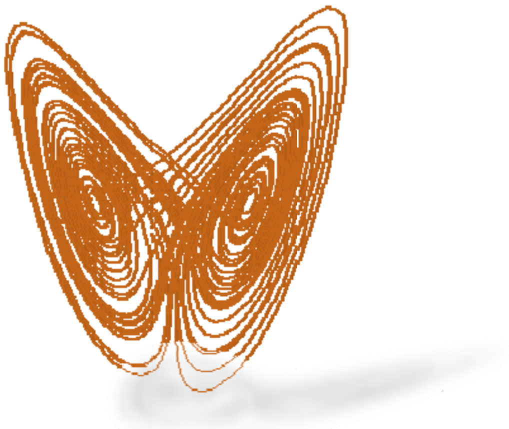
DinaZil's Notes-Heaven

מציגים

חישוביות

המבחן



סוכם, עובד והוקלד ע"י דינה זליגר

מבוסס על הרצאותיה של **פרופ' אורנה קופרמן** ותרגוליהם של **רובי למפורט ויועד לוסטיג**

Please read the following important legal information before reading or using these notes. The use of these notes constitutes an agreement to abide by the terms and conditions below, just as if you had signed this agreement.

A. THE SERVICE.

The following notes ("The service") are provided by DinaZil's Notes-Heaven ("Notes-Heaven").

B. DISCLAIMER OF WARRANTIES; LIMITATION OF LIABILITY.

Notes-Heaven does not endorse content, nor warrant the accuracy, completeness, correctness, timeliness or usefulness of any opinions, advice, content, or services provided by the Service.

YOU AGREE THAT USE OF THE SERVICE IS ENTIRELY AT YOUR OWN RISK. THE SERVICE PROVIDED IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND. NOTES-HEAVEN EXPRESSLY DISCLAIMS ALL WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION: ANY WARRANTIES CONCERNING THE ACCURACY OR CONTENT OF INFORMATION OR SERVICES. NOTES-HEAVEN MAKES NO WARRANTY THAT THE SERVICE WILL MEET YOUR REQUIREMENTS, OR THAT THE SERVICE WILL BE ERROR FREE; NOR DOES NOTES-HEAVEN MAKE ANY WARRANTY AS TO THE RESULTS THAT MAY BE OBTAINED FROM THE USE OF THE SERVICE OR AS TO THE ACCURACY OR RELIABILITY OF ANY INFORMATION OBTAINED THROUGH THE SERVICE. YOU UNDERSTAND AND AGREE THAT ANY DATA OBTAINED THROUGH THE USE OF THE SERVICE IS DONE AT YOUR OWN DISCRETION AND RISK AND THAT YOU WILL BE SOLELY RESPONSIBLE FOR ANY DAMAGE TO YOUR GPA.

NEITHER NOTES-HEAVEN NOR ANY OF ITS PARTNERS, AGENTS, AFFILIATES OR CONTENT PROVIDERS SHALL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF USE OF THE SERVICE OR INABILITY TO GAIN ACCESS TO OR USE THE SERVICE OR OUT OF ANY BREACH OF ANY WARRANTY.

C. INDEMNIFICATION.

You agree to indemnify and hold Notes-Heaven, its partners, agents, affiliates and content partners harmless from any dispute which may arise from a breach of terms of this Agreement. You agree to hold Notes-Heaven harmless from any claims and expenses, including reasonable attorney's fees and court costs, related to your violation of this Agreement.

D. OWNERSHIP RIGHTS.

The materials provided by the Service may be downloaded or reprinted for personal use only. You acknowledge that the Service contains information that is protected by copyrights, trademarks, trade secrets or other proprietary rights, and that these rights are valid and protected in all forms, media and technologies existing now or hereafter developed. You may not modify, publish, transmit, participate in the transfer or sale, create derivative works, or in any way exploit, any of the Content, in whole or in part. You may not upload, post, reproduce or distribute Content protected by copyright, or other proprietary right, without obtaining permission of the owner of the copyright or other proprietary right.

E. NO COPYING OR DISTRIBUTION.

You may not reproduce, copy or redistribute the design or layout of this service, individual elements of the design, Notes-Heaven logos or other logos appearing on this service, without the express written permission of Notes-Heaven, Inc. Reproduction, copying or redistribution for commercial purposes of the service is strictly prohibited without the express written permission of Notes-Heaven, Inc.

If you have any questions about this statement or the practices of this service you can contact

Dina Zeliger
dinaweb@gmail.com

תוכן עניינים

הקדמה

סימונים וטרמינולוגיה מתמטית

קבוצות

הגדרות:

1. **קבוצה** היא אוסף של איברים, ללא חשיבות לסדר או למספר ההופעות של כל איבר. לכל איבר בעולם או שהוא שייך לקבוצה או שלא. מסמנים זאת ע"י $a \in A$ ו-$a \notin A$ בהתאמה.
2. תהי A קבוצה. נאמר ש-B היא **תת קבוצה** של A ונסמן $B \subset A$ אם לכל $x \in B$ מתקיים $x \in A$.
3. קבוצת כל תת הקבוצות של קבוצה A נקראת **קבוצת החזקה** ומסומנת ע"י 2^A או $P(A)$.
4. תהיינה A, B קבוצות. נאמר שהן **שוות** ונסמן $A = B$ אם $B \subset A$ וגם $A \subset B$, כלומר לשתי הקבוצות יש בדיוק אותם איברים.
5. הקבוצה שאין בה אף איבר נקראת **הקבוצה הריקה** ומסומנת ע"י $\varnothing$.
6. נניח שהעולם שבו אנחנו עובדים הוא קבוצה U. תהיינה $A, B \subset U$ קבוצות בעולם.
   - איחוד: $A \cup B = \{x \in U : x \in A \vee x \in B\}$
   - חיתוך: $A \cup B = \{x \in U : x \in A \wedge x \in B\}$
   - משלים: $A^c = \{x \in U : x \notin A\}$
   - הפרש: $A \setminus B = \{x \in U : x \in A \wedge x \notin B\}$

דוגמה: תהיינה $A = \{1,2,3\}, B = \{1,2,3,4,5,6,7\}, C = \{2,4,5,8\}$. אזי:

- $3 \in A$ ואילו $3 \notin C$
- $A \subset B$
- $A \cup C = \{1,2,3,4,5,8\}$
- $B \cap C = \{2,4,5\}$
- $B \setminus A = \{4,5,6,7\}$

הגדרה: יהיו A, B קבוצות. **המכפלה הקרטזית** של A, B מסומנת ע"י $A \times B$ והיא שווה לכל הזוגות הסדורים בהם האיבר הראשון הוא מתוך A והאיבר השני הוא מתוך B. כלומר $A \times B = \{(a,b) : a \in A, b \in B\}$.

יחסים

הגדרות:

1. **יחס בינארי** R בין A ל-B הוא תת קבוצה של המכפלה $A\times B$, כלומר $R\subset A\times B$. אם a מתייחס ל-b מסמנים aRb או $R(a,b)$.
2. יחס $R\subset A\times A$ נקרא:
   - **רפלקסיבי** אם לכל $a\in A$ מתקיים aRa
   - **סימטרי** אם לכל $a,b\in A$ aRb אמ"מ bRa
   - **טרנזיטיבי** אם לכל $a,b,c\in A$ אם aRb וגם bRc אז aRc
3. יחס שהוא רפלקסיבי, סימטרי וטרנזיטיבי נקרא **יחס שקילות**.
4. עבור יחס שקילות $R\subset A\times A$ **מחלקת השקילות** של a לפי R היא הקבוצה $[a]_R=\{b\in A: aRb\}$.
5. **חלוקה** של קבוצה A היא אוסף זר של קבוצות חלקיות ל-A המכסות את A (כלומר איחודן שווה ל-A).

דוגמאות:

1. היחס $"<"\subset\mathbb{N}\times\mathbb{N}$ האומר ש-$a<b$ אמ"מ a קטן ממש מ-b
2. בגרף $G=\langle V,E\rangle$ קבוצת הצלעות E היא יחס על הקודקודים שכן $E\subset V\times V$
3. נגדיר יחס $R\subset\mathbb{N}\times\mathbb{N}$ ע"י aRb אמ"מ $a\equiv b(\mathrm{mod}\,3)$. קל לראות שהו יחס שקילות והוא מחלק את $\mathbb{N}$ לשלוש מחלקות שקילות:
   - $[7]_R=\{1,4,7,...\}$
   - $[0]_R=\{0,3,6,9,...\}$
   - $[5]_R=\{2,5,8,...\}$
4. נסתכל על היחסים הבאים על $\mathbb{N}$:

יחס	רפלקסיבי	סימטרי	טרנזיטיבי
$<$	-	-	+
$\leq$	+	-	+
$=$	+	+	+
$a\in[b-1,b+1]$	+	+	-

פונקציות

הגדרות:

1. יחס $R \subset A \times B$ נקרא **מלא** אם כל $a \in A$ משתתף ביחס.
2. יחס $R \subset A \times B$ נקרא **חד ערכי** אם לכל $a \in A$ שמשתתף ביחס קיים $b \in B$ יחיד כך ש-aRb.
3. **פונקציה** מ-A ל-B היא יחס מלא וחד ערכי $f \subset A \times B$ (לכל $a \in A$ קיים איבר יחיד $b \in B$ כך ש-$f(a,b)$). במקרה זה נסמן $f : A \to B$ ו-$f(a) = b$.
4. פונקציה $f : A \to B$ היא **חד חד ערכית** (להלן חח"ע) אם לכל $b \in B$ יש לכל היותר $a \in A$ יחיד כך ש-$f(a) = b$.
5. פונקציה $f : A \to B$ היא **על** אם לכל $b \in B$ קיים $a \in A$ כך ש-$f(a) = b$.
6. תהיינה $f : A \to B, g : B \to C$. **ההרכבה** $g \circ f$ היא פונקציה מ-A ל-C שמקיימת $g \circ f(a) = g(f(a))$ לכל $a \in A$.
7. נאמר ש-$f : A \to B$ **הפיכה** אמ"מ קיימת $g : B \to A$ כך ש-$f \circ g = id = g \circ f$. במקרה זה נסמן $g = f^{-1}$ ונאמר ש-g היא **ההופכית** של f.

משפט: תהי $f : A \to B$. אזי f חח"ע ועל אמ"מ f הפיכה.

עוצמות של קבוצות

הגדרות:

1. תהי A קבוצה סופית. **העוצמה** של A שתסומן ע"י $|A|$ היא מספר האיברים (השונים) ב-A.
2. תהיינה שתי קבוצות A, B. נאמר ש-A, B **שוות עוצמה** ונסמן $|A| = |B|$ אם קיימת פונקציה הפיכה $f : A \to B$.
3. קבוצה A תיקרא **בת מניה** אם $|A| = |\mathbb{N}|$. מסמנים $|\mathbb{N}| = \aleph_0$.
4. נאמר ש-$|A| < |B|$ אם קיימת פונקציה $f : A \to B$ שהיא חח"ע אך אינה על, או לחילופין קיימת $g : B \to A$ על שאינה חח"ע.

דוגמאות:

1. $|\mathbb{N}|=|2\mathbb{N}|$ ע"י הפונקציה $f(n)=2n$
2. $|\mathbb{Z}|=|\mathbb{N}|$. כדי להראות זאת יש להראות שניתן לסדר את איברי $\mathbb{Z}$ ברשימה וזה פשוט: $0,1,-1,2,-2,3,-3,4,-4,5,-5,...$. או באופן פורמאלי $f(x)=\left\lfloor \frac{x}{2} \right\rfloor (-1)^x$
3. $|\mathbb{N}|=|\mathbb{Q}^+|$

טענה: לכל קבוצה A מתקיים $|A|<|2^A|$

טענה: קיימת קבוצה שאיננה בת מניה.

הוכחה: נסתכל על הקטע $(0,1]$. כל $x\in(0,1]$ ניתן לייצוג יחיד ע"י $0.a_1a_2...a_n...$ כאשר $a_i\in\{0,...,9\}$. נניח בשלילה ש-$(0,1]$ בן מניה. אזי ניתן לרשום את כל האיברים שלו באופן הבא:

$$\begin{array}{l} 0.a_1^1a_2^1...a_n^1a_{n+1}^1... \\ 0.a_1^2a_2^2...a_n^2a_{n+1}^2... \\ \vdots \\ 0.a_1^na_2^n...a_n^na_{n+1}^n... \\ \vdots \end{array}$$

נסתכל על המספר $0.a_1a_2...a_k...\in(0,1]$ כאשר לכל $k\in\mathbb{N}$ מתקיים $a_k\neq a_k^k$. המספר הזה לא נמצא ברשימת המספרים שלנו, שהרי אם הוא היה רשום בשורה ה-n אז מתקיים $0.a_1a_2...a_n...=0.a_1^na_2^n...a_n^n...$, אבל לפי הגדרת המספר $a_n\neq a_n^n$ בסתירה.

☺

חלק ראשון

אוטומטים ושפות

שפות רגולריות

אוטומטים סופיים

הגדרות:

1. **אלפבית** (א"ב) הוא קבוצה סופית של סימנים $\Sigma = \{\sigma_1, ..., \sigma_n\}$. הסימנים בא"ב נקראים **אותיות**.
2. **מילה** היא סדרה סופית של אותיות $w = w_1 w_2 ... w_k$. המילה שאין בה אף אות נקראת **המילה הריקה** ומסומנת ע"י ε. אם כל אותיות מילה w הן מתוך א"ב Σ נאמר ש-w היא **מעל** Σ.
3. כל המילים מעל Σ הן $\Sigma^* = \{w : w \text{ is a word over } \Sigma\}$
4. **שפה** היא קבוצה כלשהי של מילים - $L \subset \Sigma^*$.

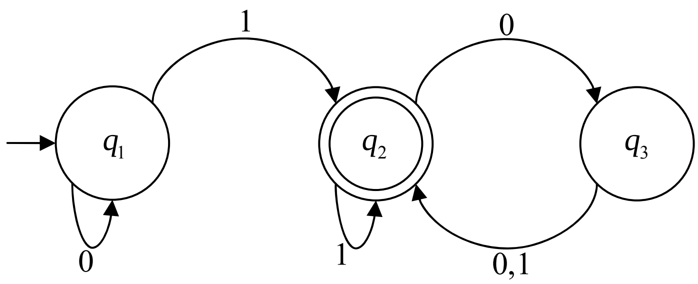
הגדרה: **אוטומט סופי דטרמיניסטי** (להלן אס"ד) הוא חמישייה $\langle Q, \Sigma, \delta, q_0, F \rangle$ כאשר:

1. Q קבוצה סופית של מצבים
2. Σ א"ב
3. $\delta : Q \times \Sigma \to Q$ פונקציית מעברים
4. $q_0 \in Q$ מצב התחלתי
5. $F \subset Q$ קבוצת מצבים מקבלים

דוגמה: $M_1 = \langle \{q_1, q_2, q_3\}, \{0,1\}, \delta, q_1, \{q_2\} \rangle$ ו- δ מוגדרת ע"י הטבלה הבאה:

δ	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

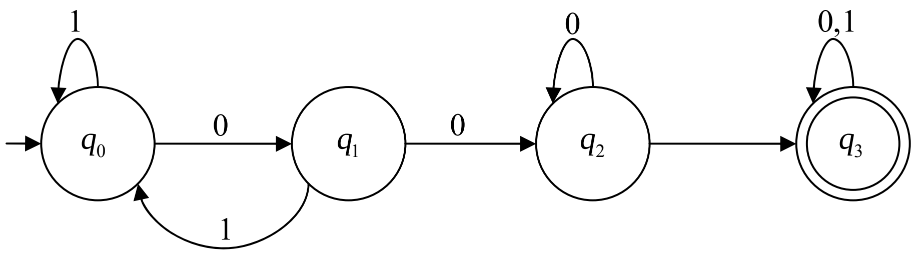
את האס"ד הזה ניתן לתאר בעזרת ציור באופן הבא:



הגדרות:

1. בהינתן מילה $w = w_1 \dots w_n$, **ריצה** r של אס"ד M על w היא סדרת מצבים $r = r_0 r_1 \dots r_n$ ב-Q כך שמתקיימות התכונות הבאות:
   - $r_0 = q_0$ (הריצה מתחילה במצב ההתחלתי)
   - לכל $1 \le i \le n-1$ מתקיים $r_{i+1} = \delta(r_i, w_{i+1})$ (הריצה מתקדמת בהתאם לפונקציית המעברים)
2. ריצה $r = r_0 \dots r_n$ היא **מקבלת** אם $r_n \in F$, אחרת r **דוחה**.
3. אס"ד M **מקבל** מילה w אם הריצה של M על w היא מקבלת, אחרת M **דוחה** את w.
4. **שפה של אס"ד** היא קבוצת כל המילים שהוא מקבל - $L(M) = \left\{ w \in \Sigma^* : M \text{ accepts w} \right\}$

דוגמה: נבנה אס"ד M_2 כך ש-$L(M_2) = \left\{ w \in \{0,1\}^* : w \text{ conatains the sequence } 001 \right\}$.



למצבים יש משמעות:

- q_0 - לפני שהרצף 001 מתחיל
- q_1 - קראנו 0
- q_2 - קראנו 00
- q_3 - קראנו 001

ניתן לראות שאחרי שכבר הגענו למצב q_3 אנחנו נישאר בו לנצח, ללא קשר לקלט. במקרה זה הסיבה לכך היא שכבר בדקנו שהרצף 001 נמצא במילה ומהרגע שמצאנו אותו לא אכפת לנו מהן שאר האותיות במילה. מצב כזה נקרא **בור מקבל**. לעומת זאת, יכול להיות מצב שברגע שהגענו אליו כבר ברור שהמילה אינה בשפה ולכן כל קלט ישאיר אותנו במצב זה. מצב כזה נקרא **בור דוחה**.

דוגמה: לא תמיד כדאי לצייר אס"ד. למשל נסתכל על השפה

$$L_n = \{w : w \text{ contains the sequence } 0^n 1\}$$

במקרה שידוע מהו n ניתן לצייר את האס"ד בלי בעייה. למשל, בגומה הקודמת ציירנו אס"ד עבור L_2. אבל באופן כללי עדיף פשוט לתת תיאור פורמאלי של האס"ד באופן הבא:

$$M_n = \langle \{q_0, q_1, \ldots, q_{n+1}\}, \{0,1\}, q_0, \delta, \{q_{n+1}\} \rangle$$

כאשר δ מוגדרת באופן הבא:

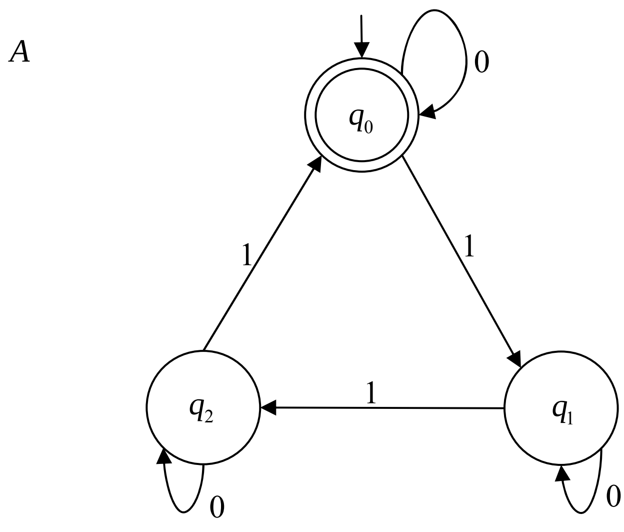
- עבור q_0 נגדיר $\delta(q_0,0) = q_1$ ו- $\delta(q_0,1) = q_0$
- עבור $1 \le i \le n-1$ נגדיר $\delta(q_i,0) = q_{i+1}$ ו- $\delta(q_i,1) = q_0$
- עבור q_n נגדיר $\delta(q_n,0) = q_n$ ו- $\delta(q_n,1) = q_{n+1}$
- q_{n+1} מצב מקבל, כלומר $\delta(q_{n+1},0) = q_{n+1} = \delta(q_{n+1},1)$

הגדרה: יהי $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ אס"ד. נגדיר פונקציה $\delta^* : Q \times \Sigma^* \to Q$ באינדוקציה על אורך מילת הקלט:

1. $\delta^*(q, \varepsilon) = q$
2. עבור $u \in \Sigma^*$ ו- $a \in \Sigma$ $\delta^*(q, ua) = \delta(\delta^*(q,u), a)$

למעשה $\delta^*(q,w)$ הוא המצב שאליו תגיע A בריצה על w שמתחילה מהמצב q. ואז $\delta^*(q_0, w) \in F$ אמ"מ $w \in L(A)$.

דוגמה: נבנה אס"ד עבור השפה $L = \left\{ w_1 \ldots w_n \in \{0,1\}^* : \sum_{i=1}^{n} w_i \equiv 0 \pmod 3 \right\}$:



פונקציית המעברים של A מוגדר ע"י $\delta(q_i,0)=q_i$ ו- $\delta(q_i,1)=q_{(i+1)\bmod 3}$ לכל $0\le i\le 2$.

נטען שלכל $w=w_1...w_n\in\{0,1\}^*$ מתקיים $\delta^*(q_0,w)=q_j$ כאשר $j=\left(\sum_{i=1}^{n}w_i\right)\bmod 3$. נוכיח זאת באינדוקציה על האורך של w.

1. אם $w=\varepsilon$ אז $\delta^*(q_0,\varepsilon)=q_0$ ואכן הסכום הוא 0.
2. נניח עבור $w_1...w_{n-1}$ ונוכיח עבור $w_1...w_n$. לפי ההגדה מתקיים $\delta^*(q_0,w_1..w_n)=\delta(\delta^*(w_1...w_{n-1}),w_n)=\delta(q_j,w_n)$ כאשר $j=\left(\sum_{i=1}^{n-1}w_i\right)\bmod 3$.
   כעת, אם $w_n=0$ אז הסכום לא משתנה וכן $\delta(q_j,w_n)=q_j$. ואם $w_n=1$ אז $\delta(q_j,w_n)=q_{(j+1)\bmod 3}$ אבל גם הסכום גדל ב-1 ולכן התווסף 1 לשארית.

הראינו שמתקיימת הטענה. בפרט מתקיים ש- $\sum_{i=1}^{n}w_i\equiv 0(\bmod 3)$ אמ"מ $\delta^*(q_0,w)=q_0$ ולכן $L(A)=L$.

הגדרה: שפה $L\subset\Sigma^*$ נקראת **רגולרית** אם קיים אס"ד M כך ש- $L(M)=L$. נסמן את מחלקת השפות הרגולריות ב-DFA.

הגדרה: תהיינה $L_1,L_2\subset\Sigma^*$ שפות כלשהן מעל Σ. נגדיר את הפעולות הבאות על השפות:

- **איחוד**: $L_1\cup L_2=\{w:w\in L_1\vee w\in L_2\}$
- **חיתוך**: $L_1\cap L_2=\{w:w\in L_1\wedge w\in L_2\}$
- **משלים**: $\overline{L_1}=\{w:w\notin L_1\}$
- **כוכב**: $L_1^*=\{w_1...w_k:0\le k\wedge\forall 1\le i\le k\ w_i\in L_1\}$

משפט: תהיינה L_1,L_2 שפות רגולריות. אזי $L_1\cup L_2$ רגולרית.

הוכחה: L_1,L_2 רגולריות. נניח בה"כ ש- L_1,L_2 שתיהן מעל אותו א"ב. אחרת נסתכל אל איחוד הא"בים שלהן. יהיו A_1,A_2 אס"דים שמקבלים אותן בהתאמה. נניח ש- $A_i=\langle Q_i,\Sigma,\delta_i,s_i,F_i\rangle$.

נגדיר אס"ד חדש $A=\langle Q,\Sigma,\delta,s_0,F\rangle$ כאשר:

- $Q=Q_1\times Q_2$
- $s_0=(s_1,s_2)$
- $\delta((q_1,q_2),\sigma)=(\delta_1(q_1,\sigma),\delta_2(q_2,\sigma))$
- $F=F_1\times Q_2\cup Q_1\times F_2$

הרעיון הוא ש-A למעשה רץ על A_1 ועל A_2 באותו שמן ומקבל אם אחד מהם הגיע למצב מקבל.

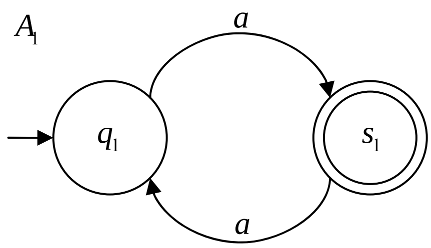
נוכיח ש-$L(A)=L(A_1)\cup L(A_2)$:

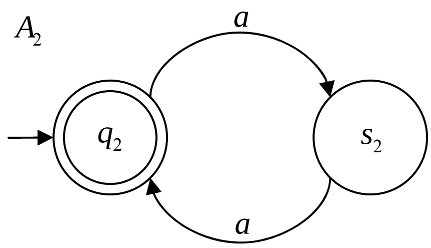
($\subset$) נניח ש-$w=w_1...w_n\in L(A)$. אזי קיימת ריצה מקבלת $r=r_0r_1...r_n$ של A על w. נסמן $r_i=(q_1^i,q_2^i)$. מהגדרת A נובע ש-$t_i=q_i^0q_i^1...q_i^n$ היא הריצה של A_i על w. מאחר ש-$r_n\in F_1\times Q_2\cup Q_1\times F_2$, נובע ש-$q_1^n\in F_1$ או $q_2^n\in F_2$, כלומר $w\in L(A_1)\cup L(A_2)$.

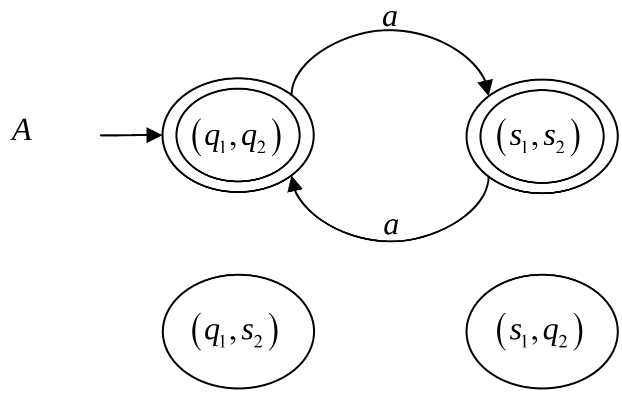
($\supset$) נניח ש-$w\in L(A_1)\cup L(A_2)$. בה"כ $w\in(A_1)$. תהי $t_i=q_i^0q_i^1...q_i^n$ הריצה של A_i על w. אזי לפי ההגדרה $r=r_0r_1...r_n$ היא הריצה של A על w כאשר $r_i=(q_1^i,q_2^i)$. אבל הנחנו ש-$w\in L(A_1)$. לכן $q_1^n\in F_1$. לכן $r_n\in F$, כלומר $w\in L(A)$.

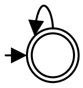
☺

דוגמה: נגדיר את השפות הבאות: $L_1=\{w:|w| \text{ is even}\},L_2=\{w:|w| \text{ is odd}\}$ מעל $\{a\}$.



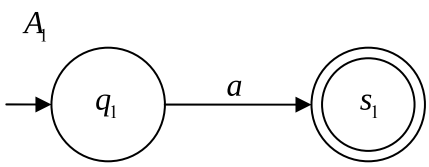


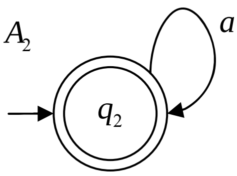


מאחר ש- $L_1 \cup L_2 = \Sigma^*$ היה מספיק מצב אחד  בשביל לתאר את השפה האס"ד שבנינו בהוכחה נקרא אוטומט המכפלה. אז אנחנו רואים שאוטומט המכפלה יכול להיות בזבזני. כמו כן, ניתן לראות כאן שקיבלנו מצבים לא ישיגים. אבל, תמיד ניתן לבנות את אוטומט המכפלה וזאת העוצמה של המשפט.

הערה: אמנם בהגדרה של אס"ד δ היא פונקצייה שלמה, אך בפועל פעמים רבות כשמציירים אס"ד לא מציירים את כל המעברים של פונקציית המעברים. אם יש אות σ ומצב q שעבורם $\delta(q,\sigma)$ לא מוגדרת, הכוונה היא שאם האס"ד נמצא במצב q והאות σ מגיעה מהקלט אז האס"ד "נתקע" ודוחה את המילה. ברור שמצב זה ניתן לתיאור ע"י פונקציית מעברים מלאה. פשוט נוסיף בור דוחה ואליו נעביר את כל הקלטים שאמורים "להיתקע". בהקשר זה, חשוב לציין שכאשר בונים את אוטומט המכפלה חשוב להשתמש בהגדרה המלאה של פונקציית המעברים.

דוגמה: $\Sigma = \{a\}, L_1 = \{a\}, L_2 = \Sigma^*$





מה קורה כאשר בונים את אוטומט המכפלה ולא משתמשים בהגדרה המלאה של δ_1?

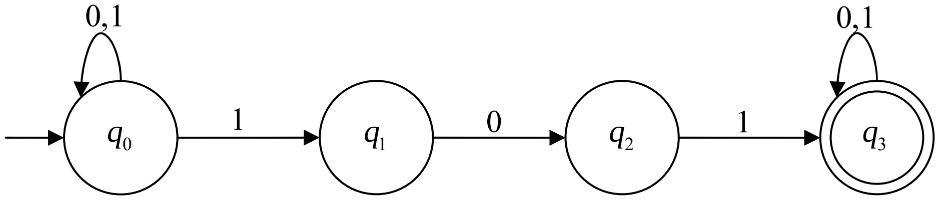
משפט: תהיינה L_1, L_2 שפות רגולריות. אזי $L_1 \cap L_2$ רגולרית.

הוכחה: באופן דומה להוכחה של המשפט הקודם, אלא שבאוטומט המכפלה $F = F_1 \cap F_2$.

☺

אי דטרמיניזם

דוגמה: נסתכל על מכונת המצבים הבאה עבור השפה $L = \{w : w \text{ contains the sequence } 101\}$

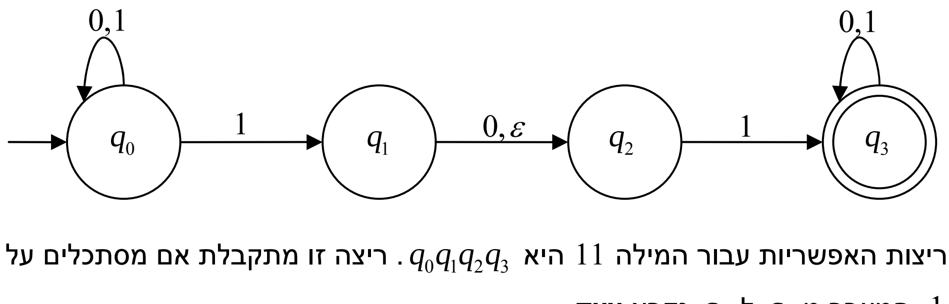


במצב q_0 כאשר מגיעה מהקלט האות 1 לא ברור האם צריך להישאר ב-q_0 או לעבור ל-q_1. למשל, עבור המילה 101 יכולות להיות שתי ריצות: $q_0q_0q_0q_1$ ו-$q_0q_1q_2q_3$. הריצה הראשונה דוחה ואילו השנייה מקבלת. בדיקה קצרה תראה שאם $w \notin L$ לא קיימת ריצה מקבלת עבורה.

הגדרות:

1. **אוטומט סופי לא דטרמיניסטי** (להלן אס"ל) הוא חמישייה $A = \langle Q, \Sigma, \delta, Q_0, F \rangle$ כאשר:
   - Q קבוצה סופית של מצבים
   - Σ א"ב סופי
   - $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ פונקציית מעברים לא דטרמיניסטית
   - $Q_0 \subset Q$ קבוצת מצבים התחלתיים
   - $F \subset Q$ קבוצת מצבים מקבלים
2. **ריצה** של אס"ל A על $w = w_1 ... w_n \in \Sigma^*$ היא סדרת מצבים $r = r_0 r_1 ... r_m$ כך ש-
   - ניתן לכתוב את w כ-$w = y_1 ... y_m$ כאשר $y_i \in \Sigma \cup \{\varepsilon\}$ (ניתן לרפד את w ע"י ε-ים)
   - $r_0 \in Q_0$ (כלומר מתחילים את הריצה ממצב התחלתי)
   - $r_{i+1} \in \delta(r_i, y_{i+1})$ (הריצה מתקדמת לפי פונקציית המעברים)
3. ריצה r של אס"ל A על $w \in \Sigma^*$ היא **מקבלת** אם $r_n \in F$. אחרת היא **דוחה**.
4. אס"ל A **מקבל** מילה w אם קיימת ריצה מקבלת של A על w. אחרת הוא **דוחה** אותה.
5. **שפה של אס"ל** A היא כל המילים ש-A מקבל - $L(A) = \{w : A \text{ accepts } w\}$.
6. נסמן את מחלקת השפות שמתקבלות ע"י אס"ל ב-NFA.

דוגמה: המשמעות של האות ε בקלט הוא שניתן לעבור למצב הבא ללא קריאה של אות מהמילה. למשל, נסתכל על האס"ל הבא:



אחת הריצות האפשריות עבור המילה 11 היא $q_0q_1q_2q_3$. ריצה זו מתקבלת אם מסתכלים על 11 כעל $1\varepsilon1$. המעבר מ-q_1 ל-q_2 נקרא **צעד** ε.

משפט: לכל אס"ל A קיים אס"ל A' ללא צעדי ε כך ש-$L(A)=L(A')$. כמו כן את A' ניתן לחשב בזמן פולינומיאלי.

הוכחה: הרעיון הוא שניתן לקשר באופן ישיר כל מצב למצב שניתן להגיע אליו ע"י סדרה של מעברי ε.

יהי $A=\langle Q,\Sigma,\delta,Q_0,F\rangle$ אס"ל. עבור מצב $q\in Q$ נגדיר את $E(q)$ להיות קבוצת המצבים q' כך שניתן להגיע מ-q ל-q' ע"י סדרת מעברי אפסילון. עבור קבוצה מצבים $S\subset Q$ נגדיר $E(S)=\bigcup_{q\in S}E(q)$.

בהינתן מצב q קל למצוא את $E(q)$ בזמן פולינומיאלית ע"י האלגוריתם הבא:

1. $S_0 \leftarrow \{q\}$
2. $i \leftarrow 0$
3. repeat
   3.1. $S_{i+1} \leftarrow S_i \cup \{q' : \exists q'' \in S_i \left(q' \in \delta(q'',\varepsilon)\right)\}$
   3.2. $i \leftarrow i+1$
   until $S_i = S_{i-1}$
4. return S_i

ברור שהאלגוריתם עוצר בזמן פולינומיאלית, שהרי לכל $0<i$ מתקיים $S_i\subset S_{i+1}$ ולכן ניתן לבצע את שלב (2.1) לכל היותר $|Q|$ פעמים. בכל איטרציה יש לעבור על כל Q ולכן לכל q נצטרך $O\left(|Q|^2\right)$ צעדים. בשביל לחשב את $E(q)$ לכל $q\in Q$ נצטרך סה"כ $O\left(|Q^3|\right)$ צעדים.

כמן כן, ברור קל להשתכנע שהאלגוריתם אכן מחזיר את $E(q)$ כפי שהגדרנו אותה למעלה.

כעת נוכל להגדיר את $A'=\langle Q',\Sigma',\delta',Q_0',F'\rangle$ באופן הבא:

- $Q'=Q$
- $\Sigma'=\Sigma$
- $\delta'(q,\sigma)=E\left(\delta(q,\sigma)\right)=\bigcup_{q'\in\delta(q,\sigma)}E(q')$
- $Q_0'=E(Q_0)=\bigcup_{q_0\in Q_0}E(q_0)$
- $F'=F$

כעת נראה ש- $L(A)=L(A')$:

($\subset$) נראה שעבור מילה $w_1...w_n$ אם יש ריצה של A עליה $r_0...r_t$ אז יש גם ריצה של A' עליה שמסתיימת ב- r_t. בפרט זה יהיה נכון עבור כל המילים שישי עבורן ריצה כך ש- $r_t\in F$ ואז נקבל ש- $L(A)\subset L(A')$. נוכיח את הטענה באינדוקציה על מספר רצפי ה- ε בריצה $r_0...r_t$.

1. אם אין בריצה מעברי ε אז $r_0...r_t$ היא גם ריצה ב- A' והטענה ברורה.
2. נניח עבור k רצפי ε ונוכיח עבור $k+1$. נסמן ב- m את האינדקס שלפני רצף ה- ε האחרון וב- l את האינדקס האחרון של רצף ה- ε האחרון. אז נסתכל בריצה $r_0...r_m r_{m+1}...r_l r_{l+1}...r_t$. לפי הנחת האינדוקציה יש ריצה ב- A' שמגיעה ל- r_m. כעת לפי הגדרת δ' יש ב- A' מעבר מ- r_m ל- r_l. המעברים שאח"כ הם אינם מעברי ε ולכן ב- A' ניתן להגיע מ- r_l ל- r_t.

($\supset$) נראה שבהינתן $w=w_1...w_n$ וריצה $r_0r_1...r_n$ של A' קיימת ריצה $x_1...x_t$ של A על w כך ש- $x_t=r_n$. נוכיח באינדוקציה על אורך המילה:

1. אם $w=\varepsilon$ אז $r_0\in Q_0'$, כלומר $r_0\in E(Q_0)$. לכן קיימים $q_0\in Q_0$ וסדרת מעברי ε $q_0x_1...x_t$ כך ש- $x_t=r_0$ וזו ריצה של A על ε שמסתיימת ב- r_0.
2. נניח שהטענה נכונה עבור $w_1...w_n$ ונוכיח עבור $w=w_1...w_nw_{n+1}$. נניח שיש ריצה $r_0r_1...r_nr_{n+1}$ של A' על w. ברור ש- $r_0...r_n$ היא ריצה של A' על $w_1...w_n$. לכן לפי הנחת האינדוקציה קיימת ריצה של A על $w_1...w_n$ שמסתיימת ב- r_n. נסמן ריצה זו ב- $x_0x_1...x_t$. ידוע ש- $r_{n+1}\in\delta'(r_n,w_{n+1})=\delta'(x_t,w_{n+1})$. לכן לפי ההגדרה $r_{n+1}\in E(\delta(r_n,w_{n+1}))$. אזי קיימים $y_1\in\delta(r_n,w_{n+1})$ וסדרת מעברי ε $y_1...y_k$ כאשר $y_k=r_{n+1}$. לכן $x_0...x_ty_1...y_k$ ריצה של A על w שמסתיימת ב- r_{n+1}.

☺

משפט: לכל אס"ל A קיים אס"ד A' כך ש- $L(A)=L(A')$.

הוכחה: יהי אס"ל $A=\langle Q,\Sigma,\delta,Q_0,F\rangle$. בה"כ ל- A אין צעדי ε, כלומר $\delta:Q\times\Sigma\to 2^Q$. נגדיר אס"ד $A'=\langle Q',\Sigma,\delta',q_0',F'\rangle$ כאשר:

- $Q'=2^Q$ כל הקבוצות החלקיות של המצבים.
- $\delta'(S,\sigma)=\bigcup_{s\in S}\delta(s,\sigma)$. למעשה זוהי קבוצת המצבים שניתן לעבור אליהם בקריאת האות σ מאחד מהמצבים שב- S.
- $q_0'=Q_0$

- $F'=\{S : S\cap F\neq\varnothing\}$

הרעיון הוא שהאס"ד A' ייקח בחשבון בריצה שלו את כל המצבים ש-A יכול היה להיות בהם. ואם בסוף הריצה יש בקבוצת המצבים לפחות מצב מקבל אחד, סימן של-A הייתה יכולה להיות ריצה מקבלת על המילה ולכן A' מקבל.

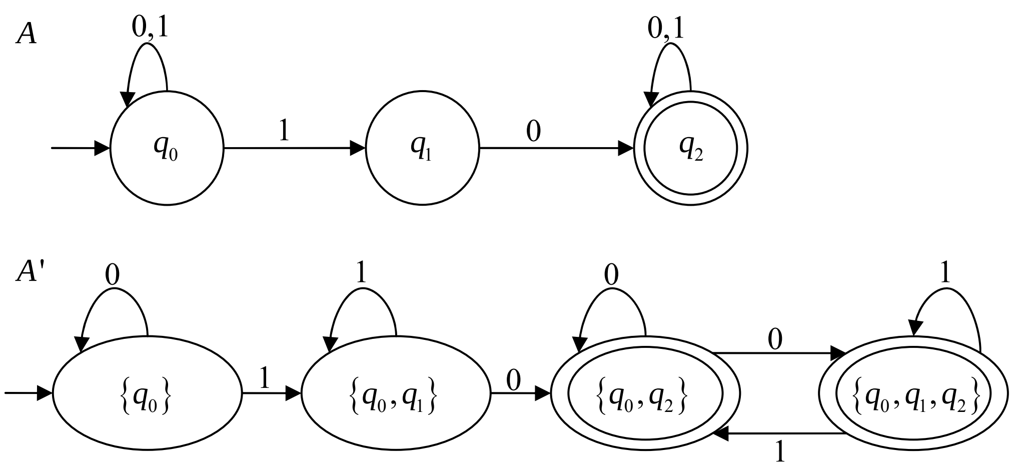
נניח ש-$w=w_1..w_n\in L(A)$. אזי קיימת ריצה מקבלת $q_0q_1...q_n$ כאשר $q_0\in Q_0, q_n\in F$ ולכל $0\le i<n$ $q_{i+1}\in\delta(q_i,w_{i+1})$. אזי הריצה של A' על w תהיה $r_0r_1...r_n$ כאשר $r_0=q_0'$ ולכל $0\le i<n$ $r_{i+1}=\bigcup_{s\in r_i}\delta(s,w_{i+1})$. לכל i מתקיים $q_i\in r_i$. לכן בפרט $q_n\in r_n$ ו-$r_n\in F'$, כלומר $w\in L(A')$.

מצד שני, אם $w\in L(A')$ אז הריצה $r_0r_1...r_n$ של A' על w היא מקבלת. כלומר $r_0=q_0'$, $r_n\in F'$ ולכל $0\le i<n$ מתקיים $r_{i+1}=\delta'(r_i,w_{i+1})=\bigcup_{s\in r_i}\delta(s,w_{i+1})$. מתוך אלה נוכל לבחור סדרה $q_0q_1...q_n$ כך ש-$q_i\in r_i$, $q_n\in F$ ו-$q_{i+1}\in\delta(q_i,w_{i+1})$ לכל $0\le i<n$. זאת היא ריצה מקבלת של A על w. לכן $w\in L(A)$.

☺

הערה: המעבר מאס"ל לאס"ד שקול מספר המצבים יכול לגדול מעריכית!!

דוגמה:



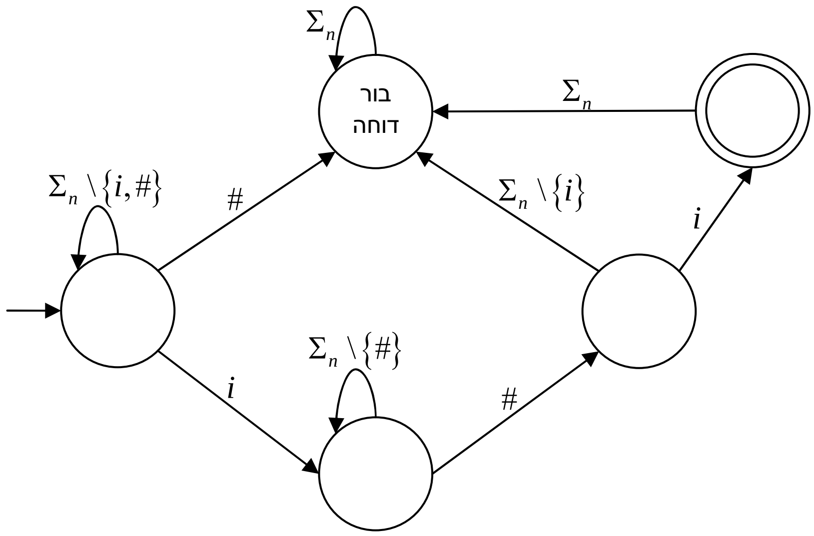
התהליך שעושים כאן נקרא **חירצון**.

משפט: יש משפחה של שפות $\{L_n\}$ כך שלכל $1\le n$ מתקיים:

1. L_n ניתנת לזיהוי ע"י אס"ל עם $O(n)$ מצבים
2. האס"ד הקטן ביותר עבור L_n צריך לפחות 2^n מצבים.

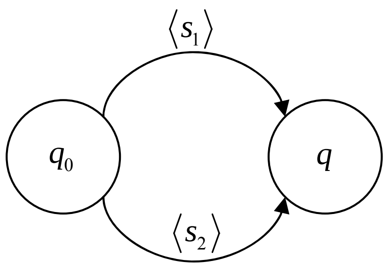
הוכחה: נגדיר $L_n=\{\sigma_1...\sigma_k\#\sigma_{k+1}:\forall 1\le i\le k\ \sigma_i\in\{1,...,n\}\wedge\sigma_{k+1}\in\{\sigma_1,...,\sigma_k\}\}$ מעל הא"ב $\Sigma_n=\{1,...,n,\#\}$.

ראשית, נראה אס"ל שמקבל את L_n עם $O(n)$ מצבים. האס"ל יפעל באופן הבא: הוא ינחש את האות האחרונה (יש n אפשרויות כאלה ולכל אחת מצב התחלתי אחד) ויפעיל את המכונה הבאה:



סה"כ באס"ל יש $3n+2$ מצבים - 3 מצבים לכל אות ועוד בור דוחה ובור מקבל. קל לראות שאכן השפה מתקבלת ע"י האס"ל הזה. אם $\sigma_1...i...\sigma_k\#i\in L_n$ אז האס"ל יבחר להתחיל את הריצה במצב ההתחלתי שמתאים לניחוש ש- i היא האות האחרונה. ואז קל לראות שהמילה אכן תתקבל ברכיב זה. ואם $w\notin L_n$ ברור שלא משנה באיזה רכיב ינחש האס"ל להתחיל לרוץ, הריצה על המילה תהיה דוחה.

כעת נראה שבאס"ד שמקבל את השפה חייבים להיות לפחות 2^n מצבים. נניח בשלילה שיש אס"ד A_n עם פחות מ- 2^n מצבים כך ש- $L(A_n)=L_n$. ל- $\{1,...,n\}$ יש 2^n קבוצות חלקיות, אבל יש פחות מ- 2^n מצבים ולכן מעיקרון שובך היונים קיימות שתי קבוצות $S_1,S_2\subset\{1,...,n\}$ כך שהמילים שמורכבות מאותיות בהן מגיעות לאותו מצב q בריצה. תהיינה $\langle s_i\rangle\in S_i$ המילים שמורכבות מאותיות S_i בסדר לקסיקוגרפי.



תהי בה"כ $\sigma \in S_1 \setminus S_2$. אזי $\langle s_1 \rangle \# \sigma \in L_n$ אבל $\langle s_2 \rangle \# \sigma \notin L_n$. נתבונן ב-$q' = \delta\left(\delta\left(q,\#\right),\sigma\right)$.

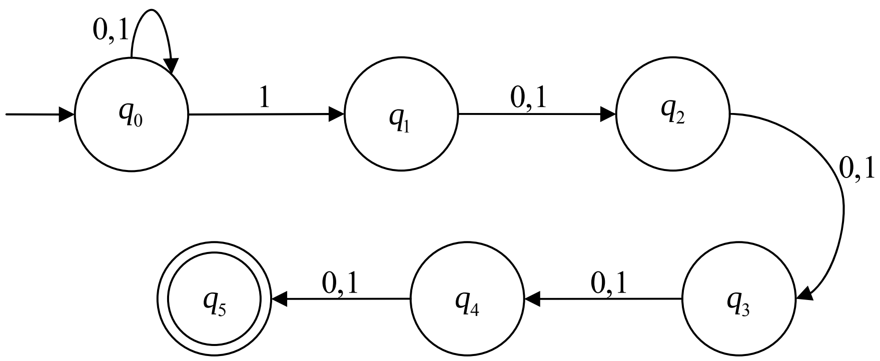
- אם $q' \in F$ נקבל ש-$\langle s_2 \rangle \# \sigma \in L_n$ אבל $\langle s_2 \rangle \# \sigma \notin L_n$.
- אם $q' \notin F$ נקבל שהריצה היחידה של $\langle s_1 \rangle \# \sigma$ דוחה וזה בסתירה לכך ש-$\langle s_1 \rangle \# \sigma \in L_n$.

בכל מקרה קיבלנו סתירה ולכן לא יכול להיות שב-A_n יש פחות מ-2^n מצבים. ☺

מסקנה: לניפוח המעריכי של החירצון יש חסם תחתון, כלומר לא יכולנו למצוא בנייה פולינומיאלית של אס"ד שקול. שהרי, אם יש בניה פולינומיאלית, אז נוכל להפעיל אותה על המשפחה $\{L_n\}$ ומובטח שקיים $0 < n$ שעבורו בבנייה פולינומיאלית לא יהיו מספיק מצבים.

דוגמה: $L = (0+1)^* 1(0+1)(0+1)(0+1)(0+1)$ של כל המילים שמופיע בהן 1 במקום החמישי מהסוף.

נתאר אס"ל A שמקבל את L:



קל לראות שאכן $L(A) = L$. אם $w_1 \ldots w_{n-5} w_{n-4} w_{n-2} w_{n-1} w_n \in L$ אז $w_{n-5} = 1$ ואז הריצה $\underbrace{q_0 \ldots q_0}_{n-4} q_1 q_2 q_3 q_4 q_5$ היא ריצה מקבלת של A. מצד שני אם $w \notin L$ אז $w_{n-5} = 0$ ואז לא משנה מתי בריצה האס"ל ינחש לעבור ל-q_1, הריצה לא תיגמר ב-q_5.

נטען שבאס"ד שמקבל את L יש לפחות 32 מצבים. נניח ש-A אס"ד כך ש-$L(A) = L$. לכל מילה $w = w_1 \ldots w_5$ נסמן $q_w = \delta^*\left(q_0, w\right)$. נטען שאם u, v מילים שונות בנות חמש אותיות אז $q_v \neq u_w$.

נניח בשלילה ש- $q_v \neq u_w$. נניח שהן שונות באות ה- i. בה"כ האות ה- i של v היא 0 ואילו האות ה- i של u היא 1. נסתכל על המילים $v0^{i-1}$ ו- $u0^{i-1}$. ברור שהריצה של A על שתי המילים האלה נגמרת באותו מצב שהרי A דטרמיניסטי. אבל $u0^{i-1} \in L$ בעוד ש- $v0^{i-1} \notin L$.

אם כך, לכל שתי מילים שונות בנות חמש אותיות יש ריצה שונה ב- A. אבל יש $2^5 = 32$ מילים כאלה. לכן יש לפחות 32 מצבים שונים ב- A!

ביטויים רגולריים

הגדרות:

1. נגדיר **ביטוי רגולרי** על א"ב Σ באופן אינדוקטיבי:
   - $a \in \Sigma$, ε ו- $\varnothing$ הם ביטויים רגולריים
   - אם r_1, r_2 ביטויים רגולריים אז גם הבאים הם ביטויים רגולריים:
     - i. $r_1 + r_2$
     - ii. $r_1 \cdot r_2$
     - iii. r_1^*
2. כל ביטוי רגולרי r מגדיר שפה $L(r) \subset \Sigma^*$ באופן הבא:
   - $L(a) = \{a\}$, $L(\varepsilon) = \{\varepsilon\}$ ו- $L(\varnothing) = \varnothing$
   - אם r_1, r_2 ביטויים רגולריים אז:
     - i. $L(r_1 + r_2) = L(r_1) \cup L(r_2)$
     - ii. $L(r_1 \cdot r_2) = L(r_1) \cdot L(r_2)$
     - iii. $L(r_1^*) = (L(r_1))^*$
3. נסמן בקיצור $r^+ = r \cdot r^*$
4. נסמן ב- REG את מחלקת השפות שמתקבלות ע"י ביטויים רגולריים.

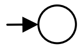
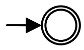
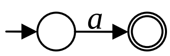
דוגמאות: נסתכל על $\Sigma = \{0,1\}$

1. שפת כל המילים שבהם מופיע 1 שני מקומות לפני הסוף מוגדרת ע"י הביטוי הרגולרי $(0+1)^* 1 (0+1)(0+1)$
2. מילים שמופיע בהן 1 אחד בדיוק מוגדרות ע"י $0^* 1 0^*$
3. מילים שמופיע בהן 1 אחד לפחות מוגדרות ע"י $(0+1)^* 1 (0+1)^*$
4. מילים שהתו האחרון בהן הוא 0 הן $(0+1)^* 0$

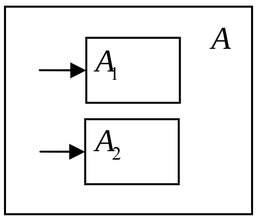
טענה: $REG = NFA$

הוכחה: נוכיח הכלה בשני הכיוונים.

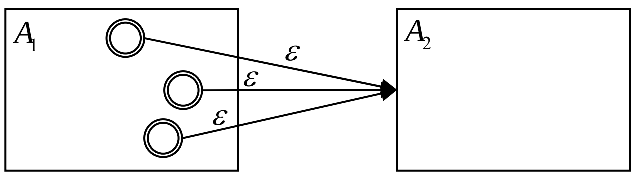
($\subset$) יהי r ביטוי רגולרי. נראה שקיים אס"ל A כך ש-$L(r)=L(A)$. נוכיח באינדוקציה על הבנייה של r:

1. עבור ביטויים רגולריים בסיסיים:
   - אם $r=\varnothing$ $L(r)=\varnothing$ מתקבלת ע"י האס"ל 
   - אם $r=\varepsilon$ אז $L(r)=\{\varepsilon\}$ מתקבלת ע"י האס"ל 
   - אם $r=a$ עבור $a\in\Sigma$ אז $L(r)=\{a\}$ מתקבלת ע"י האס"ל 

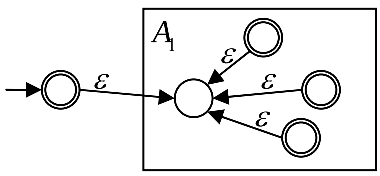
2. נניח שהטענה נכונה עבור r_1,r_2 ו-A_1,A_2 אס"לים עבור $L(r_1),L(r_2)$ בהתאמה. ניתן להניח בה"כ של-A_1 ול-A_2 מצב התחלתי יחיד, שהרי ניתן להוסיף מצב חדש שיתפקד כמצב התחלתי וממנו יצאו מעברי ε לכל המצבים ההתחלתיים המקוריים.
   - אם $r=r_1+r_2$ אז $L(r)$ מתקבלת פשוט ע"י צירוף A_1 ו-A_2 לאס"ל אחד A שיש בו שני מצבים התחלתיים.



   - אם $r=r_1\cdot r_2$ אז $L(r)$ מתקבלת ע"י האס"ל A שמתקבל מהוספת מעברי ε מהמצבים המקבלים של A_1 למצב ההתחלתי של A_2 והפיכת המצבים המקבלים של A_1 ללא מקבלים.



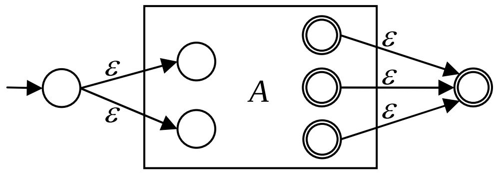
   - אם $r=r_1^*$ אז $L(r)$ מתקבלת ע"י האס"ל המוגדר כך:



($\supset$) נגדיר אוטומט לא דטרמיניסטי מוכלל (להלן אסל"מ) שהוא בדיוק כמו אס"ל אלא שעל המעברים שלו רשומים ביטויים רגולריים ולא אותיות. יש לו מצב התחלתי יחיד ללא קשתות נכנסות ומצב מקבל יחיד ללא קשתות יוצאות. מלבד מצבים אלו בין כל שני מצבים יש מעבר. אם האוטומט נמצא במצב q ונשארה המילה w, אז האוטומט יכול לעבור למצב q' עם ביטוי רגולרי r אמ"מ יש רישא של w ששייכת ל-$L(r)$. אם $r=\varepsilon$ ניתן לעבור תמיד ואם $r=\varnothing$ לא ניתן לעבור אף פעם.

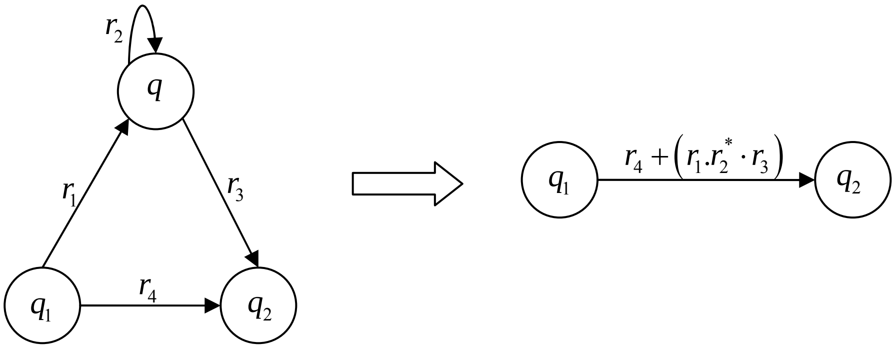
נסמן אסל"מ עם k מצבים ע"י $GNFA_k$.

נראה שלכל אס"ל עם k מצבים קיים אסל"מ שקול עם $k+2$ מצבים. זה די ברור, כי הבינתן אס"ל A נוכל להפוך אותו לאסל"מ באופן הבא:



לכל שני מצבים שאין ביניהם מעבר נוסיף קשת עם $\varnothing$. ברור שקיבלנו אסל"מ שקול.

כעת נראה שלכל $2<k$ ולכל $GNFA_k$ קיים $GNFA_{k-1}$ שקול. יהי q מצב כלשהו שאינו התחלתי ואינו מקבל. לכל זוג מצבים q_1,q_2 נבצע את ההחלפה הבא:



אחרי שנעשה זאת לכל q_1,q_2 כנ"ל נזרוק את q מהאסל"מ.

נראה שהבנייה שעשינו מקיימת ש- $L\left(GNFA_k\right)=L\left(GNFA_{k-1}\right)$:

($\subset$) אם $w\in L\left(GNFA_k\right)$ תהי $s=s_0s_1...s_n$ ריצה מקבלת. אם s לא עוברת ב- q אז s היא גם ריצה של $GNFA_{k-1}$ על w. אחרת נניח ש- $s=s_0...s_iq^+s_j...s_n$. אזי תת המילה שנקראת בין s_i ל- s_j שייכת לשפה $r_1\cdot r_2^*\cdot r_3$ (לפי הסימונים באיור). לכן בריצה של $GNFA_{k-1}$ ניתן לקרוא את אותה התת מילה במעבר מ- s_i ל- s_j.

($\supset$) אם $w\in L\left(GNFA_{k-1}\right)$ תהי $s=s_0s_1...s_n$ ריצה מקבלת. נסתכל על s_i,s_{i+1}. נניח שהמעבר ביניהם נעשה ע"י r_4. אם המעבר היה קיים גם ב- $GNFA_k$ אז אין בעיה. אחרת המעבר נעשה ע"י $r_1\cdot r_2^*\cdot r_3$ ואז ב- $GFNA_k$ ניתן לעשות את המעבר הזה דרך המצב q שהוצאנו בבנייה. לכן גם ב- $GNFA_k$ תהיה ריצה מקבלת.

כעת, באינדוקציה נוכל להפוך את $GNFA_k$ ל- $GNFA_2$. אבל $GNFA_2$ הוא מהצורה 

והשפה שלו היא $L\left(r\right)$, כלומר היא ב- REG.

☺

שפות לא רגולריות

טענה: השפה $L=\{0^n1^n : n\geq 0\}$ אינה רגולרית

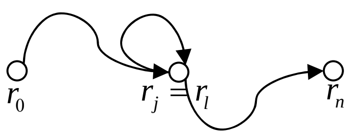
הוכחה: נניח בשלילה שקיים אס"ד A כך ש- $L(A)=L$. נניח שיש ל- A k מצבים. נסתכל במילה $0^k1^k \in L$. תהי $r=r_0r_1...r_kr_{k+1}...r_{2k}$ ריצה מקבלת של A על 0^k1^k. בגלל שהמילה היא באורך $2k$ ויש לאס"ד רק k מצבים אז לפי עיקרון שובך היונים חייב להיות בריצה מעגל. למעשה המעגל נמצע כבר בחלק $r_0r_1...r_k$ שמופיעים בו $k+1$ מצבים. לכן קיימים $0\leq i<j\leq k$ כך ש- $r_i=r_j$. נובע ש- A מקבל גם את המילה $0^{k+(j-i)}1^k$ אשר אינה בשפה, בסתירה לכך ש- $L(A)=L$.

☺

למת הניפוח: אם L רגולרית אז קיים $1\leq p$ (שנקרא לו **קבוע הניפוח**) כך שלכל $w\in\Sigma^*$ המקיימת $|w|\geq p$ קיימת חלוקה $w=xyz$ כך ש-

1. לכל $0\leq i$ $xy^iz\in L$
2. $0<|y|$
3. $|xy|\leq p$

הוכחה: יהי $A=\langle Q,\Sigma,q_0,\delta,F\rangle$ אס"ד כך ש- $L(A)=L$. נסמן $p=|Q|$. תהי $w=w_1w_2...w_n$ עבור $p\leq n$. תהי $r=r_0r_1...r_n$ הריצה של A על w. בקבוצה $\{r_0,...,r_p\}$ יש $p+1$ מצבים ולכן לפי עיקרון שובך היונים קיימים $0\leq j<l\leq p$ כך ש- $r_j=r_l$. אז נחלק את w באופן הבא: $w=xyz$ כאשר $x=w_1...w_j, y=w_{j+1}...w_l, z=w_{l+1}...w_n$. נראה שמתקיימים התנאים הדרושים:



1. החלוקה של המילה משרה חלוקה של הריצה לשלושה חלקים. לכל $0\leq i$ הריצה $r_0...r_j\left(r_{j+1}...r_l\right)^i r_{l+1}...r_n$ היא ריצה מקבלת של A על xy^iz, שהרי $r_j=r_l$. לכן, כאשר נקרא את $w_{j+1}...w_l$ שוב ושוב בכל פעם נחזור ל- r_{j+1} ונעבור בדיוק על אותם מצבים.
2. $y\neq\varepsilon$ שהרי $j<l$
3. $|xy|=l\leq p$

☺

מסקנה: תהי L שפה כך שלכל p קיימת מילה באורך p לפחות שלא ניתן לנפחה כמו בלמת הניפוח. אזי L אינה רגולרית.

טענה: $L=\{0^n1^n : n\geq 0\}$ אינה רגולרית.

הוכחה: בהינתן p נסתכל על המילה 0^p1^p. אם ננסה לחלק $0^p1^p = xyz$ כך ש- $|xy|\leq p$ ו- $y\neq\varepsilon$ בהכרח $x=0^*, y=00^*$. אז אם נסתכל על xy^2z נקבל שמספר האפסים שם גדול ממספר האחדות, כלומר $xy^2z\notin L$.

☺

טענה: $L=\{w : \text{the number of 0s equals the number of 1s}\}$ אינה רגולרית.

הוכחה א: בהינתן p נסתכל על המילה 0^p1^p. כמו בטענה הקודמת לכל חלוקה $0^p1^p = xyz$ נקבל ש- $x=0^*\ y=00^*$. לכן ב- xy^2z מספר האפסים גדול ממספר האחדות.

☺

הוכחה ב: נשים לב שמתקיים $\{0^n1^n : n\geq 0\} = L\cap L\left(0^*1^*\right)$. השפות הרגולריות סגורות לחיתוך. לכן אילו L הייתה רגולרית גם $\{0^n1^n : n\geq 0\}$ הייתה רגולרית, בסתירה לטענה הקודמת.

☺

טענה: השפה $L=\{ww : w\in\Sigma^*\}$ אינה רגולרית.

הוכחה: בהינתן p נסתכל על המילה $0^p10^p1\in L$. לכל חלוקה $0^p10^p1=xyz$ כך ש- $y\notin\varepsilon$ ו- $|xy|\leq p$ מתקיים ש- $xy^0z=xz=0^{p-|y|}10^p1$ והיא אינה מהצורה ww. כלומר, את 0^p10^p1 אי אפשר לנפח. לכן L אינה רגולרית.

☺

טענה: השפה $L=\left\{1^{n^2} : 0\leq n\right\}$ אינה רגולרית.

הוכחה: נניח בשלילה ש- L רגולרית וקבוע הניפוח שלה הוא p. נסתכל במילה $w=1^{p^2}\in L$. נניח ש- $w=xyz$ חלוקה של w כך ש- $x=1^a, y=1^b, z=1^c$. נסתכל על $xy^2z=1^{a+2b+c}$. ידוע ש- $p^2=a+b+c$ ו- $a+b\leq p$. כמו כן $0<b$ ולכל $0\leq i$ $a+ib+c=k^2$ עבור k כלשהו. אזי $p^2=a+b+c<a+2b+c$. אבל אז נקבל ש-

$$\left(p+1\right)^2 = p^2+2p+1\geq a+b+c+2a+2b+1\geq a+b+c+a+b+1>a+2b+c$$

ולכן $p^2 < a+2b+c < (p+1)^2$. בפרט לא קיים k כך ש- $k^2+a+2b+c$.

☺

טענה: $L=\{m+n=k : m,n,k \text{ represent binary numbers and equality holds}\}$ מעל הא"ב $\Sigma=\{0,1,+,=\}$ אינה רגולרית.

הוכחה: בהינתן p נסתכל על המילה $w=1^p+0=1^p$. בכל חלוקה $w=xyz$ כך ש- $|xy|\le p$ xy נמצאים בחלק 1^p ואז כמובן לא ניתן לנפח את y בשום צורה. משום ש- $1^q+0\ne 1^p$ לכל $q\ne p$.

☺

הגדרה: תהי שפה $L\subset\Sigma^*$. נגדיר יחס $\sim_L\subset\Sigma^*\times\Sigma^*$ באופן הבא: $x\sim_L y$ אמ"מ לכל $z\in\Sigma^*$ $xz\in L\leftrightarrow yz\in L$. אם קיים z שעבורו זה לא מתקיים הוא נקרא **זנב מפריד**.

דוגמה: $L=(0+1)^*0(0+1)$. אזי $111\sim_L 11$, $11\not\sim_L 10$ (כי 1 זנב מפריד) ו-$11\not\sim_L 01$ (כי ε זנב מפריד).

טענה: $\sim_L$ הוא יחס שקילות.

הוכחה: נראה שמתקיימות כל התכונות הדרושות:

1. רפלקסיביות: ברור ש- $x\sim_L x$ שהרי לכל $z\in\Sigma^*$ $xz\in L\leftrightarrow xz\in L$.
2. סימטריות: אם $x\sim_L y$ אז לכל $z\in\Sigma^*$ $xz\in L\leftrightarrow yz\in L$, כלומר $yz\in L\leftrightarrow xz\in L$ ולכן $y\sim_L x$.
3. טרנזיטיביות: אם $x\sim_L y$ וגם $y\sim_L w$ אז לכל $z\in\Sigma^*$ $xz\in L\leftrightarrow yz\in L$ וגם לכל $z\in\Sigma^*$ $yz\in L\leftrightarrow wz\in L$. כלומר לכל $z\in\Sigma^*$ $(xz\in L\leftrightarrow yz\in L)\wedge(yz\in L\leftrightarrow wz\in L)$, כלומר לכל $z\in\Sigma^*$ $xz\in L\leftrightarrow wz\in L$ ופירוש הדבר ש- $x\sim_L w$.

☺

משפט Myhill-Nerode: מספר מחלקות השקילות של $\sim_L$ הוא סופי אמ"מ L רגולרית.

הוכחה: תהי $L \subset \Sigma^*$ שפה.

($\Leftarrow$) נניח שמספר מחלקות השקילות של $\sim_L$ הוא סופי. נגדיר אס"ד $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ באופן הבא:

- Q - מחלקות השקילות של $\sim_L$
- $q_0 = [\varepsilon]$
- $\delta([x], a) = [xa]$
- $F = \{[w] : w \in L\}$

ראשית נטען ש-δ מוגדרת היטב. יש להראות שהיא אינה תלויה בבחירת הנציג. נניח ש-$[x] = [y]$. אזי לכל $a \in \Sigma$ מתקיים $xa \in L \leftrightarrow ya \in L$. נניח ש-$ya \not\sim_L xa$. אזי קיים זנה מפריד w כך שבה"כ $xaw \in L$ אבל $yaw \notin L$. אבל אז aw זנה מפריד בין x ל-y, בסתירה להנחה. לכן $[xa] = [ya]$, כלומר אכן δ אינה תלויה בבחירת הנציג.

קל לראות ש-$L(A) = L$. אם $w = w_1 \dots w_n \in L$ אז נסתכל על הריצה $r = r_0 r_1 \dots r_n$ של A על w. מתקיים $r_0 = [\varepsilon]$ ולכל $0 \le i < n$ מתקיים $r_{i+1} = [w_1 \dots w_i w_{i+1}]$, בפרט $r_n = [w] \in F$. מצד שני, אם $w \notin L$ אז $r_n = [w] \notin F$.

($\Rightarrow$) יהי $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ אס"ד כך ש-$L(A) = L$. נגדיר יחס שקילות $\sim_A$ על Σ^* ע"י $y \sim_A x$ אמ"מ $\delta^*(q_0, x) = \delta^*(q_0, y)$ (כלומר A מגיע לאותו מצב בקריאת x ובקריאת y). ברור ש-$\sim_A$ מוגדר היטב והוא אכן יחס שקילות מאחר ש-A דטרמיניסטי.

נראה שמספר מחלקות השקילות של $\sim_A$ הוא לכל הפחות מספר מחלקות השקילות של $\sim_L$ ומאחר שמספר מחלקות השקילות של $\sim_A$ חסום ע"י $|Q|$ נקבל את הטענה. נניח ש-$y \sim_A x$ ונראה ש-$y \sim_L x$. לכל $a \in \Sigma^*$ מתקיים

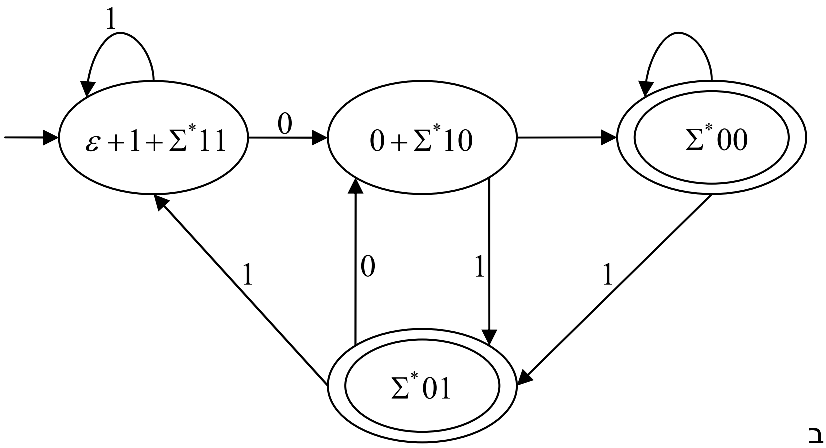
$\delta^*(q_0, xa) = \delta(\delta^*(q_0, x), a) = \delta(\delta^*(q_0, y), a) = \delta^*(q_0, ya)$. לכן $xa \in L \leftrightarrow ya \in L$, כלומר $y \sim_L x$. לכן מספר מחלקות השקילות של $\sim_A$ חסום מלמטה ע"י מספר מחלקות השקילות של $\sim_L$ וסיימנו.

☺

דוגמה: נסתכל על $L=(0+1)^*0(0+1)$. ניתן לראות ששייכות מילה לשפה תלויה אך ורק בשתי האותיות האחרונות שלה. לכן מחלקות השקילות של $\sim_L$ הן:

- $\varepsilon,1,\Sigma^*11$
- Σ^*01
- $0,\Sigma^*10$
- Σ^*00

נבנה אס"ד שמקבל את L לפי האלגוריתם שהוצג בהוכחה:



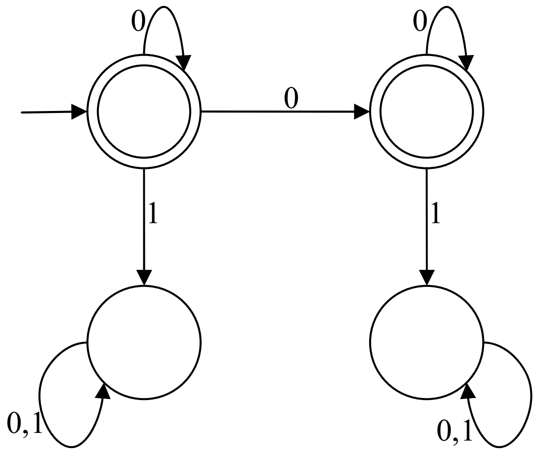
ב

זה נקרא **האוטומט הדטרמיניסטי הקנוני** של L. זה האוטומט בעל מספר המצבים המינימאלי שמקבל את L. מספר מחלקות השקילות של $\sim_L$ תלוי אך ורק ב-L ולא באוטומט שמקבל אותה. הוכחנו במשפט שאם A מקבל את L אזי מספר המצבים שלו הוא לכל הפחות מספר מחלקות השקילות של $\sim_L$. לכן אוטומט מינימאלי שמקבל את L הוא אוטומט שמספר המצבים שלו הוא כמספר מחלקות השקילות, כמו בדוגמה שלנו!

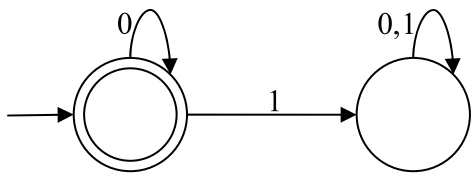
דוגמאות:

1. נוכיח ש-$L=\{0^n1^n:0\le n\}$ אינה רגולרית בעזרת משפט Myhill-Nerode. נשים לב שלכל $i\ne j$ מתקיים $0^i\not\sim_L 0^j$ משום ש-1^i הוא זנב מפריד. לכן יש ל-L אינסוף מחלקות שקילות ומכאן ש-L אינה רגולרית.
2. נוכיח ש-$L=\{ww:w\in\Sigma^*\}$ אינה רגולרית. נסתכל על המילים 0^i1 לכל $0<i$. עבור $i\ne j$ 0^i1 היא סיפא מפרידה בין 0^i1 ל-0^j1. מאחר שיש אינסוף i-ים יש אינסוף מחלקות שקילות ולכן השפה אינה רגולרית.
3. נסתכל על $L=\{0^i1^j:\gcd(i,j)=1\}$. לכל שני ראשוניים $p\ne q$ 1^p היא סיפא מפרידה בין 0^p ל-0^q. כיוון שיש אינסוף ראשוניים יש אינסוף מחלקות שקילות ולכן L אינה רגולרית.

דוגמה: נסתכל על האס"ד הבא:



השפה שלו היא 0^* אבל הוא לא מינימאלי. אוטומט מינימאלי עבור 0^* הוא:



הגדרות:

1. בהינתן אס"ד $A=\langle Q,\Sigma,\delta,q_0,F\rangle$ נגדיר על Q יחס שקילות $\sim_F$ באופן הבא:

   $p\sim_F q$ אמ"מ $\forall w\in\Sigma^*\left(\delta^*(p,w)\in F\leftrightarrow\delta^*(q,w)\in F\right)$

   ברור שזה יחס שקילות וכן אם שני מצבים הם שקולים ניתן לאחד אותם למצב אחד ולקבל אס"ד שקול.

2. לכל $0\le i$ נגדיר יחס שקילות $\sim_i$ על Q באופן הבא:

   $p\sim_i q$ אמ"מ $\forall w\in\Sigma^*\left(|w|\le i\rightarrow\left(\delta^*(p,w)\in F\leftrightarrow\delta^*(q,w)\in F\right)\right)$

   במילים, אחרות, $p\sim_i q$ אמ"מ אין סיפא מפרידה באורך לכל היותר i בין p ל- q. נשים לב ש- $\sim_F$ מתאים ל- $\sim_\infty$.

טענה: בהינתן אס"ד $A=\langle Q,\Sigma,\delta,q_0,F\rangle$ האלגוריתם הבא מוצא את מחלקות השקילות של $\sim_F$:

1. עבור $i=0$ $Q/_{\equiv_i}=\{F,Q\setminus F\}$
2. עבור $0<i$ לכל $p\equiv_{i-1}q$ בדוק לכל $a\in\Sigma$ אם $\delta(p,a)\equiv_{i-1}\delta(q,a)$
   2.1. אם כן, אז $p\equiv_i q$
   2.2. אחרת, $p\not\equiv_i q$
3. אם $Q/_{\equiv_i}=Q/_{\equiv_{i-1}}$ עצור

הוכחה: נראה את נכונות האלגוריתם בשלבים:

א. נראה ש- $p\equiv_i q$ אמ"מ $p\sim_i q$.
באינדוקציה על i:
1. עבור $i=0$ הטענה ברורה כי ε מפריד בין F ל- $Q\setminus F$
2. עבור $i>0$ מהגדרת האלגוריתם נובע ש- $p\equiv_i q$ אמ"מ

$\forall a\in\Sigma\left(\delta(p,a)\equiv_{i-1}\delta(q,a)\right)$. מהנחת האינדוקציה זה נכון אמ"מ

$\forall a\in\Sigma\left(\delta(p,a)\sim_{i-1}\delta(q,a)\right)$. מהגדרת $\sim_i$ זה נכון אמ"מ

$$\forall a\in\Sigma\forall w\in\Sigma^*\left(|w|\le i-1\to\left(\delta^*\left(\delta(p,1),w\right)\in F\leftrightarrow\delta^*\left(\delta(q,a),w\right)\in F\right)\right)$$

אמ"מ $\forall a\in\Sigma\forall w\in\Sigma^*\left(|aw|\le i\to\left(\delta^*(p,aw)\in F\leftrightarrow\delta^*(q,aw)\in F\right)\right)$ אמ"מ

$\forall z\in\Sigma^*\left(|z|\le i\to\left(\delta^*(p,z)\in F\leftrightarrow\delta^*(q,z)\in F\right)\right)$ אמ"מ $p\sim_i q$.

ב. נראה שהאלגוריתם עוצר.
נשים לב שהיחס $\equiv_{i+1}$ מעדן את היחס $\equiv_i$, כלומר בכל שלב המחלקה יכולה להתפצל. מספר הפיצולים חסום ע"י $|Q|$ ולכן מספר האיטרציות של האלגוריתם חסום ע"י $|Q|$.

ג. נראה שאם עצרנו ב- $i=k$ אז לכל $k\le j$ מתקיים $p\equiv_j q\leftrightarrow p\equiv_k q$.
באינדוקציה על j:
1. עבור $j=k$ זה כמובן ברור.
2. עבור $k<j$ מתקיים

$$\begin{aligned}
p\equiv_j q &\leftrightarrow && p\equiv_{j-1}q\wedge\left(\forall a\in\Sigma\left(\delta(p,a)\equiv_{j-1}\delta(q,a)\right)\right)\\
&\overset{IH}{\leftrightarrow} && p\equiv_k q\wedge\left(\forall a\in\Sigma\left(\delta(p,a)\equiv_k\delta(q,a)\right)\right)\\
&\overset{Q/_{\equiv_k}=Q/_{\equiv_{k-1}}}{\leftrightarrow} && p\equiv_{k-1}q\wedge\left(\forall a\in\Sigma\left(\delta(p,a)\equiv_{k-1}\delta(q,a)\right)\right)\\
&\leftrightarrow && p\equiv_k q
\end{aligned}$$

ד. מכאן נובע ש- $p\sim_F q\leftrightarrow p\sim_\infty q\leftrightarrow p\equiv_k q$ שהרי ברגע $p\equiv_k q$ זה נכון גם לכל $j\ge k$ ולכן אין בכלל מילה מפרידה.

☺

מסקנה: בהינתן אס"ד $A=\langle Q,\Sigma,\delta,q_0,F\rangle$ ניתן לבנות אס"ד מינימאלי שקול ע"י
$A'=\left\langle Q/_{\sim_\infty},\Sigma,\delta',[q_0]_{\sim_\infty},F/_{\sim_\infty}\right\rangle$ כאשר $\delta'\left([q]_{\sim_\infty},a\right)=\left[\delta(q,a)\right]_{\sim_\infty}$.

הוכחה: ראשית צריך להראות ש-δ' מוגדרת היטב, כלומר $\delta'\left([q]_{\sim_\infty},a\right)$ אינו תלוי בבחירת הנציג.
אבל זה ברור מההגדרה של $[q]_{\sim_\infty}$. לפי הגדרה לכל $p\in[q]_{\sim_\infty}$ מתקיים $\delta(p,a)=\delta(q,a)$.

ברור ש-$L(A)=L(A')$ לפי הגדרת $\sim_\infty$.

נראה ש-$x\sim_L y\leftrightarrow\delta^*(q_0,x)\sim_\infty\delta^*(q_0,y)$:

$$\begin{aligned}
x\sim_L y \quad &\leftrightarrow \quad \forall z\in\Sigma^*\left(xz\in L\leftrightarrow yz\in L\right)\\
&\leftrightarrow \quad \forall z\in\Sigma^*\left(\delta^*(q_0,xz)\in F\leftrightarrow\delta^*(q_0,yz)\in F\right)\\
&\leftrightarrow \quad \forall z\in\Sigma^*\left(\delta^*\left(\delta^*(q_0,x),z\right)\in F\leftrightarrow\delta^*\left(\delta^*(q_0,y),z\right)\in F\right)\\
&\leftrightarrow \quad \delta^*(q_0,x)\sim_\infty\delta^*(q_0,y)
\end{aligned}$$

מאחר שמספר מחלקות השקילות של $\sim_\infty$ שווה למספר מחלקות השקילות של $\sim_L$ נקבל ש-A' מינימאלי.

☺

שאלות מעניינות על אוטומטים

בהינתן אוטומט A היינו רוצים לדעת עליו כמה דברים:

1. **שייכות** של מילה לשפת האוטומט – האם $w\in L(A)$?
2. **ריקנות** – האם $L(A)=\varnothing$?
3. **אוניברסאליות** – האם $L(A)=\Sigma^*$?

לכל השאלות האלה יש תשובות פשוטות אבל היעילות של הפיתרון משתנה אם A דטרמיניסטי או לא.

1. שייכות:
   - אם A דטרמיניסטי פשוט מריצים את A על w והתשובה מתקבלת תוך $O(|w|)$ צעדים.
   - אם A לא דטרמיניסטי ניתן לפעול בשלוש דרכים. הדרך הנאיבית היא לבנות את האס"ד השקול ואז להריץ אותו. זה יעלה לנו $O\left(2^{|Q|}+|w|\right)$ צעדים. גישה אחרת היא לבנות גרסה דטרמיניסטית תוך כדי ריצה, כלומר בכל שלב ליצור רק את המצבים

שנחוצים עבור המילה הספציפית שלנו. התהליך הזה צורך רק $O\left(\left|Q\right|^2\left|w\right|\right)$ צעדים.

האופציה הכי טובה היא לבנות את אוטומט החיתוך של $\left\{w\right\}\cap L\left(A\right)$ ואז לבדוק ריקנות.

2. ריקנות: כאן התשובה לא מושפעת מהדטרמיניסטיות של האוטומט. אוטומט הוא לא ריק אמ"מ קיים מסלול ממצב התחלתי למצב מקבל כלשהו. הבדיקה הזאת היא בדיקה פולינומיאלית פשוטה, למשל ע"י DFS.
3. אוניברסאליות: יש לבדוק את הריקנות של האוטומט המשלים.
   - עבור אס"ד זה פשוט מאוד וכבר ראינו איך הופכים אס"ד לאס"ד המשלים.
   - עבור אס"ל לא ניתן לקבל את האס"ל המלים באותו אופן כמו לאס"ד. לכן אין ברירה אלא לחרצן אותו ואז לבדוק אוניברסאליות.

שפות חסרות הקשר

דקדוקים חסרי הקשר

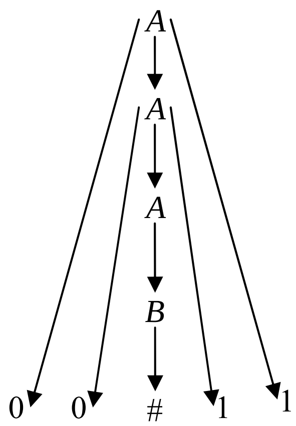
הגדרות:

1. **דקדוק חסר הקשר** הוא רביעייה $G=\langle V,\Sigma,R,S\rangle$ כאשר
   - V קבוצה סופית של משתנים
   - Σ קבוצה סופית של טרמינלים שזרה ל- V
   - R קבוצה סופית של חוקי גזירה מהצורה $(\Sigma\cup V)^{*}V(\Sigma\cup V)^{*}\to(\Sigma\cup V)^{*}$
   - $S\in V$ משתנה התחלתי
2. נאמר ש- $uAv\Rightarrow uwv$ אם $A\to w$ חוק בדקדוק כאשר $A\in V$ ו- $w,u,v\in(\Sigma\cup V)^{*}$.
   נאמר ש- $u\Rightarrow\Rightarrow v$ אם קיימת סדרה $u_1,\dots,u_k$ עבור $1\le k$ כך ש-
   $u=u_1\Rightarrow u_2\Rightarrow\dots\Rightarrow u_k=v$. תהליך זה נקרא **גזירה** של v מ- u.
3. **השפה של דקדוק** היא $L(G)=\{w\in\Sigma^{*}:s\Rightarrow\Rightarrow w\}$.
4. שפה אשר מתקבלת ע"י דקדוק חסר הקשר נקראת **שפה חסרת הקשר** (להלן שח"ה).
5. נסמן את מחלקת השפות חסרות ההקשר ב- CFL.

דוגמאות:

1. נגדיר דקדוק חסר הקשר $G=\langle V,\Sigma,R,S\rangle$ ע"י:
   - $V=\{A,B\}$
   - $\Sigma=\{0,1,\#\}$
   - חוקי הגזירה הם:
     i. $A\to 0A1$
     ii. $A\to B$
     iii. $B\to\#$
   - המשתנה ההתחלתי הוא A

למשל, הנה גזירה של המילה $00\#11$: $A\Rightarrow 0A1\Rightarrow 00A11\Rightarrow 00B11\Rightarrow 00\#11$. את הגזירה ניתן גם לתאר ע"י **עץ גזירה**:



קל לראות ש- $L(G)=\left\{0^n\#1^n : n\geq 0\right\}$.

2. נכתוב דקדוק עבור השפה $(0+1)^* 0(0+1)$
   - $V=\left\{S,A\right\}$
   - $\Sigma=\left\{0,1\right\}$
   - חוקי הגזירה:
     i. $S\rightarrow A00$
     ii. $S\rightarrow A01$
     iii. $A\rightarrow \varepsilon \mid 0A \mid 1A$
3. שפת הסוגריים המקוננים באופן חוקי: $G=\left\langle \left\{S\right\},\left\{a,b\right\},R,S\right\rangle$ כאשר חוקי הגזירה הם $S\rightarrow aSb\mid SS\mid \varepsilon$. נשים לב ששפה זו אינה רגולרית שהרי $L(G)\cap a^*b^*=\left\{a^nb^n : 0\leq n\right\}$. אילו $L(G)$ הייתה רגולרית הינו מקבלים שגם $\left\{a^nb^n : 0\leq n\right\}$ רגולרית, בניגוד למה שכבר הוכחנו קודם.
4. שפת הפלינדרומים $L=\left\{w\in\left\{0,1\right\}^* : w=w^R\right\}$ מתקבלת ע"י חוק הגזירה $S\rightarrow 0S0\mid 1S1\mid 0\mid 1\mid \varepsilon$.
5. שפת האיחוד $\left\{0^n1^n : 0\leq n\right\}\cup\left\{1^n0^n : 0\leq n\right\}$ מתקבלת ע"י חוקי הגדירה הבאים:
   - $S\rightarrow A\mid B$
   - $A\rightarrow 0A1\mid \varepsilon$
   - $A\rightarrow 1A0\mid \varepsilon$

   באותו אופן ניתן להגדיר דקדוק לכל שפה שהיא איחוד של שתי שפות הקשר. יהיו $G_1=\left\langle V_1,\Sigma_1,R_1,S_1\right\rangle$, $G_2=\left\langle V_2,\Sigma_2,R_2,S_2\right\rangle$ שני דקדוקים חסרי הקשר. אזי $L(G_1)\cup L(G_2)$ מתקבלת ע"י דקדוק $G=\left\langle V,\Sigma,R,S\right\rangle$ המוגדר באופן הבא:
   - $V=V_1\cup V_2\cup\left\{S\right\}$ כאשר S משתנה חדש והוא המשתנה ההתחלתי
   - $\Sigma=\Sigma_1\cup\Sigma_2$
   - $R=\left\{S\rightarrow S_1\mid S_2\right\}\cup R_1\cup R_2$

משפט: מחלקת השפות הרגולריות מוכלת במחלקת השפות חסרות ההקשר.

הוכחה: צריך להראות שבהינתן אס"ד $A=\langle Q,\Sigma,q_0,\delta,F\rangle$ קיים דקדוק חסר הקשר $G=\langle V,\Sigma,R,S\rangle$ כך ש- $L(A)=L(G)$.

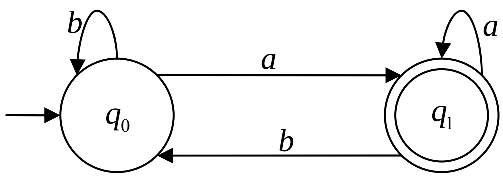
נגדיר את G באופן הבא:

1. לכל $q_i \in Q$ נגדיר משתנה v_i
2. v_0 המתאים ל- q_0 הוא המשתנה ההתחלתי
3. נגדיר את כללי הגזירה כך:
   - אם $\delta(q_i,a)=q_j$ נוסיף חוק $v_i \to av_j$
   - אם $q_i \in F$ נוסיף חוק $v_i \to \varepsilon$

נראה כעת ש- $L(A)=L(G)$. תהי $w=w_1...w_n$ ונראה ש- w $v_0 \Rightarrow\Rightarrow$ אמ"מ $w\in L(A)$.
$w\in L(A)$ אמ"מ קיימת ריצה $r_0r_1...r_n$ כך ש- $r_0=q_0$, $r_n\in F$ ו- $\delta(r_i,w_{i+1})=r_{i+1}$ לכל $0\le i<n$ וזה נכון אמ"מ קיימת שרשרת גזירות
$V_0 \Rightarrow w_1V_1 \Rightarrow w_1w_2V_2 \Rightarrow ... \Rightarrow w_1...w_nV_n \Rightarrow w_1...w_n$ כאשר $V_i=r_i$ לכל $0\le i\le n$.

☺

דוגמה: $L=(a+b)^*a$ שפת כל המילים הנגמרות ב- a.



באס"ד יש שני מצבים ולכן יתאימו להם שני משתנים $V=\{v_0,v_1\}$ וחוקי הגזירה יהיו:

1. $v_0 \to av_1 \mid bv_0$
2. $v_1 \to bv_0 \mid av_1 \mid \varepsilon$

משפט: $REG \subset CFL$

הוכחה: נראה אין בהינתן ביטוי רגולרי r נבנה דקדוק G שמגדיר את אותה השפה. נוכיח באינדוקציה על הבנייה של הביטוי הרגולרי r.

1. עבור ביטויים רגולריים בסיסיים:
   - אם $r=\varnothing$ הדקדוק הוא $S\to S$
   - אם $r=\varepsilon$ הדקדוק הוא $S\to\varepsilon$

- אם $r=a$ עבור $a\in\Sigma$ הדקדוק הוא $S\to a$

2. עבור ביטויים רגולריים מורכבים. יהיו r_1,r_2 ביטויים רגולריים ונניח שיש דקדוקים שגוזרים את אותן השפות עם משתנים התחלתיים S_1,S_2 בהתאמה.
   - אם $r=r_1+r_2$ אז הדקדוק יכלול את $S\to S_1\mid S_2$ ואז חוקי הגזירה של הדקדוק של $L(r_1)$ ושל $L(r_2)$.
   - אם $r=r_1.r_2$ אז הדקדוק יכלול את $S\to S_1S_2$ ואז חוקי הגזירה של הדקדוק של $L(r_1)$ ושל $L(r_2)$.
   - אם $r=r_1^*$ אז הדקדוק יכלול את $S\to S_1S$ ואת חוקי הגזירה של הדקדוק של $L(r_1)$.

ברור ש- $L(G)=L(r)$.

☺

הגדרה: יהי G דקדוק חסר הקשר. G ייקרא דקדוק לינארי ימני אם כל החוקים בו הם מהצורה $V\to aV'$ או $V\to\varepsilon$ כאשר V,V' משתנים ו- a ליטרל. G ייקרא דקדוק לינארי שמאלי אם כל החוקים בו מהצורה $V\to V'a$ או $V\to\varepsilon$.

מסקנה: שפה L היא רגולרית אמ"מ קיים דקדוק לינארי ימני G כך ש- $L(G)=L$.

הוכחה:

($\Leftarrow$) אם השפה היא רגולרית ראינו שקיים דקדוק G כך ש- $L(G)=L$ ומההגדרה של G נובע שהוא דקדוק לינארי ימני.

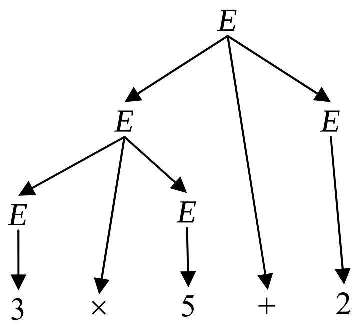
($\Rightarrow$) נשים לב שהבנייה שעשינו בהוכחת המשפט הקודם היא הפיכה. לכן בהינתן דקדוק לינארי ימני נוכל לבנות אס"ד A כך ש- $L(G)=L(A)$ ולכן $L(G)$ היא רגולרית.

☺

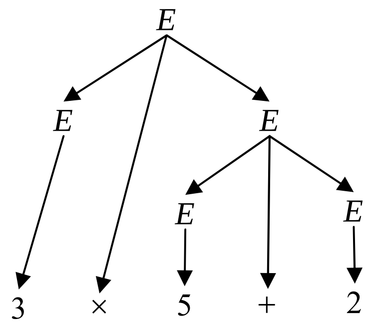
דוגמאות:

1. דקדוק לביטויים חשבוניים:
   - $\Sigma = \{0,...,9,+,\times\}$
   - $E \to E\times E \mid E+E \mid 0 \mid ... \mid 9$

   למילה $3\times 5+2$ יש שני עצי גזירה שונים:



$(3\times 5)+2$

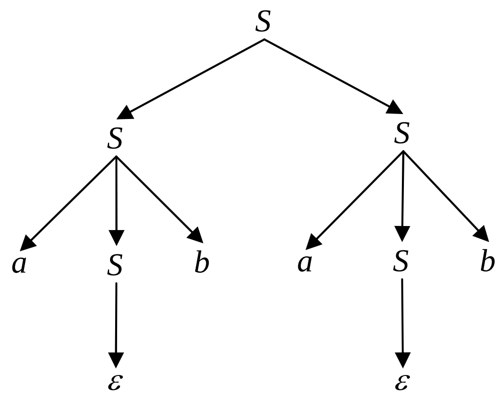


$3\times(5+2)$

   לא ברור אם הכוונה היא ל- $3\times(5+2)$ או ל- $(3\times 5)+2$.

2. אם יש שתי גזירות שונות לאותה המילה זה לא בהכרח אומר שעצי הגזירה הם שונים.
   למשל, נסתכל על שפת הסוגריים המקוננים באופן חוקי המוגדרת ע"י $S \to aSb \mid SS \mid \varepsilon$.
   למילה $abab$ יש שתי גזירות שונות:
   - $S \to SS \to aSbS \to aSbaSb \to abaSb \to abab$
   - $S \to SS \to aSbS \to abS \to abaSb \to abab$

   אבל עצי הגזירה של שתי הגזירות הם זהים:



הגדרות:

1. יהי G דקדוק חסר הקשר ונניח ש- $S \Rightarrow\Rightarrow w$. נאמר שהגזירה של w היא **שמאלית ביותר** אם בכל צעד בגזירה מוחלף המשתנה השמאלי ביותר.
2. נאמר ש- w **נגזרת באופן רב משמעי** אם יש ל- w שתי גזירות שמאליות ביותר שונות ומינימאליות.
3. דקדוק הוא **רב משמעי** אם קיימת מילה w אשר נגזרת באופן רב משמעי.

הערה: קיימות שפות חסרות הקשר אשר כל דקדוק שגוזר אותן הוא רב משמעי.

שאלות מעניינים על דקדוקים חסרי הקשר

כמו במקרה של אוטומטים היינו רוצים לדעת לבדוק שייכות של מילה לדקדוק ולבדוק ריקנות של דקדוק. לצורך המטרה נזדקק לכמה כלים חדשים.

הגדרה: דקדוק חסר הקשר הוא ב**צורה נורמאלית של חומסקי** אם כל חוקי הגזירה הם מאחת הצורות הבאות:

1. $A \to BC$ עבור $A, B, C \in V$ ו- $B, C \neq S$ כאשר S המשתנה ההתחלתי
2. $A \to a$ עבור $A \in V$ ו- $a \in \Sigma$
3. $S \to \varepsilon$

טענה: לכל דקדוק חסר הקשר G קיים דקדוק חסר הקשר שקול G' בצורה נורמאלית של חומסקי.

הוכחה: נראה איך בהינתן דקדוק G נוכל להפוך אותו לדקדוק שקול בצורה נורמאלית של חומסקי. נעשה זאת בשלבים ונשים לב שבכל שלב איננו משנים את השפה של הדקדוק. בסיום השלבים, כל חוקי הגזירה יהיו באחת הצורות הדרושות.

1. אם יש חוק גזירה שבו S מופיע בצד ימין נגדיר משתנה התחלתי חדש S' ונוסיף חוק $S' \to S$.
2. אם יש חוק $A \to \varepsilon$ עבור $A \neq S$, לכל חוק מהצורה $B \to uAv$ כאשר $u, v \in (\Sigma \cup V)^*$ נוסיף את החוק $B \to uv$. אם לא קיים חוק $B \to uAv$ כזה אז A לא מופיע בצד ימין של אף חוק גזירה ולכן אנחנו יכולים פשוט לזרוק את $A \to \varepsilon$.
3. אם יש חוק מהצורה $A \to B$, לכל חוק מהצורה $B \to u$ עבור $u \in (\Sigma \cup V)^*$ נוסיף חוק $A \to u$ (אלא אם כן, כבר טיפלנו בו קודם). כעת ניתן לזרוק את $A \to B$ מהשפה ונישאר עם דקדוק שקול.
4. לכל חוק מהצורה $A \to u_1 ... u_k$ כאשר $3 \leq k$ ו- $u_i \in \Sigma \cup V$ נחליף אותו ב-$k-1$ חוקים חדשים: $A \to u_1 A_1, A_1 \to u_2 A_2, ..., A_{k-2} \to u_{k-1} u_k$ כאשר $A_1, ..., A_{k-2}$ משתנים חדשים.
5. לכל חוק מהצורה $A \to u_1 u_2$ כאשר $u_1, u_2 \in \Sigma \cup V$, אם $u_1 \in \Sigma$ נדגיר משתנה חדש U_1 ונוסיף את החוקים $U_1 \to u_1$ ו- $A \to U_1 u_2$. כנ"ל עבור u_2.

בכל השלבים, ברור שעברנו לדקדוק שקול, ולבסוף נותרנו עם דקדוק חסר הקשר בצורה נורמאלית של חומסקי אשר שקול לדקדוק המקורי.

☺

טענה: יהי דקדוק חסר הקשר $G=\langle \Sigma, V, R, S\rangle$. אם קיימת גזירה $S \Rightarrow\Rightarrow w$ אז קיימת גזירה מהצורה $S \Rightarrow A_1A_2 \Rightarrow B_1B_2B_3 \Rightarrow ... \Rightarrow Z_1Z_2...Z_n \Rightarrow w_1w_2...w_n$.

הוכחה: ניתן להניח שהדקדוק נתון בצורה נורמאלית של חומסקי. אם קיבלנו $w_1...w_n$ סימן שיש משתנים שעברו אליהם, וכל שני משתנים באו ממשתנה אחד.

☺

מסקנה: ניתן לבדוק בקלות וביעילות שייכות של מילה לשח"ה.

הוכחה: נשתמש בתכנון דינאמי. נסמן ב-$T[i, j]$ את כל המשתנים שמהם ניתן לגזור את תת המילה $w_i...w_j$, כלומר $T[i, j]=\{A\in V : A \Rightarrow\Rightarrow w_i...w_j\}$. אנחנו רוצים לדעת אם $S \in T[1,n]$.

האלגוריתם יפעל כך:

1. אתחול: לכל $i=1,...,n$ $T[i,i]=\{A\in V : A \to w_i\}$. בגלל שהדקדוק נתון בצורה נורמאלית של חומסקי הבדיקה הזאת קלה ממש.
2. לכל $k=1,...,n-1$ (k מסמל את אורך התת מילה פחות 1)

2.1. לכל $i=1,...,n-k$ (i מסמל את המיקום בתוך התת מילה)

2.1.1. לכל $j=0,...,k-1$ (j מסמל את אורך החלק הראשון פחות 1)

2.1.1.1. אם יש חוק מהצורה $A\to BC$ כאשר $B\in T[i,i+j]$ ו-$C\in T[i+j+1,i+k]$

2.1.1.1.1. נוסיף את A ל-$T[i,i+k]$

יש באלגוריתם שלוש לולאות מקוננות והחלק הפנימי תלוי במספר חוקי הגזירה, לכן זמן הריצה של האלגוריתם הוא $O(n^3|R|)$.

☺

משפט: ניתן לבדוק ביעילות ריקנות של דקדוק חסר הקשר.

הוכחה: יהי דקדוק $G=\langle \Sigma, V, R, S\rangle$. נבנה קבוצה E של משתנים שמהם ניתן לגזור מילים. נרצה לדעת אם הקבוצה מכילה את S או לא. נגדיר סדרה של קבוצות באינדוקציה:

$$E_1=\{A\in V : \exists a\in\Sigma((A\to a)\in R)\}$$
$$E_{i+1}=E_i\cup\{A\in V : \exists B,C\in E_i((A\to BC)\in R)\}$$

נטען ש-E_i היא קבוצת כל המשתנים שמהם ניתן לגזור מילה כלשהי תוך לכל היותר i צעדים.

נוכיח באינדוקציה על i:

1. עבור $i=1$ זה נובע ישירות מההגדרה של E_1
2. נניח עבור i ונוכיח עבור $i+1$. אם $A\in E_{i+1}$ אז $A\in E_i$ כלומר ניתן לגזור ממנו מילה תוך i צעדים לכל היותר (ובפרט תוך $i+1$ צעדים לכל היותר) או קיימים $B,C\in E_i$ כך ש-$A\to BC$ חוק בדקדוק. לפי הנחת האינדוקציה מ-B ומ-C ניתן לגזור מילה תוך לכל היותר i צעדים. לכן מהחוק $A\to BC$ ניתן לגזור את המילה שהיא השרשור של שתי המילים תוך $i+1$ צעדים לכל היותר.

כעת נטען שקיים i כך ש-$E_i=E_{i+1}$, אבל זה ברור משום שלכל היותר יכול להיות $E=V$. כעת ברור ש-$E=\sum_{i=1}^{k}E_i=E_k$ כאשר $E_k=E_{k+1}$. כדי לייצר את E יש צורך בלכל היותר $|V|$ איטרציות ובכל איטרציה יש לעשות $|R|$ בדיקות. לכן, תוך זמן $O(|V||R|)$ נמצא את E. לבסוף, נותר רק לבדוק אם $S\in E$ או $S\notin E$.

☺

אוטומט מחסנית

הגדרות:

1. **אוטומט מחסנית** (להלן א"מ) הוא שישייה $A=\langle Q,\Sigma,\Gamma,\delta,Q_0,F\rangle$ כאשר:
   - Q קבוצת מצבים סופית
   - Σ א"ב סופי של השפה
   - Γ א"ב סופי של המחסנית
   - $\delta:Q\times(\Sigma\cup\{\varepsilon\})\times(\Gamma\cup\{\varepsilon\})\to 2^{Q\times(\Gamma\cup\{\varepsilon\})}$ פונקציית מעברים לא דטרמיניסטית
   - $Q_0\subset Q$ קבוצת מצבים התחלתיים
   - $F\subset Q$ קבוצת מצבים מקבלים
2. **קונפיגורציה** של א"מ A היא המצב ש-A נמצא בו והמצב של המחסנית. אם A נמצא במצב q ובמחסנית רשום $w_1...w_n$ כאשר w_1 התו האחרון שנרשם במחסנית ו-w_n התו הראשון שנרשם נסמן את הקונפיגורציה ב-$\langle q,w_1...w_n\rangle$.
3. קונפיגורציה $\langle q',s'\rangle$ היא σ**-עוקבת** ל-$\langle q,s\rangle$ אם s ניתנת לחלוקה $s=at$ כאשר $a\in\Gamma\cup\{\varepsilon\}$ ו-$t\in\Gamma^*$ כך ש-$\langle q',b\rangle\in\delta(q,\sigma,a)$ ו-$s'=bt$.
4. **ריצה** של A על מילה $w\in\Sigma^*$ הי סדרה של קונפיגורציות $c_0,...,c_m$ כך ש-w ניתנת לכתיבה כ-$w_1...w_m$ כאשר $w_i\in\Sigma\cup\{\varepsilon\}$ לכל $1\le i\le m$ ומתקיים:
   - $c_0=\langle q_0,\varepsilon\rangle$ עבור $q_0\in Q_0$

* לכל $0 \le i \le m-1$ הקונפיגורציה c_{i+1} היא w_{i+1}-עוקבת ל- c_i
5. **קונפיגורציה התחלתית** היא $\langle q, \varepsilon \rangle$ עבור $q \in Q_0$ כלשהו.
6. **קונפיגורציה מקבלת** היא $\langle q, s \rangle$ עבור $q \in F$.
7. ריצה $c_0, \ldots, c_m$ היא **מקבלת** אם c_m קונפיגורציה מקבלת.
8. השפה של א"מ היא קבוצת כל המילים שקיימת עבורן ריצה מקבלת. את השפה של A מסמנים ב- $L(A)$.

דוגמה: נתכנן א"מ עבור השפה $L = \{0^n 1^n : 0 \le n\}$. הרעיון הוא להשתמש במחסנית על מנת לזכור כמה אפסים קראנו. בכל קריאה של 0 נדחוף סימן כלשהו למחסנית. בקריאת 1 נשלוף את הסימן. אם בסוף קריאת המילה המחסנית ריקה סימן שהמילה בשפה. נשים לב לכמה פרטים:

* יש צורך בסימן שיסמן את תחילת המחסנית
* יש לוודא גם שקודם מופיעים כל האפסים ולאחר מכן כל האחדות

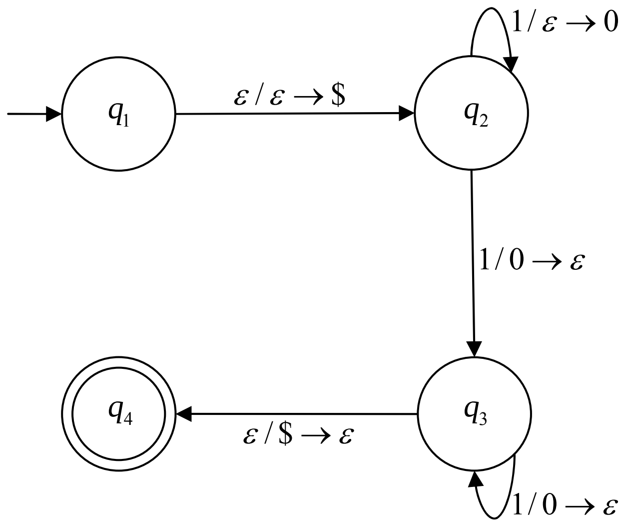
נגדיר כעת א"מ $A = \langle Q, \Sigma, \Gamma, \delta, Q_0, F \rangle$ באופן הבא:

* $Q = \{q_1, q_2, q_3, q_4\}$
* $\Sigma = \{0, 1\}$
* $\Gamma = \{0, \$\}$ כאשר $\$$ יסמן את תחילת המחסנית
* $Q_0 = \{q_1\}$
* $F = \{q_1, q_4\}$
* פונקציית המעברים מוגדרת כך:
  * $\delta(q_1, \varepsilon, \varepsilon) = \{(q_2, \$)\}$ - מאתחלים את המחסנית עם $\$$.
  * $\delta(q_2, 0, \varepsilon) = \{(q_2, 0)\}$ - כל עוד מגיעים אפסים דוחפים 0 למחסנית.
  * $\delta(q_2, 1, 0) = \{(q_3, \varepsilon)\}$ - ברגע שמגיע ה-1 הראשון שולפים 0 מהמחסנית ועוברים למצה הבא.
  * $\delta(q_3, 0, 0) = \{(q_3, \varepsilon)\}$ - כל עוד מגיעות אחדות שולפים 0 מהמחסנית.
  * $\delta(q_3, 1, \$) = \{(q_4, \varepsilon)\}$ - אם הגענו לתחילת המחסנית סימן שמספר האחדות שווה למספר האפסים והמילה בשפה.

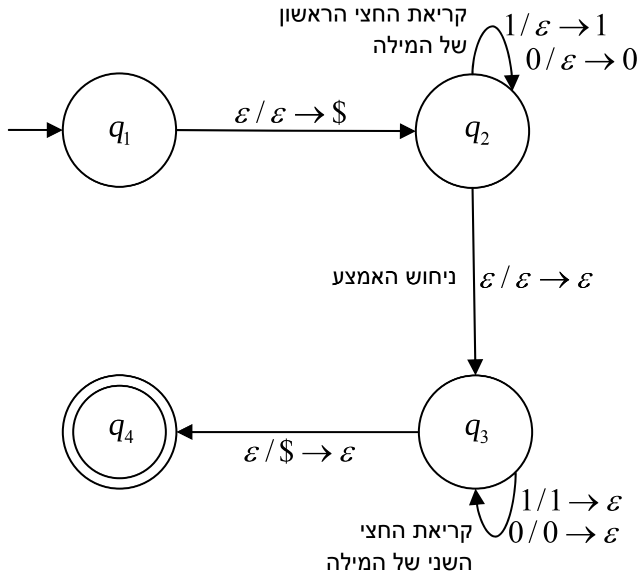
א"מ ניתן לתאר בצורה גראפית. אם $(q', \gamma') \in \delta(q, \sigma, \gamma)$ נתאר זאת כך:

$$q \xrightarrow{\sigma / \gamma \to \gamma'} q'$$

אז הנה הא"מ A כך ש- $L(A) = L$:



דוגמה: שפת הפלינדרומים באורך זוגי $L=\left\{w\cdot w^R : w\in\{0,1\}^*\right\}$. L גם מתקבלת ע"י הדקדוק $S\to 0S0\,|\,1S1\,|\,\varepsilon$. הרעיון הוא שנשמור את המילה שקראנו עד כה על המחסנית. ברגע שנחליט שהגענו לאמצע המילה נתחיל לשלוף מהמחסנית בהתאמה לאותיות שמגיעות.



משפט: מחלקת השפות חסרות ההקשר שווה למחלקת השפות שמתקבלות ע"י א"מ.

הוכחה: התבטלה בגלל השביתה (מי שמתעניין יכול למצוא אותה במהדורה הראשונה של Sipser בעמוד 106 או במהדורה השנייה בעמוד 115).

שפות לא חסרות הקשר

למת הניפוח לשפות חסרות הקשר: תהי L שפה חסרת הקשר. אזי קיים $1 \leq p$ (שנקרא לו קבוע הניפוח) כך שלכל $w \in L$ כך ש-$|w| \geq p$ קיימת חלוקה $w = uvxyz$ המקיימת:

1. לכל $0 \leq i$ $uv^i xy^i z \in L$
2. $|vy| \geq 1$
3. $|vxy| \leq p$

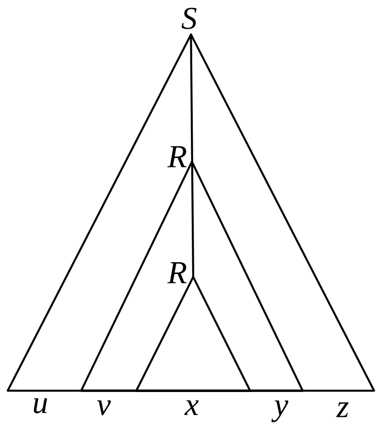
הוכחה: יהי b האורך המקסימלי של צד ימין של חוק בדקדוק שגוזר את L. נניח ש-$b \geq 2$ (אחרת כל המילים ב-L הן באורך 1 או 2 ואז הטענה ברורה).

תהי $w \in L$. בעץ הגזירה T של w יש לכל צומת לכל היותר b בנים. אז מתקיים:

- אם $\text{height}(T) = h$ אז מספר העלים בעץ הוא לכל היותר b^h.
- אם $\text{height}(T) < h$ אז מספר העלים בעץ קטן ממש מ-b^h.
- אם $\text{height}(T) \geq h$ אז מספר העלים בעץ הוא לכל הפחות b^h.

נבחר $p = b^{|V|+1}$.[1] אם $|w| \geq p$ אז גובה עץ הגזירה המינימאלי[2] הוא לפחות $|V|+1$. שהרי אורך המילה הוא מספר העלים בעץ. אם מספר העלים הוא לכל הפחות p אז הגובה הוא לכל הפחות $\log_b p = \log_b b^{|V+1|} = |V|+1$. אז יש טרמינל שבמסלול אליו מ-S יש לפחות $|V|+1$ צמתים. אבל אז לפי עיקרון שובך היונים קיים משתנה R שחוזר לפחות פעמיים ב-$|V|+1$ הרמות התחתונות.

נחלק את w כמו באופן הבא:

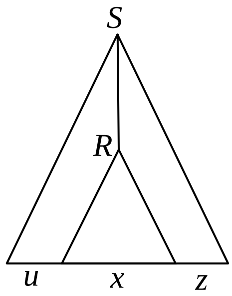


[1] נזכור ש-V הם המשתנים של הדקדוק

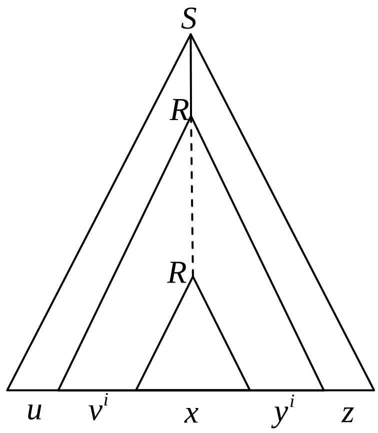
[2] כלומר אין ל-w עץ גזירה עם פחות צמתים

נראה שמתקיימים התנאים הדרושים:

- לכל $0 \le i$ $uv^i xy^i z \in L$
  - o בוודאי $uxz \in L$ משום שהיה ניתן לגזור אותה כך:



  - o אם $i > 0$ ניתן לגזור את $uv^i xy^i z$ באופן הבא:



- $|vy| > 0$ שהרי אם $v = \varepsilon = y$ אז היינו מקבלים סתירה למינימאליות של עץ הגזירה של w.
- $|vxy| \le p$ משום ש-R חוזר על עצמו ב-$|V|+1$ הרמות האחרונות ולכן vxy נגזר מ-R בעץ שגובהו לכל היותר $|V|+1$. לכן $|vxy| \le b^{|V|+1} = p$.

☺

הערה: כמו בשפות רגולריות, בלמת הניפוח משתמשים בעיקר כדי להראות ששפה היא אינה חסרת הקשר. אם בהינתן שפה, מראים שלכל p קיימת מילה באורך לפחות p אשר לא ניתן לנפח אותה, אז השפה אינה חסרת הקשר.

דוגמאות:

1. $L = \left\{a^n b^n c^n : 0 \le n\right\}$

בהינתן p נסתכל על המילה $w = a^p b^p c^p$. נניח שקיימת חלוקה $w = uvxyz$ כמו בלמת הניפוח. מאחר ש-$|vxy| \le p$ מופיעות בו לכל היותר שתי אותיות שונות. לכן, אם מנפחים את v ואת y נקבל שעבור האות השלישית יש מעט מדי אותיות!

2. $L=\left\{w\cdot w : w\in\{0,1\}^*\right\}$

בהינתן p נסתכל על המילה $w=0^p1^p0^p1^p$ ונניח ש- $w=uvxyz$.

- אם כל vxy נמצא בחצי הראשון של w אז האות אחרי האמצע של uv^2xy^2z היא 1 ולכן $uv^2xy^2z\notin L$.
- אם כל vxy נמצא בחצי השני של w אז האות לפני האמצע של uv^2xy^2z היא 0 ולכן $uv^2xy^2z\notin L$.
- אם vxy כולל אות אמצעית אז uxz מהצורה $0^p1^i0^j1^p$ עבור $i+j<2p$ ולכן $uxz\notin L$.

חלק שני

תורת החישוביות

התזה של צ'רץ' וטיורינג

מכונות טיורינג

הגדרות:

1. **מכונת טיורינג דטרמיניסטית** (להלן מט"ד) היא שביעייה $M=\left\langle Q,\Sigma,\Gamma,\delta,q_0,q_{acc},q_{rej}\right\rangle$ כאשר:
   - Σ א"ב הקלט ואינו כולל את התו _
   - Γ א"ב העבודה כאשר $\Sigma\subset\Gamma$ ו- $\_\in\Gamma$
   - $\delta:Q\times\Gamma\rightarrow Q\times\Gamma\times\{L,R\}$ פונקציית מעברים דטרמיניסטית.
   - $q_0\in Q$ מצב התחלתי
   - $q_{acc}\in Q$ מצב מקבל
   - $q_{rej}\in Q$ מצב דוחה

   למכונת טיורינג בנוסף למצבים יש סרט עבודה אינסופי שבתחילת הריצה כתובה עליו מילת הקלט. בנוסף למכונה יש ראש קורא אשר נמצא בתא הראשון של הסרט בתחילת הריצה ויכול לזוז בכל שלב או תא אחד ימינה או תא אחד שמאלה. אם $\delta(q,\gamma)=(q',\gamma',R)$ פירוש הדבר שאם המכונה נמצאת במצב q והראש הקורא רואה את התו γ אז המכונה תעבור למצב q', תכתוב במקום γ γ' ותעבור תא אחד ימינה. אם $\delta(q,\gamma)=(q',\gamma',L)$ פירוש הדבר שאם המכונה נמצאת במצב q והראש הקורא רואה את התו γ אז המכונה תעבור למצב q', תכתוב במקום γ γ' ותעבור תא אחד שמאלה.

2. **מכונת טיורינג אי דטרמיניסטית** (להלן מטא"ד) היא שביעייה $M=\left\langle Q,\Sigma,\Gamma,\delta,q_0,q_{acc},q_{rej}\right\rangle$ כאשר:
   - Σ א"ב הקלט ואינו כולל את התו _
   - Γ א"ב העבודה כאשר $\Sigma\subset\Gamma$ ו- $\_\in\Gamma$
   - $\delta:Q\times\Gamma\rightarrow 2^{Q\times\Gamma\times\{L,R\}}$ פונקציית מעברים אי דטרמיניסטית.
   - $q_0\in Q$ מצב התחלתי
   - $q_{acc}\in Q$ מצב מקבל
   - $q_{rej}\in Q$ מצב דוחה

3. **קונפיגורציה** של מכונת טיורינג כלשהי (להלן מ"ט) כוללת שלושה פרמטרים:
   - המצב שבו המכונה נמצאת
   - תוכן הסרט
   - המקום שעליו נמצא הראש הקורא

   אם המכונה נמצאת במצב q, בסרט כתוב uv והראש הקורא מצביע על האות הראשונה של v נסמן זאת ע"י uqv (כמובן, u ו-v יכולים להיות המילה הריקה).

4. **קונפיגורציה התחלתית** עבור מילת קלט w היא q_0w
5. **קונפיגורציה סופית** היא מהצורה $uq_{acc}v$ או $uq_{rej}v$
6. **קונפיגורציה מקבלת** היא מהצורה $uq_{acc}v$
7. **קונפיגורציה דוחה** היא מהצורה $uq_{rej}v$
8. יהיו $a,b,c\in\Gamma$, $v,u\in\Gamma^*$ ו- $q,q'\in Q$. אזי $uq'acv$ **קונפיגורציה עוקבת** ל- $uaqbv$ אם $\delta\left(q,b\right)=\left(q',c,L\right)$ ו- $uacq'v$ קונפיגורציה עוקבת ל- $uaqbv$ אם $\delta\left(q,b\right)=\left(q',c,R\right)$.
אם הראש הקורא מצביע לתחילת הסרט, כלומר הקונפיגורציה היא qbv אז יכולות להיות לה שתי קונפיגורציות עוקבות: $q'cv$ (אם $\delta\left(q,b\right)=\left(q',c,L\right)$) ו- $cq'v$ (אם $\delta\left(q,b\right)=\left(q',c,R\right)$).
נשים לב שאם מדובר במטא"ד יש להחליף את כל סימני השוויון בפונקציית המעברים בסימני הכלה.
9. נאמר שמ"ט M **מקבלת** את w אם קיימת סדרת קונפיגורציות $c_0c_1...c_k$ כך ש-
   - c_0 קונפיגורציה התחלתית
   - c_k קונפיגורציה מקבלת
   - לכל $0\le i<k$ c_{i+1} קונפיגורציה עוקבת ל- c_i
10. **השפה** $L\left(M\right)$ של מ"ט M היא קבוצת כל המילים ש- M מקבלת.
11. אם $L\left(M\right)=L$ נאמר ש- M **מזהה** את L.
12. שפה היא **ניתנת למנייה רקורסיבית** (להלן נל"ר) אם קיימת מ"ט שמזהה אותה. מחלקת השפות הנ"לר מסומנת ב-/ RE
13. נאמר ששפה L היא **כריעה** או **רקורסיבית** אם קיימת מט"ד M שמזהה אותה וכמו כן, M עוצרת על כל הקלטים. במקרה זה, נאמר ש- M **מכריעה** את L. מחלקת השפות הרקורסיביות מסומנת ב- R.[3]
14. נאמר שמטא"ד M מכריעה שפה אם לכל קלט כל הריצות האפשריות עוצרות.
15. מחלקת השפות שמשלימתן היא נל"ר מסומנת ע"י $co-RE=\left\{L\subset\Sigma^*:L^c\in RE\right\}$.

דוגמה:

נתכנן מט"ד M שמזהה את השפה $L=\left\{0^{2^n}:n\ge 0\right\}$.

נגדיר פרדיקט מעל הטבעיים באופן הבא: $g\left(k\right)\leftrightarrow\exists n\left(k=2^n\right)$. נשים לב שמתקיים $g\left(k\right)\leftrightarrow\left(k\ne 0\right)\wedge\left(k=1\vee g\left(\frac{k}{2}\right)\right)$.

[3] ברור ש- $R\subset RE$

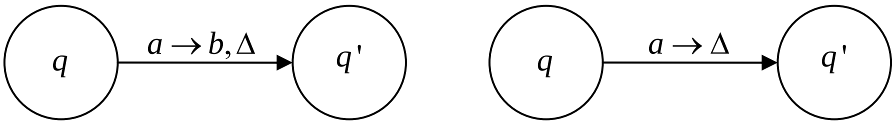
M תפעל באופן הבא:

א. סרוק את הסרט משמאל לימין ומחק כל 0 שני.
ב. אם על הסרט כתוב כעת רק 0 קבל
ג. אם בסרט כתוב מספר אי זוגי של אפסים דחה
ד. חזור עם הראש הקורה לתחילת הסרט
ה. עבור לשלב א'

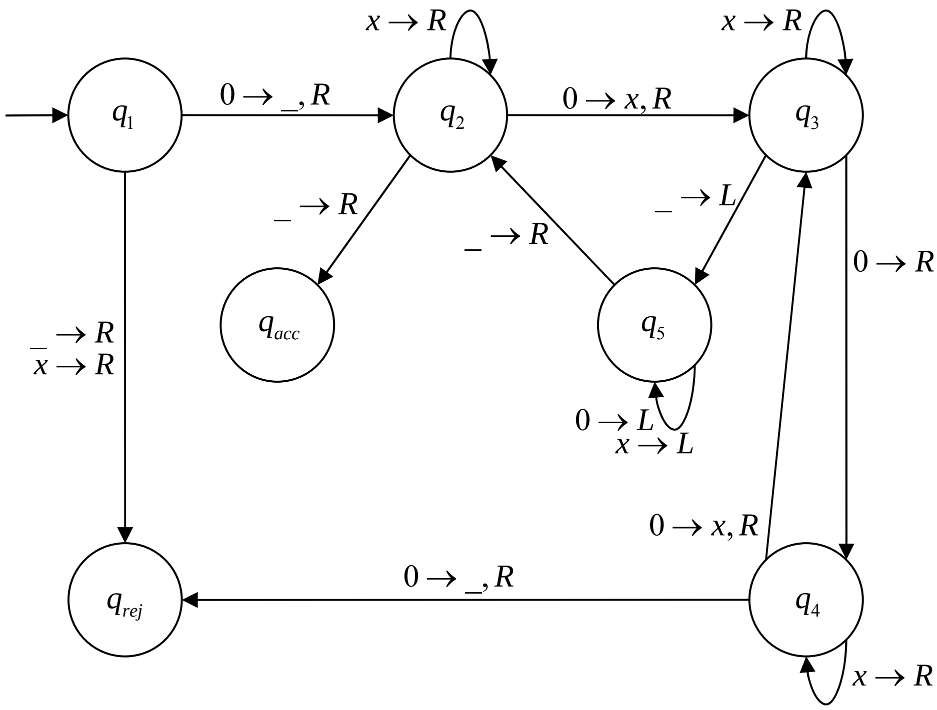
נתאר את M פורמאלית: $M = \left\langle \left\{q_1, q_2, q_3, q_4, q_5, q_{acc}, q_{rej}\right\}, \left\{0\right\}, \left\{0, \_, x\right\}, \delta, q_1, q_{acc}, q_{rej} \right\rangle$.

פונקציית מעברים ניתן לתאר בצורה גרפית באופן הבא:

$\delta(q,a) = (q', a, \Delta)$ $\qquad$ $\delta(q,a) = (q', b, \Delta)$



נצייר אם כן את M:



מעתה כאשר נתאר מ"ט, לא נתאר יותר באמצעות ההגדרה הפורמאלית אלא ניתן הסבר על אופן פעולתה בשפה עילית. ההצדקה הפורמאלית לכך תינתן כאשר נדבר על התזה של צ'רץ' טיורינג, אך כרגע קל יהיה להיווכח שאת כל הפעולות שמתוארות באלגוריתמים ניתן לבצע ע"י מ"ט.

משפט: המחלקה R סגורה תחת השלמה

הוכחה: תהי $L \in R$ ותהי $M = \left\langle Q,\Sigma,\Gamma,\delta,q_0,q_{acc},q_{rej} \right\rangle$ שמכריעה את L. נגדיר $M' = \left\langle Q',\Sigma',\Gamma',\delta',q'_0,q'_{acc},q'_{rej} \right\rangle$ באופן הבא:

- $Q' = Q$
- $\Sigma' = \Sigma$
- $\Gamma' = \Gamma$
- $\delta' = \delta$
- $q'_0 = q_0$
- $q'_{acc} = q_{rej}$
- $q'_{rej} = q_{acc}$

קל לראות ש- $L(M') = L^c$, שהרי M עוצרת על קל קלט ולכן בוודאי גם M' עוצרת על כל קלט.

יתר על כן, הריצות של M ושל M' זהות על כל קלט, אלא שמצב המקבל של M' הוא המצב הדוחה של M והמצב הדוחה של M' הוא המצב המקבל של M. לכן w מתקבלת ע"י M אמ"מ היא נדחית ע"י M'. אז M' מ"ט שמכריעה את L^c ולכן $L^c \in R$.

☺

משפט: $R = RE \cap co-RE$

הוכחה:

($\subset$) נניח ש- $L \in R$ אזי $L \in RE$ שהרי $R \subset RE$. אבל לפי המשפט הוקדם גם $L^c \in R$ ולכן $L^c \in RE$, כלומר $L \in co-RE$.

($\supset$) נניח ש- $L \in RE \cap co-RE$. אזי קיימת מ"ט M' שמזהה את L ויש מ"ט M'' שמזהה את L^c. בהינתן מילה w ידוע ש-M' עוצרת בריצה עליה אחרי מספר סופי של צעדים או M'' עוצרת בריצתה עליה אחרי מספר סופי של צעדים.

נבנה מ"ט M אשר פועלת באופן הבא:

א. נאתחל $i = 0$.
ב. נריץ את M' על w i צעדים.
ג. אם בשלב הקודם הגענו למצב מקבל M תקבל.
ד. אחרת, נריץ את M'' i צעדים.
ה. אם בשלב הקודם הגענו למצב מקבל M תדחה.
ו. אחרת נגדיל את i ב-1 ונחזור לשלב ב'.

בגלל שמובטח לנו שאו M' או M'' עוצרת אחרי מספר סופי של צעדים האלגוריתם חייב להסתיים אחרי מספר סופי של צעדים. וברור ש- M מכריעה את L.

☺

הגדרה: **מכונת טיורינג עם** k **סרטים** היא מכונה שזהה למכונת טיורינג אלא שיש לה k סרטים. לכל סרט יש ראש קורא משלו ופונקציית המעברים היא $\delta : Q\times\Gamma^k \to Q\times\Gamma^k\times\{L,R\}^k$. בכל שלב כל הראשים הקוראים נעים סימולטנית. בתחילת הריצה בסרט הראשון כתוב הקלט ושאר הסרטים ריקים.

משפט: לכל מ"ט עם k סרטים קיימת מ"ט שקולה עם סרט יחיד.

הוכחה: תהי $M=\langle Q,\Sigma,\Gamma,\delta,q_0,q_{acc},q_{rej}\rangle$ מ"ט עם k סרטים. נבנה מ"ט $M'=\langle Q',\Sigma,\Gamma',\delta',q_0',q_{acc}',q_{rej}'\rangle$ עם סרט יחיד שקולה לה. ברור שמה שאנחנו צריכים זה רק היכולת לסמלץ k סרטים על סרט יחיד. נשים לב שלמרות כל סרט הוא אינסופי, בכל שלב בכל סרט משתמשים רק במספר סופי של תאים. נוסף על היכול לשמור את כל תכולת הסרטים עלינו לדעת איפה נמצא הראש הקורא של כל אחד מהם. לצורך כך נגדיר $\dot{\Gamma}=\{\dot{w}:w\in\Gamma\}$. אם בסרט מופיעה האות $\dot{w}$ פירוש הדבר שהראש הקורא של אחד הסרטים נמצא מעליה. אז נגדיר $\Gamma'=\Gamma\cup\{\#\}\cup\dot{\Gamma}$, ואז אם בסרט ה-$i$ של M רשום $w_1^i...w_{n_i}^i$ כאשר הראש הקורא מעל $w_{h_i}^i$ אז בסרט היחיד של M יהיה רשום $\_\# w_1^1...\dot{w}_{h_1}^1...w_{n_1}^1\_\#...\#w_1^k...\dot{w}_{h_k}^k...w_{n_k}^k$. אז בתחילת הריצה הסרט ייראה כך: $\$\#\_\#...\#\_\#\_ w_{n_1}^1...w_{h_1}^1...\dot{w}_1^1$.

כדי לדמות צעד אחד של M, M' סורקת את הסרט משמאל לימין כדי לקבוע מהם הסימנים שנמצאים מתחת לראשים הקוראים. לאחר מכן M' עוברת עוד פעם על הסרט כדי לעדכן אותו בהתאם לפונקציית המעברים. אם באיזשהו שלב איזשהו ראש קורא עובר להצביע על $\#$ פירוש הדבר שב-M הוא היה אמור להצביע על תא שעוד לא נעשה בו שימוש קודם לכן. אז M' כותבת _ במקום הרלוונטי ומעבירה את כל תכולת הסרט מנקודה זו והלאה תא אחד ימינה.

ככה M' יכולה לבצע את כל מה ש- M יכולה לבצע.

נבחן כעת את המחיר של התהליך הזה. נניח ש- M עוצרת על קלט w תוך T צעדים. M' תעצור על אותו קלט תוך מספר הצעדים המקורי $\times$ מספר הצעדים שנדרש לדימוי צעד מקורי יחיד. מאחר שמספר הצעדים הוא T האורכים של הסרטים חסומים ע"י $kT+n$. התוספת של n היא למקרה שהמכונה לא קראה את כל הקלט. אבל מאחר שזה לא קורה בד"כ נוכל להניח שהאורכים חסומים ע"י kT. לכן מספר הצעדים לסימולציה של צעד מקורי היא $O(kT)$. סה"כ M' תעצור אחרי $O(kT^2)$ צעדים, וזה לא רע בכלל!

☺

משפט: לכל מטא"ד קיימת מט"ד שקולה.

הוכחה: בהינתן מטא"ד M נראה כיצד לבנות מט"ד D שקולה עם שלושה סרטים, ואז לפי המשפט הקודם נקבל את הדרוש. הרעיון הוא ש- D תנסה את כל האפשרויות עבור ריצות של M. אם באיזשהו שלב היא תגיע למצב מקבל אז D תקבל, אחרת הריצה של D לא תיגמר אף פעם.

ל- D יהיו שלושה סרטים: סרט קלט שלא כותבים עליו אלא רק קוראים ממנו, סרט עבודה שעליו יש העתק של הסרט של M עבור איזשהו ענף של עץ הריצה הלא דטרמיניסטי וסרט ניחוש ששומר את המקום של D בעץ הריצה של M על w.

יהי b הפיצול המקסימלי של פונקציית המעברים הלא דטרמיניסטית של M. אזי לכל צומת בעץ הריצה יש לכל היותר b בנים. לכל צומת בעץ ניתן כתובת שהיא מילה מעל $\Sigma_b=\{1,2,...,b\}$. הכתובת $i_1...i_k$ שייכת לצומת שמגיעים אליה כך: מהשורש עוברים לבן ה- i_1, ממנו עוברים לבן ה- i_2 וכן הלאה עד שהצומת שהגענו אליו ע"י $i_1...i_{k-1}$ עוברים לבן ה- i_k שלו. סרט הניחוש מכיל מילה מעל Σ_b כאשר המילה הריקה מתאימה לכתובת של שורש עץ הריצה.

כעת נתאר את פעולת D:

1. בהתחלה סרט הקלט מכיל את הקלט ושני הסרטים האחרים ריקים.
2. מעתיקים את סרט הקלט לסרט העבודה.
3. משתמשים בסרט העבודה כדי לסמלץ את פעולת M על w בענף שרשום בסרט הניחוש. בכל שלב פועלים לפי ההנחיות בסרט הניחוש. אם אין עוד תווים בסרט הניחוש או שסרט הניחוש מייצג כתובת לא חוקית, עוברים לשלב הבא. אם מגיעים למצב דוחה עוברים לשלב הבא. אם מגיעים למצב מקבל, מקבלים את המילה.
4. מחליפים את הכתובת בסרט הניחוש בכתובת הבאה בסדר הלקסיקוגרפי ועוברים לשלב 2.

ברור ש- $L(D)=L(M)$.

כעת, נבחן את היעילות. נניח שיש ל- M ריצה מינימאלית (כלומר קצרה ביותר) שמקבלת את w תוך T צעדים. כדי לקבל את המילה ב- D יש לעבור על כל עץ הריצה עד עומק T. כל ענף כזה הוא באורך לכל היותר T ויש לכל היותר b^T ענפים כאלה. לכן זמן העבודה של D יהיה $O(b^T)$.

☺

הערה: בסימונים של הוכחת המשפט, היינו רוצים שאם M מכריעה אז גם D מכריעה. זה מצריך תוספת קטנה ל- D. נוסיף סרט רביעי שכתוב בו אם יש המשך לרמה שאנחנו נמצאים בה עכשיו. הסרט מתאפס בכל פעם שיורדים רמה. אם הגענו למצב שאנחנו צריכים לרדת רמה אבל בסרט הרביעי כתוב שאין המשך, סימן שכל הריצות האפשריות על w הן דוחות ולכן D דוחה את w.

הגדרה: **ספרן** E הוא מ"ט עם מדפסת. המ"ט משתמשת במדפסת כאמצעי פלט ומדפיסה במדפסת מילים. שפת הספרן $L(E)$ היא כל המילים אשר בסופו של דבר יודפסו במדפסת.

משפט: $L \in RE$ אמ"מ קיים ספרן E כך ש- $L(E)=L$.

הוכחה:

($\Leftarrow$) נניח ש- M מ"א שמזהה את L. ידוע ש- Σ^* בת מניה ולכן ניתן לכתוב $\Sigma^*=\{w_0,w_1,\ldots\}$.
נגדיר ספרן E אשר מדפיס את L באופן הבא:

א. נאתחל $i=0$.
ב. נאתחל $j=0$.
ג. נריץ את M על w_j i צעדים.
ד. אם בשלב הקודם הגענו למצב מקבל נדפיס את w_j במדפסת.
ה. אם $i=j$ נגדיל את i ב- 1 ונחזור לשלב ב'.
ו. אחרת נגדיל את j ב- 1 ונחזור לשלב ג'.

ברור שהספרן ידפיס בדיוק את המילים שמתקבלות ע"י M, שהרי אם יש מילה w ב- $L(M)$ אז היא מתקבלת אחרי מספר סופי של צעדים ובסופו של דבר האלגוריתם שלנו יריץ את M על w כמספר הצעדים הדרוש. ואילו אם $w \notin L(M)$ היא לעולם לא תודפס משום שאנחנו מדפיסים רק אם M הגיעה למצב מקבל.

($\Rightarrow$) נניח שקיים ספרן E כך ש- $L(E)=L$ נראה שקיימת מ"ט M כך ש- $L(M)=L$. בהינתן w M תפעיל את E וכל פעם שהוא ידפיס מילה היא תבדוק אם המילה שהדפיס שווה ל- w. אם כן, M תקבל ואחרת תמשיך להריץ את E. ברור ש- M מזהה את L.

☺

התזה של צ'רץ' וטיורינג

התזה של צ'רץ' וטיורינג: כל פונקציה, אשר הייתה נחשבת באופן טבעי כחשיבה, ניתנת לחישוב במכונת טיורינג.

התזה יומרנית למדי והיא אומרת שכל פונקציה באשר היא, עבור כל שיטת חישוב בעבר, בהווה ובעתיד, ניתן לחשב באמצעות מכונת טיורינג. אך זוהי רק השערה. ניתן להוכיח גרסה פחות יומרנית.

הגדרה: **מכונת** RAM היא מודל חישובי שיש בו כמה מרכיבים:

- אמצעי קלט
- זיכרון
- רגיסטר בשם אקומולאטור
- קוד של תכנית שמבצעת המכונה

- רגיסטר בשם IP אשר מצביע לפקודות בתכנית

הקוד מורכב מפקודות מהצורה $operator\ [operand]$. פקודות יכולות להיות **ישירות** ו**עקיפות**.
נסמן ב- $c(i)$ את תוכן התא ה- i בזיכרון. פקודות ישירות הן מהצורה $operator\ i$ כאשר i מספר כלשהו או $operator\ c(i)$. פקודות עקיפות הן מהצורה $operator\ c(c(i))$.

הפקודות האפשריות הן:

- Load n - הכנס את n לאקומולאטור
- Store i - הכנס את ערך האקומולאטור לתא שמספרו i
- Add n - הוסף n לאקומולאטור
- Sub n - החסר n מהאקומולאטור ואם התוצאה שלילית אפס אותו
- Jump k - שנה את ערך ה- IP ל- k
- If=0 jump k - שנה את ערך ה- IP ל- k אם ערך האקומולאטור הוא 0
- If>0 jump k - שנה את ערך ה- IP ל- k אם ערך האקומולאטור גדול מ- 0
- Read – הכנס תו של קלט לאקומולאטור
- Accept – קבל את הקלט
- Reject – דחה את הקלט

כל הפקודות שמקבלות אופרנדים יכולות להיות גם ישירות וגם עקיפות. אם האופרנד הוא מספר נסמן זאת ע"י $=n$, אם האופרנד הוא ערך של תא בזיכרון נסמן $c(i)$ ואם האופרנד הוא מצביע לתא בזיכרון נסמן ע"י $*k$.

המכונה מתחילה את החישוב שלה כאשר כל הזיכרון מאותחל לאפס ו- IP מצביע לפקודה הראשונה בתכנית. המכונה מקבלת מילה אם הריצה שלה מגיעה לפקודת accept ולא מקבלת מילה אם היא נכנסת ללולאה אינסופית או מגיעה לפקודת reject.

משפט: מכונת RAM ומכונת טיורינג הם מודלים שקולים

הוכחה: ברור שמכונת RAM שהיא בעיקרה מחשב יכולה לדמות מ"ט. נראה שמ"ט יכולה לדמות מכונת RAM.

קונפיגורציה של מכונת RAM ניתן לייצג במ"ט עם 5 סרטים:

1. סרט קלט
2. סרט התכנית וה- IP. הפקודה הנכחית מסומנת ע"י תו מיוחד כלשהו, למשל

$load$	$=$	1	7	$\#$	$\widetilde{add}$	$*$	6	$\#$	...

3. סרט האקומולאטור
4. סרט הזיכרון, למשל

(	1	,	2	)	(6	,	1	7	3	)			...

5. סרט עבודה

נראה איך בהינתן קונפיגורציה של מכונת RAM אשר מיוצגת במ"ט ניתן לעבור לקונפיגורציה עוקבת. זאת בעצם תהיה סימולציה של מכונת RAM על מ"ט.

לא נעבור על כל הפקודות. הן כולם דומות אחת לשנייה. נראה, למשל, איך מבצעים את $32*$ add:

1. רצים על סרט הפקודות עד שמגיעים לפקודה עם שמסומנת כפקודה הנכחית.
2. כותבים $32*$ על סרט עזר.
3. רצים על הזיכרון ומחפשים ערך של תא 32
   א. אם מצאנו מעתיקים את תוכן התא $c(32)$ לסרט עזר
   ב. אם לא מצאנו מעתיקים 0 לסרט העזר
4. בצורה דומה מחפשים את ערך התא שמספרו כתוב על סרט העזר
5. מוסיפים את מה שכתוב בסרט העזר למה שכתוב באקומולאטור
6. מעבירים את הסימון של פקודה הנכחית לפקודה הבאה בתור

באופן דומה ניתן לסמלץ את כל הפקודות של המכונה. נשארת רק השאלה של המחיר. אם בתא i כתוב j מחיר השמירה שלהם הוא $\log i+\log j$. הזיכרון שנשמר בקונפיגורציה הוא פרופורציוני לסכום מחירי התאים בקונפיגורציית מכונת ה- RAM. לכן עבור מכונה שרצה בזמן T, אם סכום המחירים המקסימלי הוא p אז זמן הריצה של המ"ט השקולה פרופורציוני ל- $T\cdot p$. אבל למ"ט השקולה יש מספר סרטים ולכן המחיר הוא $(T\cdot p)^2$.

☺

מסקנה: כל אלגוריתם שמחשב בימינו יכול לבצע אפשר לממש בעזרת מ"ט.

אלגוריתם ניתן לתאר בשלוש רמות:

1. תיאור פורמאלי של מ"ט
2. תיאור בשפה עילית של פעולת המ"ט
3. תיאור אבסטרקטי של האלגוריתם בשפה טבעית

הבעיה היא שאלגוריתמים הם מעל גרפים, מטריצות וכו', בעוד שמ"ט הן מעל א"ב כלשהו Σ. אז אנחנו צריכים דרך לקודד את המידע לשפה פורמאלית, כדי שמ"ט יוכלו לעבוד איתו. למשל, גרף $G=\langle V,E\rangle$ ניתן לקודד באופן הבא: $\langle G\rangle=v_1\#...\#v_2\$\left(v_{i_1},v_{i_2}\right)\#...\#\left(v_{i_k},v_{i_n}\right)$. אפשר גם לקודד מ"ט אחרת, למשל כך: $\langle M\rangle=q_0\#...\#q_n\$\sigma_1\#...\#\sigma_m\$\gamma_1\#...\#\gamma_m\$...\$q_0\$q_{acc}\$q_{rej}$. הקידודים האלה הם קבועים והתחביר שלהם נתון מראש והמ"ט שפועלת עליהם יודעת את חוקי התחביר. כך, הדבר הראשון שיש לעשות הוא לעבור על הקלט ולבדוק אם הוא חוקי ואכן מייצג גרף או מ"ט או כל מבנה אחר שהמכונה מצפה לו.

דוגמה: נתאר מ"ט שמכריעה אם גרף לא מכוון הוא קשיר. ברמה האבסטרקטית היינו אומרים שצריך לבחור קודקוד, לבצע ממנו BFS ולבדוק אם כיסינו את שאר הקודקודים. בעקבות המסקנה הקודמת ברור שזהו תיאור לגיטימי. אבל לא קשה גם לאתר את זה ביותר פירוט.

מ"ט M שמכריעה את הבעיה תפעל כך: בהינתן קלט $\langle G\rangle$:

- אם מילת הקלט אינה מקודדת גרף נדחה אותה.
- נבחר קודקוד v ונסמן אותו (למשל, נחליף את האות הראשונה σ בקידוד שלו ב- $\dot{\sigma}$).
- כל עוד יש קודקודים לא מסומנים חדשים:
  - o לכל קודקוד ב- V נסמן אותו אמ"מ יש קשת בינו ובין קודקוד שכבר מסומן.
- נעבור על כל קודקודי G.
  - o אם כולם מסומנים נקבל את המילה.
  - o אחרת נדחה את המילה.

הגדרה: **מכונת טיורינג אוניברסאלית** U היא מ"ט שמקבלת כקלט קידוד של מ"ט M ומילה w ומסמלצת ריצה של M על w. ל- U יש שני סרטים: סרט קלט וסרט הסימולציה שרשומה בו הקונפיגורציה הנכחית של ריצת M על w. בהינתן $\langle M\rangle, w$ המכונה U כותבת q_0w על סרט הסימולציה. בכל שלב U קוראת אתה מצב הנכחי q והאות a שהראש הקורא מצביע עליה, מחפשת את $\delta(q,a)$ בסרט הקלט ומעדכנת את הקונפיגורציה בהתאם. אם M מגיעה ל- q_{acc} או ל- q_{reej} U מקבלת או דוחה את $\langle M\rangle, w$ בהתאמה. אם M לא עוצרת אז U לא עוצרת.

כריעות

עד כה עסקנו בשפות פורמאליות שלא קשורות כל כך לעולם האמיתי. בעקבות התזה של צ'רץ וטיורינג היינו רוצים להשתמש במכונות טיורינג כדי לתאר בעיות שיותר קשורות לחיי היום יום.

טענה: L_{DFA} שפת המילים מהצורה $\langle A\rangle, w$ כאשר $\langle A\rangle$ קידוד של אס"ד ו- $w\in L(A)$ היא כריעה.

הוכחה: הרעיון הוא לסמלץ את ריצת A על w. נתאר מ"ט M אשר מכריעה את הבעיה. למכונה יהיו שלושה סרטים: סרט קלט, סרט שכתוב בו המצב שבו אנחנו נמצאים וסרט שכתוב בו המקום בו אנחנו נמצאים בקריאת w. בסרט הקלט אנחנו נשתמש כדי לקבל את האות הבאה ולקבלת את פונקציית המעברים. בכל שלב נקרא את האות הבאה במילה ונבדוק בפונקציית המעברים לאן צריך להתקדם. אם הסימולציה הסתיימה במצב מקבל סימן ש- $\langle A\rangle, w\in L_{DFA}$, אחרת $\langle A\rangle, w\notin L_{DFA}$.

☺

טענה: L_{NFA} שפת המילים מהצורה $\langle A\rangle, w$ כאשר $\langle A\rangle$ קידוד של אס"ל ו- $w\in L(A)$ היא כריעה.

הוכחה: ראינו אלגוריתם לחירצון אס"ל ובטענה הקודמת ראינו כיצד לבדוק שייכות של מילה לאס"ד. אז נוכל לבנות מ"ט שבהינתן אס"ל A תייצר אס"ד שקול A' ולאחר מכן תבדוק אם $w\in L(A')=L(A)$.

☺

משפט: יש שפה שאינה נל"ר (ופרט אינה כריעה).

הוכחה: ניתן להניח בה"כ שהשפה היא מעל $\{0,1\}$ משום שכל שפה אחרת ניתן לקודד בבינארית. נשים לב שקבוצת כל השפות מעל $\{0,1\}$ היא אינה בת מניה. אבל כל מ"ט מתוארת ע"י רצף סופי של סימנים ולכן יש מספר בן מניה של מ"ט. מכאן שיש שפה שלא ניתן לזהות ע"י מ"ט.

☺

טענה: A_{TM} שפת כל המילים מהצורה $\langle M\rangle, w$ כאשר $\langle M\rangle$ קידוד של מ"ט ו- $w\in L(M)$ נל"ר.

הוכחה: נתאר מ"ט T אשר מזהה את A_{TM}. בהינתן קלט $\langle M\rangle, w$ T תפעל באופן הבא:

- תבדוק האם $\langle M\rangle$ קידוד חוקי של מכונת טיורינג. אם לא, תדחה.
- תסמלץ את ריצת M על w.
- אם באיזשהו שלב הסימולציה תגיע למצב מקבל, T תקבל את המילה.

T אכן מזהה את A_{TM}, שהרי אם $w\in L(M)$ אז קיימת ריצה סופית של M על w שמגיעה למצב מקבל, וכאשר הסימולציה תגיע למצב הזה היא תקבל את המילה. אם $w\notin L(M)$, הריצה של M על w לעולם לא תגיע למצב מקבל, ולכן T לא תקבל.

☺

טענה: $A_{TM}\notin R$

הוכחה: נניח בשלילה שיש מ"ט H שמקבלת כקלט מילה מהצורה $\langle M\rangle, w$ ומקיימת:

$$H\left(\langle M\rangle, w\right)=\begin{cases} accept & w\in L(M) \\ reject & else \end{cases}$$

נשתמש ב- H כדי לבנות מ"ט חדשה D אשר מקבלת כלקט תיאור של מ"ט M ופועלת כך:

$$D\left(\langle M\rangle\right)=\begin{cases} accept & M\left(\langle M\rangle\right)=accept \\ reject & else \end{cases}$$

D מוגדרת היטב משום שהנחנו ש- H עוצרת על כל הקלטים, ולכן כדי לקבל את $D\left(\langle M\rangle\right)$ פשוט נריץ את $H\left(\langle M\rangle,\langle M\rangle\right)$. וכעת ניתן להגדיר מ"ט D' שמקבלת קלט קידוד של מ"ט ופועלת כך:

$$D'\left(\langle M\rangle\right)=\begin{cases} reject & M\left(\langle M\rangle\right)=accept \\ accept & else \end{cases}$$

כעת נתבונן בריצה של D' על $\langle D' \rangle$. מתקבל:

$$D'(\langle D' \rangle) = \begin{cases} reject & D'(\langle D' \rangle) = accept \\ accept & else \end{cases}$$

וזאת סתירה. לכן לא קיימת מ"ט H שמכריעה את A_{TM}.

☺

מסקנה: $A_{TM}^c \notin RE$

הוכחה: אחרת היינו מקבלים ש- $A_{TM} \in co-RE$, ומאחר ש- $A_{TM} \in RE$ היה מתקבל $A_{TM} \in R$, בסתירה למה שהוכחנו.

☺

טענה: $HALT_{TM}$ שפת כל המילים מהצורה $\langle M \rangle, w$ כאשר $\langle M \rangle$ קידוד של מ"ט ו- M עוצרת על w היא נל"ר.

הוכחה: פשוט נריץ את M על w. אם היא עוצרת נקבל את $\langle M \rangle, w$, אחרת המכונה לא תעצור.

☺

משפט: $HALT_{TM} \notin R$

הוכחה: נניח בשלילה שיש מ"ט H שמכריעה את $HALT_{TM}$. נייצר מ"ט S שמכריעה את A_{TM}.
בהינתן קלט $\langle M \rangle, w$ S תפעל באופן הבא:

- תריץ את H על $\langle M \rangle, w$. אם H דוחה סימן ש- M לא עוצרת בריצה על w ולכן $w \notin L(M)$, כלומר $\langle M \rangle, w \notin A_{TM}$.
- אם H מקבלת, סימן ש- M עוצרת בריצה על w. לכן נוכל להריץ את M על w ומובטח שהריצה תעצור.
- אם הריצה נגמרת במצב מקבל סימן ש- $\langle M \rangle, w \in A_{TM}$.
- אחרת, $\langle M \rangle, w \notin L_{TM}$.

ברור ש- S אכן מכריעה את A_{TM} וזאת סתירה לכך ש- $A_{TM} \notin R$.

☺

מסקנה: $HALT_{TM}^{c} \notin RE$

הוכחה: ראינו ש- $HALT_{TM} \in RE$. אילו $HALT_{TM}$ הייתה גם ב- $co-RE$ היינון מקבלים ש- $HALT_{TM} \in R$, בסתירה למשפט הקודם.

☺

משפט: $Regular = \{\langle M \rangle : L(M) \in REG\}$ אינה כריעה.

הוכחה: נניח בשלילה שקיימת מ"ט H שמכריעה את $Regular$, כלומר

$$H\left(\langle M \rangle\right) = \begin{cases} acceept & L(M) \in REG \\ reject & else \end{cases}$$

נראה שקיימת מ"ט שמכריעה את A_{TM} ונקבל סתירה. בהינתן זוג $\langle M \rangle, w$ נוכל לייצר מ"ט M' אשר בהינתן קלט x פועלת כך:

- אם $x \in \{0^n 1^n : 0 \le n\}$ M' מקבלת את x.
- אחרת, M' מריצה את M על w ומקבלת את x אמ"מ M מקבלת את w.

נשים לב שאם M לא מקבלת את w אז $L(M') = \{0^n 1^n : 0 \le n\} \notin REG$ ואם M מקבלת את w אז $L(M') = (0+1)^* \in REG$.

כעת אנחנו יכולים להגדיר מ"ט S שמכריעה את A_{TM}. בהינתן קלט $\langle M \rangle, w$ S פועלת כך:

- מייצרת את M' כפי שהוגדרה לעיל ותלויה ב- M וב- w (נשים לב שייצור המכונה הוא תהליך סופי).
- מריצה את H על M'.
  - אם H מקבלת אז S מקבלת
  - אם H דוחה אז S דוחה

S אכן מכריעה את A_{TM}, שהרי אם S קיבלה, סימן ש- H קיבלה את M', כלומר $L(M') \in REG$ ולכן $w \in L(M)$. אם S דחתה אז H דחתה את M', כלומר $L(M') \notin REG$ ו- $w \notin L(M)$. אז S מזהה את A_{TM}. אבל S תמיד עוצרת משום ש- H תמיד עוצרת. ולכן S מכריעה את A_{TM}.

☺

רדוקציה

הגדרות

1. $f:\Sigma^* \to \Sigma^*$ נקראת **פונקציה ניתנת לחישוב** או **פונקציה חשיבה** אם קיימת מ"ט M_f כך שלכל קלט $w \in \Sigma^*$ M_f עוצרת ועל הסרט כתוב $f(w)$.
2. נאמר ששפה $A \subset \Sigma^*$ **ניתנת לרדוקציות מיפוי** לשפה $B \subset \Sigma^*$ אם קיימת פונקציה חשיבה $f:\Sigma^* \to \Sigma^*$ כך שלכל $w \in \Sigma^*$ מתקיים $w \in A \leftrightarrow f(w) \in B$. במקרה זה f נקראת **רדוקציה** מ-A ל-B ומסמנים $A \le_m B$.

משפט:

1. אם $A \le_m B$ ו-$B \in R$ אז $A \in R$.
2. אם $A \le_m B$ ו-$B \in RE$ אז $A \in RE$.

הוכחה: נוכיח רק את (1). (2) מתקבל באותו אופן.

תהי M_B מ"ט שמכירה את B ותהי M_f מ"ט שמחשבת את $f:\Sigma^* \to \Sigma^*$ הרדוקציה מ-A ל-B.
נבנה מ"ט M שמכריעה את A. בהינתן קלט w, M תפעל באופן הבא:

- מריצה את M_f על w.
- מריצה את M_B על תוצאת השלב הקודם.
- אם M_B קיבלה אז M מקבלת.
- אם M_B דחתה אז M דוחה.

M אכן מכריעה את A משום ש-M_B עוצרת תמיד ולכן M עוצרת תמיד, ומתקיים $w \in A \leftrightarrow f(w) \in B$.

☺

מסקנה:

1. אם $A \le_m B$ ו-$A \notin R$ אז $B \notin R$.
2. אם $A \le_m B$ ו-$A \notin RE$ אז $B \notin RE$.

הוכחה: נוכיח רק את (1). (2) מתקבל באותו אופן.

אילו היה מתקיים $B \in R$ אז לפי המשפט הקודם היינו מקבלים $A \in R$, בסתירה להנחה.

☺

דוגמאות:

1. כדי להוכיח ש- $A_{TM} \notin R$ איינו יכולים גם להשתמש ברדוקציה. ראינו כבר ש- $HALT_{TM} \notin R$ ולכן מספיק להראות ש- $A_{TM} \leq_m HALT_{TM}$.

בהינתן קלט $\langle M \rangle, w$ ל- A_{TM} עלינו לייצר קלט $f\left(\langle M \rangle, w\right)$ ל- $HALT_{TM}$ כך ש- $\langle M \rangle, w \in A_{TM} \leftrightarrow f\left(\langle M \rangle, w\right) \in HALT_{TM}$. עבור $\langle M \rangle, w$ M_f תחזיר $\langle M' \rangle, w'$ כאשר $w' = w$ ו-M' פועלת על קלט x באופן הבא:

- M' מריצה את M על x.
- אם M מקבלת את w גם M' מקבלת את x.
- אם M דוחה את x, M' נכנסת ללולאה אינסופית.

נראה ש- $\langle M \rangle, w \in A_{TM} \leftrightarrow f\left(\langle M \rangle, w\right) \in HALT_{TM}$:

$(\leftarrow)$אם $\langle M \rangle, w \notin A_{TM}$ אז $\langle M' \rangle, w \notin HALT_{TM}$ משום ש-$w \notin L(M)$ ובין אם M דוחה את w או לא עוצרת בכלל, M' נכנסת ללולאה אינסופית.

$(\rightarrow)$אם $\langle M \rangle, w \in A_{TM}$ אז $\langle M' \rangle, w \in HALT_{TM}$ משום ש- M מקבלת את w ולכן M' עוצרת.

2. נסתכל על $HALT_{TM}^{\varepsilon}$ שפת כל המילים מהצורה $\langle M \rangle$ כאשר M מ"ט שעוצרת על הקלט הריק ε אינה כריעה. ברור ש- $HALT_{TM}^{\varepsilon} \in RE$ שהרי בהינתן $\langle M \rangle$ נוכל פשוט להריץ את M על ε ולקבל אם היא עוצרת.

נראה ש- $HALT_{TM}^{\varepsilon}$ אינה כריעה ע"י הרדוקציה $HALT_{TM} \leq HALT_{TM}^{\varepsilon}$. יש להראות איך בהינתן קלט $\langle M \rangle, w$ ל- $HALT_{TM}$ ניתן לייצר קלט $f\left(\langle M \rangle, w\right)$ ל- $HALT_{TM}^{\varepsilon}$ כך ש- $\langle M \rangle, w \in HALT_{TM} \leftrightarrow f\left(\langle M \rangle, w\right) \in HALT_{TM}^{\varepsilon}$. נגדיר M_f שבהינתן $\langle M \rangle, w$ מייצרת מ"ט M' אשר פועלת כך: קודם כל היא כותבת את w על הסרט ולאחר מכן מריצה את M על הסרט.

נשים לב ש-M' פועלת על קלט ריק. כתיבת w על הסרט מובנית בהגדרה של M'. ברור ש- $\langle M' \rangle \in HALT_{TM}^{\varepsilon} \leftrightarrow \langle M \rangle, w \in HALT_{TM}$.

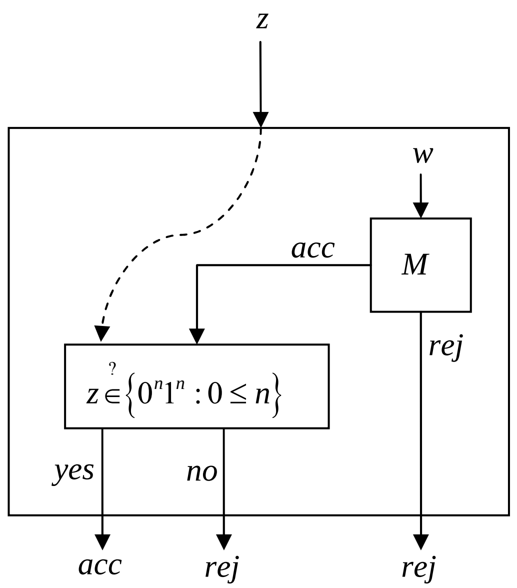
כעת, כמו בדוגמאות והמשפטים הקודמים, ברור גם ש- $HALT_{TM}^{\varepsilon} \notin co-RE$.

טענה: $Regular \notin RE \cup co-RE$.

הוכחה:

1. נראה ש- $Regular \notin RE$ ע"י הרדוקציה $A_{TM}^c \le Regular$.

בהינתן $\langle M \rangle, w$ נגדיר מ"ט M' אשר פועלת כל קלט z כך:



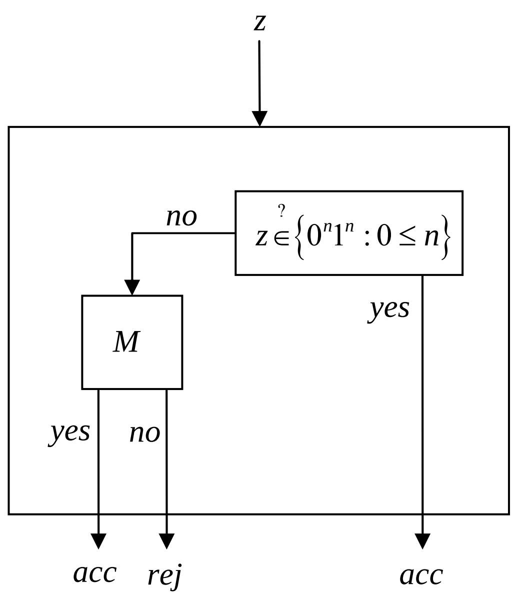
כעת, אם $w \in L(M)$ אז $L(M') = \{0^n1^n : 0 \le n\}$ והיא אינה רגולרית. ואם $w \notin L(M)$ אז יש שתי אפשרויות. אם M לא עוצרת על w אז M' לא עוצרת ואם M עוצרת אז M' דוחה. בכל אופן נקבל ש- $L(M') = \emptyset$ והיא רגולרית.

כלומר מצאנו פונקציה חשיבה $f : M \mapsto M'$ כך ש- $\langle M \rangle, w \in A_{TM}^c \leftrightarrow M' \in Regular$,

לכן $A_{TM}^c \le Regular$ ו- $Regular \notin RE$.

2. נראה ש- $Regular \notin co-RE$ ע"י הרדוקציה $A_{TM}^c \le Regular^c$.

בהינתן $\langle M \rangle, w$ נגדיר מ"ט M' אשר פועלת כל קלט z כך:



כעת, אם $w \in L(M)$ אז $L(M') = \Sigma^*$ והיא רגולרית. ואם $w \notin L(M)$ אז $L(M') = \{0^n 1^n : 0 \le n\}$ והיא אינה רגולרית.

כלומר מצאנו פונקציה חשיבה $f : M \mapsto M'$ כך ש- $\langle M \rangle, w \in A_{TM}^c \leftrightarrow M' \in Regular^c$,
לכן $A_{TM}^c \le Regular^c$ ו- $Regular \notin co-RE$.

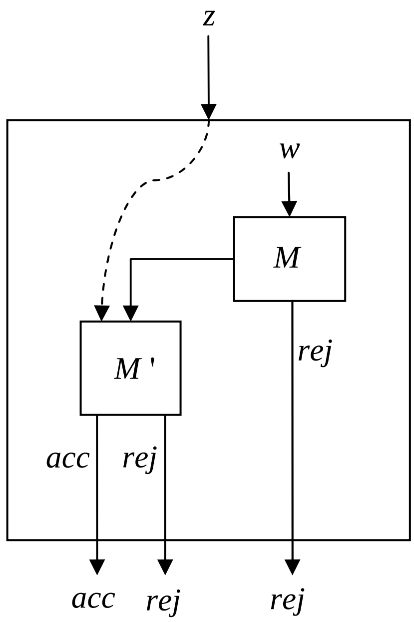
☺

הגדרה: תהי P קבוצה של מכונות טיורינג. P היא **תכונה סמנטית** אם לכל שתי מ"ט M_1, M_2 כך ש- $L(M_1) = L(M_2)$ מתקיים $M_1 \in P \leftrightarrow M_2 \in P$. במילים אחרות, P היא תכונה שתלויה אך ורק בשפה של המכונה.

משפט רייס: תהי P תכונה סמנטית שאינה ריקה ואינה כל המ"ט. אזי $L_P = \{\langle M \rangle : M \in P\}$ אינה כריעה.

הוכחה: תהי $M_\varnothing$ מ"ט כך ש- $L(M_\varnothing) = \varnothing$. בה"כ $M_\varnothing \in P$. אחרת, $M_\varnothing \in P^c$ אבל אם L_P^c כריעה גם L_P כריעה. לכן ניתן להניח שהתכונה הסמנטית מתקיימת גם ע"י $M_\varnothing$.

נוכיח ש- $L_p \notin RE$ ע"י הרדוקציה $A_{TM}^c \le_m L_p$. כיוון ש- P אינה כל המ"ט קיימת מ"ט M' כך ש- $M' \notin P$. בהינתן קלט $\langle M \rangle, w$ ל- A_{TM}^c נגדיר מ"ט T אשר בהינתן קלט z פועלת באופן הבא:



1. אם M דוחה את w אז T דוחה את z.
2. אחרת, T מפעילה את M' על z ומחזירה את התוצאה.

נראה ש- $T \in L_p \leftrightarrow \langle M \rangle, w \in A_{TM}^c$.

$(\leftarrow)$ נניח $\langle M \rangle, w \notin A_{TM}^c$. אזי $w \in L(M)$. לכן $L(T) = L(M')$ אבל $M' \notin P$ ו- P תכנה סמנטית ולכן $T \notin P$.

$(\rightarrow)$ נניח $\langle M \rangle, w \in A_{TM}^c$. אזי $w \notin L(M)$ ולכן $L(T) = \varnothing = L(M_\varnothing)$. לכן $T \in L_P$.

☺

מסקנה: כל תכונה סמנטית ש- $M_\varnothing$ מקיימת אותה היא איננה נל"ר.

הוכחה: באופן ישיר מההוכחה של משפט רייס.

☺

טענה: השפה $INF_{TM}=\{\langle M\rangle : L(M) \text{ is infinite}\}$ אינה נל"ר.

הערה: לא ניתן להוכיח את הטענה בעזרת משפט רייס משום ש- $M_\varnothing \notin INF_{TM}$.

הוכחה: נראה ש- $A_{TM}^c \le_m INF_{TM}$. בהינתן קלט $\langle M\rangle, w$ ל- A_{TM}^c נגדיר מ"ט M' שפועלת על קלט z באופן הבא:

1. M' מריצה את M על w $|z|$ צעדים.
2. אם M קיבלה, M' דוחה.
3. אחרת, M' מקבלת.

נראה ש- $M' \in INF_{TM} \leftrightarrow \langle M\rangle, w \in A_{TM}^c$:

$(\leftarrow)$ אם $\langle M\rangle, w \notin A_{TM}^c$ אז $w \in L(M)$ ולכן קיימת ריצה מקבלת של M על w באורך t. אז $L(M')=\{z:|z|<t\}$ והיא סופית, כלומר $M' \notin INF_{TM}$.

$(\rightarrow)$ אם $\langle M\rangle, w \in A_{TM}^c$ אז $w \notin L(M)$ ולכן לא קיימת ריצה מקבלת של M על w. בפרט אף ריצה באף אורך אינה מקבלת ולכן $L(M')=\Sigma^*$ והיא אינסופית, כלומר $M' \in INF_{TM}$.

☺

הגדרות:

1. בהינתן רביעייה $\langle T,H,V,T_{init}\rangle$ כאשר:

- $T=\{t_0,t_1,\ldots,t_k\}$ קבוצה סופית של אריחים
- $H,V \subset T\times T$ קבוצות סופיות של תנאים אנכיים ואופקיים בהתאמה
- $t_{init}\in T$ אריח התחלתי

ריצוף חוקי בגודל $n\times n$ הוא פונקציה $f:\{1,\ldots,n\}^2 \to T$ כך ש-

- $f(1,1)=t_{init}$
- לכל $1\le i\le n-1$ ולכל $1\le j\le n$ מתקיים $\big(f(i,j),f(i+1,j)\big)\in H$
- לכל $1\le i\le n$ ולכל $1\le j\le n-1$ מתקיים $\big(f(i,j),f(i,j+1)\big)\in V$

2. **בעיית הריצוף** $TILE$ היא קבוצת כל המילים מהצורה $\langle T,H,V,T_{init}\rangle$ כך שקיים ריצוף חוקי בגודל $n\times n$ לכל $1\le n$.

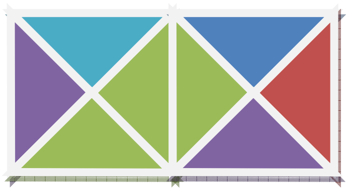
טענה: $TILE\in co-RE$

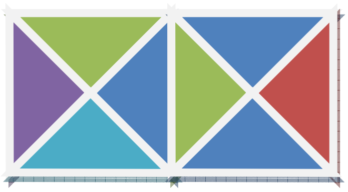
הוכחה: לכל $1\le n$ מספר הריצופים האפשריים הוא סופי. לכן, בהינתן $\langle T,H,V,T_{init}\rangle$ לכל n ניתן לבדוק אם קיים ריצוף חוקי. אם לא, $\langle T,H,V,T_{init}\rangle\notin TILE$, אחרת נעבור ל- n הבא. אם $\langle T,H,V,T_{init}\rangle\notin TILE$ בסופו של דבר האלגוריתם עצור. אם $\langle T,H,V,T_{init}\rangle\in TILE$ האלגוריתם לא יעצור אף פעם. בכל אופן, ניתן לזהות את $TILE^c$.

☺

משפט: $TILE\notin RE$

הוכחה: אנחנו לא נוכיח זאת אבל בעיית הריצוף כפי שהגדרנו אותה שקולה לקיום ריצוף חוקי לכל רבע המישור. כמו כן, נשים לב שאת בעיית הריצוף ניתן לייצג בצורה שונה. ניתן לדמיין את האריחים כמסומנים בכל צלע. ניתן להניח אריח אחד ליד אריח שני אם הצלעות שנוגעות מסומנות באותו סימון. למשל,


מצב חוקי

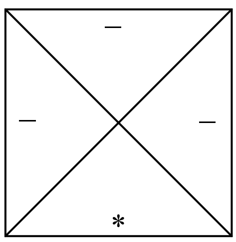

מצב לא חוקי

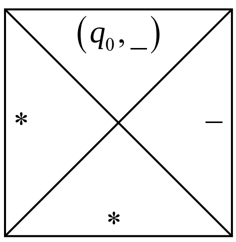
ראינו ש- $\left(HALT_{TM}^{\varepsilon}\right)^c\notin RE$, לכן מספיק להראות ש- $\left(HALT_{TM}^{\varepsilon}\right)^c\le_m TILE$.

יהי $\langle M\rangle$ קלט ל- $\left(HALT_{TM}^{\varepsilon}\right)^c$. נראה איך לבנות קלט $\langle T,V,H,t_{init}\rangle$ ל- $TILE$ כך ש- M לא עוצרת על הקלט הריק אמ"מ קיים ריצוף חוקי לכל המישור.

יהיו ארבעה סוגים של אריחים:

1. אריחים שימלאו את השורה הראשונה:

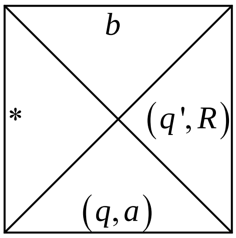




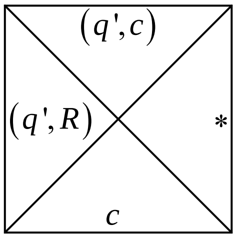
האריח ההתחלתי

2. אריחים שמתאימים לפונקציית המעברים:

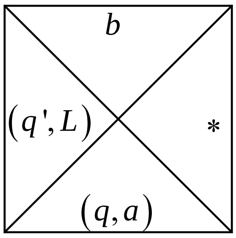
א. לכל מעבר $\delta(q,a)=(q',b,R)$ כאשר $q\notin\{q_{acc},q_{rej}\}$ נוסיף אריח:



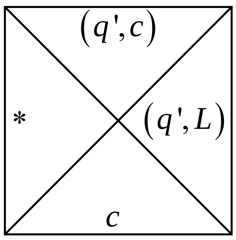
ולכל $c\in\Gamma$ נוסיף אריח:



ב. לכל מעבר $\delta(q,a)=(q',b,R)L$ כאשר $q\notin\{q_{acc},q_{rej}\}$ נוסיף אריח:

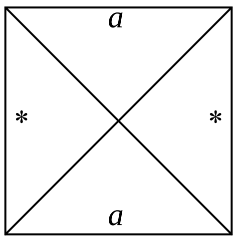


ולכל $c\in\Gamma$ נוסיף אריח:



סה"כ כל מעבר שאינו מאחד המצבים הסופיים מוסיף $|\Gamma|+1$ אריחים.

3. אריחי ריפוד: לכל $a\in\Gamma$ אריח מהצורה



בעזרת אריחים אלה ניתן לייצג קונפיגורציה של המכונה בריצה על הקלט הריק. ושתי שורות c_i,c_{i+1} הן בריצוף חוקי אמ"מ הן מתאימות לקונפיגורציות עוקבות.

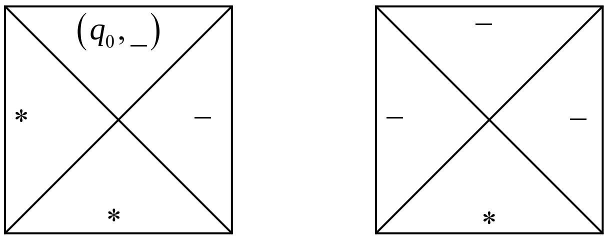
כעת, אם המכונה לא עוצרת סימן שיש לה אינסוף קונפיגורציות עוקבות ולכן ניתן לרצף את רבע המישור. ואילו אם ניתן לרצף את רבע המישור אז הריצוף מייצג אינסוף קונפיגורציות עוקבות, כלומר ריצה של המכונה אשר לא עוצרת.

☺

דוגמה: נסתכל על מ"ט שפונקציית המעברים שלה מוגדרת ע"י $\delta(q_0,\_)=(q_0,b,R)$ ו- $\delta(q_0,b)=(q_{acc},b,R)$. המכונה הזאת לא עוצרת על ε. היא פשוט תמלא את הסרט ברצף אינסופי של b-ים.

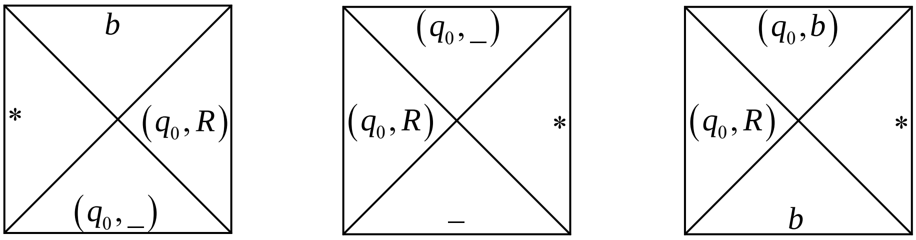
האריחים שמתאימים למ"ט זו הם:

1. שורה ראשונה:

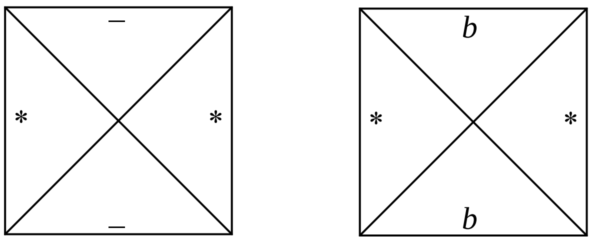


האריח ההתחלתי

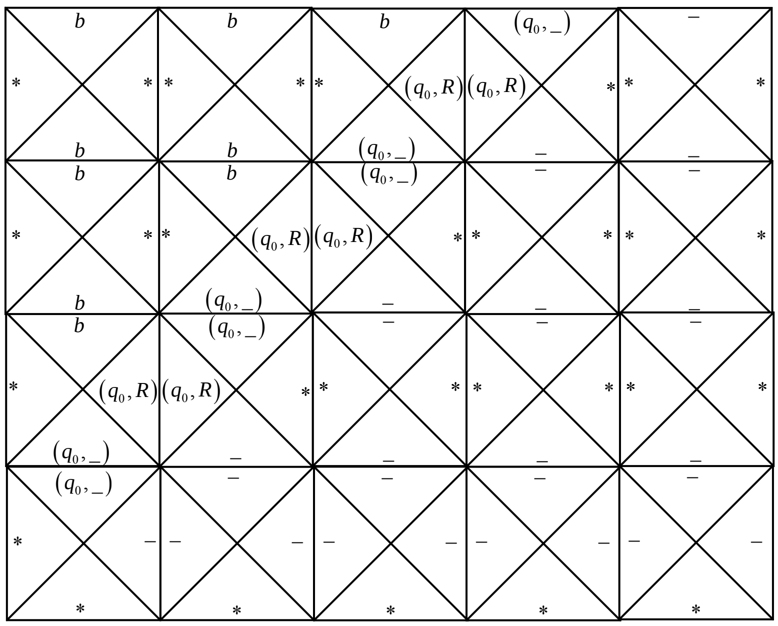
2. מעברים:



3. ריפוד:



בעזרת אריחים אלה ניתן לרצף את כל רבע המישור:



חלק שלישי

תורת הסיבוכיות

סיבוכיות זמן

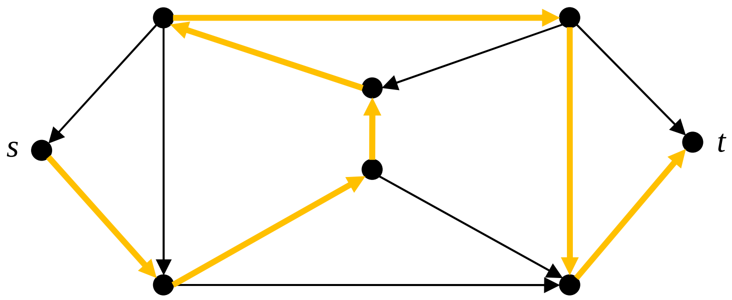
המחלקות P ו- NP

הגדרות:

1. $TIME\left(t\left(n\right)\right)$ היא מחלקת השפות אשר ניתנות להכרעה ע"י מט"ד בעלת סרט יחיד שרצה בזמן $O\left(t\left(n\right)\right)$.
2. $NTIME\left(t\left(n\right)\right)$ היא מחלקת השפות אשר ניתנות להכרעה ע"י מטא"ד בעלת סרט יחיד שרצה בזמן $O\left(t\left(n\right)\right)$.
3. $P=\bigcup_{k\in\mathbb{N}} TIME\left(n^k\right)$ היא מחלקת השפות אשר ניתנות להכרעה ע"י מט"ד בזמן פולינומיאלי.
4. $NP=\bigcup_{k\in\mathbb{N}} NTIME\left(n^k\right)$ היא מחלקת השפות אשר ניתנות להכרעה ע"י מטא"ד בזמן פולינומיאלי.

דוגמאות:

1. **מסלול המילטוני** בגרף מכוון הוא מסלול העובר דרך כל קודקודי הגרף פעם אחת בדיוק. למשל כמו בציור הבא:



$HAMPATH$ היא שפת המילים מהצורה $\langle G,s,t\rangle$ כאשר G מכוון ויש מסלול המילטוני מ- s ל- t.

יש אלגוריתם מעריכי פשוט להכרעת $HAMPATH$. פשוט נסרוק את כל המסלולים באורך $|V|$ שיוצאים מ- s ונבדוק אם המסלולים שמגיעים ל- t הם מסלולים המילטוניים. הבדיקה של כל מסלול היא פשוטה ולוקחת זמן פולינומיאלי. הצרה היא שיש מספר מעריכי של מסלולים שיוצאים מ- s.

לא ידוע כיום אם קיים אלגוריתם פולינומיאלית אשר מכריע את הבעיה!!

2. $COMPOSITE=\{x:(x=pq)\wedge(1<p,q)\}$ כאשר x נתון בייצוג בינארי.
גם במקרה זה הבעיה כריעה ע"י אלגוריתם מעריכי פשוט. נסרוק את כל המספרים $2,3,...,\sqrt{x}$ ונבדוק אם הם מחלקים את x. בגלל שהייצוג של x הוא בינארי סיבוכיות הזמן של האלגוריתם היא מעריכית. מצד שני, בהינתן p,q קל מאוד לבדוק בזמן פולינומיאלית אם אכן מתקיים $x=pq$ או לא.

הגדרה: **מוודא**[4] לשפה L הוא מ"ט A כך ש- $L=\{w:A\text{ accepts }\langle w,c\rangle\text{ for some word }c\}$.
במקרה זה c נקרא **עד**[5]. סיבוכיות המוודא נמדדת ביחס לאורך של w. **מוודא פולינומיאלי** רץ על $\langle w,c\rangle$ בזמן פולינומיאלית ב- $|w|$ לכל w (בפרט $|c|$ הוא פולינומיאלית ב- $|w|$).

דוגמאות:

1. מוודא עבור $HAMPATH$ הוא מ"ט A שהקלט שלה הוא $(\langle G,s,t\rangle,c)$ ו- A מקבל את $(\langle G,s,t\rangle,c)$ אמ"מ c הוא תיאור של מסלול המילטוני מ- s ל- t ב- G.
2. מוודא פולינומיאלי עבור $COMPOSITE$ הוא מ"ט A אשר מקבלת את $(\langle x,c=(p,q)\rangle)$ אמ"מ $x=pq$ ו- $1<p,q$.

טענה: $HAMPATH\in NP$

הוכחה: נראה מטא"ד M אשר מקבלת את $\langle G,s,t\rangle$ אמ"מ קיים מסלול המילטוני מ- s ל- t:

1. M מנחשת סדרה של קודקודים $p_1,...,p_n$ כאשר $n=|V|$.
2. אם יש קודקוד שמופיע פעמיים M דוחה.
3. אם $p_1\neq s$ או $p_n\neq t$ M דוחה.
4. אם יש $1\le i\le n-1$ כך שאין קשת מ- p_i ל- p_{i+1} אז M דוחה.
5. אחרת, M מקבלת.

ברור שכל ריצה של M עוצרת אחרי זמן פולינומיאלי ב- n:

- ניחוש n קודקודים - $O(n)$
- בדיקה אם יש קודקוד שמופיע פעמיים - $O(n^2)$
- בדיקה אם $p_1\neq s$ או $p_n\neq t$ - $O(n)$ (כי צריך לזוז על הסרט קדימה ואחורה)
- בדיקת הקשתות - $O(n^2)$

[4] verifier
[5] certificate

ברור גם ש- M מכריעה את הבעיה שהרי אם קיים מסלול המילטוני אז באחד הריצות M תנחש את המסלול ותקבל ואילו אם לא קיים מסלול המילטוני אז כל מסלול ש- M תנחש יידחה.

☺

משפט: $L \in NP$ אמ"מ קיים ל- L מוודא פולינומיאלי.

הוכחה:

$(\Leftarrow)$ נניח ש- $L \in NP$. אזי קיימת מטא"ד M כך ש- $L(M)=L$ ו- M רצה זמן פולינומיאלי באורך הקלט לכל קלט. לכן L היא למעשה שפת כל המילים אשר קיימת עבורן ריצה מקבלת של M. אבל מאחר שהריצות של M פולינומיאלית ב- $|w|$ גם הקידוד של הריצה המקבלת הוא פולינומיאלי והבדיקה היא כמובן פולינומיאלית.

$(\Rightarrow)$ נניח ש- $L=\{w: A \text{ accepts } \langle w,c\rangle \text{ for some word } c\}$. נתאר מטא"ד M אשר מכריעה את L. בהינתן w, M תנחש עד c ותבדוק אם $\langle w,c\rangle \in L(A)$.

☺

שלמות ב- NP

הגדרות:

1. נאמר ש- $f:\Sigma^* \to \Sigma^*$ **ניתנת לחישוב בזמן פולינומיאלי** אם יש מט"ד M העוצרת בזמן פולינומיאלי ומחשבת את f (כלומר, יש פולינום $t:\mathbb{N}\to\mathbb{N}$ כך שלכל w M עוצרת אחרי $t(|w|)$ צעדים).
2. תהיינה A,B שפות. נאמר ש- $f:\Sigma^* \to \Sigma^*$ היא **רדוקציה פולינומיאלית** מ- A ל- B אם היא ניתנת לחישוב בזמן פולינומיאלי ולכל $w\in\Sigma^*$ מתקיים $w\in A \leftrightarrow f(w)\in B$. במקרה זה נסמן $A\le_p B$.
3. שפה L היא NP**-קשה**[6] אם לכל שפה $L'\in NP$ מתקיים $L'\le_p L$.
4. שפה L היא NP**-שלמה**[7] אם $L\in NP$ ו- $L\in NP-hard$.

[6] $NP-hard$
[7] $NP-complete$

משפט: אם $A \le_p B$ ו- $B \in P$ אז $A \in P$.

הוכחה: תהי M_B מט"ד אשר מכריעה את B בזמן פולינומיאלי ותהי $f:\Sigma^* \to \Sigma^*$ רדוקציה פולינומיאלית מ- A ל- B. נתאר מט"ד M_A אשר בהינתן w מכריעה אם $w \in A$ או $w \notin A$.
ראשית, M_A תחשב את $f\left(w\right)$. מאחר שהחישוב לוקח זמן פולינומיאלי ב- $|w|$ גם $\left|f\left(w\right)\right|$ פולינומיאלי ב- $|w|$. כעת נשתמש ב- M_B כדי להכריע אם $f\left(w\right) \in B$ או $f\left(w\right) \in M_B$. בדיקה זו פולינומיאלית ב- $\left|f\left(w\right)\right|$ ולכן סה"כ M_A פועלת בזמן פולינומיאלי ב- $|w|$. ובגלל ש- f רדוקציה מ- A ל- B מתקיים $w \in A \leftrightarrow f\left(w\right) \in B$, כלומר M_A אכן מכריעה את A.

☺

משפט: אם $L \in NP-complete$ ו- $L \in P$ אז $P = NP$.

הוכחה: ברור ש- $P \subset NP$. יש להראות ש- $NP \subset P$. תהי $L' \in NP$. מהנתון $L' \le_p L$. אל $L \in P$ ולכן מהמשפט הקודם $L' \in P$. כלומר, $NP \subset P$.

☺

משפט: אם $L \in NP-hard$ ו- $L \le_p L'$ אז $L' \in NP-hard$.

הוכחה: תהי $L'' \in NP$. יש להראות ש- $L'' \le_p L'$. ידוע ש- $L'' \le_p L$ ו- $L \le_p L'$. נניח ש- f רדוקציה פולינומיאלית מ- L'' ל- L ו- g רדוקציה פולינומיאלית מ- L ל- L'. ברור ש- $g \circ f$ רדוקציה פולינומיאלית מ- L'' ל- L'.

☺

הגדרה: **בעיית הריצוף החסום** BT היא שפת המילים מהצורה $\left\langle T,V,H,t_{init},t_{fin},n\right\rangle$ כאשר n נתון באונארית וקיים ריצוף חוקי $f:\left\{1,...,n\right\}^2 \to T$ כך ש- $f\left(1,1\right)=t_{init}$ ו- $f\left(1,n\right)=t_{fin}$.

טענה: $BT \in NP-complete$

הוכחה:

1. נראה ש- $BT \in NP$:
נתאר מטא"ד M אשר מכריעה את BT בזמן פולינומיאלי. M תנחש ריצוף ותבדוק אם הוא מתאים. הבדיקה היא כמובן פולינומיאלית באורך הקלט. אבל נטען שגם תהליך הניחוש הוא פולינומיאלי. כדי לייצג אריח יחיד מ- T יש צורך ב- $\log|T|$ תאים ולכן ניתן לנחש אריח ב- $\log|T|$ צעדים. אנחנו צריכים לנחש n^2 אריחים ולכן זה לוקח $n^2 \log|T|$ תאים או

$n^2 \log|T|$ צעדים. ההנחה היא ש-n גדול מ-$|T|$, ולכן בגלל שאורך הקלט הוא $O(n)$ גם הניחוש לוקח זמן פולינומיאלי. כאן חושב להדגיש שהעבודה ש-n מיוצג באונארית היא אקוטית לפולינומיאליות של האלגוריתם.

2. נראה ש- $BT \in NP-hard$:

נראה שלכל $L \in NP$ מתקיים $L \le_p BT$. נגדיר שפה חדשה:

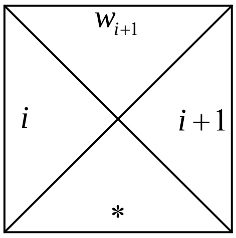
$$A_{B-TM} = \left\{ \begin{matrix} \langle M \rangle, w, 1^p : & M \text{ is a non deterministic Turing machine} \\ & \text{which accepts } w \text{ in p steps} \end{matrix} \right\}$$

ברור שלכל $L \in NP$ מתקיים $L \le_p A_{B-TM}$ כי אם $x \in L$ נעביר אותו ל- $\langle M \rangle, x, 1^{p(x)}$ כאשר $p(x)$ אורך הריצה של M על x.

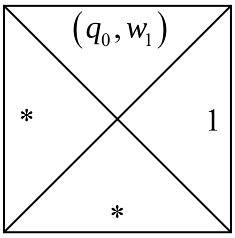
כעת נראה ש- $A_{B-TM} \le_p BT$. בהינתן $\langle M \rangle, w, 1^s$ נבנה בעיית ריצוף $BI = \langle T, H, V, t_0, t_k, 1^{s'} \rangle$ כך ש- $BI \in BT \leftrightarrow \langle M \rangle, w, 1^s \in A_{B-TM}$.

האריחים יהיו:

א. אריחים לשורה הראשונה:



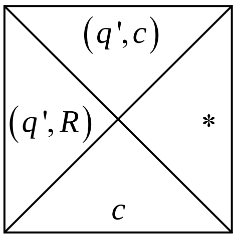
לכל $1 \le i < n$



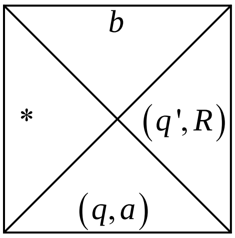
אריח התחלתי

ב. אריחים שמתאימים למעברים:

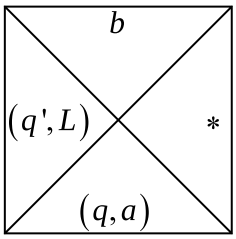
- לכל מעבר מהצורה $\delta(q,a) = (q',b,R)$

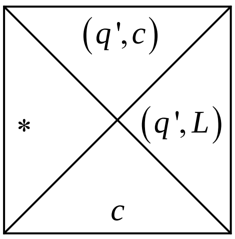


לכל $c \in \Gamma$



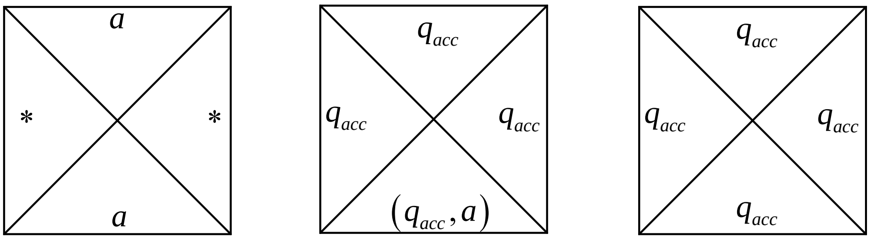
- לכל מעבר מהצורה $\delta(q,a) = (q',b,L)$





לכל $c \in \Gamma$

ג. אריחי ריפוד: לכל $a \in \Gamma$ אריח מהצורה



כעת, כמו בריצוף חסום ניתן לראות שאת האריחים הנ"ל ניתן לסדר בהתאם לריצה של M על w. כל שורה בריצוף תתאים לקונפיגורציה של M ושורת עוקבות הן קונפיגורציות עוקבות.

☺

הגדרות:

1. **משתנה בוליאני** הוא משנה אשר יכול לקבל ערכים רק מ- $\{true, false\}$ או מ- $\{1,0\}$ בהתאמה.
2. **פעולה בוליאנית** היא אחר מהפעולות $\neg, \vee, \wedge$.
3. **נוסחה בוליאנית** היא מתקבלת באופן רקורסיבי ע"י החוקים הבאים:
   i. 0 ו-1 הן נוסחאות בוליאניות
   ii. משתנה בוליאני הוא נוסה בוליאנית
   iii. אם φ, ψ נוסחאות בוליאניות אזי $\varphi \wedge \psi$, $\varphi \vee \psi$ ו-$\neg\varphi$ נוסחאות בוליאניות.
4. בהינתן נוסחה בוליאנית φ תהי Var קבוצת המשתנים שלה. **השמת אמת** ל-φ היא פונקציה $f : Var \to \{0,1\}$.
5. תהי φ נוסחה בוליאנית ותהי $f : Var \to \{0,1\}$ השמת אמת. נגדיר את **ערך האמת** של φ באופן רקורסיבי:
   i. אם $\varphi = 0$ או $\varphi = 1$ ערך האמת של φ הוא 0 או 1 בהתאמה.
   ii. אם $\varphi = x$ עבור $x \in Var$ אזי ערך האמת של φ הוא $f(x)$.
   iii. אם φ, ψ נוסחאות בוליאניות עם ערכי אמת $v(\varphi), v(\psi)$ בהתאמה
      - ערך האמת של $\neg\varphi$ הוא $1 - v(\varphi)$.
      - ערך האמת של $\varphi \wedge \psi$ הוא 1 אמ"מ $v(\varphi) = 1 = v(\psi)$, אחרת ערך האמת הוא 0.
      - ערך האמת של $\varphi \vee \psi$ הוא 0 אמ"מ $v(\varphi) = 0 = v(\psi)$, אחרת ערך האמת הוא 1.
6. נוסחה בוליאנית φ היא **ספיקה** אם קיימת השמת אמת f כך שערך האמת של φ הוא 1.
7. **בעיית הספיקות** היא $SAT = \{\langle\varphi\rangle : \varphi \text{ is satisfiable}\}$
8. **ליטרל** הוא משתנה או שלילתו
9. **פסוקית** היא נוסחה מהצורה $l_1 \vee ... \vee l_n$ כאשר $\{l_i\}$ ליטרלים.
10. נוסחה בוליאנית היא בצורת CNF אם היא מהצורה $\varphi_1 \wedge ... \wedge \varphi_n$ כאשר $\{\varphi_i\}$ פסוקיות.

11. נוסחה בוליאנית היא בצורה $nCNF$ אם היא בצורת CNF ובכל פסוקית מופיעים בדיוק n ליטרלים.
12. $nSAT = \left\{ \langle \varphi \rangle : \varphi \text{ is a satisfiable formula in } nCNF \text{ form} \right\}$

טענה: $3SAT \in NP-complete$

הוכחה:

1. בוודאי $3SAT \in NP$. בהינתן φ נוסחה ניתן פשוט לנחש הערכת אמת ולבדוק אם היא מספקת את φ. זה ייקח לכל היותר $O\left(\left|\langle\varphi\rangle\right|^2\right)$.
2. נראה ש- $BT \le_p 3SAT$ ונקבל ש- $3SAT \in NP-hard$.

בהינתן $\left\langle T,H,V,t_{init},t_{fin},n\right\rangle$ נראה שניתן לייצר בזמן פולינומיאלי נוסחה φ בצורת $3CNF$ כך ש- $\langle\varphi\rangle \in 3SAT \leftrightarrow \left\langle T,H,V,t_{init},t_{fin},n\right\rangle \in BT$.

משתני הנוסחה יהיו $x_{i,j,t}$ לכל $1 \le i,j \le n$ ולכל $t \in T$, סה"כ $n^2|T|$ משתנים. נגדיר:

א. לכל $1 \le i,j \le n$ $\varphi_{ij} = \underset{t\in T}{\vee} x_{i,j,t}$

ב. לכל $1 \le i,j \le n$ $\varphi'_{ij} = \underset{t\in T}{\wedge}\left(x_{i,j,t} \to \underset{t\ne t'}{\wedge}\left(\neg x_{i,j,t'}\right)\right)$

ג. $\varphi_{border} = x_{1,1,t_{init}} \wedge x_{1,n,f_{fin}}$

ד. לכל $1 \le i < n$ ולכל $1 \le j \le n$ $\theta_{ij} = \underset{t,t'\in H}{\vee}\left(x_{i,j,t} \wedge x_{i+1,j,t'}\right)$

ה. לכל $1 \le i \le n$ ולכל $1 \le j < n$ $\theta'_{ij} = \underset{t,t'\in H}{\vee}\left(x_{i,j,t} \wedge x_{i,j+1,t'}\right)$

כעת נגדיר $\varphi = \left(\underset{\substack{1\le i\le n\\1\le j\le n}}{\wedge}\left(\varphi_{ij} \wedge \varphi'_{\iota\varphi}\right)\right) \wedge \left(\varphi_{border}\right) \wedge \left(\underset{\substack{1\le i<n\\1\le j\le n}}{\wedge}\theta_{ij}\right) \wedge \left(\underset{\substack{1\le i\le n\\1\le j<n}}{\wedge}\theta'_{ij}\right)$.

נטען ש- $\left\langle T,H,V,t_{init},t_{fin},n\right\rangle \in BT \leftrightarrow \langle\varphi\rangle \in SAT$.

($\leftarrow$) נניח $\left\langle T,H,V,t_{init},t_{fin},n\right\rangle \in BT$. נגדיר השמת אמת $f:\left\{x_{i,j,t}\right\} \to \{0,1\}$ ע"י $f\left(x_{i,j,t}\right)=1$ אמ"מ בריצוף במקום ה-(i,j) נמצא אריח t. נטען ש- f מספקת את φ.

א. מאחר שבכל מקום (i,j) קיים לפחות אריח אחד t $v\left(\varphi_{ij}\right) = v\left(\underset{t\in T}{\vee} x_{i,j,t}\right) = 1$.

ב. מאחר שבכל מקום (i,j) קיים בדיוק אריח אחד t $v\left(\varphi'_{ij}\right) = v\left(\underset{t\in T}{\wedge}\left(x_{i,j,t} \to \underset{t\ne t'}{\wedge}\left(\neg x_{i,j,t'}\right)\right)\right) = 1$.

ג. מאחר שנתון שב-$(1,1)$ נמצא t_{init} ובמקום $(1,n)$ נמצא t_{fin} מתקיים $v\left(\varphi_{border}\right) = v\left(x_{1,1,t_{init}} \wedge x_{1,n,f_{fin}}\right) = 1$.

ד. מאחר שמתקיימים התנאים המאוזנים לכל $1\le i<n$ ולכל $1\le j\le n$

$$v\left(\theta_{ij}\right)=v\left(\bigvee_{t,t'\in H}\left(x_{i,j,t}\wedge x_{i+1,j,t'}\right)\right)=1.$$

ה. מארח שמתקיימים התנאים המאונכים לכל $1\le i\le n$ ולכל $1\le j<n$

$$v\left(\theta'_{ij}\right)=v\left(\bigvee_{t,t'\in H}\left(x_{i,j,t}\wedge x_{i,j+1,t'}\right)\right)=1.$$

לכן בסה"כ φ ספיקה ע"י f .

$(\rightarrow)$ כעת, אם φ ספיקה ע"י השמת אמת f אזי בתהליך הפוך לכיוון הקודם ניתן לבנות ריצוף חוקי.

עד כאן הראנו ש- $\langle\varphi\rangle\in SAT\leftrightarrow\langle T,H,V,t_{init},t_{fin},n\rangle\in BT$. הבעיה היא שכפי שהגדרנו את φ היא אינה בצורת $3CNF$. אבל קל לראות ש- φ שקולה לנוסחה $\psi\in 3CNF$ ואת המעבר ניתן לבצע בזמן פולינומיאלי.

☺

טענה: $HAMPATH\in NP-complete$

הוכחה:

1. ברור ש- $HAMPATH\in NP$ כי ניתן לנחש מסלול ולבדוק אם הוא מקיים את הדרישות.
2. נראה ש- $3SAT\le_p HAMPATH$ ונקבל ש- $HAMPATH\in NP-hard$ בהינתן נוסחה φ בצורת $3CNF$ נבנה גרף G ונגדיר קודקודים s,t בגרף כך שיתקיים $\varphi\in SAT\leftrightarrow\langle G,s,t\rangle\in HAMPATH$.

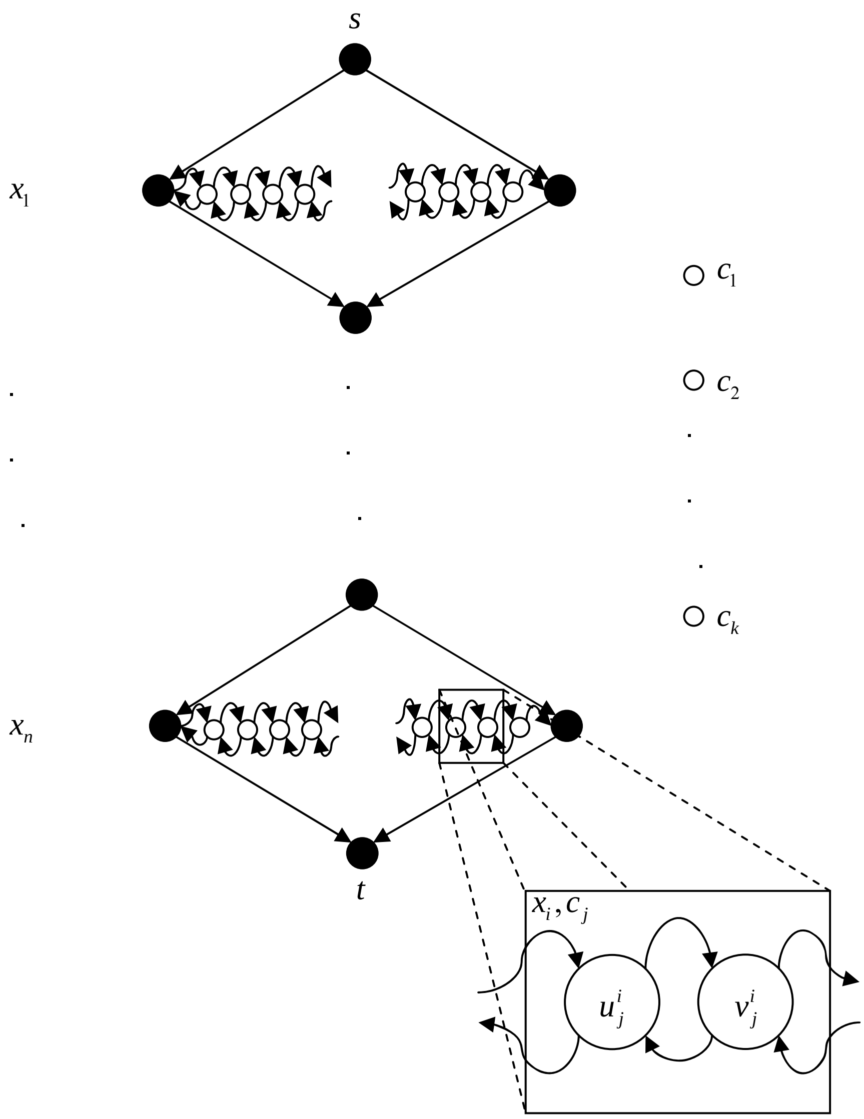
נניח ש- $\varphi=\bigwedge_{i=1}^{k}c_i$ כאשר $c_i=l_1^i\vee l_2^i\vee l_3^i$ והמשתנים הם $x_1,...,x_n$.

לכל משתנה x_i ב- φ נתאים מעוין כמו באיור ובו $3k+3$ קודקודים במאוזן: לכל פסוקית c_j נתאים זוג קודקודים שונים u_j^i,v_j^i ובין כל שני זוגות נציב קודקוד נוסף. בנוסף לכל פסוקית נוסיף קודקוד c_j.

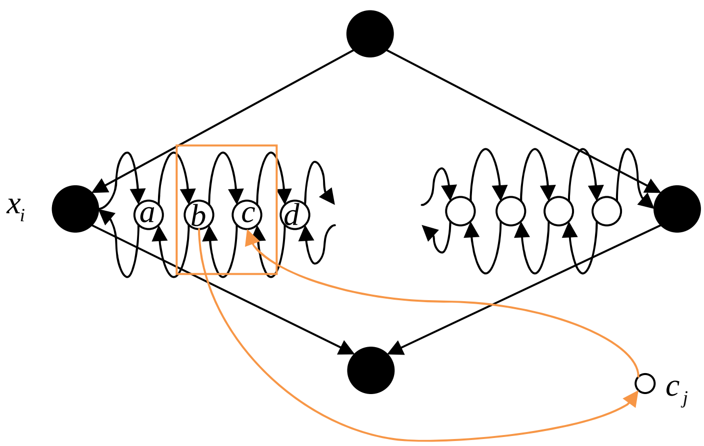
אם x_i מופיע ב- c_j נוסיף קשתות $\left(u_j^i,c_j\right)$ ו- $\left(c_j,v_j^i\right)$. אם $\neg x_i$ מופיע ב- c_j נוסיף קשתות $\left(v_j^i,c_j\right)$ ו- $\left(c_j,u_j^i\right)$.

ברור שהבנייה הזאת היא פולינומיאלית.

נניח ש- φ ספיקה ותהי $f:\{x_i\}\rightarrow\{0,1\}$ השמה מספקת. אזי יש מסלול המילטוני מ- s ל- t. המסלול עובר על כל המעוניינים בכיווניות שמתאימה להשמה: $f\left(x_i\right)=1$ אמ"מ המעבר על המעוין המתאים הוא משמאל לימין. אם פסוקית j מסתפקת ע"י משתנה x_i אז במהלך המעבר על קודקודי x_i אפשר לעבור גם על c_j (בגלל הצורה שבה בנינו את הגרף).



מצד שני, אם יש ב- G מסלול המילטוני מ- s ל- t נטען שהוא בהכרח מהצורה של המסלול שתיארנו קודם ואז גם ברור מהי ההשמה המספקת. אם המסלול הוא לא מהצורה הנ"ל פירוש הדבר שכאשר הוא הגיע לאיזה c_j הוא לא חזר ממנו מיד אלא המשיך למקום אחר.



כלומר, המסלול הגיע ל- b, המשיך ל- c_j אבל לא חזר ל- c. אבל המסלול הוא מילטוני ולכן בשלב מסוים הוא חייב לחזור ל- c. הדרכים היחידות לחזור ל- c הן מ- b, מ- c_j ומ- d. אבל ב- b וב- c_j המסלול כבר ביקר, ואילו אם נגיע ל- c מ- d אז המסלול לא יוכל להמשיך ולהגיע ל- t.

לכן, אם קיים מסלול המילטוני הוא חייב להיות מהצורה לעיל ובתהליך הפוך לתהליך שתיארנו ניתן לקבל את ההשמה המספקת.

☺

הגדרה: $SUBSET-SUM=\left\{(y_1,\ldots,y_l,t):\exists I\subset\{1,\ldots,l\}\left(\sum_{i\in I}y_i=t\right)\right\}$

טענה: $SUBSET-SUM\in NP-complete$

הוכחה: ברור ש- $SUBSET-SUM\in NP$ כי ניתן לנחש תת קבוצה ולבדוק את הנכונות. הבדיקה היא כמובן פולינומיאלית.

כדי להראות ש- $SUBSET-SUM\in NP-hard$ נראה ש- $3SAT\le_P SUBSET-SUM$. תהי נוסחה $\varphi=\bigwedge_{i=1}^{k}c_i$ כאשר $c_i=l_1^i\vee l_2^i\vee l_3^i$ מעל המשתנים $x_1,\ldots,x_n$. נבנה רשימה של l מספרים ומספר t כך ש- φ ספיקה אמ"מ קיימת תת קבוצה של מספרים שסכומם t.

נייצר טבלה עם $2(n+k)$ שורות ו- $n+k$ עמודות ונחלק אותה לארבעה חלקים בגודל $n\times k$.

- A: בעמודה i יש 1 בשורות $2i,2i-1$ ו- 0 בשאר.
- B: בעמודה j יש 1 בשורות המתאימות לליטרלים המופיעים בפסוקית c_j.
- C: כל התאים מאופסים.
- D: בעמודה j יש 1 בשורות $2j,2j-1$.

הבנייה היא כמובן פולינומיאלית.

הטבלה מייצגת $2(n+k)$ מספרים בבסיס טרינארי.

	x_1	x_2		x_n	c_1	c_2		c_k
x_1	1					1		
$\neg x_1$	1							
x_2		1						
$\neg x_2$		1				1		
			A				B	
x_n				1				
$\neg x_n$				1		1		
c_1					1			
c_1					1			
c_2						1		
c_2						1		
			C				D	
c_k								1
c_k								1

$c_2=x_1\vee\neg x_2\vee\neg x_n$

נראה ש-φ ספיקה אמ"מ יש תת קבוצה של שורות שסכומן t כאשר $t=\underbrace{11...1}_{n}\underbrace{3...3}_{k}$.

תהי f השמה מספקת ל-φ. מבין $2n$ השורות העליונות נבחר את השורות המתאימות לליטרלים שמקבלים ערך 1. בגלל ש-f השמה נבחר או את x_i או את $\neg x_i$ אבל לא את שניהם. לכן בחלק $\boxed{\begin{array}{c}A\\ \hline C\end{array}}$ סכום המשפרים הוא בדיוק $\underbrace{11...1}_{n}$. בגלל ש-f מספקת בחלק B לכל עמודה הסכום הוא בין 1 ל-3. אז מבין $2k$ השורות התחתונות נבחר את השורות כך שסכום כל עמודה ב-$\boxed{\begin{array}{c}B\\ \hline D\end{array}}$ יהיה 3.

מצד שני, נניח ש-$I\subset\{1,2,...,2n+2k\}$ כך ש-$\sum_{i\in I} y_i = t$ כאשר y_i הוא המספר של שורה i. נבנה השמה מספקת ל-φ. ההשמה תיתן ערך אמת 1 לליטרלים שנבחרו ב-A (כלומר הליטרלים שהשורות המתאימות להם נבחרו ל-I מבין $2n$ השורות העליונות). בגלל שבכל עמודה הסכום הוא 1 לא יכול להיות שניתן ערך 1 גם למשתנה וגם לשלילתו. כמו כן, בהכרח נותנים ערך לכל המשתנים. ההשמה היא מספקת משום שב-D לכל עמודה יש לפחות 1 אחד, כלומר בכל פסוקית יש ליטרל שקיבל ערך אמת 1 ולכן כל הפסוקיות מסופקות ומכאן שכל הנוסחה מסופקת.

☺

הגדרות:

1. k-**קליקה** היא גרף מלא לא מכוון עם k קודקודים.
2. $CLIQUE$ היא שפת המילים מהצורה $\langle G,k\rangle$ כך ש-G כרף לא מכוון שמכיל קליקה בגודל k.

טענה: $CLIQUE\in NP-complete$

הוכחה:

1. $CLIQUE\in NP$ שהרי ניתן פשוט לנחש קבוצה של קודקודים ולבדוק אם זו קליקה. תהליך זה הוא כמובן פולינומיאלי.
2. נראה ש-$3SAT\le_p CLIQUE$ ונקבל ש-$CLIQUE\in NP-hard$.

   בהינתן נוסחה φ בצורת $3CNF$ נבנה גרף G עם מספר k כך ש-$\varphi\in 3SAT\leftrightarrow\langle G,k\rangle\in CLIQUE$. נניח שנתונה נוסחה $\varphi=\bigwedge_{i=1}^{k}\left(l_1^i\vee l_2^i\vee l_3^i\right)$ בצורת $3CNF$. נגדיר כרף G באופן הבא:

   - הקודקודים V הם כל $3k$ הליטרלים - $\{l_1^i,l_2^i,l_3^i\}$
   - הצלעות E מחברות בין כל קודקוד לכל קודקוד פרט לליטרלים שבאותה פסוקית ופרט לליטרל ושלילתו.

הבניה היא בוודאי פולינומיאלית. יש ב- V $3k$ קודקודים וב- E יש לכל היותר $(3k)^2$ צלעות.

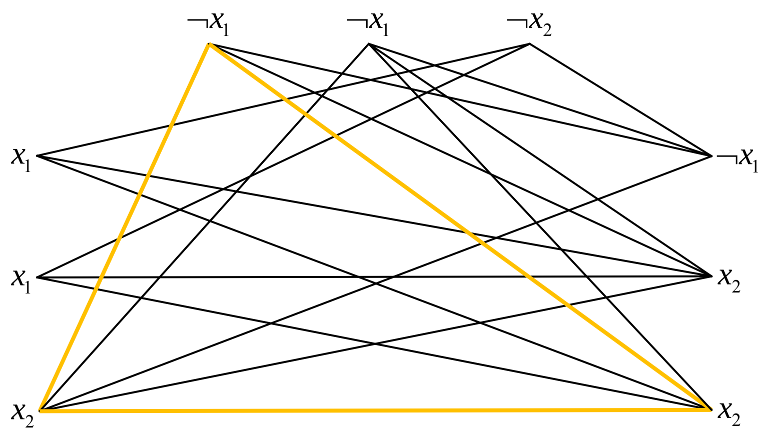
נראה ש- $\langle G,k\rangle \in CLIQUE \leftrightarrow \varphi \in SAT$.

$(\leftarrow)$ נניח $\langle G,k\rangle \in CLIQUE$. אזי יש ב- G k-קליקה. בגלל שבגרף אין צלעות בין הליטרלים של אותה פסוקית, כל k הקודקודים חייבים להיות מ- k פסוקיות שונות. כלומר מכל פסוקית בנוסחה יש לנו ליטרל אחד. נגדיר השמה ל- f שנותנת 1 לכל אחד מהליטרלים האלה ולמשתנים שלא מופיעים בין הליטרלים נותנת 0. נשים לב שההשמה אכן חוקית משום שנתון שאין ליטרל שמחובר עם שלילתו ולכן לא יכול להיות שברשימת הליטרלים שלנו יש גם משתנה וגם שלילתו ולכן לא יכול להיות שקיים משתנה שהוא ושלילתו מקבלים ערך אמת 1. ברור שההשמה הזאת מספקת משום שמכל פסוקית לקקנו ליטרל אחד והוא קיבל ערך אמת 1. סה"כ כל פסוקית סופקה בנפרד, ולכן גם כל הנוסחה סופקה.

$(\rightarrow)$ נניח ש- φ ספיקה. תהי $f:\{x_i\}\rightarrow\{0,1\}$ השמה מספקת. בכל פסוקית חייב להיות לפחות ליטרל אחד אשר מקבל ערך אמת 1 תחת f. יש k פסוקיות ולכן k ליטרלים. נטען שבין כל שני ליטרלים קיימת צלע. אחרת, או שהם משתנה ושלילתו או שהם מאותה פסוקית (כי כך הגדרנו את הגרף) אבל אף אחד מהתנאים הנ"ל לא מתקיים.

☺

דוגמה: נניח שהנוסחה היא $\varphi=(x_1\vee x_1\vee x_2)\wedge(\neg x_1\vee\neg x_2\vee\neg x_2)\wedge(\neg x_1\vee x_2\vee x_2)$. אזי הגרף המתאים לה הוא:



לנוסחה זו יש השמה מספקת לפי הגרף: $f(x_1)=0$ ו- $f(x_2)=1$.

הגדרות:

1. קבוצת קודקודים $S\subset V$ בגרף לא מכוון $G=\langle V,E\rangle$ נקראת **בלתי תלויה** אם לכל $u,v\in S$ $(u,v)\notin E$.
2. IS קבוצת כל המילים מהצורה $\langle G,k\rangle$ כך ש- G גרף לא מכוון שמכיל קבוצה בלתי תלויה בגודל k.

טענה: $IS\in NP-complete$

הוכחה:

1. ברור ש- $IS\in NP$ משום שניתן לנחש קבוצת קודקודים הגודל k ולבדוק אם היא בלתי תלויה.
2. נראה ש- $CLIQUE\le_p IS$ ונקבל ש- $IS\in NP-hard$.

בהינתן $\langle G,k\rangle$ נבנה $\langle G',k\rangle$ כך ש- $\langle G',k\rangle\in IS\leftrightarrow\langle G,k\rangle\in CLIQUE$. נגדיר את G' להיות הגרף המשלים, כלומר $G'=\langle V,V\times V\setminus E\rangle$. כעת ברור מההגדרה של קליקה ושל קבוצה בלתי תלויה שב- G יש k-קליקה אמ"מ ב- G' יש קבוצה בלתי תלויה בגודל k. שהרי, $S\subset V_G$ היא קליקה אמ"מ $S\subset V_{G'}$ היא בלתי תלויה.

☺

הגדרות:

1. יהי $G=\langle V,E\rangle$ גרף לא מכוון. **כיסוי** של G הוא קבוצה $C\subset V$ כך שלכל $(u,v)\in E$ מתקיים $u\in C$ או $v\in C$.
2. $VC=\{\langle G,k\rangle : G \text{ is a graph with a vertex cover of size } k\}$

טענה: $VC\in NP-complete$

הוכחה:

1. ברור ש- $VC\in NP$ משום שניתן לנחש קבוצה בגודל k ולבדוק אם היא כיסוי.
2. נראה ש- $IS\le_p VC$ ונקבל ש- $VC\in NP-hard$.

בהינתן $\langle G,k\rangle$ נסתכל על $\langle G,|V|-k\rangle$. ברור ש- $\langle G,k\rangle\in IS$ אמ"מ $\langle G,|V|-k\rangle\in VC$, שהרי S קבוצה בלתי תלויה ב- G אמ"מ אין צלעות בין איברי S. זה נכון אמ"מ כל צלע ב- G נוגעת לפחות בקודקוד אחד של S^c וזה אמ"מ S^c כיסוי. ברור כמובן שהבנייה פולינומיאלית, ובזאת סיימנו.

☺

סיבוכיות זיכרון

משפט Savitch

הגדרות:

1. בהינתן מ"ט M העוצרת על כל קלט, סיבוכיות הזיכרון של M היא פונקציה $s:\mathbb{N}\rightarrow\mathbb{N}$ כך ש-$s(n)$ חסם על מספר התאים בסרט בהם M משתמשת בריצה על קלט באורך n.
2. $SPACE(s(n))$ היא מחלקת כל השפות אשר ניתנות להכרעה ע"י מט"ד עם סיבוכיות זיכרון $s(n)$.
3. $NSPACE(s(n))$ היא מחלקת כל השפות אשר ניתנות להכרעה ע"י מטא"ד עם סיבוכיות זיכרון $s(n)$.
4. $PSPACE=\bigcup_{k\in\mathbb{N}} SPACE(n^k)$
5. $co-PSPACE=\{L:L^c\in PSPACE\}$
6. $NPSPACE=\bigcup_{k\in\mathbb{N}} NSPACE(n^k)$
7. $co-NPSPACE=\{L:L^c\in NPSPACE\}$

טענה: $TIME(f(n))\subset SPACE(f(n))$

הוכחה: תהי $L\in TIME(f(n))$. קיימת מט"ד M כך שלכל קלט באורך n M עוצרת אחרי לכל היותר $f(n)$ צעדים. בכל צעד M כותבת לתוך תא אחד בזיכרון, ולכן משתמשת בלכל היותר $f(n)$ תאים. כלומר $L\in SPACE(f(n))$.

☺

טענה: $SPACE(f(n))\subset TIME\left(f(n)\cdot 2^{O(f(n))}\right)$

הוכחה: תהי M מט"ד אשר בהינתן קלט באורך n משתמשת בלכל היותר $f(n)$ תאים. אם מספר התאים חסום ע"י $f(n)$ אזי מספר הקונפיגורציות האפשרי של המכונה הוא חסום ע"י $|Q|f(n)|\Gamma|^{f(n)}$, שהרי קונפיגורציה כוללת מידע על המצב בו המכונה נמצא, מיקום הראש הקורא על הסרט, והתוכן של הסרט. כעת, אם נתון שהמכונה עוצרת על כל קלט, אזי לא יכול להיות שבריצה שלה היא מגיעה לאותה קונפיגורציה פעמיים, שהרי אז היא הייתה נכנסת ללולאה אינסופית. לכן

המכונה מגיעה לכל אחת מהקונפיגורציות לכל היותר פעם אחת. ולכן זמן הריצה חסום ע"י מספר הקונפיגורציות האפשריות. מאחר ש- $|Q|,|\Gamma|$ הם קבועים נקבל שזמן הריצה חסום ע"י $f(n)\cdot 2^{O(f(n))}$.

☺

דוגמה: SAT ניתנת לפיתרון בסיבוכיות זמן לינארית. בהינתן קלט $\langle\varphi\rangle$ נוכל לפעול כך:

1. לכל השמת אמת f למשתנים $x_1,...,x_m$ נשערך את φ ביחס ל- f.
2. אם השערוך היה אמת נקבל את $\langle\varphi\rangle$, אחרת נעבור להשמה הבאה.
3. אם כל ההשמות שוערכו לשקר, נדחה את $\langle\varphi\rangle$.

הזיכרון שדרוש:

- השמת האמת הנוכחית - m תאים
- שערוך הנוסחה לפי ההשמה הנוכחית - $O(|\varphi|)$ תאים

סה"כ הזיכרון הדרוש הוא לינארי בגודל הקלט. כלומר, $SET \in LINEAR-SPACE$.

הגדרות:

1. $ALL_{DFA}=\{\langle A\rangle : A \text{ is a deterministic automaton and } L(A)=\Sigma^*\}$
2. $ALL_{NFA}=\{\langle A\rangle : A \text{ is a non deterministic automaton and } L(A)=\Sigma^*\}$

טענה: יהי $A=\langle Q,\Sigma,\delta,Q_0,F\rangle$ אס"ל. אזי $L(A)\neq\Sigma^*$ אמ"מ קיימת מילה w באורך לכל היותר $2^{|Q|}$ כך ש-$w\notin L(A)$.

הוכחה:

($\Leftarrow$) אם $L(A)\neq\Sigma^*$ פירוש הדבר ש- $L(A)^c\neq\varnothing$. באוטומט שמקבל את $L(A)^c$ יש לכל היותר $2^{|Q|}$ מצבים, שהרי נוכל לעבור לאס"ד השקול ואז לעבור למשלים. כדי לדעת אם קיימת מילה ב- $L(A)^c$ יש לבדוק אם קיים מסלול באס"ד מהמצב ההתחלתי לאחד המצבים המקבלים. אורך מסלול (ולכן גם ארוך המילה שמתקבלת) הוא לכל היותר $2^{|Q|}$.

($\Rightarrow$) זה ברור מההגדרה.

☺

טענה: יהי $A=\langle Q,\Sigma,\delta,Q_0,F\rangle$ אס"ל. קיימת מילה w באורך לכל היותר $2^{|Q|}$ כך ש- $w\notin L(A)$ אמ"מ קיימת סדרה של קבוצות של מצבים $S_0,S_1,...,S_{2^{|Q|}}$ שמתאימה לריצה של האוטומט הדטרמיניסטי השקול המתקבל ע"י חירצון A ויש $0\le i\le 2^{|Q|}$ כך ש- $S_i\cap F=\varnothing$.

הוכחה:

($\Leftarrow$) נניח שקיימת מילה כנ"ל באורך $i\le 2^{|Q|}$. אזי הריצה $r_0r_1...r_i$ של האס"ד השקול על w נגמרת במצב r_i שאינו מצב מקבל. אבל כל r_k הוא למעשה קבוצה S_i של מצבים של A ולפי תהליך החירצון לא יכול להיות ב- r_i מצב מקבל, כי אחרת r_i היה מקבל. לכן $S_i\cap F=\varnothing$.

($\Rightarrow$) נניח שקיימת סדרה כנ"ל. אזי המילה שמתאימה למעברים $S_0S_1...S_i$ היא מילה שאינה ב- $L(A)$.

☺

מסקנה: $ALL_{NFA}^{c}\in SPACE\left(O\left(|Q|\right)\right)$

הוכחה: נתאר מטא"ד M אשר מכריעה את ALL_{NFA}^{c}. בהינתן קלט $\langle A\rangle$ אשר מקודד אס"ל M תפעל כך:

1. תכתוב על הסרט את המצבים ב- Q_0
2. תאתחל מונה ל- 0
3. כל עוד המונה קטן מ- $2^{|Q|}$
   - אם הקבוצה S על הסרט מקיימת $S\cap F=\varnothing$ תקבל את $\langle A\rangle$
   - אחרת, תנחש אות a, תעדכן את הקבוצה על הסרט ל- $\delta(S,a)$ ותגדיל את המונה ב-1
4. תדחה

ברור שהאלגוריתם נכון לפי הטענה הקודמת. נספור את הזיכרון שנחוץ:

- קבוצת מצבים - $O\left(|Q|\right)$
- מונה - $O\left(|Q|\right)$ כי ניתן לספור בבסיס בינארי
- מצביעים עבור העדכונים - $O(1)$

סה"כ האלגוריתם פועל בסיבוכיות זיכרון $O\left(|Q|\right)$.

☺

משפט Savitch: לכל $S:\mathbb{N}\to\mathbb{N}$ כך ש-$n\le s(n)$ מתקיים $NSPACE(s(n))\subset SPACE(s^2(n))$.

הוכחה: בהינתן מטא"ד M עם סיבוכיות זיכרון $s(n)$ נבנה מט"ד שקולה M' עם סיבוכיות זיכרון $s^2(n)$. בה"כ ל-M יש קונפיגורציה התחלתית יחידה c_{init} וקונפיגורציה מקבלת יחידה c_{acc}. כמו כן, יהי d קבוע כך שאין ל-M יותר מ-$2^{d\cdot s(n)}$ קונפיגורציות שונות בריצה על קלט באורך n (קיים כזה משום שמספר הקונפיגורציות חסום ע"י $|Q|s(n)|\Gamma|^{s(n)}$).

נגדיר שגרה דטרמיניסטית $\text{reach}(c_1,c_2,t)$ אשר מכריעה האם ניתן לעבור מקונפיגורציה c_1 לקונפיגורציה c_2 תוך לכל היותר t צעדים. השגרה תעבוד באופן הבא:

1. אם $t=1$ בדוק האם $c_1=c_2$ או שיש מעבר של צעד אחד ביניהן
2. אם $1<t$ אז לכל קונפיגורציה c של M המשתמשת ב-$s(n)$ תאים:
   2.1. הרץ את $\text{reach}\left(c_1,c,\left\lceil\frac{t}{2}\right\rceil\right)$
   2.2. הרץ את $\text{reach}\left(c,c_2,\left\lceil\frac{t}{2}\right\rceil\right)$
   2.3. אם שתי הריצות קיבלו, קבל
3. דחה

נכונות האלגוריתם ברורה. ננתח את סיבוכיות הזיכרון שלו. עומק הרקורסיה הוא $\log t$. בכל קריאה שומרים שתי קונפיגורציות ומונה שזה לוקח זיכרון $O(\log t+s(n))$. לכן, סה"כ סיבוכיות השטח היא $O((\log t+s(n))\log t)$.

אם M' תפעיל את $\text{reach}\left(c_{init},c_{acc},2^{d\cdot s(n)}\right)$ נקבל בדיוק את הדרוש משום שלכל $w\in\Sigma^*$ M' מקבלת את w אמ"מ M מקבלת את w. נשים לב שאם $t=2^{d\cdot s(n)}$ אז $\log t=d\cdot s(n)$ ו-M' רצה בסיבוכיות זיכרון $O(s^2(n))$.

☺

מסקנות:

1. $NPSPACE=PSPACE$
2. $co-NPSPACE=co-PSPACE$
3. $co-PSPACE=PSPACE$

הוכחה:

1. המעבר ממטא"ד למט"ד שקולה מצריך רק העלאה פולינומיאלית של סיבוכיות הזיכרון. לכן, כל בעיה שניתן לפתור ע"י מטא"ד בסיבוכיות זיכרון פולינומיאלית ניתן לפתור גם ע"י מט"ד בסיבוכיות זיכרון פולינומיאלית.

2. $co-PSPACE=\left\{L:L^c\in PSPACE\right\}=\left\{L:L^c\in NPSPACE\right\}=co-NPSPACE$
3. אם מט"ד M מכריעה שפה L בסיבוכיות זיכרון פולינומיאלית אז ניתן להכריע גם את L^c בסיבוכיות זיכרון פולינומיאלית, שהרי נגדיר מט"ד M' שזהה ל- M מלבד החלפת המצב המקבל במצב הדוחה ולהפך. בפרט M' ו- M פועלות באותה סיבוכיות זיכרון.

☺

שלמות ב- $PSPACE$

הגדרות:

1. שפה L היא $PSPACE$-קשה ($PSPACE-hard$) אם לכל $L'\in PSPACE$ מתקיים $L'\leq_p L$
2. שפה L היא $PSPACE$-שלמה ($PSPACE-complete$) אם $L\in PSPACE$ וגם $P\in PSPACE-hard$.

משפט: אם L שפה $PSPACE$-שלמה ו- $L\in P$ אז $P=PSPACE$

הוכחה: ברור ש- $P\subset PSPACE$. נראה ש- $PSPACE\subset P$. תהי $L'\in PSPACE$. אזי $L'\leq_p L$. תהי f הרדוקציה הפולינומיאלית. נוכל להכריע את L' באופן הבא: בהינתן w נחשב את $f\left(w\right)$. זה לוקח זמן פולינומיאלי ולכן גם מקום פולינומיאלי. כעת, נשתמש באלגוריתם הפולינומיאלי של L כדי להכריע את $f\left(w\right)$, וזה ייתן לנו הכרעה של L'. נשים לב שמאחר ש- $\left|f\left(w\right)\right|$ פולינומיאלי ב- $\left|w\right|$ גם הריצה של המט"ד של L על $f\left(w\right)$ לוקחת זמן פולינומיאלי. לכן $L'\in P$.

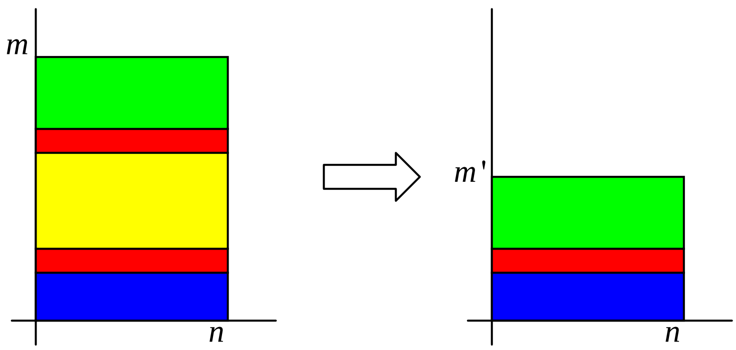
☺

הגדרה: בעיית הריצוף החסום SBT היא שפת כל המילים מהצורה $\left\langle T,H,V,t_{in},t_{fin},n\right\rangle$ כאשר n מספר שנתון באונארית וקיים ריצוף חוקי בגודל $n\times m$ עבור m כלשהו כך ש- $f\left(1,1\right)=t_{in}$ ו- $f\left(1,m\right)=t_{fin}$.

טענה: בסימונים של ההגדרה, השאלה האם קיים ריצוף חוקי בגודל $n\times m$ עבור m כלשהו שקולה לשאלה האם קיים ריצוף חוקי בגודל $n\times m$ עבור $m\le|T|^n$.

הוכחה:

($\Leftarrow$) נניח שקיים ריצוף חוקי עבור $|T|^n<m$ ונראה שקיים $m'\le|T|^n$ כך שיש ריצוף חוקי עבורו. נשים לב שאם יש $|T|$ אריחים שונים ו- n תאים שיש לרצף, אזי מספר הריצופים השונים שקיימים עבור שורה הוא $|T|^n$. לכן, אם קיים ריצוף שבו יותר מ- $|T|^n$ שורות בהכרח קיימות שתי שורות $1\le i<j\le m$ אשר חוזרות על עצמן. נשים לב שאת כל השורות שבין i ו- j (כולל i אך לא כולל j) ניתן להוריד מהריצוף ולהישאר עם ריצוף חוקי, שהרי שורה $i-1$ (אם קיימת כזאת) התאימה לשורה i ושורה $j+1$ (אם קיימת כזאת) התאימה לשורה j. כעת נוכל לחזור על התהליך הזה עד שכל השורות בריצוף יהיו שונות. מאחר שיש לכל היותר $|T|^n$ שורות שונות נקבל ריצוף חוקי עם $m'\le|T|^n$ שורות.



($\Rightarrow$) ברור.

☺

טענה: $SBT\in PSPACE-complete$

הוכחה:

1. ראשית, יש להראות ש- $SBT\in PSPACE$
מאחר ש- $PSPACE=NPSPACE$ מספיק להראות ש- $SBT\in NPSPACE$. נבנה מטא"ד M שמקבלת כקלט מילה $\langle T,H,V,t_{in},t_{fin},n\rangle$ ומכריעה אותה באופן הבא:

   1. לכל $1\le m\le|T|^n$
      1.1. לכל ריצוף אפשרי בגודל $n\times m$
         1.1.1. בדוק האם הריצוף חוקי
         1.1.2. אם כן, קבל את המילה
         1.1.3. אחרת, עבור לריצוף הבא
   2. דחה

בעקבות הטענה הקודמת, ברור שהאלגוריתם פועל. נסמן ב- k את אורך מילת הקלט ונחשב את סיבוכיות הזיכרון שלו:

- יש צורך במונה אשר יכול לספור עד $|T|^n$. בשביל לייצג את המספר הזה בבינארית יש צורך ב- $\log|T|^n = n\log|T|$ תאים. מאחר ש- n נתון באונארית, זה לוקח $O\left(k^2\right)$ תאים.
- בשביל לבדוק אם ריצוף הוא חוקי יש צורך בשמירה של שתי שורות בלבד ויכולת לזכור את הריצופים שכבר בדקנו. מאחר ש- n נתון באונארית צריך $O\left(k\right)$ תאים כדי לשמור את המידע הזה.

סה"כ סיבוכיות הזיכרון הוא פולינומיאלית.

2. שנית, יש להראות ש- $SBT \in PSPACE-hard$ אבל לא נעשה זאת במסגרת זו.

☺

הגדרות:

1. **נוסחה בוליאנית** היא נוסחה שמופיעים בה 0 , 1, משתנים בוליאניים והקשרים $\wedge ,\vee ,\neg$.
2. בהינתן נוסחה φ במשתנים $x_1,...,x_n$ נאמר שהמשתנה x_i **מכומת** אם φ מהצורה $\exists x_i\psi\left(x_1,...,x_n\right)$ או $\forall x_i\psi\left(x_1,...,x_n\right)$. משתנה שאינו מכומת נקרא **משתנה חופשי**.
3. **משפט** הוא נוסחה שכל משתניה מכומתים.
4. $TQBF$ היא שפת כל המילים מהצורה $\langle\varphi\rangle$ כאשר φ משפט נכון.

משפט: $TQBF \in PSPACE-complete$

הוכחה:

1. $TQBF \in PSPACE$:

בהינתן נוסחה $\forall x\varphi\left(x\right)$ או $\exists x\varphi\left(x\right)$ כאשר $\varphi\left(x\right)$ נוסחה שהמשתנה החופשי היחיד שלה הוא x , נשערך את $\varphi\left(0\right)$ ואת $\varphi\left(1\right)$ באופן רקורסיבי. עומק הרקורסיה הוא מספר המשתנים. לכן כדי לעבור על כל עץ הרקורסיה צריך לכל ענף רק $O\left(n\right)$ תאים כאשר n מספר המשתנים.

2. $TQBF \in PSPACE-hard$:

נראה ש- $SBT \leq_p TQBF$. הרעיון הוא לבנות נוסחה $\psi_{c_1,c_2,t}$ שתהיה נכונה אמ"מ יש מעבר חוקי בריצוף משורה c_1 לשורה c_2 כך שיש t שורות ביניהן לכל היותר.

יש $|T|$ אריחים ובכל שורה n מקומות. לכן מספר השורות השונות הוא לכל היותר $|T|^n$.

מכאן שכל שורה ניתן לקודד ע"י $l = \log|T|^n = n\log|T|$ משתנים. תחת ההנחה ש- $|T| < n^2$ זה מספר פולינומיאלי באורך הקלט.

לנוסחה יהיו l משתנים וכל השמת אמת למשתנים תקודד שורה בריצוף.

נגדיר באופן רקורסיבי:

- $\psi_{c_1,c_2,1} = (c_1 = c_2) \vee (c_2 \text{ can be legally placed above } c_1)$
- $\psi_{c_1,c_2,t} = \exists x \left(\psi_{c_1,c,\left\lceil \frac{t}{2} \right\rceil} \wedge \psi_{c,c_2,\left\lceil \frac{t}{2} \right\rceil} \right)$

אנחנו נתעניין ב- $\psi_{c_{init},c_{fin},|T|^n}$ לצורך הרדוקציה. הבעיה היא שהיא לו פולינומיאלית משום שבצורה הנאיבית שבה הגדרנו את המשוואות החישוב לוקח $O(t \log t)$ ובמקרה שלנו $t = |T|^n$.

נשים לב שניתן לכתוב באופן שקול
$\psi_{c_1,c_2,t} = \exists c \forall c_2 \forall c_4 \left((c_3 = c_1 \wedge c_4 = c) \vee (c_3 = c \wedge c_4 = c_2) \right) \to \psi_{c_3,c_4,\left\lceil \frac{t}{2} \right\rceil}$ והחישוב של נוסחה זו לוקח $O(\log t)$ ובמקרה שלנו $O(n)$.

☺

המחלקות L ו- NL

הגדרות:

1. המחלקה $LOGSPACE$ (או L) היא מחלקת כל השפות A כך שיש מט"ד M שמכריעה את L ובנוסף לסרט הקלט משתמשת בסרט נוסף באורך $O(\log n)$ לכל קלט באורך n.
2. המחלקה $NLOGSPACE$ (או NL) היא מחלקת כל השפות A כך שיש מטא"ד M שמכריעה את L ובנוסף לסרט הקלט משתמשת בסרט נוסף באורך $O(\log n)$ לכל קלט באורך n.

דוגמאות:

1. $\{0^n 1^n : 0 \le n\} \in L$. נשים לב שבשביל לספור עד n בבינארית יש צורך ב- $\log n$ תאים. אז נוכל לקרוא את החלק הראשון של המילה ולספור על סרט העבודה כמה אפסים הופיעו. לאחר מכן לספור כמה אחדות הופיעו ובסוף להשוות.
2. $PATH = \{\langle G,s,t \rangle : G \text{ is a graph and there is a path between } s \text{ and } t\}$

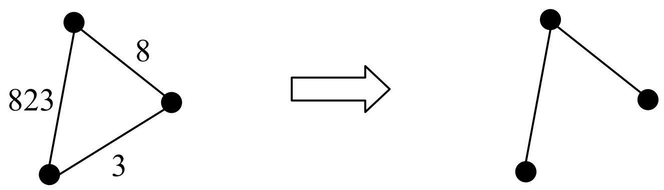
קל לראות ש- $PATH \in NP$ כי בהינתן רשימת קודקודים ניתן לוודא שהיא אכן מסלול כנדרש. למעשה, אפילו $PATH \in P$ שהרי ניתן לבצוע BFS או DFS על הגרף ואלה הם אלגוריתמים פולינומיאליים.
נראה ש- $PATH \in NL$. המכונה תשמור על הסרט את $\langle s \rangle$ ואת $\langle t \rangle$ לצורך ההשוואה וגם את הקודקוד הנוכחי במסלול שהיא מנחשת. כמו כן, היא שומרת מונה שסופר עד $|V|$ כדי שהמכונה תפסיק לנחש. בכל שלב המכונה מנחשת קודקוד הבא במסלול. אם באיזה שלב

ניחשנו את t סימן שסיימנו. אם יש n קודקודים אז צריך $O(\log n)$ תאים כדי לייצג קודקוד ו- $\log n$ תאים כדי לספור עד n. לכן סה"כ המכונה משתמשת $O(\log n)$ זיכרון נוסף על הקלט.

הגדרות:

1. **משרן במקום לוגריתמי**[8] הוא מט"ד M בעלת שלושה סרטים: RO (אשר ניתן רק לקרוא ממנו), R/W ו- WO (אשר ניתן רק לכתוב עליו) כאשר עבור קלט באורך n בסרט R/W יש $O(\log n)$ תאים. נאמר שהמכונה מחשבת פונקציה $f:\Sigma^* \to \Sigma^*$ אם לכל מילה $w\in\Sigma^*$ הכתובה בסרט RO M עוצרת עם פלט $f(w)$ בסרט WO.
2. $A' \le_L A$ אם קיים משרן שמחשב פונקציה $f:\Sigma^* \to \Sigma^*$ כך ש- $w\in A' \leftrightarrow f(w)\in A$.
3. שפה A היא NL**-קשה**[9] אם לכל שפה $A'\in NL$ מתקיים $A' \le_{Lm} A$.
4. שפה A היא NL**-שלמה**[10] אם $A\in NL$ ו- A NL-קשה.

דוגמה: נתכנן משרן אשר מקבל כקלט גרף עם משקלות $\langle V,E,w\rangle$ ופולט גרף לא ממושקל $\langle V,E'\rangle$ כך ש- $e\in E'$ אמ"מ $e\in E$ ו- $w(e)>7$. למשל,



המשרן משתמש בשלושה סרטים: סרט קלט ארוך, סרט עבודה קצר וסרט פלט ארוך. בתחילה הוא מעתיק את הצמתים לסרט הפלט. לאחר מכן הוא עובר על הקשתות כדי להחליט האם להעתיק אותה לסרט הפלט. בשביל זה מעתיקים את המשקל שלה לסרט העבודה ומשווים ל- 7. נשים לב שאם אורך הייצוג של המשקל ארוך מהייצוג של 7 אנחנו כבר יכולים לדעת ש- $w(e)>7$.

משפט: אם $A\le_L B$ ו- $B\in L$ אז $A\in L$.

הערה: אל לנו להתבלבל. ההוכחה הבאה לא נכונה:

בהינתן M_B עבור B נבנה M_A עבור A: על קלט w המכונה M_A תריץ את המשרן המובטח לחישוב $f(w)$ ותריץ את M_B על $f(w)$.

הבעיה עם ההוכחה היא שלא מובטח לנו דבר על האורך של $f(w)$. לכן בסופו של דבר M_A אכן תכריע את A אך יכול להיות שהשיכרון שהיא תשתמש בו לא יהיה לוגריתמי.

[8] Log space transducer
[9] $NL-hard$
[10] $NL-complete$

הוכחה: נתאר מט"ד M_A שתכריע את A. על קלט w המכונה תחשב בכל איטרציה את האות מתוך $f(w)$ שאותה M_B צריכה כרגע. החישוב הזה לוקח זיכרון לוגריתמי לפי ההנחה. כש-M_B תסיים את החישוב שלה נדע אם M_A צריכה לקבל או לא. ייתכן שנריץ את המשרן הרבה מאוד פעמים אבל בכל זאת מבחינת זיכרון הוא ישתמש רק ב-$O(\log n)$ משום שכל איטרציה משתמשת רק ב-$O(\log n)$.

☺

טענה: $PATH$ היא NL-שלמה.

הוכחה: כבר ראינו שהיא ב-NL. אז נראה שהיא NL-קשה. תהי $A \in NL$ ותהי M_A מ"ט שמכריעה אותה בשטח $O(\log n)$. בהינתן קלט w ל-M_A נייצר $\langle G,s,t \rangle$ כך ש-M_A מקבלת את w אמ"מ $\langle G,s,t \rangle \in PATH$.

הצמתים של G יהיו קונפיגורציות של M_A בריצה על w. בה"כ יש קונפיגורציה התחלתית יחידה s וקונפיגורציה מקבלת יחידה t. המעברים בין הקונפיגורציות יתאימו להליכה במסלול בגרף. סה"כ יש $|Q| \cdot n \cdot |\Gamma|^{O(\log n)}$ קונפיגורציות אפשריות. לכן בשביל לייצג קונפיגורציה צריך סרט באורך $\log\left(c_1 \cdot n \cdot c_2^{O(\log n)}\right) = \log c_1 n + \log c_2^{O(\log n)} = O(\log n)$.

המשרן עובר בסדר לקסיקוגרפי כל המילים באורך $c \log n$ ומעתיק אותן לסרט הפלט שלו אם הן מייצגות קונפיגורציה (אלה הקודקודים של הגרף). אח"כ הוא עובר על כל זוגות המילים ובודק אם הן מייצגות קונפיגורציות עוקבות ואם כן הוא מעתיק את הזוג לסרט הפלט (אלה הקשתות בגרף). לאחר מכן הוא מזהה את הקונפיגורציה ההתחלתית והמקבלת ומעתיק גם אותן.

ברור ש-M_A מקבלת את w אמ"מ $\langle G,s,t \rangle \in PATH$.

☺

מסקנה: $NL \subset P$

הוכחה: אם $A \in NL$ אז $A \leq_L PATH$. אבל $PATH \in P$ והרדוקציה שהראנו למעלה היא פולינומיאלית בזמן. לכן $A \in P$.

☺

משפט היררכיית הזיכרון

הגדרה: פונקציה $f:\mathbb{N}\to\mathbb{N}$ המקיימת $\log n\le f(n)$ מקראת ניתנת לבנייה בזיכרון[11] אם קיימת מכונת טיורינג M_f שבהינתן קלט אונארי 1^n מחשבת את $f(n)$ בייצוג בינארי במגבלת זיכרון $O(f(n))$.

משפט היררכיית הזיכרון: לכל פונקציה $f:\mathbb{N}\to\mathbb{N}$ הניתנת לבנייה בזיכרון קיימת שפה L_f הניתנת להכרעה במגבלת זיכרון $O(f(n))$ אך לא במגבלת זיכרון $o(f(n))$.

הוכחה: נתאר שפה A אשר מתקבלת ע"י מ"ט D במגבלת זיכרון $O(f(n))$ אך שונה מכל שפה שמוכרעת בעזרת מגבלת זיכרון של $o(f(n))$.

בהינתן קלט w המכונה D תפעל באופן הבא:

1. יהי n האורך של w.
2. מחשבת את $f(n)$ שהיא ניתנת לבנייה בזיכרון ומקצה בדיוק $f(n)$ תאים על הסרט. אם בשלב מאוחר יותר המכונה תנסה לכתוב מעבר לתאים אלה היא תדחה את w.
3. אם w אינה מהצורה $\langle M\rangle 10^*$ עבור מ"ט M כלשהי D תדחה את w.
4. D מסמלצת את ריצת M על w וסופרת את מספר הצעדים. אם המונה עולה על $2^{f(n)}$ D דוחה את w.
5. אם M דוחה D מקבלת ואם M מקבלת D דוחה.

ברור ש-D עוצרת תמיד ולכן היא מכריעה את השפה $L(D)$. כמו כן, ברור שהיא עובדת במגבלת זיכרון $O(f(n))$.

נראה שלא קיימת מ"ט שמעריכה את $L(D)$ במגבלת זיכרון $o(f(n))$. נניח בשלילה ש-M מכריעה את $L(D)$ במגבלת זיכרון $g(n)$ כאשר $g(n)=o(f(n))$. קיים קבוע n_0 שהחל ממנו $g(n)<f(n)$. לכן D יכולה לסמלץ את M כל עוד הקלט שלה הוא באורך n_0 ומעלה. נסתכל על ריצת D על הקלט $\langle M\rangle 10^{n_0}$. בוודאי D יכולה לסמלץ את כל הריצה. לכן D תחזיר תוצאה הפוכה משל M על אותו הקלט, בסתירה לכך ש-M מכריעה את $L(D)$.

☺

[11] Space constructible

מסקנות:

1. עבור $k_1 < k_2$ $SPACE\left(n^{k_1}\right) \subsetneq SPACE\left(n^{k_2}\right)$
2. לכל $k \in \mathbb{N}$ $SPACE\left(n^{k}\right) \subsetneq SPACE\left(n^{\log n}\right)$
3. $PSPACE \subsetneq SPACE\left(n^{\log n}\right)$
4. $SPACE\left(n^{\log n}\right) \subsetneq SPACE\left(2^{n}\right)$
5. $PSPACE \subsetneq SPACE\left(2^{n}\right)$

בהצלחה!!!

DinaZil's Notes-Heaven