את החלק שהוא x בקלט. מכל עלה נוציא מסלול עם c מצבים כך שהחיצים בו הם בדיוק כמו במסלול שעשינו כדי להגיע לאותו עלה – זה מייצג את ה-u השניה בקלט.



אכן מדובר בכמות קבועה של מצבים, $n = |\Sigma|, c$ קבועים.

אפשר לראות שכל מילה מהצורה uxu תתקבל על ידי האוטומט כי עבור כל מילה u ניקח את המסלול המתאים בעץ למילה הזאת, ומשם יש רק אפשרות אחת למסלול מקבל והיא לקלוט רק את המילה שנקלטה בהתחלה.