(1) 26.02.07
חישוביות

סודחה
סמסטר ב'

ברוכים הבאים לחישוביות!

www.cs.huji.ac.il/~compu

המרצה: אורנה קופרמן   orna@cs.huji.ac.il

שעת קבלה - יום ב'   8-10

בקורס יהיו שני בחנים שיהיו מגן. לראשון כבר יש תאריך - 30/04
הבחנים הם אמריקאים. מי שעשה את כל שיעורי הבית מלאו טפש
במיוחד צריך להצליח בבחן. הבחנים יהיו 15%.
תרגילים - צריך להגיש את כלם פרט לשניים וצריך לעבור את
כולם בשביל לגשת למבחן.

הספר העיקרי של הקורס - Sipser

☆ ☆ ☆ ☆

מה לומדים בקורס בחישוביות?

מבוא ← מבנ"ת ← אלגוריתמים ← חישוביות

מבוא: איך להגיד למחשב מה לעשות ומשם מודל

מבנ"ת: איך להגיד למחשב מה לעשות אבל ביעילות! לשם מיון

אלגוריתמים: פתרון בעיות השגת מטרה יעילות, ללא שימוש טכני

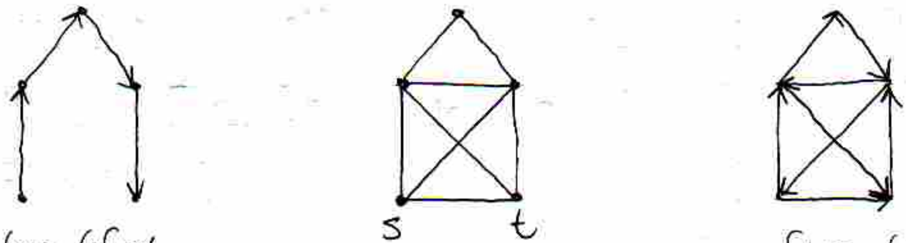
חישוביות: מה מחשבים יכולים ולא יכולים לעשות ובאיזה מחיר?

הקורס נחלק לשלושה חלקים
1) מודלי חישוב (אוטומטים)
2) חישוביות (מה המחשב יכול ומה לא)
3) סיבוכיות (מה המחיר של החישובים)

סיבוכיות: נתון גרף $G = (V, E)$ ונתונים שני קודקודים מיוחדים - $s,t \in V$. יש כל מיני שאלות שאפשר לשאול.

- **מסלול אוילר** - האם יש מסלול מ-s ל-t העובר על כל קשת בדיוק פעם אחת?
- **מסלול המילטוני** - האם יש מסלול מ-s ל-t העובר של כל קודקוד בדיוק פעם אחת?

מסלול אוילר זה בעצם כל החידות שאנחנו מכירים מהיסודי - האם אפשר לצייר איזו צורה בקו אחד.



מסלול המילטון   מסלול אוילר

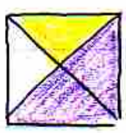
בעית מסלול אוילר ניתנת לפתרון בזמן ליניארי $O(|V|)$ - כי זה תלוי בדרגות של הקודקודים.

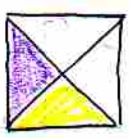
אפשר להוכיח שאר בעית מסלול המילטון אפשר לפתור רק ע"י בדיקת כל האפשרויות.

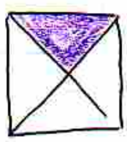
סיווג של בעיות למחלקות סיבוכיות:

- לא לשאול ליותר מדי (אם בעיה לא ניתנת לפתרון פולינומיאלי אז חבל לבזבז זמן מחפש פיתרון (כזה).
- קירובים / אלגוריתמים רנדומליים.
- לנצל קושי של בעיות (בעיקר בהצפנה)

תישבוריות: דוגמה - נתונים אריחים $t_1, \dots, t_n$.
כל אריח מאופיין ע"י ארבעה צבעים.
האם לכל n ניתן לרצף ריבוע n×n כך שאריחים שכנים יסכימו על הצבע (מותר לסובב אריחים).


א'


ב'


ג'

למשל בדוגמה הנ"ל אפשר לעשות זאת לכל n באופן הבא:

ג	א			
ב	ג	א		
א	ב	ג	א	
ג	א	ב	ג	א

ככה אפשר להמשיך באלכסונים

אפשר להוכיח שבעיה הזאת לא קיים אלגוריתם שאומר אם ריצוף הוא אפשרי או לא. מניחים שיש אלגוריתם ואז מראים שיש קלט שיכול לבלבל אותו.

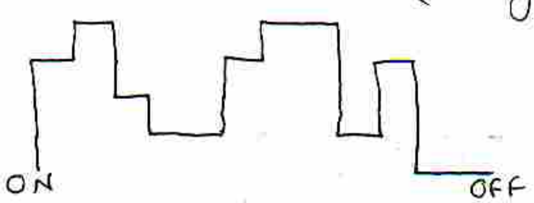
דוגמה נוספת - בעיית העצירה. רוצים תכנית שמקבלת כקלט:
- תכנית ב Java (P)
- קלט x ל-P

והתכנית מכריעה האם P עוצרת על X.
לבעיה הזאת גם כן אין אלגוריתם לפיתרון!

אוטומטים
automata

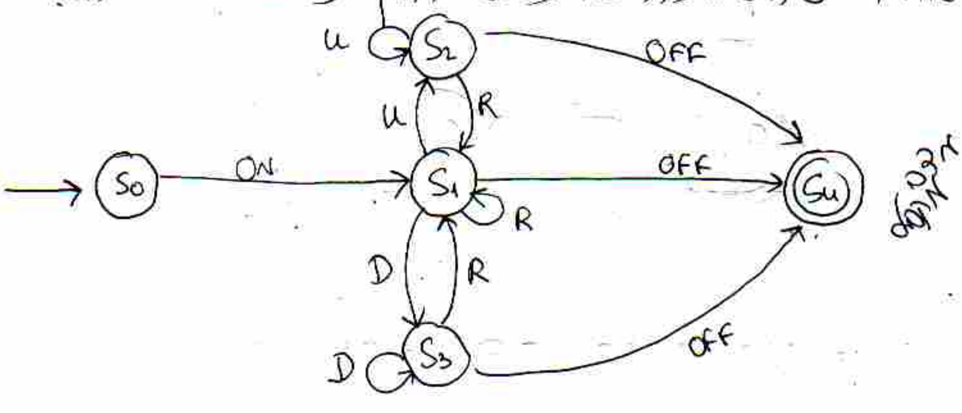
אוטומט הוא מודל בסיסי - "מחשב עם זיכרון מצומצם".

דוגמה: יש עט וניתן לתת לו 6 פקודות: ON, OFF, R, L, D, U
רוצים מערכת שמקבלת סדרת פקודות ומכריעה האם המילה חוקית (= מציירת skyline). למשל



אלה לא חוקיים: ⇅ וחייבים להתחיל עם ON ולסיים עם OFF.

נצייר אוטומט שמזהה את זה (אם אין חץ שיוצא ממצב עבור קלט כלשהו זה אומר שהאות הזאת לא חוקית במצב הזה).



הגדרות

1) אלפבית (א"ב): קבוצה סופית של אותיות $\Sigma=\{\sigma_1,\ldots,\sigma_n\}$

דוגמאות:
$\Sigma=\{0,1\}$
$\Sigma=\{0,1\}^4 \quad |\Sigma|=16$
$\Sigma=\{OFF, ON, U, D, L, R\}$

2) מילה: סדרה סופית של אותיות $W=\sigma_1\sigma_2\ldots\sigma_k$

- המילה הריקה - ε
- כל המילים מעל א"ב Σ: $\Sigma^*=\{w : w \text{ מילה מעל } \Sigma\}$

3) שפה $L\subseteq\Sigma^*$ (אנחנו נדבר רק על שפות פורמליות)

אוטומט הוא חמישיה $M=\langle Q,\Sigma,\delta,q_0,F\rangle$ כאשר

- Q - קבוצה סופית של מצבים
- Σ - א"ב
- δ - פונקציית מעברים - $\delta: Q\times\Sigma\to Q$
- q_0 - שייך ל-Q והוא המצב ההתחלתי
- F - המצבים הסופיים (המקבלים). תת קבוצה של Q.

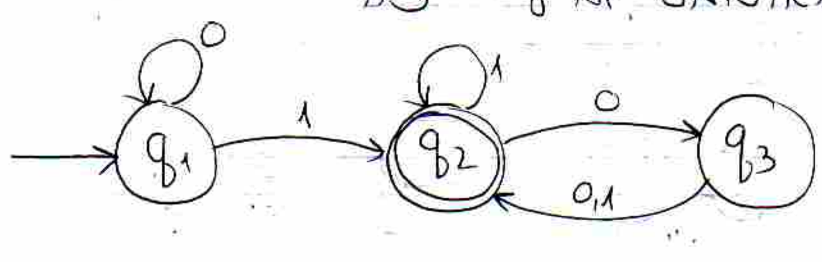
האוטומט הנ"ל הוא דטרמיניסטי לכל קלט מתאימה ריצה אפשרית יחידה בלבד.

דוגמה: $M_1=\langle\{q_1,q_2,q_3\},\{0,1\},\delta,q_1,\{q_2\}\rangle$

כאשר 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

אפשר לתאר את האוטומט גם ע"י ציור



כל אוטומט מגדיר שפה - קבוצת המילים מ-Σ^* שהוא מקבל

(3) בהינתן מילה $w = w_1 w_2 \dots w_n \in \Sigma^*$ **ריצה** של M על w היא

סדרת מצבים $r_0, r_1, \dots, r_n$ ב-Q כך ש-

1) $r_0 = q_0$ (הריצה מתחילה במצב ההתחלתי)

2) לכל $0 \le i \le n-1$ מתקיים $r_{i+1} = \delta(r_i, w_{i+1})$

למשל הריצה של M_1 על 00110 היא q_1, q_1, q_2, q_2, q_3.

הריצה r **מקבלת** אם $r_n \in F$, אחרת r **דוחה**.

האוטומט M מקבל את w אם הריצה של w מקבלת.

<u>שפה</u> של אוטומט M היא קבוצת המילים ש-M מקבל

$L(M) = \{w : M \text{ מקבל את } w\}$

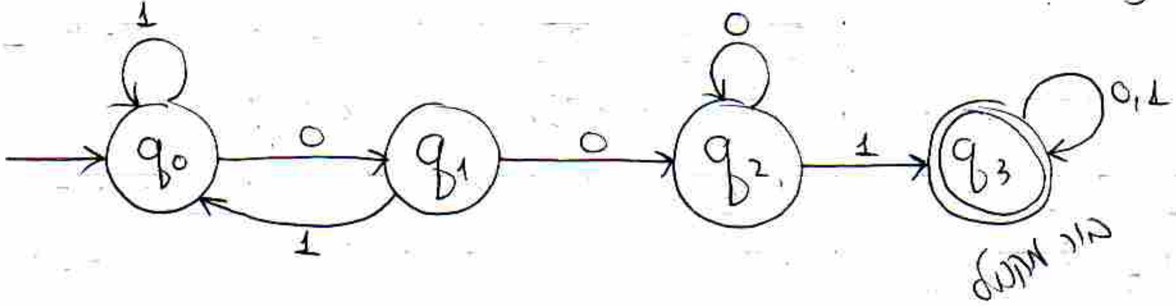
למשל השפה של M_1: "יש לפחות 1 אחד, ואחרי ה-1 האחרון יש מספר זוגי של 0-ים".

$\boxed{\text{דוגמה:}}$ נבנה M_2 כך ש-

$L(M_2) = \{w : w \text{ מכילה את תת המילה } 001\}$

כשאנחנו קוראים מילה אנחנו יכולים להיות באחד מהמצבים הבאים ואלה גם יהיו המצבים של האוטומט:

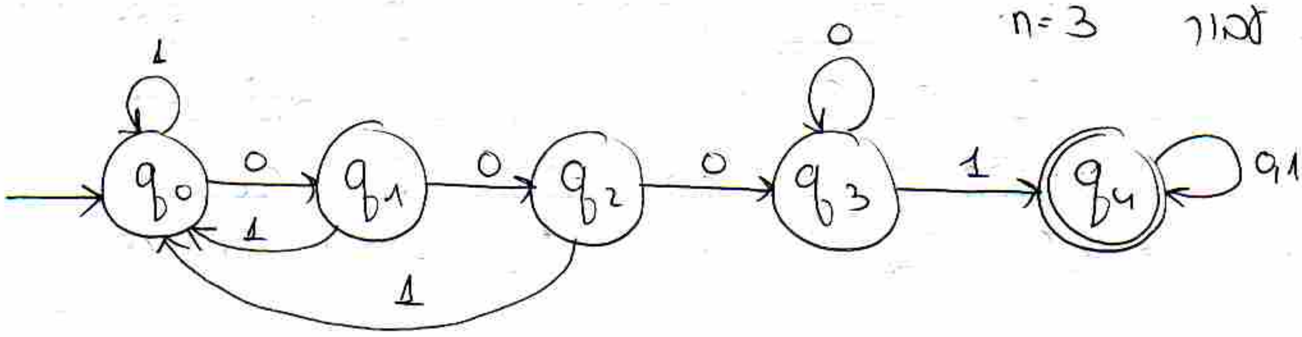
- לפני ש-001 מתחיל
- קראנו 0
- קראנו 00
- קראנו 001



מתי כדאי לצייר אוטומט?

לו היינו רוצים $L(M_n) = \{w : w \text{ מכילה את } 0^n 1\}$

לדוגמא עבור $n=3$



עכשיו שהבנו איך זה עובד אנחנו מוכנים לתאר את האוטומט הכללי:

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

נגדיר את δ:

$$\begin{cases} \delta(q_0,0) = q_1 \\ \delta(q_0,1) = q_0 \end{cases}$$

עבור $1 \le i \le n-1$

$$\begin{cases} \delta(q_i,0) = q_{i+1} \\ \delta(q_i,1) = q_0 \end{cases}$$

$$\begin{cases} \delta(q_n,0) = q_n \\ \delta(q_n,1) = q_{n+1} \\ \delta(q_{n+1},0) = \delta(q_{n+1},1) = q_{n+1} \end{cases}$$

דוגמה: $L = \{0^n 1^n : n \in \mathbb{N}\}$

את השפה הזאת אי אפשר לקבל ע"י אוטומט סופי!

$L \subseteq \Sigma^*$ היא **רגולרית** אם קיים אוטומט סופי M כך ש-$L(M)=L$.

השפה מהדוגמה האחרונה אינה רגולרית!

תכונות סגור של שפות רגולריות

- איחוד: בהינתן L_1, L_2 שפות מעל Σ

$$L_1 \cup L_2 = \{w : w \in L_1 \vee w \in L_2\}$$

- חיתוך: $L_1 \cap L_2 = \{w : w \in L_1 \wedge w \in L_2\}$
- שרשור: $L_1 \cdot L_2 = \{w_1.w_2 : w_1 \in L_1 \wedge w_2 \in L_2\}$
- כוכב: $L^* = \{w_1 \dots w_k : k \ge 0 \wedge \forall i\ w_i \in L\}$

דוגמה: $L_1 = \{1, 333\}$, $L_2 = \{22, 4444\}$, $\Sigma = \{1,2,3,4\}$

$L_1 \cup L_2 = \{1, 22, 333, 4444\}$ $L_1 \cap L_2 = \emptyset$ $L_1 \cdot L_2 = \{122, 14444, 33322, 3334444\}$

$L_1^* = \{\varepsilon, 111, 1111, 11333, 1333333, 33311333$, and infinitely many more$\}$

④ 28.02.07
תשובות.

הודעה: התרגול של יום רביעי 08^{00}-10^{00} עבר לשפרינצק 217

דיברנו על שפה $L \subseteq \Sigma^*$ ופעולות שאפשר לעשות על שפות:
איחוד, חיתוך, שרשור, כוכב.

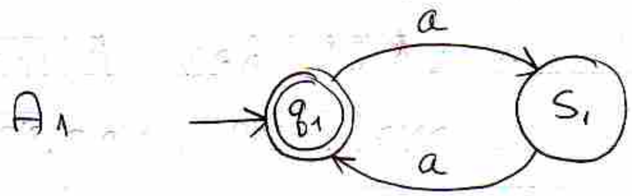
תכונות סגור של שפות רגולריות (=שפות שיש אוטומט שמקבל אותן)
מה זה בכלל סגור? למשל המספרים הטבעיים סגורים לכפל.
לכל $m,n \in \mathbb{N}$ גם $m \cdot n \in \mathbb{N}$. אבל הם לא סגורים לחילוק, כי
למשל $1/2 \notin \mathbb{N}$
שפות רגולריות הן סגורות תחת הפעולות שהזכרנו לעיל.

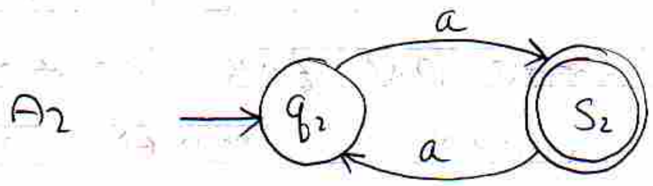
משפט: אם L_1, L_2 שפות רגולריות אז $L_1 \cup L_2$ רגולרית.
הוכחה: מהנתון קיימים אוטומטים A_1 עבור L_1
A_2 עבור L_2
נבנה אוטומט A עבור $L_1 \cup L_2$.
נניח ש-
$$A_1 = \langle Q_1, \Sigma, \delta_1, s_1, F_1 \rangle$$
$$A_2 = \langle Q_2, \Sigma, \delta_2, s_2, F_2 \rangle$$
נשים לב שעשינו כאן הנחה שהם מוגדרים על אותה שפה. זה
לא מגביל את הכלליות כי אם השפות הן Σ_1 ו- Σ_2 אפשר
להסתכל על האיחוד $\Sigma = \Sigma_1 \cup \Sigma_2$ ולומר שהאוטומטים הם מעל
שפה זאת.
חוץ מזה אנחנו מניחים שהאוטומטים הם מלאים וגם זה בסדר כי אם
יש מצב שבו לא יכול לקבל קלט מסוים אז פשוט נוסיף בור דוחה לאוטומט
שזה מצב שאליו מגיעים כל הקלטים הלא חוקיים ונתקעים שם.
נגדיר $A = \langle Q, \Sigma, \delta, s_0, F \rangle$ כאשר
$$Q = Q_1 \times Q_2$$
$$s_0 = \langle s_1, s_2 \rangle$$
$$\delta(\langle q_1, q_2 \rangle, \sigma) = \langle \delta_1(q_1, \sigma), \delta_2(q_2, \sigma) \rangle$$
$$F = F_1 \times Q_2 \cup Q_1 \times F_2$$

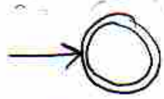
דוגמה:

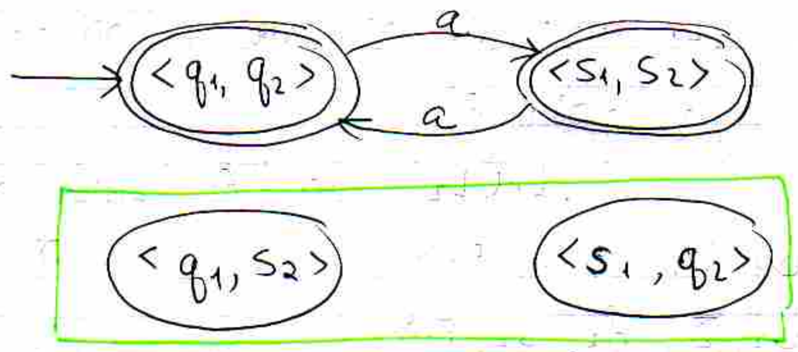
$\Sigma=\{a\} \quad L_1 = \{w : |w| \equiv 0 \pmod 2\}$

$\Sigma=\{a\} \quad L_2 = \{w : |w| \equiv 1 \pmod 2\}$





האיחוד $L_1 \cup L_2$ זה פשוט כל המילים. אז מספיק מצב אחד בשביל לתאר אותה.  אבל באוטומט המכפלה יש ארבעה מצבים. חלק מאלה יש בו מצבים לא ישיגים.



אלו מצבים לא ישיגים אז לא חייבים לצייר אותם.

אנחנו הראינו שאוטומט המכפלה לא נותן אוטומט מינימלי ואנחנו עוד נלמד איך למזער אוטומטים.

נוכיח כעת שהבנייה שלנו נכונה. צריך להוכיח ש- $L(A)=L(A_1)\cup L(A_2)$

($\subseteq$) $L(A) \subseteq L(A_1)\cup L(A_2)$

תהי $w=w_1 \dots w_n$ מילה ב- $L(A)$. ויש ריצה $r= r_0,\dots,r_n \in Q^{n+1}$ מקבלת של A על w. נסמן $r_i=\langle q_1^i, q_2^i\rangle$. נשים לב ש- $t_1 = q_1^0, q_1^1, \dots, q_1^n$ היא הריצה של A_1 על w. זה נובע פשוט מהגדרת המצבים שלנו. הגדרנו את המצב ההתחלתי של A להיות $\langle S_1, S_2\rangle \Leftarrow q_1^0 = S_1$ (הריצה מתחילה במצב ההתחלתי של A_1) והגדרנו $\langle q_1^{i+1}, q_2^{i+1}\rangle = \langle \delta_1(q_1^i, w_{i+1}), \delta_2(q_2^i, w_{i+1})\rangle$ לכן הריצה מתקדמת לפי δ_1 כי אנחנו מסתכלים על ההטלה של האיבר הראשון.

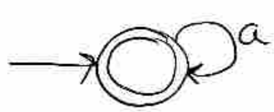
באותו אופן יודעים שהסדרה $t_2 = q_2^0, q_2^1, \ldots, q_2^n$ היא הרידה של A_2 על w. ובטוח אחת מהרידות האלה מקבלת. מאחר ש-r מקבלת אז $q_1^n \in F_1$ או $q_2^n \in F_2$ ולכן קיבלנו ש- $w \in L(A_1)$ או $w \in L(A_2)$, כלומר $w \in L(A_1) \cup L(A_2)$.

(ב) $L(A_1) \cup L(A_2) \subseteq L(A)$

באופן דומה לסעיף הקודם. אם $w = w_1 \ldots w_n$ מילה שמתקבלת ע"י A_1 או ע"י A_2, ונראה שהיא מתקבלת גם ע"י A.
תהי t_1 הרידה של A_1 על w ותהי t_2 הרידה של A_2 על w. אזי הרידה של A על w היא שילוב של t_1 ו-t_2 (לפי קואורדינטות) והיא רידה מקבלת של A על w.

למה התעקשנו שהמעברים יהיו שלמים? - כדי שהבנייה שלנו לא תיתקע. נניח ש- $\Sigma = \{a\}$ ו- $L_1 = \Sigma^*$ ו- $L_2 = \{a\}$.
אז

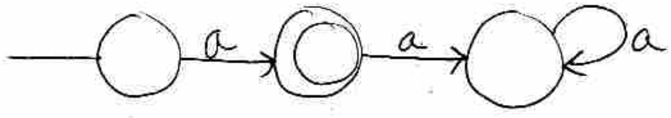
A_1



A_2

עכשיו אם נעשה איחוד הוא עדיין Σ^* אבל נשרטט את אוטומט המכפלה אז הוא ייתקע אחרי שנקרא a אחת. לכן A_2 חייב להיות עם בור דוחה:

A_2



נשים ♡ שבאותו אופן יכולנו לבנות גם את אוטומט החיתוך - רק שהמצבים המקבלים יהיו $F_1 \times F_2$ שזה המצבים שבהם גם A_1 קיבל וגם A_2 קיבל.
אפשר גם לעשות XOR של השפות - כל המילים שמתקבלות ע"י בדיוק אחת מהשפות ואז המצבים המקבלים יהיו $(F_1 \times Q_2 \cup Q_1 \times F_2) \setminus (F_1 \times F_2)$.

מאור לשרשור - קודם ניזכר מה זה שרשור. שרשור זה נשים מילים של השפה הראשונה ואז מילים של השפה השנייה. למשל,

$L_1 = \{ w : |w| \equiv 0 \pmod 2 \}$

$L_2 = \{ w : |w| \equiv 0 \pmod 5 \}$

אז למשל $a^9 = \underline{aaaa}\,\underline{aaaaa}$ כן

מילה בשרשור.

אוטומטים אי-דטרמיניסטים

זה אוטומט שיכול לנחש לאן ללכת.

- קבוצה של מצבים התחלתיים
- פונקציית המעברים מוגדרת כ- $\delta: Q \times \Sigma \to 2^Q$

כלומר מצב זה יכול להיות ממופה ל<u>קבוצה</u> של מצבים.

אנחנו נראה איך אוטומט אי-דטרמיניסטי אפשר להפוך לאוטומט דטרמיניסטי אבל גם הם מגדירים שפות רגולריות

פתק שמח

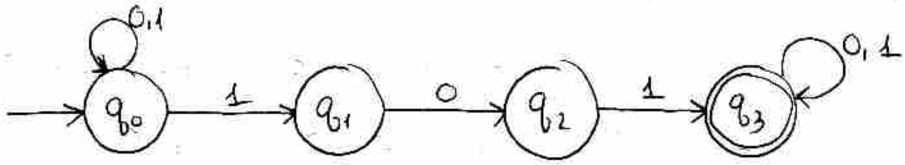
(6) 7/3/07

חישוביות

אוטומטים לא דטרמיניסטיים

את האוטומט הלא דטרמיניסטי נכיר ע"י דוגמא.

$L=\{w : 101 \text{ מכילה } w\}$



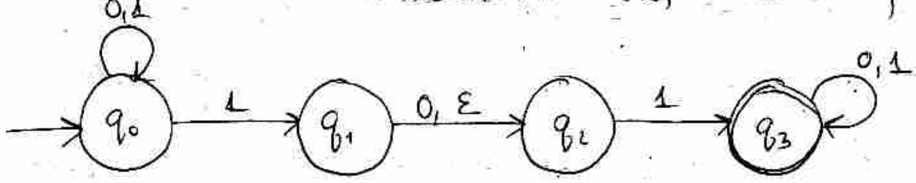
לאוטומט לא דטרמיניסטי יכולות להיות כמה אפשרויות מעבר ממצב ואות ולכן יכולות להיות מספר ריצות על מילת הקלט. למשל עבור המילה 110100 יכולה להיות הריצה $\langle q_0q_0q_0q_0q_0q_0\rangle$ אבל יכולה גם להיות הריצה $\langle q_0q_0q_1q_2q_3q_3q_3\rangle$. כל פעם שנותנים לאוטומט מילה הוא מנחש לאן כדאי להתקדם.

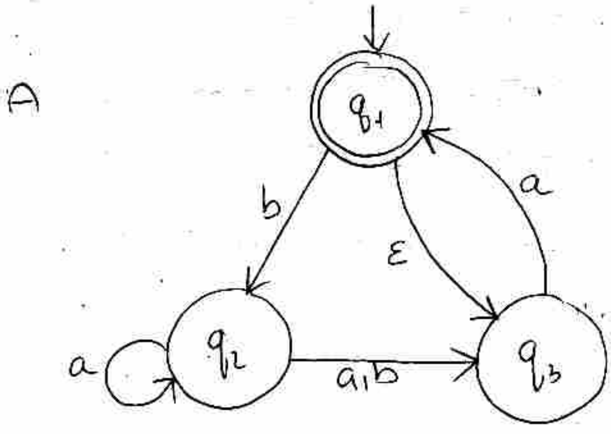
נאמר שמילה מתקבלת ע"י האוטומט אם קיימת ריצה מקבלת עבור המילה הנ"ל. אז שפת האוטומט היא קבוצת המילים שמתקבלות.

<u>צעדי ε</u> - לאוטומט מותר לדלג למצבים עוקבים בלי להתקדם בקריאת הקלט. בצעד זה כאילו שבין שתי האותיות מופיעה המילה הריקה. לאוטומט מותר לבצע צעדי ε רק אם רשום ε על הצלע המתאימה.

למשל אם $L=\{w : 11 \text{ או } 101 \text{ מכילה } w\}$

אז האוטומט הבא מקבל את השפה:





<u>דוגמא</u>

ε ✓

a ✓

b X

baba ✓

__הגדרה__ __עצמיין אי-חורצני__ הוא חמישייה $A = \langle Q, \Sigma, \delta, Q_0, F \rangle$

כאשר - $Q_0 \subseteq Q$ קבוצה של מצבים התחלתיים

$\delta: Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ פונקציית מעברים לא חורצנית.

__ריצה__ של A על $w = w_1 \dots w_n \in \Sigma^*$ היא סדרת מצבים $r = r_0 r_1 \dots r_m$

כך ש- (1) ניתן לכתוב את w כ- $w = y_1 y_2 \dots y_m$ כאשר $y \in \Sigma \cup \{\varepsilon\}$

(כלומר דוחפים ε-ים בין האותיות)

(2) $r_0 \in Q_0$ (הריצה מתחילה ממצב התחלתי)

(3) לכל $0 \le i < m$ $r_{i+1} \in \delta(r_i, y_{i+1})$ (הריצה מתקדמת לפי δ)

הריצה r __מקבלת__ אם $r_m \in F$.

A __מקבל__ את w אם קיימת ריצה מקבלת של A על w.

__קיצורים__ - עצמיין סופי תורצני DFA deterministic finite automaton

עצמיין סופי אי-חורצני NFA nondeterministic finite automaton

__משפט:__ לכל NFA A קיים DFA A' כך ש- $L(A) = L(A')$

__הוכחה:__ נתון NFA $A = \langle Q, \Sigma, \delta, Q_0, F \rangle$ ונניח של-A אין צעדי ε, כלומר $\delta: Q \times \Sigma \to 2^Q$ (ניתן להניח זאת כי בתרגול נראה שניתן בקלות להסיר צעדי ε מהישור עם אוטומט שקול).

נבנה DFA $A' = \langle Q', \Sigma, \delta', q_0', F' \rangle$ שקול (כלומר $L(A) = L(A')$).

הרעיון הוא ש- $Q' = 2^Q$ כלומר כל מצב יתאים לקבוצה של מצבים ב-A.

הריצה של A' מגיעה במצב $S \in 2^Q$ אחרי קריאת w אמ"מ S היא הקבוצה של כל המצבים ב-Q ש-A עשויה להיות בהם אחרי קריאת w.

נכתוב את הרעיון פורמלית. נרחיב את $\delta: Q \times \Sigma \to 2^Q$ לקבוצות של מצבים ומילים.

$$M: 2^Q \times \Sigma^* \longrightarrow 2^Q$$

$M(S,w) =$ כל המצבים ש-A יכול להגיע אליהם, בריצה כלשהי שמתחילה ב-S וקוראת w.

נוכיח את M באינדוקציה על האורך של המילה.

① $M(S,\varepsilon) = S$

② עבור אות $\sigma\in\Sigma$: $M(S,\sigma)=\{q : \exists s\in S \ q\in\delta(s,\sigma)\} = \bigcup_{s\in S}\delta(s,\sigma)$

③ עבור מילה w ואות σ : $M(S,w.\sigma) = M(M(S,w),\sigma)$

נגדיר את δ' כך ש- $\delta'(q_0',w)=M(Q_0,w)$ (*) ואז נגדיר את קבוצת המצבים המקבלים ע"י $F'=\{S : S\cap F\neq\emptyset\}$ שהרי אם ב-A' אם הגענו למצב מקבל זה אומר שב-A היתה ריצה עבור w שקיבלה אותה.

עכשיו נגדיר את $\delta': 2^Q\times\Sigma\to 2^Q$ ע"י $\delta'(S,\sigma)=M(S,\sigma)$.

בכאן סיימנו את הבניה. כעת נוכיח את הנכונות:

נוכיח את (*) באינדוקציה על $|w|$.

אם - $|w|=0$ אז המצב ההתחלתי של A' הוא Q_0

- $|w|=k+1$ מהנחת האינדוקציה אם $w=w'\sigma$ אז

$$\delta'(q_0',w')=M(Q_0,w')$$

ואז מהגדרת δ' : $\delta'(q_0',w\sigma)=\delta'(\delta'(q_0,w'),\sigma)=M(Q_0,w'\sigma)$

11

נשים ♡ שקרה פה דבר עצוב - נאלצנו להגדיל אקספוננציאלית את מספר המצבים.

(8) 12.03.07

חישוביות

יום הולדת שמח לענת!

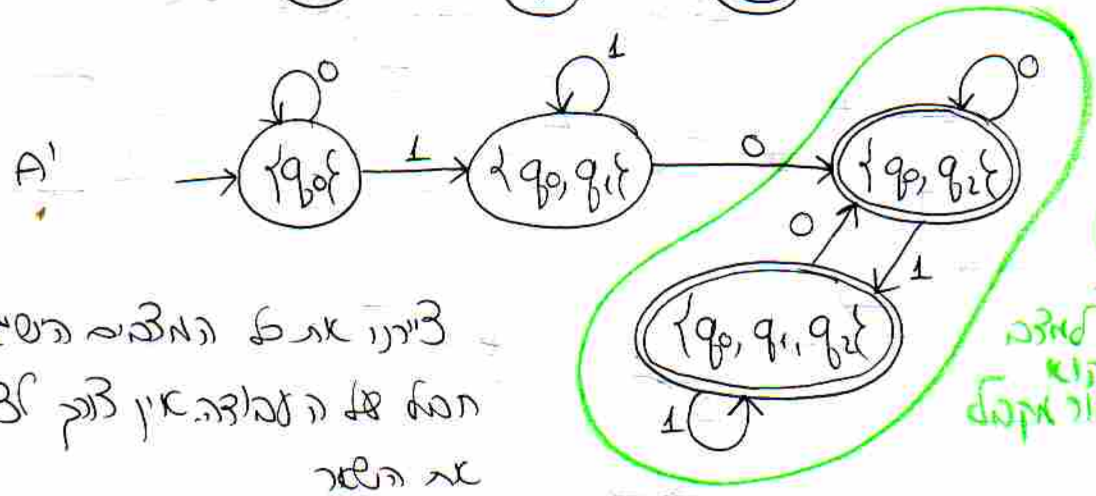
בשיעור הקודם ראינו איך אפשר להעביר עצמיון לא חורצני לעצמיון חורצני. אם A לא חורצני אז מגדירים

$$A' = \langle 2^Q, \Sigma, \delta', Q_0, F' \rangle$$

כאשר $\delta' : 2^Q \times \Sigma \to 2^Q$ מוגדרת ע"י $\delta'(S,\sigma) = \bigcup_{s \in S} \delta(s,\sigma)$.

$$F' = \{ S : S \cap F \neq \emptyset \}$$

דוגמה:



את שני אלה אפשר לאחד למצב אחד והוא יהיה בור מקבל

ציירנו את כל המצבים הישיגים. חלק של העבודה אין צורך לצייר את השאר

התהליך שעשינו נקרא חירצון. מה שמטריד בחירצון הוא הגדלה המעריכית של המצבים.

יש שני דברים שאפשר לעשות אולי:

- להראות איך אפשר לצמצם את העצמיון
- להוכיח שיש חסם תחתון

משפט: יש משפחה של שפות $L_1, L_2, \ldots, L_n$ כך שלכל $n \geq 1$:

1) L_n ניתנת לזיהוי ע"י NFA עם $O(n)$ מצבים

2) ה-DFA הקטן ביותר עבור L_n צריך לפחות 2^n מצבים.

נשים לב: המשפט אומר שיש חסם תחתון לניפוח אקספוננציאלי, שהרי אם יש בניה פולינומיאלית נפעיל אותה על המשפחה $L_1, \ldots, L_n, \ldots$ ונמצא שקיים $i \geq 0$ שעבורו נקבל סתירה.

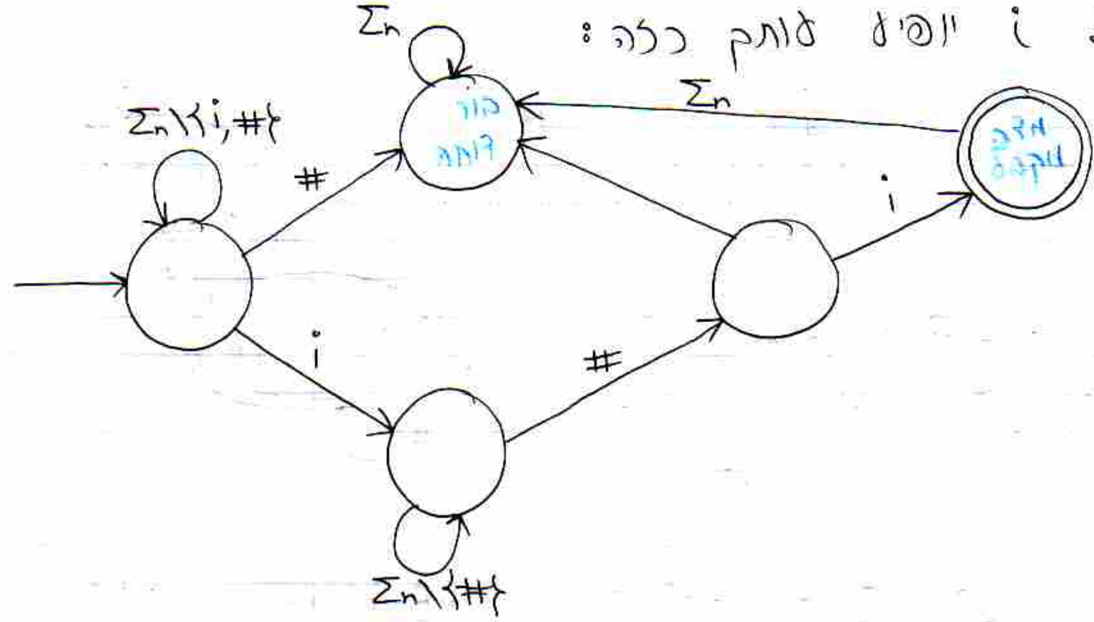
כמו כן צריך משפחה שלמה של שפות ולא די בשפה אחת, כי משפה יחידה אי אפשר להסיק שאכן מדובר בניפוח מעריכי ולא בקבוע כפלי גדול...

__הערה:__ נגדיר $\Sigma_n = \{1,2,\dots,n,\#\}$

$$L_n = \{\sigma_1\sigma_2\dots\sigma_k\#\sigma_{k+1} : \forall_{1\le i\le k}\ \sigma_i\in\{1,\dots,n\} \wedge \sigma_{k+1}\in\{\sigma_1,\dots,\sigma_k\}\}$$

למשל:
$1242\#4 \in L_4$
$1222\#4 \notin L_4$
$13323\#1 \in L_4$
$1\#\#1 \notin L_4$

(1) נראה NFA עם $O(n)$ מצבים עבור L_n:
הוא מנחש ש- i הוא האות האחרונה.
והיו לו n מצבים התחלתיים בהתאם לניחוש
ש- $1,\dots,n$ יכולות להיות האותיות האחרונות.
לכל i יופיע עותק כזה:



סה"כ בכל העצמיון יהיו $3n+2$ מצבים - $3n$ לכל האותיות
בתוספת מצב מקבל ובור דוחה.
קל לראות שהעצמיון הזה באמת מתאים לשפה שהגדרנו.
הדרך היחידה שלו לקבל מילה היא אם הוא ניחש נכון את האות
האחרונה והיא באמת הופיעה אחרי הסולמית הראשונה.

(2) DFA צריך לפחות 2^n מצבים:
נניח בשלילה שיש DFA A_n עם פחות מ- 2^n מצבים עבור L_n.
$\Leftarrow$ יש שתי קבוצות $S_1,S_2\subseteq\{1,\dots,n\}$ שונות כך שהמילים שמורכבות
מהאותיות בהן מגיעות לאותו מצב.

$$\delta^*(q_0,\langle S_1\rangle) = \delta^*(q_0,\langle S_2\rangle) = q$$

$\langle S_i\rangle =$ "המילה המורכבת" מאותיות S_i בסדר לקסיקוגרפי.
זה נובע מעיקרון שובך היונים משום שיש 2^n קבוצות כאלה ופחות
מ- 2^n מצבים.

תהי בה"כ $\sigma \in S_1 \setminus S_2$. נתבונן ב-

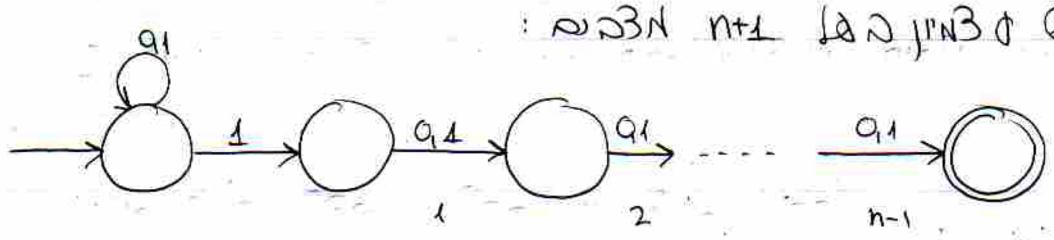
$\delta(\delta(q,\#),\sigma) = q'$ = לאן הולך A_n מקוראו $\sigma\#$ מהמצב q.

- 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 שהמילה היחידה של $\langle S_1\rangle\#\sigma$ נדחתה וזה בסתירה לכך ש- $\langle S_1\rangle\#\sigma \in L_n$.

קיבלנו סתירה בכל מקרה. לכן לא יכול להיות שקיים ϕ3מיין חורצני עם פחות מ- 2^n מצבים.

עכשיו אנחנו באמת יודעים שיש חסם תחתון.
אם $f:\mathbb{N}\to\mathbb{N}$ פולינומיאלית שמתיימרת להיות חסם עליון לחוצין אז נתבונן במה שקורה כשמפעילים את האלגוריתם על L_n הנ"ל. הסדרה $f(n_1), f(n_2), \dots$ כאשר n_i מספר המצבים ב-NFA עבור L_i מקיימת שקיים i רק ש- $f_n(i)<2^i$ וזאת סתירה למשפט.

בתרגול: $L_n = \{$כל המילים שבהן יש 1 במקום ה-n לפני הסוף$\}$
לשפה כי יש ϕ3מיין בעל $n+1$ מצבים:



ב- DFA עבור שפה זו חייבים להיות לפחות 2^n מצבים.

ביטויים רגולריים

דוגמאות

1. "1 במקום הלפני לפני אחרון" $(0+1)^*\cdot 1\cdot(0+1)\cdot(0+1)$
2. מילים עם 1 אחד בדיוק $0^*\cdot 1\cdot 0^*$
3. לפחות 1 אחד $0^*1(0+1)^* = (0+1)^*\cdot 1\cdot(0+1)^*$

הגדרה: ביטוי רגולרי (ב"ר) מעל א"ב Σ

1) $a\in\Sigma$, ε, $\emptyset$ הם ביטויים רגולריים

2) אם r_1, r_2 הם ב"ר הם גם r_1+r_2 $(r_1\cup r_2)$

$r_1\cdot r_2$

r_1^*

כל ב"ר מגדיר שפה: $L(r)\subseteq\Sigma^*$

$L(a)=\{a\}$ $\qquad L(r_1+r_2)=L(r_1)\cup L(r_2)$

$L(\varepsilon)=\{\varepsilon\}$ $\qquad L(r_1\cdot r_2)=L(r_1)\cdot L(r_2)$ ★ בפ"כ אסוציאטיבית ".."

$L(\emptyset)=\emptyset$ $\qquad L(r_1^*)=(L(r_1))^*$

סדר הפעולות מהקושר יותר לפחות: $*$, $\cdot$, $+$.

דוגמאות:

- "מילים שמסתיימות ב-0": $(0+1)^*0$
- L_n מהמסמך התחתון: $\bigcup_{i\in\{1,\dots,n\}}(1+\dots+n)^*\,i\,(1+\dots+n)^*\#i$ ⇐ זה כמו $\sum_{i=1}^{n}$

שפות לא רגולריות

ראינו שאוטומט לא דטרמיניסטי לא יותר חזק מאוטומט דטרמיניסטי אבל הוא יכול להיות יעיל יותר.

ננסה לראות שיש שפות שהם לא יכולים לקבל.

נתבונן בשפה $L=\{0^n1^n : n\geq 0\}$

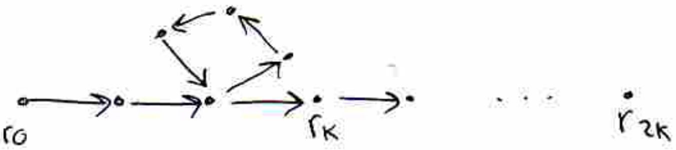
טענה: אין ל-L DFA

הוכחה: נניח בשלילה שיש DFA A כך ש-$L(A)=L$.

נניח שיש ל-A k מצבים. נתבונן ב-$w=0^k1^k$. המילה w מתקבלת ע"י A ⇐ יש ריצה מקבלת $r=r_0r_1\dots r_kr_{k+1}\dots r_{2k}$.

לפי עקרון שובך היונים חייב להיות בריצה מעגל. שהרי יש k מצבים שונים אבל רק ב-$r_0\dots r_k$ יש $k+1$ מצבים.

קיימים $0 \le i < j \le k$ רק ש- $q = r_i = r_j$.



$\Leftarrow$ העצמיין מקבל גם את $0^{k+(j-i)}1^k$ למרות שהיא אינה בשפה.

הנחת היסוד שגויה סתירה לכך שהעצמיין חוקי.

למת הניפוח Pumping Lemma

אם L רגולרית אז קיים $1 \le p$ (שנקרא גם קבוע הניפוח) כך שלכל מילה w המקיימת $|w| \ge p$, יש חלוקה $w = x \cdot y \cdot z$ רק ש-

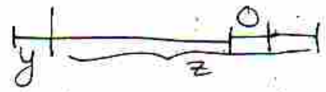
1) לכל $0 \le i$ $xy^iz \in L$
2) $|y| > 0$
3) $|x \cdot y| \le p$ (יתכן $x = \varepsilon$, $y = \varepsilon$)

דוגמה $L = (0+1)^*0(0+1)$

נראה שתנאי הלמה מתקיימים:

ניקח $p=3$. לכל $w \in L$ רק ש- $|w| \ge 3$. נתבונן בחלוקה

$x = \varepsilon$ $\quad |y| = 1$



ברור ש- $|y| > 0$ (כי $|y| = 1$)

ברור ש- $|x \cdot y| \le 3$ (כי $|x| = 0$, $|y| = 1$)

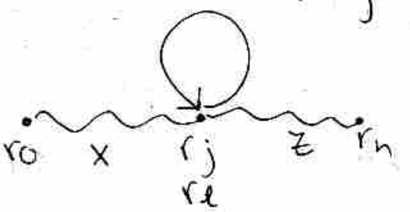
האורך של z לפחות 2 לכן כשמנפחים את y הניפוח לא נוגע בשתי האותיות האחרונות.

(11) 14.3.07
חישוביות

דיברנו על <u>למת הניפוח</u>: אם L רגולרית, אז יש p כך שכל מילה
w כך ש- $|w| \ge p$ ניתנת לחלוקה $w=xyz$ כך ש-
1) לכל $i \ge 0$ $xy^iz \in L$
2) $y \ne \varepsilon$ $(|y|>0)$
3) $|xy| \le p$

אנחנו נשתמש בלמה בעיקר בכיוון ההפוך. נשתמש בה כדי להוכיח ששפות אינן רגולריות.

<u>הוכחת הלמה</u>: יהי $A=\langle Q,\Sigma,q_0,\delta,F\rangle$ DFA שמזהה את L.
(יכולנו להוכיח שקיים NFA אבל ככה החיים שלנו קלים יותר וזה הרי שקול אז לא משנה). נסמן $p=|Q|$. נתבונן במילה
$w=w_1w_2\dots w_n$ עבור $n \ge p$. נניח שהריצה של A על w
היא $r=r_0r_1r_2\dots r_p\dots r_n$. נתבונן בקבוצה $\{r_0,r_1,\dots,r_p\}$.
יש בה $p+1$ מצבים אבל $|Q|=p$ ⇐ לפי עיקרון שובך
היונים חייבים להיות $j<l \le p$ כך ש- $r_j=r_l$.
נחלק את המילה באופן הבא:



$\overbrace{w_1w_2\dots w_j}^{x}\ \overbrace{w_{j+1}\dots w_l}^{y}\ \overbrace{w_{l+1}\dots w_n}^{z}$

r_0 r_1 r_j r_l

נראה שמתקיימים תנאי הלמה:
1) החלוקה של המילה בעצם משרה חלוקה של הריצה לשלושה חלקים.
לכל $i \ge 0$ הסדרה $r_0\dots r_j(r_{j+1}\dots r_l)^i r_{l+1}\dots r_n$ היא ריצה מקבלת
של A על xy^iz. למה זאת באמת ריצה חוקית? ברור
ש- $r_0\dots r_j$ הוא חלק ריצה חוקי ומשם אפשר לעבור ל- r_{j+1}.
מ- r_l אפשר לעבור ל- r_{l+1}. למה אפשר לעבור מ- r_l ל- r_{j+1}?
משום שהנחנו ש- $r_j=r_l$ ומ- r_j אפשר לעבור ל- r_{j+1}.
כמובן, גם במקרה... ש- $i=0$ אפשר לעבור מ- w_j ל- w_{l+1}
מסיבה דומה.

2) $y \neq \varepsilon$ שהרי $j < l$.

3) $|x \cdot y| = l \le p$.

בזאת הוכחנו את למת הניפוח.

שימוש בלמת הניפוח

כך נוכיח ששפה אינה רגולרית. נראה שלכל p תהיה מילה שלא ניתנת לנפח. נוכל להראות את זה כך: אם $w \in L$ כך ש- $|w| \ge p$ אז לכל חלוקה $w = xyz$, אם $|y| > 0$ ו- $|xy| \le p$ אז קיים $i \ge 0$ כך ש- $xy^iz \notin L$.

<u>דוגמה:</u> $L = \{0^n1^n : n \ge 0\}$ אינה רגולרית.

בהינתן p נתבונן במילה $w = 0^p1^p$. נשים ♡ ש- $w \in L$ ו- $|w| \ge p$. לכל חלוקה $w = xyz$ כך ש- $|y| > 0$ ו- $|xy| \le p$ מתקיים $y \in 0^+$ (כלומר $y \in 0 \cdot 0^*$). לכן לכל $i > 1$ $xy^iz \notin L$ כי יש יותר אפסים מאחדות.

<u>דוגמה:</u> $L = \{w : \#0 = \#1\}$

זו שפה לא רגולרית. אפשר להראות את זה בשתי דרכים.

1) $\{0^n1^n : n \ge 0\} = L \cap 0^*1^*$

מכיוון שהשפות הרגולריות סגורות לחיתוך והכל שכזו ורנו להיות ש-L רגולרית. כי אם כן, היינו מקבלים ש- $\{0^n1^n\}$ הייתה רגולרית ואת זה כבר הוכחנו שלא נכון.

2) נשתמש בלמת הניפוח. בהינתן p נתבונן ב- $w = 0^p1^p$ ונשים לב שכמו קודם, לכל חלוקה $w = xyz$ כך ש- $|y| > 0$ ו- $|xy| \le p$ מביאות ש- $y \in 0^+$ ולכן $xy^iz \notin L$ לכל $i > 1$, כי יש יותר אפסים מאחדות.

<u>דוגמה:</u> $L = \{w \cdot w \; : \; w \in \Sigma^*\}$

בהינתן p נתבונן ב- $w = 0^p 1 0^p 1$. לכל חלוקה
$w = xyz$ המקיימת $|y|>0$ ו- $|xy| \le p$ מתקיים
$y \in 0^+$. ונכון ש- $xz \notin L$.

$$xz = 0^{p-|y|} 1 0^p 1$$

והוא כמובן אינו מהצורה ww.

הערה חשובה: למת הניפוח היא אינה אמ"מ. יש שפות לא רגולריות שמקיימות את למת הניפוח.

<u>אפיון שפות רגולריות</u> (מייהיל לחזור על זה לפני השיעור הבא)

בהינתן שפה $L \subseteq \Sigma^*$ נגדיר יחס $\sim_L \subseteq \Sigma^* \times \Sigma^*$:

$x \sim_L y$ אמ"מ לכל $z \in \Sigma^*$ $xz \in L \Leftrightarrow yz \in L$

המשמעות האינטואיטיבית היא שאין ל-x ול-y זנב מפריד.
כלומר לכל זנב הן בשפה או לא בשפה ביחד.

לדוגמה נסתכל על $(0+1)^*0(0+1)$.

לכל $z \in \Sigma^*$ $11z \in L \Leftrightarrow 111z \in L$

שהרי זה מתקיים אמ"מ $z \in L$. $\Leftarrow$ $11 \sim_L 111$.

לעומת זאת $10 \not\sim_L 11$ כי 1 הוא זנב מפריד:

$$101 \in L \;,\; 111 \notin L$$

ועוד דוגמה $01 \not\sim_L 11$ כי ε זנב מפריד.

ביום שני נראה ששפה היא רגולרית אמ"מ מספר מחלקות השקילות של היחס הזה הוא סופי.

13

19.3.07

חישוביות

אפיון שפות רגולריות

הגדרנו את היחס $\sim_L \subseteq \Sigma^* \times \Sigma^*$:

$x \sim_L y$ אמ"מ לכל $z \in \Sigma^*$ $\quad xz \in L \iff yz \in L$.

דוגמה: $L = (0+1)^* 0 (0+1)$

$11 \sim_L 111$

$10 \not\sim_L 11$ (כי $101 \in L$, $111 \notin L$)

$01 \not\sim_L 11$ (כי ε זנב מפריד)

נשים ♡ ש- $\sim_L$ יחס שקילות:

1) רפלקסיבי: ברור שלכל $x \in \Sigma^*$ מתקיים $x \sim_L x$

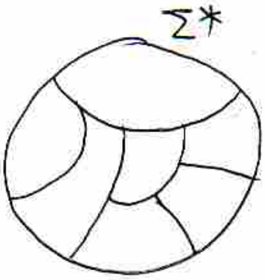
2) סימטרי: אם $x \sim_L y$ ברור שגם $y \sim_L x$.

3) טרנזיטיבי: אם $x \sim_L y$ ו- $y \sim_L w$ אז $x \sim_L w$: נניח בשלילה שיש z כך ש- $wz \in L$ ו- $xz \notin L$ (בלי הגבלת הכלליות)
נתבונן ב- yz. מכיוון ש- $x \sim_L y$ אז $yz \notin L$.
מצד שני, מכיוון ש- $w \sim_L y$ אז $yz \in L$. סתירה.

למה זה טוב יחס שקילות? כי אז יש מחלקות שקילות:

$[w] = \{x \in \Sigma^* : w \sim_L x\}$

Σ^*



אנחנו לא יודעים דבר על המחלקות האלה- מספר, גודל.

נמצא את מחלקות השקילות של $\sim_L$ עבור
$L = (0+1)^* 0 (0+1)$

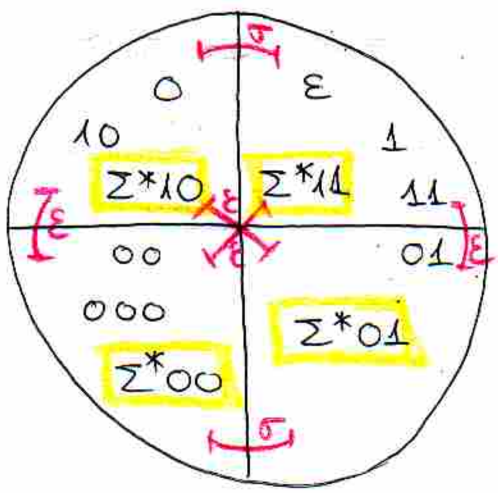
$\varepsilon \not\sim_L 0$ כי $01 \in L$ ו- $1 \notin L$

$1 \sim_L \varepsilon$

$00 \not\sim_L 0$ כי ε זנב מפריד

$01 \not\sim_L \varepsilon$

..



אפשר לשים ♡ שזה תלוי רק ב-2 האותיות האחרונות.

משפט: Myhill - Nerode

מספר מחלקות השקילות של $\sim_L$ סופי אמ"מ L רגולרית.

הוכחה:

($\Rightarrow$) יהי $A=\langle Q,\Sigma,\delta,q_0,F\rangle$ עצמיין חורצני עבור L.
נגדיר יחס $\sim_A\subseteq\Sigma^*\times\Sigma^*$ ע"י $x\sim_A y$ אמ"מ $\delta(q_0,x)=\delta(q_0,y)$
(כלומר האוטומט מגיע לאותו מצב בקריאת x ו-y). זה מוגדר היטב
כי העצמיין חורצני. ברור ש-$\sim_A$ יחס שקילות.
נטען שמספר מחלקות השקילות של $\sim_A$ הוא לפחות כמספר מחלקות השקילות
של $\sim_L$. נראה שאם $x\sim_A y$ אז $x\sim_L y$, כלומר לכל $z\in\Sigma^*$
$yz\in L \Leftrightarrow xz\in L$. אבל זה ברור כי $\delta(q_0,xz)=\delta(q_0,yz)$
שהרי העצמיין חורצני. $\Leftarrow$ $\sim_A$ הוא עידון של $\sim_L$.
כעת נשים לב שמספר מחלקות השקילות של $\sim_A$ הוא $\le |Q|$. למעשה
הוא שווה למספר המצבים הישיגים: זאת מפשט ההגדרה של היחס.
$\Leftarrow$ מספר מחלקות השקילות של $\sim_L$ הוא סופי, (לכל היותר $|Q|$)

($\Leftarrow$) נניח שמספר מחלקות השקילות של $\sim_L$ הוא סופי. נגדיר DFA

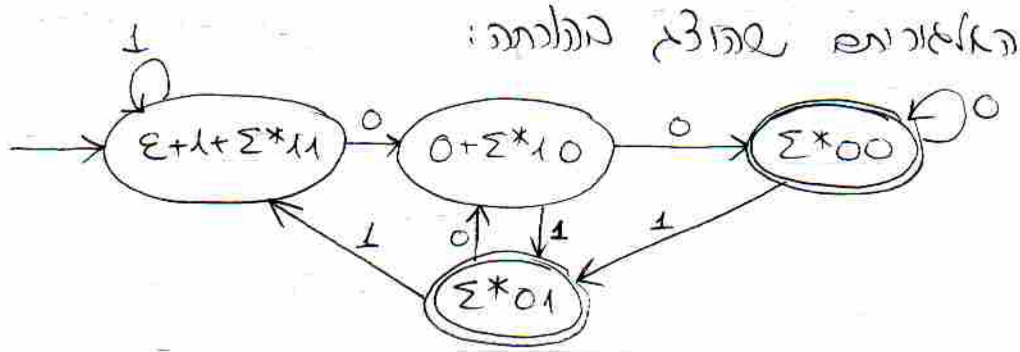
$$A=\langle Q,\Sigma,\delta,q_0,F\rangle$$

כאשר:
- Q - מחלקות השקילות של $\sim_L$.
- $q_0=[\varepsilon]$
- $\delta([x],a)=[xa]$. נשים לב שההגדרה לא תלויה בבחירת הנציג. אם $x\sim_L y$ אז $xa\sim_L ya$ (כי כל מפריד z ל-xa, ya משרה מפריד az ל-x,y).
- $F=\{[w] : w\in L\}$ (מתקיים $\delta(q_0,w)=[w]$)

העצמיין הזה מקבל את L לכן היא רגולרית.

נבנה DFA עבור $L=(0+1)^*0(0+1)$

ראינו שיש ל-L ארבע מחלקות שקילות, אז נבנה את ה-DFA לפי האלגוריתם שהוצג בהרצאה:



זה נקרא האוטומט החסכוני הקנוני של L.

שימושים של משפט Myhill-Nerode

1. הוכחת רגולריות / אי-רגולריות של שפות.

דוגמה: $\{0^n1^n : n\in\mathbb{N}\}$ אינה רגולרית. לכל $i\neq j$ $0^i \not\sim_L 0^j$ שהרי 1^i הוא מפריד. ⟸ יש אינסוף מחלקות שקילות.

2. צמצום אוטומטים חסכוניים - מסתבר שיש דבר כזה אוטומט חסכוני, קנוני מזערי.

$\sim_A$ - יחס מעל מילים ב-Σ^* תלוי במימוש, כלומר ב-A

$\sim_L$ - תלוי רק בשפה. מספר מחלקות השקילות הוא מספר המצבים באוטומט מזערי.

שפות חסרות הקשר
context free languages

כמו ששפה רגולרית מוגדרת ע"י אוטומט, שפה חסרת הקשר מוגדרת ע"י דקדוק חסר הקשר.

$$\boxed{\begin{array}{l} A \to 0A1 \\ A \to B \\ B \to \# \end{array}}$$

דוגמה:

בדקדוק יש:
- משתנים (A,B) + משתנה התחלתי
- טרמינלים $(0,1,\#)$
- חוקי גזירה.

דוגמא לגזירת מילה בשפה:

$A \rightarrow 0A1 \rightarrow 00A11 \rightarrow 000A111 \rightarrow 000B111 \rightarrow 000\#111$

אפשר כבר לנחש ש- $L=\{0^n\#1^n : n\geq 0\}$.

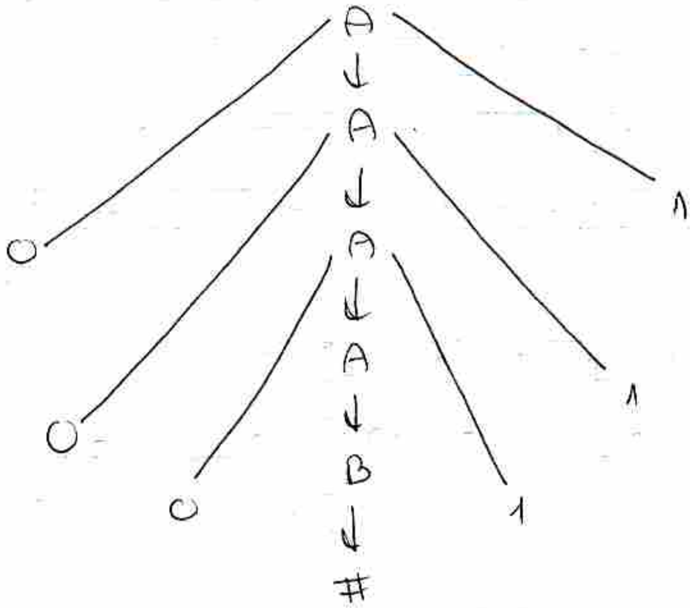
גזירת מילה בשפה:

1) התחל מהמשתנה ההתחלתי

2) כל עוד יש במילה משתנים

- מצא משתנה במילה:
- הפעל את אחד מחוקי הגזירה הרלוונטיים

אפשר לתאר גזירה של מילה ע"י עץ גזירה. למשל עבור 000#111



היתרון של עץ גזירה הוא שרואים בו את כל הגזירה.

CFL = מחלקת השפות הניתנות להגדרה ע"י דקדוק חסר הקשר.

סיפור: היה ויכוח גדול אם שפות טבעיות הן רגולריות או לא.

ובכן, הן לא רגולריות

הנה טיל: missile

והנה נגד טילים anti-missile

והנה טיל נגד טילים" anti-missile missile

וכן הלאה טיל נגד טיל נגד טיל נגד טיל, מילה מילה מילה,

יוצא $a^n b^{n+1}$ וזה לא רגולרי!

הגדרה: דקדוק חסר הקשר הוא רביעיה:

$$G = \langle V, \Sigma, R, S \rangle$$

משתנים, טרמינלים, חוקי גזירה, משתנה התחלתי

אם $w,u,v \in (V \cup \Sigma)^*$ ו- $A \to w$ חוק בדקדוק

אז $uAv \Rightarrow uwv$ "uAv מייצר את uwv".

נאמר ש- $u \stackrel{*}{\Rightarrow} v$ אם יש סדרה $u_1, u_2, \dots, u_k$ עבור

$1 \le k$ רק ש- $u = u_1 \Rightarrow u_2 \Rightarrow u_3 \Rightarrow \dots \Rightarrow u_k = v$

השפה של הדקדוק G היא $L(G) = \{w : S \stackrel{*}{\Rightarrow} w, w \in \Sigma^*\}$.

דוגמה: דקדוק עבור $(0+1)^*0(0+1)$

$V = \{S, A\}$ $\quad S \to A00$

$\Sigma = \{0, 1\}$ $\quad S \to A01$

(אפשר גם לכתוב בקיצור $S \to A00 | A01$)

$A \to \varepsilon | 0A | 1A$

דוגמה: $G = \langle \{S\}, \{a,b\}, R, S \rangle$

$R: S \to aSb | SS | \varepsilon$

למשל בשפה: $abab, aaabbb, aababb$

זוהי שפת הסוגריים המקוננים באופן חוקי.

היא לא רגולרית כי $L(G) \cap a^*b^* = \{a^n b^n : n \ge 0\}$

וחיתוך של שפות רגולריות לא יכול להיות לא רגולרי.

דוגמה: דקדוק עבור $\{0^n 1^n : n \ge 0\} \cup \{1^n 0^n : n \ge 0\}$

$S \to A | B$

$A \to 0A1 | \varepsilon$

$B \to 1B0 | \varepsilon$

21.3.07
חישוביות

דקדוקים חסרי הקשר

שפת הפלינדרומים - $\Sigma=\{a,b\}$

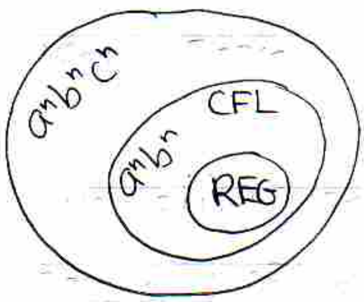
- $S \to aSa \mid bSb \mid a \mid b \mid \varepsilon$

. $L=\{a^nb^nc^n : n\ge 0\}$

נשים ❤ ש- $L=\{a^nb^nc^*: n\ge0\}\cap\{a^*b^nc^n: n\ge 0\}$

אז אם נוכיח ששפות חסרות הקשר סגורות לחיתוך נקבל ש-L חסרת הקשר. אבל, זה לא נכון - שפות חסרות הקשר אינן סגורות לחיתוך. האם זה אומר ש-L אינה שפה חסרת הקשר? לא! אבל במקרה L הזו באמת לא חסרת הקשר ונוכיח את זה בעתיד.

ראינו שיש שפות חסרות הקשר שאינן רגולריות - למשל $\{a^nb^n\}$. האם יש שפה רגולרית שאינה חסרת הקשר? האינטואיציה אומרת שלא.



משפט $REG \subseteq CFL$

הוכחה נראה איך בהינתן עצמיין סופי חורגני אנחנו מגדירים דקדוק חסר הקשר.

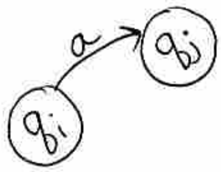
בהינתן $A=\langle Q,\Sigma,q_0,\delta,F\rangle$ נבנה $G=\langle V,\Sigma,R,S\rangle$ כך ש- $L(A)=L(G)$.

__הרעיון__: משתנה V_i עבור כל מצב q_i

השפה הנגזרת מהדקדוק עם משתנה התחלתי V_i =

= השפה המתקבלת ע"י A עם מצב התחלתי q_i

נגדיר את כללי הגזירה R:



- אם $\delta(q_i,a)=q_j$ נוסיף חוק $V_i\to aV_j$
- המשתנה ההתחלתי: V_0 (המתאים ל- q_0)
- אם $q_i\in F$ נוסיף חוק $V_i\to\varepsilon$.

נראה שגם $L(A)=L(G)$. תהי $W=W_1W_2\dots W_n$.

נראה ש- $V_0 \overset{*}{\Rightarrow} W \iff W\in L(A)$:

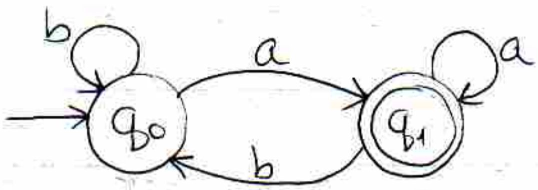
$W\in L(A)$ אמ"מ קיימת ריצה $r_0,r_1,r_2\dots r_n$ כך ש- $r_0=q_0$, $r_n\in F$

ו- $\delta(r_i,w_{i+1})=r_{i+1}$, כלומר $q_0=r_0 \xrightarrow{w_1} r_1 \xrightarrow{w_2} r_2 \xrightarrow{w_3} \dots \xrightarrow{w_n} r_n \in F$

אמ"מ קיימת סדרת גזירות $V_0 \Rightarrow w_1V_1 \Rightarrow w_1w_2V_2 \Rightarrow \dots \Rightarrow w_1w_2\dots w_nV_n \Rightarrow w_1w_2\dots w_n$

אמ"מ $V_0 \overset{*}{\Rightarrow} W$.

דוגמה $L=(a+b)^*a$ כל המילים שנגמרות ב-a



abaa
$q_0\, q_1\, q_0\, q_1\, q_1$

$V_0 \to aV_1 \to abV_0$
$\to abaV_1 \to$
$\to abaaV_1 \to abaa$

$$V_0 \to aV_1 \mid bV_0$$
$$V_1 \to aV_1 \mid bV_0 \mid \varepsilon$$

תרגיל מעניין: לבנות דקדוק שגוזר את המילה מהסוף להתחלה.

הגדרה דקדוק לינארי ימני הוא דקדוק שבו כל החוקים מהצורה $V\to\varepsilon$ או $V\to aV'$

באופן דומה דקדוק לינארי שמאלי הוא דקדוק שבו כל החוקים מהצורה $V\to\varepsilon$ או $V\to V'a$

מסקנה שפה היא רגולרית אמ"מ יש לה דקדוק לינארי ימני

הוכחה

($\Leftarrow$) ראינו בהוכחה איך בונים דקדוק לינארי ימני

($\Rightarrow$) פשוט נבנה אוטומט לפי הכללים שהוצגו בהוכחה.

(17)

רבוי משמעויות

Ambiguity

The girl touches the boy with the flower

לא ברור אם הילד הוא עם פרח

או שהילדה עם פרח.

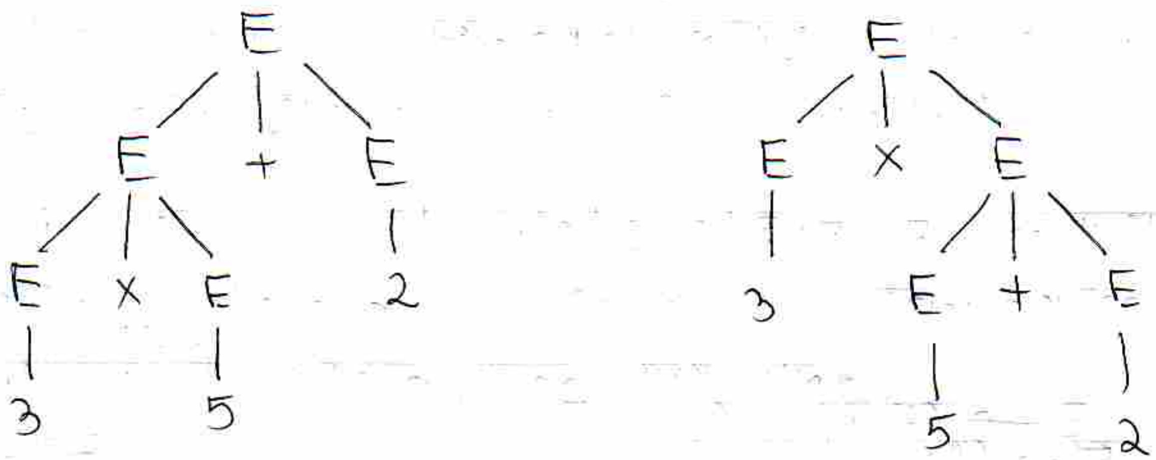
אולי כאן קשה לנו להבין מה הבעיה, אבל אותה הבעיה

בדיוק יש גם בביטוי הזה: $3\times5+2$

לא ברור אם הכוונה היא ל- $(3\times5)+2$ או ל- $3\times(5+2)$.

נכתוב דקדוק לביטויים חשבוניים $\Sigma=\{0,\dots,9,+,\times\}$

$$E\to E+E\mid E\times E\mid 0\mid 1\mid\dots\mid 9$$



דקדוק רב משמעי: יכול לייצר שני עצי גזירה שונים לאותה מילה.

נשים ♥: זה שונה מההגדרה הפורמלית קיום שתי גזירות שונות)

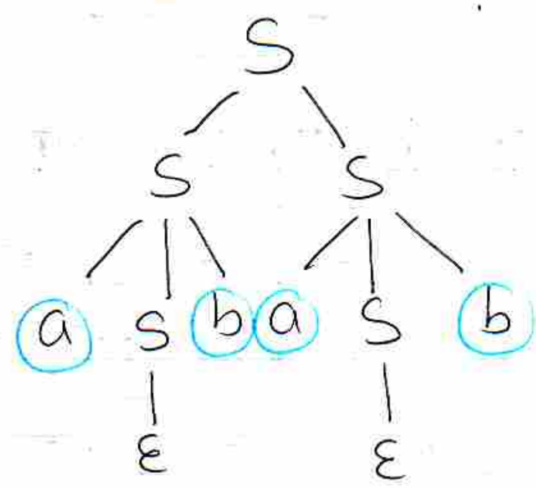
למשל, $S\to aSb\mid SS\mid\varepsilon$ שפת הסוגריים המאוזנים

יש שתי גזירות ל- abab

$$S\to SS\to aSbS\to aSbaSb\to abaSb\to abab$$

$$S\to SS\to aSbS\to abS\to abaSb\to abab$$

אבל, לשתי הגזירות יש את אותו העץ:



הגדרה: נאמר שהגזירה של w היא שמאלית ביותר אם בכל צעד בגזירה מוחלף המשתנה השמאלי ביותר.

דוגמה: הגזירה השנייה של abab מהדוגמה הקודמת היא שמאלית ביותר.

הגדרה: נאמר ש-w נגזרת באופן רב משמעי אם יש ל-w שתי גזירות שמאליות ביותר (שונות ומינימליות)

ראינו ש- $E \to E \times E \mid E+E \mid 0 \mid 1 \mid \dots \mid 9$ רב משמעי.
נשארת השאלה האם קיים דקדוק שקול שהוא לא רב משמעי.

<u>ניסיון 1</u>:

$$E \to (E \times E) \mid (E+E) \mid 0 \mid \dots \mid 9$$
$$(\to \varepsilon$$
$$) \to \varepsilon$$

<u>ניסיון 2</u>:

$$E \to P \mid S \mid 0 \mid \dots \mid 9$$
$$S \to E+E$$
$$P \to E \times E$$

<u>ש"ב</u>: לבדוק בבית אם הניסיונות האלה טובים או לא.

(18) 26.3.07
תשובות

יום הולדת שמח לעצמי!

שבוע שעבר דיברנו על ריבוי משמעויות.

ראינו דקדוק שמגדיר ביטויים חשבוניים

$$E \to E+E \mid E\times E \mid 0 \mid 1 \mid \dots \mid 9$$

אז למשל $3\times5+2$ לא ברור אם זה $(3\times5)+2$ או $3\times(5+2)$

זה שהדקדוק הספציפי הזה הוא לא חד משמעי זה לא אומר שלא קיים
דקדוק חד משמעי לשפת הביטויים החשבוניים.

היו לנו שני ניסיונות בשבוע שעבר אבל שניהם לא היו טובים - בשניהם
אי אפשר היה להבדיל בין שני הביטויים למעלה.

לשפה הנ"ל יש דקדוק חד משמעי שהוא אסוציאטיבי מימין - כלומר
הסוגריים תמיד מתרכזים מצד ימין:

$$E \to D+E \mid D\times E$$
$$D \to 0 \mid 1 \mid \dots \mid 9$$

<u>הגדרה</u>: דקדוק הוא רב-משמעי אם יש מילה w עם שתי גזירות שמאליות
ביותר שונות.

<u>הערה</u>: קיימות שפות שכל דקדוק שגוזר אותן הוא רב משמעי.

<u>תזכורת</u>: $L=\{0^n1^n2^n : n\geq 0\} \notin CFL$

נתבונן ב- $L'=\{0^i1^j2^k : i=j \vee j=k\}$

ברור שמתקיים $L\subseteq L'$ (רואים $i=j=k$) ורואים $L'\not\subseteq L$

שהרי $01\in L'$ אבל $01\notin L$.

$$L'=\{0^n1^n2^* : n\geq 0\} \cup \{0^*1^n2^n : n\geq 0\}$$

אלה הן שתי שפות חסרות הקשר $\Rightarrow$ האיחוד שלהן L' חסרת הקשר.

והנה דקדוק עבורה:

$$S \to S_0^1 \mid S_0^2$$
$$S_0^1 \to T_0^1 R \qquad S_0^2 \to L T_0^2$$
$$T_0^1 \to 0T_0^1 1 \mid \varepsilon \qquad T_0^2 \to 1T_0^2 2 \mid \varepsilon$$
$$R \to 2R \mid \varepsilon \qquad L \to 0L \mid \varepsilon$$

הדקדוק הזה הוא רב-משמעי: יש שתי גזירות שמאליות שונות של מילה ב-L.
אבל גם לא קיים בכלל דקדוק חד משמעי לשפה הזאת כי דקדוק כזה "דומה" לדקדוק זמור L אשר אינה חסרת הקשר.

עצמיון מחסנית pushdown automata

זה מאוד דומה לעצמיון סופי תורגני אלא שלעצמיון מחסנית יש מחסנית בעומק לא חסום שמשמשת כזיכרון שאנחנו יכולים לכתוב לראש שלו ולקרוא ממנו.

דוגמה: PDA עבור $\{0^n1^n : n\geq 0\}$

הרעיון:
- בקריאת 0, נדחוף 0 למחסנית
- בקריאת 1, נשלוף 0 מהמחסנית
- בנוסף צריך לשמור על המבנה 0^*1^*

הגדרה: עצמיון מחסנית הוא שישיה $A=\langle Q,\Sigma,\Gamma,\delta,Q_0,F\rangle$

כאשר
- Q קבוצת מצבים סופית
- Σ א"ב של השפה
- Γ א"ב של המחסנית
- $\delta: Q\times(\Sigma\cup\{\epsilon\})\times(\Gamma\cup\{\epsilon\})\to 2^{Q\times(\Gamma\cup\{\epsilon\})}$

$\langle q',\gamma'\rangle\in\delta(q,\sigma,\gamma)$ - אם A במצב q, קורא σ והאות בראש המחסנית היא γ, אז A עובר למצב q', שולף את γ, דוחף במקומה את γ'.
- $Q_0\subseteq Q$ מצבים התחלתיים
- $F\subseteq Q$ מצבים מקבלים

קונפיגורציה של A היא המצב שבו A נמצא והמצב של המחסנית

$s=abc\in\Gamma^*$ $\quad\langle q,s\rangle\xrightarrow{\sigma}\langle q',s'\rangle$

$\langle q',s'\rangle$ היא σ-עוקבת של $\langle q,s\rangle$ אם A במצב q ותוכן המחסנית הוא s, אז יש אפשרות לקרוא σ, לעבור ל-q' ולעדכן את המחסנית ל-s'. כלומר s ניתן לחלוקה $s=at$ כאשר $a\in\Gamma\cup\{\epsilon\}$, $t\in\Sigma^*$

רק ש- $\langle q',b\rangle \in \delta(q,\sigma,a)$. ו- $s'=bt$

<u>ריצה</u> של A על מילה $w\in\Sigma^*$ היא סדרה של קונפיגורציות $c_0,\dots,c_m$ רק ש- w ניתנת לחלוקה כ- $w_1\dots w_m$

$(w_i\in\Sigma\cup\{\epsilon\})$ ו:

1) $c_0=\langle q_0,\epsilon\rangle$ עבור $q_0\in Q_0$

2) לכל $0\le i\le m-1$ הקונפיגורציה c_{i+1} היא w_{i+1} עוקבת ל- c_i.

<u>קונפיגורציה התחלתית</u> היא $\langle q,\epsilon\rangle$ עבור $q\in Q_0$

<u>קונפיגורציה מקבלת</u> היא $\langle q,s\rangle$ עבור $q\in F$

הריצה $c_0,\dots,c_m$ <u>מקבלת</u> אם c_m מקבלת

<u>דוגמה:</u> אפיון עבור $\{0^n1^n : n\ge 0\}$

$A=\langle Q,\Sigma,\Gamma,Q_0,\delta,F\rangle$

$Q=\{q_1,q_2,q_3,q_4\}$

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

$\Gamma=\{0,\$\}$ אנו שמים סימון תחתית למחסנית

$F=\{q_1,q_4\}$

1) $\delta(q_1,\epsilon,\epsilon)=\{\langle q_2,\$\rangle\}$ מאתחלים את המחסנית

לא מתקדמים בקריאת הקלט

לא שולפים דבר מהמחסנית

דוחפים $\$$ למחסנית

$\langle q_2,\$\rangle$ היא ϵ- עוקבת ל- $\langle q_1,\epsilon\rangle$.

2) $\delta(q_2,0,\epsilon)=\{\langle q_2,0\rangle\}$

כל עוד מגיעים אפסים דוחפים 0 למחסנית בלי קשר למה שיש בה.

$\langle q_2,00\$\rangle$ היא 0- עוקבת ל- $\langle q_2,0\$\rangle$

3) $\delta(q_2, 1, 0) = \{\langle q_3, \varepsilon\rangle\}$

יש לו 2 שימושים
- הגבלה: אפשר לקרוא 1 רק אם יש במחסנית 0
- שליפה: כל פעם שקוראים 1 שולפים 0

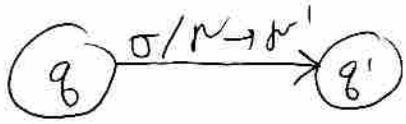
לא דוחפים כלום

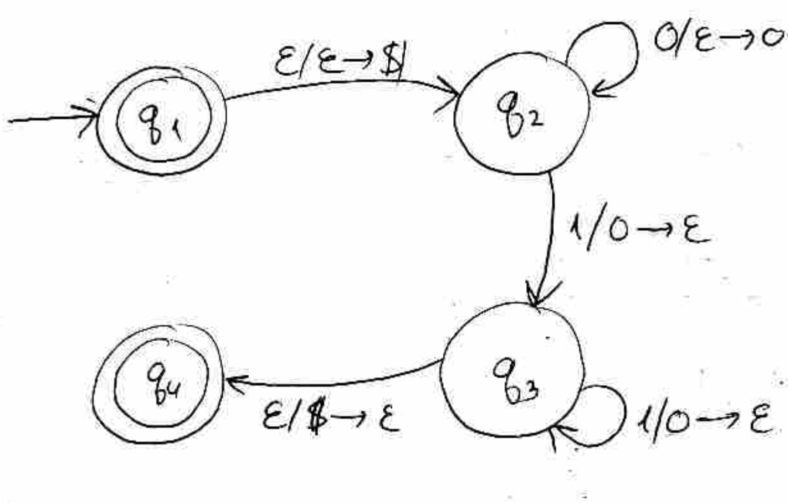
4) $\delta(q_3, 1, 0) = \{\langle q_3, \varepsilon\rangle\}$

נשים ♡ שלא הגדרנו $\delta(q_3, 1, \$)$ - זה בגלל שהאוטומט לא מוכן לקבל מילים כאלה - שיש בהן יותר מדי 1ים.

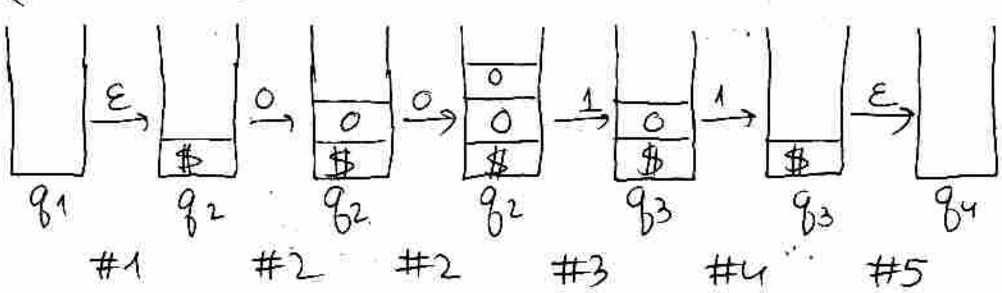
5) $\delta(q_3, \varepsilon, \$) = \{\langle q_4, \varepsilon\rangle\}$

נצייר את האוטומט כאשר $\langle q', \mu'\rangle \in \delta(q, \sigma, \mu)$ מצויר ע"י





ריצה על 0011 (בעצם הריצה היא על $\varepsilon 0011 \varepsilon$)



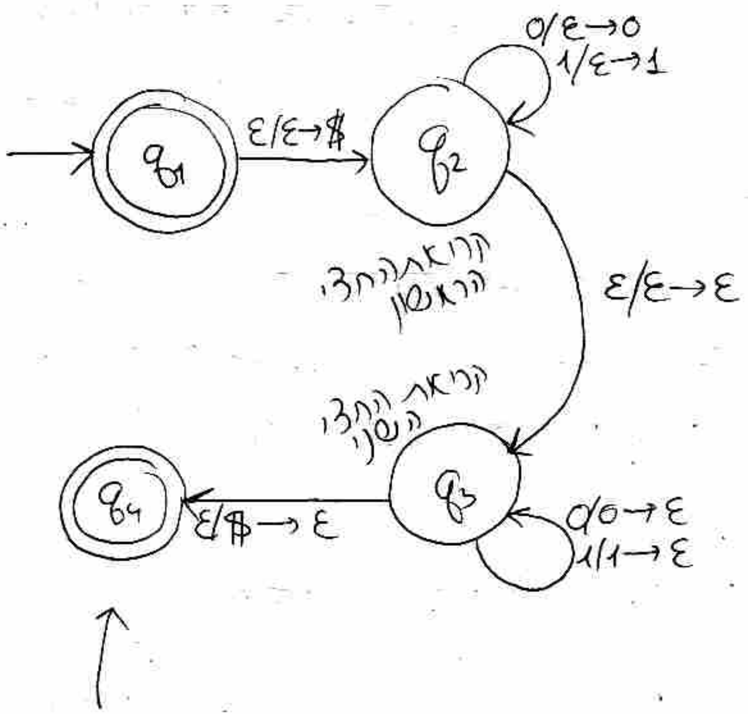
די ברור למה האוטומט הזה הוא אכן מתאים לשפה.
אם מילה בשפה אז יש לה ריצה בדומה ל 0011 והיא נגמרת במצב מקבל.
ואם ריצה נגמרת במצב מקבל אז זה או q_1 או q_4, אם זה q_1 אז המילה היא ε ואם זה q_4 אז מספר ה-0-ים חייב להיות זהה למספר ה-1-ים.

<u>דוגמה:</u> $L = \{w \cdot w^R : w \in \Sigma^*\}$

שפת הפלינדרומים באורך זוגי.

$$S \to 0S0 \mid 1S1 \mid \varepsilon$$

<u>הרעיון:</u> נדחוף את האותיות הנקראות למחסנית. כשננחש שהגענו לאמצע, נוציא אותיות מהמחסנית ונוודא שהן מסכימות עם אותיות הקלט
(כאן חוסר הדטרמיניות כבר מתחיל להיות חשוב).



אם במצב הזה מגיעים סוף אותיות אז האוטומט נתקע ולא מקבל את המילה.

ריצה מקבלת של 011110 :

$(q_1,\varepsilon) \xrightarrow{\varepsilon} (q_2,\$) \xrightarrow{0} (q_2,0\$) \xrightarrow{1} (q_2,10\$) \xrightarrow{1} (q_2,110\$)$

$\xrightarrow{\varepsilon} (q_3,110\$) \xrightarrow{1} (q_3,10\$) \xrightarrow{1} (q_3,0\$) \xrightarrow{0} (q_3,\$) \xrightarrow{\varepsilon} (q_4,\varepsilon)$

ריצה שמנחשת שהמילה מחולקת כך $01|1110$ תגיע למצב $(q_2,10\$)$ משם תנחש שזה האמצע ותגיע ל- $(q_3,0\$)$ ואז היא תיתקע.

<u>דוגמה</u> $L=\{ww : w\in\Sigma^*\}$

מילים באורך זוגי כך שהחצי הראשון שווה לחצי השני.

השפה הזאת אינה חסרת הקשר

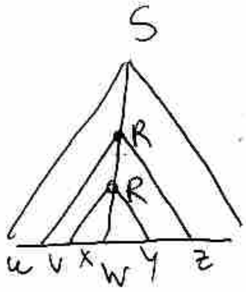
<u>למת הניפוח לשפות חסרות הקשר</u> אם L שפה חסרת הקשר אז קיים $p\geq 1$ (קבוע הניפוח) כך שלכל $w\in L$ כך ש $|w|\geq p$ קיימת חלוקה $w=uvxyz$ המקיימת:

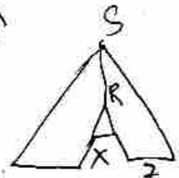
1) לכל $i\geq 0$ $uv^ixy^iz\in L$

2) $|vy|>0$ אומר לא יתכן ששניהם ε

3) $|vxy|\leq p$

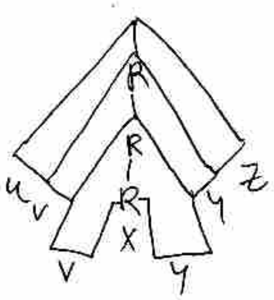
<u>הרעיון</u> נתבונן במילה w ארוכה. אם $w\in L$ יש לה עץ גזירה. $\Leftarrow$ יש מסלול מ-S לטרמינל ובמסלול שהוא ארוכה יש משתנה R שחוזר במסלול לפחות פעמיים. נתבונן בחלוקה המושרית מהעץ (כמו בציור משמאל) ונראה שהיא מקיימת את כל התנאים.





- עבור $i=0$ $uxz\in L$ הנה עץ הגזירה שלו

$i>1$ (סתם עוד עץ).

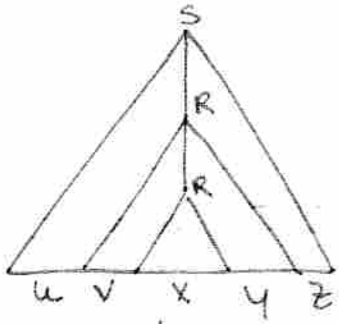


נותר להשלים את הפרטים הקטנים.

(21) 28.3.07
חישוביות

- למת הניפוח לשפות חסרות הקשר -

אם L שפה חסרת הקשר, יש $p \ge 1$ כך שלכל מילה w, אם $|w| \ge p$
אז קיימת חלוקה $w = uvxyz$ כך ש-



איור 1.

(1) לכל $0 \le i$ $uv^ixy^iz \in L$
(2) $|vy| \ge 1$
(3) $|vxy| \le p$

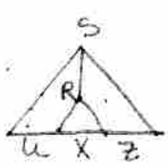
<u>הוכחה</u>: יהי b האורך המקסימלי (מספר האותיות ב- $\Sigma \cup V$) של צד ימין של חוק בדקדוק המגדיר את L. נניח $b \ge 2$ (אחרת כל המילים בדקדוק הן באורך 1 או 0 ואז הטענה ברורה).

בעץ הגזירה של w, לכל צומת יש b בנים לכל היותר. נתבונן בעץ גזירה T. אם $height(T)=h$ אז $\#leaves(T) \le b^h$
אם $height(T)<h$ אז $\#leaves(T)<b^h$ ואם $height(T) \ge h$ אז $\#leaves(T) \ge b^h$. (נבחר $p = b^{|V|+1}$. אזי גובהו של כל עץ גזירה של w כך ש- $|w| \ge p$ הוא לפחות $|V|+1$.

נראה חלוקה של w כמו $w=uvxyz$ המקיימת את תנאי הלמה.
במסלול מ-S לטרמינל, כל הצמתים פרט לאחרון מסומנים במשתנים.
יש $|V|+1$ צמתים $\Leftarrow$ מעיקרון שובך היונים יש משתנה שחוזר על עצמו.
יהי T עץ גזירה מינימלי עבור w (כלומר אין ל-w עץ גזירה עם פחות צמתים). גובהו של T לפחות $|V|+1$. במסלול מ-S לטרמינל שאורכו לפחות $|V|+1$ נבחר משתנה R שחוזר ב- $|V|+1$ הרמות התחתונות.

ניקח חלוקה כמתואר באיור 1. נראה שכל התנאים מתקיימים.

(1) לכל $i \ge 0$ $uv^ixy^iz \in L$.
- $i=0$ הנה עץ גזירה של uxz:



- $i>1$ אפשר להמשיך לגזור R מ-R.

② $|vy|>0$ אם $v=\varepsilon=y$ אז כן סתירה למינימליות של τ

③ $|vxy|\le p$ מכיוון שבחרנו את R ראשונה שחוזר ב- $|V|+1$ הרמות האחרונות אז vxy (נוצר מ-R בעץ שגובהו לכל היותר $|V|+1$ $\Leftarrow$ $|vxy|\le b^{|V|+1}=p$.

- שימוש בלמת הניפוח -

כדי להראות ש-L אינה חסרת הקשר: לכל $p\ge 1$ יש מילה w כך ש- $|w|\ge p$ ולכל חלוקה $w=uvxyz$ אם $|vy|>0$ ו- $|vxy|\le p$, אז קיים i כך ש- $uv^ixy^iz\notin L$.

דוגמה: $L=\{a^nb^nc^n : n\ge 0\}$

בהינתן p (הנתון בלמה) המילה $w=a^pb^pc^p\in L$. כמובן $|w|\ge p$. (נראה שכל חלוקה לא טריוויאלית אפשר לנפח ולצאת מהשפה. נניח ש- $w=uvxyz$ כך ש- $|vy|>0$ ו- $|vxy|\le p$. אזי ב- vxy יש לכל היותר 2 אותיות שונות ולכן אם מנפחים את הקבוצות שלו, הניפוח יפגע את האות השלישית.

דוגמה: $L=\{ww : w\in\Sigma^*\}$ $\Sigma=\{0,1\}$

בהינתן p (הנתון ב- $w=0^p10^p1$

יש חלוקה טובה: $\underbrace{0^{p-1}}_{u}\underbrace{0}_{v}\underbrace{1}_{x}\underbrace{0}_{y}\underbrace{0^{p-1}1}_{z}$ - מתקיימים שלושת התנאים!!

אנחנו בבעיה. נמצא מילה אחרת: $0^p1^p0^p1^p$.

פירוקים אפשריים:

① כש vxy בחצי הראשון של w $\Leftarrow$ האות אחרי האמצע של uv^2xy^2z הוא 1.

② כש vxy בחצי השני של w $\Leftarrow$ האות לפני האמצע היא 0.

③ כש vxy מכיל את האמצע $\Leftarrow$ uxz מהצורה $0^p1^i0^j1^p$ עבור $i+j<2p$.

(22) 25.5.07
חישוביות

כמעט בטוח שהמבחן הפעם יכלול שלושה שבועות קדימה.
בקשר לחומר, הקורס נחלק ל-3 חלקים: - אוטומטים
- חישוביות
- סיבוכיות

לפני השביתה היינו בפתיחה של הוכחה שדקדוק חסר הקשר שקול לאוטומט מחסנית. אנחנו נדלג על ההוכחה. ההוכחה לא פשוטה. היא נמצאת בספר וכדאי לעיין בה.
חוץ מזה בד"כ השבועיים האחרונים מכיוון שיש חומר שהוא העשרה, אז זה הקורבן הידוע של השביתה וחוץ מזה נספק את כל מה שצריך.
בזאת נגמר החומר של אוטומטים.

הודעה:
יום שישי 9-11
שיעור השלמה
כמעט בטוח
שבכימיה 7.
תהיה הודעה באתר הקורס.

תורת החישוביות

ראינו שני מודלי חישוב:
NDFA - שפות רגולריות
PDA - שפות חסרות הקשר

מבחינה כזו היו סוגי שאלות:
- איך לתאר דקדוק של שפה?
- שאלות בעלות אופי של חסם תחתון על הגודל של האוטומט
- להוכיח ששפה היא לא חסרת הקשר/רגולרית

המודל הבא שנלמד הוא מכונת טיורינג. זהו כבר הדבר האמיתי. מכונת טיורינג עושה כל מה שמחשב יכול לעשות ואז כבר השאלות יהיו מעניינות יותר קשורות למציאות. האם בעיה נתונה לפתרון במחשב? האם האלגוריתם חייב להיות אקספוננציאלי?

<u>דוגמה:</u> נסתכל על השפה $L = \{w\#w : w \in (0+1)^*\}$.
זאת לא שפה חסרת הקשר (אפשר להוכיח עם למת הניפוח). בפרט היא גם לא רגולרית.

אבל קל מאוד לכתוב תכנית שמקבלת כקלט מילה מעל $\Sigma=\{0,1,\#\}$ ופולטת "כן" אמ"מ המילה ב-L. למשל אפשר לעשות את זה עם שני מצביעים שעוברים על המילה, או שאפשר לעבור עליה כמה פעמים. וזה בדיוק הדברים שהיו חסרים לנו במודלים הקודמים.

המודל: מכונת טיורינג (Alan Turing 1936)

למכונה יש כל מיני מאפיינים חדשים:

- סרט עבודה אינסופי
- יכולת לקרוא/לכתוב מ/על הסרט
- יכולת להזיז את הראש הקורא ימינה ושמאלה.
- מצבי קבלה ודחייה

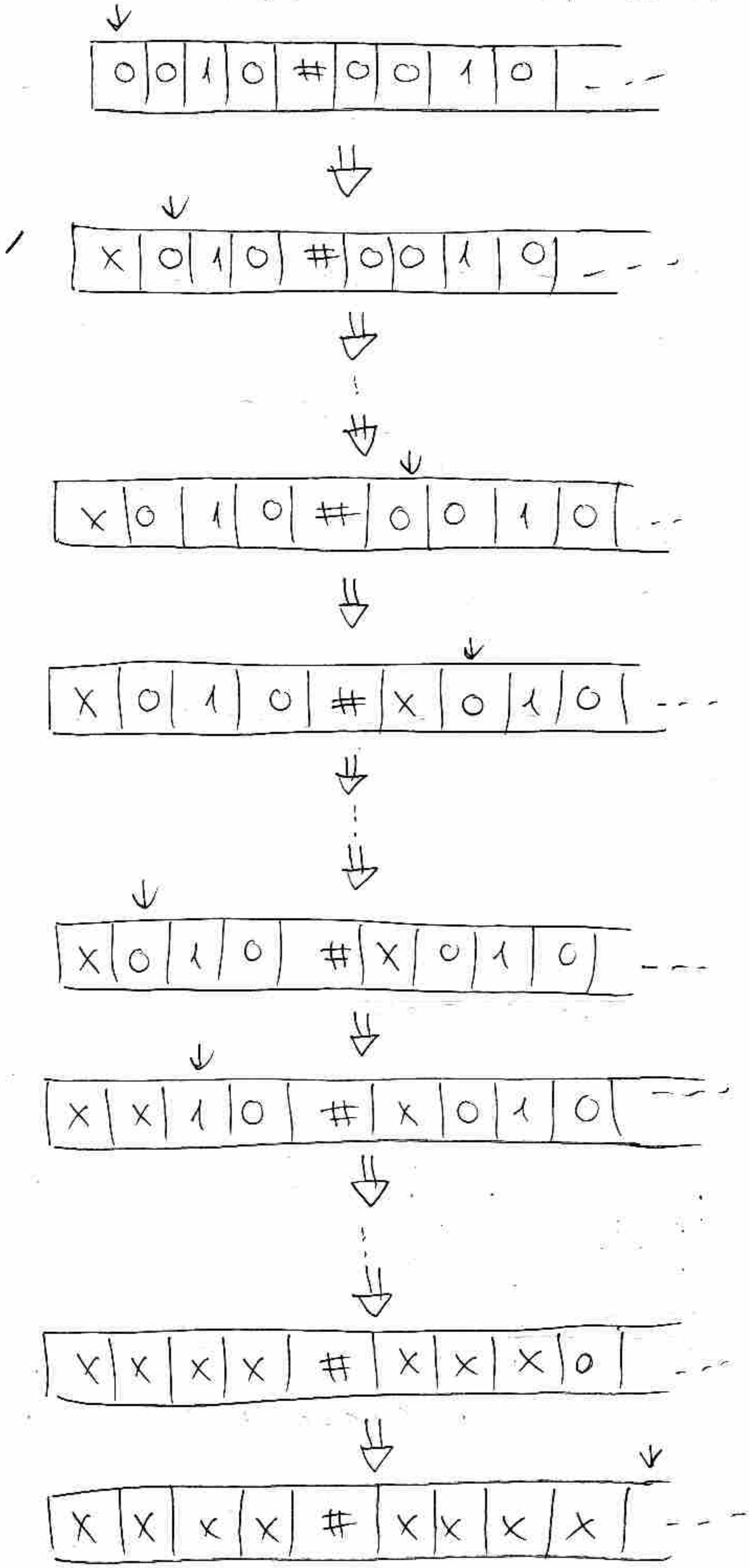
בכל שלב של החישוב היא קוראת את הסרט, זזה ימינה או שמאלה וכותבת במקום את מה שהיה.

דוגמה מכונת טיורינג (להלן מ"ט) עבור L

M תפעל כך:

- סורקת את הקלט ומוודאת שיש # אחת בדיוק, ואם לא, דוחה את המילה.
- זיגזג בין מיקומים תואמים בסרט הקלט תואמים:
  1) התא הראשון והתא הראשון אחרי ה-#
  2) (אח"כ) התא הראשון שלא מחוק והתא הראשון שלא מחוק אחרי ה-#.
- אם הם מסומנים באותה אות, מוחקת אותם וממשיכה, אחרת דוחה.
- כשנמחקו כל התאים משמאל ל-# בודקת האם יש תאים שלא מחוקים מימין. אם יש דוחה, אחרת מקבלת.

דוגמה לריצה של M על 0010#0010.



ואז היא רואה שאין עוד ומחליטה שהיא מקבלת!

הגדרה פורמלית של מכונת טיורינג

מכונת טיורינג היא שביעייה

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej} \rangle$$

כאשר:

- Σ – א"ב הקלט ואיננו כולל את האות "$\sqcup$" (רווח)
- Γ – א"ב העבודה כאשר $\Sigma \subseteq \Gamma$ ו- $\sqcup \in \Gamma$
- δ – פונקציית מעברים $\delta: Q\times\Gamma \to Q\times\Gamma\times\Delta$

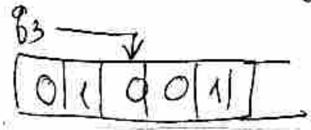
כאשר $\Delta = \{L, R\}$

אם $\delta(q,a) = \langle q', b, L\rangle$ זה אומר שכאשר M
במצב q וקוראת a, אז M עוברת למצב q',
כותבת b במקום a וזזה עם הראש הקורא
תא אחד שמאלה.

זה במקרה שהמכונה דטרמיניסטית. אם היא לא
דטרמיניסטית אז $\delta: Q\times\Gamma \to 2^{Q\times\Gamma\times\Delta}$

<u>קונפיגורציה</u> של מ"ט היא – המצב
- תוכן הסרט
- מיקום הראש הקורא

בהינתן מצב q ומילים $u,v\in\Gamma^*$ uqv היא
קונפיגורציה שבה: - המצב q
- תוכן הסרט uv
- הראש הקורא מצביע על האות הראשונה של v.



למשל אם אנחנו במצב הבא
אז הקונפיגורציה היא $01q_3001$.

<u>קונפיגורציה התחלתית</u> היא $\sqcup\sqcup\sqcup\, q_0 w \sqcup\sqcup\sqcup$ (על מילת קלט w)

<u>בקונפיגורציה סופית</u> המצב הוא q_{acc} (ואז היא מקבלת)
או שהמצב הוא q_{rej} (ואז היא דוחה)

קונפיגורציות עוקבות

יהיו $a,b,c \in \Gamma$, $v,u \in \Gamma^*$, $q,q' \in Q$

יש שתי אופציות לקונפיגורציות עוקבות

1) אם $\delta(q,b)=(q',c,L)$ אז

$$(*)\quad uaqbv \to uq'acv$$

2) אם $\delta(q,b)=(q',c,R)$ אז

$$(**)\quad uaqbv \to uacq'v$$

יש מקרה פרטי שהראש הקורא מצביע על התא השמאלי אז הראש לא זז ויהיה $qbv \to q'cv$ או

$$qbv \to cq'v$$

M מקבלת את w אם קיימת סדרת קונפיגורציות $C_0C_1\ldots C_k$ כך ש- C_0 התחלתית (q_0w)

C_k מקבלת (המצב הוא q_{acc})

לכל $i \geq 0$ הקונפ' C_{i+1} עוקבת ל- C_i

השפה של M היא $L(M)$ - שפת כל המילים ש- M מקבלת.

נאמר ש- M מזהה את L אם $L(M)=L$.

שפה L היא ניתנת למניה רקורסיבית (recursively enumerable) (להלן RE או נל"ן) אם קיימת מ"ט המזהה את L.

נשים ♡: יש 3 גורלות אפשריים לריצה:

1) מגיעה לקונפ' מקבלת
2) מגיעה לקונפ' דוחה
3) נכנסת ללולאה אינסופית (לא עוצרת)

נאמר ש- M מכריעה את L אם M מזהה את L ובנוסף M עוצרת על כל הקלטים. זאת כמובן דרישה חזקה יותר מנק' זיהוי, כי בכלל לא מובטח שמכונות עוצרות. וזה בגלל שבעיית העצירה אינה כריעה. המכונה לא יכולה לדעת שהיא נכנסה ללולאה אינסופית.

שפה L היא <u>רקורסיבית</u> (R) אם קיימת מכונת טיורינג המכריעה אותה.

ברור ש- $R \subseteq RE$. נראה בעתיד ש- $RE \nsubseteq R$ (בגלל בעיית העצירה).

<u>דוגמה</u>: נצייר מ"ט M שמכריעה את $L=\{0^{2^n} : n \geq 0\}$

הרעיון: נגדיר predicate מעל הטבעיים

$good(k) \Leftrightarrow \exists n \in \mathbb{N}\ k=2^n$ (שימו ♡ ש-

$good(k) \leftrightarrow (k=1 \vee good(\frac{k}{2})) \wedge k \neq 0$

לכן M תפעל כך:

א) סרוק את הסרט משמאל לימין ומחק כל 0 שני...

ב) אם היה בקלט בשלב א' רק 0 אחד, קבל

ג) " " מספר אי-זוגי של אפסים, דחה

ד) תחזיר את הראש הקורא לתחילת הסרט

ה) חזור לשלב א'

נתאר את M פורמלית:

$$M = \langle Q, \Sigma, \Gamma, q_1, q_{acc}, q_{rej} \rangle$$

$$\Sigma = \{0\}$$

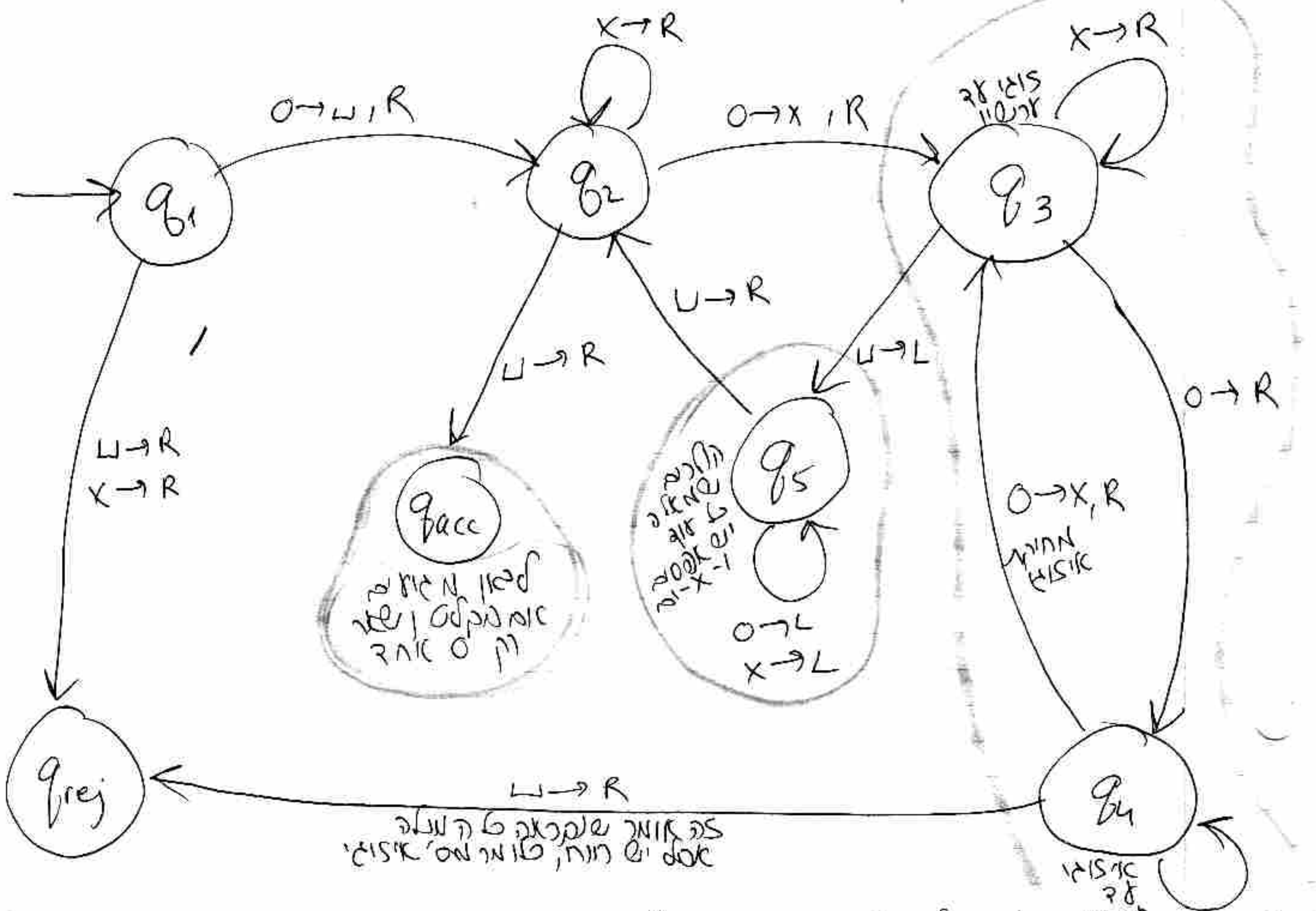
$$\Gamma = \{0, \sqcup, x\}$$

$$Q = \{q_1, \dots, q_5, q_{acc}, q_{rej}\}$$

איך מציירים את המצבים של המכונה?

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

$q \xrightarrow{a \to b, \Delta} q'$ $\qquad$ $q \xrightarrow{a \to \Delta} q'$



הערה: נשים ♡ שהמכונה באמת דטרמיניסטית ופונק' המעברים שלמה (כלומר התחום הוא $Q\times\Gamma$).

תכונות:

RE - יש מכונת טיורינג שמזהה

R - יש מכונת טיורינג שמכריעה.

עכשיו נגדיר סוג

co-RE - כל השפות L כך ש- $\Sigma^*\setminus L\in RE$

כלומר $L\in RE$ אמ"מ $\bar{L}\in co\text{-}RE$

<u>משפט:</u> $R = RE\cap co\text{-}RE$

1) אם $L\in R \Rightarrow L\in RE$ כמובן

אבל גם $L\in co\text{-}RE$ (כי נקח את מכונת הטיורינג שמכריעה את L, נחליף את q_{acc} ב- q_{rej} ונקבל מ"ט שמכריעה את $\bar{L}$).

(26) 30.5.07
תשעיות

המחלקות RE ו- R

ריצה של מ"ט היא סדרה $c_0, c_1, \ldots$ של קונפיגורציות כך
ש- 1) c_0 היא קונפיגורציה התחלתית
2) c_{i+1} נובעת מ-c_i לכל $i \geq 0$
3) הסדרה אינסופית (אינה מכילה קונפיגורציה סופית)
או הסדרה סופית ומסתיימת בקונפיגורציה סופית.

- $L \in RE$ אמ"מ קיימת מ"ט המזהה (מקבלת את כל המילים בשפה ורק אותן) את L.
- $L \in R$ אמ"מ קיימת מ"ט המכריעה (עוצרת ע"כ הקלט ומזהה) את L.
- $L \in co\text{-}RE$ אמ"מ $\Sigma^* \setminus L \in RE$

<u>משפט:</u> $RE \cap co\text{-}RE = R$

<u>הוכחה:</u>
1) $L \in R \Rightarrow$ נראה שיש מ"ט M שמזהה את L וגם מ"ט M' שמזהה את $comp(L)$. מההנחה קיימת M שמכריעה את L. היא כמובן גם מזהה את L. M' תתקבל ע"י החלפת q_{acc} ו- q_{rej}. נשים לב ש-M' מכריעה את $comp(L)$. (למעשה הוכחנו פה ש- R סגור תחת השלמה).
2) נתון שיש מ"ט M שמזהה את L ויש M' שמזהה את $comp(L)$. נבנה T שמכריעה את L. בהינתן קלט w, יודעים ש- M עוצרת על w או M' עוצרת על w. עבור $i=1,2,\ldots$ T מריצה את M על w i צעדים. אם M קיבלה, עוצרת ומקבלת את w. אם M לא קיבלה, מריצה את M' על w i צעדים. אם M' קיבלה אז T עוצרת ודוחה את w. אם M' לא קיבלה מגדילים את i. הלולאה הלא נגמרת כל אינסופית כי מובטח לנו ש- M או M' בסופו של דבר תעצורנה. (!!)

הערה: לא תיארנו בהוכחה מ"ט T. תיארנו אלגוריתם הנראה טריוויאלי שברור לנו איך לממש אותו ב-מ"ט. אנחנו נניח שמ"ט שקולה למחשב ולכן באמת ההוכחה הלא תהיה חסרה לגמרי.

הגדרה: ספרן (Enumerator) הוא מ"ט עם מדפסת E והיא מייצרת רשימה של מילים (אלה שהיא מדפיסה). ייתכן שהיא לא תעצור לעולם והשפה של E היא כל המילים שיודפסו באופן של ספר ורק הן. אין שום דרישה על הסדר - מילה יכולה להיות מודפסת כמה פעמים וזה לא משנה לנו.

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

הוכחה: בהינתן E נייצר M שמקבלת את L.
על קלט w M מריצה את E. בכל פעם ש- E מדפיסה מילה x, M בודקת האם $w=x$. אם כן, M עוצרת ומקבלת. (נשים לב ש-M רק מקבלת, היא לא מכריעה).
מצד שני, בהינתן M נבנה E כך ש- $L(E)$ הן כל המילים ש-M עוצרת ומקבלת עליהן. יהי $w_0, w_1, w_2, \ldots$ סידור של כל המילים ב-Σ^* (למשל הסדר הלקסיקוגרפי תמיד נוח). E תפעל כך: עבור $i=1,2,\ldots$ הרץ את M על $w_1, w_2, \ldots, w_i$ i צעדים. הדפס כל מילה ש-M מקבלת. אם M מקבלת את w_i אז w_i בסופו של דבר יודפס (ומותר שזה יודפס אינסוף פעמים), שהרי אם M מקבלת את w_i ב-j צעדים אז באיטרציה $\max(i,j)$ תודפס w_i.
אם M לא מקבלת את w אז w לא תודפס ברשימה, כלומר אנחנו מדפיסים רק את מה ש-M מקבלת.

יום שני - 9-11 ..קודה שיעור השלמה

(27) 1|6|07
חישוביות

התיזה של Church -ו Turing

ב-8 באוגוסט 1900 היה קונגרס מתמטי חשוב והילברט עלה לבמה והציג רשימה מאוד מפורסמת של בעיות פתוחות במתמטיקה. מבין 23 הבעיות שהציג רובן כבר נפתרו אבל חלקן התגלו כ"מקולקלות".

הבעיה העשירית של הילברט היא: לתאר אלגוריתם שבהינתן פולינום במספר משתנים p, יכריע האם ל-p שורש שלם.

האלגוריתם המיידי שעולה בראש הוא לנסות את כל האפשרויות, אבל הבעיה היא שאם אין פתרון שלם אז הוא לא יעצור אף פעם.

הילברט לא השתמש אף במילה "אלגוריתם" אלא ב"תהליך שאותו ניתן להכריע אחרי מספר סופי של צעדים". היום ידוע שאין אלגוריתם שפותר את הבעיה הזאת.

ב 1936 טיורינג הראה שיש אלגוריתם אמ"מ יש מ"ט שפותרת את הבעיה. צ'רץ' השתמש בכלי שנקרא λ-calculus והוא הראה בעצם את אותו הדבר.

למשל, במקרה של הפולינום הבעיה היא בעצם להכריע את השפה

$$L = \{ \langle p \rangle : \text{פולינום עם שורש שלם } p \}$$

ברור ש- $L \in RE$. נניח ש-p ל-m משתנים. פשוט נעבור על כל הצירופים האפשריים ל-m המשתנים וזה לא בעיה כי Z^m בת-מניה.

אבל, עבור $m=1$ הבעיה היא כריעה. יש איזה משפט מתמטי שתוחם את השורשים (אם יש כאלה) בתחום סופי. אז אם מדובר בשורשים שלמים (שזה לא משהו מסובך) פשוט עוברים על כל השלמים בתחום הזה, ואם אף אחד מהם לא היה שורש אז יודעים שגם לא נמצא שורש במקום אחר. ואם אחד מהם היה שורש אז סיימנו.

תאור אלגוריתם ע"י מכונת טיורינג

1) תאור פורמלי - $M=\langle Q,\Sigma,\Gamma,\delta,q_0,q_{rej},q_{acc}\rangle$
2) רמת המימוש - תאור בשפה עלית של פעולת מכונת הטיורינג
3) רמה אבסטרקטית - תאור האלגוריתם בשפה טבעית

אלגוריתמים הם משהו גרפים, מטריצות וכו'. אבל מה הוא משהו א"ב כלשהו Σ. זאת הבעיה כי ממש לא תמיד קידוד עדינים של אלגוריתמים בשפה של מכונות טיורינג, כי עוף מתאימים את זה מראש.
למשל אם יש גרף $G=(V,E)$ אפשר לקודד אותו ע"י

$$\langle G\rangle = V_1\#V_2\#\dots\#V_n\$(V_1,V_2)(V_1,V_n)\dots$$

אפשר אפילו לקודד מכונת טיורינג אחרת:

$$\langle M\rangle = \sigma_1\#\sigma_2\#\dots\#\sigma_n\$\gamma_1\#\gamma_2\#\dots\#\gamma_m\$\dots\$q_0\$q_{acc}\$q_{rej}$$

מכונת הטיורינג צריכה לדעת מה לא קידוד חוקי והדבר הראשון שהוא עושה תמיד הוא לסרוק את הקלט ולבדוק אם הוא חוקי בכלל.

מכונת טיורינג המכריעה האם גרף לא מכוון הוא קשיר

$A=\{\langle G\rangle :$ G גרף לא מכוון קשיר $\}$

ברמה האבסטרקטית היינו אומרים כזה דבר:
- בחר קודקוד כלשהו
- תעשה ממנו BFS
- תבדוק אם ניסינו את כולם.

אבל זה מאוד אבסטרקטי. איך נעשה זה מכונת טיורינג??

M תפעל כך: יש קלט $\langle G\rangle$ המקודד גרף
- אם מילת הקלט אינה מקודדת גרף נדחה אותה. זאת לא בעיה כי זה בסה"כ סינטקס. המכונה יודעת מה בדיוק צריך להגיע. למשל, אם פתאום יש 2 סולמיות ברצף היא תדחה, או שבצלעות יש קודקוד שלא הופיע בקודקודים היא גם תדחה.
- בחר קודקוד V וסמן אותו
- תחזור עד שלא מסומנים קודקודים חדשים:

לכל קודקוד בG סמן אותו אמ"מ יש קשת בינו ובין קודקוד מסומן

- עבור כל σ קידודי σ. אם כולם מאושרים, קבל. אחרת, דחה.

גם כאן יש פרטים שלא ר"כ ברור איך מממשים במכונת טיורינג.

- "סמן אותו" פירושו להחליף את האות σ הראשונה, בקידוד של v באות σ̇.

- לפי סדר, נמצא קידוד של סימון, נסמן אותו כקידוד ונבדק ע"י החלפה σ ב-σ̄.

שפות/בעיות כריעות

בעיות מתורת האוטומטים

① בעיית השייכות: בהינתן DFA A ומילה w האם $w \in L(A)$?

② בעיית הריקנות: בהינתן A האם $L(A) \neq \emptyset$?

סימון: $L_{DFA} = \{\langle A\rangle, w : w \in L(A) \text{ ו- } A \text{ קידוד של DFA}\}$

<u>טענה:</u> $L_{DFA} \in R$

<u>הוכחה:</u> נבנה מ"ט M שעל קלט $\langle A\rangle w$ תסמלץ ריצה של A על w. לא בעיה. יהיו לנו 3 סרטים. בראשון הקלט, בשני המצב שבו אנחנו נמצאים ובשלישי מיקום הראש בקלט.

אם הסימולציה הסתיימה במצב מקבל, מקבלים ואם היא הסתיימה במצב דוחה, דוחים.

סימון: $L_{NFA} = \{\langle A\rangle, w : w \in L(A) \text{ ו- } A \text{ קידוד של NFA}\}$

<u>טענה:</u> $L_{NFA} \in R$

<u>הוכחה:</u> אפשרות אחת היא קלאסית: ליצור DFA שקול ל-A ואז להמשיך כמו בהוכחה הקודמת.

אפשרות אחרת היא לבנות אוטומט on-the-fly. כלומר, לא לחשב מצב אחד בסרט המצבים, אלא לתחזק סט של המצבים שבהם A עשוי להיות בריצה כשהוא עומד במקום הראש הנוכחי.

אי כריעות

משפט: יש שפה $L\notin R$

הוכחה: משיקולי מניה. קבוצת השפות מעל $\{0,1\}$ הנה לא בת-מניה, שהרי מספר כל תתי הקבוצות של וקטורים מעל $\{0,1\}$ וזה $\aleph = 2^{\aleph_0}$ אלף שני, ואילו תיאור של מ"ט הוא סופי, לכן אוסף כל מכונות הטיורינג הוא בן מניה. $\Leftarrow$ חייבת להיות שפה שאין עבורה מ"ט. ☺

דוגמה: $A_{TM} = \{\langle M\rangle, w : w\in L(M)$ -ו מ"ט של קידוד $M\}$ אינה ב-R.

נשים ♡ ש-$A_{TM}\in RE$ שהרי מ"ט שמכריעה את A_{TM} פשוט מסמלצת ריצה של M על w ומוצאת תשובה עבור $w\in L(M)$.

אבל נוכל ש- $\overline{A_{TM}} = \{\langle M\rangle, w : M(w)\neq accept$ -ו מ"ט של קידוד $M\}$ אינה ב-RE שהרי אם ניתוח ב-RE היינו מקבלים ש-$A_{TM}\in R$ (כי $RE \cap co\text{-}RE = R$). אבל, מה שרשמנו כאן הוא לא מדויק $\overline{A_{TM}}$ משום שפורמלית היו צריכים שיהיו בה גם מילים שלא עונות $\langle M\rangle$ כקידוד של מ"ט, אבל מאחר שבדיקת סינטקס היא לא מה שמעניין בבעיה אנחנו מתעלמים מזה. לא קשה לראות שגם אם עדיין היינו מתחשבים במילים הללו, כל מה שאמרנו הוא נכון.

אנו ננסה כעת להראות ש- $A_{TM}\notin R$. נניח בשלילה ש- $A_{TM}\in R$ $\Leftarrow$
יש מ"ט H שמקבלת כקלט $\langle M\rangle, w$ -ו:

$$H(\langle M\rangle, w) = \begin{cases} accept & M(w) = accept \\ reject & \underbrace{M(w) \neq accept}_{\text{כלומר } M \text{ עוצרת של } W \text{ ויוצאת בדחייה או שהיא נכנסת ללולאה אינסופית..}} \end{cases}$$

א-H נבנה מ"ט D, שמקבלת כקלט תאור $\langle M\rangle$ של מ"ט ופועלת רק:

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

D פשוט תפעל רק: מקבלת כקלט $\langle M\rangle$ והיא תריץ את $H(\langle M\rangle, \langle M\rangle)$

(29)

מ-D נבנה 'D שמקבלת כקלט תאור $\langle M \rangle$ של מ"ט

$$D'(\langle M \rangle) = \begin{cases} \text{reject} & M(\langle M \rangle) = \text{accept} \\ \text{accept} & M(\langle M \rangle) \neq \text{accept}. \end{cases}$$

'D מתקבלת מ-D ע"י החלפת q_{acc} ו- q_{rej}!

עכשיו נתבונן מה קורה כש 'D על $\langle D' \rangle$. נקבל ש- ,

$$D'(\langle D' \rangle) = \begin{cases} \text{reject} & D'(\langle D' \rangle) = \text{accept} \\ \text{accept} & D'(\langle D' \rangle) \neq \text{accept} \end{cases}$$ (לא נקרא לסיום)

וזאת סתירה!!!

מעכשיו יהיה לנו מאוד קל להראות ששפות הן לא ב- R או לא ב-RE

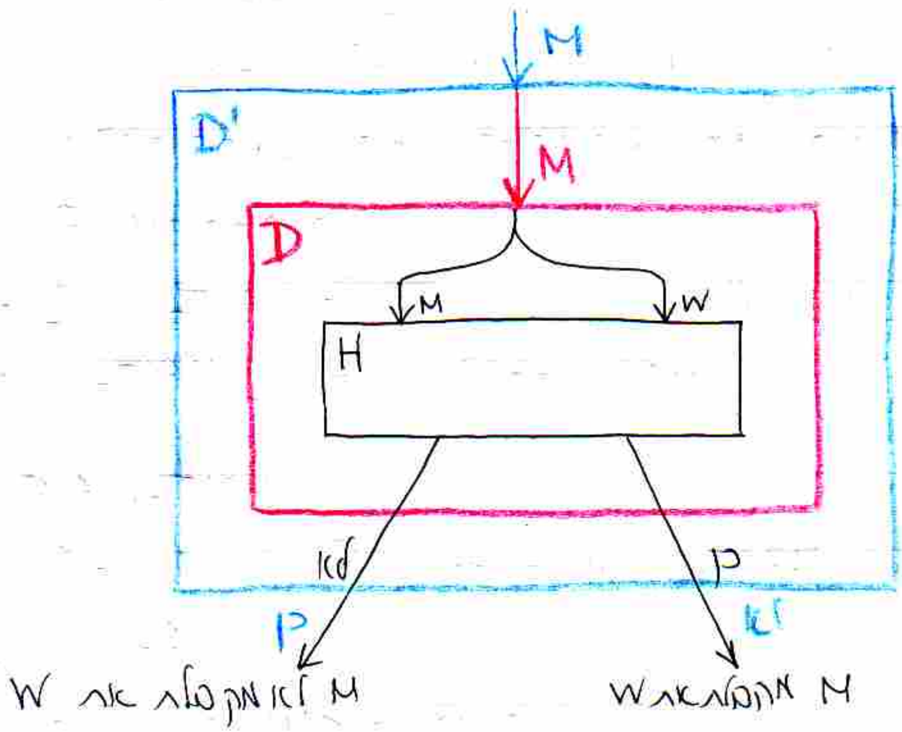
וזאת ע"י רדוקציה ובנית שרשור המכונות עליהן את הדברים האלה.

(30) 4.6.07
חישוביות

ביום שישי ראינו שיש שפות שאינן כריעות:

$A_{TM} = \{\langle M\rangle, w :$ M קידוד של מ"ט ו- $w \in L(M)\}$

הנחנו בשלילה ש-A_{TM} כריעה והגענו לסתירה בתהליך הבא:



אם H מכריעה את A_{TM} הצלחנו להגדיר D' רק ש-

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

וזאת סתירה!

<u>מושג</u>: <u>מ"ט אוניברסלית</u> U היא מ"ט המקבלת כקלט מ"ט M ומילה w והיא מסמלצת ריצה של M על w.

ל-U יש שני סרטים: 1) הקלט $\langle M\rangle, w$

2) הקונפיגורציה הנוכחית של M על w (סרט הסימולציה)

בהינתן $M, \langle w\rangle$ המכונה U כותבת q_0w על סרט הסימולציה.

בכל שלב ש-U קוראת את המצב הנוכחי q והאות a שהראש הקורא מצביע עליה, מחפשת את $\delta(q,a)$ בסרט הקלט ומעדכנת את הקונפיגורציה בהתאם.

- אם M מגיעה ל- q_{acc} , U מקבלת את $\langle M\rangle, w$
q_{rej} , U דוחה את $\langle M\rangle, w$
- אם M לא עוצרת אז U לא עוצרת.

<u>דוגמה:</u> $HALT_{TM} = \{\langle M\rangle, w :$ M עוצרת על $w\}$

קל לראות ש- $HALT_{TM}$ היא ניתנת למניה רקורסיבית. פשוט נריץ את M על w ואם נעצר יופי ואם לא אז לא משנה.
אולם אם השפה הנ"ל הייתה רקורסיבית אז היינו מקבלים סתירה לכך ש-A_{TM} אינה רקורסיבית.

<u>הרעיון:</u> נראה ש- $HALT_{TM}$ קשה לפחות כמו A_{TM}. לו היינו יודעים להכריע את $HALT_{TM}$ אז היינו מצליחים להכריע גם את A_{TM}.

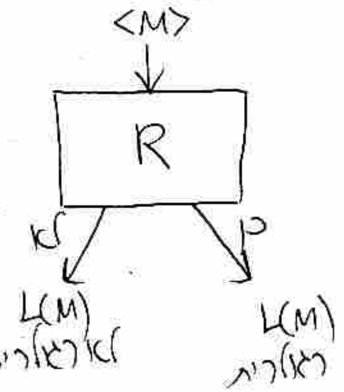
<u>משפט:</u> $HALT_{TM} \notin R$

<u>הוכחה:</u> בשלילה יש מ"ט R המכריעה את $HALT_{TM}$. נייצר מ"ט S המכריעה את A_{TM}.
S פועלת על קלט $\langle M\rangle, w$ באופן הבא:
מריצה את R על $\langle M\rangle, w$. אם R דוחה יודעים ש- M לא עוצרת על w $\Leftarrow$ M לא מקבלת את w $\Leftarrow$ S דוחה.
אם R מקבלת אז M עוצרת על w. אז S מריצה את M על w ומובטח לנו שהיא תעצור. אז S תדחה/תקבל בהתאם לריצה של M על w.

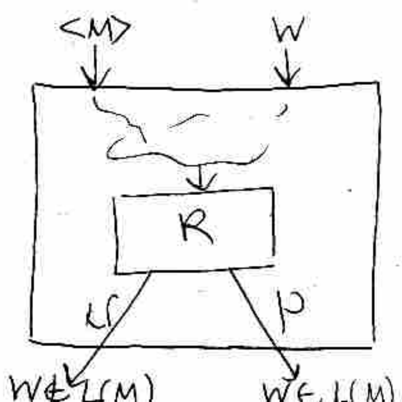
<u>משפט:</u> $Regular = \{\langle M\rangle :$ רגולרית $L(M)\} \notin R$

<u>הוכחה:</u> ע"י רדוקציה ל- A_{TM}. נראה ש-

אם היה



אז היה



<u>הרעיון:</u> נבנה מכונה M' ונראה את R כך שבאמת $L(M')$
רגולרית אמ"מ $(\langle M\rangle,w)\in A_{TM}$?

M' תפעל על קלט x כך:

1) אם x מהצורה 0^n1^n M' מקבלת את x (לא תלוי ב- $\langle M\rangle,w$)

2) אחרת, M' מריצה את M על w ומקבלת את x אמ"מ M מקבלת את w.
(לא תלוי ב-x. תלוי ב- $\langle M\rangle,w$ ויתכן ש-M' לא תעצור).

נשים ♥:

- אם M לא מקבלת את w אז
$$L(M')=\{0^n1^n : n\geq 0\}\notin REG$$
- אם M כן מקבלת את w אז
$$L(M')=(0+1)^*\in REG$$

וקיבלנו את מה שצריך! רק נבנה את המכונה S שמכריעה את A_{TM}:

בהינתן $\langle M\rangle,w$:

1) מייצרת את $M'(M,w)$

2) מריצה את R על M'.

אם R מקבלת אז S מקבלת

אם R דוחה אז S דוחה

S תמיד עוצרת כי R תמיד עוצרת. זה בסתירה לכך ש- $A_{TM}\notin R$.

☺

פונקציות ניתנות לחישוב

<u>הגדרה:</u> $f:\Sigma^*\to\Sigma^*$ היא פונקציה ניתנת לחישוב (computable function) אם קיימת מ"ט M_f כך שעל קלט $w\in\Sigma^*$, M_f עוצרת עם $f(w)$ על הסרט.

למשל $f: x\#y \to x+y$ כאשר x ו- y נתונים ע"י יצוג אונרי.

f: מ"ט $\to$ מ"ט כך ש $\langle M\rangle$ עוברת ל-$\langle M'\rangle$ כך ש-M'
לעולם לא מנסה ליפול שמאלה מהסרט.

הגדרה: שפה $A \subseteq \Sigma^*$ ניתנת לרדוקציה מיפוי לשפה $B \subseteq \Sigma^*$ $(A \le_m B)$ אם יש פונקציה ניתנת לחישוב $f: \Sigma^* \to \Sigma^*$ כך שלכל $w \in \Sigma^*$

$$w \in A \leftrightarrow f(w) \in B$$

במקרה זה f נקראת רדוקציה מ-A ל-B.

רדוקציה היא בעצם המרה של שאלות על חברות ב-A לשאלות על חברות ב-B.

משפט: אם $A \le_m B$ ו-$B \in R$ אז $A \in R$. (חסם עליון)

הוכחה: בהינתן M_B שמכריעה את B ופונקציה ניתנת לחישוב $f: \Sigma^* \to \Sigma^*$ כך שלכל $w \in \Sigma^*$ $w \in A$ אמ"מ $f(w) \in B$

נבנה M_A שמכריעה את A.

על קלט w, M_A מחשבת את $f(w)$ ומריצה את M_B על $f(w)$.

מסקנה: אם $A \le_m B$ ו-$A \notin R$ אז $B \notin R$. (חסם תחתון)

כלי מרכזי להוכחת אי כריעות

נחזור לדוגמה שראינו לאי כריעות: $HALT_{TM}$: נראה ש- $A_{TM} \le_m HALT_{TM}$ ולפי המסקנה והעובדה ש- $A_{TM} \notin R$ ינבע ש- $HALT_{TM} \notin R$:

אז נבנה פונקציה חשיבה f כך ש-

$$f(\langle M \rangle, w) \in HALT_{TM} \leftrightarrow \langle M \rangle, w \in A_{TM}$$

בהינתן $\langle M \rangle, w$ M_f תחזיר M' ו-w' כך: עבור קלט x

1) M' מריצה את M על x

2) אם M מקבלת את x גם M' מקבלת

3) אם M דוחה את x M' נכנסת ל-loop

4) $w' = w$

32

מייצרים M' כך ש- $M',w \in HALT_{TM}$ אמ"מ $M,w \in A_{TM}$:

א) אם $\langle M\rangle,w \in A_{TM}$ אז $\langle M'\rangle,w \in HALT_{TM}$:

כי אז M' תריץ את M על w ואם M מקבלת אז M' מקבלת.

2) אם $\langle M\rangle,w \notin A_{TM}$ אז $\langle M'\rangle,w \notin HALT_{TM}$:

אם זה בגלל ש- M לא עוצרת על w אז זה "יתקע" בשלב 1.

אם היא הגיעה לבור דוחה אז נתקע בשלב 3.

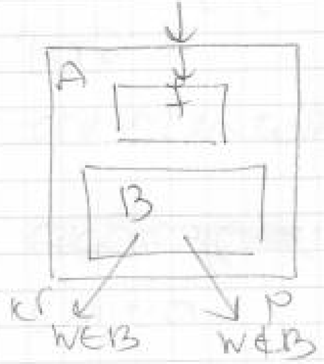
$\Rightarrow$ M' לא עוצרת בכל אופן. ∇

(33) 6/6/07
חישוביות

הגדרנו את הרדוקציה: $A \le_m B$ אם יש פונקציה ניתנת לחישוב $f:\Sigma^* \to \Sigma^*$
כך שלכל $w \in \Sigma^*$ $w \in A \Leftrightarrow f(w) \in B$

אומרים ש-A ניתנת לרדוקציית מיפוי ל-B. משפט: כי אם $A \le_m B$
ו-B הכריעה אז A הכריעה. ואם A לא הכריעה
אז B לא הכריעה.

בצורה דומה אם B ני"ר אז גם A ני"ר (אותה הוכחה בדיוק).
ואז אם A לא ני"ר אז B לא ני"ר.



שאלה: $A_{TM} \le_m HALT_{TM}$

למה? כי אנחנו יודעים ש A_{TM} אינה הכריעה
וזה יאפשר לנו להסיק ש- $HALT_{TM}$ אינה הכריעה.

איך? אנחנו צריכים למצוא {מ"ט ומילה} ← {מ"ט ומילה} f
כך שמתקיים

$$\langle M\rangle, w \in A_{TM} \leftrightarrow f(\langle M\rangle, w) \in HALT_{TM}$$

נתאר את M_f (המכונה שמחשבת את $f(\langle M\rangle, w)$) היא מקבלת
קלט $\langle M\rangle, w$ ומייצרת פלט $\langle M'\rangle, w'$.

M' תפעל באופן הבא: על קלט x:
1) מריצה את M על x
2) אם M מקבלת את x גם M' מקבלת את x
3) אם M דוחה את x אז M' נכנסת ללולאה אינסופית.

M_f מחזירה את $w' = w$.

אז M_f מ"ט תקינה ומתכנסת: לכן f פונקציה חשיבה.
אז נשאר לנו להראות שאם M מקבלת את w אמ"מ
M' עוצרת על w'.

אבל זה ברור. אם M מקבלת את w אז נראה איך M' פועלת על
w'=w קלט היא מריצה את M על w ועוצרת, לכן מקבלת,
בפרט עוצרת. ואם M לא מקבלת את w אז M' תריץ את
M על w ומכיוון אם M דוחה או M לא עוצרת M' לא תעצור!

רדוקציה: $HALT^{\varepsilon}_{TM} = \{ \langle M \rangle : M \text{ עוצרת על הקלט } \varepsilon \}$

רוצים להסיק ש- $HALT^{\varepsilon}_{TM}$ אינה כריעה. נראה את הרדוקציה

$$HALT_{TM} \le_m HALT^{\varepsilon}_{TM}$$

בהינתן קלט $\langle M \rangle, w$ ל- $HALT_{TM}$ נראה איך לייצר $f(\langle M \rangle, w) = M'_{(w,\langle M \rangle)}$

כך ש- M' עוצרת על ε $\Longleftrightarrow$ M עוצרת על w

M' תפעל באופן הבא: כותבת את w על הסרט ואז מריצה את M.

ברור שכאשר מריצה אותה על w הריצה של M' היא סופית.

f - ל נתונים $\langle M \rangle$ ו- w אז אין בעיה להעתיק את w לסרט.

ברור שזאת באמת רדוקציה.

עד עכשיו ~~דיברנו~~ הרבה על אוטומטים, אבל אנחנו רוצים דבר של החיים האמיתיים.

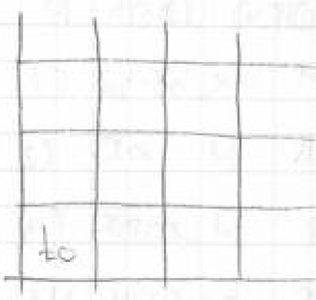
אנחנו נראה שבעיית הריצוף אינה ב- RE.

<u>הקלט</u>: $T = \{t_0, \dots, t_k\}$ קבוצה סופית של אריחים.

$H, V \subseteq T \times T$ תנאים אופקיים ומאונכים.

<u>ריצוף $n\times n$ חוקי</u>: זו פונקציה $f : \{1,\dots,n\}^2 \to T$

כך ש- $f(1,1) = t_0$



ולכל i,j $\quad H(f(i,j), f(i+1,j))$ $\quad 1 \le i \le n-1,\ 1 \le j \le n$

$V(f(i,j), f(i,j+1))$ $\quad 1 \le i \le n,\ 1 \le j \le n-1$

<u>הבעיה</u>: האם השפה הבאה היא כריעה / נ"ר ?

$Tile = \{ \langle T, H, V, t_{init} \rangle :$ קיים ריצוף $n\times n$ חוקי לכל $n \ge 1 \}$

קל לראות ש- $Tile \in co\text{-}RE$ אפשר לכל n הבדיקה היא סופית. אז נבדוק את כל ה- n-ים ואם באיזה שלב נראה שאין ריצוף אז נעצור.

ביום שני נראה רדוקציה $\overline{HALT^{\varepsilon}_{TM}} \le_m Tile$. ראינו ש- $HALT^{\varepsilon}_{TM} \notin R$. קל

לראות ש- $HALT^{\varepsilon}_{TM} \in RE$ $\Longleftarrow$ $\overline{HALT^{\varepsilon}_{TM}} \notin RE$

(34) 11/6/07
חישוביות

בעיית הריצוף:

$T=\{t_1,\dots,t_k\}$ קבוצה של אריחים

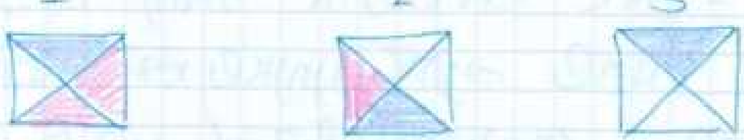
$H,V\subseteq T\times T$ חוקים לריצוף אופקי ואנכי

t_0 נמצא בתא $(1,1)$ ושואלים אם קיים ריצוף חוקי לכל $n\in\mathbb{N}$.

למשל,
$T=\{1,2,3\}$
$t_0=3$
$H=\{(3,1),(1,2),(2,3),(2,1)\}$
$V=\{(1,2),(3,2),(2,3)\}$

את המקרה הזה אפשר לתאר בעזרת אריחים עם צבעים

1 2 3



כשמניחים שני אריחים אחד ליד השני צריך שהצבעים של הצלעות יסכימו.
ספציפית למקרה הזה קיים ריצוף חוקי לכל n.

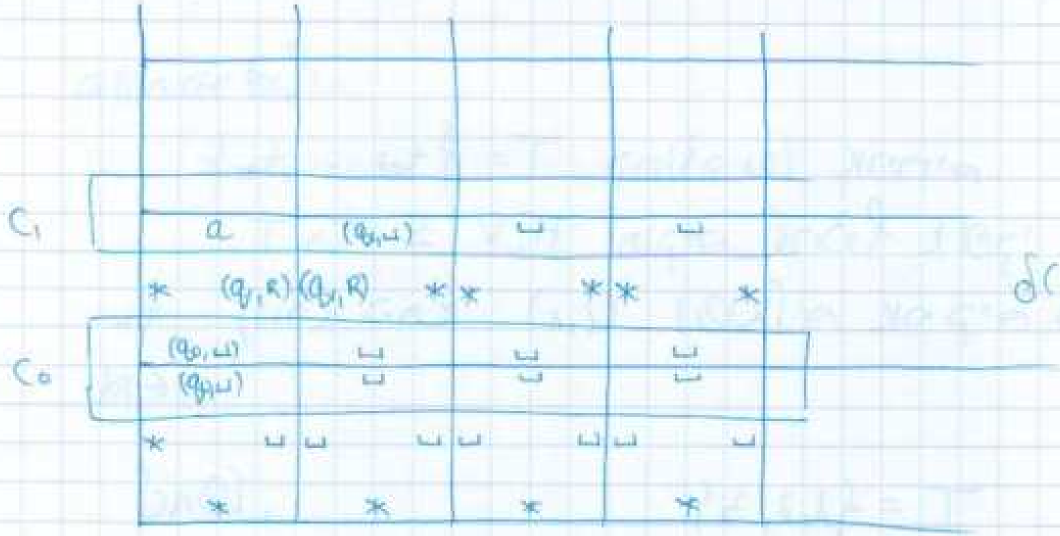
נראה שזאת בעיה קשה ע"י שנראה את הרדוקציה
$\overline{HALT_{TM}^{\varepsilon}}\le_m TILE$

כאשר:

$TILE=\{\langle T,H,V,t_0\rangle :$ לכל n קיים ריצוף חוקי$\}$
$\overline{HALT_{TM}^{\varepsilon}}=\{\langle M\rangle :$ M לא עוצרת על $\varepsilon\}\notin RE$

בהינתן M נרצה לייצר T,H,V,t_0 כך ש-M לא עוצרת על ε אמ"מ $\langle T,H,V,t_0\rangle\in TILE$.

הרעיון הוא שהאריחים איכשהו יקודדו את M וריצוף לא חסום יתאים לריצה אינסופית.



$\delta(q_0,\sqcup)=(q_1,a,R)$

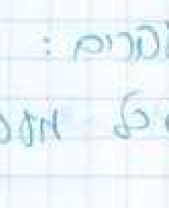
ראשית: הבעיה TILE שקולה לכך שיש ריצוף חוקי לכל רבע המישור.

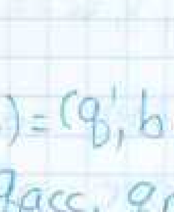
ברור שאם יש ריצוף חוקי לכל רבע המישור אז יש ריצוף חוקי לכל n. הכיוון ההפוך נובע מהלמה של קניג (מי שמכיר את הלמה - זה תרגיל נחמד. מי שלא מכיר אותה - לא משנה).

בכל אופן הרעיון הוא שאנחנו טוענים שתמיד אפשר להאריך ריצוף n×n כלשהו לריצוף $(n+1)\times(n+1)$. שהרי אחרת בשלב כלשהו היינו נתקעים, כלומר את n×n אי אפשר להאריך, ונסתכל על ריצוף $(n+1)\times(n+1)$ (הוא קיים לפי ההנחה). אבל אז אם נסתכל על n×n ממנו נקבל ריצוף n×n שאותו אפשר להאריך, בסתירה להנחה.

בכל אופן יש מלאכה.

(1) מרצפות השורה הראשונה:



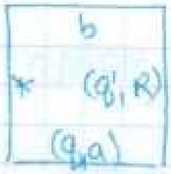


- מרצפת התחלתית.

(2) מרצפות המעברים:

2R: לכל מעבר $\delta(q,a)=(q',b,R)$ כאשר $q\notin\{q_{acc},q_{rej}\}$

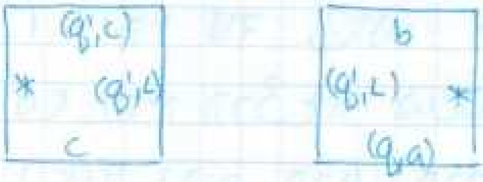
נוסיף מרצפות



ולכל $c \in \Gamma$ נוסיף מרצפה

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

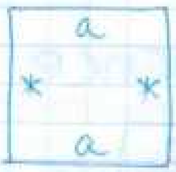
נוסיף מרצפות



סה"כ לכל מעבר מוסיף $1+|\Gamma|$ מרצפות חדשות.

③ מרצפות ריפוד:

לכל a מרצפה



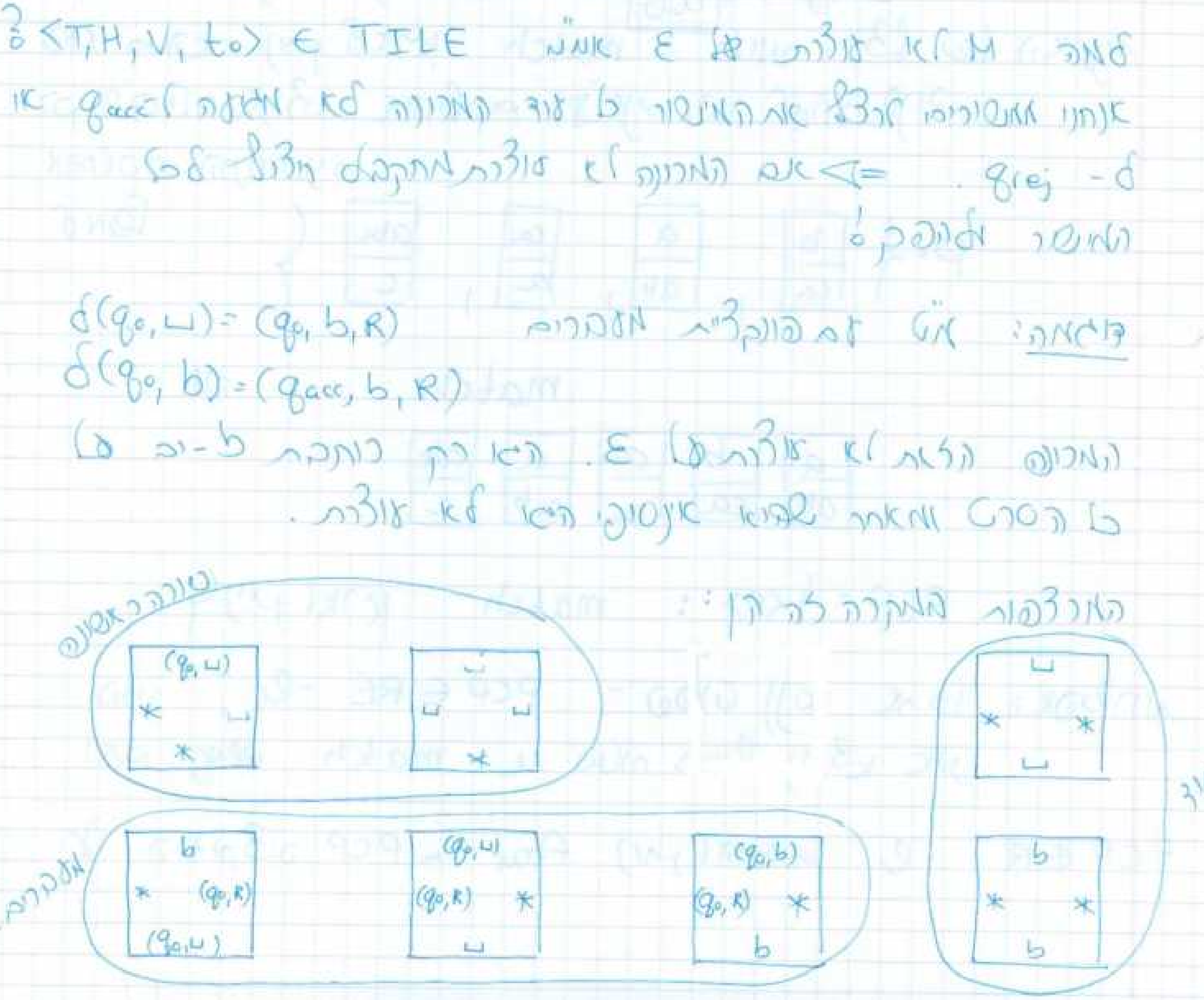
למה M לא עוצרת על ε אמ"מ $\langle T,H,V,t_0\rangle \in TILE$?

אנחנו מאשרים לרצף את המישור כל עוד המכונה לא מגיעה ל-q_{acc} או ל-q_{rej}. $\Leftarrow$ אם המכונה לא עוצרת מתקבל ריצוף של כל המישור ולהפך!

דוגמה: מ"ט עם פונקציית מעברים
$\delta(q_0,\sqcup)=(q_0,b,R)$
$\delta(q_0,b)=(q_{acc},b,R)$

המכונה הזאת לא עוצרת על ε. היא רק כותבת b-ים על כל הסרט ומאחר שהוא אינסופי היא לא עוצרת.

המרצפות המתאימות לה הן:



קל לראות שקיים רידוקציה חוקית לבעיית המישור בעזרת האריחים הנ"ל.

הערה: אין את הבעיה הזאת ב- sipser אבל יש באינטרנט.

הודעות: - ביום שישי - תרגול חזרה
- לא יהיו עוד שיעורי השלמה
- לא תהיה הרצאה בשבוע לפני המבחן
(אולי יהיה תרגול חזרה)

Post Correspondence Problem (PCP)

נתון: אוסף אבני דומינו שכתובים עליהן רצפי אותיות
שואלים האם ניתן לארגן match - כלומר רצף של האבנים
כך שהמילה שנוצרת מהחלקים העליונים שווה למילה שנוצרת
מהחלקים התחתונים.

למשל

$$P = \left\{ \left[\frac{b}{ca}\right], \left[\frac{a}{ab}\right], \left[\frac{ca}{a}\right], \left[\frac{abc}{c}\right] \right\}$$

הנה match:

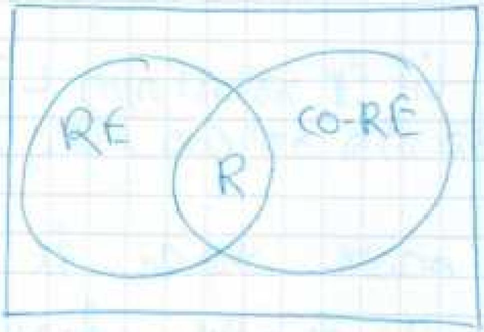
$$\left[\frac{a}{ab}\right] \left[\frac{b}{ca}\right] \left[\frac{ca}{a}\right] \left[\frac{a}{ab}\right] \left[\frac{abc}{c}\right]$$

$PCP = \{\langle P \rangle :$ ניתן לארגן match $\}$

ברור ש- $PCP \in RE$ - פשוט ננסה את כל האפשרויות
ואם קיים match אז בסוף נמצא אותו!

ע"י הרדוקציה $A_{TM} \le_m PCP$ (נראה בהמשך) ש- $PCP \notin R$.

תורת הסיבוכיות



מה קורה בתוך R?

- איפיון 1: REG, CFL
- איפיון 2: כמה משאבים צריך כדי להכריע את השפה?

דוגמה: אלגוריתם להכרעת השפה $L=\{0^n1^n : n\geq 0\}$

בהינתן $w\in(0+1)^*$ מ"ט שמכריעה השפה תפעל כך:

1) נוודא ש- $w\in 0^*1^*$
2) כל זוג אפשר, נמחק 0 ראשון ו-1 ראשון
3) נקבל אם המחיקות של ה-0-ים וה-1-ים הסתיימו בו זמנית.

מה הסיבוכיות של האלגוריתם?

שלב 1: $O(n)$

שלב 2: $\left.\begin{array}{l}\text{כל מחיקה } O(n) \\ \frac{n}{2} \text{ מחיקות}\end{array}\right\} \Rightarrow O(n^2)$

שלב 3: $O(1)$.

האם אפשר בפחות מ- $O(n^2)$?

1) עם שני סרטים אפשר ב- $O(n)$ (אבל המעבר משני סרטים לסרט יחיד מעלה את הזמן בריבוע)
2) עם סרט אחד אפשר ב- $O(n\log n)$ ע"י בדיקת התאמה של זוגיות, חלוקה ב-2 וחוזר חלילה.

לא ניתן להכריע בפחות מ- $O(n\log n)$. יש משפט שאומר שאם L ניתנת להכרעה ב- $o(n\log n)$ אז L רגולרית. מאחר ש $\{0^n1^n\}$ אינה רגולרית, היא אינה ניתנת להכרעה ב- $o(n\log n)$.

$$TIME(t(n)) = \{L : \text{ ניתנת להכרעה ע"י מ"ט דטרמיניסטית } \text{ בעלת סרט יחיד הרצה בזמן } O(t(n))\}$$

למשל, $\{0^n1^n\} \in TIME(n\log n)$

$$NTIME(t(n)) = \{L \text{ ניתנת הכרעה ע"י מ"ט אי-דטרמיניסטית בעלת סרט יחיד הרצה בזמן } O(t(n))\}$$

למ"ט א"ד יש פונק' מעברים $\delta: Q\times\Gamma \to 2^{Q\times\Gamma\times\Delta}$

היא מקבלת מילה אם יש מסלול מקבל עבורה.

מ"ט א"ד מכריעה שפה אם כל החישובים של מילים ב-Σ^* עוצרים.

זמן הריצה של המכונה על מילה הוא מספר הצעדים המקסימלי מחישוב כלשהו.

לכל מ"ט א"ד שרצה בזמן $t(n)$ קיימת מ"ט דטרמיניסטית שקולה שרצה בזמן $2^{t(n)}$.

המחלקות P ו-NP

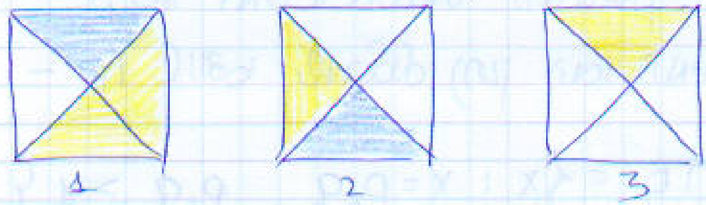
$P = \bigcup_k TIME(n^k)$ = קבוצת השפות הכריעות בזמן פולינומיאלי ע"י מ"ט דטרמיניסטית.

$NP = \bigcup_k NTIME(n^k)$ = קבוצת השפות הכריעות בזמן פולינומיאלי ע"י מ"ט א"ד.

13.6.07

חישוביות

תיקון הדוגמה מהשיעור הקודם:



$H = \{(2,1), (1,2), (3,1), (2,3)\}$

$V = \{(1,2), (2,3), (3,1)\}$

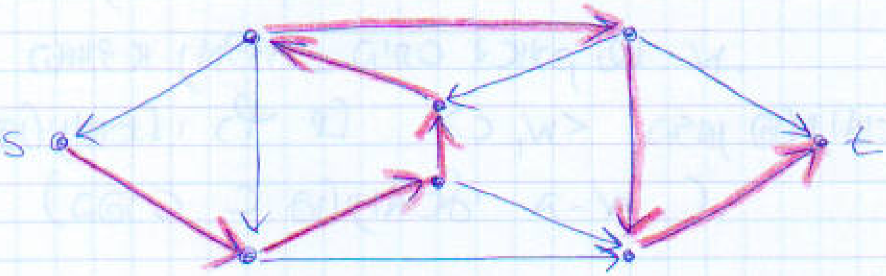
* * * * *

המחלקה NP

$P = \bigcup_k PTIME(n^k)$ (מיון, קשירות בגרף, ...)

$NP = \bigcup_k NPTIME(n^k)$

מסלול המילטוני בגרף מכוון -
מסלול העובר דרך כל קודקודי הגרף, כל קודקוד פעם אחת.
למשל,



HAMPATH = {<G, s, t>: G גרף מכוון ויש מסלול המילטוני מ-s ל-t}

- אלגוריתם מעריכי להכרעת HAMPATH:
נסרוק את כל המסלולים באורך n מ-s, ונבדוק את המילטוניותם של אלה שמגיעים ל-t.

האם קיים אלגוריתם פולינומיאלי? זאת בעיה פתוחה...

נשים ❤ : - קשה למצוא מסלול המילטון (מעריכי)

- קל לוודא שמסלול נתון הוא המילטון (פולינומיאלי)

עוד דוגמה: COMPOSITE $= \{x : x = p\cdot q \ , \ p,q > 1\}$

מספרים פריקים

בדיקה ע"י ניסיון למצוא מחלק $2,3,\dots,\sqrt{x}$ היא מעריכי באורך הייצוג הבינארי של x

לעומת זאת, בהינתן p,q קל לוודא ש- $x=pq$.

לא כל הבעיות קלות לבדיקה (משלים הבעיה שאינן כריעות).

למשל, HAMPATH. אין עד קצר שעוזר להשתכנע מאנו שאין מסלול המילטון מ-s ל-t. לעומת זאת, כן יש עד קצר שממנו אפשר להשתכנע שיש מסלול המילטוני מ-s ל-t.

<u>מוודא</u> (verifier) עבור שפה L הוא אלגוריתם A רק ש-

$L = \{w :$ A מקבל $\langle w,c\rangle$ עבור מילה כלשהי $c\}$

c נקרא עד (certificate).

סיבוכיות המוודא נמדדת ביחס לאורך של w.

מוודא פולינומיאלי רץ על $\langle w,c\rangle$ בזמן פולינומיאלי ב-w לכל w (בפרט c פולינומיאלי ב-w).

מוודא עבור HAMPATH: אלגוריתם A שיקבל אליו הגו $(\langle G,s,t\rangle , c)$ ו- A מקבל את $(\langle G,s,t\rangle , c)$ אם c הוא תאור של מסלול המילטוני מ-s ל-t ב-G.

מוודא פולינומיאלי עבור COMPOSITE: יקבל את $(\langle x, \overbrace{p,q}^{c} \rangle)$ אם $x=pq$ ו- $p,q>1$

אורן: היי דינה!
אל תסתכלי עליי,
אני במצב דוחה...
HAHA

שתי הגדרות:

(1) המחלקה NP: $\bigcup_k NPTIME(n^k)$

(2) המחלקה NP: שפות עם מוודא פולינומיאלי.

$HAMPATH \in NP$ (1): נראה מ"ט א"ד שרצה בזמן פולינומיאלי המכריעה את HAMPATH

- בהינתן $\langle G,s,t \rangle$ יהי n מספר הקדקודים ב- G.

1) נחש סדרה $p_1, \dots, p_n$ של קדקודים.
2) אם יש קדקוד שמופיע פעמיים דחה
3) אם $p_1 \neq s$ או $p_n \neq t$ דחה
4) אם יש $1 \le i \le n-1$ כך שאין קשת מ- p_i ל- p_{i+1} דחה
5) קבל

משפט: NP לפי (1) = NP לפי (2)

הוכחה בתרגול בית.

הרעיון: (1 ← 2) בהינתן מוודא A, מכונת הטיורינג תנחש עד (פולינומיאלי) ותפעיל את A

(2 ← 1) העד יהיה הריצה המקבלת.

שלמות ב- NP

(Cook, Levin שנות ה-70)

יש שפות (בעיות) ב- NP שאם נמצא פתרון פולינומיאלי עבורן, ינבע ש- P=NP. אלה השפות השלמות ב- NP (NP Complete).

(1) כדי להוכיח ש- P=NP די למצוא שפה NP-C אחת L, ולהוכיח ש- $L \in P$.

(2) כדי להוכיח ש- $P \neq NP$ די למצוא שפה L אחת ב- NP ולהוכיח ש- $L \notin P$.

(3) מכיוון שמאמינים ש- $P \neq NP$ הטענה $L \in NP\text{-}C$, נקבל אינדיקציה לאי קיום אלגוריתם מהיר, לקירובים, להסתפקות: קושי (הנדסה) ...

(39) 18.6.07
חישוביות

ראינו: קל להכריע $P = \bigcup_k TIME(n^k)$

קל לאמת $NP = \bigcup_k NTIME(n^k)$

כמובן $P \subseteq NP$. לא ידוע אם $NP \subseteq P$.

<u>משפט</u>: אם $3SAT \in P$ אז $P=NP$. (cook, Levin)

<u>בעיית הספיקות</u>: נתונה נוסחה עם n משתנים ושואלים האם יש השמה שמספקת אותה.

למשל $\varphi \equiv (x \wedge \bar{y}) \vee (\bar{x} \wedge z)$ עבור φ לזאת יש השמה מספקת $x=1, y=0, z=0$. השמה לו היא עד קצר לספיקות.

נגדיר פורמלית את הבעיה.

- משתנה בוליאני: מקבל ערכים מ- $\{true, false\}$ או $\{0,1\}$.
- פעולות בוליאניות: $\wedge$ and, $\vee$ or, $\neg$ not
- נוסחה בוליאנית: מתקבלת ממשתנים בוליאניים ע"י הפעלת פעולות בוליאניות. למשל הדקדוק הבא יוצר את הנוסחאות:

$$B \to 0 \mid 1 \mid Var \mid (B \vee B) \mid (\neg B) \mid (B \wedge B)$$
$$Var \to x, y, z, \dots$$

- השמה אחת: $f : Var \to \{0,1\}$
- <u>בעיית הספיקות</u>: $SAT = \{\langle\varphi\rangle :$ φ נוסחה ספיקה $\}$

תשתית להוכחת משפט קוק-לוין:

<u>רדוקציה פולינומיאלית</u>

ראינו $A \le_m B \leftrightarrow$ קיימת פונקציה חשיבה $f:\Sigma^* \to \Sigma^*$ כך שלכל $w \in \Sigma^*$ $w \in A \leftrightarrow f(w) \in B$

כאן אין שום דרישה על הזמן שלוקח לחשב את הפונקציה. ← ברמנטש

$f:\Sigma^* \to \Sigma^*$ נתנת לחישוב <u>בזמן פולינומיאלי</u> אם יש מ"ט M העוצרת בזמן פולינומיאלי ומחשבת את f (כלומר יש $t: N \to N$! t - פולינום שלכל w, M עוצרת אחרי $t(|w|)$ צעדים).

הגדרה: רדוקציה פולינומיאלית: $A \le_p B$ אמ"מ קיימת פונקציה ניתנת לחישוב בזמן פולינומיאלי $f:\Sigma^* \to \Sigma^*$ כך שלכל $w \in \Sigma^*$

$$w \in A \leftrightarrow f(w) \in B$$

<u>משפט</u>: אם $A \le_p B$ ו- $B \in P$ אז $A \in P$.

<u>הוכחה</u>: הנה מ"ט הרצה בזמן פולינומיאלי ומכריעה את A: על קלט w מחשבת את $f(w)$ ומריצה את המכונה של B על $f(w)$. מאחר ש- f פולינומיאלית, מובטח ש- $|f(w)|$ פולינומיאלי ב- $|w|$. ואז המכונה של B רצה בזמן פולינומיאלי של $|f(w)|$. סה"כ כל העניין הוא פולינומיאלי. ☺

<u>הגדרה</u>: 3CNF = צורה נורמלית לנוסחאות בוליאניות (כמו שלמדנו במערכות ספרתיות maxterms). למשל,

$$(x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee x_4)$$

ליטרל = משתנה או שלילתו (x_i או $\bar{x}_i$)

פסוקית = איווי (OR) בין מספר ליטרלים

נוסחה ב- CNF (conjunctive normal form) = גימום (AND) של פסוקיות

נוסחה ב- 3CNF = גימום של פסוקיות עם 3 ליטרלים כל אחת

$3SAT = \{\langle\varphi\rangle :$ φ נוסחה ב- 3CNF ספיקה $\}$

ידוע ש- $2SAT \in P$.

<u>הגדרה</u>: שפה L היא NP-שלמה אם:

1. $L \in NP$ (חסם עליון)
2. L היא NP-קשה (חסם תחתון) (כלומר לכל בעיה $L' \in NP$ מתקיים $L' \le_p L$).

<u>משפט</u>: אם $L \in NP\text{-Complete}$ ו- $L \in P$ אז $P=NP$

<u>הוכחה</u>: תהי $L' \in NP$. אזי $L' \le_p L$ (כי L NP-קשה)

⇐ אם $L \in P$ אז $L' \in P$ ⇐ $NP \subseteq P$.

<u>בעיית הריצוף התחום</u>

קלט: קבוצה סופית T של אריחים

$H,V \subseteq T \times T$ תנאי שכנות במאונך ובמאוזן

t_{init} - האריח במקום $(1,1)$

t_{fin} - האריח במקום $(1,n)$

n = נתון באונרית

פלט: האם ניתן לרצף מאופן חוקי ריבוע $n \times n$ רק ש- במקום $(1,1)$ יש t_{init} ובמקום $(1,n)$ יש t_{fin}.

$BT=\{<T,V,H,t_{init},t_{fin},n>:$ יש ריצוף חוקי $n \times n\}$

כמובן $BT \in NP$. מ"ט לא-דטרמיניסטית פשוט תנחש ריצוף ותבדוק אם הוא מתאים. ומאוד חשוב ש-n נתון באונרית. כך מה שמאפשר לנו לעשות בדיקות על n^2 אריחים.

הסבר של העניין עם הבדיקה: נניח שיש מכונה שמקבלת n ומבצעת:

```
for i=1 to n
    print *
```

אם n נתון בבסיס 10 אז האורך של קלט הוא $\log_{10} n$.
כך אם אורך הקלט הוא k מבצעים 10^k פעולות - מעריכי!
באופן דומה, אם n נתון בבסיס 2 אז האורך הוא $\log_2 n$
ואז אם אורך הקלט הוא k מבצעים 2^k פעולות - מעריכי!
<u>אבל</u>, אם n נתון באונרית אז אורך הקלט ומספר הפעולות שמבצעים שווים - כלומר הזמן הוא לינארי (פולינומיאלי).

בתרגול ראינו ש- $BT \in NP\text{-hard}$:

בהינתן מ"ט א"ד M עם סיבוכיות זמן פולינומיאלית $p(n)$ ומילה $w \in \Sigma^*$, נייצר $\langle T, H, V, t_{init}, t_{fin} \rangle$ כך ש-

$$w \in L(M) \leftrightarrow \langle T, H, V, t_{init}, t_{fin}, p(n) \rangle \in BT$$

זוהי השפה הנוספת שראינו שהיא NP קשה.

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

משתני הנוסחה: לכל $i,j \in \{1,\dots,n\}$ ואריח $t \in T$ יהיה משתנה $X_{i,j,t}$. נגדיר f כך ש- $f(X_{i,j,t})=1$ אם בהשמה המושרית מ-f האריח t נמצא במקום/משבצת (i,j). מספר המשתנים הוא $|T| \cdot n^2$.

① לכל מקום מתאים לפחות אריח אחד

φ_{ij} $$\bigvee_{t \in T} X_{i,j,t}$$

② בכל מקום יש לכל היותר אחת אחד

φ'_{ij} $$\bigwedge_{t \in T} \left(X_{i,j,t} \rightarrow \bigwedge_{t' \neq t} \overline{X_{i,j,t'}} \right)$$

השמה מספקת משרה ריצוף לוח מהרה חוקי, אבל ריצוף.

④ $$\varphi_{border} \equiv \left(X_{n,1,t_{fin}} \wedge X_{1,1,t_{init}} \right)$$

③ מתקיימים תנאי שכנות מאוזנים

$\Theta_{i,j}$ $$\bigvee_{(t,t') \in H} \left(X_{i,j,t} \wedge X_{i+1,j,t'} \right)$$

⑤ מתקיימים תנאי שכנות מאונכים

$\Theta'_{i,j}$ $$\bigvee_{(t,t') \in V} \left(X_{i,j,t} \wedge X_{i,j+1,t'} \right)$$

נגדיר:

$$\varphi = \bigwedge_{i,j \in \{1,\dots,n\}} \left(\varphi_{i,j} \wedge \varphi'_{i,j} \right) \wedge \varphi_{border} \wedge \bigwedge_{\substack{1 \leq i < n \\ 1 \leq j \leq n}} \Theta_{ij} \wedge \bigwedge_{\substack{1 \leq i \leq n \\ 1 \leq j < n}} \Theta'_{ij}$$

<u>טענה:</u> φ ספיקה אמ"מ $\langle T, V, H, t_{init}, t_{fin}, n\rangle \in BT$

<u>הוכחה:</u>

($\Leftarrow$) תהי f השמה מספקת. אתנאים א' ו-ב' f משרה ריצוף. אתנאים ג'-ה' הריצוף תקין.

($\Rightarrow$) יהי g ריצוף $(g: \{1,\dots,n\}^2 \to T)$. ההשמה $f: Var \to \{0,1\}$ המושרית ע"י g מקיימת את φ.

☺

<u>מסקנה:</u> $SAT \in NP\text{-}Hard$

לא ניכנס לזה, אבל ניתן לתרגם את φ בזמן פולינומיאלי ל-3CNF ולהראות כך ש-3SAT היא NP-קשה.

<u>דוגמה:</u> ריצוף של BT עם $n=2$

$\langle \{a,b\},\ H=\{(a,b),(a,b)\},\ V=\{(a,b),(b,a)\},\ 1,\ 2,\ 11 \rangle$

b	a
a	b

המשתנים הם: $x_{1,1,a}, x_{1,1,b}, x_{1,2,a}, x_{1,2,b}, x_{2,1,a}, x_{2,1,b}, x_{2,2,a}, x_{2,2,b}$

<u>משפט:</u> אם $L \in NP\text{-}hard$ ו- $L \le_p L'$ אז $L' \in NP\text{-}hard$

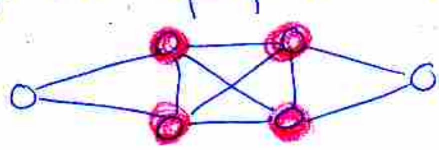
<u>הוכחה:</u> צריך להראות שלכל $L'' \in NP$ מתקיים $L'' \le_p L'$. ידוע ש- $L'' \le_p L$. ידוע גם $L \le_p L'$. אבל רדוקציה פולינומיאלית היא טרנזיטיבית $\Leftarrow$ $L'' \le_p L'$.

☺

עוד בעיה NP-Complete:

$CLIQUE = \{\langle G, k\rangle :$ G גרף לא מכוון המכיל קליקה בגודל k $\}$

קליקה היא $C \subseteq V$ קבוצה של קודקודים כך שיש קשת בין כל שני קודקודים בקבוצה. k-קליקה היא קליקה בגודל k.



① $CLIQUE \in NP$: המוודא הא"ז תנתח k-קליקה ותבדוק אותה.

② $CLIQUE \in NP\text{-hard}$. נראה ש- $3SAT \leq_p CLIQUE$

בהינתן נוסחה φ ב-3CNF נמצא $\langle G, k\rangle$

כך ש-φ ספיקה אמ"מ יש ב-G k-קליקה.

<u>הערה</u>: במקרה כזה לא חשוב מאיזה "גודל" נתון k, שהרי אם k גדול מאוד הגרף ברור שהתשובה היא לא.

הנוסחה φ שראינו בהרצאה מ-BT לא הייתה ב-3CNF

כך נעבור ל-3CNF:

(א) עוברים ל-CNF (ראו בעיה)

(ב) עוברים מ-CNF ל-3CNF. למשל

$$\underbrace{x_1 \vee x_2 \vee x_3 \vee x_4}_{\text{ספיקה}} \Longleftrightarrow \underbrace{(x_1 \vee x_2 \vee z) \wedge (\bar{z} \vee x_3 \vee x_4)}_{\text{ספיקה}}$$

עבור פסוקים עם m ליטרלים מוסיפים m-3 משתנים חדשים

$$a_1 \vee a_2 \vee \dots \vee a_m$$

$$\Rightarrow (a_1 \vee a_2 \vee z_1) \wedge (\bar{z}_1 \vee a_3 \vee z_2) \wedge (\bar{z}_2 \vee a_4 \vee z_3) \wedge \dots \wedge$$
$$\wedge (\bar{z}_{m-3} \vee a_{m-1} \vee a_m)$$

(42) 20.6.07
חישוביות

שפות NP-שלמות

CLIQUE = {<G,k> : G גרף לא מכוון שמכיל k-קליקה}

<u>טענה:</u> CLIQUE NP-שלמה

<u>הוכחה:</u>

(1) $CLIQUE \in NP$: כי אם יש k-קליקה ב-G אז יש עד קצר לכך - הקליקה עצמה.

(2) $CLIQUE \in NP\text{-hard}$: נראה $3SAT \leq_p CLIQUE$

בהינתן נוסחה φ ב-3CNF נבנה גרף G עם מספר k כך ש-φ ספיקה ↔ ב-G יש k-קליקה. נניח ש-

$$\varphi = (l_1^1 \vee l_2^1 \vee l_3^1) \wedge (l_1^2 \vee l_2^2 \vee l_3^2) \wedge \dots \wedge (l_1^k \vee l_2^k \vee l_3^k)$$

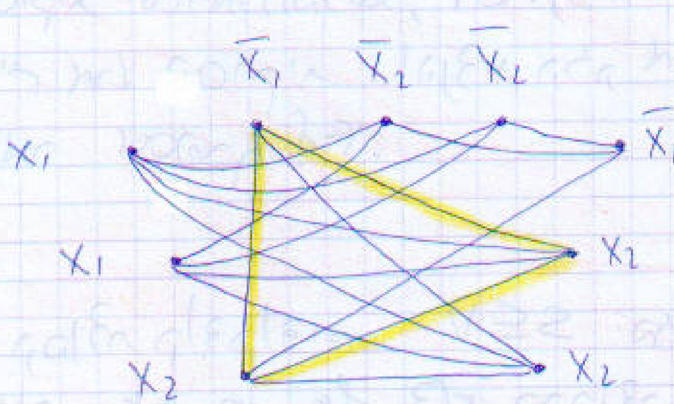
נגדיר את G באופן הבא:

V = כל הליטרלים $l_1^1, l_2^1, l_3^1, \dots, l_1^k, l_2^k, l_3^k$ סה"כ 3k

$E = V \times V$ מלבד צלעות בין ליטרלים באותה פסוקית ומלבד צלעות בין משתנה לשלילתו.

k הוא מספר הפסוקיות ב-φ

<u>דוגמא:</u> $\varphi = (x_1 \vee x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2 \vee x_2)$



k=3

<u>השמה מספקת:</u>
$f(x_1) = 0$
$f(x_2) = 1$

הרדוקציה הזו פולינומיאלית כי ב-V יש 3k קודקודים (שלה פחות מגודל הנוסחה) וב-E יש לכל היותר $(3k)^2$ צלעות וגם זה פולינומיאלי.

נראה ש-φ ספיקה $\longleftrightarrow$ יש ב-G k-קליקה.

נניח ש-φ ספיקה. תהי $f:\{x_i\}\to\{0,1\}$ השמה מספקת.

$\Leftarrow$ לכל פסוקית יש לפחות ליטרל אחד שמקבל ערך 1 תחת f. נבחר ליטרל יחיד כזה מכל פסוקית. קבוצת הקודקודים המתאימה היא k-קליקה. יש שתי סיבות שבגללן לא תהיה צלע בין שני ליטרלים - שהם מאותה פסוקית, אבל זה לא המצב שלנו

- שהם משתנה ושלילתו, אבל גם זה לא יתכן כי הנחנו שההשמה נותנת 1 לכל הליטרלים שבחרנו.

$\Leftarrow$ בין כל שני ליטרלים שבחרנו יש צלע, כלומר זו k-קליקה.

כעת, נניח שיש k קליקה בגרף. מאותו טיעון למעלה בהכרח k הקודקודים האלה הם מפסוקיות שונות. נגדיר השמה f שנותנת 1 לכל אחד מהליטרלים האלה ושאר המשתנים שלא הופיעו בקליקה לא חשוב. אז לפי ההגדרה ההשמה הזו מספקת (מכיוון שבכל פסוקית בדיוק ליטרל שמקבל 1).

נשים לב שאכן מקבלים כאן השמה חוקית. משום שהליטרלים היו בקליקה לא יתכן שיש ביניהם ליטרל ושלילתו (כי לפי בניית G אין ביניהם צלע). וההשמה מספקת כי מכיוון שקודקודים מאותה פסוקית לא מחוברים, ויש לנו k-קליקה אז בהכרח יש לנו נציג מכל פסוקית, והנציג הלז מקבל ערך 1 בהשמה.

$\Leftarrow$ ההשמה מספקת ל-φ.

<u>הגדרה</u>: קבוצת קודקודים $S\subseteq V$ בגרף $G=\langle V,E\rangle$ נאמנות נקראת בלתי תלויה אם אין אף צלע בקבוצה. כלומר לכל $u,v\in S$ $(u,v)\notin E$.

$IS=\{\langle G,k\rangle :$ G גרף לא מכוון המכיל קבוצה בת"ל בגודל k $\}$

טענה: IS NP-שלמה.

הוכחה:

1) ברור ש- $IS\in NP$ - ננחש קבוצה בגודל k ונבדוק שאין בה שני קודקודים מחוברים. הבדיקה היא כמובן פולינומית.

2) $IS\in NP\text{-hard}$ - נראה ש- $CLIQUE\le_p IS$ זה די פשוט. בהינתן $\langle G,k\rangle$ נבנה $\langle G',k'\rangle$ כך ש- ב- G יש k-קליקה אמ"מ ב- G' יש קבוצה בת"ל בגודל k.

אז ניקח את G' להיות הגרף המשלים, כלומר $G'=\langle V, V\times V\setminus E\rangle$ ו- $k'=k$.

עכשיו ברור מההגדרה של קליקה ושל קבוצה בת"ל שב- G יש k-קליקה אמ"מ ב- G' יש קבוצה בת"ל בגודל k. $S\subseteq V_G$ היא קליקה אמ"מ $S\subseteq V_{G'}$ היא בת"ל.

11

הגדרה: כיסוי (בקודקודים) של גרף $G=\langle V,E\rangle$ היא קבוצה $C\subseteq V$ כך שלכל $(u,v)\in E$ מתקיים $u\in C$ או $v\in C$ או שניהם (כלומר C היא קבוצת קודקודים שנוגעת בכל הצלעות בגרף).

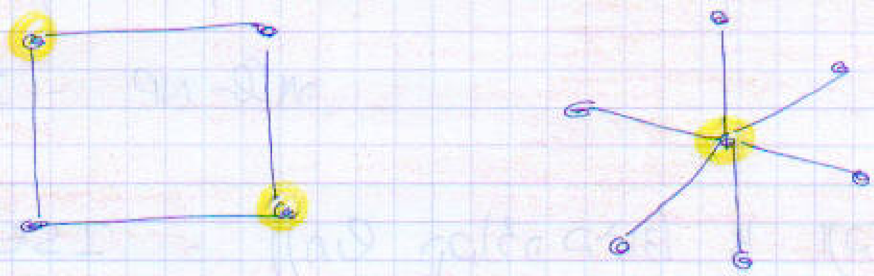
$VC=\{\langle G,k\rangle :$ G גרף לא מכוון עם כיסוי בגודל k $\}$

טענה: VC NP-שלמה.

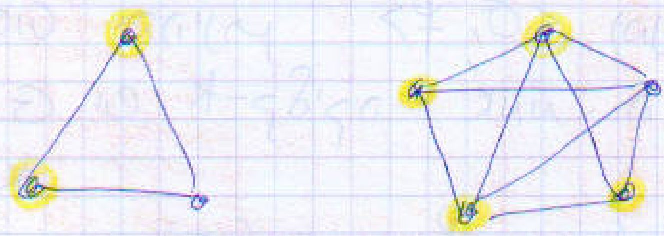
הוכחה:

1) $VC\in NP$ כי ננחש קבוצה בגודל k ונבדוק אם היא כיסוי. זאת כמובן בדיקה פולינומיאלית והכיסוי יהיה עד קבלה.

דוגמאות:



לגרף שלם עם k קודקודים יש כיסוי עם $k-1$ קודקודים.



(2) בהינתן (G,k) נבנה (G',k') כך ש-

ב-G יש קב' בת"ת בגודל k $\longleftrightarrow$ ב-G' יש כיסוי בגודל k'.

ניקח $G'=G$

$k'=n-k$ כאשר $n=|V|$.

<u>למה</u>: בגרף לא מכוון G

קבוצה $S\subseteq V$ היא בת"ת $\longleftrightarrow$ $\bar{S}=V\setminus S$ כיסוי.

<u>הוכחה</u>: S קבוצה בת"ת ב-G

$\longleftrightarrow$ אין צלעות בתוך S

$\longleftrightarrow$ כל צלע נוגעת לפחות בקודקוד אחד ב-$\bar{S}$

$\longleftrightarrow$ $\bar{S}$ כיסוי.

הלמה מוכיחה לנו את הרדוקציה. ברור כמובן שהיא פולינומיאלית.

סיכום ביניים: $3SAT \le_p CLIQUE \le_p IS \le_p VC$

(44) 25.6.07
תישמעות

סוף

השבוע - שיעורים + תרגולים כרגיל
שבוע הבא - הרצאה רק ביום שני ואין תרגולים (הרצאה אחרונה)
שבוע הבא הבא - תרגול חזרה.

סיבוכיות זיכרון Space complexity

עד עכשיו דיברנו על זמן, אבל גם לזיכרון יש חשיבות.

הגדרה: בהינתן מ"ט M העוצרת על כל קלט, סיבוכיות הזיכרון של M היא הפונקציה $S: \mathbb{N} \to \mathbb{N}$ כך ש $S(n)$ חסם על מספר התאים מסרט בהם M משתמשת בריצה על קלט באורך n.

$SPACE(S(n)) = \{ L :$ L ניתנת להכרעה ע"י מ"ט עם זיכרון $S(n)$ $\}$

עבור מכונה דטרמיניסטית -

1) מה עדיף, זמן $f(n)$ או שטח $f(n)$?
טענה: $TIME(f(n)) \subseteq SPACE(f(n))$
זה ברור כי בכל צעד ניתן לבקר על תא אחד לכל היותר. אז בזמן $f(n)$ ניתן לבקר לכל היותר $f(n)$ תאים.

2) $SPACE(f(n)) \subseteq TIME(f(n) \cdot 2^{O(f(n))})$
כמה קונפיגורציות שונות יש למ"ט הרצה בשטח $f(n)$?
קונפיגורציה מורכבת מצב שאנחנו, איפה הראש הקורא, ומה רשום על הסרט. ⟸ מס' הקונפיגורציות הוא $|Q| \cdot f(n) \cdot |\Gamma|^{f(n)}$
אם אנחנו במ"ט דט' והגענו פעמיים לאיזו קונפיגורציה אז אנחנו נחזור אליה שוב ושוב. אז אם החישוב עוצר, לא ייתכן להיות שהגענו לאותה קונפיגורציה פעמיים. ⟸ חסם על זמן הריצה הוא מס' הקונפיגורציות. אבל $|Q|$ ו- $|\Gamma|$ קבועים ⟸ $f(n) \cdot 2^{O(f(n))}$

דוגמה: נוכיח SAT ניתנת לפתרון במקום ליניארי.
בהינתן קלט <φ>, עבור נוסחה בוליאנית φ
1) לכל השמה אמת f למשתנים $x_1...x_m$ נשערך את φ ביחס ל-f.
2) אם השערוך ל-true, קבל. אחרת נעבור להשמה הבאה.
3) בהשמה האחרונה, אם גם היא נכשלה, דחה.
המקום הדרוש:
- השמת האמת הנוכחית (m תאים)
- שערוך הנוסחה לפי ההשמה הנוכחית ($O(|φ|)$ תאים)
אז $SAT \in NP\text{-}Complete$ וגם $SAT \in Linear\text{-}Space$.

דוגמה: אוטומט A הוא ריק אם $L(A) = \emptyset$
אוטומט A הוא אוניברסלי אם $L(A) = \Sigma^*$

בהינתן A דטרמיניסטי הכרע האם $L(A) = \Sigma^*$
$ALL_{DFA} = \{<A> : L(A) = \Sigma^*$ דטרמיניסטי A $\}$
זאת בעיה קלה. $L(A) = \Sigma^*$ אמ"מ $L(\bar{A}) = \emptyset$.
והמשלים ניתן לבנות בקלות, בגלל שהוא דטרמיניסטי. אין בעיה לבנות את $\bar{A}$ ולבדוק אם $L(\bar{A}) = \emptyset$ (זה הפסוק)(בודק אם יש מסלול מ-q_0 למצב מקבל כלשהו). אז $ALL_{DFA} \in P$.

ומה לגבי אוטומטים לא דטרמיניסטיים? זה קצת יותר מסובך
$ALL_{NFA} = \{<A> : L(A) = \Sigma^*$ - ו א"ס A $\}$
$\overline{ALL_{NFA}} = \{<A> : L(A) \neq \Sigma^*$ - ו א"ס A $\}$
נשים לב שאנחנו לא יודעים אם $ALL_{NFA} \in NP$ או $ALL_{NFA} \in Co\text{-}NP$.
אבל יש לנו אלגוריתם לא דטרמיניסטי שהוא באורך פולינומיאלי:

טענה: $L(A) \neq \Sigma^*$ אמ"מ קיימת מילה w באורך $2^{|Q|}$ לכל היותר כך ש- $w \notin L(A)$.

הוכחה:

($\Rightarrow$) לא ברור!

($\Leftarrow$) $L(A) \neq \Sigma^*$ $\Rightarrow$ $L(\bar{A}) \neq \emptyset$. מכיוון של- $\bar{A}$ יש $2^{|Q|}$ מצבים לכל היותר אז $L(\bar{A}) \neq \emptyset$ אם יש מילה w באורך $2^{|Q|}$ רק ש- $w \in L(\bar{A})$. (!!)

כלומר: קיימת מילה w באורך $2^{|Q|}$ לכל היותר רק ש- $w \notin L(A)$ אמ"מ קיימת סדרה של קבוצות של מצבים $S_0, S_1, \ldots, S_{2^{|Q|}}$ שמתאימה לריצה של האוטומט הדטרמיניסטי המתקבל מ-A ע"י הפעלת ה- subsetconstruction ויש i רק ש- $S_i \cap F = \emptyset$

(כלומר $S_0 = Q_0$ ולכל $i \geq 0$ קיימת אות σ_i רק ש- $S_{i+1} = \delta(S_i, \sigma_i)$ ויש i רק ש- $S_i \cap F = \emptyset$ - כ"א הריצה של האס"ד $detA$ על $\sigma_0 \ldots \sigma_i$ מגיעה למצב לא מקבל.

עכשיו נראה מ"ט א"ד עבור ALL_{NFA}:

על קלט $\langle A \rangle$ המקודד NFA:

1) כתוב שהסרט את המצבים ב- Q_0

2) אתחל מונה ל-0

3) כל עוד המונה קטן מ- $2^{|Q|}$ (לוקח $|Q|$ תאים)

– אם הקבוצה S של הסרט מקיימת $S \cap F$, דחה את $\langle A \rangle$ (כי מצאנו מילה שנדחית)

– אחרת, נחש אות a, עדכן את הקבוצה של הסרט ל- $\delta(S, a)$ והגדל את המונה ב-1.

4) דחה.

סיבוכיות הזיכרון:

- קבוצת מצבים $O(|Q|)$
- מונה $O(|Q|)$
- מצביעים עבור הזיכרונים - $O(1)$

סה"כ - $O(|Q|)$

משפט <u>Savitch</u>: לכל $s: \mathbb{N} \to \mathbb{N}$ כך ש- $s(n) \geq n$

$$NSPACE(s(n)) \subseteq SPACE(s^2(n))$$

(כאן SPACE רק עם מ"ט א"ד)

<u>הוכחה</u>:

בהינתן מ"ט א"ד M עם סיבוכיות זיכרון $s(n)$ נבנה מ"ט דטרמיניסטית שקולה עם סיבוכיות זיכרון $s^2(n)$.

הנחות בה"כ על M:

1- יש קונפיגורציה מקבלת יחידה C_{acc}

2 - יש קונפיגורציה התחלתית יחידה C_{init}

3 - d קבוע כזה כך שאין ל- M יותר מ- $2^{d \cdot s(n)}$ קונפיגורציות שונות (יש כזה כי מספר הקונפיגורציות חסום ע"י $|\Gamma|^{s(n)} \cdot s(n) \cdot |Q|$)

נבנה שגרה דטרמיניסטית $reach(C_1, C_2, t)$ שמכריעה האם ניתן לעבור מקונפיגורציה C_1 לקונפיגורציה C_2 ב- t צעדים לכל היותר.

סיבוכיות המקום של reach - $O(\log t + s(n) \cdot \log(t))$

המכונה 'M תריץ את $reach(C_{init}, C_{acc}, 2^{d \cdot s(n)})$

נשים ♡:

1) לכל w 'M מקבלת את w אמ"מ M מקבלת את w.

2) סיבוכיות הזיכרון של 'M:

$$O(\log_2 2^{ds(n)} + s(n) \cdot \log(2^{ds(n)})) = O(s^2(n))$$

אז נתאר את השיטה reach.

הרעיון הוא לעבור על כל הקונפיגורציות של M ולבדוק האם $reach(C_1, c, \lceil \frac{t}{2} \rceil)$ וגם $reach(c, C_2, \lceil \frac{t}{2} \rceil)$.

1) אם $t=1$ בדוק האם $C_1 = C_2$ או שיש מעבר של צעד אחד מ- C_1 ל- C_2.

2) אם $t>1$ אז לכל קונפיגורציה c של M המשתמשת ב- $s(n)$ תאים:

2.1) הרץ את $reach(C_1, c, \lceil \frac{t}{2} \rceil)$

2.2) הרץ את $reach(c, C_2, \lceil \frac{t}{2} \rceil)$

2.3) אם שתי הריצות קיבלו, קבל

3) דחה

ניתוח הסיבוכיות:

- עומק הרקורסיה הוא כמובן $\log t$. ואם $t = 2^{d s(n)}$
אז העומק הוא $\log 2^{d \cdot s(n)} = d \cdot s(n)$
- בכל קריאה שומרים שתי קונפיגורציות ומונה - $O(\log t + s(n))$
סה"כ סיבוכיות השטח $O(\log t + s(n)) \log t$

מסקנה: - $\bigcup_k SPACE(n^k) = PSPACE = NPSPACE = \bigcup_k NSPACE(n^k)$

- $co\text{-}NPSPACE = co\text{-}PSPACE$
$co\text{-}PSPACE = PSPACE$

(47) 27.6.07
חישוביות

בשיעור שעבר ראינו:

משפט Savitch: $NSPACE(S(n)) \subseteq SPACE(S^2(n))$

בפרט
$$\begin{array}{ccc} NPSPACE & = & PSPACE \\ \shortparallel & & \shortparallel \\ co\text{-}NPSPACE & & co\text{-}PSPACE \end{array}$$

עוד עובדות של PSPACE:

1. $P \subseteq PSPACE$ (לא ידוע אם כל הכלה ממש)
2. $NP \subseteq PSPACE$ (לא ידוע אם כל הכלה ממש)
3. $PSPACE \subseteq EXPTIME$ (לא ידוע אם כל הכלה ממש)

בינתיים תמונת העולם שלנו היא כזאת:

$$P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME \subseteq R$$

ולא ידוע אם ההכלות הן ממש או לא.

ידוע ש- $P \neq EXPTIME$

שלמות ב- PSPACE

שפה L היא PSPACE-שלמה אם:

1) $L \in PSPACE$
2) $L \in PSPACE\text{-hard}$, כלומר לכל $L' \in PSPACE$ $L' \le_p L$.

<u>משפט</u>: אם L היא PSPACE-שלמה ו- $L \in P$ אז $P = PSPACE$

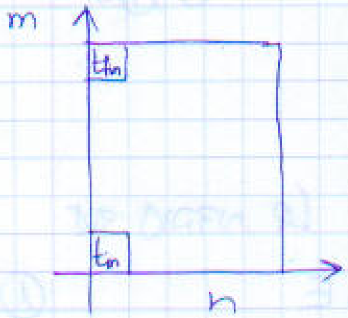
<u>הוכחה</u>: תהי $L' \in PSPACE$. נראה אלגוריתם פולינומיאלי ל- L': בהינתן w, נחשב(*) $f(w)$ ונריץ את האלג' הפולינומיאלי של L. ☺

<u>הערה</u>: אם בהגדרה היינו דורשים ש- $L' \le_{PSPACE} L$ אז (*) היה נופל.

SPACE-BOUNDED TILING

$I = \langle T, H, V, t_{in}, t_{fin}, n \rangle$

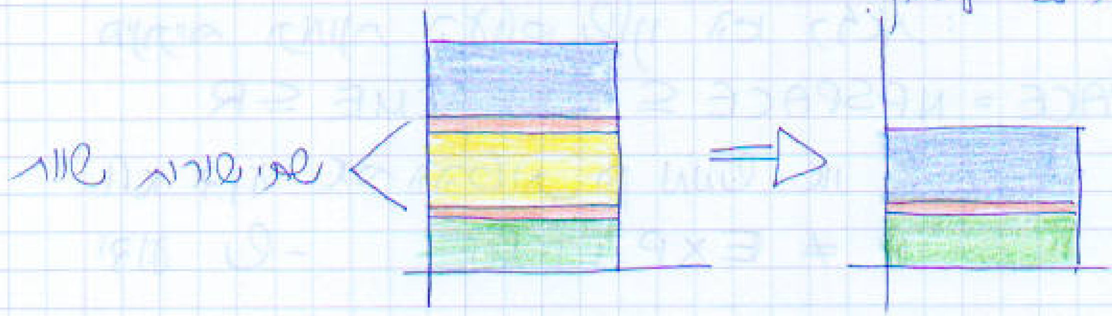
נתון באונרית



האם יש ריצוף $m\times n$ חוקי עבור m כלשהו.

שאלה שקולה: האם יש ריצוף חוקי $m\times n$ עבור $m \le |T|^n$.

למה זה שקול? מספר השורות השונות שיכולות להיות הוא $|T|^n$. אם יש ריצוף חוקי $m\times n$ עבור m כלשהו, אז יש ריצוף חוקי $m' \times n$ שבו אין שורה שחוזרת על עצמה.



לא קשה לראות ש-SBT $\in$ PSPACE-complete.

Truth Quantified Boolean Formulas

נוסחה בוליאנית = נוסחה שמופיעים בה 0, 1, משתנים בוליאניים והקשרים $\wedge, \vee, \neg$.

עבור משתנה x נוסיף כמתים בוליאניים $\forall x$, $\exists x$

משפט = נוסחה שכל משתניה מכומתים.

למשל $\forall x \exists y ((x \vee y) \wedge (\bar{x} \vee \bar{y}))$

$TQBF = \{\langle \varphi \rangle :$ משפט נכון $\varphi\}$

משפט: TQBF ∈ PSPACE-Complete

הוכחה:

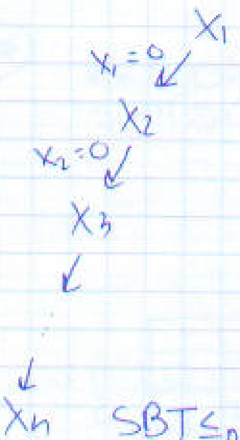
① TQBF ∈ PSPACE : בהינתן נוסחה $\forall x \varphi(x)$ או $\exists x \varphi(x)$ כאשר $\varphi(x)$ נוסחה שכל משתניה פרט ל-x מכומתים (כלומר רק x משתנה חופשי)

נעריך באופן רקורסיבי את $\varphi(0)$ ו-$\varphi(1)$

עומק הרקורסיה הוא מספר המשתנים.

כל צומת זוכר את ההצבה עד אליו ובעלים משערכים ביטוי מעל קבועים.

אבל רגע אנחנו צריכים לשמור רק מידע באורך פולינומיאלי.



② TQBF ∈ PSPACE-hard : נראה ש- $SBT \le_p TQBF$

הרעיון הוא שנבנה נוסחה $\psi_{c_1,c_2,t}$ שתהיה נכונה אמ"מ יש מעבר משורה c_1 לשורה c_2 רק שיש t שורות לכל היותר בניהן.

יש $|\Gamma|$ אותיות לכל שורה באורך n. יש $|\Gamma|^n$ שורות

$\Rightarrow$ ניתן לקודד שורה ע"י $\log|\Gamma|^n$ משתנים. $l=n\log|\Gamma|$

לנוסחה יהיו l משתנים וכל השמה אמת למשתנים תקודד שורה בריצה

$$\psi_{c_1,c_2,t} = \exists c \; \psi_{c_1,c,\frac{t}{2}} \wedge \psi_{c,c_2,\frac{t}{2}}$$

קיימת השמה אמת למשתנים.

$\psi_{c_1,c_2,1} = (c_1=c_2) \vee$ ניתן לשים את c_2 אחרי c_1

בשביל הרדוקציה נבדוק את $\psi_{c_{init},c_{fin},|\Gamma|^n}$

יש לנו בעיה: הרדוקציה אינה פולינומיאלית כי $t=|\Gamma|^n$.

$$\psi_{c_1,c_2,t} = \exists c \, \forall c_3 \, \forall c_4 \left[\left((c_3=c_1)\wedge(c_4=c)\right) \vee \left((c_3=c)\wedge(c_4=c_2)\right)\right] \longrightarrow \psi_{c_3,c_4,\frac{t}{2}}$$

$$\Longrightarrow |\psi_{c_1,c_2,t}| = O(\log t)$$

(49) 2/7/07 חישוביות

האחות התאומה של אורנה, תמי, מעבירה את השיעור!

הודעות:
- אין תרגולים השבוע
- בשבוע הבא יש תרגולי חזרה

מבנה המבחן:
- 5 מתוך 6 שאלות
- 2 אוטומטים
- 2 חישוביות
- 2 סיבוכיות

מחלקות Log Space ו- NLog Space

- נתייחס לסיבוכיות מקום
- מותר להשתמש בזיכרון נוסף שהוא לוגריתמי בגודל הקלט (מלבד סרט הקלט).

בזמן כמובן אין מה לדבר על סיבוכיות לוגריתמית כי לפחות צריך לקרוא את הקלט וזה דורש n צעדים. אז אין כאן מה לדבר על זמן תת-לינארי.

אופן החישוב: - סרט קלט שיש בו קלט מאורך n
- סרט עבודה שאפשר לעשות עליו כל מה שרוצים אבל הוא מאורך $O(\log n)$.

דוגמה: $EQ = \{0^n1^n : n \in N\}$

ראינו אלגוריתם שרץ בזמן $O(n\log n)$ אבל סיבוכיות המקום הייתה $O(n)$ כי העתקנו את הקלט.

נשים ♡ שלייצוג המספר n נדרשים $\log_2 n$ מקום.

1

נוכל לפתור את הבעיה בסיבוכיות מקום $O(\log n)$.
האלגוריתם סופר את מספר האפסים והאחדים ובסוף משווה את המונים. בסרט העבודה נרשום את מספר האפסים אח"כ סימן הפרדה ואח"כ מספר האחדים. סה"כ נשתמש ב- $O(\log n)$ תאים בסרט העבודה.

<u>הגדרות:</u>

- המחלקה LOGSPACE (ולפעמים מסומנת גם ב- L):
כל השפות שיש מ"ט דטרמיניסטית שמכריעה אותן תוך שימוש בזיכרון נוסף $O(\log n)$. (אין חשיבות לזמן זמן לה תמיד יוצא ב- P)
- המחלקה NLOGSPACE (NL): כנ"ל אבל מ"ט א"ד.

<u>דוגמה:</u>

PATH={<G,s,t> : G גרף, s,t קודקודים ויש מסלול ביניהם}

אבחנות פשוטות:

- $PATH \in NP$ כי בהינתן עד (מסלול מ-s ל-t) אפשר לוודא שזה אכן מסלול
- $PATH \in P$ למשל DFS או BFS

נראה ש- $PATH \in NL$ מה אפשר לשמור ב- $O(\log n)$ ביטים?
- כל מספר קבוע של צמתים או קשתות.

המכונה תנחש מסלול מ-s ל-t. בכל רגע תשמור רק את הצומת הנוכחי במסלול. בהתחלה נרשום בסרט העבודה את <s> ובכל שלב תחליף בצומת הנוכחי.
בנוסף נחזיק בסרט העבודה מונה שמאותחל ל-0 וגדל ב-1 בכל איטרציה. למה? כי אם קיים מסלול אז קיים מסלול פשוט שיש בו לכל היותר n צמתים. אם מגיעים לשלב שנרשם <t> בסרט העבודה אז הקלט ב- PATH. אם המונה הגיע ל- |V| אז הקלט לא ב- PATH.

צריך לוודא כמה דברים:

1) שימוש ב- $O(\log n)$ מקום: ✓
כי שומרים רק צומת אחד ומונה אחד שגדל עד $|V|$ לכל היותר (ומשתמש גם את t ו-n לצורך ההשוואה)

2) קיימת ריצה מקבלת אמ"מ יש מסלול מ-s ל-t: ✓
($\Leftarrow$) ברור כי אם הגענו ל-t אז כל הצמתים שעברנו מהגרף הם מהווים מסלול
($\Rightarrow$) המכונה א"ד $\Leftarrow$ אם יש מסלול אז יש מסלול פשוט ואותו המכונה יכולה לנחש. בכל צעד יהיה רשום לנו הצומת הנוכחי במסלול ובסוף נגיע ל-$\langle t \rangle$ ולכן קיימת ריצה מקבלת.

NL - Completeness

<u>הגדרה</u>: שפה A היא שלמה ב- NL אם מתקיים:
- $A \in NL$
- $A \in NL\text{-hard}$, לומר לכל שפה $A' \in NL$ $A' \le A$
כאשר הרדוקציה משתמשת רק ב- $O(\log n)$ זיכרון

Log Space Transducer
(מתמר מקום לוגריתמי)

מתמר הוא מ"ט דטרמיניסטית M בעלת שלושה סרטים:
RO, R/W, WO.
בסרט R/W יש $O(\log n)$ תאים.
נאמר שהמכונה מחשבת פונקציה $f: \Sigma^* \to \Sigma^*$ אם לכל מילה $w \in \Sigma^*$ הכתובה בסרט ה- RO M עוצרת עם $f(w)$ כתוב על בסרט ה- WO.

דוגמה - Log Space Transducer

קלט: גרף עם משקלות $\langle V,E,W\rangle$ $\quad W: E\to N$

פלט: גרף לא ממושקל $\langle V,E'\rangle$ רק ש- $e\in E' \iff e\in E, W(e)>7$

למשל: קלט פלט

המתמר משתמש בשלושה סרטים: קלט (ארוך), עבודה (קצר), פלט (ארוך)
תחילה נעתיק את הצמתים לסרט הפלט. אח"כ נעבור על הקשתות.
כדי להחליט האם להעתיק קשת לסרט הפלט, נכתוב את המשקל שלה על סרט העבודה ונשווה ל-7.

<u>הערה</u>: משקלים בני 4 ביטים או יותר לא יועתקו לסרט העבודה בשלמותם. כבר בביט הרביעי ניתן להסיק ש- $w(e)>7$ ואז לא צריך להעתיק את הקשת.

<u>הגדרה</u>: $A'\le_L A$ אם קיים מתמר העובד במקום לוגריתמי ומחשב פונקציה f רק ש- $w\in A' \iff f(w)\in A$

<u>משפט</u>: אם $A\le_L B$ ו- $B\in L$ אז $A\in L$

<u>הוכחה</u>: בהינתן M_B עבור B נבנה M_A עבור A:

- ניסיון 1: על קלט w המכונה M_A תריץ את המתמר המחשב את $f(w)$ ותריץ את M_B על $f(w)$.

הניסיון שגוי כי ערכו $f(w)$ נחשב חלק מהחישוב ואינו הפלט. המתמר משתמש ב- $O(\log|w|)$ מקום רק כדי לייצר את $f(w)$ אבל עכשיו נצטרך כשלב ביניים לכתוב את $f(w)$.

- ניסיון 2 (מוצלח): על קלט w המכונה תחשב באיטרציה את האות מתוך $f(w)$ שאותה M_B צריכה כרגע (אף פעם לא נרשום את כל $f(w)$) ייתכן שנריץ את המתמר המון פעמים ושיהיו ביטים שנחשב מספר פעמים אבל מותר לנו! מכונה זו מכריעה את A תוך שימוש ב- $O(\log n)$ מקום.

(51) 1

השפה הבאה היא NL-complete:

$PATH = \{\langle G,s,t\rangle :$ קדקודים שמסלול ביניהם s,t , גרף $G\}$

נראה ש- $PATH \in NL\text{-hard}$ ע"י שנראה רדוקציה Log Space מכל שפה A ב- NL.

תהי M_A מ"ט ("תמך א"ד) המכריעה את A במקום $O(\log n)$.
בהינתן קלט w ל- M_A נייצר $\langle G,s,t\rangle$ כך ש- M_A מקבלת את w אמ"מ $\langle G,s,t\rangle \in PATH$

צמתי G יהיו קונפיגורציות של M_A בריצה על w

s - קונפיגורציה התחלתית
t - קונפיגורציה מקבלת
ניתן להניח שהן יחידות.

איך נשמור קונפיגורציה? - מצב, מיקום ראש קורא, תוכן הסרט
סה"כ יש $|Q|\cdot n\cdot|\Gamma|^{O(\log n)}$ קונפיגורציות.
בשביל לייצג קונפיגורציה צריך סרט בגודל:

$$\log(c_1\cdot n\cdot c_2^{O(\log n)}) = \log c_1 n + \log c_2^{O(\log n)} = O(\log n)$$

המשרן: עובר בסדר לקסיקוגרפי על כל המילים באורך $c\cdot\log n$ ומעתיק לסרט הפלט את כל המילים שמייצגות קונפיגורציה (אלו הצמתים של G)

אח"כ המשרן עובר על כל זוגות המילים האלו ובודק האם הזוג מייצג שתי קונפיגורציות עוקבות. אם כן, מעתיק לסרט הפלט (אלו הקשתות). בהמשך מזהה את הקונפיגורציה ההתחלתית והמקבלת ומעתיק גם אותן.

לסיכום, לכל $A \in NL$, $A \leq_L PATH \Rightarrow PATH \in NL\text{-complete}$

משפט: $NL \subseteq P$

הוכחה: (ידוע ש- $L \subseteq P$) בהינתן בעיה ב- NL נוריד אותה ב- Log Space ל- PATH. אבל $PATH \in P$ והרדוקציה שמשמשת היא פולינומיאלית $\Rightarrow$ הבעיה ב- P.

ידוע $L \subseteq NL \subseteq P$

לא ידוע האם יש שוויונים!

סוף