

סוכם על ידי אבי שוע –

shuaavi@gmail.com

<http://www.cs.huji.ac.il/~shuaavi>

אני מקווה שהסיכומים יעזרו לכם ולעוד רבים – אבל אני חייב להעיר – יתכן, ואפילו סביר, שיש בה טעויות. **אני (ואף אחד אחר) לא לוקח אחריות** לאף ציון / בזק שיגרם בגלל הסיכום.

אבי שוע

28.2.2007

מתרגל: רובי.

אתר: <http://www.cs.huji.ac.il/~compu>

מייל: robil@cs.huji.ac.il

היום נדבר על:

1. קבוצות, יחסים, עצמות
2. אוטומטים סופיים דטרמיניסטיים

קבוצה: אוסף של איברים, ללא חשיבות לסדר או למספר ההופעות של כל איבר.

לכן - $\{a, b, c\} = \{c, b, a\} = \{c, a, b\}$.

יהו A, B קבוצות, אזי $A \times B$ היא המכפלה **הקרטזית** והיא כל הזוגות הסדורים בהם האיבר הראשון הוא מ A והשני מ B .

$$A \times B = \{\langle a, b \rangle \mid a \in A \wedge b \in B\}$$

יחס בינארי R בין A ל B הוא קבוצה חלקית למכפלה הקרטזית, כלומר $R \subseteq A \times B$.

אם $\langle a, b \rangle \in R$ נסמן $R(a, b)$ או aRb .

לדוגמא: מתקיים $\mathbb{N} \times \mathbb{N} \subset '<'$. נסמן $5 < 7$ או $(5, 7) <$.

יחס נוסף: יהי $G = \langle V, E \rangle$, אזי $E \subseteq V \times V$.

יחס $R \subseteq A \times A$ נקרא:

1. רפלקסיבי – אם לכל $a \in A$ מתקיים aRa
2. סימטרי - $aRb \Rightarrow bRa$
3. טרנזיטיבי $aRb \wedge bRc \Rightarrow aRc$

כעת יש כמה דוגמאות של יחסים והאם הם רפלקסיביים, סימטריים, וכיו"ב. קיים בסיכום דיסקרטית למעוניינים (או אצל דינה).

הגדרה: יחס שקילות – רפלקסיבי, סימטרי, טרנזיטיבי. יחס שקילות מחלק קבוצה **למחלקות שקילות**.

הגדרה פורמלית: – יהי R יחס שקילות על A , $a \in A$, אזי $[a]_R = \{b \in A \mid aRb\}$ **היא מחלקת**

השקילות של A .

הגדרה: **חלוקה** של קבוצה היא אוסף זר של קבוצות חלקיות ל A , המכסות את A .

טענה: אוסף מחלקות השקילות לפי יחס השקילות R על A מהווה חלוקה של A .

דוגמא: $R = " = \text{mod } 3 "$ על $\mathbb{N}$. אזי $[7]_R = \{1 + 3a \mid a \in \mathbb{N}\}$.

הגדרה: **פונקציה** מ A ל B (נסמן $f : A \rightarrow B$) הוא יחס $f \subseteq A \times B$ שבו לכל $a \in A$ יש איבר יחיד $b \in B$ כך ש afb .

פונקציה תיקרא **חח"ע** (חד חד ערכית) אם לכל $a_1 \neq a_2 \in A$ מתקיים $f(a_1) \neq f(a_2)$.

פונקציה תיקרא **על** אם לכל $b \in B$ קיים $a \in A$ כך ש $f(a) = b$.

אם פונקציה היא חח"ע ועל, נאמר שהיא הפיכה – כלומר קיימת $f^{-1} : B \rightarrow A$ כך

$$f^{-1}f = ff^{-1} = Id$$

עבור קבוצה סופית A העוצמה של A , $|A|$ היא מספר האיברים (השונים) שבה.
הגדרה: עבור שתי קבוצות A, B כלשהן (לאו דווקא סופיות), נאמר ש $|A| = |B|$ (שוות עוצמה) אם יש פונקציה חז"ע ועל ביניהן.

$$|N| = |2N| \text{ (זוגיים)}$$

$$|N| = |Z|$$

$$\text{גם } |N| = |Q|.$$

דוגמאות נוספות והסברים כאן (או אצל דינה)

http://he.wikipedia.org/wiki/%D7%94%D7%9E%D7%9C%D7%95%D7%9F_%D7%A9%D7%9C_%D7%94%D7%99%D7%9C%D7%91%D7%A8%D7%98

הגדרה: A היא **בת מניה** אם היא סופית או זהה בעוצמתה לקבוצת הטבעיים. נסמן $|N| = \aleph_0$ (העוצמה האינסופית הקטנה ביותר).

טענה: קיימת קבוצה שאינה בת מניה - $\mathbb{R}$.

הוכחה: מספיק להראות שאין פונקציה חז"ע ועל $f: \mathbb{N} \rightarrow \mathbb{R}$. נניח בשלילה שיש פונקציה כזו,

$$f: \mathbb{N} \rightarrow [0, 1] \text{ נגדיר } a_n = (f(1), f(2), \dots)$$

נניח בה"כ כי הסדרה נראית כך:

$$f(1) = 0.13569...$$

$$f(2) = 0.75963...$$

מהנחת השלילה, מופיעים בה כל המספרים הממשיים בין $[0, 1]$.

קנטור מציע את **שיטת הלכסון**: עבור כל מספר i , נסתכל בספרה ה i אחרי היחידה. כעת, נחליף כל פעם את הספרה ה i בספרה אחרת (אם היה רשום שם $1 -$ נשים 2 , וכך הלאה). את כל הספרות שהחלפנו, ניקח למספר חדש (הספרה הראשונה שלו - תהיה הספרה שבחרנו כשונה מהספרה הראשונה של הראשון, וכך הלאה). נניח בשלילה שהמספר הנ"ל מופיע במקום ה k בסדרה - זה לא יכול להיות, כי הספרה ה k ית של x שונה מהספרה ה k ית של אותו מספר.

הוכחה אלטרנטיבית: תהי S קבוצה - אזי 2^S היא קבוצת העצמה (כל התת קבוצות של S). הוכחנו בדיסקרטית כי $|2^S| > |S|$.

אוטומטים סופיים דטרמיניסטיים - DFA

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

- Q - קבוצת מצבים
- Σ - א"ב של הקלט
- $\delta: Q \times \Sigma \rightarrow Q$ - פונקציית מעבר
- q_0 - מצב התחלתי
- $F \subseteq Q$ - מצביים מקבלים.

עבור מילה $w = w_1 w_2 \dots w_n \in \Sigma^*$ (תזכורת: Σ^* - אוסף כל המילים, כולל הריקה, מעל Σ).

ריצה של A היא סדרה $r = r_0 r_1 \dots r_n$ כאשר:

$$1. r_0 = q_0,$$

$$\forall 1 \leq i \leq n, r_i = \delta(r_{i-1}, w_i) \quad 2.$$

נגדיר: $\delta^* : Q \times \Sigma^* \rightarrow Q$ באופן רקורסיבי:

$$\delta^*(q, \varepsilon) = q$$

$$\delta^*\left(q, \underbrace{ua}_{\substack{u \in \Sigma^* \\ a \in \Sigma}}\right) = \delta(\delta^*(q, u), a)$$

נגדיר יחס שקילות - $A \subseteq \Sigma^* \times \Sigma^*$, אזי אם $w \in \Sigma^*$, אזי $[w]_A = \{u \in \Sigma^* : \delta^*(q_0, w) = \delta^*(q_0, u)\}$ (כל המילים שמגיעות לאותו מצב).

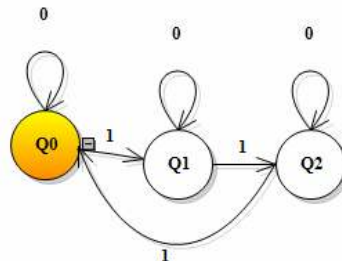
בעצם, מתקיים $L(A) = \bigcup_{\delta^*(q_0, w)} [w]_A$ (כזכור $L(A)$ - היא השפה של A).

דוגמא לאוטומט:

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

$$L = \left\{ w = w_1 \dots w_n \mid \sum_{i=1}^n w_i = 0 \bmod 3 \right\}$$

הוא יראה כך: (נקרא לו A)



טענה I: $L(A) = L$

טענה II: $\{w : \delta^*(q_0, w) = q_0\} = L$

$$\delta^*(q_0, w) = q_j$$

$$q_j = \sum_{i=1}^n w_i \bmod 3 \quad \text{טענה III:}$$

הוכחה: באינדוקציה על אורך w , כלומר $|w|$:

בסיס: $|w| = 0$ ולכן $w = \varepsilon$, ואכן $\delta^*(q_0, \varepsilon) = q_0$

צעד האינדוקציה:

$$\delta^*(q_0,w)=\delta\big(\delta^*(q_0,w_1...w_{n-1}),w_n\big)=\delta\left(q_{\sum_{i=1}^{n-1}w_i\text{ mod }3},w_n\right)=\begin{cases}q_{\sum_{i=1}^{n-1}w_i\text{ mod }3},w_n=0\\q_{\sum_{i=1}^{n-1}w_i+1\text{ mod }3},w_n=1\end{cases}$$

6.3.2007

ציטוט השיעור: 'דברו איתי! אני יושב פה ואמלל, מדבר עם עצמי' (מתרגל אלמוני לחישוביות)

תרגול 2

אוטומטים לא דטרמיניסטיים – הסרת צעדי ε .

נניח ואנו מתעניינים בשפה $\{u \circ v \mid u \in L_1, v \in L_2\}$.

נשים לב שאי אפשר פשוט 'לשרשר' את האוטומטים – לדוגמא, אם $L_1 = \text{מס' אפסים זוגי}$, $L_2 = \text{מספר אפסים שמתחלק ב-5}$ – אם המילה היא 10 פעמים 0, אזי שרשור לא יפתור לנו את הבעיה – אבל היא שייכת לשפה – אם נחתוך את זה ל 0 אפסים (L_1), ו 10 ל L_2 . (כן יעבוד אם המילה היא 7 פעמים 0, אז כן).

הבעיה היא מתי עוברים מהאוטומט של L_1 לזה של L_2 . זו לא בעיה קלה. כדי לפתור את זה, עוברים למודל לא דטרמיניסטי. בהמשך, נוכיח שמודל לא דטרמיניסטי שקול לדטרמיניסטי, ולכן זה פתרון טוב.

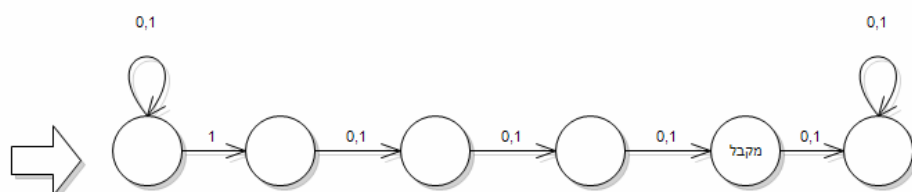
אוטומט סופי לא דטרמיניסטי (NFA):

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

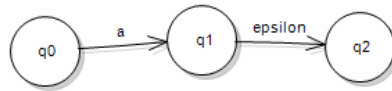
- Q – קבוצת מצבים
 - Σ – א"ב. נסמן $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$
 - $Q_0 \subseteq Q$ – קבוצת מצבים תחיליים – שונה מ DFA!!!
 - $\delta : Q \times \Sigma_\varepsilon \rightarrow 2^Q$ (קבוצת תתי הקבוצות של Q) – פונקציית המעברים (או יחס המעברים). ניתן לרשום גם $\delta \subseteq Q \times \Sigma_\varepsilon \times Q$
 - $F \subseteq Q$ – זו קב' מצבים מקבלים.
- $r = r_0 \dots r_n$ סדרת מצבים היא ריצה של A על המילה w אם ניתן לכתוב את w כ $w = y_1 \dots y_n$ כשלכל $1 \leq i \leq n$ $y_i \in \Sigma_\varepsilon$ ומתקיים:
1. $r_0 \in q_0$
 2. לכל $1 \leq i \leq n$ מתקיים $r_i \in \delta(r_{i-1}, y_i)$

ריצה היא ריצה מקבלת אם $r_n \in F$ (כלומר – המצב האחרון שייך למצבים המקבלים). מילה מתקבלת ע"י האוטומט A אם קיימת עלייה ריצה מקבלת.

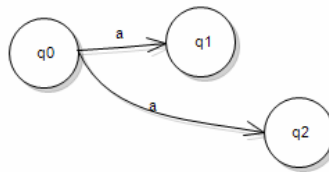
נניח שאנו רוצים לתאר את השפה $L = \text{כל המחרוזות של } 0,1 \text{ ים שבהם מופיע } 1 \text{ בספרה החמישית לפני סוף המילה. אוטומט מתאים הוא:}$



הסרת צעדי ε



נבנה ויש לנו חלק באוטומט כדוגמת



אז הוא שקול ל

אם q_1 היה מצב התחלתי, אזי היינו הופכים את q_2 גם להתחלתי.

משפט: בהנתן אוטומט NFA $A = (Q, \Sigma, Q_0, \delta, F)$ ניתן לבנות אוטומט NFA $A^* = (Q, \Sigma, Q_0^*, \delta^*, F)$ כך שב A^* אין מעברי ϵ וכן $L(A) = L(A^*)$, ויתר על כן – הבניה היא בזמן פולינומיאלי.

הגדרה: יהי q מצב, אזי $E(q) = \{q' \in Q \mid * \}$ כך שיש מסלול במעברי ϵ מ q ל q' ב 0 או יותר מעברים).

עבור $S \subseteq Q$ נגדיר $E(S) := \bigcup_{q \in S} E(q)$.

אלגוריתם:

$E(q)$
 $S_0 \leftarrow \{q\}$
 repeat
 $S_{i+1} \leftarrow S_i \cup \{q \mid \exists q'' \in S_i, q' \in \delta(q'', \epsilon)\}$
 until $S_{i+1} = S_i$

סימון – עבור i שמוחזר נסמן $S_i = S_\infty$.

טענה: חישוב $E(q)$ עוצר.

הוכחה: לכל i מתקיים $S_i \subseteq S_{i+1}$. לכן, אם האלגוריתם לא עצר ב S_{i+1} , אזי יש לפחות מצב אחד שאינו ב S_i .

משום שלכל i מתקיים $S_i \subseteq Q$ ברור שהאלגוריתם עוצר תוך לכל היותר $|Q| = n$ צעדים. כל צעד לוקח זמן פולינומיאלי $O(n^2)$. נחשב את $E(q)$ לכל מצב q סה"כ חישוב כל ה $E(q)$ ים לוקח $O(n^4)$.

נכונות: מוכיחים באינדוקציה ש S_i היא קבוצת המצבים שניתן להגיע אליהם מ q ב i צעדים (או פחות).

בהינתן $NFA A = (Q, \Sigma, Q_0, \delta, F)$ (עם מעברי ε) נגדיר $NFA A^* = (Q, \Sigma, Q_0^*, \delta^*, F)$ ללא מעברי ε , כך:

$$Q_0^* = E(Q_0) = \bigcup_{q \in Q_0} E(q), \quad \Sigma \text{ הוא } A^* \text{ של } Q.$$

$$\delta^* \text{ מוגדר ע"י } \forall r \in \Sigma, \delta^*(q, r) = E(\delta(q, r)) = \bigcup_{q' \in \delta(q, r)} E(q')$$

$$L(A^*) = L(A)$$

עבור מילה $w = w_1 \dots w_n$ מעל Σ יש ריצה $r_0 \dots r_n$ של A^* על w המסתיימת במצב q אם יש ריצה $t_1 \dots t_k$ של A על w .

אם יש ריצה של A^* על w המסתיימת ב q אז יש ריצה של A על w המסתיימת ב q .
נוכיח באינדוקציה על אורך המילה $w = w_1 \dots w_n$.

בסיס: עבור $n = 0$ מתקיים $w = \varepsilon$ ריצה של A^* על ε היא מצב התחלתי, $q \in Q_0^*$, וע"פ ההגדרה קיים $q_0 \in Q_0$ כך ש $q_0 \in E(q_0)$ לכן קיים מסלול של מעברי ε $x_0 = q_0$ כך ש $x_l = q$ ולכן יש ריצה של A על ε המסתיימת ב q .

שלב האינדוקציה: נניח נכונות עבור n , נוכיח עבור $n + 1$. $w = w_1 \dots w_{n+1}$ ויש ריצה של A^* $r_0 \dots r_{n+1}$ ($r_{n+1} = q$). כלומר ריצה של A^* על המילה $w_1 \dots w_n$ לכן יש ריצה של A על $w_1 \dots w_n$ המסתיימת ב r_n . נסמן $z_0 \dots z_l$ ו $z_l = r_m$. $r_{m+1} \in \delta^*(r_n, w_{n+1})$. כמו במקרה בסיס, יש $q' \in \delta(q, w_{n+1})$ ויש מסלול $q' = a_1 \dots a_m = q$ של מעברי ε ולכן $z_0 \dots z_l a_1 \dots a_m$ ריצה של A על w שמסתיימת ב q .

כיוון שני – לא נוכיח פורמלית, אבל אם יש ריצה של A על w המסתיימת ב q , אז יש ריצה של A^* על w המסתיימת ב q . נוכיח באינדוקציה על מספר רצפי ε ים במילה.

13.3.2007

ביטויים רגולריים

סינטקס	סמנטיקה
ε	$\{\varepsilon\}$
$\emptyset$	$\emptyset$
$\forall a \in \Sigma, a$	$\{a\}$
אם r_1, r_2 ביטויים רגולריים ששפותיהן L_1, L_2 $r_1 + r_2$	$L_1 \cup L_2$
אם r_1, r_2 ביטויים רגולריים ששפותיהן L_1, L_2 $r_1 \cdot r_2$	$L_1 \cdot L_2 = uv \mid u \in L_1, v \in L_2$
הכוכב של קליני - r_1^*	$\{w \mid \exists k \geq 0, w = w_1 \dots w_k, w_i \in L\}$

L^+ - שרשור בין 1 ל ∞ של L . (לא נדרש - ניתן למימוש באמצעות L וכוכב).

דוגמאות: $(0+1)^*$ - כל המילים מעל $\{0,1\}$.

$0(0+1)^*$ - כל המילים שמופיע בהם 0.

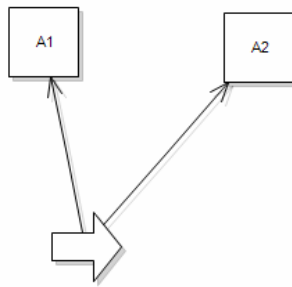
$(0+\varepsilon)(1+\varepsilon)$ - $\varepsilon, 0, 1, 01$.

$$REG = \{L \subseteq \Sigma^* \mid \text{exists regular expression } r \text{ and } L(r) = L\}$$

נראה להוכיח כי $REG = NFA (= DFA)$.

נוכיח $REG \subseteq DFA$. פשוט נבנה אוטומטים לכל אחד מהמצבים:





נניח שיש $A_{1,2}$ כך ש $L(A_i) = L(r_i)$ עבור $i = 1, 2$.
 שרשור – למד בכיתה.
 הכוכב של קליני -

למה 1: לכל NFA A בן m מצבים, קיים NFA שקול A' בן $m + 2$ מצבים כך של A' יש מצב התחלתי אחד שאף מעבר (קשת) לא נכנס אליו, וכן יש לו מצב מקבל אחד שאין ממנו מעברים. כלל.
 הוכחה: נוסיף מצב התחלתי חדש, נעבור למקוריים בצעדי ϵ , ובנוסף – מצב מקבל חדש, שנחבר אותו לכל הקודמים בצעדי ϵ , ונהפוך את כל המקבלים הקודמים נהפוך ללא מקבלים.

נשנה את האוטומט כפי שמוגדר בלמה, נהפוך את המצב הראשון למקבל, ונוסיף מצב אפסילון מהמצב המקבל (היחיד) לראשון (היחיד).

הראנו כי $REG \subseteq NFA$. נותר להראות כי $NFA \subseteq REG$.

לשם כך נגדיר את GNFA –
 כמו NFA מלבד:

1. יש מצב התחלתי אחד בלבד שאין מעברים לתוכו
2. יש מצב סופי אחד בלבד שאין מעברים ממנו
3. כל מעבר מסומן לא רק באות או ϵ אלא באיזשהו ביטוי רגולרי.

ריצה על מילה $w \in \Sigma^*$ היא סדרת מצבים $r_0..r_n$ כך שניתן לכתוב את w כ $w_1...w_n$ (תת מילה) ולכל i , המילה w_i שייכת לביטוי הרגולרי על המעבר בין r_{i-1} ל r_i .

למה 2: לכל GNFA עם שני מצבים יש ביטוי רגולרי עם אותה שפה.
 הוכחה: זה האופציה:



אסטרטגיית הוכחה: יהי A NFA עם m מצבים – המטרה – מציאת ביטוי רגולרי שקול.
 למה 1 מלמדת אותנו שאפשר לקבל מ A' GNFA עם $m + 2$ מצבים.
 למה 2 מלמדת אותנו כי אם יש A'' עם שני מצבים יש ב"ר שקול.

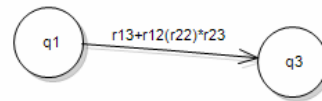
נותר להפוך GNFA עם $m + 2$ מצבים ל GNFA עם 2 מצבים. נעשה זאת רקורסיבית.

למה 3: לכל A GNFA עם m מצבים, עבור $m > 2$ קיים GNFA A' עם $n - 1$ מצבים כך ש $L(A) = L(A')$.

אינטואיציה:

יש לנו לפחות 3 מצבים q_1, q_2, q_3 . נניח שיש קשתות בין כולן. על הקשת בין q_i ל q_j יש את r_{ij} (ביטוי רגולרי). נשים לב שיתכן גם r_{ii} .

נניח שאנו רוצים להעניף את q_2 . בה"כ הוא לא התחלתי ולא מקבל. ברור שאי אפשר פשוט למחוק אותו – צריך לספק דרך אלטרנטיבית לו. נעשה זאת כך – (חשוב לשים לב ל $(r_{22})^*$) – המעבר מ q_1 ל q_3 יהפך ל $r_{13} + r_{12}(r_{22})^*r_{23}$.



אבל לא העפנו אותו מכל ריצה אפשרית. צריך לבצע את התהליך הנ"ל לכל זוג קודקודים, כולל q_1, q_1 (לדוגמא - $(r_{11} = r_{11} + (r_{12}r_{21})^*)$).

פורמלית: לכל זוג קודקודים q_1, q_3 (יתכן ש $q_1 = q_3$) ומצב q_2 שאינו מצב התחלתי / מקבל נסמן $f(q_1, q_2, q_3) = r_{13} + r_{12}(r_{22})^* r_{23}$. נביט בGNFA בו הורדנו את q_2 וכל קשת בין שני מצבים q_i, q_j החלפנו את הביטוי הרגולרי ב $f(q_i, q_2, q_j)$. נסמן ב' A את הGNFA שהתקבל.

$L(A) \subseteq L(A')$ - בהינתן $w \in L(A)$ יש ריצה $t_0 \dots t_n$ של A על w ולכן יש חלוקה של $w = w_1 \dots w_n$ כך שלכל i , w_i שייך לשפת הביטוי r_i על המעבר מ t_{i-1} ל t_i . נוכיח שיש על w ריצה מקבלת של A' המסתיימת ב t_n באינדוקציה על מספר רצפי ה q_2 בריצה $t_0 \dots t_n$.
 בסיס: טריוויאלי

צעד אינדוקציה: ע"פ הגדרת f (בדיוק כמו בהוכחת צעדי $\mathcal{E}$). נתבונן ב $t_0...t_n$. נניח ש $t_i, t_{i+1}...t_{i+l}$ רצף ה q_2 האחרון. ע"פ הנחת האינדוקציה קיימת ריצה של A' על $w_1...w_i$ המסתיימת ב t_i . נתבונן במילה $w_{i+1}...w_{i+l} \in L(r_{22})$, וכך $w_{i+2} = L(r_{22})$. באופן כללי - $w_{i+l} \in L(r_{2n})$. (או פרמוטציה כלשהיא של האינדקסים). לכן השרשור שייך לביטוי הרגולרי $r_{j2}r_{22}^*r_{2k}$.

ההוכחה בכיוון השני מאוד דומה. אין כאן עומק, אבל יש המממממממממממממוץ אינדוקסים.

למת הניפוח:

אם L רגולרית, אז קיים $p \geq 1$ (קבוע הניפוח) כך שלכל מילה w המקיימת $|w| \geq p$, יש חלוקה של

$$w = x \cdot y \cdot z \quad w$$

$$.xy^iz \in L, \forall i \geq 0 \quad .1$$

$$2. |y| > 0.$$

$$3. |xy| \leq p$$

שימושים – בעיקר כדי להוכיח ששפה אינה רגולרית. לדוגמא – נביט בשפה $L = \{1^{n^2} \mid n \geq 0\}$ נוכיח שאינה רגולרית. נניח על הדרך השלילה ש L רגולרית, אז קיים p כבלמת הניפוח. נביט במילה $1^{p^2} \in L$. ע"פ למת הניפוח, $1^{p^2} = xyz$ כך ש $|y| \geq 1$, $|xy| \leq p$, ובפרט $|y| \leq p$. ע"פ למת הניפוח, $xyyz \in L$.

מה המילה הארוכה ממש מ 1^{p^2} הקצרה ביותר ב L ? $1^{(p+1)^2} = 1^{p^2+2p+1}$. אבל, אורך $xyyz$ הוא $|xyz| + |y|$, לכן $xyyz$ אינה בשפה. סתירה.

21.3.2007

הנושאים להיום:

1. משפט Myhill-Nerole (MN)
2. מינימיזציה של אוטומטים
3. שימושים במשפט MN
4. אלוריתמים ושימושים באוטומטים

משפט MN

הגדרה: בהינתן שפה L נגדיר את יחס השקילות $\sim_L$ באופן הבא: $x, y \in \Sigma^*$
$$x \sim_L y \Leftrightarrow (\forall z \in \Sigma^*, x \cdot z \in L \Leftrightarrow yz \in L)$$

מחלקות השקילות של w לפי $\sim_L$ הוא $[w]_{\sim_L} = \{w' \mid w' \in \Sigma^*, w \sim_L w'\}$

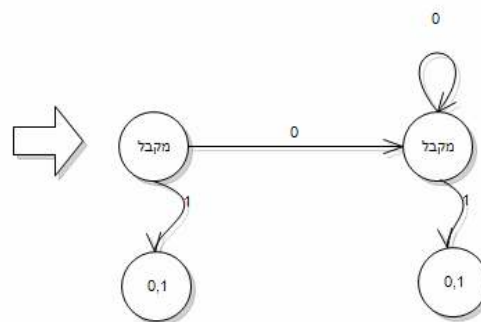
משפט: L רגולרית $\Leftrightarrow$ מספר מחלקות השקילות של $\sim_L$ הוא סופי.

תוצאה מההוכחה: בכל DFA עבור שפה רגולרית L מספר המצבים של $A \leq$ מספר מחלקות השקילות

$\sim_L$.

עבור האוטומט המינימלי מתקיים שיויון.

נתבונן ב



השפה שלו היא 0^* .

אבל כמובן יש אוטומט קטן יותר:



$$w \in \Sigma^*, q \in Q, a \in \Sigma, \delta^*(q, a) = \delta(q, a) \quad \text{בהינתן A DFA, נגדיר}$$

$$\delta^*(q, aw) = \delta^*(\delta(q, a), w)$$

נגדיר את יחס השקילות $\sim_F$ על Q באופן הבא :

$$p, q \in Q, p \sim_F q \Leftrightarrow \begin{cases} \forall w \in \Sigma^* \\ \delta^*(p, w) \in F \Leftrightarrow \delta^*(q, w) \in F \end{cases}$$

(שני מצבים הם שקולים, אם כל ריצה שמתחילה מהראשון מגיעה על מילה מסויימת למצב מקבל אממ כל ריצה שמתחילה מהשני מגיעה על אותה מילה למצב מקבל).

הרעיון של מינימציה של אוטומטים: נשים את כל המצבים באותה מחלקת שקילות, ולאט לאט נחלק אותם שוב ושוב.

לכל $i \geq 0$ נגדיר את היחס $\sim_i$ על Q באופן הבא:

$$p \sim_i q \quad - \quad p, q \in Q, p \sim_i q \Leftrightarrow \{ \forall |w| \leq i, \delta^*(p, w) \in F \Leftrightarrow \delta^*(q, w) \in F \}$$

$$\Leftrightarrow \text{אין סיפא מפרידה באורך } i \geq \text{ בין } p \text{ ל } q.$$

$$Q / \equiv_0 = \{F, Q \setminus F\}, i = 0 \quad \text{עבור}$$

עבור $i > 0$ לכל $p \equiv_{i-1} q$, בדוק לכל $a \in \Sigma$ אם $\delta(p, a) \equiv_{i-1} \delta(q, a)$. אם כן, $p \equiv_i q$ אחרת $p \not\equiv_i q$.

אם $Q \equiv_i Q_{i-1}$ עצור.

טענות:

1. לכל $i \geq 0$ $p \equiv_i q \Leftrightarrow \delta^*(p, w) \in F \Leftrightarrow \delta^*(q, w) \in F \Leftrightarrow \forall |w| \leq i, \delta^*(p, w) \in F \Leftrightarrow \delta^*(q, w) \in F$
2. האלגוריתם עוצר
3. אם עצרנו ב $i = u$ אז לכל $j > u$ מתקיים $p \equiv_j q \Leftrightarrow p \equiv_u q$
4. נבנה אוטומט ונוכיח מינימליות.

הוכחה:

1. הוכחה: באינדוקציה i

a. עבור $i = 0$ - החלוקה שלנו היא למצב מקבלים וללא מקבלים, ולכן זה תקין.

b. עבור $i > 0$ -

$$p \equiv_i q \Leftrightarrow \forall a \in \Sigma, [\delta(p, a) \equiv_{i-1} \delta(q, a)] \Leftrightarrow$$

$$\forall a \in \Sigma [\forall |w| \leq i-1, \delta^*(\delta(q, a), w) \in F \Leftrightarrow \delta^*(\delta(q, a), w) \in F] \Leftrightarrow$$

$$\forall a \in \Sigma [\forall |w| \leq i-1, \delta^*(p, aw) \in F \Leftrightarrow \delta^*(q, aw) \in F] \Leftrightarrow$$

$$\forall |z| \leq i, \delta^*(p, z) \in F \Leftrightarrow \delta^*(q, z) \in F$$

2. היחס $\equiv_{i+1}$ מעדן את היחס $\equiv_i$. בכל שלב (פרט לאחרון) באלגוריתם לפחות מחלקה אחת מתפצלת. מספר הפיצולים חסום ע"י $|Q|$ ועל כן מספר האיטרציות של האלגוריתם $|Q| \geq$

$$O(|Q|(|Q|^2|\Sigma|)) = O(|Q|^3|\Sigma|) \text{ זמן ריצה:}$$

3. **טענה:** אם עצרנו ב $i = u$ אז לכל $j > u$ מתקיים $p \equiv_j q \Leftrightarrow p \equiv_u q$

הוכחה: באינדוקציה על $j = u - j$ - ברור.

$$p \equiv_j u \Leftrightarrow \begin{cases} p \equiv_{j-1} q \\ \forall a \in \Sigma, \delta(p, a) \equiv_{j-1} \delta(q, a) \end{cases} \Leftrightarrow_{\text{induction hypothesis}}$$

$$\begin{cases} p \equiv_k q \\ \forall a \in \Sigma, \delta(p, a) \equiv_k \delta(q, a) \end{cases} \Leftrightarrow_{\substack{\text{Stopped after } k \\ (\equiv_k) = (\equiv_{k-1})}} \text{ **צעד:**}$$

$$(p \equiv_{k-1} q) \wedge (\forall a \in \Sigma, \delta(p, a) \equiv_{k-1} \delta(q, a)) \Leftrightarrow p \equiv_k q$$

$$\underline{p \equiv_k q \Leftrightarrow p \sim_\infty q \Leftrightarrow p \sim_F q} \text{ **ולכן**}$$

4. נגדיר $A' = \langle \frac{Q}{\sim_\infty}, \Sigma, \delta', [q_0]_{\sim_\infty}, \frac{F}{\sim_\infty} \rangle$ נגדיר $\delta'([q], a) = [\delta(q, a)]$

הוכחת מינימליות:

טענה: לכל $x, y \in \Sigma^*$ מתקיים $x \sim_L y \Leftrightarrow \delta^*(q_0, x) \equiv_\infty \delta^*(q_0, y)$

$$x \sim_L y \Leftrightarrow (\forall z \in \Sigma^*, xz \in L \Leftrightarrow yz \in L) \Leftrightarrow$$

$$\forall z \in \Sigma^*, \delta^*(q_0, xz) \in F \Leftrightarrow \delta^*(q_0, yz) \in F \Leftrightarrow$$

$$\forall z \in \Sigma^*, \delta^*(\delta^*(q_0, x), z) \in F \Leftrightarrow \delta^*(\delta^*(q_0, y), z) \in F \Leftrightarrow \text{ **הוכחה:**}$$

$$\delta^*(q_0, x) \sim_\infty \delta^*(q_0, y)$$

שימושים במשפט MN

a. לכל $0^i 1$, $L_1 = \{ww \mid w \in \Sigma^*\}$, לכל $i > 0$, עבור $i \neq j$, $0^i 1$ היא סיפא מפרידה בין $0^i 1, 0^j 1$.

b. $L_2 = \{0^i 1^j \mid \gcd(i, j) = 1\}$ לכל שני מספרים ראשוניים $p_1 \neq p_2$, 1^{p_1} סיפא מפרידה בין $0^{p_1}, 0^{p_2}$.

שאלות על אוטומטים

a. שייכות של מילה לשפת אוטומט - $w \in L(A)$

i. עבור DFA - פשוט מריצים - זמן ריצה $O(|W|)$

ii. עבור NFA

1. אופציה נאיבית – בונים DFA שקול. - $O(2^{|Q|} + w)$

2. שומר בכל רגע נתון את כל המצבים שאני יכול להיות בהם. $O(|W| \cdot |Q|^2)$

3. $\{w\} \cap L(A) \neq \emptyset \Leftrightarrow w \in L(A)$ - בניית אוטומט כזה ובדיקה האם

הוא ריק מתבצעת ב $O(|Q||w|)$

b. ריקנות - $L(A) = ? \emptyset$

i. בעזרת DFS (Depth first search) - $O(|Q|)$ - בודקים האם יש מסלול מהתחלה

עד

c. אוניברסליות $L(A) = ? \Sigma^*$

i. בודקים ריקנות של המשלים. אפשרי רק ב DFA. ב NFA נהפוך אותו קודם ל DFA.

שפות חסרות הקשר

- דקדוק חסר הקשר (ח"ה) בצורה נורמלית של חומסקי
- שייכות של מילה לדקדוק
- ריקנות של דקדוק

$$G = \left\langle \underbrace{\Sigma}_{\text{Alphabet}}, \underbrace{V}_{\text{Variables}}, \underbrace{S}_{\text{Starting Variable}}, \underbrace{R}_{\text{Derivation laws}} \right\rangle$$

נסמן דקדוק כך:

הגדרה: דקדוק ח"ה בצורה נורמלית של חומסקי אם כל חוקי הגזירה שלו הם מאחת הצורות הבאות:

- $A \rightarrow BC$, כך ש $A, B, C \in V, B, C \neq S$
- $A \rightarrow a$, $A \in V, a \in \Sigma$
- $S \rightarrow \varepsilon$

טענה: לכל דקדוק ח"ה G יש דקדוק ח"ה G' בצורה נורמלית של חומסקי כך ש $L(G) = L(G')$

הוכחה:

נטפל בכללים מהצורה:

1. $A \rightarrow \varepsilon$
2. $A \rightarrow B$
3. $A \rightarrow BCD \dots$ (יותר מ-2 משתנים)
4. $A \rightarrow bC$

ראשית, נגדיר משתנה תחילי חדש S' , ונוסיף חוק $S' \rightarrow S$ (כדי שלא נגזור כלום לתוך S).
כעת,

1. לכל $A \rightarrow \varepsilon, S \neq A$, עבור כל חוק מצורה $B \rightarrow uAv$ ($u, v \in (\Sigma \cup V)^*$) נוסיף את

החוק $B \rightarrow uv$.

הערה: עבור $B \rightarrow uAvAw$ נוסיף $B \rightarrow uvw | uAvw | uvAw$

עבור $B \rightarrow A$ - אם כבר טיפלנו ב $B \rightarrow \varepsilon$ - לא נוסיף אותו.

2. לכל $A \rightarrow B$ - עבור כל חוק $B \rightarrow u$ ($u \in (\Sigma \cup V)^*$) נוסיף את $A \rightarrow u$ (אלא אם כן טיפלנו בו קודם).

3. לכל $A \rightarrow u_1 u_2 \dots u_k$, $k \geq 3$, כך ש $u_i \in (E \cup V)$ (נשים לב - ללא כוכב).

נחליף אותו לרשימת הכללים הבאה: $A \rightarrow u_1 A_1$, $A \rightarrow u_2 A_2$, $A_1 \rightarrow u_{k-1} u_k$, $A_{k-2} \rightarrow u_{k-1} u_k$, ..., כאשר $A_1 \dots A_{k-2}$ משתנים חדשים.

4. לכל $A \rightarrow u_1 u_2$ $u_i \in (V \cup \Sigma)$ - אם $u_i \in \Sigma$ נגדיר משתנה חדש U_1 ונוסיף את

החוקים $A \rightarrow U_1 u_2$, $U_1 \rightarrow u_1$, וכן עבור u_2 .

בעיית השייכות

בהינתן דקדוק G , האם $w \in L(G)$?

טענה: אם קיימת גזירה $w \Rightarrow S$ אז קיימת גזירה מהצורה $(|w| = n)$
 $S \rightarrow A_1 A_2 \rightarrow B_1 B_2 B_3 \rightarrow \dots \rightarrow \underbrace{Z_1}_{w_1} \underbrace{Z_2}_{w_2} \dots \underbrace{Z_n}_{w_n}$
 פשוט נבדוק את כל המקרים.

אופציה טובה יותר:
 (משתמש בתכנון דינאמי)
 יהו G, w .

נגדיר טבלה המוגדרת כך
 $T[i, j] =$ כל המשתנים שמהם ניתן לגזור את תת המילה $w_i \dots w_j$, כלומר -
 $\{A \in V \mid A \Rightarrow w_i \dots w_j\}$. ברור שאנו מתעניינים רק במקרים בהם $j \geq i$.
 נמלא קודם כל את האלכסון $T[i, i]$ (כל i), ומשם נמשיך.

אלגוריתם:
אתחול: לכל $i \in 1 \dots n$ $T[i, i] = \{A \in V \mid A \rightarrow w_i\}$
 לכל $k = 1, \dots, n-1$ (אורך התת מילה פחות 1)
 לכל $i = 1, \dots, n-k$ ($i =$ מיקום התת מילה שאנו מפצלים)
 לכל $j = 0, \dots, k-1$ (j - אורך החלק הראשון פחות אחד)
 אם יש חוק מהצורה $A \rightarrow BC$ כאשר $B \in T[i, i+j]$
 ו $C \in T[i+j+1, i+k]$, הוסף את A ל $T[i, i+k]$.

סדר גודל זמן ריצה - $n^3 |R|$.

ריקנות של דקדוק
 נרצה לשאול - $L(G) = \emptyset$?
 נשים לב כי אם יש גזירה $w \Rightarrow S$ עבור w כלשהיא, אז קיימת מילה כלשהי עם עץ גזירה בעומק $\geq |V| + 2$ (הרי כל משתנה נהפך מתישהו לטרמינל, אז אם משתנה נגזר לאחר, אז אפשר להגיע ישר לטרמינל).
 על כן, פשוט ננסה את כל הגזירות האפשרויות עד עומק $|V| + 2$, ונראה אם הגענו למילה כלשהיא.

אופציה טובה יותר: נבנה קבוצה E (Effective) של משתנים שמהם ניתן לגזור מילים.
 נניח ש G בצורת חומסקי.
 נגדיר:

$$E_1 = \{A \in V \mid A \rightarrow a, a \in \Sigma\}$$

$$E_{i+1} = E_i \cup \{A \in V \mid A \rightarrow BC, B, C \in E_i\}$$

טענה: $E_i =$ כל המשתנים שמהם ניתן לגזור מילה תוך i צעדים (עץ גזירה בעומק i).

אלגוריתם
 אם $E_i = E_{i+1}$ - עצור - אם $S \notin E_i$, אי אפשר לגזור מילה. אם בשלב כלשהו $S \in E_i$, מסקנה שאפשר לגזור מילה, וכך אנו עונים על שאלת הריקנות.

זמן ריצה: מספר האיטרציות יהיה לכל היותר כמספר המשתנים – ולכן $|V||R|$.

בעיית האוניברסליות

האם $L(G) = \Sigma^*$ - לא ידוע אלגוריתם שפותר את זה. (הוכח שאין כזה).

נזכור כי $REG \subseteq CFL$

בהמשך נוכיח כי אוטומט מחסנית שקול ל-CFL. עבור כל ביטוי רגולרי יש לנו דקדוק שגוזר את השפה:

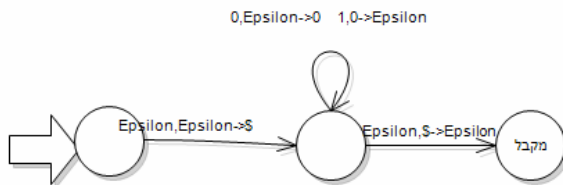
$S \rightarrow S$	$\emptyset$
$S \rightarrow \varepsilon$	ε
$S \rightarrow a$	$a \in \Sigma$
$S_{r_1+r_2} \rightarrow S_{r_1} S_{r_2}$	$r_1 + r_2$
$S_{r_1 r_2} \rightarrow S_{r_1} S_{r_2}$	$r_1 r_2$
$S_{r_1}^* \rightarrow S_{r_1} S_{r_1}^* \varepsilon$	r_1^*

דוגמאות

נכתוב אוטומט מחסנית עבור שפת הסוגריים המכוננים חוקית. פורמלית

$\Sigma = \{0,1\}$. (0 – סוגריים שמאלי, 1 סוגריים ימני)

$$L = \{w \in \{0,1\}^* \mid \#0 \geq \#1 \text{ for every beginning, and overall } \#0 = \#1\}$$



30.5.2007

מכונות טיורינג

על סדר היום

- שקילות בין מכונת טיורינג עם k סרטים למכונה עם סרט יחיד.
- שקילות בין מ"ט לא דרטמניסטית למכונה דרטמניסטית.

עד כה ראינו כי $REG=DFA=NFA$

ראינו כי $CFL=PDA$.

כעת אנו עולים רמה, ומגיעים לTM (Turing Machine).

ראינו כי $0^n 1^n \in CFL$ אבל אינה רגולרית, אבל $0^n 1^n 2^n \in TM$ אבל לא ל CFL .

תזכורת:

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

ראינו כי $\Sigma \subset \Gamma$, וכי $_ \in \Gamma \setminus \Sigma$.

קונבנציה: הסרט מסתיים ב- (סיום הקלט הוא ב-)

נגדיר קונפיגורציה כ:

1. מצב המכונה
2. תוכן הסרט
3. מיקום הראש

נרשום כך: $c = w_1 q_{17} w_2 w_3 \dots w_4$

	q_{17}				
	↓				
w_1	w_2	...	w_4	—	

לדוגמא: את

פונקציית המעברים מוגדרת כך:

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

בהמשך לדוגמא, אם $\delta(q_{17}, w_2) = \langle q_5, t_L \rangle$, אז המעבר יהיה כזה:

	q_{17}				
	↓				
w_1	w_2	...	w_4	—	

 $\Rightarrow$

	q_5				
	↓				
w_1	t	...	w_4	—	

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

מה זה בדיוק: הקלט נתון בסרט הראשון. יש k ראשים. הפונקציה היא

$$\delta: Q \times \Gamma^k \rightarrow Q \times (\Gamma \times \{L, R\})^k$$

הוכחה:

הרעיון היה לרשום על הסרט היחיד שלנו את כל k הסרטים. נניח כי $\# \notin \Gamma$, ונוסיף אותו – והוא יהיה הסימן שמפריד בין הסרטים.

נגדיר את $\dot{\Gamma}' = \Gamma \cup \{\#\} \cup \dot{\Gamma}$ - לכל אות, אפשר לשים אותה עם נקודה למעלה – מה שמסמן, שהראש של הסרט הרלוונטי הוא עליה).

נגדיר את $Q' = \underbrace{\{q_{right}, q_{left}\}}_{\text{moving left or right}} \times Q \cup \underbrace{\Gamma^k}_{\text{letter at all heads}} \times \underbrace{\{0,1\}^k}_{\substack{\text{Is the head} \\ \text{at the rightmost position} \\ \text{for each tape? If true,} \\ \text{should shift right}}} \times \underbrace{\{L,R\}^k}_{\substack{\text{Where the heads} \\ \text{Should move?}}}$ (כך נוכל לשמור

את המידע באיזה מצב אנו בכל אחד ב k הסרטים, בתור המצב שלנו). נשים לב שבהגדרת Q' אנו שומרים את כל המידע הדרוש. עלינו להראות:

1. מעבר לקונפיגורציה תחילית (ברור)
2. מעברים (איך לעבור סרט לסרט) (מתואר לעיל).
3. מקבלים / דוחים – כמו קודם.

סיבוכיות

נניח שמ"ט עם k סרטים עוצרת על קלט w תוך T צעדים. תוך כמה צעדים תעצור על w מכונה עם סרט יחיד?

תשובה: מספר הצעדים המקורי $(T) \times$ מספר הצעדים שנדרש לסמלץ צעד מקורי יחיד.

כמה צעדים נדרש לסמלץ צעד מקורי יחיד: אורך כל סרט מקורי הוא לכל היותר T (שכן לא היינו יכולים לעבור T מקומות ב T צעדים). לכן, אורך הסרט המאוחד הוא $k \cdot T + n$ (כאשר n הוא הקלט ההתחלתי, במקרים בהם T הוא קטן זה משפיע, בד"כ n זניח). לכן, בכל מקרה, מספר הצעדים הנדרש הוא $O(k \cdot T)$ לכל צעד מקורי הוא.

לכן, בסה"כ, המעבר מ k סרטים לאחד הוא $O(k \cdot T^2)$.

מ"ט לא דטרמניסטי

ניזכר – באוטומט דטרמניסטי $\delta: Q \times \Sigma \rightarrow Q$, בלא דטרמנסטי זה $\delta: Q \times E \rightarrow 2^Q$.

באופן דומה, במ"ט דטרמניסטי $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$ ובמ"ט לא דטרמניסטי, $\delta: Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L,R\}}$.

לדוגמא - $\delta(q_0, a) = \{\langle q_5, b, R \rangle, \langle q_{17}, w, R \rangle\}$

במ"ט לא דטרמניסטי, $w \in L(M)$ אם קיימת ריצה מקבלת של M על w .

טענה: לכל מ"ט לא דטרמניסטי קיימת מ"ט דטרמניסטי שקולה.

הוכחה: נגדיר מכונה עם 3 סרטים:

- סרט קלט
- סרט עבודה
- סרט ניחוש

נשים לב שמ"ט לא דטרמניסטי מגדירה עץ (ניחשים). אפשר לעבור על עץ כ DFS או BFS. מכיוון שיתכן ענף אינסופי, DFS לא מתאים, אז נעבוד ב BFS.

בכל התחלה של סימולציה, נעתיק את סרט הקלט לסרט העבודה, ונבחר ריצה ע"פ סרט הניחוש.

נניח שמספר הבחירות המקסימלי בפונקציית המעברים הוא b (b for branching).

לדוגמא, אם נרשום בתא הראשון של סרט הניחוש 1, הדבר מסמל שבחרתי ב branch הראשון שהגעתי אליו לאופציה הראשונה. אם לא הגעתי למצב מקבל, אני אעבור לאופציה השניה (נזכור כי עובדים

ב BFS), רושם 2, וכך הלאה, עד b .

כשהגעתי לב, עובר לניחוש הבא, וכך הלאה... נמשיך עד שאני מגיע למצב מקבל, או עד הסוף (או עד האינסוף).

הוכחנו כי מכונה דטרמניסטי עם 3 סרטים שקולה למכונה עם סרט אחד, ולכן סיימנו ©.

זמן ריצה: נניח שיש למ"ט לא דטרמניסטי ריצה מינימלית (באורכה) שמקבלת את w תוך T צעדים. תוך כמה צעדים w תתקבל במכונה הדטרמניסטיה השקולה?

עץ הניחוש הוא בעומק T , ולכן יש בו b^T עלים. לא ידוע פתרון טוב יותר.

הגדרה: מ"ט היא מכונה מכריעה אם היא עוצרת על כל קלט.

מ"ט היא מכונה מכריעה אם לכל קלט כל הריצות עוצרות.

נרצה שאם הפכנו מכונה לא דטרמניסטית לדטרמניסטית, היא תישער מכריעה. נוסיף סרט המשך. מה הוא יעשה? כל פעם שיוורדים רמה, הוא יתאפס. אם הגענו למצב לא דוחה באחד המצבים, נדליק אותו. אם בסיום הרמה הוא דלק, אז לא ממשיכים (בעצם, אם כל המצבים באותה רמה דוחים, אז גם המכונה הדטרמניסטית תדחה).

6.6.2007

על סדר היום:

- רדוקציות
- הגדרה
- השפה $Regular \notin R \cup co-RE$
- משפט Rice
- $INF_{TM} \notin RE$ (אם נספיק)

תזכורת:

R – שפות ניתנות להכרעה ע"י מכונת טיורינג
RE – שפות הניתנות לזיהוי ע"י מכונת טיורינג

רדוקציה

אינטואיציה: נאמר ש $L_1 \leq L_2$ אם קיימת פונקציה ניתנת לחישוב f , אשר מקיימת

$$\forall x, f(x) \in L_2 \Leftrightarrow x \in L_1.$$

פורמלית: נאמר ש L_1 ניתנת לרדוקציה (רדוקציית מיפוי) ל L_2 , ונסמן $L_1 \leq_M L_2$ אם קיימת פונקציה

$$חשיבה^1 \quad f: \Sigma^* \rightarrow \Sigma^* \quad \text{כך ש} \quad f(x) \in L_2 \Leftrightarrow x \in L_1.$$

מהי פונקציה חשיבה: קיימת מ"ט שלכל x מחשבת את $f(x)$ ועוצרת.

טענה: אם $L_1 \leq_M L_2$, אזי L_2 יותר קשה מ L_1 .

דוגמא: אם $L_2 \in R, L_1 \leq_M L_2 \Rightarrow L_1 \in R$.

טענה: אם $L_1 \notin RE, L_1 \leq_M L_2 \Rightarrow L_2 \notin RE$.

הוכחה: נניח בשלילה ש $L_2 \in RE$, עם מכונה M שמכריעה. אבל יש פונקציה ניתנת לחישוב f

המקיימת $\forall x, f(x) \in L_2 \Leftrightarrow x \in L_1$, ואז אפשר להשתמש בM כדי להכריע את L_1 , בסתירה לכך

ש $L_1 \notin RE$.

ניזכר בשפה $Regular = \{\langle M \rangle \mid L(M) \in REG\}$

טענה: $Reg \notin RE \cup co-RE$.

הוכחה:

ראשית נראה כי (1) $Reg \notin RE$, ולאחר מכן נראה כי (2) $\overline{REG} \notin RE$.

תזכורת:

$$A_{TM} = \{\langle M \rangle, w \mid \langle M \rangle \text{ is turing machine} \wedge w \in L(M)\}$$

$$\overline{A_{TM}} = \{\langle M \rangle, w \mid w \in L(M)\}$$

בכיתה הוכחנו כי $A_{TM} \in RE \setminus R$, ומאחר שהוכחנו $L \in R \Leftrightarrow L \in RE \wedge \overline{L} \in RE$,

נובע כי $\overline{A_{TM}} \notin RE$.

(1) נראה כי $\overline{A_{TM}} \leq Reg$. נתאר את הפונקציה f . בהינתן $\langle M \rangle, w$, תחזיר את המכונה

M_1 הבאה:

¹ ניתנת לחישוב

$\langle M \rangle, w \in \overline{A_{TM}} \Leftrightarrow f(\langle M \rangle, w) = \langle M_1 \rangle \in REG$
 כלומר - $w \notin L(M) \Leftrightarrow L(M_1) \in REG$.

איך תעבוד $M_1(\langle M \rangle, w)$ על קלט z ?

- אם M מקבלת את w
- אם $z \in 0^n 1^n$
- כן - קבל
- לא - דחה
- אם M לא מקבלת את w (ועוצרת), דחה.

כעת, נניח ש $w \notin L(M)$. אם היא דוחה, אז M_1 תדחה (לכל קלט z). אם היא מתבדרת, גם M_1 מתבדר על כל קלט $\Leftarrow M$ לא מקבלת את $w \Leftarrow M_1$ לא מקבלת אף $z \Leftarrow L(M_1) = \emptyset \Leftarrow L(M_1) \in REG$.

אם $w \in L(M) \Leftarrow M_1$ מקבלת את z אם $z \in 0^n 1^n$, כלומר $L(M_1) = 0^n 1^n \Leftarrow L(M_1) \notin REG$.

מה הראנו? כי $\langle M \rangle, w \in \overline{A_{TM}} \Leftrightarrow f(\langle M \rangle, w) = \langle M_1 \rangle \in REG$ כנדרש ☺

מה נעשה אם $\langle M \rangle, w$ הוא לא יצוג של מכונת טיורינג (syntax error)? לדוגמא, אם נקבל (0^{15}) ? מחזיר אוטומט רגולרי כלשהו.

כעת, נוכיח כי $Regular \notin co-RE$, כלומר $REGULAR \notin RE$. נעשה רדוקציה -

$\overline{A_{TM}} \leq REGULAR$. נראה פונקציה שבהנתן $\langle M \rangle, w$ מחזירה $\langle M_2 \rangle$ כך ש $\langle M, w \rangle \in \overline{A_{TM}} \Leftrightarrow \langle M_2 \rangle \in REGULAR$.

מה תעשה $M_2(\langle M \rangle, w)$ על קלט z ?

- אם $z \in 0^n 1^n$ - קבל
- אחרת
- הרץ את M על w
- החזר את תוצאת ההרצה

בואו ונבדוק:

אם $\langle M \rangle, w \in \overline{A_{TM}}$, כלומר $w \notin L(M)$, ואז M_2 מקבלת את z אם

$z \in 0^n 1^n \Leftarrow L(M_2) = 0^n 1^n \Leftarrow \langle M_2 \rangle \in REGULAR$.

אם $\langle M \rangle, w \in \overline{A_{TM}}$, אזי $w \in L(M) \Leftarrow M_2$ מקבלת כל z , ואז $\Sigma^* \Leftarrow L(M_2) = \Sigma^* \Leftarrow \langle M_2 \rangle \notin REGULAR$ רגולרית.

משפט RICE: כל תכונה סמנטית לא טריוויאלית של מכונות טיורינג איננה כריעה בעברית:

תכונה: קבוצה $P = \{M\}$ לכל מכונה אפשר לשאול $M \in P$.

סמנטית: תלויה אך ורק בשפה. לדוגמא - $L(M_1) = L(M_2) \rightarrow M_1 \in P \Leftrightarrow M_2 \in P$ (לא תלויה בקידוד, אלא בשפה בלבד).

לא טריוויאלית: קיימת מכונה $M \in P$ וקיימת $M' \notin P$ (אינה ריקה ולא כל המכונות).

כלומר, מהמשפט נובע כי $L_P = \{\langle M \rangle \mid M \in P\} \notin R$ לכל P .

הוכחה:

תהי P תכונה סמנטית לא טריוויאלית של מ"ט. נניח בה"כ כי $M_\emptyset \in P$ ($L(M_\emptyset) = \emptyset$) – כלומר, כל המכונות ששפתם ריקה שייכות ל- P . נוכיח ש $L_P \notin RE$.

הערה: $L \in R \Leftrightarrow L \in RE \wedge \bar{L} \in RE$. מאחר ו $M_\emptyset$ שייך ל- P או $\bar{P}$, נראה על הצד

ש $M_\emptyset$ שייך אליו (ואת זה מותר בה"כ), ומאחר ונוכיח שהוא לא שייך ל- RE , אזי $L \notin R$.

כיוון ש P לא טריוויאלית, קיימת מ"ט $M' \notin P$ כך ש $M' \notin P$. נראה רדוקציה מ $\bar{A}_{TM} \leq L_P$. תעבוד

כך – בהינתן $\langle M \rangle, w$ נחזיר M_3 כך ש $\langle M_3 \rangle \in L_P \Leftrightarrow \langle M \rangle, w \in \bar{A}_{TM}$.

נבנה אותה כך:

על קלט z :

- אם M דוחה את w , דחה
- אחרת

○ הפעל את M' על z , והחזר את התוצאה.

אם $\langle M \rangle, w \in \bar{A}_{TM}$, אזי M לא מקבלת את w , ואז $L(M_3) = \emptyset$ ו $M_3 \in P \Leftrightarrow \langle M_3 \rangle \in L_P$.

אם $\langle M \rangle, w \notin \bar{A}_{TM}$, אזי M מקבלת את w , ומהבניה $L(M) = L(M')$ ו $M_3 \notin P \Leftrightarrow \langle M_3 \rangle \notin L_P$. הראינו רדוקציה כנדרש.

(הצדקה נוספת לבה"כ:)

אם $M_\emptyset \in P$, אזי $L_P \notin RE$. אחרת, $M_\emptyset \notin P$, ואז $\bar{L}_P \notin RE$, ובכל מקרה - $L_P \notin R$.

עוד רדוקציה אחת והולכים:

$INF_{TM} = \{\langle M \rangle \mid L(M) \text{ is infinite}\}$

טענה: $INF_{TM} \notin RE$.

הוכחה: נראה כי $\bar{A}_{TM} \leq INF_{TM}$. בהינתן $\langle M \rangle, w$ הרדוקציה מחזירה $\langle M_4 \rangle$ כך ש

$w \in L(M) \Leftrightarrow |L(M_4)| = \infty$

$M_4(\langle M \rangle, w)$ תעבוד כך על קלט z :

- נריץ את M על w $|z|$ צעדים.
- אם התקבל:

○ נחזיר Reject

- אחרת (דחה, או לא סיים להריץ)

○ החזר accept

אם $M \Leftarrow \langle M \rangle, w \in \overline{A_{TM}}$ לא מקבלת את w , אזי M_4 תחזיר לכל קלט 'accept', כלומר

$$L(M_4) = \Sigma^* \Rightarrow |L(M_4)| = \infty$$

אם $\langle M \rangle, w \notin \overline{A_{TM}}$, אזי M מקבלת את w אחרי k צעדים, אזי M_4 תחזיר *Accept* למספר סופי של מילים (עד אורך k).

הראנו רדוקציה כנדרש.

13.6.2007

על מה נדבר היום?

- P, NP , רדוקציות פולינומיות
- בעיית הריצוף החסום
- שפות שקשורות לדקדוקים חסרי הקשר.

במה הקורס עוסק:

- אוטומטים (+ מחסניות) – REG, CFL
- חישוביות – $RE, co-RE, R$. מה ניתן לחשב?
- סיבוכיות – מתעסק ב R – כמה זה עולה לנו
 - זמן
 - זכרון

הגדרנו

$$\{ \text{השפות שעבורן קיימת מ"ט דטרמיניסטית שמכריעה קלט באורך } n \text{ תוך } t(n) \} = TIME(t(n))$$

$$\{ \text{צעדים } O(t(n)) \}$$

$$\{ \text{השפות שעבורן קיימת מ"ט לא דטרמיניסטית שמכריעה קלט באורך } n \text{ תוך } t(n) \} = NTIME(t(n))$$

$$\{ \text{צעדים } O(t(n)) \}$$

$$P = \bigcup_{\mathbb{N} \ni k > 0} TIME(n^k)$$

נגדיר:

$$NP = \bigcup_{\mathbb{N} \ni k > 0} NTIME(n^k)$$

ברור כי $NP \subseteq P$. השאלה שעדיין לא הוכרעה היא אם הן שוות $NP = P$

רדוקציות פולינומיות

שפה L_1 ניתנת לרדוקציה פולינומית לשפה L_2 , ונסמן $L_1 \leq_p L_2$ אם קיימת פונקציה חשיבה בזמן

פולינומי $f: \Sigma^* \rightarrow \Sigma^*$ כך שלכל $x \in \Sigma^*$ מתקיים $x \in L_1 \Leftrightarrow f(x) \in L_2$.

חשיבה בזמן פולינומי = קיימת מ"ט דטרמיניסטית שרצה בזמן פולינומיאלי ומחשבת את f .

ברור כי אם $L_1 \leq_p L_2$ אזי $L_1 \in P \Rightarrow L_2 \in P$.

השאלה - $NP \subseteq P$

NP Complete

שפה L היא NP-שלמה, ונסמנה NPC אם:

1. $L \in NP$

2. L NP-קשה: לכל $L' \in NP$ אזי $L' \leq_p L$

טענה: אם L היא NP-שלמה, ו $L \in P$ אז $NP \subseteq P$, ואז $P = NP$.

נראה שפה NPC

נגדיר

$BT = \{ \langle T, H, V, t_0, t_k, 1^s \rangle \mid * \}$ - קיים ריצוף חוקי עבור הריבוע $s \times s$ כך שבמיקום $(1,1)$ נמצא t_0 , ובמקום $(1,s)$ נמצא t_k .

$$T = \{t_0, t_1, \dots, t_k\}$$

$$H, V \subseteq T \times T$$

H, V מסמנים את הזוגות שמותר שיהיו צמודים אופקית ואנכית.

נשים לב שזה דומה ל $TILE = \{ \langle T, H, V, t_0 \rangle \mid * \}$ - קיים ריצוף חוקי עבור הריבוע $n \times n$ לכל n כך שבמקום $(1,1)$ נמצא t_0 . הוכחנו כי $TILE \notin RE$.

מבנה ההוכחה – נראה $\overline{HALT_{TM}^\epsilon} \leq TILE$

בהינתן קלט $\langle m \rangle$ נבנה מופע של בעיית ריצוף I , כך ש $\langle M \rangle \in \overline{HALT_{TM}^\epsilon} \Leftrightarrow I \in TILE$.

איך נבנה את הריצוף:

(q_1, a) * (q_1, L) a	b (q_1, L) * $(q_5, -)$			
a (q_5, R) $(q_0, -)$	$(q_5, -)$ (q_5, R) * —	— * — *	— * — *	...
$(q_0, -)$ * — *	— * — *	— * — *	...	

נשים לב, שכל שורה מייצגת קונפיגורציה (c_i) . כאשר $c_i \in \Gamma^* \cdot (Q \times \Gamma) \cdot \Gamma^*$, לדוגמא

$$c_{17} = abba \langle q_7, d \rangle ccda$$

בשרטוט לעיל, $\delta(q_5, -) = \langle q_1, b, L \rangle$ ו $\delta(q_0, -) = \langle q_5, a, R \rangle$.

רשימת מרצפות:

$$t_0 = \begin{array}{|c|} \hline - \\ \hline - \quad * \\ \hline \end{array} \quad . \quad \begin{array}{|c|} \hline (q_0, -) \\ \hline * \quad - \\ \hline * \\ \hline \end{array} \quad .1$$

2. לכל מעבר אשר מסיט את הראש ימינה, כלומר $\delta(q, a) = \langle q', b, R \rangle$ נרצף בצורה

$$\begin{array}{|c|} \hline (q', c) \\ \hline (q', R) * \\ \hline c \\ \hline \end{array} \quad \text{לכל } c \in \Gamma \text{ נוסיף את המרצפת} \quad \begin{array}{|c|} \hline b \\ \hline (q', R) * \\ \hline (q, a) \\ \hline \end{array}$$

3. באופן דומה לכל מעבר אשר מסיט את הראש שמאלה.

$$\begin{array}{|c|} \hline a \\ \hline * * \\ \hline a \\ \hline \end{array} \quad \text{לכל } a \in \Gamma \text{ (מילוי), נמצא}$$

נשים לב, שמאחר ו δ לא מוגדר עבור q_{acc}, q_{rej} , אזי אם אפשר לרצף את C_i , אז מגיעים לקונפיגורציה זו, כלומר אפשר לרצף עד אינסוף $\Leftrightarrow$ המכונה לא עוצרת.

נחזור לבעיית הריצוף BT -

$$BT = \{ \langle T, H, V, t_0, t_k, 1^s \rangle | * \} \quad \text{קיים ריצוף חוקי עבור הריבוע } s \times s \text{ כך שבמיקום } (1,1) \\ \text{נמצא } t_0, \text{ ובמקום } (1, s) \text{ נמצא } t_k.$$

$$T = \{t_0, t_1, \dots, t_k\}$$

$$H, V \subseteq T \times T$$

נשים לב – הקלט באונרית - 1^s אני רושם ע"י s אחדות.

טענה: BT היא NP שלמה.

הוכחה:

1. נראה כי $BT \in NP$ - כלומר קיימת מ"ט לא דטרמניסטית שמכריעה את BT בזמן פולינומיאלי.

איך נעשה זאת? המכונה תנחש ריצוף אקראי, ותבדוק אם הוא חוקי. היא תקבל / תדחה בהתאם. זמן ריצה - $O(s^2)$. גודל הקלט הוא לפחות s (הקלט הוא באונרית, כזכור).

2. BT היא NP שלמה. נראה שלכל $L \in NP$ מתקיים $L \leq_p BT$. איך נעשה זאת? נראה

$$L \leq_p A_{TM}^{bounded} \leq_p BT$$

a. מה זה $A_{TM}^{bounded}$?

$$A_{TM}^{bounded} = \left\{ \langle M, w, p_{olinom} \rangle \mid M \text{ is non deterministic MT which } \right. \\ \left. \text{accepts } w \text{ at } O(p(|w|)) \right\}$$

ראשית, נראה $L \leq_p A_{TM}^{bounded}$. זה ממש ברור – נתון לנו הפולינום שחוסם את L , אז נחשב

אותו על $|w|$ ואת התוצאה נשמור בצורה אונרית, ואז נריץ M על w עם חסימה של $p(|w|)$ צעדים. ברור כי זוהי רדוקציה פולינומית. חשוב להבין – לאחר הרדוקציה, $p(|w|)$

הוא פשוט $1^{p(|w|)}$ - מספר.

כעת, נראה את הרדוקציה $BT \leq_p A_{TM}^{bounded}$.

בהינתן $\langle M, w, 1^s \rangle$ נבנה כעת ריצוף $BI = \langle T, G, V, t_i, t_k, 1^{s'} \rangle$ כך ש M מקבלת את w תוך s צעדים אם קיים ריצוף חוקי עבור הריבוע $s' \times s'$ ($s' := 2s$).
נעשה משהו דומה למה שעשינו ב $TILE$:

(q_1, a) $*$ (q_1, L) $*$ a	b (q_1, L) $*$ $(q_5, -)$			
a (q_5, R) $*$ $(q_0, -)$	$(q_5, -)$ $*$ (q_5, R) w_2	$-$ $*$ $-$ $*$ w_3	$-$ $*$ $-$ $*$ w_4	...
(q_0, w_1) $*$ 1 $*$	w_2 1 2 $*$	w_3 2 3 $*$	...	

כלומר – בדיוק כמו קודם, עם הבדלים דקים בשורה הראשונה:

רשימת מרצפות:

$$t_0 = \begin{bmatrix} (q_0, -) \\ * & & 1 \\ & * & \end{bmatrix}, \begin{bmatrix} w_{i+1} \\ i & & i+1 \\ & * & \end{bmatrix}, \dots, 1.$$

$$\begin{bmatrix} - \\ - & & - \\ & * & \end{bmatrix} \text{ מהצורה:}$$

2. מעבר ימינה – לכל $\delta(q, a) = \langle q', b, R \rangle$ נייצר

$$\begin{bmatrix} b \\ * & & (q', R) \\ & (q, a) & \end{bmatrix}, \forall c \in \Gamma, \begin{bmatrix} (q', c) \\ (q', R) & & * \\ & c & \end{bmatrix}$$

3. באופן דומה בשמאל.

לכשנגיע למרצפת של q_{acc} , נעשה אותה בכל אותה שורה (שורה מקבלת שלמה) ו**מעליה** – וכך נעצור.
נשים לב – אנו יכולים לקבל קלטים לא דטרמיניסטים, ונשתמש בחוסר דטרמיניזם שלנו כדי לפתור אותה.

15.9.2007

היום נדבר על:

- מכונת RAM
- ליכסון

מה זה מכונת RAM ?

שרטוט אצל [דינה](#).

הסבר על מחשב – CPU, RAM, פקודות, אסמבלי. קיצר, מחשב. מבנה של פקודה:

הוראה, אופרנד (ערך להוראה)

במכונת ה-RAM שלנו יהיה רק רגיסטר אחד ששמו אקומלטור.

נסמן $c(i)$ את הערך שנמצא בתא i ב-RAM.

פקודות אצלנו:

1. Load – טען מ

a. כתובת אבסולוטית (17)

2. Add -

a. מקבל מספר אמיתי, ומוסיף אותו לאקומלטור – נסמן כ 17 =

b. תוסיף ערך של תא (אבסולוטי) לאקומלטור – נסמן כ 17.

c. תוסיף את הערך של התא הנמצא במיקום $c(i)$ (הוא מצביע) לאקומלטור.

נסמן כ 17*.

Load operand – הכנס את ערך האופרנד לאקומלטור

Store operand – הכנס את ערך האקומלטור לתא שמספרו כערך האופרנד

add operand – הוסף לאקומלטור את ערך האופרנד

sub operand – החסר מהאקומלטור את ערך האופרנד (אם התוצאה שלילית, ערך האקומלטור 0).

jmp operand – שנה את ערך ה-IP (instruction pointer) לערך האופרנד

if=0 jump operand אם ערך האקומלטור 0 שנה את ערך ערך ה-IP לערך האופרנד.

if>0 jump operand כיו"ב

read – הכנס תו אחד של קלט לאקומלטור

accept – קבל את הקלט

reject – דחה את הקלט

אבל מה עם קלט ופלט?

נניח שיש לנו אמצעי קלט (stream).

פלט – נאפשר ביט.

מכונת ה-RAM מתחילה כשכל הזכרון ריק.

נראה שמכונת טיורינג מסוגלת לעשות כל מה שמכונת RAM עושה.

נראה יצוג של קונפיגורציה של מכונת RAM באמצעות קונפיגורציה של מ"ט עם 10 סרטים.

נקדיש סרטים שונים לתפקידים שונים:

1. סרט קלט

2. סרט תוכנית (לא נרשום עליו את המילה 'load', נשתמש ב-yey.enum). אז לאחר הפקודה

יהיה סוג האופרנד, והערך. לאחר כל פקודה, יהיה 'י' מפריד. חשוב להיות מפריד – ערך התא

הוא אינסופי, כי הזכרון אינסופי, ולכן כתובת זכרון היא אינסופית.

איך נסמן את ip – נשים נקודה מעל הפקודה שעומדת להתבצע.

3. סרט האקומלטור – יכיל את ערך האקומלטור

4. סרט הזכרון – מאחר וכל תא מכיל מספר טבעי לא חסום – לכן נפריד בין התאים בצורה הזו - $(3,17)$ - כתובת התא היא 3, הערך הוא 17. נשים לב שזה בעצם 6 תאים בזכרון.
5. סרטי עבודה.

דוגמא של מימוש פקודה:

Add*32

1. רצים על סרט הקפודות, עד שמגיעים לפקודה עם נקודה מעל
2. כותבים $*32$ על סרט עזר
3. רצים על הזכרון ומחפשים ערך של תא 32
4. אם מצאנו מעתיקים את תוכן התא $c(32)$ לסרט עזר
- a. אם לא מצאנו, מעתיקים 0 לסרט העזר
5. בצורה דומה מחפשים את ערך התא שמספרו כתוב על סרט העזר
6. מוסיפים את מה שכתוב בסרט העזר (תוכן התא הרצוי) למה שכתוב באקומלטור
7. מעבירים את הנקודה בסרט הפקודות לפקודה הבאה.

נגדיר את המחיר של תא i שבתוכו תוכן j להיות $\lg i + \lg j$ (כתיב בינארי, יענו).
הזכרון הנשמר בקונפיגורציה הוא פרופורציוני לסכום מחירי התאים בקונפיגורציית מכונת ה-RAM. מזה נובע שעבור מכונת RAM רצה $T(n)$ על קלט w וסכום המחירים המקסימלי בחישוב הוא p , אזי זמן הריצה של מכונת הטיורינג המסמלצת הוא פרופורציונלי ל $T \cdot p$.
נזכר, כי אנו מדברים על מכונת טיורינג עם k סרטים – כשעוברים למכונת טיורינג רגילה, צריך להעלות בריבוע.

ליכסון

נוכיח כי $A_{TM} \notin R$. הרעיון הוא להגדיר טבלה – למעלה נרשום את כל מכונות הטיורינג מעל Σ^* (נזכור שמספר מכונות הטיורינג הוא בן מניה). למטה נרשום את כל Σ^* (גם בן מניה)

$\Sigma^* \setminus \text{Turing machine}$	m_1	m_2	$\rightarrow$
ε			
a			
$\downarrow$			

$$A_{ij} = \begin{cases} 1, & \text{if the } i\text{'th word is being accepted by the } j\text{'th machine} \\ 0, & \text{otherwise} \end{cases} \quad \text{נגדיר את}$$

ואת	$B =$		M_1	M_2	$\rightarrow$
		$\langle M_1 \rangle$			
		$\langle M_2 \rangle$			
		$\downarrow$			

נגדיר את

$$B_{ij} = \begin{cases} 1, & \text{if } \langle M_i \rangle \text{ is accepted by } M_j \\ 0, & \text{otherwise} \end{cases}$$

נשים לב – לחשב את A_{ij} בהינתן i, j זה לחשב את A_{TM} .
נשים לב – אם אפשר לחשב את A , אז ברור שאפשר לחשב את B (מוכלת).

נראה שאי אפשר לחשב את B (ולכן אי אפשר לחשב את A).
 נתבונן בטבלה הבאה:

	M_1	M_2	$\rightarrow$
$\langle M_1 \rangle$			
$\langle M_2 \rangle$			
$\downarrow$			

. ברור כי $B_{ij} \in \{0,1\}$ (חוץ מהעמודה והשורה הראשונה כמובן)

נגדיר את מכונת הטיורינג M_* כך $M_*(\langle M_i \rangle) = \begin{cases} 0, B_{ii} = 1 \\ 1, B_{ii} = 0 \end{cases}$. הסתירה היא שאין אף עמודה כזו

(ע"פ הגדרתה), ולכן M_* לא נמצאת בטבלה B, והגענו לסתירה. (למה M_* ולא M_k - כי היא לא נמצאת בטבלה מהגדרה...)

20.6.2007

היום נראה עוד שתי שפות NP-שלמות:

1. HAMPATH
2. SUBSET-SUM

הגדרה: מסלול המילטוני מ s ל t בגרף רגיל מכוון G הוא מסלול מ s ל t שעובר בכל קודקודי G בדיוק פעם אחת.

נגדיר: $HAMPATH = \{ \langle G, s, t \rangle \mid * \}$ - G גרף מכוון המכיל מסלול המילטוני מ s ל t

טענה: HAMPATH היא NP שלמה

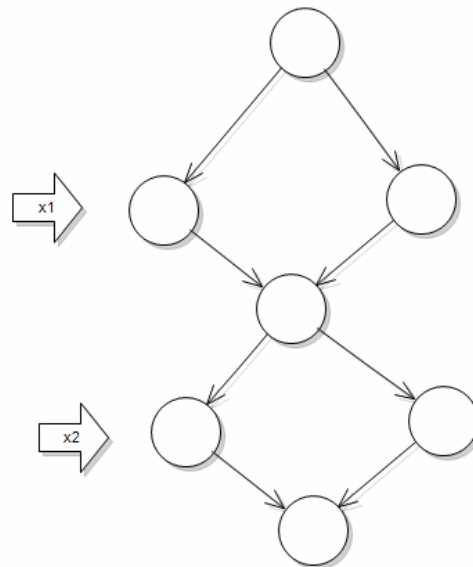
הוכחה:

1. $HAMPATH \in NP$ - נגדיר מסלול, ונבדוק אם הוא המילטוני מ s ל t .

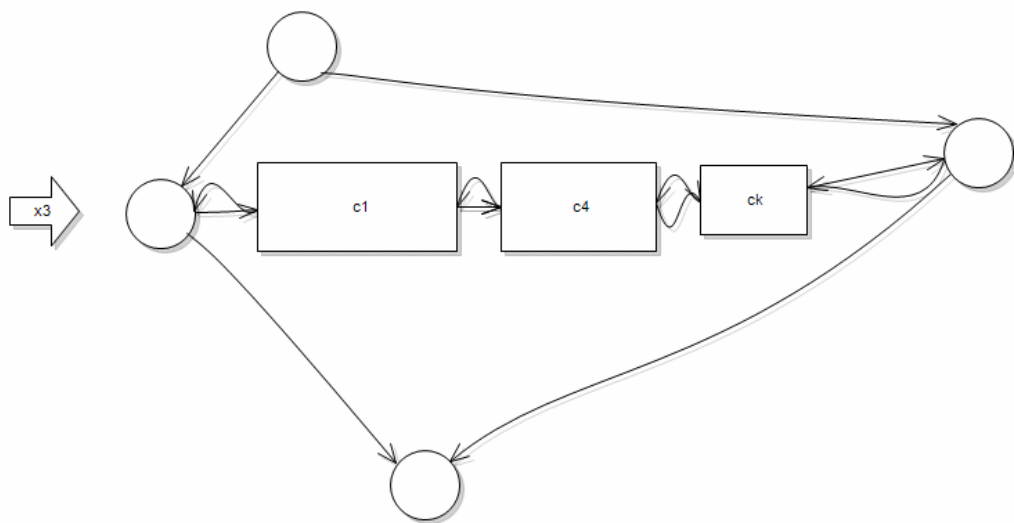
$HAMPATH \in NP - hard$

נראה כי $3SAT \leq_p HAMPATH$. בהינתן נוסחה φ ב $3CNF$ נבנה גרף G ונגדיר קדקדים s, t כך ש: φ ספיקה $\leftrightarrow$ ב G יש מסלול המילטוני מ s ל t .

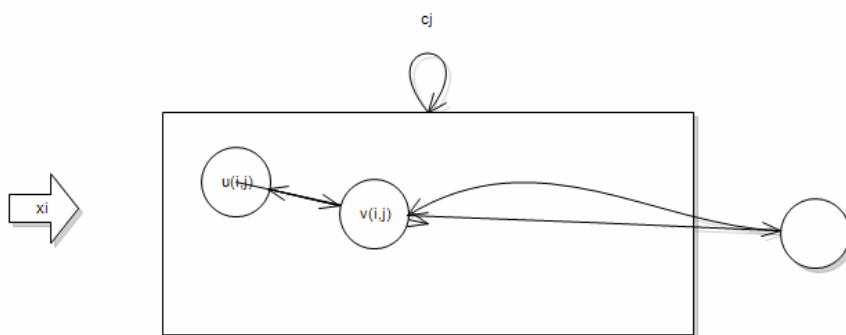
נגדיר $\varphi = (l_1^1 \vee l_2^1 \vee l_3^1) \wedge \dots \wedge (l_1^k \vee l_2^k \vee l_3^k)$ מעל משתנים $x_1, \dots, x_n$ נבנה גרף כך:



איך יראה מעוין כזה לדוגמא? כל מעוין יהיה מאותה צורה, כאשר באמצע יהיה קדקדים עבור הפסוקיות בהן הוא משתתף(?).



איך נראה ריבוע כזה לדוגמא?



קצת יותר מסודר:

- לכל משתנה φ נתאים מעוין שבו $3k + 3$ קדקדים במאונך (ראה לעיל)
- לכל פסוקית j נתאים זוג קודקדים שכנים u_j^i, v_j^i ובכין כל שני זוגות קדקוד רווח.
- בנוסף, לכל פסוקית j נוסיף קדקד c_j .

כעת, אם x_i מופיע ב c_j , נוסיף מעברים $u_j^i \rightarrow c_j \rightarrow v_j^i$.

אם $\overline{x_i}$ מופיע ב c_j , נוסיף $v_j^i \rightarrow c_j \rightarrow u_j^i$. (נשים לב – כיוונים הפוכים).

(יתכן ואצל דינה זה יותר מסודר...)

נשים לב, שברור שבניה כזו היא פולינומית בגודל הנוסחה.

נגדיר s את העליון ביותר, את t בתור התחתון ביותר.

כעת, נראה כי φ ספיקה $\leftrightarrow$ ב G יש מסלול המילטוני מ s ל t .

הוכחה:

עוברים על הקודקודים בזיגזג. או משמאל לימין, או מימין לשמאל (קטונתי מלצייר זאת). בכל הקדקדים חוץ מ c_i אין בעיה.

מסלולים בהם אני עובר משמאל לימין במעוין x_i , משמעות הדבר ש $x_i = 1$, והפוך כמובן. נשים לב – בהשמה מספקת, צריך להיות בכל פסוקית ליטרל אחד שמקבל ערך true.

הוכחה:

$\Leftarrow$ תהי φ ספיקה – תהי $f: \{x_i\} \rightarrow \{0,1\}$ השמה מספקת. המסלול יעבור על כל המעוינים בהתאם להשמה ($f(x_i) = 1 \Leftrightarrow$ מעבר על מעוין x_i משמאל לימין).

אם פסוקית j מסתפקת ע"י משתנה x_i , אז במהלך המעבר על קדקדי x_i נעבור גם ב c_j (זה אפשרי, בגלל הבניה).

$\Rightarrow$ נניח שבG יש מסלול המילטוני מ s ל t .

אם המסלול "יפה" (מהמבנה שאנו רוצים) – אזי יש לנו השמה מספקת (מהבניה).

אם לא, נראה שהוא לא המילטוני. נתבונן במרובע כלשהו, ממנו הלכנו ל c_j , אבל לא חזרנו למרובע. הנקודה היא שצריך לחזור אליו – כדי לבקר בכל שאר הקדקדים בשורה. אבל כל שאר הקדקדים שאני מחובר אליהם נוצלו (כי הגעתי לאחד שממנו ברחתי), ולכן אני לא יכול.

עוד רדוקציה

נגדיר את השפה $SUBSET - SUM = \{(y_1, \dots, y_l, t) \mid *\}$ - קיים $I \subseteq \{1, \dots, l\}$ כך

$$(\sum_{i \in I} y_i = t \text{ ש})$$

טענה: $SUBSET - SUM$ היא NP שלמה.

הוכחה: ראשית, ברור כי $SUBSET - SUM \in NP$ - ננחש עד, ונבדוק.

כעת, נראה כי $SUBSET - SUM \in NP - hard$. גם כאן נראה $3SAT \leq_p SUBSET - SUM$.

נתונה נוסחה: $\varphi = (l_1^1 \vee l_2^1 \vee l_3^1) \wedge \dots \wedge (l_1^k \vee l_2^k \vee l_3^k)$ מעל משתנים $x_1, \dots, x_n$. נבנה רשימה של l מספרים ומספר t כך ש φ ספיקה $\Leftrightarrow$ קיימת תת קבוצה של המספרים שסכומה t .

נגדיר טבלה:

A	B
C	D
T	T

$$A = \begin{array}{c|cccccccc} & x_1 & x_2 & \dots & x_n & & & & \\ \hline x_1 & 1 & & & & & & & \\ \overline{x_1} & 1 & & & & & & & \\ x_2 & & 1 & & & & & & \\ \overline{x_2} & & 1 & & & & & & \\ \dots & & & \ddots & & & & & \\ x_n & & & & 1 & & & & \\ \overline{x_n} & & & & 1 & & & & \\ \hline & & & & & & & & \end{array}$$

A: בעמודה i יש 1 בשורות $2i-1, 2i$ ו-0 בשאר.

$$B = \begin{array}{c|cccccccc} & c_1 & c_2 & \dots & c_k & & & & \\ \hline x_1 & & 1 & & & & & & \\ \overline{x_1} & & & & & & & & \\ x_2 & & & & & & & & \\ \overline{x_2} & & 1 & & & & & & \\ \dots & & & & & & & & \\ x_n & & 1 & & & & & & \\ \overline{x_n} & & & & & & & & \\ \hline & & & & & & & & \end{array}, c_2 = (x_1 \vee \overline{x_2} \vee x_n)$$

B יוגדר כך: בעמודה j יש 1 בשורות המתאימות לליטרלים המופיעים בפסוקית c_j .

C: כולו 0.

D: בעמודה j יש 1 בשורות $2j-1, 2j$.

t : n אחדות בהתחלה (מתחת ל A, C), ולאחר מכן 3ים (סכימת העמודה מעליו).

ברור כי הבניה פולינומית

טענה: φ ספיקה $\Leftrightarrow$ יש תת קבוצה של שורות שסכומן t (יש להתבונן ב t כמספר עשרוני, וגם על כל שורה).

$\Leftarrow$ תהי f השמה מספקת ל φ . מבין $2n$ השורות העליונות, נבחר את השורות שעבורן ההשמה נותנת ערך 1.

מאחר ו f השמה (נבחר x_i או $\overline{x_i}$) בחלק $\frac{A}{C}$ סכום המספרים הוא בדיוק $\underbrace{1\dots 1}_n$.

$\Leftarrow$ בגלל ש f מספקת, בחלק B לכל עמודה הסכום הוא בין 1 ל 3. מבין $2k$ (חלק $\begin{bmatrix} C & D \end{bmatrix}$),

נבחר את השורות כך שסכום כל העמודות ב $\frac{B}{D}$ יהיה 3.

$\Rightarrow$ נניח ש $I \subseteq \{1, 2, \dots, 2n + 2k\}$, כך ש $\sum_{i \in I} y_i = t$. נבנה השמה מספקת ל φ :

▪ ההשמה תיתן ערך 1 לליטרלים שהשורות המתאימות להם נבחרו ל I (מבין $2n$ השורות העליונות, כמובן).

בגלל שסכום כל עמודה ב A הוא 1, קיבלנו השמה חוקית.

בגלל שסכום כל עמודה ב $\begin{array}{|c|} \hline B \\ \hline D \\ \hline \end{array}$ הוא 3, ו D תורמת לכל היותר 2 בכל עמודה, נקבל שסכום כל

עמודה ב B (בשורות שבחרנו) הוא לפחות 1. $\Leftarrow$ כל הפסוקיות מסופקות תחת ההשמה המתאימה.

27.6.2007

הודעות:

תרגיל 10 כן להגשה

על סדר היום:

1. משפטי ההיררכיה
2. ALL_{NFA} היא Pspace – שלמה
3. $TIME(n^2) \subsetneq TIME(n^3)$

ניזכר בהוכחה של קנטור לכך ש $|\mathbb{N}| < |\mathbb{R}|$.

הוכחנו בצורה דומה כי $A_{TM} \notin R$. כיצד? נגדיר טבלה

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	...	
M_1				
M_2				
...				

- בשורות, כל מכונות הטיורינג. בעמודות – כל הקלטים – אנו

נתבונן רק בקלטים שמייצגים מכונות טיורינג.

נניח בשלילה כי $A_{TM} \in R$ - כלומר, היא מסוגלת למלא את הטבלה בזמן סופי.

נייצר מכונה דומה - M_j , שתחזיר על כל $\langle M_i \rangle$ את ההפך ממה שמוחזר ע"י A_{TM} . אבל מהגדרתה, היא לא קיימת בטבלה – ולכן, לא קיימת.

נשים לב – למה אוסף מכונות הטיורינג הוא בר מניה? $\langle M \rangle \subseteq \Sigma^*$. מאחר ויש איזומורפיזם בין $\langle M \rangle$ ל M , אזי ברור כי אוסף מכונות הטיורינג בר מניה.

הגדרה: פונקציה $f: \mathbb{N} \rightarrow \mathbb{N}$ המקיימת $f(n) \geq \lg n$ נקראת space construction (ניתנת לבניה בזכרון) אם קיימת מ"ט M_f שבהנתן קלט אונרי 1^n מחשבת את $f(n)$ בייצוג בינארי במגבלת זכרון $O(f(n))$.

לדוגמא: $f(n) = \lg n$. נתון n באונרית - $\underbrace{11\dots1}_n$ - ואני רוצה לחשב את $\lg n$. איך נעשה זאת? פשוט מאוד, נרשום מונה בבינארי, ונספור – כל פעם נגדיל ב1. כמובן שמגבלת הזכרון שלנו היא $\lg n = O(f(n)) = O(\lg n)$.
(נשים לב – נתייחס לסרט הקלט כreadonly, ויהיה סרט עבודה החסום ב $O(f(n))$).

נתבונן כעת ב $f(n) = n^2$. ראשית, נמיר את המספר לבינארי, ואז נכפיל בכתיב בינארי. ברור כי הפלט יהיה בגודל $2\lg n = O(\lg n) = O(n^2)$.

משפט היררכיית הזכרון

לכל פונקציה f שהיא space constructible, קיימת שפה L_f הניתנת להכרעה במגבלת זכרון $O(f(n))$ אך לא במגבלת זכרון $o(f(n))$.

מסקנות:

- עבור $k_1 < k_2 \in \mathbb{N}$ מתקיים $SPACE(n^{k_1}) \subsetneq SPACE(n^{k_2})$
- לכל $k \in \mathbb{N}$ מתקיים $SPACE(n^k) \subsetneq SPACE(n^{\lg n})$
- לכן, $\bigcup_{k \geq 0} SPACE(n^k) = PSPACE \subseteq SPACE(n^{\lg n})$
- ו $SPACE(n^{\lg n}) \subsetneq SPACE(2^n)$
- ועל כן - $PSPACE \subsetneq SPACE(2^n)$ (וזו מסקנה חשובה!)

חשוב להבין – ההגדרה כאן של $o(f(n))$ היא כזו: $\frac{g(n)}{f(n)} \rightarrow_{n \rightarrow \infty} 0$.
 $g(n) = o(f(n)) \leftrightarrow \frac{g(n)}{f(n)} \rightarrow_{n \rightarrow \infty} 0$
לדוגמא: אם $f(n) = n^3$, אזי לפי ההגדרות כאן - $n^3 = O(n^3)$ אבל $n^3 \neq o(n^2)$.

הוכחה:

אינטואיציות:

נגדיר L_f - תהיה מה ש $\langle M \rangle$ עושה על עצמה, ומחזירה הפוך.

בעיות: מה אם M לא עוצרת?

מה אם M חורגת מ $O(f(n))$?

כלומר, אם מדובר על $f = n^3$, נצטרך למצוא מכונה שתעבוד עם $c \cdot n^3$ זכרון, אבל אי אפשר לסמלץ אותה עם $c_2 \cdot n^2$ זכרון לדוגמא.

נשים לב – מספר הקונפיגורציות היא $|\Sigma|^{f(n)} \cdot |Q|$. אפשר להגיד שזה $O(2^{2 \cdot f(n)})$.

פורמלית

נגדיר M_f הפועלת במגבלת זכרון $O(f(n))$ ומכריעה שפה שאינה כריעה במגבלת זכרון

$o(f(n))$ על קלט w

1. מחשבת $n = |w|$

2. מחשבת את $f(n)$ (אפשרי מאחר ו $f(n)$ space constructible). נסמן על סרט החישוב

קטע באורך $f(n)$. (אם במהלך הסימולציה חורגים מהקטע המסומן, M_f תדחה).

3. אם w אינה מהצורה $10^* \langle M \rangle$ עבור M כלשהיא - M_f דוחה.

4. M_f מסמלצת את M על w תוך כדי ספירת צעד החישוב. אם מגיעים ל $2^{f(n)}$ - M_f דוחה.

5. אם M מקבלת M_f דוחה, ולהפך.

צ"ל:

1. M_f כריעה במגבלת זכרון $O(f(n))$.

a. השתמשנו ב $\sim 3f(n)$ - לכן אנחנו סבבה ☺

2. L_f אינה כריעה במגבלת זכרון $o(f(n))$.

a. נניח בשלילה ש' M מכריעה את L_f במגבלת זכרון $g(n) = o(f(n))$. $\Leftarrow$

קיים n_0 כך שלכל $n > n_0$ מתקיים $g(n) < f(n)$. לכן על כל קלט בגודל

$n_0 < n$ יש ל M_f משאבים לסמלץ את הרצת M' על הקלט עד תומה. ניקח לדוגמא

את המילה $w = \langle M' \rangle 10^{n_0}$. M_f תסמלץ את M' על w , ותחזיר תשובה הפוכה.

$\Leftarrow M'$ לא מכריעה את L_f (כזכור $L_f = L(M_f)$).

טענה: ALL_{NFA} היא Pspace – שלמה

$$ALL_{NFA} = \{ \langle A \rangle \mid L(A) = \Sigma^* \wedge A \text{ is NFA} \}$$

שפה L היא $PSPACE$ שלמה אם:

1. $L \in PSPACE$

2. לכל $L' \in PSPACE$ מתקיים $L' \leq_p L$.

ראשית, ראינו בכיתה כי $ALL_{NFA} \in PSPACE$.

כעת נראה כי ALL_{NFA} היא $PSPACE$ קשה.

כלומר, לכל $L \in PSPACE$ נראה רדוקציה מ ALL_{NFA} ל L . למעשה נראה רדוקציה

$$M \leq_p \overline{ALL_{NFA}}$$

(מספיק לנו, כי $PSPACE$ סגורה למשלים).

נתונה $\overline{L} \in PSPACE$. כלומר, קיימת מכונת טיורינג דטרמיניסטית M הפועלת במגבלת זכרון

פולינומיאלית $s(n)$ ומכריעה את $\overline{L}$.

בהינתן מילה w באורך n , נבנה במגבלת זכרון $O(s(n))$ NFA A כך

$$w \in L(M) \Leftrightarrow L(A) \neq \Sigma^*$$

A יקבל סדרת קונפיגורציות של M - $\#c_0 \#c_1 \# \dots \#c_l \#$

נגדיר A שמקבל חישובים 'לא טובים' של M על w :

1. c_0 לא קונפיגורציה תחילית

2. קיים $0 \leq i < l$ כך ש c_{i+1} לא קונפיגורציה עוקבת אחרי c_i

3. c_l לא קונפיגורציה עוקבת.

איך יראה A ? ינחש אם הוא הולך לקבל בגלל סעיף 1, 2 או 3, ואז יבדוק אותו – אם הוא מגלה שמשוהו לא בסדר, אז הוא יקבל (נכנס לבור מקבל). שרטוט של האוטומט – אצל דינה. ברור שהוא יהיה בגודל

$$O(s(n))$$