מודלים חישוביים וחישוביות - פרופ' אורנה קופרמן - סמסטר א' 2022

~

הרצאות ותרגולים

מסכם: יחיאל מרצבך

הקורס מתרחש כעת ולכן הסיכום מתעדכן מדי פעם.

תאריך עדכון:

24.12.2021

תוכן העניינים

חלק I

הרצאות

הרצאה מס' 1: הקדמה

יום שני

11.10.21

הקורס בחישוביות ממשיך את רצף קורסי התיאוריה במדעי המחשב, כאשר במהלך הקורס נתמקד ביכולת הביצוע של המחשבים ובמחיר של ביצוע זה - בזמן ובזיכרון.
עד כה, התייחסנו בעיקר לחסמים עליונים, וכעת נרצה גם להוכיח כי אי אפשר למצוא חסם טוב יותר.
הקורס מתחלק לשלושה חלקים:

1. מודלים חישוביים (אוטומטים, דקדוקים).
2. חישוביות.
3. סיבוכיות ("באיזה מחיר?").

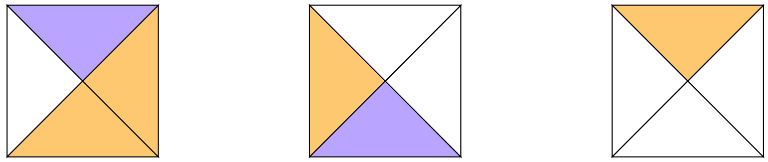
מוטיבציה ללמידת סיבוכיות

נציג כאן טעימה מחלק הסיבוכיות, שבו נמתקד בסוף הקורס.
דוגמה
נתון גרף $G = \langle V, E\rangle$ לא מכוון.
האם ניתן למצוא האם קיים **מעגל אוילר** ב-G (מעגל שעובר בכל הקשתות של G, בכל קשת פעם אחת)?
קיים אלגוריתם ליניארי למציאת מעגל אוילר, כיוון שקיים אפיון מתמטי לכך: יש מעגל אוילר $\Leftrightarrow$ כל הדרגות של הקודקודים זוגיות.
לעומת זאת, לגבי **מעגל המילטון** (מעגל שעובר בכל הקודקודים, בדיוק פעם אחת), קיים רק אלגוריתם אקספוננציאלי - שבודק את כל האפשרויות, ב-$|V|!$.
דוגמה נוספת
יהיו $p, q \in \mathbb{N}$, ואנו רוצים להחזיר את $n = p \cdot q$. קיים אלגוריתם פולינומיאלי לפתרון בעיה זו.
לעומת זאת, בהינתן n, אם נרצה להחזיר $p, q \neq 1$, כך ש-$n = p \cdot q$, לא ידוע על אלגוריתם פולינומיאלי.

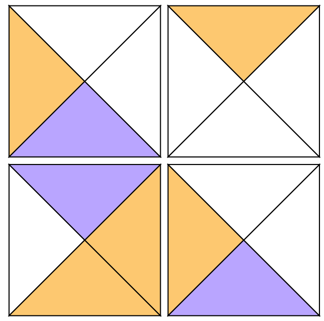
אם נרצה לעשות סיווג של בעיות למחלקות סיבוכיות, מצד אחד לא נשאף ליותר מדי, ולכן יש לגיטימציה לקירובים, ומצד שני ננצל קושי קיים (קריפטוגרפיה).

מוטיבציה ללמידת חישוביות

דוגמה
נתונה קבוצה T של אריחים, כאשר כל אריח מחולק לארבע, וכל אחד צבוע בצבע אחר.
עלינו להכריע האם קיים ריצוף חוקי $n \times n$ לכל $n \geq 1$, כאשר ריצוף חוקי הוא כי אריחים סמוכים 'מסכימים' על הצבע.
עבור הקלט הבא מסתבר שהתשובה היא כן:



לדוגמה, ניתן לרצף ריבוע 2×2 באופן הבא:



נוכיח כי אין אלגוריתם לפיתרון בעיה זו.

דוגמה נוספת - בעיית העצירה

נתונה תוכנית מחשב P עם קלט x לתוכנית P, ונרצה להכריע האם P עוצרת על x.

1 מודלים חישוביים

1.1 שפות רגולריות ואוטומטים דטרמיניסטיים (DFA)

על מנת לסבר את האוזן, נתחיל בדוגמה.

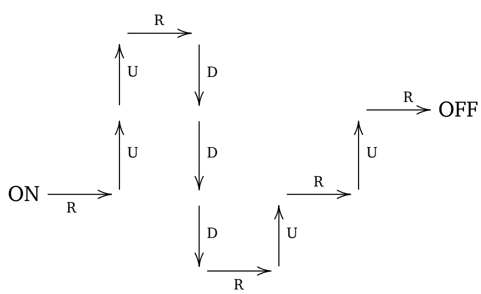
דוגמה

נתון עט דיגיטלי, שיכול לקבל 6 פקודות בכל רגע נתון - ON, OFF, U, D, R, L.

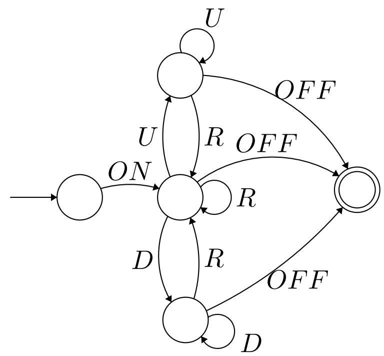
נאמר כי סדרת פקודות חוקית אם היא:

- מתחילה ב-ON ומסתיימת ב-OFF.
- אחרי פקודת U אין פקודת D ולהפך.
- היא מציירת קו רקיע משמאל לימין.

כאשר ניתן לראות דוגמה לקו רקיע בדוגמה הבאה:



אוטומט מגדיר סדרת פקודות חוקית, כפי שנגדיר ונדגים להלן, אך אפשר לראות זאת גם בציור הבא:



הגדרה

אלפבית (א״ ב) הוא קבוצה סופית ולא ריקה של אותיות $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$.

דוגמה

$\Sigma = \{0,1\}$ או $\Sigma = \{0,1\}^4$.

הגדרה

מילה היא סדרה סופית של אותיות, המילה הריקה תסומן בתור ε.

הגדרה

הקבוצה $\Sigma^\star$ היא אוסף כל המילים מאורך סופי מעל Σ.

הגדרה

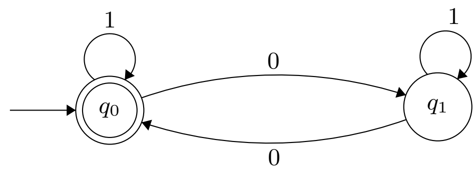
שפה L היא קבוצה של מילים. כלומר, $L \subseteq \Sigma^\star$.

הגדרה

אוטומט דטרמינסטי automaton או DFA הוא חמישיה $A = \langle Q, \Sigma, \delta, q_0, F\rangle$ כאשר:

1. Q היא קבוצה סופית של מצבים.
2. Σ היא א״ב.
3. $\delta : Q \times \Sigma \to Q$ היא פונקציית מעברים.
4. $q_0 \in Q_0$ הוא מצב התחלתי.
5. $F \subseteq Q$ קבוצת מקבלים\סופיים.

דוגמא (A_1):



בדוגמא הזאת יש אוטומט דטרמיניסטי סופי עם 2 מצבים, כלומר $Q = \{q_0, q_1\}$.
הא"ב הוא $\Sigma = \{0, 1\}$, המצב ההתחלתי הוא q_0 והמצבים המקבלים הם $F = \{q_0\}$.

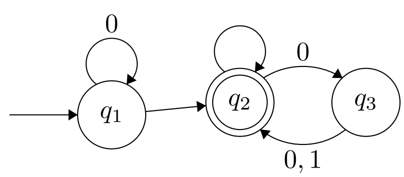
הגדרה
בהינתן מילה $w = w_1 \cdot w_2 \cdot w_3 \cdots w_n \in \Sigma^\star$, ריצה של A על w היא סדרה $r = q_0, q_1, \dots q_n$ של מצבים כך ש:

1. r מתחילה במצב ההתחלתי q_0.
2. לכל $0 \leq i < n$ מתקיים כי $q_{i+1} = \delta\left(q_i, w_{i+1}\right)$.

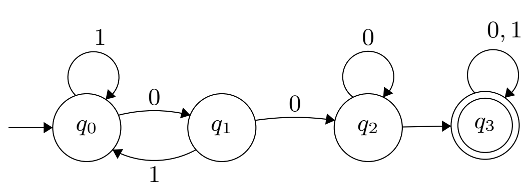
הגדרה
נאמר כי r היא ריצה **מקבלת**, אם $q_n \in F$.
האוטומט A **מקבל** את w, אם הריצה של A היא **מקבלת**.
נאמר כי השפה של A היא כל המילים w כך ש-A **מקבל** את w:

$$L\left(A\right) = \{w \mid w \text{ מקבל את } A\}$$

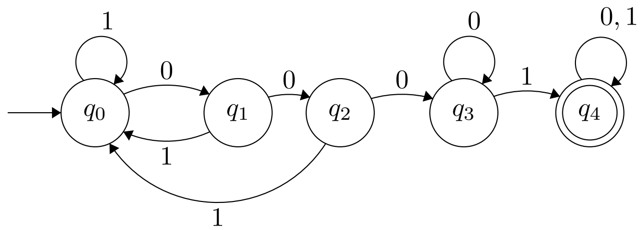
בדוגמה הקודמת, למשל, השפה היא כל המילים כך שיש בהם מספר זוגי של 0-ים.
דוגמת הרצה נוספת
נריץ דוגמה שנייה:



כאן $L\left(A_2\right)$ היא כל המילים שיש להן לפחות 1 אחד ואחרי ה-1 האחרון יש מספר זוגי של 0-ים (ייתכן 0).
דוגמה שלישית
נרצה אוטומט A_3 כזה ש-w מכילה את המילה 001. הדוגמה לכך היא:



כעת, נרצה אוטומט A_4 כך ש-w מכילה את תת המילה 0001, והדוגמה לכך הינה:



אבחנה

ככלל, נוכל לומר שההגדרה של האוטומט $L\left(A_n\right)$ כך ש-w מכילה את תת המילה 0^n1, הינה:

$$A = \left\langle \left\{q_0, \ldots, q_{n+1}\right\}, \left\{0, 1\right\}, \delta_n, q_0, \left\{q_{n+1}\right\} \right\rangle$$

כאשר:

1. $\delta\left(q_{n+1}, 0\right) = \delta\left(q_{n+1}, 1\right) = q_{n+1}$ עבור $0 \leq i < n$.
2. $\delta\left(q_i, 0\right) = q_{i+1}$.
3. $\delta\left(q_i, 1\right) = q_0$.
4. $\delta\left(q_n, 0\right) = q_n$.
5. $\delta\left(q_n, 1\right) = q_{n+1}$.

דוגמה אחרונה

אוטומט עבור $L_{eq} = \left\{0^n 1^n : n \geq 0\right\}$ - אין אוטומט עבור L.

הגדרה

שפה $L \subseteq \Sigma^\star$ היא **רגולרית** אם קיים אוטומט A כך ש-$L\left(A\right) = L$.

1.1.1 פעולות על שפות

באפשרותנו לבצע מספר פעולות על שפות. ראשית, איחוד, חיתוך, ושאר פעולות שאפשר לבצע על קבוצות. שנית, קיימות גם שתי פעולות נוספות, שנגדיר אותן כעת.

הגדרה

יהיו L_1, L_2 שפות. נגדיר את **השרשור** להיות:
$L_1 \cdot L_2 = \left\{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\right\}$.

דוגמה

אם $w_1 = \sigma_1, \sigma. \ldots . \sigma_n$ ו-$w_2 = \sigma'_1, \sigma'_2, \ldots, \sigma_{m,}$, נקבל כי $w_1 \cdot w_2 = \sigma_1, \sigma. \ldots . \sigma_n, \sigma'_1, \sigma'_2, \ldots, \sigma_{m,}$.

הגדרה

תהי L שפה. נגדיר את פעולת **הכוכב** להיות:

$L^* = \{w_1 \cdot w_2 \ldots \cdot w_k \mid k \geq 0 \ \ , w_i \in L, 1 \leq i \leq k\}$

נבחין כי L^* היא סופית, רק כאשר $L = \emptyset$ ו-$L = \{\varepsilon\}$.

במקרה זה $L^* = \{\varepsilon\}$.

1.1.2 תכונות סגור של השפה הרגולריות

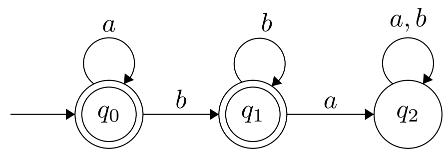
הרצאה מס' 2:

יום רביעי

13.10.21

מוטיביציה

נתבונן לרגע באוטומט הבא - A_1:



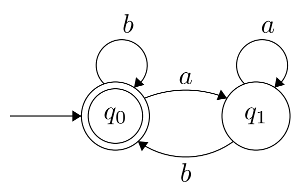
השפה $L(A_1)$ מורכבת משרשרת (ייתכן ריקה) של a-ים ואחר כך שרשרת (ייתכן ריקה) של b-ים.

כלומר, ניתן לרשום זאת גם בתור $a^i, b^j, i, j \geq 0$.

כעת, נשאל את השאלה ההפוכה, מהו האוטומט A_2 של השפה $L(A_2)$:

"אחרי כל a יש בסופו של דבר b" כאשר $\Sigma = \{a, b\}$.

נקבל את האוטומט הבא:



אם היינו מרחיבים את השפה, כלומר כעת $\Sigma = \{a, b, c\}$, לא היה משתנה דבר.

לעומת זאת, אם היינו לוקחים את השפה L "מספר ה-a-ים ב-w שווה למספר ה-b-ים ב-w", בלתי אפשר לצייר לו אוטומט, ושאלה זו מקבילה למעשה לאוטומט שראינו - $\text{Leq} = a^n \cdot b^n$. נוכל להבחין כי למעשה $\text{Leq} = L \cap L(A_1)$.

מכאן יש לנו מוטיבציה ל"תכונות סגור", שהרי ידוע כי Leq לא רגולרית, וגם ידוע כי $L(A_1)$ רגולרית, ואם נוכל להוכיח כי חיתוך שפות רגולריות הוא רגולרי, סימן ש-L לא רגולרית, ונוכל להוכיח זאת בקלות.

אבחנה

כשאנו מתייחסים ל"תכונות סגור", או "קבוצה סגורה לפעולה", הכוונה שהתוצאה של הפעלת פעולה כלשהי על איברים בקבוצה, נשארת בקבוצה.

למשל, המספרים הטבעיים סגורים לכפל כי אם $x, y \in \mathbb{N}$ אזי גם $x \cdot y \in \mathbb{N}$, אבל הם לא סגורים לחילוק כי $\frac{x}{y} \notin \mathbb{N}$.

נרצה כעת להראות כי הפעולות שהראינו קודם על שפות, סגורות בשפה.

משפט

אם L_1 רגולרית, ו-L_2 רגולרית, אזי $L_1 \cup L_2$ רגולרית.

כלומר, השפות הרגולריות סגורות לאיחוד.

הוכחה

יהי $A_1 = \left\langle Q_1, \Sigma, \delta_1, S_1^0, F_1 \right\rangle$ אוטומט DFA כלשהו עבור L_1 ויהי $L_2 = \left\langle Q_2, \Sigma, \delta_2, S_2^0, F_2 \right\rangle$ אוטומט DFA כלשהו, עבור L_2.[1]

נבנה $A = \left\langle Q, \Sigma, \delta, S^0, F \right\rangle$ כך ש-$L(A) = L(A_1) \cup L(a_2)$, ונוכל לעשות זאת באמצעות אוטומט המכפלה:

$$Q = Q_1 \times Q_2 = \{\langle q_1, q_2 \rangle \mid q_1 \in Q_1, q_2 \in Q_2\}$$

הרעיון: A מבקר במצב $\langle q_1, q_2 \rangle$ אחרי קריאת $w \in \Sigma^*$, אם A_1 מבקר ב-q_1 אחרי קריאת w ו-A_2 מבקר ב-q_2 אחרי קריאת w.

כעת, נגדיר את δ ו-F :

$$\delta\left(\langle q_1, q_2 \rangle, \sigma\right) = \langle \delta\left(q_1, \sigma\right), \delta\left(q_2, \sigma\right) \rangle = S_0 = \left\langle S_1^0, S_2^) \right\rangle$$

וגם:

$$F = \{\langle q_1, q_2 \rangle \mid q_2 \in F_2 \vee q_1 \in F_1\} = (F_1 \times Q_2) \cup (Q_1 \times F_2)$$

פונקציית המעברים שלמה (מוגדרת לכל מצב ואות).

הוכחת נכונות של הבנייה

תהי $w = \sigma_1 \cdot \sigma_2 \ldots, \sigma_n \in \Sigma^*$ מילה.

נתבונן בריצה של A על w:

$$r = \left\langle q_1^0, q_2^0 \right\rangle, \left\langle q_1^1, q_2^1 \right\rangle, \left\langle q_1^2, q_2^2 \right\rangle, \ldots, \left\langle q_1^n, q_2^n \right\rangle$$

ונשים לב כי מתקיים ש-$r_1 = q_1^0\, q_1^1\, q_1^2\, q_1^3\, \ldots\, q_1^n$ היא הריצה של A_1 על w, וגם מתקיים כי $r_2 = q_2^0\, q_2^1\, q_2^2\, q_2\, \ldots\, q_2^n$ היא הריצה של A_2 על w (ההטלה של r על האיבר הראשון או השני בכל זוג).

מהגדרת F מתקיים כי r מקבלת אם"ם r_1 או r_2 מקבלות.

אם כך, סיימנו את ההוכחה, כנדרש.

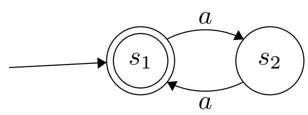
דוגמה

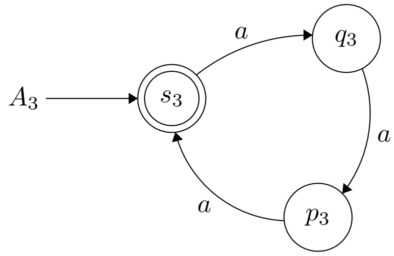
תהי $\Sigma = \{a\}$.

$L_1 = \{w \mid \ w = 0 \mod 2\}$ ו-$L_2 = \{w \mid \ w = 0 \mod 3\}$.

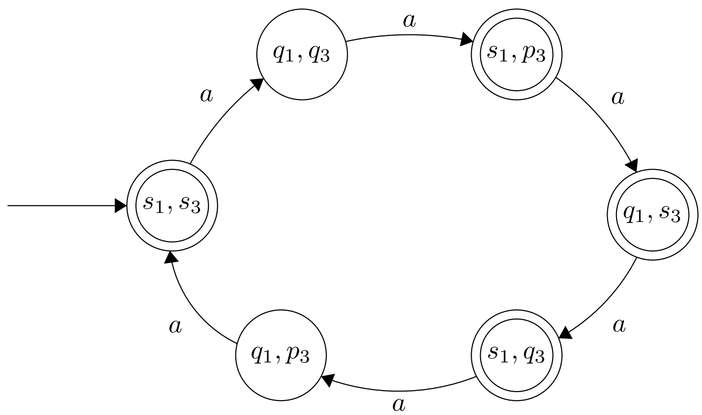
האוטומטים הם:

[1] הנחנו ש-L_1 ו-L_2 מעל אותו א"ב Σ, כיוון שאחרת נוכל לקחת את $\Sigma = \Sigma_1 \cup \Sigma_2$ ולא תהיה בעיה.





והאוטומט שנבנה יהיה אוטומט האיחוד של $A_1 \cup A_3$, כלומר כל המילים שאורכן זוגי או כפולה של 3:



על מנת לבנות אוטומט של חיתוך, נצטרך לבנות את אוטומט המכפלה עם $F = F_1 \times F_2$.
כדי לבנות אוטומט של השלמה, נבנה את אותו אוטומט, עם $\widetilde{F} = Q \setminus F$ (כל המילים שלא התקבלו באוטומט המקורי).

הרצאה מס' 3:

יום שני

18.10.21

(הערה, החלק הזה עד אוטומט דטרמינסטי לא לגמרי ברור)
נתבונן באוטומט הבא:
אם $w \in L(A_1)$ אזי $w \in L(A_2)$. כלומר $A_1 \to A_2$.
על מנת לייצר DFA עבורו, נצטרך:

- איחוד של $\widetilde{A_1}$ ו-A_2.
- אוטומט מכפלה עם $F = ((Q_1 \setminus F_1) \times Q_2) \cup (Q_1 \times F_2)$

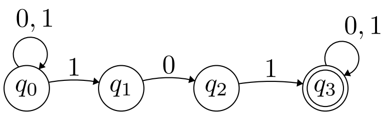
מה אם נרצה לבנות אוטומט עבור a^{10}? אז נרצה לשרשר את $\left(a^2\right)^5$ עם ε, אבל מצד שני, נצטרך לדעת מתי 'נגמרת המילה'.
נשים לב כי באופן כללי, על שפות מעל א"ב $\Sigma = \{a\}$, השפה מתאימה לתת קבוצה של $\mathbb{N}$ ("איזה אורכים נכנסים לשפה").

1.2 אוטומט סופי לא דטרמיניסטי (NFA)

מוטיבציה

נתבונן באוטומט הבא.

$\Sigma = \{0,1\}$ והשפה תהיה כל המילים שמכילות את תת המילה 101. האוטומט המתאים הוא:



זהו אוטומט שלפעמים עובד, כאשר בריצה מסוימת, אם האוטומט 'מנחש', ייתכן והמילים יהיו בשפה. כלומר, ההבדל הוא שקודם לכן היה אפשרות **חד משמעית** לאן ללכת, ואילו כאן אנחנו מאפשרים באמצעות מספר כלשהו, למשל, ללכת לשני מצבים.

כמו כן, יש צעדי ε, שמאפשרים 'לקפוץ' ממצב אחד לאחר. במצב כזה, השפה היא כל המילים שמכילות 101 או 11. דבר זה מוביל אותנו להגדרה הבאה:

הגדרה

אוטומט סופי **לא דטרמיניסטי** (NFA) הוא חמישיה $A = \langle Q, \Sigma, \delta, Q_0, F \rangle$ כאשר:

- Q קבוצה סופית של **מצבים**.
- Σ הוא **א"ב**.
- $Q_0 \subseteq Q$ קבוצה סופית של **מצבים התחלתיים**.
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ **פונקציית מעברים**.
- $F \subseteq Q$ קבוצת **מצבים מקבלים**.

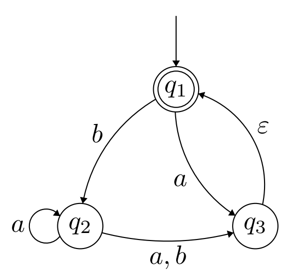
הגדרה

ריצה של A על **מילה** $w = \sigma_1 \ldots \sigma_n$ היא סדרת מצבים $r = r_0, r_1, \ldots, r_m$ (עבור $n \le m$).
כך שניתן לכתוב את w כ- $y_1\, y_2\, y_3\, \ldots y_m$ ו-$y_i \in \Sigma$ וגם $r_0 \in Q_0$.
כמו כן, לכל $0 \le i < m$ מתקיים כי $r_{i+1} \in \delta\left(\sigma_i, y_{i+1}\right)$.
נאמר ש-A **מקבלת** את אם קיימת ריצה מקבלת של A על w.

האוטומט הקודם הוא דוגמה לאוטומט לא דטרמיניסטי.

דוגמה נוספות

אוטומט מעל $\Sigma = \{a,b\}$, בתיאור הבא:



אילו מילים בשפה? $\varepsilon, a, baba, \varepsilon a$.

משפט

לכל NFA יש DFA שקול.

הוכחה

נוכיח את המשפט ללא צעדי ε. בתרגול נראה כיצד ניתן לעבור מ-NFA עם צעדי ε ל-NFA ללא צעדי ε ובכך נשלים את ההוכחה.

בהינתן NFA עם $A = \langle Q, \Sigma, Q_0, \delta, F \rangle$ נבנה DFA עם $A' = \langle Q', \Sigma, q_0, \rho, F' \rangle$ כך ש-$L(A') = L(A)$.

הבנייה

נגדיר כי $Q' = 2^Q$ ו-A' נמצא במצב $S \in 2^Q$ אחרי קריאת w, אם"ם A עשוי להימצא בכל אחד ממצבי S אחרי קריאת w.

כמו כן, מתקיים כי $q_0 = Q_0 \in 2^Q$ - כל מצב יהיה בעצם קבוצה של Q.

מעבר לזה, נגדיר $\rho : 2^Q \times \Sigma \to 2^Q$ פונקציית **המעברים** הדטרמיניסטית של A', כך ש-$\rho(S, \sigma) = \bigcup_{s \in S} \delta(s, \sigma)$.

כלומר, מדובר בכל המצבים שאפשר להגיע אליהם מאחד המצבים מ-S.

לבסוף, נגדיר את $F' = \{S : S \cap F \neq \emptyset\}$ - כלומר, כל **המצבים שמכילים מצב מקבל** באוטומט ההתחלתי.

בנייה זו נקראת **Subset Construction**.

הוכחת נכונות הבנייה

ראינו בתרגול כי ניתן להרחיב את ρ ל-ρ^* כאשר $\rho^* : Q' \times \Sigma^* \to Q'$, דהיינו $\rho^*(S, w) = S'$, כשנמצאים במצב $S \in Q'$ וקוראים w, מגיעים למצב $S' \in Q'$.[2]

כמו כן, ניתן להרחיב את פונקציית המעברים גם ב-NFA. כלומר, $\delta^* : 2^Q \times \Sigma^* \to 2^Q$, כך ש-$\delta^*(S, w)$ היא קבוצת המצבים שניתן להגיע אליהם בקריאת w ממצב ב-S.

ההגדרה תהיה באינדוקציה על $|w|$:

לכל קבוצה $S \in 2^Q$, יתקיים כי $\delta^*(S, \varepsilon) = S$ ומאידך, $\delta^*(S, w \cdot \sigma) = \bigcup_{t \in \delta^*(S,w)} \delta(t, \sigma)$.

טענה

לכל מילה $w \in \Sigma^*$ יתקיים כי $\delta^*(Q_0, w) = \rho^*(q_0, w)$

(הצד השמאלי הוא מצבים ש-A עשויה לבקר בהם אחרי קריאת w, והצד הימני הוא המצב - קבוצה של מצבי A-ש-A' נמצאת בו אחרי קריאת w).

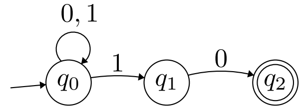
הוכחה

באינדוקציה על $|w|$.

מטענה זו נובעת נכונות הבנייה:

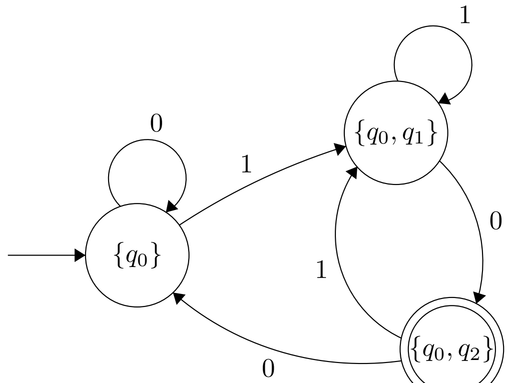
אנו יודעים כי $w \in L(A)$ אם ורק אם $\delta^*(Q_0, w) \cap F \neq \emptyset$ אם ורק אם $\rho^*(q_0, w) \in F'$ אם"ם $w \in L(A')$.

על מנת להסביר זאת, נתבונן בדוגמה הבאה:



[2] הרחבה של פונקציית המעברים לפונקציית מעברים על מילה

השפה היא מילים שמסתיימות ב-10. מדובר באוטומט לא דטרמינסטי.
על מנת לייצר אוטומט דטרמיניסטי, נוכל לבנות את האוטומט הבא:



1.2.1 חסם תחתון לדטרמינזציה[3]

הרצאה מס' 4:
יום רביעי
20.10.21

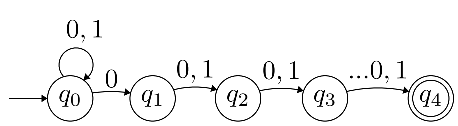
המעבר שעשינו קודם לכן מ-NFA ל-DFA דרש 'ניפוח אקספוננציאלי'. כעת נראה שיש חסם תחתון לפעולת הדטרמינזציה הזו ("חרצון" בעברית), ולא קיימת בנייה טובה יותר מה-**Subset Construction**.
דוגמה קונקרטית של מספרים לא תעזור כאן כיוון שאנחנו רוצים להוכיח באופן כללי כי **לא קיים** פולינום $P:\mathbb{N}\to\mathbb{N}$ כך שבהינתן NFA עם n מצבים, קיים DFA שקול עם $P(n)$ מצבים. הדרך לעשות זאת היא לא דרך דוגמה ספציפית, אלא באמצעות משפחה של שפות.

דוגמה

נראה משפחה של שפות L_1, L_2, L_3 כך ש-L_i, לכל $1 \le i \le n$:

- יש ל-L_n NFA עם $O(n)$ מצבים.
- ה-DFA הקטן ביותר עבור L_n צריך לפחות 2^n מצבים.

נניח בשלילה כי יש פולינום p כך ש-DFA $\overset{p(n)}{\to}$ NFA.
בהכרח, קיים n_0 כזה ש-$p(n_0) < 2^{n_0}$. נתבונן ב-L_{n_0} בו יש NFA עם n_0 מצבים ל-L_{n_0} אבל כל DFA צריך לפתור 2^{n_0} מצבים, שזה יותר מ-$p(n_0)$.
נבחר $\Sigma = \{0,1\}$ ואת השפה להיות כל המילים שיש בהן 0 במקום ה-n מהסוף.
האוטומט המתאים הוא:



טענה

כל DFA שנסמנו D_n המקבל את השפה L_n, מכיל לפחות 2^n מצבים.

הוכחה

נניח בשלילה כי יש D_n שמזהה את L_n ויש לו פחות מ-2^n מצבים. אם נתבונן בכל המחרוזות באורך n' בהכרח, מעיקרון שובך היונים, יש שתי מחרוזות $x, y \in \Sigma^*$ כך ש- $x \neq y$ וגם שתיהן מסיימות באותו המצב של A. נבחר

[3]הערת המסכם: שיניתי כאן קצת את סדר ההרצאות, כך שההגדרות של הביטויים הרגולריים מופיעות לאחר הוכחת החסם התחתון.

את ה- $1 \le i \le n$ הימני ביותר כך ש-$x[i] \neq y[i]$. נניח, בלי הגבלת הכלליות כי ב-x רשומה בתא ה-i הספרה 0 ואילו ב-y רשומה בא ה-i הספרה 1.

נוסיף מחרוזת u, ונתבונן בשתי המחרוזות החדשות xu ו-yu (כאשר u באורך $i-1$). נבחין כעת כי במקום ה-n מהסוף של xu נמצאת הספרה 0, ואילו ב-yu יש את הספרה 1. אם כך $xu \notin F$ ואילו $yu \in F$. כלומר, קיבלנו שתי ריצות שמסיימות באותו המצב (בתוספת u, ששווה בשניהם), אך האחת מקבלת והאחת לא.

הנחת השלילה קרסה ומכאן שכל D_n מכיל לפחות 2^n מצבים.

1.3 ביטויים רגולריים

הרצאה מס' 5:

יום שני

25.10.21

(מוקלט)

הגדרה

בהינתן א"ב Σ, ביטוי רגולרי מוגדרת רקורסיבית:

- $a \in \Sigma$, ε ו-$\emptyset$ הם ביטויים רגולריים.
- אם r_1 ו-r_2 ביטויים רגולריים כך גם:
  - $r_1 \cup r_2$ ביטוי רגולרי.
  - $r_1 \cdot r_2$ ביטוי רגולרי.
  - r_1^* ביטוי רגולרי.

כל ביטוי רגולרי r, מגדיר את השפה $L(r)$:

- $L(\emptyset) = \emptyset$, $L(\varepsilon) = \{\varepsilon\}$, $L(a) = \{a\}$.
- $L(r_1 + r_2) = L(r_1) \cup L(r_2)$.
- $L(r_1 \cdot r_2) = L(r_1) \cdot L(r_2)$.
- $L(r_1^*) = (L(r_1))^*$.

לשם הנוחות, נסמן מעתה ביטוי רגולרי בתור 'ב"ר'. כמו כן, נסמן $r_1^+ = r_1 \cdot r_1^*$.

דוגמאות

מילים מעל $\Sigma = \{0,1\}$.

- מילים עם 0 יחיד, יצוינו בתור $1^* \cdot 0 \cdot 1^*$.
- מילים עם לפחות 0 אחד יצוינו בתור $(0+1)^* \cdot 0 \cdot (0+1)^*$.
- מילים עם לפחות 0 אחד בשלושת האותיות האחרונות יצוינו בתור
  $\left((0+1)^* \cdot 0 \cdot (0+1) \cdot (0+1)\right) + \left((0+1)^* \cdot 0 \cdot (0+1)\right) + \left((0+1)^* \cdot 0\right)$
  או בתור $(0+1)^* \cdot 0((0+1)(0+1) + (0+1) + \varepsilon)$
  או $(0+1)^*0(0+1+\varepsilon)(0+1+\varepsilon)$.
  הביטוי המשלים (מילים שאין בהם 0 באחת משלושת המילים האחרונות) הוא:
  $\varepsilon + 1 + 11 + (0+1)^* 111$.
- מילים שאין בהם את הרצף 00 או הרצף 11 יצוינו בתור $(\varepsilon + 1)^* (10)^* (\varepsilon + 0)$.

- מילים באורך זוגי, יצוינו בתור $((0+1)\cdot(0+1))^*$.

משפט

לכל שפה $L \subseteq \Sigma^*$, L רגולרית אם"ם יש ב"ר r כך ש-$L(r)=L$.

הוכחה

נציג רק את מבנה ההוכחה הכללי.

בכיוון הראשון $\Rightarrow$ נצטרך לתרגם ב"ר לאוטומטים.

בכיוון השני $\Leftarrow$ נצטרך לתרגם אוטומט לב"ר.

ההוכחה עצמה מופיעה בתרגול.

1.4 למת הניפוח ומשפט מייהל - נרוד

1.4.1 למת הניפוח

מוטיבציה

כבר ציינו כי השפה $L=\{0^n1^n \mid n\geq 0\}$ איננה רגולרית. אבל לא הוכחנו זאת בצורה קונסטרוקטיבית.

נרצה למצוא דרך להוכיח זאת בצורה פורמלית ומלאה.

טענה

השפה $L=\{0^n1^n \mid n\geq 0\}$ איננה רגולרית.

הוכחה

נניח בשלילה כי יש DFA שמתאים לשפה[4] שנסמנו ב-A, כך ש-$L(A)=L$.

יהי p מספר המצבים ב-A, ונתבונן במילה $w=0^p1^p$. בהכרח קיימת ריצה $r=q_0,q_1,\ldots,q_{2p}$ של A על w.

מכאן עולה כי $q_{2p}\in F$. לאוטומט יש p מצבים ולכן קיימים $0\leq l<j\leq p$ כך ש-$q_j=q_p$ (מגיעים לאותו מצב פעמיים בקריאת 0, מעיקרון שובך היונים, שהרי קוראים $p+1$ מצבים).

כעת נבחר $j=p-l$, (השמטנו את הסיבוב) - האוטומט A יקבל גם את המילה 0^j1^p, שהרי גם עם פחות 0-ים המילה מתקבלת.

אך לפי הגדרת השפה, אמורות להתקבל מילים רק בעלות מספר אפסים ואחדים שונה. כלומר, קיבלנו סתירה.

מעבר לכך, קיימת סתירה לדטרמיניזם, שכן ישנן שתי 'דרכים' לבחור אפסים, האחת שממשיכה את הסיבוב והאחת שמתקדמת לכיוון ה-1-ים.

מסתבר שההוכחה הזאת והשימוש בעקרון שובך היונים אינם ספציפיים בשפה זו.

משפט (למת הניפוח)

אם L רגולרית, אז קיים $p\geq 1$ קבוע ניפוח, כך שלכל $w\in\Sigma^*$, אם $|w|\geq p$, אזי קיימת חלוקה xyz, כך ש-$w=xyz$ וגם:

- $|y|>0$ ($y\neq\varepsilon$).
- $|xy|\leq p$ (מותר כי $z=\varepsilon$ או $x=\varepsilon$).
- לכל $i\geq 0$, המילה $xy^iz\in L$ (למשל $xyyyz\in L$).

[4] יש להבחין שכאשר מניחים בשלילה, עדיף לעבוד עם המודל החלש יותר, שכן קל יותר למצוא הפרכה לדטרמיניסטיות.

הוכחה

תהי L שפה רגולרית. נצטרך למצוא p מתאים, שיוכיח את הטענה.

יהי A DFA עבור L. נבחר את p להיות $|Q| \in A$ (מספר המצבים).

כעת, נתבונן ב-$w \in L$, כך ש-$|w| \geq p$. בריצה של A על w קיימים בהכרח l, j כך ש-$1 \leq l < j \leq p$.

נסמן $w = w_1, \ldots, w_n$ ונקבל:

$$\underbrace{q_0 \xrightarrow{w_0} \cdots \xrightarrow{w_l} q_l}_{x} \underbrace{\xrightarrow{w_{l+1}} \cdots \xrightarrow{w_j} q_j}_{y} \underbrace{\xrightarrow{w_{j+1}} \cdots \xrightarrow{w_n} q_n}_{z}$$

נבחין כי המצבים שמסומנים בירוק שווים, לפי מה שהסברנו מקודם ("עשינו סיבוב"). נראה שאכן מתקיימים התנאים:

- $|y| > 0$ כי $l < j$ ולכן $y = j - l$.
- $|xy| \leq p$ כי $j \leq p$ ולכן החזרה תקרה "בתוך" p הצעדים הראשונים.
- לכל $i \geq 0$, המילה $xy^iz \in L$ כי $q_1, \ldots q_l, (q_{l+1}, \ldots, q_j)^i, q_{j+1}, q_n$ ריצה מקבלת שהרי $q_n \in F$.

דוגמה לשימוש בלמה

ניקח את השפה שכוללת את כל המילים עם 0 במקום הלפני אחרון, כלומר את $L = (0+1)^*0(0+1)$.

נראה ש-L מקיימת את למת הניפוח. נבחר את $p = 3$, ונראה שלכל מילה $w \in L$ ניתן לחלק ל-xyz כנדרש.

נבחר את $x = \varepsilon$, את $y = w[1]$ ו-$z = w[2 - |w|]$ (כל השאר).

נראה שתנאי הלמה מתקיימים:

- אכן $|y| > 0$ מההגדרה, שכן $|y| = 1$.
- $|xy| \leq 3$ שכן $|xy| = 1$.
- לכל $i \geq 0$, מתקיים כי $xy^iz \in L$, שכן הסיפא z לא משתנה וה-0 נמצא בתוכה.

1.4.2 משפט מייהל-נרוד

הרצאה מס' 6:

יום רביעי

27.10.21

(מאיה)

הגדרה

יהי Σ א"ב ותהא $L \subseteq \Sigma^*$. יחס השקילות מייהל-נרוד של L מוגדר כך: לכל $x, y \in \Sigma^*$ נגדיר כי $x \sim_L y$ אם ורק אם לכל $z \in \Sigma^*$ מתקיים כי $y \cdot z \in L \Leftrightarrow x \cdot z \in L$.

דוגמה

ניקח את השפה $(0 \cup 1)^* \cdot 0 \cdot (0 \cup 1)$.

נבחין כי יש 0 במקום אחד לפני האחרון.

- $111 \sim_L 11$ כי הן לא בשפה, אבל אם נוסיף את אותה סיפא לשתיהן (למשל 01), הן יהיו בשפה.
- $11 \not\sim_L 10$ כי למשל $111 \notin L$ אבל $101 \in L$.

טענה

לכל $L \subseteq \Sigma^* v$ היחס $\sim_L$ הוא יחס שקילות.

הוכחה

ראשית, נבחין $\sim_L$ רפלקסיבי. כלומר $x \sim_L x$, כי הרי $xz \in L \Leftrightarrow xz \in L$ לכל $z \in \Sigma^*$ (טאוטולוגיה).

כמו כן, $\sim_L$ סימטרי כי הרי מהסימטריה של האופרטור $\Leftrightarrow$ עולה כי

$(yz \in L \Leftrightarrow xz \in L) \Leftrightarrow (xz \in L \Leftrightarrow yz \in L)$.

לבסוף, נשאר להוכיח כי $\sim_L$ טרנזיטיבי. אם נניח בשלילה שלא, אזי קיימות w_1, w_2, w_3 ב-Σ^* כך ש-$w_1 \sim_L w_2$ וגם $w_2 \sim_L w_3$ אבל $w_1 \not\sim_L w_3$.

כלומר, קיים $z \in \Sigma^*$ כך שבלי הגבלת הכלליות $w_1 z \in L$ אבל $w_3 z \notin L$. אמנם, מכך ש-$w_2 \sim_L w_3$, עולה כי $w_2 z \notin L$, ומכאן ש-$w_1 \sim_L w_2$ נדע כי $w_1 z \notin L$, בסתירה.

אבחנה

יחס שקילות מחלק את "העולם" למחלקות שקילות זרות. כלומר, קבוצת איברים שמקיימות את היחס, יוצרות חלוקה של L.

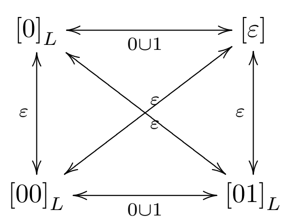
נסמן ב-$[w]_L$ את מחלקת השקילות של w ב-L, ונוכל לקחת נציג ספציפי מקבוצת מחלקת השקילות.

דוגמה

עבור השפה $(0 \cup 1)^* \cdot 0 \cdot (0 \cup 1)$(שפה שבה יש 0 במקום הלפני אחרון), מחלקות השקילות הן:

$S_1 = \varepsilon \cup 1 \cup \Sigma^* 11,\ S_2 = 0 \cup \Sigma^* 10,\ S_3 = \Sigma^* \cdot 00,\ S_4 = \Sigma^* \cdot 01$.

עלינו להראות שיש זנב מפריד בין כל אחת מבין הקבוצות, באמצעות נציג ממחלקת השקילות:



במקרה הקודם, היה מספר סופי של מחלקות שקילות. אמנם, ייתכן שיהיו אינסוף מחלקות שקילות.

למשל, ניקח את השפה $L = \{0^n 1^n \mid n \geq 0\}$. אם נביט במחלקת השקילות של היחס $\sim_L$, נקבל למעשה כי לכל $i < j$ מתקיים כי $0^i \not\sim_L 0^j$.

נשים לב כי המילה 1^i היא **זנב מפריד** בין $0^j, 0^i$, כי $0^i 1^i \in L$, אבל $0^j 1^i \notin L$. כיוון שיש אינסוף זוגות i, j כאלה, עולה כי יש ל-$\sim_L$ אינסוף מחלקות שקילות מייהל-נרוד.

מסתבר שזה לא במקרה, ומכך נגיע למשפט הבא.

משפט מייהל-נרוד

תהא $L \subseteq L^*$ שפה, אזי L רגולרית אם ורק אם יש ב-$\sim_L$ מספר סופי של מחלקות שקילות מייהל-נרוד.[5]

הוכחה

בכיוון הראשון $\Leftarrow$:

נניח כי L רגולרית, ונביט ב-DFA שמוגדר על ידי $A = \langle Q, \Sigma, \delta, q_0, F \rangle$, כך ש- $L(A)$.

[5] האינטואיציה תבוא בהמשך.

נגדיר יחס שקילות חדש $\sim_A \subseteq \Sigma^* \times \Sigma^*$ כך: לכל $x, y \in \Sigma^*$ מתקיים כי $x \sim_A y \Leftrightarrow \delta^*(q_0, x) = \delta^*(q_0, y)$.
כלומר, $x \sim_A y$ אם"ם x ו-y מגיעים **לאותו מצב** בריצה של A מ-q_0.
כעת, נרצה לטעון כי אם $x \sim_A y$ אזי גם $x \sim_L y$. ברגע שנוכיח זאת נסיים את ההוכחה, שהרי מספר מחלקות השקילות של $\sim_L$ הוא לכל היותר מספר מחלקות השקילות של $\sim_A$ וזה מספר סופי כי $|Q|$ סופי.
נבחין כי לכל $z \in \Sigma^*$ עולה כי:

$$\delta^*(q_0, xz) = \delta^*(\delta^*(q_0, x), z) = \delta^*(\delta^*(q_0, y), z) = \delta^*(q_0, yz)$$

כלומר הוכחנו את הנדרש.

בכיוון השני $\Leftarrow$:
נניח כי יש ל-L מספר סופי של מחלקות מייהל-נרוד ונראה כי L רגולרית.
נבנה ל-L DFA כלדהלן: $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ שמוגדר על ידי:

- Q - המצבים יהיו כל מחלקות השקילות MN של L.
- q_0 יהיה $[\varepsilon]$ - מחלקת השקילות של ε.
- δ יאובחן על ידי: לכל מחלקת שקילות $[w]$ ואות $\sigma \in \Sigma^*$ יתקיים כי $\delta([w], \sigma) = [w\sigma]$.
  נשים לב כי δ לא תלויה בנציג, שהרי אם $x \sim_L y$, עולה כי גם $x\sigma \sim_L y\sigma$, לכל $\sigma \in \Sigma^*$ (אם בשלילה יש z כי $xaz \in L$ ו-$yaz \notin L$, אז az זנב מפריד בין x ל-y).
- F יהיה $F = \{[w] \mid w \in L\}$.

נרצה לטעון כי $L(A) = L$. דבר זה נובע מכך ש-$\delta(q_0, w) = [w]$.
נוכיח זאת באינדוקציה על $|w|$:
<u>בסיס האינדוקציה</u>
עבור $|w| = 0$, עולה כי $w = \varepsilon$ ולכן $\delta(q_0, \varepsilon) = [\varepsilon]$.
<u>צעד האינדוקציה</u>
נניח כי הטענה נכונה לכל מילה w באורך שקטן שווה מ-n ונוכיח על $n+1$..
ניקח w כך ש-$|w| = n+1$, נסמן $w = w'\sigma$ כך ש-$|w'| \leq n$ ונקבל, מההגדרה כי:

$$\delta^*(q_0, w) = \delta^*(q_0, w'\sigma) \overset{\overbrace{\text{הגדרה}}^{}}{\underset{}{\overset{\downarrow}{=}}}$$

$$\delta^*(\delta^*(q_0, w'), \sigma) \overset{\overbrace{\text{הנחה}}^{}}{\overset{\downarrow}{=}}$$

$$\delta([w']_L, \sigma) \overset{\overbrace{\text{הגדרה}}^{}}{\overset{\downarrow}{=}}$$

$$[w'\sigma]_L$$

כלומר, מכאן עולה כי לפי הגדרת F, בהכרח $L(A) = A$, כנדרש.

1.4.3 מציאת אוטומט דטרמיניסטי מינימלי

הרצאה מס' 7:

יום שני

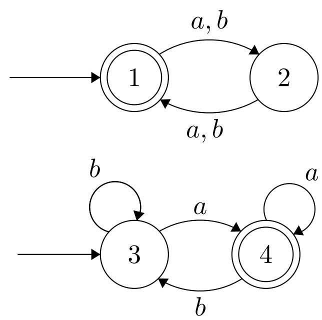
01.11.21

מוטיבציה

ניקח את $\Sigma=\{a,b\}$ ואת L: כל המילים באורך זוגי שמסתיימות ב-a.

נבחין כי מדובר בשפה רגולרית, שכן ניתן להתבונן על L בתור $L=L_1\cap L_2$ כך ש-L_1 אלו השפה של המילים באורך זוגי, ו-L_2 הן המילים שמסתיימות ב-a, וכפי שאנחנו יודעים שפות רגולריות סגורות לחיתוך.

שתי השפות הללו מיוצגות בתור:



נתבונן במספר דוגמאות לבדוק אילו מילים נמצאות ביחס מייהל-נרוד:

- $ab\not\sim_L aa$ כי אם ε זנב מפריד: $ab\varepsilon\notin L$ ואילו $aa\varepsilon\in L$.
- $b\sim_L a$ כי לכל $z\in\Sigma$ יתקיים כי $az\in L$ אם"ם $|z|$ אי זוגי ו-z מסתיימים ב-a ולכן $bz\in L$.
  מכאן עולה כי ברגע שמדובר במילה אי זוגית, אין משמעות ל-a.

האלגוריתם שלנו יבדוק למעשה את הדבר האחרון - אילו מצבים 'מיותרים' וניתן לאחד ביניהם. כלומר, נרצה למצוא את אוסף המצבים המינימלי.

אלגוריתם למזעור אוטומט דטרמיניסטי

נתון $A=\langle Q,\Sigma,q_0,\delta,F\rangle$. נגדיר סדרה $=_i\subseteq Q\times X$ של יחסים מעל Q כך ש-$1\le i$.

נרצה למעשה כי $q=_i q'\Leftrightarrow$ לכל מילה w, אם $|w|\ge i$ אזי $\delta^*(q',w)\in F\Leftrightarrow\delta^*(q,w)\in F$ - הם ביחס אם סיפא של מילה תגיע משניהם למצב מקבל.

נגדיר זאת אינדוקטיבית:

$q=_0 q'$ אם"ם $(q\in F\Leftrightarrow q'\in F)$. למקרה זה יש שתי מחלקות שקילות למעשה, אם $q\in F$ ואם $q\in Q\setminus F$.

$q=_{i+1}q'$ אם"ם $q=_i q'$ וגם לכל $\sigma\in\Sigma$ יתקיים כי $\delta(q,\sigma)=_i\delta(q',\sigma)$.

טענה

נרצה להוכיח כי $q=_i q'$ אם"ם לכל w, אם $|w|\le i$, אזי $\delta^*(q,w)\in F\Leftrightarrow\delta^*(q',w)$.

הוכחה

באינדוקציה על i.

<u>בסיס האינדוקציה</u>

עבור $i=0$, מתקיים כי $\delta^*(q,\varepsilon)=q\in F\Leftrightarrow\delta^*(q',\varepsilon)=q'\in F$.

<u>צעד האינדוקציה</u>

נניח כי הטענה נכונה עבור כל $|w| \leq i$ ונוכיח עבור $i+1$.
כלומר כל שעלינו לבדוק הוא את ה'אות האחרונה'.
כעת, יהיו q ו-q' כך ש-$q =_i q'$. תהי w מילה באורך $i+1$.

- אם $|w| \leq i$ אזי מהנחת האינדוקציה $\delta^*(q,w) \in F \Leftrightarrow \delta^*(q',w) \in F$.
- אם $|w| = i+1$, אזי $w = \sigma x$ עבור x כך ש-$|x| = i$. במקרה זה יתקיים כי $\delta^*(q,\sigma x) = \delta^*(\delta(q,\sigma), x)$. כעת, מהגדרת $=_{i+1}$ מתקיים כי $\delta(q,\sigma) =_i \delta(q',\sigma)$ ולכן מההנחה בהכרח $\delta^*(q',\sigma x) = \delta^*(\delta(q',\sigma), x)$, כנדרש.

מתי האלגוריתם יעצור? נרצה לטעון כי לסדרה $=_i$ יש נקודת שבת, כלומר נקודה בה $=_i$ זהה ל-$=_{i+1}$. כלומר, קיים i שיותר ממנו לא נפצל יותר.

טענה

לסדרה $=_i$ יש נקודת שבת.

הוכחה

בכל איטרציה שאיננה נקודת שבת, מתפצלת לפחות מחלקה אחת. ה'חסם העליון' הוא מספר המצבים, שהרי לא ניתן לפצל מצב בפני עצמו.

סמנטיקה

נאמר כי $q =_A q'$ אם ורק אם $q =_i q'$ עבור האיטרציה i שבה הגענו לנקודת שבת.

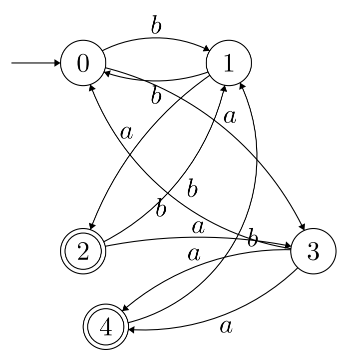
טענה - אוטומט מינימלי

יהיו $Q' = \{S_1, \ldots, S_n\}$. האוטומט המינימלי A' יוגדר על ידי $A' = \langle Q', \Sigma, [q_0], \delta', F' \rangle$, כאשר:

- Q' - מחלקות השקילות של $=_A$.
- δ' - $\delta'([q], a) = [\delta(q,a)]$.
- F' - $\{[q] \mid q \in F\}$.

דוגמה

השפה שראינו מקודם הינה:

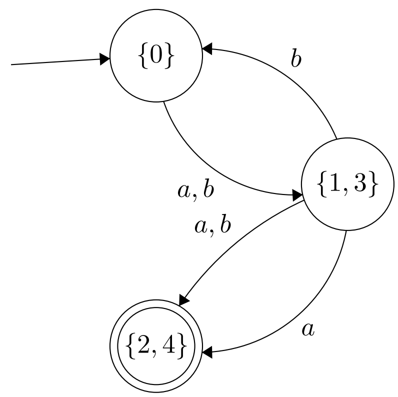


נתבונן בריצת האלגוריתם:

1. בתחילה, מתקיים כי $=_0$ הינו $\{\{2,4\}, \{0,1,3\}\}$, כי הרי החלוקה היא למצבים המקבלים והמצבים הלא מקבלים.

2. נבדוק האם $1 =_1 3$. אכן, מתקיים כי $\delta(1,a) =_0 \delta(3,a)$ שהרי $2 =_0 4$. כמו כן, מתקיים כי $\delta(1,b) =_0$ $\delta(3,b)$ ו-$0 =_0 0$.
   מאידך, $0 \neq_0 3$ כי הרי $\delta(0,a) \neq_0 \delta(3,a)$ כי $3 \neq_0 4$.
   לכן נקבל בסך הכל כי $=_1$ הינו $\{\{2,4\},\{0\},\{1,3\}\}$.
3. מתברר כי $=_2 = =_1$.

ואכן, האוטומט המינימלי יהיה:



1.5 שפות חסרות הקשר

נתחיל קודם כל מדוגמה.

דוגמה

ניקח דקדוק שיוגדר על ידי: $A \to 0A1,\ A \to B,\ B \to \#$.
מסתבר שהדקדוק הזה מגדיר שפה מעל הא"ב $\Sigma = \{0,1,\#\}$.
כי למשל, יתקיים:

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

במצב האחרון קיבלנו רק אותיות שנמצאות בא"ב ללא משתנים, ואז למעשה קיבלנו מילה שנמצאת בשפה.
כך נגדיר למעשה את ה"דקדוק", כעת בצורה פורמלית.

הגדרה

דקדוק חסר הקשר (להלן ח"ה) G מוגדר על ידי $G = \langle V, \Sigma, R, S \rangle$ כאשר:

- V משתנים.
- Σ א"ב.
- R הן חוקי גזירה מהצורה $V \to (V \cup \Sigma)^*$.
- $S \in V$ משתנה התחלתי.

הגדרה

אם $w,u,v\in(V\cup\Sigma)^*$ ו-$A\to w$ חוק בדקדוק, אזי נאמר ש-$uAv\Rightarrow uwv$ **מייצר** את w.

אם $u,v\in(V\cup\Sigma)^*$ נאמר כי $u\overset{*}{\Rightarrow}v$ אם יש סדרה $u=u_1\Rightarrow u_2\Rightarrow u_3\Rightarrow\ldots\Rightarrow u_k=V$ ו-$1\le k$.

השפה של G היא: $L(G)=\left\{w\in\Sigma^*\mid S\overset{*}{\Rightarrow}w\right\}$.

דוגמה

נתבונן ב-$G=\langle\{S,A\},\{a,b\},R,S\rangle$ כאשר R מוגדר על ידי $S\to AaA$ ו-$A\to\varepsilon\mid aA\mid bA$.

מה השפה של G? כל המילים שיש בהם a, כלומר למעשה $L(G)=(a+b)^*a(a+b)^*$.

הגדרה

דקדוק **תלוי** הקשר הוא דקדוק בו, בהינתן חוק $a\to b$, ייתכן כי $a\in V^*$.

כהערה, נאמר כי השפה שראינו מקודם היא $L(G)=\{0^n\#1^n\mid n\ge 0\}$, כי מספר האפסים שמוסיפים שווה למספר האחדות.

דוגמה נוספת

הרצאה מס' 8:

יום רביעי

03.11.21

ניקח את הדקדוק חסר ההקשר שמוגדר על ידי $S\to 0S0\mid 1S1\mid\varepsilon$.

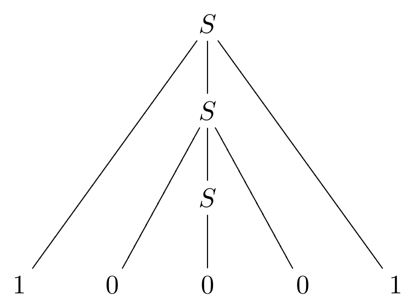
מדובר בשפה של 'פלינדרום באורך זוגי' מעל $\Sigma=\{0,1\}$, כי למשל מתקיים: $S\to 0S0\to 01\varepsilon 10\to 0110$.

אם היינו רוצים פלינדרומים באורך כלשהו, היה עלינו להוסיף כי $S\to 0\mid 1$.

בהתבסס על הדוגמה האחרונה, נציג אופן נוסף שבו אפשר 'לגזור' מילים בשפה.

עצי גזירה ועיבוד שפות טבעיות

נניח ונרצה לגזור את המילה 10101 באמצעות השפה שראינו בדוגמה האחרונה. נקבל את העץ הבא:



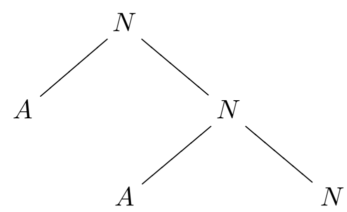
דבר זה נקרא עץ גזירה.

המוטיבציה להשתמש בו קשורה לעיבוד שפות טבעיות, שבעצמן קשורות באופן הדוק לשפות חסרות הקשר.

כי למשל אם ניקח את הדקדוק בשפה האנגלית, נוכל להגדיר אותו בתור

Noun $\to$ Noun Adj
Adj $\to$ big|red
Noun $\to$ big|cat

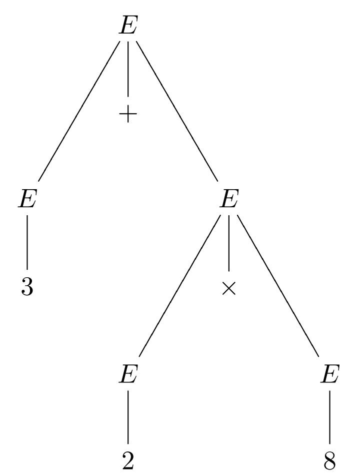
ואז אם נרצה לגזור את המשפט big red dog, אפשר להסתכל על זה בתור עץ גזירה:



נעיר כי שפות טבעיות אינן רגולריות מהסיבה הפשוטה שיש צורך ב'זיכרון', ולעיתים ייתכן ריבוי משמעות.

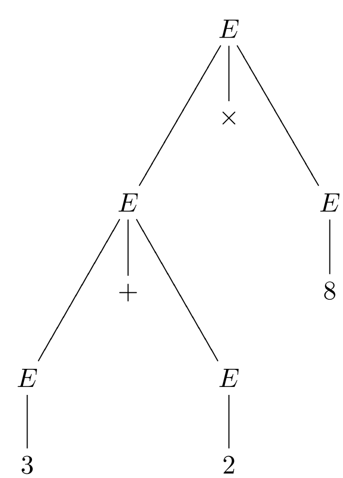
ריבוי משמעות

למשל, אם ניקח את השפה $E \to E \times E \mid E + E$, ונרצה לגזור את הביטוי $3 + 2 \times 8$, ייתכנו שני ביטויים:



וביטוי זה שקול ל-$3 + (2 \times 8)$.

מאידך, נוכל גם לקחת את הביטוי הזה:



כשביטוי זה שקול ל-$8 \times (2+3)$.

עוד דוגמה לדקדוק

ניקח את הדקדוק שמוגדר על ידי $S \to aSb \mid SS \mid \varepsilon$. אילו מילים בשפה?
$aabb, abab, s \to SS \to \ldots \in L$ אבל $abba \notin l$. כלומר, מדובר בשפת הסוגריים המקוננים באופן חוקי (באמצעות קידוד של a ו-b עם סוגריים מתאימים).
נעיר כי מדובר בשפה לא רגולרית. מספיק להוכיח באמצעות תכונות סגור, כי הרי החיתוך של שפה זו עם a^*b^* הוא השפה a^nb^n.

משפט

$\text{REG} \subseteq \text{CFL}$ - השפות הרגולריות כולן שפות חסרות הקשר.

הוכחה

בהינתן DFA שמוגדר על ידי A, נגדיר דקדוק חסר הקשר G כך ש-$L(G) = L(A)$.
יהי $A = \langle Q, \Sigma, q_0, \delta, F\rangle$ ונגדיר $G = \langle V, \Sigma, R, S\rangle$ על ידי $V = \{V_q \mid q \in Q\}$ (גוזר מלים שמתקבלות ממצב q).
כמו כן $S = V_{q_0}$ ו-R הוא $V_q \to aV_{q'}$: לכל מעבר $\delta(q,a) = q'$ נוסיף את החוק $V_q \to aV_{q'}$. בנוסף, אם $q \in F$ אזי $V_q \to \varepsilon$.
נקבל כי אם $r = q_0, \ldots, q_m$ ריצה של A על $w = \sigma_1, \ldots, \sigma_m$, נקבל כי $V_{q_0} \Rightarrow \sigma_1 V_{q_1} \Rightarrow \sigma_1\sigma_2 V_{q_2} \Rightarrow \ldots \Rightarrow$ $\sigma_1\sigma_2\ldots\sigma_m$.
מעבר לזה, מדובר בדקדוק לינארי ימיני.

הגדרה

דקדוק **ליניארי ימני** הוא דקדוק שכל החוקים שלו מהצורה $A \to aB$.

1.5.1 למת הניפוח לשפות חסרות הקשר ואוטומט מחסנית

אם נרצה לראות דוגמה לשפה שאינה חסרת הקשר, נוכל לקחת את השפה $L = \{a^nb^nc^n \mid n \geq 0\}$. האינטואיציה היא שיש צורך ב'שלושה מונים', ואילו בשפות חסרות הקשר יש שני מונים (ובאוטומטים מונה אחד).
נרצה להוכיח כי השפה $L = \{ww \mid w \in \Sigma^*\}$ אינה חסרת הקשר. נוכל לעשות זאת באמצעות למת הניפוח לשפות חסרות הקשר.

למת הניפוח לשפות חסרות הקשר

אם L ח"ה, אזי קיים $1 \leq p$, כך שלכל מילה w אם $|w| \geq p$ יש חלוקה $w = uvxyz$ וכך ש:

1. $|vxy| \leq p$.
2. $|vy| > 0$.
3. $uv^ixy^iz \in L$ לכל $0 \leq i$.

אוטומט מחסניות

עד כה עסקנו באוטומטים רגילים, שלהם כאמור אין זיכרון.
אוטומט המחסנית עובד אחרת - יש לו אפשרות למצוא את האיבר האחרון שהוכנס.
לאוטומט כזה יש אפשרות לאבחן את השפה $\{0^n1^n \mid n \geq 0\}$ - בקוראו 0, דוחף a למחסנית, בקוראו 1, מוציא a מהמחסנית. אם הגיע לסוף המילה וגם המחסנית ריקה, סימן שהמילה בשפה.

2 תורת החישוביות

הרצאה מס' 9:

יום שני

08.11.21

הקדמה

במהלך חלק זה, נרצה קודם כל לבדוק 'מה מחשבים יכולים לעשות'. לשם כך, נרצה 'להתקרב' למחשב קצת יותר.

2.1 מכונת טיורינג

המודל של מכונת טיורינג מאפשר לנו לבדוק מה היכולת של מחשבים.
בדומה לאוטומט, קיים במכונת רצף של אותיות, אלא שבשונה מאוטומט, אפשר להכניס רווחים, וגם מדובר בסרט אינסופי (לאחר האותיות מופיעים אינסוף רווחים).
בנוסף, ישנה יכולת לכתוב על הסרט, ולא רק לקרוא כמו באוטומט, וגם קיימת אפשרות לתזוזה עם ה'ראש הקורא' שמאלה וימינה.

		$\Rsh$	q				
a	b	b	a	$\sqcup$	$\sqcup$	$\sqcup$	$\ldots$

כדי לאפיין מתי 'נגמרת' המילה, נגדיר מצבי קבלה ודחייה, שיעזרו לנו.

דוגמה

ניקח למשל את השפה הבאה - $L = \left\{ w\#w \mid w \in (0+1)^* \right\}$ - מדובר בשפה לא רגולרית (ואפילו לא חסרת הקשר), אך בכל זאת תוכנית מחשב יכולה לבצע זאת.
למשל, מכונת טיורינג עם הסרט הבא:

$\Rsh$	q								
0	0	1	1	#	0	1	1	0	$\sqcup$

ואז נגדיר את האלגוריתם הבא:

אלגוריתם 1 אלגוריתם למציאת $w\#w$

1. סרוק את הסרט וודא שיש לפחות # אחת.
2. כל עוד יש אותיות לא מחוקות משמאל ל-# הראשונה:
   (א) זגזג בין מיקומים תואמים לפני ואחרי ה-# הראשונה, וודא שמסומנים באותה אות.
   (ב) אם מצאנו חוסר התאמה: דחה.
3. אם נותרו אותיות לא מחוקות מימין ל-#: דחה.
4. אחרת: קבל.

שימו לב שעדיין לא מדובר בהגדרה פורמלית מלאה ונראה הגדרה מלאה בהמשך.

הגדרה

מכונת טיורינג (להלן - מ"ט). היא שביעייה $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}} \rangle$ כאשר:

- Q קבוצת מצבים סופית.
- Σ א"ב הקלט, כך ש-$\sqcup \notin \Sigma$.
- Γ א"ב העבודה, כך ש-$\sqcup \in \Gamma$ וכך ש-$\Sigma \subseteq \Gamma$.
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ מוגדרת על ידי $\delta(q, a) = \langle q', b, R \rangle$ - כש-M במצב q, הראש הקורא מצביע על תא שכתוב בו a, אזי M עובר למצב q', כותבת b במקום a וזזה עם הראש הקורא תא אחד ימינה.
- $q_0 \in Q$ מצב התחלתי.
- $q_{\text{acc}} \in Q$ מצב מקבל.
- $q_{\text{rej}} \in Q$ מצב דוחה.

הגדרה

קונפיגורציה של מכונת טיורינג מוגדרת על ידי **המצב**, **תוכן הסרט ומיקום הראש הקורא**.

נתאר קונפיגורציה כך: עבור $v, u \in \Gamma^*$ ומצב $q \in Q$, נקבל vqu, קונפיגורציה שבה המצב הוא q, תוכן הסרט $v \cdot u$ והראש מצביע על האות הראשונה ב-u:

$$
\begin{array}{cccccc}
 & & & q & & \\
 & & & \downarrow & & \\
\hline
\multicolumn{1}{|c|}{a} & \multicolumn{1}{c|}{b} & \multicolumn{1}{c|}{\ldots} & \multicolumn{1}{c|}{b} & \multicolumn{1}{c|}{c} & \multicolumn{1}{c|}{\ldots} \\
\hline
 & v & & & u &
\end{array}
$$

הקונפיגורציה ההתחלית של M על מילה $w \in \Sigma^*$ היא $q_0 w$.

בהינתן $a, b, c \in \Gamma$ ו-$u, v \in \Gamma^*$ ו-$q, q' \in Q$, והקונפיגורציה ההתחלתית היא $uaqbv$ - כלומר מצביעים על b:

$$
\begin{array}{cccc}
 & & q & \\
 & & \downarrow & \\
\hline
\multicolumn{1}{|c|}{u} & \multicolumn{1}{c|}{a} & \multicolumn{1}{c|}{b} & \multicolumn{1}{c|}{v} \\
\hline
\end{array}
$$

אם $q \in \{q_{\text{acc}}, q_{\text{rej}}\}$ (מצב עצירה), אין קונפיגורציה עוקבת והריצה מסתיימת. אחרת, אם $\delta(q, b) = (q', c, L)$ (לך ל-q, שים שם c ולך אחד שמאלה ל-q') אזי $uaqbv \to uq'acv$:

$$
\begin{array}{cccc}
 & q' & & \\
 & \downarrow & & \\
\hline
\multicolumn{1}{|c|}{u} & \multicolumn{1}{c|}{a} & \multicolumn{1}{c|}{c} & \multicolumn{1}{c|}{v} \\
\hline
\end{array}
$$

ואם $\delta(q, b) = (q', c, R)$ אזי $uaqbv \to uacq'v$, בצורה דומה.

כשהקונפיגורציה היא qav ו-$\delta(q, a) = \delta(q', b, L)$ אז הקוניפגורציה העוקבת היא $\delta(q, b)$ (לא נופלים מהסרט שמאלה).

הגדרה
ריצה של M על מילה $w = w_1 w_2, \ldots w_n \in \Sigma^*$ היא סדרה $r = c_0, c_1, c_2, \ldots, c_m$ של קונפ', כך ש-$c_0 = q_0 w$ הקונפ' ההתחלתית של M על w, ולכל $0 \leq i \leq M$ מתקיים כי c_{i+1} עוקבת ל-c_i, ו-c_m היא **קונפ' עוצרת** (המצב שלה הוא q_{acc} או q_{rej}).
השפה של M מוגדרת על ידי $L(M) = \{w \mid w$ יש ריצה מקבלת של M על $w\}$.

נשים לב כי יש שלושה גורלות לריצה של M על w:

1. עוצרת ומקבלת.
2. עוצרת ודוחה.
3. לא עוצרת.

דבר זה מביא אותנו שוב לבעיית העצירה (שניפגש בה שוב עוד בהמשך). בבעייה זו, הקלט הוא תכנית מחשב P וקלט X ל-P ונרצה לבדוק האם P עוצרת על X.

דוגמה
מ"ט M עבור $L = \left\{ w\#w \mid w \in (0+1)^* \right\}$ ו-M שיוגדר על ידי:

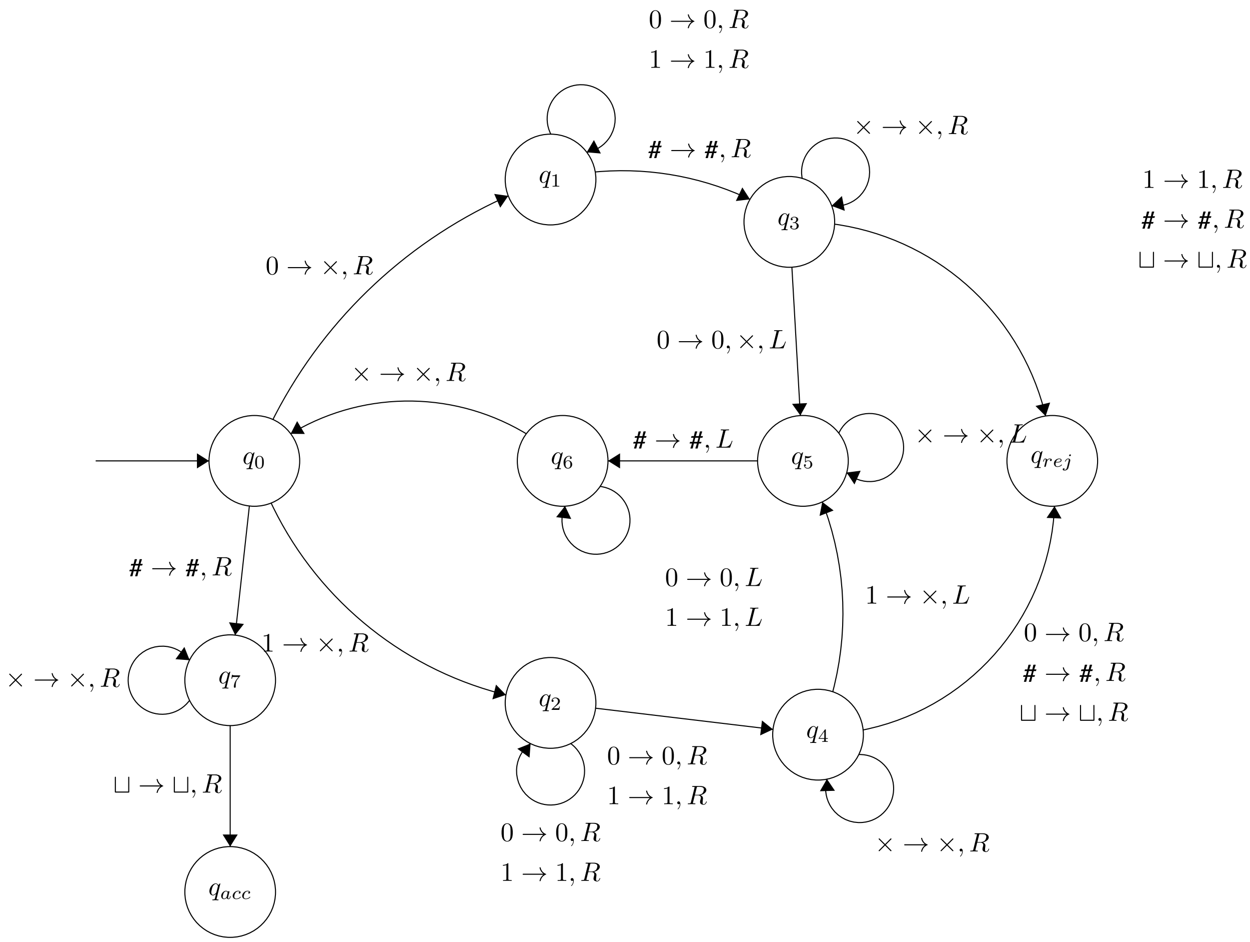
$$M = \langle Q, \Sigma = \{0, 1, \#\}, \Gamma = \{0, 1, \#, x, \sqcup\}, q_0, \delta, q_{\text{acc}}, q_{\text{rej}} \rangle$$

כיצד נוכל לתאר את האלגוריתם דלעיל בתור פעולות של מכונת טיורינג?

אלגוריתם 2 מכונת טיורינג למציאת $w\#w$

1. אם התא הנוכחי מסומן ב-#, בדוק האם יש 0 או 1 מימינו.
   (א) אם אין: קבל.
   (ב) אחרת: דחה.
2. אם התא הנוכחי מסומן ב-0, לך ימינה עד ל-#, ואחר כך ימינה עד לתא הלא מחוק הראשון.
   (א) אם מסומן ב-$1, \sqcup, \#$: דחה.
   (ב) אחרת: מחוק אותו ועבור ל-4.
3. כמו 2, כשהתא הנוכחי מסומן ב-1 (מצפים לראות 1 אחרי #).
4. לך שמאלה (על מחוקים) עד ל-# ואז לך שמאלה עד המחוק הימני לפני ה-# ואז חזור תא אחד ימינה. חזור ל-1.

כעת, נצייר זאת:



הגדרה

נאמר שמ"ט M **מזהה** את השפה $L \subseteq \Sigma^*$ אם $L(M) = L$.

נאמר ששפה היא 'ניתנת למנייה רקורסיבית', כלומר ב-RE, אם קיימת מ"ט M שמזהה את L.

נאמר שמ"ט M **מכריעה** את השפה $L \subseteq \Sigma^*$ את השפה $L \subseteq \Sigma^*$ אם M מזהה את L ובנוסף M עוצרת על כל קלט.

נאמר ששפה היא רקורסיבית כלומר, ב-R, אם קיימת מ"ט שמכריעה אותה.

נבחין כי מההגדרה R⊆RE אבל ההפך לא בהכרח נכון.

המחלקה co-RE הינה המשלים של RE.

מההגדרה עולה כי אם $L \in$ co-RE אזי $\overline{L} \in$ RE, כלומר יש מ"ט M עבור $\overline{L}$ כזו שלכל $w \in \Sigma^*$, אם $w \notin L$ אזי M עוצרת ומקבלת, ואם $w \in L$, אזי M דוחה או לא עוצרת.

הרצאה מס' 10: יום רביעי 10.11.21

משפט

$\mathrm{R} = \text{co-RE} \cap \mathrm{RE}$.

הוכחה

נוכיח זאת באמצעות הכלה דו כיוונית.
בכיוון הראשון, נוכיח כי $\mathrm{R} \subseteq \mathrm{RE} \cap \text{co-RE}$.
תהי $L \in \mathrm{R}$. מההגדרה בהכרח כי $L \in \mathrm{RE}$. נשאר לנו להוכיח כי $L \in \text{co-RE}$. נוכל לעשות זאת אם נראה כי $\overline{L} \in \mathrm{R} \subseteq \mathrm{RE}$.

למה

$\overline{L} \in \mathrm{R}$.

הוכחה

בהינתן מכונה M שמכריעה את L, אז המכונה $\tilde{M}$ שמתקבלת מ-M על ידי החלפת q_{acc} ו-q_{rej}, מכריעה את $\overline{L}$ (אם יודעים רק ש-M מזהה את L, אז ההחלפה הזאת לא בהכרח תעזור!).

בכיוון השני, נרצה להראות כי $\mathrm{RE} \cap \text{co-RE} \subseteq \mathrm{R}$.
כלומר, עלינו להוכיח כי אם $L \in \mathrm{RE}$ וגם $L \in \text{co-RE}$ אזי $L \in \mathrm{R}$.
בהינתן מכונה M שמזהה את L (יש כזו כי $L \in \mathrm{RE}$) ומכונה $\overline{M}$ שמזהה את $\overline{L}$ (יש כזו, מסיבה הפוכה).
נבנה כעת מכונה M' שמכריעה את L. M' תפעל כך (הרצה במקביל):
עבור $1 \leq i$:

1. הרץ את M על w i צעדים, אם קיבלה, עצור וקבל.
2. הרץ את $\overline{M}$ על w i צעדים, אם קיבלה, עצור ודחה.

נבחין כי M' בטוח עוצרת: $w \in L$ או $w \notin L$. מתקיים כי M עוצרת על w או $\overline{M}$ עוצרת על w. ולכן בהכרח קיים i כך ש-M עוצרת על w אחרי i צעדים, או $\overline{M}$ עוצרת על w אחרי i צעדים.
כמו כן, M' מזהה את L , שהרי היא עוצרת ומקבלת בדיוק את כל המילים ב-L.
אם כך, בהכרח M' מכריעה את L, כפי שרצינו לבנות.

2.1.1 ספרן (Enumerator)

הגדרה

מודל **הספרן** (**Enumerator**), מורכב ממ"ט ללא קלט, עם מדפסת, שמדפיסה מילים ב-Σ^*.
השפה של הספרן E היא מילים שמודפסות בסופו של דבר (ייתכן שמילה תודפס מספר - אפילו ∞- פעמים).

משפט

$L \in \mathrm{RE}$ אם ורק אם יש ספרן E כזה ש-$L = L(E)$.

הוכחה

נבנה הוכחה באמצעות הכלה דו כיוונית. בכיוון הראשון, נניח כי יש ספרן E עבור L ונבנה מ"ט שמזהה את L.
המ"ט M תפעל כך: בהינתן מילה w, M מריצה את E. כל פעם ש-E מדפיסה מילה x, M בודקת האם $w = x$.
אם כן, עוצרת ומקבלת (את w). אם לא, ממשיכה להריץ את E.
אם $w \in L$, אזי w תודפס ע"י E ולכן M תעצור ותקבל את w.
אם $w \notin L$, אזי M לא תקבל את w (תתקע, בהנחה ש-E נתקעת).
בכיוון השני, בהינתן מ"ט M שמזהה את L, נבנה ספרן E עבור L.
תהי $w_1, w_2, w_3, \ldots$ סדרה של כל המילים ב-Σ^* (יש כזה, כי Σ היא בת מנייה).
נוכל להציע כמה פתרונות רעים:

נריץ את m על w_1, אם קיבלה, נדפיס את w_1. אם לא, נעבור ל-w_2. מדובר בפיתרון רע[6] שכן ייתכן כי w_2 לא תעצור.
פתרון רע נוסף הוא לרוץ צעד 1 על $w_1, \ldots,$ ולעבור לצעד הבא. מדובר בפתרון רע שכן זו סדרה אינסופית ולא נגיע ל-2.

הפתרון הנכון הוא ליצור מונה $1 \leq i$ ובאיטרציה ה-i לרוץ על $w_1, \ldots, w_i$ i צעדים. אם M קיבלה את w_j, נדפיס את w_j (למעשה, אנחנו עוברים על כל המילים האפשריות בשפה).
נבחין כי אם $w \in L$, אזי יש איטרציה i כך שנרוץ על w את מספר הצעדים הדרוש לריצה המקבלת ולכן למעשה w תודפס (למעשה תודפס ∞ פעמים).
מצד שני, אם $w \notin L$, אזי w לא תודפס.

הרצאה מס' 11:

יום שני 15.11.21

2.1.2 קידוד אלגוריתמים ואוטומטים

הבעייה ה-10 של הילברט
בשנת 1900, הילברט ניסה לתאר אלגוריתם, כך שבהינתן פולינום במספר משתנים, יכריע האם יש לו שורש שלם. הוא ניסה לחפש תהליך שאותו ניתן להכריע אחרי מספר סופי של פעולות. למעשה, **בלתי אפשרי למצוא אלגוריתם כזה** ואת זה נוכיח בקרוב (ספויילר!).

התזה של צ'רץ' וטיורינג
צ'רץ' וטיורינג הוכיח למעשה כי **אלגוריתם** שקול **להכרעה על ידי מכונת טיורינג**. דבר זה יעזור לנו בהמשך.

ישנן שלוש רמות של תיאור אלגוריתם:

1. באמצעות מכונת טיורינג.
2. באמצעות תיאור הפעולה של מכונת טיורינג.
3. בפסאודו קוד (שפה עילית).

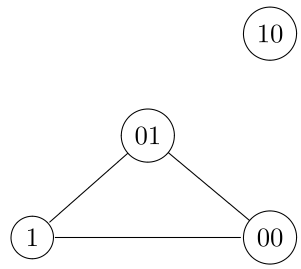
אנו יודעים כי אלגוריתמים פועלים על פולינומים, גרפים, מטריצות ועוד. מכונות טיורינג, לעומת זאת, מקבלות רק מילה ב-Σ^*. נרצה להיווכח שאכן ניתן לקודד כל אלגוריתם למכונת טיורינג:

- נסמן ב-$\langle A \rangle$ את הקידוד של A.
- המכונת טיורינג בודקת שהקידוד נכון.

למשל, יכולנו לקחת את הדוגמאות הללו:
$L = \{\langle G \rangle \mid$ גרף לא מכוון וקשיר $G\}$ או את $\overline{L} = \{\langle G \rangle \mid$ גרף לא מכוון ולא קשיר $G\}$ ולקודד אותן.
הרעיון של הקידוד הוא "תרגום" של מכונת הטיורינג לאיזשהו טקסט שמסמן את הפלט שהיא מוציאה.

דוגמאות לקידודים
נוכל לקחת את הגרף G הבא:

[6] "אל תכתבו את זה במבחן!" (א.ק)



ולקודד אותו באמצעות $\underbrace{v_1\#v_2\#v_3\#\dots\#v_n}_{\text{קודקודים}}\$\underbrace{v_{i1}\#v_{i2}\dots\$}_{\text{קשתות}}$.

הקידוד יהיה $\langle G\rangle \in (0+1+\#+\$)^*$ ו:

$$\underbrace{00\#01\#10\#11\#\$}_{\text{קודקודים}}\underbrace{00\#01\ \$\ 01\#11\ \ \$\ 00\#11\#\ \$}_{\text{קשתות}}$$

נבנה אלגוריתם כפי שאנחנו מכירים:

אלגוריתם 3 אלגוריתם לבדיקת קשירות

---

1. $C=\emptyset, T=\{v_0\}$.
2. כל עוד $T\neq\emptyset$:
   (א) $v=pop\,(T)$.
   (ב) $push\,(C,v)$.
   (ג) לכל $u\in V\setminus(C\cup T)$:
      i. אם $E\,(v,u)$, אז $push\,(T,u)$.
3. אם $C=V$, תקבל, אחרת תדחה.

נרצה כעת לבנות מכונת טיורינג מתאימה.
קודם כל נראה מהו א"ב העבודה - $\Gamma=\Sigma\cup\{0,1\}\times\{T,C,A\}$. כמו כן, יתקיים כי קודקוד $v\in T$ אם הביט הראשון שלו מסומן ב-T. (הכוונה ב-A היא active).
כעת נתאר את פעולת המכונה:

אלגוריתם 4 מכונת טיורינג למציאת קשירות

1. T-סמן את הקודקוד הראשון (= "בחר את v_0").
2. כל עוד יש קודקודים T-מסומנים (= "כל עוד $T \neq \emptyset$"):
   (א) A-סמן קודקוד T-מסומן.
   (ב) עבור על רשימת הקודקודים.
   אם יש קודקוד לא מסומן, בדוק האם יש קשת בינו לבין הקודקוד ה-A-מסומן. אם כן, T-סמן אותו.
   (ג) C- סמן את הקודקוד שהיה A-מסומן.
3. אם כל הקודקודים C-מסומנים - קבל. אחרת - דחה.

אפשר לקודד גם פולינומים, למשל $L = \{\langle p\rangle \mid$ יש ל-p שורש שלם$\}$.

בעיית הכרעה וחיפוש

עד כה דיברנו בעיקר על בעיות הכרעה. אמנם, מסתבר שאפשר להמיר בעיות חיפוש גם לרצף של בעיות הכרעה.

קידוד אוטומטים

נוכל לקדד אוטומט $A = \langle \Sigma, Q, q_0, \delta, F\rangle$ ו-$A_{\text{DFA}} \{\langle A, w\rangle \mid w \in L(A), A \text{ DFA}\}$.
ואז:

$$\sigma_1\#\sigma_2\#\ldots\#\sigma_n\$q_0\#q_1\#\ldots\#q_m\$q_0\#q_i\#\sigma_j\#q_l\$\ldots\$q_{i_1}\#\ldots\#q_{i_k}\$w\$$$

טענה

$A_{\text{DFA}} \in \mathrm{R}$.

הרעיון

מכונת טיורינג יכולה לסמלץ ריצה של A על w ולענות כמוה.

פרטי מימוש:

1. אתחול:
   (א) כתוב q_0 על סרט נוסף.
   (ב) כתוב את w על סרט נוסף.
2. כל עוד קריאת w לא הסתיימה:
   (א) מצא את המעבר שמתאים למצב שבסרט הנוסף והאות שהראש הקורא בסרט של w מצביע עליה.
   (ב) עדכן את המצב והראש הקורא על w.
3. קבל אם"ם המצב בסרט הנוסף שייך ל-F.

2.1.3 אי כריעות

טענה

לכל א"ב סופי Σ בגודל 2 או יותר יש שפה $L \notin \mathrm{R}$.

הוכחה

משיקולי ספירה. ראינו בעבר כי יש $2^{\aleph_0}$ שפות (ראו בתרגול 1).

מכיוון שניתן לקודד כל מ"ט מעל א"ב Σ על א"ב סופי $\Sigma \cup \{\#, \$\}$, אז יש לכל היותר $\aleph_0$ מ"ט. אם כך, יש יותר שפות ממכונות טיורינג ואם כך יש לכל היותר $\aleph_0$ שפות שניתנות להכרעה.

טענה

$A_{\text{TM}} = \{\langle M, w\rangle \mid w \in L(M)\text{-ו מ"ט M}\} \notin \text{R}$.

הוכחה

נראה ש-$A_{\text{TM}} \notin \text{R}$.

נשים לב כי $A_{\text{TM}} \in \text{RE}$. מ"ט M' שמזהה את A_{TM} פועלת כך: מסמלצת ריצה של M על w (למשל על ידי שמירה של הקונפ' הנוכחת בסרט נוסף). אם M מקבלת את w, אזי M' מקבלת את $\langle M, w\rangle$.

מצד שני, $A_{\text{TM}} \notin \text{R}$.

שלב I

נניח בשלילה שהיא כן שייכת. אזי יש מ"ט H כך שלכל קלט $\langle M, w\rangle$ H עוצרת ו:

$$H(\langle M, w\rangle) = \begin{cases} \text{acc} & M(w) = \text{acc} \\ \text{rej} & M(w) \neq \text{acc} \end{cases}$$

שלב II

נבנה מ-H מ"ט D שמקבלת בקלט רק מ"ט M ופועלת כך:

$$D(\langle M\rangle) = \begin{cases} \text{acc} & M\langle m\rangle = \text{acc} \\ \text{reg} & M\langle m\rangle \neq \text{acc} \end{cases}$$

יש מכונה כזאת, כי במקום w נכנס למכונה H את הקידוד של המכונה M על w.

שלב III

נבנה מ-D מ"ט $\widetilde{D}$ שמקבלת בקלט מ"ט M ופועלת כך:

$$\widetilde{D}(\langle M\rangle) = \begin{cases} \text{rej} & M(\langle M\rangle) = \text{acc} \\ \text{acc} & M(\langle M\rangle) \neq \text{acc} \end{cases}$$

$\widetilde{D}$ מתקבלת מ-D על ידי החלפת q_{acc} ו-q_{rej}.

אם M מזהה את L , אזי $\widetilde{M}$ לא בהכרח מזהה את L. מצד שני, אם M מכריעה את L, $\widetilde{M}$ מכריעה את $\overline{L}$.

שלב IV

נתבונן בריצה של $\widetilde{D}$ על $\left\langle \widetilde{D} \right\rangle$ (המקרה בו $\langle M\rangle$ שבקלט היא $\left\langle \widetilde{D} \right\rangle$):

$$\widetilde{D}\left(\left\langle \widetilde{D} \right\rangle\right) = \begin{cases} \text{rej} & \widetilde{D}\left(\left\langle \widetilde{D} \right\rangle\right) = \text{acc} \\ \text{acc} & \widetilde{D}\left(\left\langle \widetilde{D} \right\rangle\right) \neq \text{acc} \end{cases}$$

אם כך, קיבלנו סתירה!

הרצאה מס' 12:

יום רביעי

17.11.21

אבחנה

להוכחה מהסוג שעשינו כאן קוראים 'הוכחה בלכסון'. מדוע? אם ניזכר בהוכחה שעשינו באינפי 1, שאין סידור של $[0,1]$, נראה כי למעשה הסתכלנו על האלכסון והראינו שאין דרך להציג את המספר האלכסוני - אין תצוגה מתאימה. במידה מסוימת, גם ההוכחה שלנו כעת הייתה הוכחה מסוג זה, כי גם כאן אנחנו יכולים להסתכל על סידור של כל מכונות הטיורינג בעולם, כאשר בכניסה ה-(i,j) בטבלה נאמר האם M_i מקבלת את $\langle M_j\rangle$.

המכונה H ממלאת את הטבלה ב-rej והמכונה D היא למעשה האלכסון, כי היא מקבלת את M_i ומריצה אותו על $\langle M_i\rangle$. המכונה $\widetilde{D}$ לעומת זאת, מוציאה פשוט את ההפך מ-D.

ניתן לראות שלא ניתן למצוא סידור על האלכסון ל-$\widetilde{D}\left(\left\langle\widetilde{D}\right\rangle\right)$.

מצד אחר, אנחנו יודעים כי R=RE$\cap$coRE ומכאן נובע כי $\begin{matrix} A_{\text{TM}}\notin \text{R} \\ A_{\text{TM}}\in \text{RE}\end{matrix} \Rightarrow A_{\text{TM}}\notin \text{coRE}$ כלומר $\overline{A_{\text{TM}}}\notin \text{RE}$.

נתבונן במכונה $\text{HALT}_{\text{TM}}=\{\langle M,w\rangle \mid w \text{ עוצרת על } M\}$ - מ"ט שמריצה את M על w. אם M מגיעה ל-q_{acc} או ל-q_{rej}, עוצרת ומקבלת.

כיצד נוכל להוכיח זאת? באמצעות רדוקציה.

טענה

$\text{HALT}_{\text{TM}}\notin \text{R}$.

הוכחה

תהי M מ"ט שמכריעה את HALT_{TM}. נבנה בעזרתה מ"ט M' שמכריעה את A_{TM}.

בהינתן $\langle M,w\rangle$, המכונה M' פועלת כך: היא מריצה את M על $\langle M,w\rangle$. מובטח שתעצור. אם M דוחה, אז M' דוחה את $\langle M,w\rangle$ (כי יודעים ש-M לא עוצרת על w ולכן M לא מקבלת את w).

אם M מקבלת את $\langle M,w\rangle$, נריץ (ללא חשש) את M על w. מובטח שנעצור, ונענה כאן $M(w)$.

2.2 רדוקציית מיפוי

הגדרה

עבור א"ב Σ, נאמר ש-$f:\Sigma^*\to\Sigma^*$ **ניתנת לחישוב** (computable) אם קיימת מ"ט M_f שבהינתן קלט $x\in\Sigma^*$, עוצרת עם $f(x)$ על הסרט.

דוגמה

תהי $f:\mathbb{N}\times\mathbb{N}\to\mathbb{N}$ המוגדרת על ידי $f(x,y)=x+y$.

אם x ו-y נתונים באונרית $111\ldots1\#11\ldots1$ נוכל למחוק את הסולמית ולהזיז אחד שמאלה, ואז למעשה נקיים את הפונקציה, והמ"ט תעצור על $f(x)$, כפי שרצינו.

הגדרה
עבור א"ב Σ, ושתי שפות $A,B\subseteq\Sigma^*$, נאמר ש-A **ניתנת לרדוקציית מיפוי** ל-B $A\leq_m B$ ("A קלה יותר מ-B") אם קיימת פונקצייה ניתנת לחישוב $f:\Sigma^*\to\Sigma^*$ כך שלכל $x\in\Sigma^*$ יתקיים כי $x\in A\Leftrightarrow f(x)\in B$.

הרצאה מס' 13:

יום שני

22.11.21

משפט הרדוקציה
אם $A\leq_m B$ ו-$B\in\mathrm{R}$, אזי $A\in\mathrm{R}$.

הוכחה
תהי M_B מ"ט שמכריעה את B. תהי M_{R} מ"ט שמחשבת פונקציה קלטים ל-$A\to$קלטים ל-B : f שבזכותה $A\leq_m B$.
תהי M_A שמכריעה את A: בהינתן קלט w, מריצה את M_{R}, מריצה את M_{B} על $f(w)$ ועונה כמוה.

משפט הרדוקציה 2
אם $A\leq_m B$ ו-$A\notin\mathrm{R}$ אזי $B\notin\mathrm{R}$.

אינטואיציה

לפני שאנו ניגשים לעובי הקורה של רדוקציות מיפוי, מסובכות יותר ופחות, חשוב שנבין מה עלינו לעשות במהלך הרדוקציה.
מה יש בידינו? שתי שפות, A ו-B, שיש פונקציית מיפוי ביניהם, או שעלינו למצוא אותה. לפי ההגדרה, כפי שראינו, זה אומר שיש פונקציה $f:\Sigma^*\to\Sigma^*$, **שאם נפעיל את הפונקציה הזאת על ערכים מ-A, נקבל בהכרח ערכים ששייכים ל-B.**
הפונקציה הזאת **חשיבה** - יש מ"ט שעוצרת עם $f(x)$ **על כל קלט** x. העובדה שהפונקציה עוצרת על כל קלט מאפשרת לנו למעשה למפות ערכים מ-A לערכים ב-B. אם אכן נתון לנו ש-B מכריעה גם היא, כל שנותר לנו הוא להפעיל את f על קלטים מ-A (גם אם A לא עצרה לנו, אחרי ההפעלה היא תעצור), והופ, הכרענו את A. אם נרצה להשתמש במשפט הרדוקציה, עלינו פשוט למצוא פונקציית מיפוי כזאת (ולהוכיח שהיא חשיבה!) ולהשתמש בנתונים שיש לנו כבר על B.
אם נרצה **לשלול** תכונות של B, בהינתן תכונות של A, נבצע את הפעולה ההפוכה. אם למשל אנו יודעים כי A לא ניתנת להכרעה, אזי אם B הייתה ניתנת להכרעה, פשוט היינו לוקחים קלטים מ-A, מפעילים עליה את פונקציית המיפוי, ומכריעים את הקלטים.
חשוב תמיד לזכור מה הקלטים של A ומה הקלטים של B, וליצור פונקציה מתאימה על פי זה.

שלילת הכרעת $\mathrm{HALT_{TM}}$
נוכיח כי $A_{\mathrm{TM}}\leq_m \mathrm{HALT_{TM}}$ ונרצה להוכיח כי $\mathrm{HALT_{TM}}\notin\mathrm{R}$, באמצעות משפט הרדוקציה ההפוך.
אם ניקח קלטים ל-$A\to$קלטים ל-B : f כאשר $A=A_{\mathrm{TM}}$ ו-$B=\mathrm{HALT_{TM}}$ ו-f פונקציית הרדוקציה מוגדרת על ידי $f(\langle M,w\rangle)=\langle M',w'\rangle$. נשים לב כי $\langle M,w\rangle$ היא בתחום הקלטים ל-A_{TM} וכי $\langle M',w'\rangle$ שייכים לתחום הקלטים של $\mathrm{HALT_{TM}}$, שהרי מדובר בקשר שבין מכונה ומילה - האם המכונה מקבלת את המילה או לא.
<u>בנייה</u>
M' תתקבל מ-M על ידי החלפת המצב q_{rej} במצב חדש q_{loop} כזה ש-$\delta(q_{\mathrm{loop}},\sigma)=\langle q_{\mathrm{loop}},\sigma,R\rangle$, כמו כן, $w'=w$.

<u>נכונות</u>

א. תחילה, נוכיח כי f חשיבה. בהכרח קיימת מ"ט M_{R} שמחשבת את f: היא עוברת על התיאור של M, מוסיפה מצב q_{loop} במקרה שמגיעים ל-q_{rej}, מוסיפה מעבר $\langle q_{\text{loop}}, \sigma, q_{\text{loop}}, \sigma, R\rangle$ לכל $\sigma \in \Gamma$ (המעבר שתיארנו לעיל), משנה מעברים שהלכו ל-q_{rej} כך שילכו ל-q_{loop}, ומעתיקה את w ל-w'.

ב. כעת, נוכיח כי אכן מדובר בפונקציית מיפוי. כלומר כי $x \in A_{\text{TM}} \Leftrightarrow f(x) \in \text{HALT}_{\text{TM}}$.
כלומר, נרצה להוכיח כי $\langle M, w\rangle \in A_{\text{TM}} \Leftrightarrow f(\langle M, w\rangle) \in \text{HALT}_{\text{TM}}$:

- אם $\langle M, w\rangle \in A_{\text{TM}}$, כלומר מ"ט M עוצרת על w, אזי הריצה של M' על w' זהה לריצה של M על w, כי הרי לא הגענו ל-q_{rej} ולכן M' עוצרת על w'. כלומר, $\langle M', w'\rangle \in \text{HALT}_{\text{TM}}$.
- אם $\langle M, w\rangle \notin A_{\text{TM}}$, אזי אם M נתקעת על w אזי גם M' נתקעת על w' (אותה ריצה) ולכן לא עוצרת, ואם M דוחה את w, אזי M' מגיעה ל-q_{loop} ונתקעת ולכן בהכרח המכונה M' נתקעת על w' ולא עוצרת, כלומר $\langle M', w'\rangle \notin \text{HALT}_{\text{TM}}$.

שלילת כריעות $\text{HALT}_{\text{TM}}^{\varepsilon}$

ניקח את:

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

נבחין כי השפה שייכת ל-RE. מאידך, נשים לב כי היא איננה שייכת ל-R.
נוכל להוכיח זאת באמצעות ה'רדוקציה ההפוכה', ולהראות כי $\text{HALT}_{\text{TM}} \leq_m \text{HALT}_{\text{TM}}^{\varepsilon}$ (ולמצוא f כמו שראינו).
נבנה מכונה M', כך ש-$f(\langle M, w\rangle) = \langle M', w'\rangle$. בהינתן $\langle M, w\rangle$ שהינם קלטים ל-HALT_{TM}, המכונה M' כותבת w על הסרט, וזה עם הראש הקורא לאות הראשונה של w, עוברת למצב ההתחלתי של M.
f ניתנת לחישוב: נוסיף $|w|+1$ מצבים, בהם M'כותבת w על הסרט וגם מצבים לחזור עם הראש הקורא שמאלה, ואת התיאור של M.
הנכונות נובעת מכך ש-M' מסמלצת ריצה של M על w, ולכן M עוצרת על w אם"ם M'עוצרת על ε.

דוגמה

$\text{HALT}_{\text{TM}}^{aba} = \{\langle M\rangle \mid \text{עוצרת על } w \ M\}$. במקרה שלנו, M' תבדוק שכתוב aba על הסרט, מוחקת אותו, כותבת את w וזזה שמאלה.

שלילת כריעות REG_{TM}

ניקח את $\text{REG}_{\text{TM}} = \{\langle M\rangle \mid \text{רגולרית } L(M)\}$ - כל **המכונות** ששפתם רגולרית.
נראה ש-$A_{\text{TM}} \leq \text{REG}_{\text{TM}}$ ולכן $\text{REG}_{\text{TM}} \notin R$.
עלינו למצוא f כך ש-$\langle M, w\rangle \in A_{TM}$ אם ורק אם $f(\langle M, w\rangle) = \langle M'\rangle \in \text{REG}_{\text{TM}}$.

<u>בנייה</u>

ניקח M' שפועלת על $x \in \{0,1\}^*$ כך:

1. אם $x \in \{0^n 1^n \mid n \geq 0\}$, אזי M' מקבלת את x.
2. אחרת, M' מריצה את M על w ועונה כמוה (M' מקבלת את x אם"ם M קיבלה את w).

נבחין כי f חשיבה (בהינתן M, w ניתן לייצר את M' ולעצור תמיד).

<u>נכונות</u>

נוכיח כי M מקבל את w אם ורק אם $L(M')$ רגולרית.

- אם $\langle M,w\rangle\in A_{\text{TM}}$ כלומר, M מקבל את w, אזי $L\left(M'\right)=\left(0+1\right)^{*}$ בהכרח, כי הרי לכל $x\in\left(0+1\right)^{*}$, אם $x\in0^{n}1^{n}$ אזי M' תקבל את x בשלב הראשון. בכל מקרה, יתקיים כי $L\left(M'\right)$ רגולרית, כלומר $\langle M'\rangle\in\text{REG}_{\text{TM}}$.
- אם $\langle M,w\rangle\notin A_{\text{TM}}$, אזי $L\left(M'\right)=\left\{ 0^{n}1^{n}\mid n\geq0\right\}$ כי הרי ישנן שתי אפשרויות:
  - אם $x\in0^{n}1^{n}$ אזי M' מקבלת את x בשלב 1 ולכן רק $0^{n}1^{n}$ מתקבל.
  - אם $x\notin0^{n}1^{n}$, אזי M' לא מקבלת את x בשלב הראשון וגם לא בשלב השני ולכן בהכרח M' תיתקע או תידחה. כלומר, קיבלנו כי במצב זה $L\left(M'\right)$ איננה רגולרית, כלומר $\langle M'\rangle\notin\text{REG}_{\text{TM}}$.

משפט רדוקציה ל-RE ול-coRE

1. אם $B\leq_{m}A$ ו-$B\in$RE אזי $A\in$RE.
2. אם $B\leq_{m}A$ ו-$B\in$coRE אזי $A\in$coRE.

דוגמה - שלילת זיהוי REG_{TM}

נרצה כעת להראות כי $\text{REG}_{\text{TM}}\notin\text{RE}$.
אנו יודעים כי $\overline{A_{\text{TM}}}\notin\text{RE}$ (ראינו כבר כי $A_{\text{TM}}\notin\text{coRE}$ ולכן זה נובע באופן ישיר). אם נראה כי $\overline{A_{\text{TM}}}\leq_{m}\text{REG}_{\text{TM}}$ נוכל להסיק כי $\text{REG}_{\text{TM}}\notin\text{RE}$ כיוון ש-$A_{\text{TM}}\leq_{m}\overline{\text{REG}_{\text{TM}}}$ שקול ל-$\overline{A_{\text{TM}}}\leq_{m}\text{REG}_{\text{TM}}$.
f מוגדרת על ידי $f\left(\langle M,w\rangle\right)\to\langle M'\rangle$. נרצה לבנות את M' בצורה כזאת ש-M מקבלת את w אם"ם $L\left(M'\right)$ לא רגולרית.

<u>בנייה</u>
M' תוגדר כך - בהינתן $x\in\left(0+1\right)^{*}$:

- אם $x\in\left\{ 0^{n}1^{n}\mid n\geq0\right\}$ אזי M' מריצה את M על w ועונה כמוה (M' מקבלת את x, אם M מקבלת את x).
- אחרת: M' דוחה את x.

כלומר, אם M מקבלת את w, אזי $L\left(M'\right)$ לא רגולרית ואם אם M לא מקבלת את w, אזי $L\left(M'\right)$ רגולרית.
בסופו של דבר, הראינו כי $\text{REG}_{\text{TM}}\in\overline{\text{RE}\cup\text{coRE}}$ (גם אי אפשר לזהות אותה, וגם אי אפשר לזהות את המשלים שלה).

הרצאה מס' 14:

יום רביעי

24.11.21

הוכחת $\text{INF}_{\text{TM}}\in\overline{\textbf{RE}\cup\textbf{coRE}}$.
נתבונן בשפה $\text{INF}_{\text{TM}}=\left\{ \langle M\rangle\mid L\left\langle M\right\rangle \text{ אינסופית}\right\}$ - כל המכונות ששפתן אינסופיות.
אם היינו צריכים לבדוק האם שפה של אוטומט היא אינסופית, היינו צריכים למצוא האם יש מעגל בגרף של האוטומט, נרצה לשאול את עצמנו האם ניתן לעשות אותו דבר גם על מכונות טיורינג, באמצעות גרף הקונפיגורציות, למשל.

אמנם, נראה כי $\text{INF}_{\text{TM}}\notin\text{RE}$ וגם $\text{INF}_{\text{TM}}\notin\text{coRE}$ - כלומר, לא ניתן לזהות האם שפה של מכונת טיורינג היא אינסופית, וגם לא לזהות האם היא סופית. נעשה זאת באמצעות שתי רדוקציות.

תחילה, נעשה רדוקציה באמצעות A_{TM}: נראה כי $\text{INF}_{\text{TM}}\notin\text{coRE}$ -INF_{TM} ונוכיח כי $A_{\text{TM}}\leq_{m}\text{INF}_{\text{TM}}$ כלומר ניקח $\langle M,w\rangle$ ונייצר מהם M', כך ש-$L\left(M'\right)$ אינסופית אם ורק אם M מקבלת את w (אנחנו מתחילים להתעלם מ-f, אך ברור כי מתחת לפני השטח קיימת f שמעבירה את $\langle M,w\rangle$ ומייצרת את $\langle M'\rangle$.

בנייה
בהינתן קלט $x\in\Sigma^*$ (א"ב של M', לבחירתנו), M' מתעלמת מ-x לגמרי, ומריצה את M על w ועונה כמותה.

נכונות

- אם $\langle M,w\rangle\in A_{\text{TM}}$, כלומר M מקבלת את w, אזי $L\left(M'\right)=\Sigma^*$ כי למעשה השפה של M' היא כל המילים שמתקבלות. כלומר $L\left(M'\right)\in \text{INF}_{\text{TM}}$.
- אם $\langle M,w\rangle\notin A_{\text{TM}}$ אזי $L\left(M'\right)=\emptyset$ ואז $L\left(M'\right)\notin \text{INF}_{\text{TM}}$.

נבצע רדוקציה באמצעות $\text{HALT}_{\text{TM}}\leq_m \overline{\text{INF}_{\text{TM}}}$ ונראה כי $\text{INF}_{\text{TM}}\notin \text{RE}$.

בנייה
נציג g חשיבה, כך ש-$g\left(\langle M,m\rangle\right)\to\langle M'\rangle$
על קלט x, המכונה M' פועלת כך:
M' מריצה את M על w במשך $|x|$ צעדים. אם M עצרה על w במהלך $||x||$ הצעדים, המכונה דוחה את x. אחרת, M מקבלת את x.

נכונות
נרצה להוכיח כי $\langle M,w\rangle\in \text{HALT}_{\text{TM}}$ אם ורק אם $\langle M'\rangle\notin \text{INF}_{\text{TM}}$.

- אם $\langle M,w\rangle\in \text{HALT}_{\text{TM}}$, אזי קיים $m\geq 0$ כך ש-M עוצרת על w אחרי m צעדים. לכל קלט x, אם $|x|<m$ אזי M לא תעצור על w תוך $|x|$ צעדים, ולכן $x\in L\left(M'\right)$. מצד שני, אם $|x|\geq m$, אזי M מספיקה לעצור על w ולכן $x\notin L\left(M'\right)$. כלומר, קיבלנו כי אם $\langle M,w\rangle\in \text{HALT}_{\text{TM}}$ אזי $L\left(M'\right)=\Sigma^{<m}$ - כל המילים באורך קטן מ-m ומדובר בשפה סופית, ולכן $\langle M'\rangle\notin \text{INF}_{\text{TM}}$.
- אם $\langle M,w\rangle\notin \text{HALT}_{\text{TM}}$ אזי $L\left(M'\right)=\Sigma^*$, כי לכל x, M לא תעצור על w תוך $|x|$ צעדים ולכן M' תקבל את x ולכן $\langle M'\rangle\in \text{INF}_{\text{TM}}$.

היינו יכולים לעשות את הרדוקציה הראשונה גם באמצעות הרדוקציה השנייה, אם היינו משנים זאת לקבלה ולא לעצירה.

הרצאה מס' 15:

יום שני

29.11.21

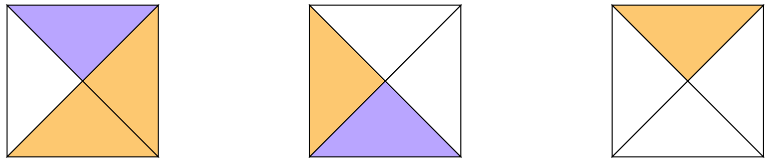
2.2.1 בעיית הריצוף

ניזכר בבעיית הריצוף, עליה דיברנו בתחילת הקורס.

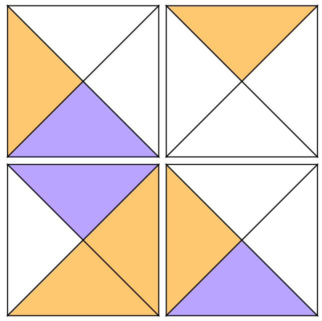
קלט:
קבוצה סופית T של אריחים $T=\{t_0,t_1,\ldots,t_n\}$
תנאי שכנות במאוזן ובמאונך $H\subseteq T\times T$ ו-$V\subseteq T\times T$.
אריח התחלתי $t_{\text{init}}\in T$.

דוגמה
נניח כי ריצוף חוקי הוא כי אריחים סמוכים 'מסכימים' על הצבע.

עבור הקלט הבא מסתבר שהתשובה היא כן:



לדוגמה, ניתן לרצף ריבוע 2×2 באופן הבא:



הגדרה

TILE=$\{\langle T, V, H, t_{\text{init}} \rangle \mid 1 \leq n$ לכל $n \times n$ יש ריצוף חוקי$\}$.

ריצוף חוקי $n \times n$ מוגדר על ידי פונקציה $f_i : \{1 \dots n\} \times \{1 \dots n\} \to T$ כאשר לכל $1 \leq i \leq n$ מתקיים כי $1 \leq j \leq n$ וגם כי $f(1,1) = t_{\text{init}}$.

וגם $(f(i,j), f(i,j+1)) \in V$ ו-$(f(i,j), f(i+1,j)) \in H$.

טענה

יש ריצוף חוקי $n \times n$ לכל n, אם"ם יש ריצוף חוקי (אינסופי) לרבע המישור החיובי.

הוכחה

תחילה, נזכיר את הלמה של קניג.

הלמה של קניג

בכל עץ אינסופי עם דרגת פיצול סופית, יש מסלול אינסופי.

אם יש ריצוף $n \times n$ לכל $1 \leq i \leq n$, נבנה עץ אינסופי כזה:

- ברמה i: ריצופים חוקיים $i \times i$.
- הבנים של קודקוד f (ריצוף חוקי $i \times i$): ריצופים חוקיים $i+1 \times i+1$ שמרחיבים את f .

העץ מקיים את תנאי הלמה של קניג ולכן יש מסלול אינסופי, כלומר יש ריצוף אינסופי של רבע המישור.

בכיוון השני, אם יש ריצוף אינסופי של רבע המישור, אזי נתבונן על הרישא שלו.

הוכחת האפיון של TILE

נרצה להוכיח כי TILE $\in$ coRE, כלומר, כי $\overline{\text{TILE}} \in \text{RE}$ (קיים m שאין לו ריצוף חוקי).

נבנה מ"ט שמזהה את $\overline{\text{TILE}}$:

עבור $m \geq 1$ המכונה בודקת את כל הריצופים $m \times m$. אם מוצאת ריצוף חוקי כלשהו, אזי מגדילה את m ב-1.
אם בדקה את כל הריצופים $m \times m$ (יש מספר סופי - לכל היותר $|T|^{m \times m}$) וכולם לא חוקיים, מקבלת.
כעת, נראה כי $\text{RETILE} \notin$ ונראה זאת באמצעות רדוקציה ל-$\overline{\text{HALT}^{\varepsilon}_{\text{TM}}}$ (כל מכונת הטיורינג שלא עוצרות על ε).
נרצה להראות כי אם היינו יכולים לפתור את בעיית הריצוף, היינו יכולים גם לפתור את בעיה זו, כלומר נרצה למצוא f כך ש-$\langle M \rangle \in \overline{\text{HALT}^{\varepsilon}_{\text{TM}}}$ אם ורק אם $f(\langle M \rangle) = (T, H, V, t_{\text{init}}) \in \text{TILE}$.

רעיון הרדוקציה

נזכיר כי קונפיגרוציה של מכונת טיורינג מוגדרת על ידי $\Gamma^* \cdot (0 \times \Gamma) \cdot \Gamma^*$ ולאחר מכן אינסוף אותיות ב-Γ. למשל, קונפיגורציה התחלתית על המילה aba היא $(q_0, a) \cdot b \cdot a \cdot \sqcup \ldots \sqcup$. זוג קונפיגורצות עוקבות הן כמעט זהות - ההבדל היחיד הוא מיקום הראש והמצב הנוכחי של כל אחת, והוא מוגדל על פי δ.
נוכל להמיר כל קונפיגורציה לתיאור של אריחים
כל קומה בריצוף תהיה קונפיגרוציה, וכל מעבר בין קומות יהיה מעבר בין קונפיגורציות עוקבות.
הקומה הראשונה תהיה הקונפיגורציה ההתחלתית של M על ε, נוכל להגדיר אילו קונפיגורציות עוצרות ואילו לא.

** חלק זה חסר - יושלם בקרוב!!!**

הגדרת האריחים

מרצפות מסוג 2R: לכל מעבר $\delta(q, a) = (q', b, R)$ עבור $q \neq q_{\text{acc}}, q_{\text{rej}}$ נוסיף $|\Gamma| + 1$ מרצפות:
מרצפות מסוג 2L: לכל מעבר $\delta(q, a) = (q', b, L)$ עבור $q \neq q_{\text{acc}}, q_{\text{rej}}$ נוסיף $|\Gamma| + 1$ מרצפות:

נבחין כי הרדוקציה ניתנת לחישוב, בהינתן M ניתן לייצר את $\langle T, V, H, t_{\text{init}} \rangle$.
למעט בטור השמאלי ביותר (ניתן לפתרון ע"י הגדרה מורכבת יותר):

- בכל קומה יש $*$ בכל הצדדים, למעט במקום אחד (מעבר של הראש הקורא).
- קומה: קונפיגרוציה. מעבר בין קומות: מעבר בין קונפיגורציות.

מכך עולה כי M לא עוצרת על ε גורר כי יש ריצוף אינסופי.
בכיוון השני יש ריצוף אינסופי, אזי M לא עוצרת על ε.

בעיית ההתאמה של פוסט (Post Correspondence Problem).

קלט: אוסף סופי של אבני דומינו, e_i שמורכבים מ-$u_i, d_i \in \Sigma^*$.
פלט: $e_{i_1}, e_{i_2}, \ldots, e_{i_m}$ כך ש-$u_{i_1} \cdot u_{i_1} \ldots \cdot u_{i_m} = d_{i_1} \cdot d_{i_2} \cdot \ldots \cdot d_{i_m}$.

ראשית, נראה כי הבעייה ניתנת לזיהוי (שייכת ל-RE). נוכל לבנות מ"ט שתעבור על כל הסדרות בגודל $1, 2, 3, \ldots$, אם יש match, תמצא אותו ותעצור.
כעת, נראה כי המשלים של הבעייה אינו ניתן לזיהוי. כלומר $\text{PCP} = \{\langle e_1, \ldots, e_n \rangle \mid e_1, \ldots, e_n$ קיים match ב$\} \notin$ coRE.
הרדוקציה תיעשה באמצעות A_{TM}.

הרצאה מס' 16:

יום רביעי

01.12.21

3 תורת הסיבוכיות

בחלק הזה נעשה אפיון של שפות כריעות.

למעשה, כבר ראינו כמה אפיונים, למשל REG או CFL, שהרי אלו שפות כריעות.

כעת, נראה אפיון לפי משאבים: זמן ומקום.

3.1 מבוא לסיבוכיות

סיבוכיות זמן

הגדרה

חסם עליון על אלגוריתם הוא מציאת סיבוכיות זמן **ישיגה** באלגוריתם.

חסם תחתון על אלגוריתם הוא מציאת סיבוכיות הזמן הטוב ביותר האפשרית.

דוגמה

נתבונן לרגע במ"ט שמכריעה את $\{0^n 1^n \mid n \geq 0\}$:

1. בודקת שהקלט מהצורה 0^*1^*.
2. כל עוד אפשר: מוחקת 0 ראשון ו-1 ראשון.
3. מקבלת אם המחיקות בשני הצדדים הסתיימו יחד.

ניתוח הסיבוכיות על קלט באורך n:

1. $O(n)$ צעדים.
2. לכל היותר, n איטרציות שכל אחת מהן היא $O(n)$ צעדים, ולכן $O\left(n^2\right)$.

אם כך, סיבוכיות הזמן היא $O\left(n^2\right)$.

הגדרה

עבור פונקציה $t : \mathbb{N} \to \mathbb{N}$ מחלקת הסיבוכיות, נגדיר את מחלקת הסיבוכיות $\mathrm{TIME}\left(t\left(n\right)\right)$, שהינן כל השפות שניתנות להערכה על ידי מכונות טיורינג דטרמיניסטיות עם סרט יחיד, הרצות על קלט באורך n, לכל היותר $O\left(t\left(n\right)\right)$ צעדים.

נשים לב כי השפה שראינו מקודם ניתנת להכרעה ע"י מ"ט עם שני סרטים בזמן ליניארי.

משפט

לכל מ"ט מרובת סרטים שעובדת בזמן $t\left(n\right)$, יש מ"ט שקולה בעלת סרט יחיד שעובדת בזמן $O\left(t^2\left(n\right)\right)$.

ניתן להכריע את השפה שראינו מקודם ב-$O\left(n \log n\right)$ צעדים. בכל איטרציה מוחקים חצי מה-0-ים וה-1-ים. ישנן $\log\left(n\right)$ איטרציות כשבכל איטרציה $O\left(n\right)$ צעדים.

משפט
אם L ניתנת להכרעה בזמן $O\left(n\log n\right)$ אזי L רגולרית.

ראינו בתרגול את ההגדרה של מ"ט אי דטרמיניסטיות:

הגדרה
מכונת טיורינג אי דטרמינסטית היא שביעייה:

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

כאשר δ מוגדרת על ידי:

$$\delta:Q\setminus\left\{q_{\text{rej}},q_{\text{acc}}\right\}\times\Gamma\to2^{Q\times\Gamma\times\{L,R\}}\setminus\{\emptyset\}$$

ראינו כי למכונה זו קיים עץ ריצות, לפי הקונפיגורוציות שקיימות על ריצות המילה. כמו כן, אמרנו ש-M **מכריעה** את L, אם M עוצרת על כל מילה בכל החישובים על המילה. מעבר לכך, הראינו כי $w\in L$ אם קיים חישוב מקבל כלשהו.

הגדרה
עבור פונקציה $t:\mathbb{N}\to\mathbb{N}$ מחלקת הסיבוכיות, נגדיר את מחלקת הסיבוכיות $\text{TIME}\left(t\left(n\right)\right)$, שהינן כל השפות שניתנות להערכה על ידי מכונות טיורינג אי דטרמיניסטיות עם סרט יחיד, הרצות על קלט באורך n, לכל היותר $O\left(t\left(n\right)\right)$ צעדים בכל הריצות.

3.1.1 המחלקות P ו-NP

הגדרה
המחלקה P היא מחלקת השפות שניתנות להכרעה בזמן פולינומיאלי, כלומר $\text{P}=\bigcup\limits_k\text{TIME}\left(n^k\right)$.

הגדרה
המחלקה NP היא מחלקת השפות שניתנות להכרעה בזמן פולינומיאלי עם מ"ט א"ד, כלומר $\text{NP}=\bigcup\limits_k\text{NTIME}\left(n^k\right)$.

הגדרה
המחלקה EXPTIME היא מחלקת השפות שניתנות להכרעות בזמן אקספוננציאלי, כלומר $\text{EXPTIME}=\bigcup\limits_k\text{TIME}\left(2^{n^k}\right)$.

טענה
$\text{P}\subseteq\text{NP}\subseteq\text{EXPTIME}$.

הוכחה

P ⊆ NP יחסית פשוט, שהרי כל דבר שאפשר לעשות עם מכונת טיורינג דטרמינסטית, אפשר לעשות עם מכונה אי דטרמינסטית.
את ההכלה השנייה נוכיח בהמשך.

השאלה האם $\stackrel{?}{=}$ NPP היא שאלה פתוחה במדעי המחשב, ואחת השאלות הפתוחות הקיימות היום.

משפט

אם L ניתנת להכרעה ע"י מ"ט א"ד בזמן $t(n)$, אזי L ניתנת להכרעה על ידי מ"ט דטרמיניסטית בזמן $2^{O(t(n))}$.

טענה

P ⊆ NP ⊆ EXPTIME.

הוכחה

P ⊆ NP יחסית פשוט, שהרי כל דבר שאפשר לעשה על ידי מ"ט דטרמינסטית בזמן $2^{O(t(n))}$ צעדים.

הרצאה מס' 17:

יום שני

06.12.21

3.2 המחלקה NP

הגדרה

מסלול המילטוני בגרף הוא מסלול שעובר בכל הקודקודים, בדיוק פעם אחת.

3.2.1 שייכות D-ST-HAMPATH ל-NP

נתבונן בשפה $\{\langle G,s,t\rangle\}$ =D-ST-HAMPATH כך שבגרף המכוון יש מסלול המילטון מ-s ל-t.

נבחין כי D-ST-HAMPATH∈ EXPTIME שהרי נוכל לבנות מ"ט שמכריעה את D-ST-HAMPATH בזמן אקספוננציאלי:

- המכונה עוברת על כל הסדרות ב-$|V|^n$. אם יש סדרה שמתחילה ב-s, מסתיימת ב-t ומהווה פרמוטציה של V וקודקודים עוקבים יש ביניהם קשת - אזי מקבלת.

כהערה, נשים לב כי לא פולינומיאלי קשה למצוא, למשל לבדוק האם יש מסלול המליטון. פולינומיאלי קל לאמת: כלומר, קל לבדוק האם סדרה של קודקודים היא מסלול המילטון.

נראה כי D-ST-HAMPATH∈ NP:
מ"ט א"ד שרצה בזמן פולינומיאלי, תעבוד כך:

1. מנחשת סדרה $v_1,v_2,\ldots,v_n$ של קודקודים.
   (א) אם $v_1 \neq s$ - דוחה.
   (ב) אם $v_n \neq t$ - דוחה.
   (ג) אם יש קודקוד שחוזר על עצמו יותר מפעם אחת - דוחה.
   (ד) אם יש $1 \leq i \leq n$ כך ש-$(v_i,v_{i+1}) \notin E$ - דוחה.
   (ה) אחרת, מקבלת.

3.2.2 **אפיון של המחלקה NP** על ידי מאמת\מוודא **(Verifier)**

הגדרה
עבור שפה L: **מוודא** עבור L, הוא מ"ט דטרמיניסטית V כך ש:

$$L = \{w \mid \text{קיים } c \in \Sigma^* \text{ כך ש-} V \text{ מקבלת את } \langle w, c\rangle\}$$

ל-$c \in \Sigma^*$ שבתוך $\langle w, c\rangle$, נקרא עֵד (או Certificate).

דוגמה - מוודא עבור D-ST-HAMPATH
נוכל למצוא מ"ט V כך ש:

$$L(V) = \{\langle (G, s, t), \pi\rangle \mid \pi \text{ מסלול המילטון ב-} G \text{ מ-} s \text{ ל-} t\}$$

דוגמה נוספת
נתבונן בשפה:

$$\text{COMPOSIME} = \{x \mid x \in \mathbb{N}, \text{ יש } p, q \in \mathbb{N} \text{ כך ש-} p, q \neq 1 \text{ ו-} p \cdot q = x\}$$

אלגוריתם אקספוננציאלי עבור מציאת השפה:

1. עוברים על כל $p = 2, 3 \ldots, \sqrt{x}$ ובודקים האם p מחלק את x.

מכיוון ש-x נתון בבינארית, אורך הקלט הוא $\log_2 x$ ולכן מדובר באלגוריתם אקספוננציאלי.
קשה להכריע האם x פריק אבל קל לבדוק שעֵד p הוא כזה ש-p מחלק את x, ללא שארית, וכך נוכל למצוא את המוודא עבור השפה:

$$V = \{\langle x, p\rangle \mid x = 0 \mod p\}$$

הגדרה
סיבוכיות של מוודא היא זמן הריצה ביחס למילה w.

הגדרה
נאמר כי מוודא הוא **מוודא פולינומיאלי** אם הוא רץ על $\langle w, c\rangle$ בזמן פולינומיאלי ב-$|w|$.
במילים אחרות, נאמר כי קיים מוודא פולינומיאלי ל-L אם:

$$L = \{w \mid \text{קיים } c \text{ פולינומיאלי ב-} w \text{ כך ש-} V \text{ מקבלת בזמן פולינומיאלי את } \langle w, c\rangle\}$$

משפט

$L \in \mathrm{NP}$ אם"ם יש ל-L מוודא פולינומאלי.

הוכחה

$\Rightarrow$ יש ל-L מוודא פולינומיאלי V, כלומר יש פולינום $t_V : \mathbb{N} \to \mathbb{N}$ שחוסם את זמן הריצה של V. נבנה מ"ט א"ד שמכריעה את L בזמן פולינומיאלי: בהינתן מילה w, מנחשת עד c באורך **קטן שווה** מ-$t_V(w)$, מריצה את V על $\langle w, c\rangle$ ועונה כמו V.

למעשה, קיבלנו מ"ט V, כך שיש פולינום $t_V : \mathbb{N} \to \mathbb{N}$ כשעל קלט (w, c) V רצה לכל היותר $t_V(|w|)$ צעדים ו:

$$L = \{w \mid \text{קיים } c \text{ כזה כך ש-} |c| \le t_V(w) \text{ מקבלת את } (w, c)\}$$

ולכן $L \in \mathrm{NP}$.

$\Leftarrow$ יש מ"ט M שרצה בזמן פולינומיאלי ו$L(M) = L$.

המוודא הוא:

$$L(V) = \{(w, r) \mid \text{ריצה מקבלת של } M \text{ על } w \; r\}$$

מתקיים, אם כך:

$$L = \{w \mid \text{יש } r \text{ כך ש-} V \text{ מקבל את } (w, r)\}$$

ברור כי V פולינומאלי כי M פולינומאלית (האורך של r חסום על ידי אורכי הריצות של M + בדיקה שאכן הריצה מקבלת).

נשים לב כי על פניו, לא ברור האם כל בעיה שב-EXPTIME היא ב-NP:

$$\overline{\text{D-ST-HAMPATH}} = \{\langle G, s, t\rangle \mid G \text{ מכוון ואין מסלול המילטוני מ-} s \text{ ל-} t\}$$

לא ברור ש-$\overline{\text{D-ST-HAMPATH}} \in \mathrm{NP}$, שהרי לא ברור שיש עֵד קצר שמשכנע שאין מסלול המילטון.

3.3 רדוקציות פולינומיאליות

הגדרה

נאמר כי שפה L היא NP- שלמה אם:

1. חסם עליון -NP $L \in \mathrm{NP}$.
2. חסם תחתון - L היא NP-קשה - אם $L \in \mathrm{P}$ אז $\mathrm{P} = \mathrm{NP}$.

רדוקציות פולינומיאליות

ראינו כי $A \le_m B$ אם יש פונקציה f, ניתנת לחישוב כך שלכל $w \in \Sigma^*$, $w \in A \Leftrightarrow f(w) \in B$.

הגדרה
נאמר כי f ניתנת לחישוב בזמן פולינומיאלי אם יש פולינום $t:\mathbb{N}\to\mathbb{N}$ ומ"ט M_t שעל קלט w, עוצרת אחרי $t(w)$ עם $f(w)$ על הסרט.

הגדרה
נאמר כי $A\leq_P B$ אם יש פונקציה f ניתנת לחישוב בזמן פולינומיאלי כך שלכל $w\in\Sigma^*$, יתקיים כי $w\in A \Leftrightarrow f(w)\in B$.

משפט
אם $A\leq_P B$ ו-$B\in P$ אזי $A\in P$.

הוכחה
בהינתן M_B, מ"ט דטרמיניסטית שמכריעה את B בזמן פולינומיאלי, M_t מ"ט דטרמיניסטית שמחשבת בזמן פולינומיאלי פונקציה f כך ש-$f(w)\in B \Leftrightarrow w\in A$.
נבנה M_A שמכריעה את A בזמן פולינומיאלי: בהינתן w, תריץ את M_t, תקבל את $f(w)$ (נשים לב כי $|f(w)|$ פולינומיאלי ב-$|w|$) ותריץ את M_f על $f(w)$ ותענה כמותה.

הגדרה
L היא NP-קשה, אם לכל $L'\in \text{NP}$ יתקיים כי $L'\leq_P L$.

טענה
אם L היא NP-קשה לפי הגדרת השלימות, אם L היא NP- קשה לפי ההגדרה הנוכחית.

הגדרה נוספת
L היא NP קשה אם יש שפה L'' כך ש-L'' היא NP- קשה לפי ההגדרה האחרונה ו-$L''\leq_P L$.

נשים לב כי אם L היא NP קשה לפי ההגדרה האלטרנטיבית אזי L היא NP-קשה לפי ההגדרה השנייה, כי לכל $L'\in \text{NP}$ אנו יודעים כי $L'\leq_P L''$ ולכן $L''\leq_P L$, מהטרנזיטיביות של $\leq_P$ עולה כי $L'\leq_P L$.

הרצאה מס' 18:

יום רביעי

08.12.21

הגדרה
SAT היא ספיקות של נוסחאות בלוגיקה פסוקית, בעלת התכונות הבאות:

- משתנים בוליאנים $x_i\in\{\mathbb{T},\mathbb{F}\}$.
- נוסחא בוליאנית פסוקית:
  - משתנה בוליאני הוא נוסחא: אם φ_1 ו-φ_2 נוסחאות, אזי גם $\neg\varphi_1$ ו-$\varphi_1\vee\varphi_2$ ו-$\varphi_1\wedge\varphi_2$ נוסחאות.
- $\text{SAT}=\{\langle\varphi\rangle \mid \text{נוסחה ספיקה } \varphi\}$.

הגדרה

- משתנים בוליאנים $x_i \in \{\mathbb{T}, \mathbb{F}\}$
- ליטרלים - משתנה x_i או שלילתו $\neg x_i$.
- פסוקית בנוסחת 3-CNF : אוסף של שלושה ליטרלים ודיסיונקציה (או אימום) ביניהם. למשל, $(x_1 \vee \neg x_n \vee x_4)$.
- קוניונקציה (או גימום) של פסוקיות. למשל: $(x_1 \vee \neg x_n \vee x_4) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$.

בשנות ה-70, קוק ולוין הוכיח כי SAT $\in$ P אם"ם NP=P, כלומר מדובר בבעיית NP-קשה.

רדוקציה מבעיית הקליקה (CLIQUE) ל-3-SAT

קלט

גרף לא מכוון $G = \langle V, E \rangle$, $K \in \mathbb{N}$.

פלט

קליקה בגודל K.

נרצה להראות כי הבעיה הזאת היא NP- קשה ונראה זאת באמצעות רדוקציה ל-3-SAT.

כלומר, כי CLIQUE $\leq_P$ 3-SAT.

בנייה

בהינתן $\varphi \in$ 3CNF, שנתון על ידי $\varphi = \left(\ell_1^1 \vee \ell_1^2 \vee \ell_1^3\right) \wedge \left(\ell_1^2 \vee \ell_2^2 \vee \ell_2^3\right) \wedge \ldots \wedge \left(\ell_m^1 \vee \ell_m^2 \vee \ell_n^3\right)$,

נבנה $\langle G, K \rangle$ כך ש-$G = \langle V, E \rangle$,ו-V מוגדר על ידי:

$$V = \left\{\ell_1^1, \ell_1^2 \ldots, \ell_m^1, \ell_m^2, \ell_m^3\right\}$$

ו-E מוגדר על ידי:

$V \times V \setminus (\{(v_1, v_2) \mid$ הליטרלים של v_1ו-v_2 מתאימים למשתנה ושלילתו$\} \cup$

$\{(v_1, v_2) \mid$ הליטרלים של v_1ו-v_2 מתאימים לאותה פסוקית$\})$

נכונות

1. הרדוקציה פולינומיאלית - $|V| = 3m$ ויש מספר פולינומיאלי של מבחנים בהגדרת E.
2. הרדוקציה נכונה - (משמרת שייכות) $(G, K) \in$ CLIQUE $\Leftrightarrow \varphi \in$ 3-SAT.

(א) נניח כי φ ספיקה, כלומר, קיימת $f : \{x_1, \ldots, x_n\} \to \{\mathbb{F}, \mathbb{T}\}$ שמספקת את φ.

בכל פסוקית $\ell_i^1 \vee \ell_i^2 \vee \ell_i^3$ יש $1 \leq j \leq 3$ כך ש-$f\left(\ell_i^j\right) = \mathbb{T}$ - אם $\ell_i^j = x$, אזי $f(x) = \mathbb{T}$, ואם $\ell_i^j = \overline{x}$, אזי $f(x) = \mathbb{F}$.

נראה כי יש קשתות בין כל נציגי הפסוקיות:

- הן בפסוקיות שונות.

- מכיוון שנבחרו על סמך f, כל המשתנים מופיעים.
- אם הם מסכימים על ההשמה, יש קשת, ולכן יש $K=m$ קודקודים עם K-קליק.

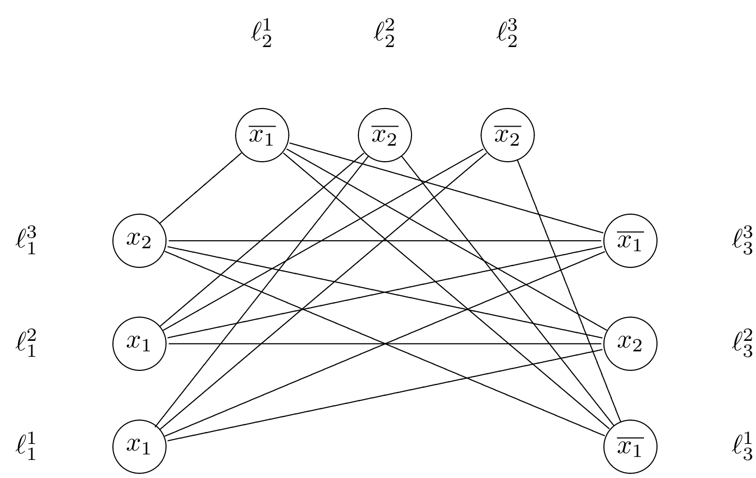
(ב) נניח שיש ב-G קליקה בגודל K. מהגדרת E, יש לכל פסוקית, לכל היותר (בדיוק, כי $K=m$) נציג אחד ב-S.
כל משתנה, אם ליטרלים שמתאימים לו משתתפים בקליקה, הם מסכימים על ההשמה, לכן הקליק משרה $f:\{x_1,\ldots,x_n\}\to\{\mathbb{F},\mathbb{T}\}$.
היא מספקת כיוון שכל פסוקית משרה ערך אחד ולכן הפסוקית ספיקה.

דוגמה

ניקח את הקלט הבא:

$$\varphi=(x_1\vee x_1\vee x_2)\wedge(\overline{x_1}\vee\overline{x_2}\vee\overline{x_2})\wedge(\overline{x_1}\vee x_2\vee x_2)$$

ובאמצעות הרדוקציה, נוכל לבנות את הקליקה הבאה:



הרצאה מס' 19:

יום רביעי

13.12.21

3.3.1 משפט קוק לוין

אם SAT∈P אזי P=NP, כלומר SAT הינו NP-שלימה.

הוכחה

תהי $L\in$ NP. בהינתן מילה $w\in\Sigma^*$, ניתן בזמן פולינומאלי למצוא נוסחה φ כך ש-$w\in L$ אם"ם φ ספיקה.
הרעיון הינו כזה: φ תבדוק אם יש ריצה מקבלת של M על w. אם יש ריצה מקבלת, אזי יש סדרה של קונפ' $c_0,c_1,\ldots,c_m$ כך ש-c_0 היא הקונפ' ההתחלתית של M על w, לכל $0\le i\le m$ הקונפ' c_{i+1} עוקבת לקונפ' c_i, והקונפ' c_m מקבלת. במהלך ההוכחה, ננסה להפוך קונפיגורציה של מכונת טיורינג, ל-3-CNF, ולהראות כי כל התנאים שהראינו אכן מתקיימים.

נבחין כי המכונה M, מ"ט א"ד עם פונקציית זמן פולינומיאלית $t:\mathbb{N}\to\mathbb{N}$ קיימת כי $L\in\text{NP}$.
נזכיר כי אנו מציגים קונפיגורציות באמצעות $S=\Gamma\cup Q\cup\{\#\}$,
כאשר קונפיגורציה תתואר בתור $\#\delta_1\ldots,\delta_k q'\delta_{k+1}\ldots\delta_n\#$ (מילה לפי הראש, מצב, מילה אחרי הראש). החסם על זמן הריצה יכתיב אורך קונפיגורציה, ובהינתן $n=|w|$, כיוון ש- M עוצרת על w ב-$t(w)$ צעדים, לכן היא משתמשת לכל היותר ב-$t(n)$ תאים.
כעת, נרצה כי הנוסחה תתאר **מטריצה**: לכל מילה תהיה מטריצה ייחודית משלה.
המשתנים - לכל כתובת (i,j) במטריצה ואות $s\in S$, יהיה המשתנה $X_{i,j,s}$ שמביע: האות s נמצאת בכתובת ה-i,j (**שימו לב, כאן i זה עמודות, ו-j שורות!**).
גובה הטבלה - מתאים לזמן הריצה, ולכן חסום על ידי $t(n)$.
אורך הטבלה - $t(n)+2$, כיוון שמוסיפים סולמית בהתחלה, וסולמית בסוף, ועוד אחד לכיף.
אם כך, הטבלה תראה כך:

#	q_0	w_1	w_2	$\ldots$	w_n	$\sqcup$	$\ldots$	$\sqcup$	#

הנוסחה φ תהיה גימום של הביטויים הבאים:

$$\varphi=\underbrace{\varphi_{\text{cell}}}_{\text{ההשמה מתאימה למילוי המטריצה}}\wedge\underbrace{\varphi_{\text{init}}}_{\text{בשורה הראשונה יש } q_0}\wedge$$
$$\underbrace{\varphi_{\text{move}}}_{\text{טיפוס בין שורות תואם למטריצות עוקבות}}\wedge\underbrace{\varphi_{\text{acc}}}_{\text{מגיעים לקונפ' מקבלת}}$$

נבחין כי יש מספר פולינומיאלי של משתנים $(t(n)+2)\cdot t(n)\cdot|S|$.
כיצד φ_{cell} נראה? כדי שתהיה השמה מתאימה לכל משתנה, כל משתנה, או כל תא, אמור לקבל ערך יחיד. ולכן נגדיר אותו להיות:

$$\varphi_{\text{cell}}=\underbrace{\left(\bigwedge_{\substack{1\le i\le t(n)+3\\ 1\le j\le t(n)}}\bigvee_{s\in S}X_{i,j,s}\right)}_{\text{בתא ה-}i,j\text{ יש לפחות אות אחת}}\wedge\underbrace{\left(\bigwedge_{s_1,s_2\in S,\ s_1\neq s_2}\left(\overline{X_{i.j,s_1}}\vee\overline{X_{i.j,s_2}}\right)\right)}_{\text{כל אות מופיעה בדיוק פעם אחת}}$$

כיצד φ_{init} נראה?

$$\varphi_{\text{init}}=X_{1,1,\#}\wedge X_{2,1,q_0}\wedge X_{3,1,w_1}\wedge\ldots\wedge X_{n+2,1,w_n}\wedge X_{n+3,1,\sqcup}\wedge\ldots\wedge X_{t(n)+2,1,\sqcup}\wedge X_{t(n)+3,1,\#}$$

הבדיקה הראשונה צריכה לוודא כי $X_{1,1,q_0}$ מקבל ערך $\mathbb{T}$ וכל השאר מקבלים ערך $\mathbb{F}$.
כיצד φ_{acc} נראה?

$$\varphi_{\text{acc}}=\bigvee_{\substack{2\leq i\leq t(n)+2\\ 1\leq j\leq t(n)}}X_{i,j,q_{\text{acc}}}$$

כלומר, הבדיקה הזאת מוודא כי יש כתובת שמאכלסת את q_{acc}, כלומר שיש בה ערך $\mathbb{T}$.

נשים לב כי $\varphi_{\text{acc}},\varphi_{\text{init}},\varphi_{\text{cell}}$ כולן פולינומיאלות ב-n. בנוסף, שלושתן ב-CNF, כי הן מהצורה של $(\wedge\ldots\vee)\wedge(\vee\ldots\wedge)$.
כל שנותר לנו הוא לעשות את פונקציית המעברים, כלומר את q_{move}.
כיצד q_{move} נראה? "כל החלונות בגדול 3×2 שמותרים" - כלומר שייכים לקבוצה W של חלונות אפשריים, ולמעשה:

$$\varphi_{\text{move}}=\bigwedge_{\substack{1\leq i\leq t(n)+1\\ 1\leq j\leq t(n)+1}}legal\left(i,j\right)$$

כאשר נגדיר:

$$\text{legal}\left(i,j\right)=\bigvee_{s_1,\ldots,s_6\in S}X_{i,j,s_1}\wedge X_{i+1,j,s_2}\wedge X_{i+2,j,s_3}\wedge X_{i,j+1,s_4}\wedge X_{i+1,j+1,s_5}\wedge X_{i+2,j+1,s_6}$$

באופן מפורט יותר, ניזכר בדרך שבה קונפיגורציות עוברות. אם הסמן נמצא בתוך קונפיגרוציה מסוימת $w_iq_tw_j$, אזי הוא יכול לעבור שמאלה או ימינה במעבר הבא, כלומר $w_iw_kq_t$ למשל, אך אם הוא לא נמצא, עלינו **להעתיק את הקונפיגורציה** בחלון, לקומה הבאה (מלבד מקרה של 'ימינה', שיש בקצה השמאלי, ו'שמאלה', שיש בקצה הימני).
למשל, זהו חלון חוקי:

a	b	c
a	b	c

וזהו חלון לא חוקי:

a	q_0	c
a	b	c

טענה
יש קבוצה W של חלונות חוקיים, בגודל פולינומיאלי בקלט (ב$|w|$), כאשר:

$$\varphi_{\text{move}} = \bigwedge_{\substack{1 \le i \le t(n)+1 \\ 1 \le j \le t(n)+1}} legal\,(i,j)$$

הוכחה

לכל $a,b,c,d,e \in \Gamma$:

לכל מעבר, $(q_2,b,R) \in \delta\,(q_1,a)$ נוסיף חלונות:

d	c	b
d	c	q_1

q_2	c	d
a	c	d

c	b	q_2
c	q_1	a

b	q_2	c
q_1	a	c

לכל מעבר $(q_2,b,L) \in \delta\,(q_1,a)$ נוסיף חלונות:

c	b	d
q_1	a	d

e	d	q_2
e	d	c

d	q_2	c
d	c	q_1

q_2	c	b
c	q_1	a

הטיפול ב-# מתבצע באמצעות הוספת אפשרות ל-# בחלונות שבהם אין שינוי במצב.

מדובר בזמן פולינומיאלי כי לכל מעבר מוסיפים מספר קבוע (תלוי ב-$|\Gamma|$ של חלונות אבל $|\Gamma|$ קבוע)).

לא הראינו כי q_{move} ב-CNF, ניתן לעשות זאת אך **לא נראה זאת בשלב זה**.

<u>נכונות</u>

אם $w \in L$ יש ריצה מקבלת של M על w ולכן יש דרך למלא את המטריצה באופן שמייצג סדרת קונפ' שהיא ריצה חוקית מקבלת ולכן φ ספיקה.

אם φ ספיקה, אז' φ_{cell} מתארת מטריצה ו-φ_{init} גוררת כי יש בשורה הראשונה את הקונפ' ההתחלתית של M על w, ו-φ_{move} גוררת כי טיפוס במטריצה מתאים לקונפ' עוקבות, ו-φ_{acc} גוררת כי מגיעים לקונפ' מקבלת, ולכן $w \in L$.

כל שנותר לנו הוא להראות כיצד ניתן לעבור מ-CNF ל-3-CNF.

מעבר מ-CNF ל-3-CNF

בהינתן נוסחה φ ב-CNF, נרצה לייצר בזמן פולינומיאלי נוסחה φ' ב-3-CNF, כך ש-φ ספיקה אם"ם φ' ספיקה.

<u>בנייה</u>

לכל פסוקית c_j, אם $n_j = 3$, הפסוקית נשארת. אם $n_j < 3$, מכפילים ליטרלים, כלומר $x_1 \vee x_2 \rightarrow x_1 \vee x_1 \vee x_2$.

אם $n_j > 3$, מוסיפים משתני עזר: פסוקית עם n_j ליטרלים תומר ב-n_{j-2} פסוקיות עם $n_j - 3$ משתנים חדשים:

$$\begin{aligned} & a_1 \vee a_2 \vee a_3 \vee \ldots \vee a_q \rightarrow \\ & (a_1 \vee a_2 \vee z_1) \wedge (\overline{z_1} \vee a_3 \vee z_2) \wedge \ldots \wedge (\overline{z_{q-3}} \vee a_{q-1} \vee a_q) \end{aligned}$$

אם כך, הראינו כי ניתן לעבור מתצוגה כללית של CNF ל-3-CNF, בזמן פולינומיאלי, ולכן הוכחנו את המשפט, כנדרש.

3.3.2 הוכחת NP-קשה באמצעות רדוקציה פולינומיאלית

הרצאה מס' 20:

יום רביעי

15.12.21

הגדרה
יהי $G = \langle V, E \rangle$ גרף לא מכוון. קבוצה $S \subseteq V$ תקרא בלתי תלוייה אם לכל $v_1, v_2 \in S$ יתקיים כי $(v_1, v_2) \in E$.

כעת, נגדיר את השפה:

$$\text{IS} = \{ \langle G, K \rangle \mid 1 \leq k, k \text{ בגודל IS } G\text{-גרף לא מכוון, יש ב} G \}$$

ניתן להראות כי קיימת רדוקציה ISCLIQUE $\leq_p$. כדי להראות כי L' היא NP-קשה מספיק להראות כי ישנה רדוקציה פולינומילאית בינה ובין שפה ב-NP-קשה. הרדוקציה קיימת כיוון ש-S הוא קליקה ב-G אם"ם S הוא IS ב-G', גרף הקשתות המשלימות.

רדוקציה מ-CLIQUE ל-3-SAT להוכחת NP-קשה

נראה כי 3-SAT$\leq$CLIQUE באמצעות רדוקציה פולינומיאלית.
נתונה $\varphi \in$ 3-CNF כך ש-$\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$. לכל פסוקית C_j יש לכל היותר $F_j = \left\{ f_j^1, f_j^2, \ldots, f_j^7 \right\}$ השמות חלקיות למשתני C_j שמספקות את C_j) לכל היותר, כי ייתכן שבפסוקית אחת מופיעה אותו משתנה פעמיים).
נגדיר $G = \langle V, E \rangle$ כך ש-$V = \bigcup\limits_{1 \leq j \leq m} F_j$ (כל השמה חלקית היא קודקוד) - ויש לכל היותר $7m$ קודקודים.,
כמו כן, $\left(f_{j_1}^{i_1}, f_{j_2}^{i_2} \right) \in E$ אם ורק אם $f_{j_1}^{i_1}$ ו-$f_{j_2}^{i_2}$ מסכימות על המשתנים המשותפים (אין קשתות בין $f_i^{j_1}$ ו-$f_i^{j_2}$ ואין קשתות אם באחד יש משתנה ובשני שלילתו). בנוסף, יתקיים כי $K = m$.
הרדוקציה פולינומיאלית (כי השתמשנו רק בדברים פולינומיאלים בגודל הקלט).
<u>נכונות</u>

1. אם φ ספיקה, אז יש ב-G K-קליקה, כי אם φ ספיקה על ידי $f : X \to \{0, 1\}$, אזי לכל $1 \leq j \leq m$ יש ליטרל ש-f מספקת, ולכן יש i כך ש-f_j^i מסכימה עם f, כלומר בקבוצה F_j יש קודקוד שמסכים עם f ולא סותרים אחד את השני, שניקח אותו לקליקה. מדובר בקליקה, כי כל הקודקודים שנבחרו מסכימים עם f ולכן מסכימים זה עם זה ויש קשת ביניהם.

2. אם יש ב-G K-קליקה, ולכן φ ספיקה, כי ב-K-קליקה חייב להיות נציג אחד מכל קבוצה F_j (אין קשתות בין ההשמות החלקיות שמתאימות לאותה פסוקית), ולכן יש קשתות בין כל הנציגים ואין סתירות בהשמות החלקיות, ולכן האיחוד של ההשמות החלקיות משרה השמה מספקת.

אפשר לעשות גם רדוקציה הפוכה, 3-SAT $\leq_p$ CLIQUE, שונה מזו שראינו אצל קוק-לוין.
בניסיון ראשון: ננסה להגדיר לכל $i \in V$ משתנה X_i(כך ש-$X_i = \mathbb{T}$ אם ורק אם $i \in S$). אמנם, הקושי הוא למצוא כי הקליקה הוא בגודל K, ולכן הרדוקציה במקרה זה לא תהיה פולינומיאלית (עלינו לעשות choose שהוא לא פולינומיאלי).
בניסיון שני: לכל קודקוד $X \in V = \{1, \ldots, n\}$ בין $1 \leq j \leq K$, כך ש-$X_{ij} = \mathbb{T}$ אם קודקוד V_i נמצא בקליקה בתור הקודקוד ה-j בקליקה.
מתקיים כי לכל $1 \leq j \leq K$ יש בדיוק $1 \leq i \leq n$ כך ש-$X_{ij} = \mathbb{T}$. כלומר, יתקיים כי $\bigwedge\limits_{1 \leq j \leq K} \bigvee\limits_{1 \leq i \leq n} X_{ij}$ - כלומר יש לפחות i אחד שנבחר, ומצד שני $\bigwedge\limits_{1 \leq j \leq K} \bigwedge\limits_{1 \leq i_1, i_2 \leq n} \overline{X_{i_1 j}} \vee \overline{X_{i_2 j}}$ - כלומר יש רק אחד.
אם $X_{i_1 j_1} = \mathbb{T}$ ו-$X_{i_2 j_2} = \mathbb{T}$, אזי יש קשתות בין i_1 ו-i_2:

$$\bigwedge_{(i_1,i_2)\in(V\times V)\setminus E}\ \bigwedge_{1\le j_1\le j_2\le K} \overline{X_{i_1j_1}} \vee X_{i_2j_2}$$

כלומר, לא יכול להיות ש-$X_{i_1j_1}$ ו-$X_{i_2j_2}$ נבחרו למרות שיש ביניהם קשת.

הרצאה מס' 21:

יום שני

20.12.21

3.3.3 בעיית Subset-Sum

קלט: קבוצה $A=\{a_1,a_2,\dots,a_n\}\in\mathbb{N}$ ומספר יעד $s\in\mathbb{N}$.

פלט: האם יש $B\subseteq A$ כך ש-$\sum\limits_{i\in B} i=s$.

דוגמה

$A=\{5,2,10,4,7\}$ ו-$s=16$ - אפשרי כי $2+10+4$.

עבור $s=20$, אין דוגמה כזו, ואין מנוס מלעבור על כל האפשרויות.

טענה

השפה SubsetSum שטוענת כי קיים $B\subseteq A$ כך ש-$\sum\limits_{i\in B} i=s$ הינה שפה NP-קשה.

הוכחה

קודם כל, נבחין כי הינה ב-NP:

מוודא פולינומיאלי עבור SS הינו:

$$V=\left\{\langle A,s,B\rangle \mid B\subseteq A\ \ \sum_{i\in B} i=s\right\}$$

דבר זה נכון בין אם המספרים ב-A בבינארית, ובין אם נתונים באונרית.

נראה כעת קושי ב-NP. כלומר, נראה כי מדובר בשפה NP-קשה באמצעות רדוקציה 3SAT$\le_P$SS.

בהינתן φ ב-3-CNF, נבנה (בזמן פולינומיאלי) קבוצה A ומספר יעד s, כך ש-φ ספיקה אם"ם $(A,s)\in$ SS.

<u>בנייה</u>

נניח כי φ מעל n משתנים, כך ש-$X=\{x_1,\dots,x_n\}'$, ושיש בה m פסוקיות, כך שמתקיים כי $\varphi=c_1\wedge c_2\dots\wedge c_m$.

- כל משתנה x_i כך ש-$1\le i\le n$, משרה שני מספרים, t_i ו-f_i.
- כל פסוקית c_j, כך ש-$1\le j\le m$, משרה שני מספרים p_j ו-q_j.

שני המספרים יהיו בבסיס 10.

מילוי הטבלה

עבור המשתנה x_i:

- הספרה ה-i-ית של t_i ו-f_i מסומנת ב-1. כל שאר הספרות הן 0.
- הספרה $n+j$ עבור $1\le j\le m$:

- ב-t_i:
  - 1 אם x_i מופיע ב-c_j.
  - 0 אם x_i לא מופיע ב-c_j.
- ב-f_i:
  - 1 אם $\overline{x_i}$ מופיע ב-c_j.
  - 0 אם $\overline{x_i}$ לא מופיע ב-c_j.

עבור המשתנה c_i:

- הספרה ה-$n+j$ של p_i ו-q_j היא 1. כל שאר הספרות הן 0.
- הספרה ה-i-ית של p_j וq_j, לכל $1 \le i \le n$, היא 0.

מספר היעד s:

- עבור $1 \le i \le n$, הספרה ה-i היא 1, עבור $1 \le j \le m$, הספרה ה-$n+j$ היא 3.

הרדוקציה פולינומיאלית (שהרי ממלאים טבלה בגודל $O\left((n+m)^2\right)$.
הרדוקציה נכונה:

- אם φ ספיקה, אזי תהי $f_i : X \to \{\mathbb{F}, \mathbb{T}\}$, השמה שמספקת את φ. נבחר את B כך: לכל $1 \le i \le n$, אם $f(x_i) = \mathbb{T}$, נוסיף ל-B את t_i. אם $f(x_i) = \mathbb{F}$, נוסיף ל-B את f_i.
  - לכל $1 \le j \le m$, נתבונן בפסוקית c_j.
    - אם f מספקת ליטרל אחד, נוסיף ל-B את p_j ו-q_j.
    - אם f מספקת שני ליטרלים, נוסיף ל-B את p_j.
    - אם f מספקת שלושה ליטרלים, לא נוסיף את p_j או את q_j.

    (אין מצב ש-f לא מספקת אף ליטרל)
- אם קיים B כך ש-$\sum\limits_{i \in B} i = s$, אזי נשים לב כי B מכיל בדיוק אחד מתוך t_i ו-f_i לכל $1 \le i \le n$, ולכן הוא משרה השמה. תת הקבוצה של B שמגיעה מ-$\{p_1, q_1, \ldots, p_m, q_m\}$, תורמת לכל היותר ספרה $n+j$-ית, עבור $1 \le j \le m$, לכל היותר 2, ולכן בחרנו לפחות שורה אחת עם ליטרל שמסתפק בכל פסוקית, ולכן ההשמה מספקת.

□

דוגמה

ניקח את הנוסחה הבאה:

$$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee x_3)$$

	1	2	3	1	2	3	4
t_1	1	0	0	1	0	0	1
f_1	1	0	0	0	1	1	0
t_2	0	1	0	1	0	1	0
f_2	0	1	0	0	1	0	1
t_3	0	0	1	1	1	0	1
f_3	0	0	1	0	0	1	0
q_1	0	0	0	1			
p_1	0	0	0	1			
q_2	0	0	0		1		
p_2	0	0	0		1		
q_3	0	0	0			1	
p_3	0	0	0			1	
q_4	0	0	0				1
p_4	0	0	0				1

s יוגדר על ידי $s = \underbrace{\underline{111}}_{n}, \underbrace{\underline{3333}}_{m}$, בהינתן השמה של הנוסחה, למשל $x_1 \to \mathbb{F}, x_2 \to \mathbb{T}, x_3 \to \mathbb{T}$, מילאנו את הטבלה כך.

3.4 סיבוכיות זיכרון

נרצה לבדוק מהו שטח הנדרש - מהו צוואר הבקבוק שלא ניתן לפתור על ידי סבלנות.

הגדרה

בהינתן מ"ט חד-סרטית M, העוצרת על כל קלט, סיבוכיות הזיכרון של M: פונקציה $S : \mathbb{N} \to \mathbb{N}$ כך ש-$S(n)$ הוא מספר התאים ש-M משתמשת בהם בריצתה על מילה באורך n.

הגדרה

עבור $n \le S(n)$ נגדיר את $\text{SPACE}(S(n))$ להיות

$$\{L \mid \text{יש מכונה } M \text{ דטר' שמכריעה את } L \text{ בסיבוכיות זיכרון } O(S(n))\}$$

הגדרה

עבור $n \le S(n)$ נגדיר את $\text{NSPACE}(S(n))$ להיות

$$\{L \mid \text{יש מכונה } M \text{ ייתכן, דטר' שמכריעה את } L \text{ בסיבוכיות זיכרון } O(S(n))\}$$

חשוב לציין כי ההערה $n \le S(n)$ תבוא לידי ביטוי יותר בהמשך.

3.4.1 קשרים בין סיבוכיות זמן ומקום

טענה

לכל $f(n)$, מתקיים כי $\text{TIME}(f(n)) \subseteq \text{SPACE}(f(n))$.

הוכחה

מכונה שעוצרת תוך $f(n)$ צעדים, לא יכולה להשתמש ביותר ב-$f(n)$ תאים.

טענה

לכל $f(n)$ יתקיים כי $\text{SPACE}(f(n)) \subseteq \text{TIME}\left(2^{f(n)}\right)$.

הוכחה

למ"ט דטרמיניסטית עם סיבוכיות זיכרון $S(n)$ ישנם - $\underbrace{|Q|}_{\text{מצב}} \cdot \underbrace{|\Gamma|^{S(n)}}_{\text{א"ב העבודה}} \cdot \underbrace{S(n)}_{\text{מיקום הראש הקורא}}$.

זה מספיק לנו, כי מדובר בחסם על זמן הריצה, שהרי M לא חוזרת על אותה קונפ' פעמיים.

כעת יתקיים כי:

$$c_1 \cdot c_2^{S(n)} \cdot S(n) = c_1 \cdot 2^{S(n)\log c_2} \cdot 2^{\log S(n)} = S^{O(S(n))}$$

דוגמה

נראה כי SAT ניתנת להכרעה בשטח ליניארי.

הרעיון: נעבור כל השמות האמת האפשריות, אם נגיע להשמה מספקת, נעצור ונקבל. אם נסיים את המעבר, נעצור ונדחה.

תהי φ נוסחה מעל $X = \{x_1, \ldots x_n\}$. יהי $f_0, f_1, \ldots, f_{2^n-1}$ סידור של השמות אפשריות ל-X, כזה שיש פונקציה $g : F \to F \cup \{1\}$, כך שבהינתן f_i: אם $i < 2^n - 1$, אזי $g(f_i) = f_{i-1}$, ואם $i = 2^{n-1}$, אזי $g(f_i) = 1$.

M שמכריעה את SAT בשטח ליניארי:

1. כותבת על הסרט את f_0.
2. אם ההשמה f שכתובה על הסרט היא 1, עוצרת ודוחה.
3. משערכת את φ לפי ההשמה שכתובה על הסרט (f) - בכל האיטרציות משתמשת באותן תאים לפעולת השערוך. אם φ מסתפקת, עוצרת ומקבלת. אחרת / $f = g(f)$(מעדכנת את ההשמה הנבדקת) וחוזרת ל-2.

הזכרון הדרוש הוא:

- לחישוב g, נדרש מקום ליניארי ב-n.
- לפעולת השערוך, נדרש מקום ליניארי ב-$|\varphi|$.

סך הכל, קיבלנו מספר ליניארי בקלט של תאים.

דוגמה נוספת

ראינו כי $\text{VAL} = \{\langle\varphi\rangle \mid \varphi \text{ כל השמה מספקת את}\}$ לא שייכת ל-NP. מצד שני, הדרך שראינו מקודם תעבור גם על VAL. כלומר, ניתנת לחישוב במקום פולינומיאלי בקלט.

הרצאה מס' 22:

יום רביעי

22.12.21

משפט

מתקיים כי NP⊆PSPACE.

הוכחה

תהי $L\in \mathrm{NP}$. אם כך, בהכרח קיים פולינום p ומוודא V כך ש-

$L=\{w\mid (w,c)$ קיים עד c באורך לכל היותר $p(|w|)$ כך ש-V מקבל את $\}$.

תהי מ"ט M שמכריעה את L בשטח פולינומיאלי: נניח כי העדים הם מעל הא"ב Σ'.

בהינתן w, המכונה M עוברת על כל המילים c מעל Σ' באורך לכל היותר $p(|w|)$, מריצה את V על (w,c). אם V קיבל, אזי M מקבלת. אם V לא קיבל, מוחקת את הסרט, חוזרת עם הראש הקורא לתחילת האזור של הרצת V, ועוברת לעד העוקב. אם עברנו על כל העדים האפשריים, עוצרת ודוחה.

נבחין כי הזיכרון של M הינו:

- המנגנון לשמירה על c ומעבר לעד העוקב לוקח $O(p(|w|))$ תאים.
- הרצה של V - מכיוון ש-V עובדת בזמן פולינומיאלי, גם השטח הנדרש להרצה שלה הוא פולינומיאלי.

3.4.2 ניתוח מקום של $\mathrm{EMPTY_{NFA}}$

נגדיר את השפה -

$\mathrm{EMPTY_{NFA}}=\{\langle A\rangle\mid L(A)=\emptyset$ הוא NFA ו-$A\}$ וגם את השפה:

$\overline{\mathrm{EMPTY_{NFA}}}=\{\langle A\rangle\mid L(A)\neq\emptyset$ הוא NFA ו-$A\}$.

נבחין כי שתיהן ב-P. מדוע? $L(A)\neq\emptyset$ אם ורק אם בגרף של A יש מסלול מ-Q_0 ל-F, שמדובר בזמן פולינומיאלי.

כיון ש-P סגורה למשלים, אזי גם השפה השנייה היא ב-P.

אם כך, היא בהכרח ב-PSPACE.

3.4.3 ניתוח מקום של $\mathrm{ALL_{NFA}}$

נגדיר את $\mathrm{ALL_{NFA}}=\{\langle A\rangle\mid L(A)=\Sigma^*$ הוא NFA ו-$A\}$ וגם את השפה:

$\overline{\mathrm{ALL_{NFA}}}=\{\langle A\rangle\mid L(A)\neq\Sigma^*$ הוא NFA ו-$A\}$. שפה זו למעשה אומרת כי יש מילה ש-A לא מקבל.

נבחין כי $L(A)=\Sigma^*$ אם ורק אם $L(\overline{A})=\emptyset$.[7]

אלגוריתם ב-EXPTIME להכרעת $\mathrm{ALL_{NFA}}$:

- בהינתן אוטומט NFA A, נבנה את $\overline{A}$ (אם ל-A יש n מצבים, אזי ל-$\overline{A}$ יש 2^n מצבים).
  נבדוק ריקנות של $\overline{A}$. אנו יודעים כי $L(A)=\Sigma^*$ אס"ם $L(\overline{A})=\emptyset$, ולכן נענה בהתאם.

האם $\overline{\mathrm{ALL_{NFA}}}\in\mathrm{NP}$?

אם היינו יודעים כי "אם A הינו NFA ו-$L(A)\neq\Sigma^*$, אזי יש מילה w כך ש-$w\notin L(A)$ ו-$|w|$ פולינומיאלי ב-$|A|$",

אזי היינו יודעים כי $\overline{\mathrm{ALL_{NFA}}}\in\mathrm{NP}$.

בהינתן A שהינו NFA ומילה w, לכאורה, ניתן לבדוק בזמן פולינומיאלי אם $w\in L(A)$. נשים לב כי $w\in L(A)$ אס"ם $L\left(\underbrace{A_w}_{\text{אוטומט שמקבל רק את } w}\right)\cap L(A)\neq\emptyset$. את זה ניתן לעשות באמצעות אוטומט מכפלה $L(A_w\times A)$ ובדיקתו האם הוא ריק (בניית אוטומט מכפלה לוקחת זמן פולינומיאלי בקלט).

[7] $\overline{A}$ הוא NFA כזה ש-$L(\overline{A})=\Sigma^*-L(A)$ וכיוון ש-A הוא א"ד, לא מדובר ב-$\tilde{A}$.

אבל **האמת היא שזה לא אפשרי** (ונראה דוגמה נגדית).

מה שאנחנו כן יודעים, אנו יודעים כי אם $L(A) \neq \Sigma^*$, אזי יש מילה באורך לכל היותר 2^n (אקספוננציאלי) כזו ש-$w \notin L(A)$ ול-A יש n מצבים. למה? כי המילה w מתקבלת ב-$\overline{A}$: כלומר, יש מסלול פשוט מ-q_0 ל-F ב-$\overline{A}$, שמתקבל מהפעלת ה-SubsetConstruction, שלו יש 2^n מצבים.[8]

[8]כאן היה חלק של העשרה, שלא הושלם ולכן לא הבאתי אותו במלואו.

חלק II

תרגולים

תרגול מס' 1:

יום שני

11.10.21

(בדר)

1 מודלים חישוביים

1.1 חזרה על תורת הקבוצות

1. קבוצה היא אוסף של איברים, כך שלכל איבר x וקבוצה A מתקיים כי $x \in A$ או $x \notin A$ אך לא שניהם.

2. פעולות על קבוצות:

   - **האיחוד** הוא כל האיברים שנמצאים ב-A או ב-B:
   $A \cup B = \{x : x \in A \;\; \vee \;\; x \in B\}$
   - **החיתוך** הוא כל האיברים שנמצאים ב-A וגם ב-B.
   $A \cap B = \{x : x \in A \wedge x \in B\}$.
   - **המשלים** $\overline{A}$ הוא קבוצת כל האיברים שלא נמצאים ב-A.
   - **ההפרש** $A \setminus B$ הוא קבוצת כל האיברים שנמצאים ב-A ולא ב-B:
   $A \setminus B = \{x : \;x \in A \wedge x \notin B\}$.
   - **הכלה** $A \subseteq B$: נאמר כי A מוכלת ב-B אם:
   $\forall x \;\; x \in A \Rightarrow x \in B$.
   - **שוויון**: נאמר כי A שווה ל-B אם"ם:
   $A \subseteq B \wedge B \subseteq A$.

3. קבוצת **החזקה** מוגדרת על ידי $2^A = P(A) = \{B \mid B \subseteq A\}$.

4. המכפלה הקרטזית $A \times B$ היא קבוצות כל הזוגים הסדורים מ-A ו-B:
$A \times B = \{(a,b) : a \in A, b \in B\}$
דוגמה: $\{1,2\} \times \{a,b,c\} = \{(1,a),(2,a),(1,b),(2,b),(1,c),(2,c)\}$
אנו מרשים גם מכפלה קרטזית ארוכה, כך שכל אחד מהאלמנטים הוא סדרה בפני עצמו. למשל:
$A_1 \times A_2 \times \ldots \times A_n = \{(a_1,\ldots,a_n) : \forall i, a_i \in A_i\}$.

5. יחס מעל קבוצה A הוא תת הקבוצה $R \subseteq A \times A$.

   - R הוא רפלקסיבי, אם $\forall a \in A, (a,a) \in R$.
   - R הוא סימטרי, אם $\forall a,b \in A$ מתקיים כי $(a,b) \in R \rightarrow (b,a) \in R$.
   - R הוא טרנזיטיבי, אם $(a,b) \in R, \; \wedge (b,c) \in R \rightarrow (a,b) \in R$.
   דוגמה: $A = \{1,2,3,4\}$ ו-$R = \{(a,b) \in A \times A : |a-b| \leq 1\}$ -
   רפלקסיבי, סימטרי, אך לא טרנזיטיבי ($(1,2) \in R, (2,3) \in R, (1,3) \notin R$)

6. **העוצמה** של קבוצה, היא **המדד** של כמה איברים יש בקבוצה.
אם מדובר בקבוצה אינסופית, נאמר כי קבוצות שוות בעוצמתם, אם קיימת פונקציה חח"ע ועל ביניהם.
נאמר כי $|A| < |B|$ אם קיימת פונקציה כנ"ל ואין פונקציה על.
הערה
העוצמה של $\mathbb{N}$ הינה $\aleph_0$, והיא שווה גם לעוצמת $\mathbb{Q}$ ו-$\mathbb{Z}$. כלומר, קיימת העתקה חח"ע ועל בין הטבעיים ובין הרציונליים.
מאידך, $\left|2^{\mathbb{N}}\right| = 2^{\aleph_0} = |[0,1]|$ - כלומר, יש העתקה חח"ע מ-$\mathbb{N}$ ל-$2^{\mathbb{N}}$ אבל אין העתקה על ביניהם.[9]

1.2 מבוא לשפות

1.2.1 הגדרות ופעולות

תזכורת להגדרות:

1. **אלפבית** (א"ב) הוא קבוצה סופית ולא ריקה של אותיות $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$.
דוגמה: $\Sigma = \{a, b\}$.
לכל $n \geq 1$ נתבונן ב-Σ^n.
נגדיר את Σ^0 בתור כל המילים באורך 0, דהיינו המילה הריקה $Z^0 = \{\varepsilon\}$.
נגדיר את Σ^* בתור אוסף כל המילים הסופיות מעל Σ, כלומר $\Sigma^\star = \bigcup\limits_{n=0}^{\infty} \Sigma^n$.
דוגמה: למשל $\Sigma^2 = \{(a,b), (b,a,), (b,b), (b,a)\}$

2. **מילה** היא סדרה סופית של אותיות, המילה הריקה תסומן בתור ε.
כיוון ש-Σ^* מכילה את כל המילים בעלות אורך סופי, היא סגורה לשרשור.
דוגמה: $abb \cdot bab = abbbab$.

3. **שפה** מעל א"ב Σ היא קבוצה $L \subseteq \Sigma^*$.
דוגמאות:
   - $L_1 = \{\varepsilon, a, aa, b\}$.
   - $L_2 = \{w \mid a \text{ מתחילה עם } w\}$.
   - $L_3 = \{\varepsilon\}$.
   - $L_4 = \emptyset$.
   - $L_5 = \{w \mid |w| < 24\}$.

4. ניתן לבצע על שפות כל פעולה שאפשר לבצע על קבוצות וגם **שרשור,** שמוגדר על ידי:
$L_1 \cdot L_2 = \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$.
דוגמה: $L = \{ww \mid w \in \Sigma^*\}$
$\overline{L} = \left\{x = x_1, \ldots x_m \mid x_1 \cdots x_{n/2} \neq x_{n/2+1} \cdots x_n \text{ וגם } n\text{-זוגי או ש} n \text{ אי זוגי}\right\}$.
$L \cdot \overline{L} = \{wwxx \mid w \in \Sigma^*, x \in \Sigma^*\}$.

[9] בהמשך, נוכיח דבר זה, במה שייקרא **משפט החיסור.**

עוצמות ושפות

אם נחזור לרגע לעוצמות וקבוצות, נוכל לשאול: "כמה מילים יש ב-Σ^*?"
התשובה לכך היא $\aleph_0$ וניתן להראות זאת באמצעות סידור המילים לפי האורך ובסדר לקסיקוגרפי.
נוכל גם לשאול "כמה שפות יש?" ונקבל: $\left|2^{\Sigma^*}\right| = \left|2^{\mathbb{N}}\right| = 2^{\aleph_0}$.
אם כך, כיוון ש-$2^{\aleph_0} > \aleph_0$, עולה כי יש יותר שפות מעל Σ מאשר מילים מעל Σ, כלומר, אין ההעתקה חח"ע ועל בין מספר השפות למספר המילים.

האם כל השפות הן רגולריות?

כפי שאנחנו יכולים לשער, התשובה היא לא. הסיבה לכך היא שיש $2^{\aleph_0}$ שפות, ורק $\aleph_0$ שפות רגולריות.
כלומר, כיוון שכל DFA ניתן לייצוג באמצעות מספר בינארי סופי, יש בסך הכל מספר בן מנייה של שפות.
אמנם, זו אינה הוכחה קונסטרוקטיבית, ובהמשך נראה כיצד אפשר להראות ששפות מסוימות **אינן רגולריות**.

1.2.2 הפונקציה δ^*

נתבונן באוטומט כלשהו, ונניח כי כעת האוטומט $\mathcal{A}$ הוא במצב q, והאות שנקרא כעת תהיה σ. המצב הבא ניתן לתיאור באמצעות הפונקציה שראינו, $\delta : Q \times \Sigma \to Q$, ויהיה למעשה $\delta\left(q, \sigma\right)$.
על מנת להרחיב את אוצר הכלים שלנו ולאפשר התייחסות רקרוסיבית לקריאת מספר אותיות, נגדיר כעת פונקציה חדשה, ש'מחברת' בין מצבים ומילים.

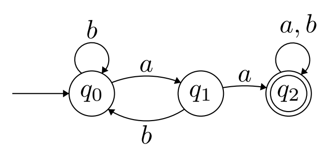
הגדרה

יהי $A = \langle Q, \Sigma, \delta, q_0, F\rangle$. נגדיר את **פונקציית המעברים המוכללת** δ^* להיות:

$$\delta^*\left(q, w\right) = \begin{cases} q & w = \varepsilon \\ \delta\left(\delta^*\left(q, w'\right), \sigma\right) & w = w' \cdot \sigma,\ w' \in \Sigma^*, \sigma \in \Sigma \end{cases}$$

דוגמה

נתבונן באוטומט $\mathcal{A}$ הבא:



נניח כי אנחנו נמצאים ב-q_1 ונרצה לקרוא את המילה ba. לפי ההגדרה לעיל, נקבל:

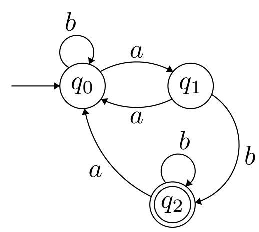
$$\delta^*\left(q_1, ba\right) = \delta\left(\delta^*\left(q_1, b\right), a\right) = \delta\left(\delta\left(\delta^*\left(q_1, \epsilon\right), b\right), a\right) = \delta\left(\delta\left(q_1, b\right), a\right) = \delta\left(q_0, a\right) = q_1$$

אבחנה

בהינתן Q ו-Σ, לא כל פונקציה $Q \times \Sigma^* \to Q$ היא פונקציה δ^* של DFA כלשהו. עלינו קודם כל להגדיר את ה-δ שלו, ורק לאחר מכן נקבל את ה-δ^* שנגזרת ממנו באופן ישיר.
אם נבחר את δ^* באופן ישיר מההגדרה, עלולה להתקבל הגדרה לא תקינה.

1.2.3 הוכחת שפה של אוטומט

בכיתה הראינו מספר DFA-ים והצבענו על השפה שלהם, אך לא הוכחנו זאת פורמלית. בחלק זה נראה, באופן חד פעמי, כיצד יש להוכיח זאת פורמלית, ומכאן והלאה ניתן יהיה להשתמש בהסבר בלבד.
נתבונן באוטומט $\mathcal{A}$ מעל $\Sigma = \{a, b\}$, כך שהשפה מורכבת ממילים בעלות מספר אי זוגי של a, שמסתיימות ב-b.
דהיינו, פורמלית: $\mathcal{A} = \langle \{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\} \rangle$, ובציור:



לפני שניגש להוכחה הפורמלית, ננסה לזהות מספר מצבים בצורה אינטואיטיבית, כך למשל, אם נקרא a מ-q_0, נלך למצב אחר, ואם נקרא a ממצב אחר, נחזור ל-q_0. כך נוכל 'לספור' את מספר האותיות, ולזכור אותם לפי המצבים.

טרמינלוגיה
עבור $w \in \Sigma^*$ ואות $\sigma \in \Sigma$, נסמן את מספר ההופעות של σ בתור $\#_\sigma (w)$.

טענה
לכל $w \in \Sigma^*$, מתקיים:

1. $\delta^* (q_0, w) = q_0$ אם"ם $\#_a (w)$ הוא זוגי.
2. $\delta^* (q_0, w) = q_1$ אם"ם $\#_a (w)$ הוא אי זוגי ו-w מסתיים ב-a.
3. $\delta^* (q_0, w) = q_2$ אם"ם $\#_a (w)$ הוא אי זוגי ו-w מסתיים ב-b.

הוכחה
נוכיח זאת באינדוקציה על $|w|$.

<u>בסיס האינדוקציה, עבור $|w| = 0$:</u>
$\delta^* (q_0, w) = d^* (q_0, \varepsilon) = q_0$, ואכן $\#_a (w) = 0$, כלומר זוגי.

<u>צעד האינדוקציה:</u>

נניח כי $|w| > 0$ ולכן w יכול להיכתב בתור $w = u\sigma$ כאשר $u \in \Sigma^*$ ו-$w \in \Sigma$.
נניח באינדוקציה מלאה כי הטענה נכונה לכל $|u| < |w|$ ונוכיח עבור $|w|$.
נחלק למקרים:

- אם $\sigma = a$:

1. $\delta^*(q_0, w) = q_0 \Leftrightarrow \delta^*(q_0, ua) = q_0 \Leftrightarrow \delta(\delta^*(q_0, u), a) = q_0 \Leftrightarrow \delta^*(q_0, u) \in \{q_1, q_2\}$.
מצעד האינדוקציה, עולה כי $\#_a(u)$ הוא אי זוגי. בתוספת a, נקבל כי $\#_a(ua)$ הוא זוגי.
2. $\delta^*(q_0, w) = q_1 \Leftrightarrow \delta^*(q_0, ua) = q_1 \Leftrightarrow \delta(\delta^*(q_0, u), a) = q_1 \Leftrightarrow \delta^*(q_0, u) = q_0$.
מצעד האינדוקציה, עולה כי $\#_a(u)$ הוא זוגי. בתוספת a, נקבל כי $\#_a(ua)$ הוא אי זוגי ו-w מסתיימת ב-a.
3. $\delta^*(q_0, w) = q_2 \Leftrightarrow \delta^*(q_0, ua) = q_2 \Leftrightarrow \delta(\delta^*(q_0, u), a) = q_2 \Leftrightarrow \delta^*(q_0, u) = q_2$.
כיוון של-q_2 אין מעברים נכנסים עם a, אין אפשרות לכך.

- אם $\sigma = b$:

1. $\delta^*(q_0, w) = q_0 \Leftrightarrow \delta^*(q_0, ub) = q_0 \Leftrightarrow \delta(\delta^*(q_0, u), b) = q_0 \Leftrightarrow \delta^*(q_0, u) = q_0$.
מצעד האינדוקציה, עולה כי $\#_a(u)$ הוא אי זוגי. נקבל כי $\#_a(ua)$ הוא אי זוגי.
2. $\delta^*(q_0, w) = q_0 \Leftrightarrow \delta^*(q_0, ub) = q_0 \Leftrightarrow \delta(\delta^*(q_0, u), b) = q_1$.
כיוון של-q_1 אין מצבים נכנסים עם b, אין אפשרות לכך.
3. $\delta^*(q_0, w) = q_2 \Leftrightarrow \delta^*(q_0, ub) = q_2 \Leftrightarrow \delta(\delta^*(q_0, u), b) = q_2 \Leftrightarrow \delta^*(q_0, u) \in \{q_1, q_2\}$
מצעד האינדוקציה, עולה כי $\#_a(u)$ הוא זוגי. נקבל כי $\#_a(ua)$ הוא אי זוגי ו-w מסתיימת ב-b.

אם כך, באמצעות הנחת האינדוקציה והצעד, הראינו כי הטענה נכונה לכל $|w|$.

1.3 אוטומט לא דטרמיניסטי (NFA)

תרגול מס' 2:

יום שני

18.10.21

(בדר)

תחילה, נזכיר את ההגדרה שראינו בכיתה:

הגדרה
אוטומט סופי לא דטרמיניסטי (NFA) הוא חמישיה $A = \langle Q, \Sigma, \delta, Q_0, F\rangle$ כאשר:

- Q קבוצה סופית של **מצבים**.
- Σ הוא **א"ב**.
- $Q_0 \subseteq Q$ קבוצה סופית של **מצבים התחלתיים**.
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ **פונקציית מעברים**.
- $F \subseteq Q$ קבוצת **מצבים מקבלים**.

ראינו בהרצאה כי קיימים NFA עם מעברי ε, שאותו אפשר לסמן בתור ε-NFA, כך שניתן לקפוץ בין מצבים.
ניתן לתרגם ε-NFA ל-NFA רגיל, ונוכיח זאת כעת, פורמלית.

טענה
כל ε-NFA ניתן להמרה ל-NFA.

הוכחה

לכל מצב $q \in Q$ נגדיר את:

$$E\left(q\right)=\{q'\in Q\mid q' \text{ ישיג מ-} q \text{ בשימוש רק במעברי } \varepsilon\}$$

כעת, נוכל להגדיר את $\mathcal{B}$ NFA:

$$\mathcal{B}=\left\langle Q,\Sigma,\eta,\bigcup_{q\in Q_0}E(q),F\right\rangle$$

כאשר $\eta\left(q,\sigma\right)=\bigcup_{s\in\delta(q,\sigma)}E\left(s\right)$ לכל $q\in Q$ ו-$\sigma\in\Sigma$.

בתרגיל בית נוכיח כי $L\left(\mathcal{B}\right)=L\left(\mathcal{A}\right)$. מעבר לכך, ניתן בזמן פולינומי לחשב את $\mathcal{B}$ בהינתן האוטומט $\mathcal{A}$.

1.3.1 תכונות סגור בשימוש ב-NFA

ראינו בשיעור כי שפות רגולריות סגורות לאיחוד. כעת נראה דרך נוספת, קלה יותר, שמשתמשת ב-NFA.

טענה

לכל שתי שפות רגולריות, L_1,L_2 מעל Σ מתקיים כי $L_1\cup L_2$ רגולרית.

הוכחה

יהיו $L_1,L_2\subseteq\Sigma^*$, אזי:

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

אנחנו רוצים להראות כי קיים NFA שמזהה את $L_1\cup L_2$.

יהיו $\mathcal{A}=\langle Q,\Sigma,q_0,\delta,F\rangle$ ו-$\mathcal{B}=\langle S,\Sigma,s_0,\eta,G\rangle$ אוטומטים תואמים, כך ש-$L\left(\mathcal{A}\right)=L_1$ ו-$L\left(\mathcal{B}\right)=L_2$. נניח בה"כ כי $S\cap Q=\emptyset$.

נגדיר כעת NFA חדש שיסומן בתור $\mathcal{C}$ ויוגדר על ידי $\mathcal{C}=\langle Q\cup S,\Sigma,\{q_0,s_0\},\alpha,F\cup G\rangle$, כאשר α הינו:

$$\alpha(q,\sigma)=\begin{cases}\{\delta(q,\sigma)\} & \text{if } q\in Q\\ \{\eta(q,\sigma)\} & \text{if } q\in S\end{cases}$$

על מנת להראות כי $L_1\cup L_2=L\left(\mathcal{C}\right)$, נשתמש בהכלה דו כיוונית.

<u>בכיוון הראשון $\Leftarrow$, נראה כי $L_1\cup L_2\subseteq L\left(\mathcal{C}\right)$:</u>

יהי $x=\sigma_1\cdot\sigma_2\cdot\ldots\cdot\sigma_m\in L_1\cup L_2$ כך שלכל $i\in[m]$ יתקיים כי $\sigma_i\in\Sigma$.

דהיינו, x היא מילה מאורך m ששייכת ל-$L_1\cup L_2$, כלומר, x שייכת ל-L_1 או x שייכת ל-L_2. נניח, בלי הגבלת הכלליות, כי $x\in L_1$.

נתבונן בריצה של $\mathcal{A}$ על x, שאנו יודעים שהיא ריצה מקבלת. כלומר:

- קיימת סדרת מצבים $r=r_0,r_1,\ldots,r_m$ כך ש-$r_0=q_0$ ו-$r_m\in F$.

□ לכל $0 \leq i < m$ יתקיים כי $\delta\left(r_i, \sigma_{i+1}\right) = r_{i+1}$.

כעת, אנו יודעים כי $r_0 = q_0 \in \{q_0, s_0\}$ וכמו כן $r_m \in F \subseteq F \cup G$ ולכן r היא ריצה מקבלת של $\mathcal{C}$ על x (=הן המצב ההתחלתי, הן המצב המקבל, והן שאר המצבים נמצאים ב-$\mathcal{C}$ ולכן מדובר בריצה מקבלת גם ב-$\mathcal{C}$).

<u>בכיוון השני $\Rightarrow$, נראה כי $L\left(\mathcal{C}\right) \subseteq L_1 \cup L_2$</u>:
יהי $y = \sigma_1 \cdot \sigma_2 \cdot \ldots \cdot \sigma_m \in L\left(\mathcal{C}\right)$ באורך m. אזי יש ריצה מקבלת r כלשהי של $\mathcal{C}$ על y.
כלומר, $r_0 \in \{q_0, s_0\}, r_1, \ldots, r_m \in Q \cup S,\ r_m \in F \cup G$ וגם $r_{i+1} \in \alpha\left(r_i, \sigma_{i+1}\right)$ לכל $0 \leq i < m$.
מההגדרה של $\mathcal{C}$, r_m שייכת ל-F או ל-G. לכן נניח בה"כ כי $r_m \in F$. כמו כן, מההגדרה של α מתקיים כי $Q \cap S = \emptyset$, ולכן $r_m \in F \setminus G$.
מכיוון ש-$\mathcal{C}$ משמר את המעברים של $\mathcal{A}$, כל המעברים האלו נמצאים ב-A ולכן נקבל כי $r_{m-1} \in Q$ וגם $\delta\left(r_{m-1}, \sigma_m\right) = r_m$.
מכאן ניתן להראות באינדוקציה כי $r_0 = q_0$ וכל המעברים בריצה r קיימים גם ב-$\mathcal{A}$.
אם כך, מדובר בריצה מקבלת של $\mathcal{A}$ על y, כלומר $y \in L_1 \subseteq L_1 \cup L_2$.

טענה

לכל שתי שפות רגולריות L_1, L_2 מעל Σ מתקיים כי $L_1 \cdot L_2$ רגולרית.

הוכחה

יהיו $L_1, L_2 \subseteq \Sigma^*$ שפות רגולריות. נתבונן ב-DFA-ים המתאימים להם:

$$\mathcal{A} = \langle Q, \Sigma, q_0, \delta, F\rangle$$
$$\mathcal{B} = \langle S, \Sigma, s_0, \eta, G\rangle$$

נתבונן ב-$L_1 \cdot L_2$ בתור:

$$L_1 \cdot L_2 = \{w = \sigma_1 \cdots \sigma_n : \exists 1 \leq k \leq n, \sigma_1 \cdots \sigma_k \in L_1, \wedge, \sigma_{k+1} \cdots \sigma_n \in L_2\}$$

נגדיר כעת את ה-NFA המתאים שהינו $\mathcal{C} = \langle Q \cup S, \Sigma, \{q_0\}, \alpha, G\rangle$, כאשר α מוגדר על ידי:

$$\alpha\left(q, \sigma\right) = \begin{cases} \{\delta\left(q, \sigma\right)\} & q \in Q \\ \{\eta\left(q, \sigma\right)\} & q \in S \end{cases}$$

בנוסף, לכל $q \in F$ נקבל כי $\alpha\left(q, \varepsilon\right) = \{s_0\}$ ולכל $q \in \left(Q \cup S\right) \setminus F$ נקבל כי $\alpha\left(q, \varepsilon\right) = \emptyset$.
נרצה כעת להוכיח כי $L\left(\mathcal{C}\right) = L_1 \cdot L_2$.

<u>בכיוון הראשון, $\Leftarrow$ נראה כי $L_1 \cdot L_2 \subseteq L\left(\mathcal{C}\right)$</u>:
תהי $w \in L_1 \cdot L_2$ ויהיו $u \in L_1$ ו-$v \in L_2$ כך ש-$w = u \cdot v$.
נתבונן בריצות של $\mathcal{A}$ ו-$\mathcal{B}$ על u ו-v בהתאמה. נסמן את הריצה של $\mathcal{A}$ על u ב-$r = a_0, a_1, \ldots, a_{|u|}$ ואת הריצה של $\mathcal{B}$ על v בתור $p = b_0, b_1, \ldots, b_{|v|}$. בהכרח מתקיים כי $a_{|u|} \in F$ ו-$b_{|v|} \in G$.
כעת, נחבר את שתי הריצות r ו-p ונוסיף ריצת ε ביניהן, ונקבל $a_0, a_1, \ldots, a_{|u|}, \varepsilon, b_0, b_1, \ldots, b_{|v|}$.

כיוון ש-$\mathcal{C}$ משמר את המעברים של $\mathcal{A}$ו-$\mathcal{B}$, נקבל כי הריצות r ו-p קיימות ב-$\mathcal{C}$. בנוסף, מכיוון ש-$a_{|u|} \in F$, יש מעבר ε מ-$a_{|u|}$ ל-b_0, שמסתיימת במצב ב-G, נקבל שהיא ריצה מקבלת של $\mathcal{C}$ על $u \cdot v$.

<u>בכיוון השני, $\Rightarrow$ נראה כי $L(\mathcal{C}) \subseteq L_1 \cdot L_2$:</u>
תהי $u \in L(\mathcal{C})$ ותהי $r_0, r_1, \ldots, r_m$ ריצה מקבלת של $\mathcal{C}$ על w.
אנו יודעים כי $r_m \in G$ וגם $q_0 = r_0$, ולכן כיוון שבה"כ $Q \cap S = \emptyset$, בהכרח קיים $0 \leq k \leq m-1$ כך ש-$\{r_0, \ldots r_k\} \subseteq Q$ וגם $\{r_{k+1}, \ldots, r_m\} \subseteq S$ ו-$r_k \to r_{k+1}$ הוא המעבר ε היחיד בריצה r. כלומר, בהכרח $r_k \in F$ וגם $r_{k+1} = s_0$. מכיוון ש-$\mathcal{C}$ משמר מעברים של $\mathcal{A}$ ו-$\mathcal{B}$, נקבל כי $r_0, r_1, \ldots, r_k$ היא ריצה מקבלת של $\mathcal{A}$ על $w_1 \cdot w_2 \cdot \ldots w_k$ ו$r_{k+1}, \ldots, r_m$ היא ריצה מקבלת של $\mathcal{B}$ על $w_{k+1}, \ldots, w_{m-1}$.
סך הכל $w = w_1, w_2, \ldots, w_{m-1}$ היא ריצה מקבלת של $L_1 \cdot L_2$, כנדרש.

טענה
אם L רגולרית, אזי L^* רגורלרית.
הוכחה
בתרגיל נראה את ההוכחה המלאה, כאן נראה רק את הבנייה עצמה.
נבנה $\mathcal{A}' = \langle \Sigma, Q \cup \{q_{\text{ start}}\}, \delta', \{q_{\text{ start}}\}, \{q_{\text{ start}}\} \rangle$ כך ש-q_{start} הוא מצב חדש, כך שלכל $q \in Q \cup \{q_{\text{start}}\}$ יתקיים:

$$\delta'(q, \sigma) = \begin{cases} \{\delta(q, \sigma)\} & \text{if } q \in Q \\ \emptyset & \text{if } q = q_{\text{ start}} \end{cases}$$

וגם נגדיר:

$$\delta'(q, \epsilon) = \begin{cases} \emptyset & \text{if } q \in Q \backslash F \\ \{q_{\text{ start}}\} & \text{if } q \in F \\ \{q_0\} & \text{if } q = q_{\text{ start}} \end{cases}$$

1.4 ביטויים רגולריים

תרגול מס' 3:
יום שני
25.10.21
(בדר)

עד כה דיברנו על אוטומטים, אבל כיצד מחשבים יכולים לחשב זאת? ראינו בהרצאה שיש דרך נוחה יותר לתאר שפות רגולריות, באמצעות ביטויים רגולריים.

תזכורת להגדרות

בהינתן א"ב Σ, ביטוי רגולרי מוגדרת רקורסיבית:

- $a \in \Sigma$, ε ו-$\emptyset$ הם ביטויים רגולריים.
- אם r_1ו-r_2 ביטויים רגולריים כך גם:
  - $r_1 \cup r_2$ ביטוי רגולרי.
  - $r_1 \cdot r_2$ ביטוי רגולרי.
  - r_1^* ביטוי רגולרי.

כל ביטוי רגולרי r, מגדיר את השפה $L(r)$:

- $L(\emptyset) = \emptyset$, $L(\varepsilon) = \{\varepsilon\}$, $L(a) = \{a\}$.
- $L(r_1 + r_2) = L(r_1) \cup L(r_2)$.
- $L(r_1 \cdot r_2) = L(r_1) \cdot L(r_2)$.
- $L(r_1^*) = (L(r_1))^*$.

דוגמא

ניקח את $r = a^* \cdot (a \cup b)$, ואז נקבל כי:

$$L(a^* \cdot (a \cup b)) = L(a^*) \cdot L(a \cup b) = L(a)^* \cdot (L(a) \cup L(b)) =$$
$$\{a\}^* \cdot (\{a\} \cup \{b\}) = \{a^*\} \cdot \Sigma = \left\{a^k \mid k \geq 0\right\} \cdot \Sigma$$

משפט

L היא רגולרית אם ורק אם קיים ביטוי רגולרי r כך ש-$L(r) = L$.

הוכחה

נתחיל מהכיוון הפשוט בו קודם כל נראה כי לכל ביטוי רגולרי יש אוטומט שקול, כך שכל שפה שמוגדרת על ידי הביטוי הרגולרי, היא רגולרית.

למה

לכל ביטוי רגולרי r, קיים NFA שנסמנו $\mathcal{A}_r$, כך ש-$L(r) = L(\mathcal{A}_r)$.

הוכחה

נוכיח זאת באינדוקציה על מבנה הביטוי r:

<u>בסיס האינדוקציה</u>

- אם $r = \emptyset$, אזי $\mathcal{A}_r$ יהיה ה-NFA שמקבל את השפה הריקה.
- אם $r = \varepsilon$, אזי $\mathcal{A}_r$ יהיה ה-NFA שמקבל את $\{\varepsilon\}$.
- אם $r = a \in \Sigma$, אזי $\mathcal{A}_r$ יהיה ה-NFA שמקבל את $\{a\}$.

<u>צעד האינדוקציה</u>

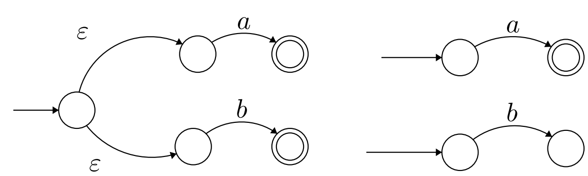
נניח כי הטענה נכונה לכל ביטוי שקטן מ-$|r|$:

- אם $r = s \cup t$, אזי $\mathcal{A}_r$ יהיה ה-NFA שמקבל את $L(\mathcal{A}_s) \cup L(\mathcal{A}_t)$.
  מסגירות שפות רגולריות תחת איחוד, בהכרח קיים NFA שמקיים
  $L(\mathcal{A}_r) = L(\mathcal{A}_s) \cup L(\mathcal{A}_t) = L(s) \cup L(t) = L(s \cup t) = L(r)$.
- אם $r = s \cdot t$, אזי $\mathcal{A}_r$ יהיה ה-NFA שמקבל את $L(\mathcal{A}_s) \cdot L(\mathcal{A}_t)$.
  מסגירות שפות רגולריות תחת שרשור, בהכרח קיים NFA שמקיים
  $L(\mathcal{A}_r) = L(\mathcal{A}_s) \cdot L(\mathcal{A}_t) = L(s) \cdot L(t) = L(s \cdot t) = L(r)$.
- אם $r = s^*$, עלינו לבנות NFA עבור $L(\mathcal{A}_s)^*$. נבנה זאת בתרגיל.

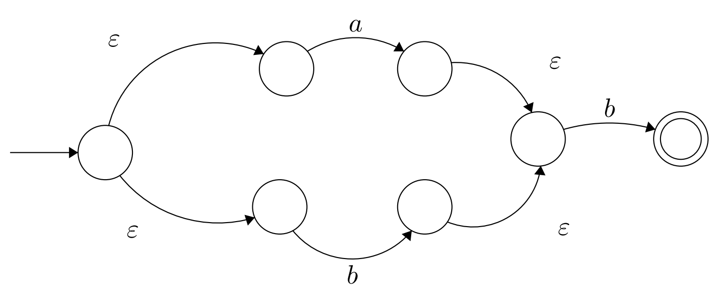
דוגמה לכיוון הראשון

ניקח את הא"ב $\Sigma = \{a, b\}$ ואת הביטוי הרגולרי $r = ((a \cup b) \cdot b)^*$. נראה זאת בשלבים:

שלב I: a, b. שלב II: $a \cup b$.



שלב III: $(a \cup b) \cdot b$



הגדרה

נאמר כי NFA הוא NFA מוכלל, או GNFA, אם על הצלעות מופיעים ביטויים רגולריים במקום אותיות.

נניח כי בה"כ, לכל GNFA מתקיים כי:

1. יש מצב התחלתי יחיד, שאין קשתות שנכנסות אליו.
2. יש מצב מקבל יחיד שאין קשתות שיוצאות ממנו.

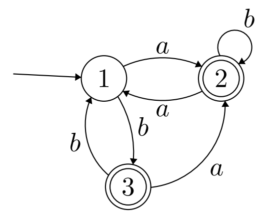
למה

לכל DFA $\mathcal{A}$ קיים ביטוי רגולרי r כך ש-$L(\mathcal{A}) = L(r)$.

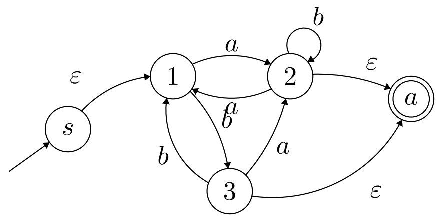
הוכחה

נדגים רק את רעיון ההוכחה[10].
הרעיון הוא להשתמש באלגוריתם שמקבל אלגוריתם סופי לא דטרמיניסטי ומחזיר ביטוי רגולרי.
באלגוריתם, ניקח קשתות באוטומט ונחליף אותם למסלול ארוך יותר, שמוותר על כל המצבים באמצע.
נדגים את הביצוע באמצעות דוגמה:

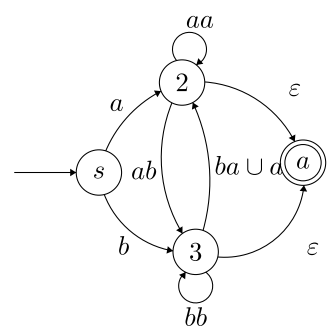
שלב 0: המצב ההתחלתי



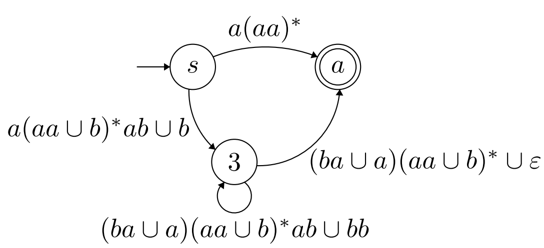
שלב 1: הוספת מצבי התחלה וסיום



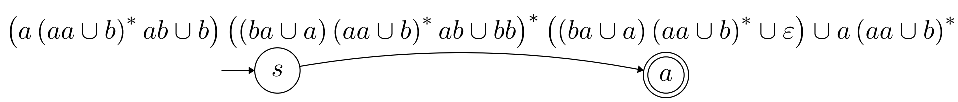
שלב 2: הוצאת קודקוד 1



שלב 3: הוצאת קודקוד 2



שלב 4: הוצאת קודקוד 3



1.5 למת הניפוח

ראינו כבר כמה פעמים שיש שפות לא רגולריות, והסברנו זאת גם בתרגול הראשון, בהתבסס על גודל השפות.
הזכרנו אז שלא מדובר בהוכחה קונסטרוקטובית.
בהרצאה ראינו דוגמה לשפה כזאת $L = \{0^n 1^n \mid n \in \mathbb{N}\}$ והוכחנו באמצעות למת הניפוח שהיא לא שפה רגולרית.

טענה (למת הניפוח)

תהי L שפת רגולרית, אזי ניתן לנפח את L, כלומר קיים $p > 0$ קבוע ניפוח, כך שלכל $w \in L$ שמתקיים עבורה כי $|w| > p$, קיימים $x, y, z \in \Sigma^*$ ו-$w = xyz$ וגם מתקיים:

1. $|y| > 0$.
2. $|xy| \leq p$.

[10] "לא נוכיח כי זה מאוד מאוד משעמם" (ב.א.ר)

3. לכל $i \in \mathbb{N} \cup \{0\}$ יתקיים כי $xy^i z \in L$.

באמצעות למת הניפוח ניתן להוכיח ששפה היא לא רגולרית, אך לא להוכיח כי שפה היא רגולרית.

דוגמה 1

תהי $L_1 = \left\{1^{n^2} \mid n \geq 0\right\}$. L_1 איננה רגולרית, ונשתמש בלמת הניפוח על מנת להוכיח זאת.
נניח בשלילה כי L_1 היא רגולרית, ויהי p קבוע ניפוח עבורה. נתבונן במילה $w = 1^{p^2}$.
מלמת הניפוח, עולה כי ניתן לרשום את w בתור xyz, כך ש-$|xy| \leq p$.
נוכל לרשום את כל אחד מהמספרים הללו בתור $x = 1^j, y = 1^k, z = 1^l$, כך ש-$j+k \leq p$, $k > 0$ ו-$j+k+l = p^2$.
אם ננפח זאת עם $i = 2$, נקבל $xy^2z = 1^j 1^k 1^k 1^l = 1^{p^2+k}$.
מצד שני, מתקיים:

$$\underbrace{p^2 < p^2 + k}_{\text{הוספת מחובר}} \underbrace{\leq}_{\text{נתון}} \underbrace{p^2 + p < p^2 + 2p + 1}_{\text{הוספת חיובי}} = (p+1)^2$$

מכאן עולה כי $p^2 + k$ הוא לא ריבוע של מספר טבעי (כי הוא בין שני ריבועים של מספר טבעי...), ומכאן נובע כי xy^2z לא ב-L_1, בסתירה ללמת הניפוח.

אבחנה

אנחנו יכולים להשתמש בלמת הניפוח באמצעות הוכחה בשלילה על מנת להוכיח ששפה L היא לא רגולרית.
הכיוון ההפוך לא נכון: יש שפות לא רגולריות שמקיימות את למת הניפוח.

דוגמה 2

תהי $\Sigma = \{0, 1\}$ ונוכיח כי השפה $L_2 = \{w \in \Sigma^* \mid \#_0(w) = \#_1(w)\}$ איננה רגולרית.
נניח בשלילה כי L_2 רגולרית ויהי p קבוע ניפוח. נתבונן במילה $w = 0^p 1^p \in L_2$. בבירור $|w| > p$, כך שנוכל לרשום את $w = xyz$, וכל התנאים לעיל מתקיימים.
ננפח את p עם $i > 1$ ונקבל את המילה $xy^i z \in L_2$. כיוון ש-$|xy| \leq p$, נקבל כי xy מכילה רק אפסים, ולכן נוכל לרשום את כל אחד מהאיברים בתור $x = 0^j, y = 0^k, z = 0^l 1^p$, כאשר $j, l \geq 0, k > 0$ וגם $j + k + l = p$.
לכן יתקיים כי $xy^i z = 0^j 0^{ik} 0^l 1^p = 0^{p+(i-1)k} 1^p$. כיוון ש-$i > 0$ ו-$k > 0$, יתקיים בפרט כי מספר האפסים גדול ממספר האחדים, בסתירה.

תרגול מס' 4:

יום שני

01.11.21

(בדר)

אבחנה

אנחנו אומרים שפונקציה g היא $\omega(n)$ אם לכל $c > 0$ יש $n \in \mathbb{N}$ כך ש-$g(n) > cn$.
בפרט, מתקיים כי $\lim\limits_{n \to \infty} \frac{g(n)}{n} = \infty$.

1.5.1 טענות נוספות ללמת הניפוח

טענה

תהי $f:\mathbb{N}\to\mathbb{N}$ מונוטונית עולה, כך ש-$f=\omega\left(n\right)$. אזי $L_f=\left\{a^{f(n)}\mid n\in\mathbb{N}\right\}$ איננה רגולרית.

למה

תהי $f:\mathbb{N}\to\mathbb{N}$ פונקציה מונוטונית עולה כך ש-$f\left(n\right)=\omega\left(n\right)$, אזי לכל $N,k\in\mathbb{N}$ קיים $n>N$ כך ש-$f\left(n+1\right)-f\left(n\right)>k$.

הוכחה

נניח בשלילה שלא. אזי קיימים $N,k\in\mathbb{N}$ כך שלכל $n>N$ מתקיים כי $f\left(n+1\right)-f\left(n\right)\leq k$.
נבחין כי מכך עולה שסדרת ההפרשים **חסומה** (כיוון שיש מספר סופי של הפרשים עד N, ומנקודה זו והלאה חסומים על ידי N).
כלומר, לפי ההגדרה קיים $M\in\mathbb{N}$ כך שלכל $n\in\mathbb{N}$ מתקיים כי $f\left(n+1\right)-f\left(n\right)\leq M$. כלומר:

$$f\left(2\right)\leq f\left(1\right)+M,\ f\left(3\right)\leq f\left(2\right)+M\leq f\left(1\right)+2M,\ldots,f\left(n\right)\leq f\left(1\right)+\left(n-1\right)\cdot M$$

אם נחלק את שני הצדדים ב-$\left(n-1\right)$, נקבל כי $\frac{f(n)}{n-1}\leq M+\frac{f(1)}{n-1}$, בסתירה לכך ש-$\lim\limits_{n\to\infty}\frac{f(n)}{n}=\infty$.

הוכחה לטענה

תהי $f\left(n\right)=\omega\left(n\right)$. נרצה להראות כי L_f איננה מקיימת את התנאים של למת הניפוח.
נניח בשלילה כי למת הניפוח מתקיימת ויהי $p>0$ קבוע הניפוח.
נפעיל את הלמה שהוכחנו זה עתה עם $N=k=p$ ונקבל כי קיים $n>p$ כך ש-$f\left(n+1\right)>f\left(n\right)+p$. מכיוון ש-$f$ היא מונוטונית עולה, מתקיים בהכרח כי $f\left(n\right)>n>p$.
כעת, תהי מילה $a^{f(n)}$, אזי מלמת הניפוח קיימים x,y,z כך ש-$a^{f(n)}=xyz$, וכך ש-$0<\left|y\right|<\left|xy\right|<p$ ולכל $i\in\mathbb{N}\cup\{0\}$ בהכרח $xy^iz\in L_f$.
תהי $m=\left|y\right|$ (אורך המעגל שניפחנו), אזי לכל $i=2$ נקבל כי $a^{f(n)+m}=xy^2z\in L_f$. מצד שני, $f\left(n\right)<f\left(n\right)+m\leq f\left(n\right)+p<f\left(n+1\right)$. כלומר, קיבלנו כי $\left(f\left(n\right)+m\right)$ לא בתמונה של f כי בין $f\left(n\right)$ ל-$f\left(n+1\right)$ קיימת $f\left(n+m\right)$, בסתירה.

1.6 משפט מייהל-נרוד

1.6.1 מחלקות מייהל-נרוד

נזכיר את ההגדרה שראינו בכיתה ליחס מייהל-נרוד:

הגדרה

יהי Σ א"ב ותהא $L\subseteq\Sigma^*$. יחס השקילות מייהל-נרוד של L מוגדר כך: לכל $x,y\in\Sigma^*$ נגדיר כי $x\sim_L y$ אם ורק אם לכל $z\in\Sigma^*$ מתקיים כי $x\cdot z\in L\Leftrightarrow y\cdot z\in L$.

כלומר, לפי ההגדרה עולה כי x ו-y שקולים אם אין **זנב מפריד** ביניהם. לכל $x\in\Sigma^*$ אנחנו מגדירים את $[w]=\{x\mid w\sim_L x\}$ בתור מחלקות השקילות של w.
כמו כן, הוכחנו את המשפט הבא.

משפט

תהא $L \subseteq L^*$ שפה, אזי L רגולרית אם ורק אם יש ב-$\sim_L$ מספר סופי של מחלקות שקילות מייהל-נרוד.

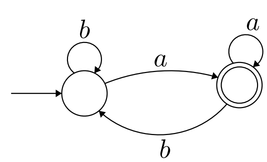
בשונה מלמת הניפוח, משפט זה נותן לנו תיאור שלם של מחלקות רגולרית, אז נוכל להשתמש במשפט זה כדי להוכיח ששפה היא רגולרית או להוכיח שהיא לא רגולרית.

דוגמה

תהי השפה $L = \left\{a^k \mid k \text{ אינה חזקה של } 2\right\}$. נבדוק האם מדובר בשפה רגולרית, באמצעות מחלקות השקילות.
נגדיר סדרה של מילים $\{w_i \mid i \in \mathbb{N}\}$ כך שלכל $m \neq n$ יתקיים כי $w_m \not\sim_L w_n$, כלומר כי קיים זנב מפריד ביניהם.
לכל $n \neq m \in \mathbb{N}$ נתבונן במילים $w_n = a^{2^n}$ ו-$w_m = a^{2^m}$. בה"כ, נניח כי $n < m$. נקבל כי הזנב המפריד הוא $z = a^{2^n}$. כי הרי $2^n + 2^n = 2^{n+1}$, אבל $2^m + 2^n = 2^n\left(2^{m-n}+1\right)$.
כלומר $a^{2^n} \cdot a^{2^n} = a^{2^{n+1}}$, ומאידך $a^{2^m} \cdot a^{2^n} = a^{2^n\left(2^{m-n}+1\right)}$. סך הכל קיבלנו כי $a^{2^m} \cdot 2^{2^n} \in L$, כי $2^n\left(2^{m-n}+1\right)$ לא חזקה של 2, ומאידך $a^{2^n} \cdot a^{2^n} \notin L$.
הראינו שיש אינסוף מחלקות שקילות, כי למעשה כל $n, m \in \mathbb{N}$ מגדירה מחלקת שקילות כיוון שמגדירה זנב מפריד, וכיוון שיש אינסוף מחלקות שקילות, לא מדובר בשפה רגולרית.

דוגמה נוספת (חיובית)

נתבונן בשפה $L = \left\{w \in \{a,b\}^* \mid w \text{ מסתיימת עם } a\right\}$. נראה שיש רק שתי מחלקות שקילות (L ו-L משלים) וממילא מדובר בשפה רגולרית.
ראשית, ברור כי $L \cup I = \Sigma^*$. כעת, נראה ש-L היא מחלקת שקילות. לכל $x, y \in L$ מתקיים כי x ו-y מסתיימות ב-a. תהי $z \in \Sigma^*$ וניקח $xz \in L$. ישנן שתי אפשרויות: z מסתיימת ב-a או ש-z היא המילה הריקה ולכן ניתן לראות בבירור כי גם $yz \in L$ (הכיוון השני דומה). אם כך, כל שתי מילים ב-L שקולות ולכן מדובר במחלקת שקילות.
מצד שני, ניקח $x, y \in I$ ו-$z \in \Sigma^*$ - בהכרח כיוון ש-$x \notin L$, אם $xz \in L$ אז z חייבת להסתיים ב-a וממילא גם $yz \in L$.
לכל $x \in L$ וגם $y \in I$ אזי הזנב $z = \varepsilon$ הוא זנב מפריד בין x ו-y.
האוטומט עבור השפה הוא מינימלי, שכן הוא עם שני מצבים:



1.6.2 שאלות חזרה לכיף

שאלה 1

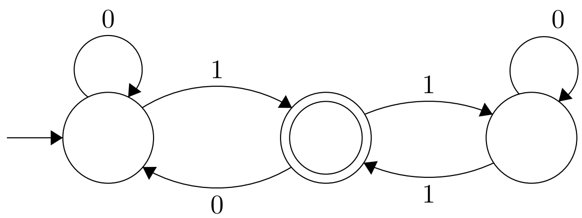
יהי $A = \langle Q, \{0,1\}, q_0, \delta, F\rangle$ עם $|Q| = r$ וגם $0^r1^r \in L(A)$.
אילו מהטענות הבאות נכונה בהכרח?

1. $L(0^*1^*) \subseteq L(\mathcal{A})$.
2. $L(\mathcal{A}) \subsetneq L(0^*1^*)$.

3. 1 לא בהכרח נכון אבל לכל $i \geq 1$ מתקיים כי $0^{ir}1^{ir} \in L(A)$.

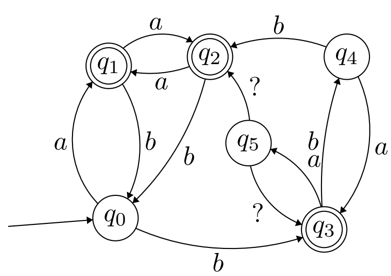
4. 1 לא בהכרח נכון אבל קיים $k \geq 1$ כך שלכל i מתקיים כי $0^{r+ik}1^{r+k} \in L(A)$.

טענה 1 איננה נכונה, כי נוכל לקחת אוטומט של מספר זוגי של אפסים.
טענה 2 אינה נכונה, מאותה סיבה שטענה 1 לא נכונה.
טענה 3 לא בהכרח נכונה, כי אם ניקח $i = 2$ ו-$r = 3$, בדוגמה הבאה, הטענה לא תהיה נכונה:



הטענה הנכונה היא בהכרח 4.

שאלה 2
נתבונן באוטומט הדטרמינסטי הבא.



נתון כי ל-$L(A)$ יש 4 מחלקות שקילות מייהל-נרוד. מהם הערכים החסרים?

1. $\delta(q_5, a) = q_2, \delta(q_5, b) = q_3$.

2. $\delta(q_5, a) = q_3, \delta(q_5, b) = q_2$.

3. תשובות א' וב' אפשריות.

4. אף אחת מהתשובות לא נכונה.

תשובה 3 היא הנכונה. מדוע? באמצעות האלגוריתם שראינו בכיתה, נוכל למצוא את הערכים ששקולים $=_i$ עד שנגיע לנקודת שבת.
אנחנו יודעים שאם נבצע את תהליך המינימיזציה, נשאר עם 4 מחלקות שקילות.
$\sim_A^0$: $\{\{q_1, q_2, q_3\}, \{q_0, q_4, q_5\}\}$.
$\sim_A^1$: נבחין כי q_1 ו-q_2 נשארים במצב מקבל עם a, ועוברים למצב לא מקבל עם b, אבל q_3 עובר למצב לא מקבל עם a (זנב מפריד).
$\{\{q_1, q_2\}, \{q_3\}, \{q_0, q_4, q_5\}\}$.
$\sim_A^2$: נבחין כי בהכרח q_0 ו-q_4 לא שקולים, כי q_4 עם a עובר ל-q_3 ואילו q_0 עם a עובר ל-q_1.
על מנת שנסדר זאת בבירור, נראה כי $\delta(q_4, a) = q_3, \delta(q_4, b) = q_2$ וגם $\delta(q_0, a) = q_1, \delta(q_0, b) = q_3$.
לכן נמצא דרך ש-q_5 תהיה שקולה לאחד מהם. אם $\delta(q_5, a) = q_2, \delta(q_5, b) = q_3$, אזי q_5 שקול ל-q_0 (כי שנהם מגיעים למחלקת q_1 ו-q_3 בהתאמה).

מצד שני, אם $\delta(q_5,a)=q_3, \delta(q_5,b)=q_2$, אזי q_5 שקול ל-q_4, כי שניהם מגיעים בדיוק לאותם מצבים.

תרגול מס' 5:

יום שני

08.11.21

(בדר)

1.7 שפות חסרות הקשר

נזכיר כי בכיתה דיברנו על שפות חסרות הקשר ודקדוקים.

נתבונן בדוגמה על מנת לרענן את ההגדרות.

דוגמה

ניקח את הדקדוק עם החוקים $S \to 0S1 \mid B \mid \varepsilon$ ו-$B \to \#$ ונוכל לקבל:

$$S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 00B11 \Rightarrow 00\#11$$

נרענן את ההגדרות:

תזכורת להגדרות

דקדוק חסר הקשר (להלן ח"ה) G מוגדר על ידי $G=\langle V,\Sigma,R,S\rangle$ כאשר:

- V משתנים.
- Σ א"ב.
- R הן חוקי גזירה מהצורה $V \to (V\cup\Sigma)^*$.
- $S\in V$ משתנה התחלתי.

אם $w,u,v\in(V\cup\Sigma)^*$ ו-$A\to w$ חוק בדקדוק, אזי נאמר ש-$uAV \Rightarrow uwV$ **מייצר** את w.

אם $u,v\in(V\cup\Sigma)^*$ נאמר כי $u \overset{*}{\Rightarrow} v$ אם יש סדרה $u=u_1\Rightarrow u_2\Rightarrow u_3\Rightarrow\ldots\Rightarrow u_k=V$ ו-$1\le k$.

השפה של G היא: $L(G)=\left\{w\in\Sigma^* \mid S\overset{*}{\Rightarrow} w\right\}$.

כעת, לאחר התזכורת, נציג מספר דוגמאות.

דוגמאות

נציג דקדוקים חסר הקשר עבור השפות הבאות:

1. עבור $\left\{a^nb^{2n} : n\ge 0\right\}$ יתקיים כי $S\to aSbb \mid \varepsilon$[11]
2. עבור $\left\{a^ib^j : j\ge i\right\}$ יתקיים כי $\begin{matrix} S\to aSbT \mid bT \mid \varepsilon \\ T\to Tb \mid \varepsilon \end{matrix}$.
3. עבור $\left\{a^ib^jc^jd^i : i,j\ge 0\right\}$, יתקיים כי $\begin{matrix} S\to aSd \mid T \mid \varepsilon \\ T\to bTc \mid \varepsilon \end{matrix}$

[11] ניתן להוכיח זאת פורמלית באמצעות אינדוקציה על עומק העץ.

אבחנה

מדוע אנו קוראים לדבר זה דקדוק חסר הקשר? הסיבה לכך היא שאנחנו מתבוננים בכל אות בפני עצמו, בלי להסתכל על האותיות האחרות ולכן למעשה אנחנו חסרי הקשר או קונטקסט. אכן, יש שפות שהן תלויות הקשר, למשל $A0B1 \to B11$ - שפות אלו בעלות מגוון עשיר יותר של שפות.

1.7.1 תכונות סגור

אחרי שראינו את מחלקת CFL, הדבר הטבעי הוא לחקור את תכונות הסגור שלה.

טענה

אם L_1, L_2 דקדוקים חסרי הקשר אזי $L_1 \cup L_2$ חסר הקשר.

הוכחה

יהיו $G_1 = \langle V_1, \Sigma, R_1, S_1\rangle$ ו-$G_2 = \langle V_2, \Sigma, R_2, S_2\rangle$ דקדוקים חסרי הקשר, כך ש-$L_1 = L\left(G_1\right)$ ו-$L_2 = L\left(G_2\right)$.
נניח בה"כ כי $V_1 \cap V_2 = \emptyset$ (אחרת, כפי שראינו בעבר, נוכל לשנות את שמות המשתנים ב-V_2).
נייצר דקדוק חסר הקשר G עבור $L_1 \cup L_2$ כלדלהן: יהי S משתנה חדש, כך ש-$S \notin V_1 \cup V_2$ ו-G מוגדר על ידי:

$$G = \langle V_1 \cup V_2, \Sigma, R_1 \cup R_2 \cup \{S \to S_1 \mid S_2\}, S\rangle$$

מכאן עולה, לפי ההגדרה, כי קיבלנו את שני הדקדוקים, שכן $S \to S_1 \mid S_2$.

טענה

אם L_1, L_2 דקדוקים חסרי הקשר, אזי $L_1 \cdot L_2$ חסר הקשר.

הוכחה

יהיו $G_1 = \langle V_1, \Sigma, R_1, S_1\rangle$ ו-$G_2 = \langle V_2, \Sigma, R_2, S_2\rangle$ דקדוקים חסרי הקשר, כך ש-$L_1 = L\left(G_1\right)$ ו-$L_2 = L\left(G_2\right)$.
נניח בה"כ כי $V_1 \cap V_2 = \emptyset$ (אחרת, כפי שראינו בעבר, נוכל לשנות את שמות המשתנים ב-V_2).
נייצר דקדוק חסר הקשר G עבור $L_1 \cdot L_2$ כלדלהן. יהי S משתנה חדש, כך ש-$S \notin V_1 \cup V_2$ ו-G מוגדר על ידי:

$$G = \langle V_1 \cup V_2, \Sigma, R_1 \cup R_2 \cup \{S \to S_1 \cdot S_2\}, S\rangle .$$

מכאן עולה, לפי ההגדרה, כי קיבלנו את שרשור הדקדוקים, שכן $S \to S_1 \cdot S_2$.

מה קורה לגבי משלים, חיתוך וכוכב? נראה בהמשך.

1.7.2 הצורה הנורמלית של חומסקי

לפעמים נוח לעבוד עם דקדוק חסר הקשר מצורה מסוימת. בחלק זה נראה 'צורה נורמלית' מסוג מסוים, ונראה שלכל דקדוק חסר הקשר יש הצגה שקולה בצורה הנורמלית. מעבר לכך, נראה גם דרך לעבור בין שתי התצוגות.

הגדרה

דקדוק חסר הקשר G ניתן להצגה **בצורה הנורמלית של חומסקי**, אם כל כללי הגזירה שלו הם מהצורה הבאה:

1. $S \to \varepsilon$ כאשר S הוא משתנה גזירה התחלתי.
2. $A \to BC$ כאשר $B, C \in V \setminus S$.
3. $A \to \alpha$ כאשר α הוא טרמינל.

טענה

לכל דקדוק חסר הקשר G יש תצוגה שקולה G' מהצורה הנורמלית של חומסקי.

הוכחה

נוכיח זאת בשלבים.

1. הגדרת משתנה התחלתי חדש - נתחיל באמצעות הוספת משתנה התחלתי חדש S_0 והוספת כלל הגזירה $S_0 \to S$. דבר זה מבטיח לנו כי המשתנה ההתחלתי לא נגזר והשפה לא משתנה.
2. הסרת כל כללי ה-ε. נוריד את כל הכללים מהצורה $A \to \varepsilon$ כאשר $A \neq S_0$ ו-$A \in V$ ונעבור על כל הכללים שמכילים את A. עבור כל כלל כזה, נוסיף אוסף של כללים שיהוו את הקומבניציות שבהן A מוחלפת על ידי ε.
3. הורדת כללים מהצורה $A \to B$. לכל כלל $A \to B$ עלינו להחליף את B עם הגזירות שאפשריות לפי B. נמחק את כל $A \to B$ ונוסיף כל דבר ש-B יכול לגזור ל-A.
4. הורדת כללים ארוכים. לכל כלל מהצורה $A \to V_1 V_2 \cdots V_k$ כך ש-$k \geq 3$ נייצר משתנים חדשים $U_2, U_3, \ldots, U_{k-1}$. נייצר את הכללים הבאים:

$$A \to V_1 U_2$$
$$U_2 \to V_2 U_3$$
$$\vdots$$
$$U_{k-1} \to V_{k-1} V_k$$

5. הורדת אותיות. לכל אות $\sigma \in \Sigma$ נוסיף את x_σ ואת הכלל $x_\sigma \to \sigma$ ונחליף את המופעים של σ ב-x_σ.

דוגמא

נתבונן בדקדוק הבא
$$S \to ASA \mid aB$$
$$A \to B \mid S$$
$$B \to b \mid \epsilon$$
ונעבור לפי הכללים:

1. נוסיף $S_0 \to S$.
2. נעבור על כל כללי ה-ε:.
   - כיוון ש-$B \to \varepsilon$, נוריד אותו, ונוסיף $A \to \varepsilon$.
   - נוריד את $A \to \varepsilon$, ונחליף את הכלל הראשון בכלל הבא $S \to ASA \mid AS \mid SA \mid aB|a$.

3. נוריד את הכללים הקצרים $S_0 \to S, A \to B, A \to S$ ונקבל את ארבעת הכללים הבאים:

$$S_0 \to ASA \,|\, AS \,|\, SA \,|\, aB \,|a$$
$$S \to ASA \,|\, AS \,|\, SA \,|\, aB|a$$
$$A \to b \,|\, ASA|AS|SA|aB \mid a$$
$$B \to b$$

4. נוריד את הכללים הארוכים באמצעות יצירת כלל בודד שהינו $V \to SA$, ואז נקבל:

$$S_0 \to AV|AS|SA|aB|a$$
$$S \to AV|AS|SA|aB|a$$
$$A \to b|AV|AS|SA|aB \mid a$$
$$V \to SA$$
$$B \to b$$

5. לבסוף, נוריד את האותיות. נגדיר את $X_a \to a$ ואת $X_b \to b$:

$$S_0 \to AV|AS|SA\,|X_aB|\,a$$
$$S \to AV|AS|SA\,|X_aB|\,a$$
$$A \to b|AV|AS|SA|X_aB \mid a$$
$$V \to SA$$
$$B \to b$$
$$X_a \to a, \quad X_b \to b$$

2 תורת החישוביות

2.1 מכונות טיורינג

הגדרה

שתי מכונות M_1 ו-M_2 הן שקולות אם הן מזהות את אותה השפה, כלומר $L(M_1) = L(M_2)$, מקבלות את אותן מילים, דוחות את אותן מילים, וגם עוצרות על אותן מילים.

הגדרה

שני מודלים $\mathcal{X}$ ו-$\mathcal{Y}$ הם שקולים אם לכל מכונה מסוג $\mathcal{X}$ קיימת מכונה שקולה מסוג $\mathcal{Y}$ ולכל מכונה מסוג $\mathcal{Y}$ קיימת מכונה שקולה מסוג $\mathcal{X}$.

2.1.1 מכונות טיורינג עם שני סרטים

מכונת טיורינג עם שני סרטים היא מכונת טיורינג רגילה, רק עם שני סרטים, כשלכל ראש יש את ראש הקריאה שלו. באתחול, הקלט מופיע בסרט הראשון, ואילו הסרט השני ריק.
פונקציית המעברים השתנתה, כך שהיא מאפשרת לנו לקרוא, לכתוב ולהזיז את הראשים בשני הסרטים בו זמנית. פורמלית, מתקיים כי $\delta : Q \times \Gamma^2 \to Q \times \Gamma^2 \times \{R, L\}^2$.
כלומר, אם ניקח למשל את הביטוי $\delta\left(q, \gamma_1, \gamma_2\right) = \left(q', \gamma_1', \gamma_2', L, R\right)$, הכוונה היא שאם המכונה במצב q, ראש 1 ב-γ_1 וראש 2 ב-γ_2 אזי המכונה עוברת ל-q', כותבת γ_1' בסרט 1, מזיזה ראש 1 שמאלה, כותבת γ_2' בסרט 2 ומזיזה ראש 2 ימינה.
נבחין כי היתרון המשמעותי הוא הוספת הראשים ולא הסרטים (שזו משימה די פשוטה).

טענה

לכל מכונת טיורינג, M יש מ"ט עם שני סרטים M' השקולה לה.

טענה

לכל מ"ט M עם שני סרטים $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}} \rangle$ יש מ"ט M' (עם סרט אחד) ששקולה לה.

הוכחה

נגדיר $M' = \left\langle Q', \Sigma, \Gamma', \delta', q_0', q_{acc}', q_{rej}' \right\rangle$ כך ש- $\Gamma' = (\Gamma \times \Gamma \times \{0,1\} \times \{0,1\}) \cup \Sigma \cup \sqcup$. נרצה שיתקיים כי האות $(a, b, 0, 1)$ במקום ה-i בסרט של M' (יש לה סרט יחיד) תתפרש כ"בסרט הראשון של M במקום ה-i, כתוב כעת a והראש הקורא לא נמצא שם, ובמקום ה-i בסרט השני של M כתוב כעת b והראש הקורא נמצא שם". איך נעשה זאת? באמצעות פעולתה של M.
בהינתן קלט $w = \sigma_1\sigma_2 \ldots \sigma_n$, המכונה M תחליף כל σ_i:

1. בשלב האתחול: כאשר $i > 1$, נחליף ב-$(w_i, \sqcup, 0, 0)$ ועבור $i = 1$ נחליף ב-$(\sigma_1, \sqcup, 1, 1)$. כלומר, בשלב זה שני הראשים הקוראים של M **במקום הראשון ולא בשום מקום אחר**. בסרט הראשון במקום ה-i כתוב σ_i ובסרט השני במקום ה-i כתוב σ_i ובסרט השני לא כתוב כלום.

2. בשלב הסימולציה:

(א) M' תסרוק את הסרט שלה ותחפש את "מיקום הראש הקורא של M'". כלומר, אות מהצורה $(\gamma, *, 1, *)$ ותקרא את האות שכתובה כעת בסרט 1, שם. לאחר מכן, M' תעבור למצב שלמעשה מקודד את העובדה שמצאה את המיקום וקראה שם γ כלשהי.

(ב) M' תסרוק שוב את הסרט ותחפש אחר $(*, \beta, *, 1)$.

(ג) M' תסרוק את פונקציית המעברים δ של M ותחליט מה לכתוב במקום מה שהיה עכשיו.

(ד) M' תסרוק את הסרט שלה, תמצא את הראש הראשון, תעדכן את γ להיות γ' לפי δ, ותעביר את הראש הראשון ל-L או R לפי δ.

(ה) M' תסרוק את הסרט שלה, תמצא את הראש השני, תעדכן את β להיות β' לפי δ, ותעביר את הראש השני ל-L או R לפי δ.

(ו) M' תבדוק האם δ עוברת ל-q_{acc} או ל-q_{rej} לפי δ, ואם לא - תחזור לתחילת השלב.

מבחינת הדגמה, כך זה נראה לאחר השלב הראשון:

M

σ_1	σ_2	$\dots$	σ_n

M'

$(\sigma_1, \sqcup, 1, 1)$	$(\sigma_2, \sqcup, 0, 0)$	$\dots$	$(\sigma_n, \sqcup, 0, 0)$

לאחר סימולציה, אם נניח המכונה M בסרט הראשון כתבה γ ועברה ימינה, ובסרט השני עברה ימינה וכתבה שם β, אז יתקיים:

M

γ	σ_2	$\dots$	σ_n
β			

M'

$(\gamma, \beta, 0, 1)$	$(\sigma_2, \sqcup, 1, 0)$	$\dots$	$(\sigma_n, \sqcup, 0, 1)$

הסיבה שבגללה העברנו את הראש הקורא לא בהתאמה, כלומר התא השני עכשיו הוא $1, 0$ ולא $1, 1$, היא כיוון שיש קידוד בין הראשים, וקפיצה ימינה לא בהכרח 'קופצת' ימינה בסרט החדש, אלא במיקום כלשהו אחר.
M' מקבלת, לא עוצרת או דוחה, את w אם M מקבלת, לא עוצרת או דוחה את w, ולכן המכונות שקולות.

עלות התרגום

אם M רצה t צעדים על w, אזי M' עשתה בכל צעד לכל היותר $8t$ צעדים, ולכן זמן הריצה הוא $O\left(t^2\right)$.
אם היו לנו k סרטים, זמן הריצה היה $O\left(t^k\right)$, שכן במקרה של שני סרטים, הא"ב שלנו התרחב ל-$|\Gamma|^2 \cdot 2^2$, ואם היו k סרטים, היינו צריכים להרחיב זאת ל-$|\Gamma|^2 \cdot 2^k$ - כלומר מדובר בסדר גידול אקספוננציאלי ביחס למספר הסרטים.

תרגול מס' 6:

יום חמישי

18.11.21

(מאיה)

תזכורת

ניזכר כי המחלקה RE היא מחלקת השפות שיש מכונת טיורינג שמזהה אותם, והמחלקה R זו המחלקה כך שקיימת מכונת טיורינג שמכריעה לגביהן. כמו כן, ראינו את coRE - כלומר, כל השפות כך שהמשלים שלהם ב-RE.

מעבר לזה, ראינו כי R=coRE∩RE.

מחלקות אלו חשובות כיוון שראינו שיש שפות שלא ניתן להכריע לגביהן. דבר זה למעשה מעיד על העובדה שיש פעולות שמחשבים לא יודעים לעשות, כי הרי מחשבים שקולים למכונות טיורינג (למשל, לא ניתן לבדוק האם בקוד מסוים יש לולאה אינסופית).

2.1.2 תכונות סגור של RE

טענה

אם $L_1, L_2 \in \mathrm{RE}$ אזי גם $L_1 \cup L_2 \in \mathrm{RE}$ (כלומר, קיימת גם מכונת טיורינג שמזהה את האיחוד של השפות).

הוכחה

נשים לב כי אי אפשר, בהינתן w קודם, להריץ את M_1 ואז את M_2, כי M_1 עלולה לא לעצור על w, אפילו אם M_2 מקבלת.
לכן, נצטרך למצוא רעיון אחר. נבנה מקום טיורינג M שפועלת כך:

1. שומרת את הקלט w במקום בטוח בתחילת הסרט. אחרי זה שומרת מונה צעדים, שמאותחל לאפס.
2. בכל שלב M תסמלץ את ריצת M על w, במשך כמות הצעדים שכתובה במונה הצעדים.
   (א) אם M_1 הגיעה למצב מקבל או דוחה, M תקבל או תדחה.
   (ב) אם לא, תמחק את תוכן הסרט הלא שמור, תסמלץ את ריצת M_2 באותה כמות צעדים, תקבל או תדחה ותגדיל את המונה ב-1 ותמשיך כך.

דוגמה נוספת להרצה במקביל

טענה

אם $L_1, L_2 \in \mathrm{RE}$ אזי גם $L_1 \cdot L_2 \in \mathrm{RE}$.

הוכחה

נשים לב כי אפשר לבנות מכונת טיורינג M_1 שמזהה את $L' = \{w_1\#w_2 \mid w_1 \in L_1, w_2 \in L_2\}$.
M' תסמלץ את M_1 על w_1 ואם M_1 מקבלת, היא תסמלץ את M_2 על w_2 ואם תקבל אז M' תקבל.
אמנם, יש בעיה באמירה זו, כי איננו יודעים מתי w_1 נגמרת ומתי w_2 מתחילה.
לכן, "נריץ במקביל" את M על כל החלוקות האפשריות של w כך.
המכונה M תחזיק:

- תיאור של M' ושל w בסרט ראשון.
- בסרט השני תחזיק "מונה חלוקות" ומונה צעדים ששניהם יאותחלו לאפס.
- בסרט השלישי יהיה סרט סימולציה.

M בכל שלב תריץ את M' על החלוקה ה-i j צעדים, ותגדיל את i אם M' לא קיבלה. אם $|w| = i$, אז את M'תחזיר את i ל-0 ותגדיל את j ב-1 ותמשיך כך.
(M מאתחלת את סרט שלוש בכל איטרציה).
אם באיזה שלב M' מקבלת, M מקבלת.

טענה

M מזהה את $L_1 \cdot L_2$.

הוכחה

אם $w \in L_1 \cdot L_2$, אזי קיימת חלוקה $w = w_1 \cdot w_2$ כך ש-$w_1 \in L_1$ ו-$w_2 \in L_2$ ואז M' מקבלת את $w_1\#w_2$ אחרי n צעדים, ולכן גם M תקבל את $w_1 w_2$ אחרי n צעדים בריצה על החלוקה הזאת.
אם $w \notin L_1 \cdot L_2$ אז אין חלוקה כפי שעשינו קודם, ולכן M תרוץ לנצח על w.

2.1.3 קידוד

נתבונן בקידוד של מ"ט טיורינג על $\Sigma = \{0, 1, \#\}$ ו-$M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}} \rangle$.
ראשית, נקודד את המצבים על ידי מספרים בינארים בסדר עולה מופרדים על ידי #:

$$0\#1\#00\ldots\#\#$$

לאחר מכן ### להפריד בין הביטויים.

כעת, נקודד את Σ, Γ, כאשר $\Sigma \subseteq \Gamma$. ניתן לקודד את Γ על ידי סטרינגים בינאריים באורך לכל היותר Γ, מופרדים על ידי #. ולאחר מכן ###.

דוגמה

$\Sigma = \{a, b\}$ ו-$\Gamma = \{0, 1, \sqcup\} \cup \Sigma$. מכאן עולה כי $|\Gamma| = 5$ ונקודד על ידי סטרינגים בינאריים באורך $\log(5) = 3$. ונקבל:

$$\underbrace{000}_{a}\#\underbrace{001}_{b}\#\underbrace{010}_{0}\#\underbrace{011}_{1}\#\underbrace{100}_{\sqcup}\#\#\#$$

כעת נקודד את δ.

את המעבר $\delta(q, \sigma) = (q', \sigma', L)$ נקודד כך:

$$\underbrace{\langle q \rangle}_{\text{הקידוד של } q} \# \underbrace{\langle \sigma \rangle}_{\text{הקידוד של } \sigma} \# \langle q' \rangle \# \langle \sigma' \rangle \# \langle L \rangle \#\#\#$$

כאשר $\langle L \rangle = 0$ ו-$\langle R \rangle = 1$. בין המעברים השונים של δ יש ## ולבסוף יש ###.

בסופו של דבר, נקודד את $q_{\text{rej}}, q_{\text{acc}}, q_0$ בצורה דומה:

$$\langle q_0 \rangle \#\#\# \langle q_{\text{acc}} \rangle \#\#\# \langle q_{\text{rej}} \rangle$$

2.1.4 מכונה אוניברסלית

נרצה כעת לבנות מ"ט U שמקבלת בקלט קידוד של מ"ט אחרת $\langle M \rangle$ ומילת קלט $\langle w \rangle$ ועונה כמוה.

כלומר U מקבלת או דוחה או עוצרת אם M מקבלת או דוחה או עוצרת על w.

ל-U יהיו שלושה סרטים:

1. התיאור של M.
2. סרט העבודה של M.
3. המצב הנוכחי של M + חישובים.

הפעולה של U תהיה כזאת:

1. תסרוק את סרט 1 ותמצא את תחילת w.
2. תעתיק את w לסרט 2 ותאתחל את ראשים $1, 2$.
3. סורקת את סרט 1, מוצאת את q_0 ותעתיק אותו לסרט 3.
4. בכל איטרציה:
   (א) נשווה את המצב בסרט 3 ל-q_{rej} או q_{acc} ונדחה או נקבל בהתאם.
   (ב) נסרוק את סרט 1 למצוא את תחילת δ.

(ג) נשווה את המצב בסרט 3 והאות מתחת לראש בסרט 2, לכל המעברים של δ לפי הסדר, עד שנמצא את המעבר הנכון.

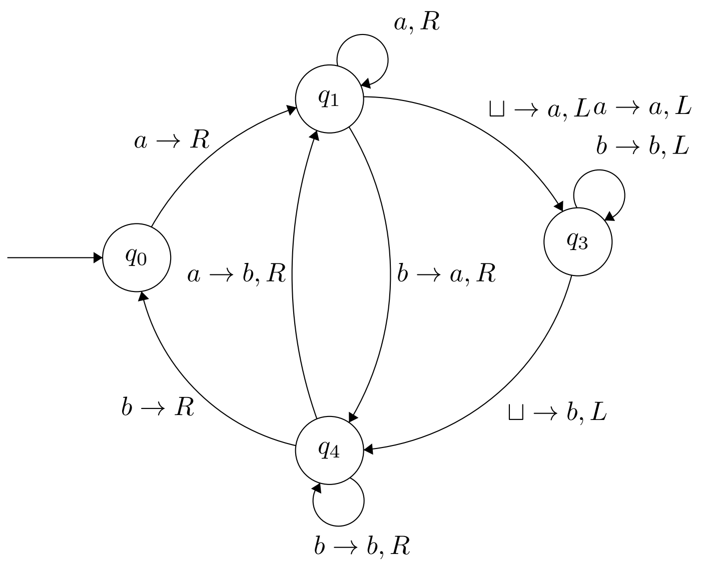
(ד) נחליף את האות בסרט 2 באות הנכונה לפי δ.

(ה) נזיז את הראש ימינה או שמאלה.

(ו) נעדכן את המצב בסרט 3 לפי δ.

2.1.5 כיף עם מכונות טיורינג

נרצה לבנות מכונת טיורינג שמזיזה את הקלט ימינה וכותבת \$ בהתחלה (אפשר להשתמש בה כתת תוכנה בתוכנה הגדולה).
נניח שאנחנו מעל $\{a, b\}$.
נבנה את המכונה הבאה:



מה קורה במכונה זו? אנחנו שמים \$ בהתחלה ומזיזים את הכל ימינה.

אפשר להשתמש במכונה זו כתת מכונה למכונות אחרות.
דוגמה
מ"ט שמחשבת את $f(n) = n + 1$ עבור n בינארי.
איך המכונה הזאת עובדת? המכונה תתחיל ב-q_0 ותסרוק את הסרט עד שתגיע ל-$\sqcup$ ואז תעבור ל-q_1, ותלך שמאלה.
ב-q_1, אם רואים 1 מחליפים ב-0 וממשיכים שמאלה. אם רואים 0, מחליפים ב-1, עוברים שמאלה ועוצרים.
אם מגיעים לתחילת הסרט (רואים \$) , מזיזים הכל ימינה ומוסיפים 1.

2.2 רדוקציות מיפוי

תרגול מס' 7:

יום שני

22.11.21

(בדר)

ניזכר בהגדרות שראינו בהרצאה:

הגדרה
עבור א"ב Σ, נאמר ש-$f:\Sigma^*\to\Sigma^*$ **ניתנת לחישוב** (computable) אם קיימת מ"ט M_f שבהינתן קלט $x\in\Sigma^*$, עוצרת עם $f(x)$ על הסרט.

הגדרה
עבור א"ב Σ, ושתי שפות $A,B\subseteq\Sigma^*$, נאמר ש-A **ניתנת לרדוקציית מיפוי** ל-B $A\le_m B$ ("A קלה יותר מ-B") אם קיימת פונקציה ניתנת לחישוב $f:\Sigma^*\to\Sigma^*$ כך שלכל $x\in\Sigma^*$ יתקיים כי $x\in A\Leftrightarrow f(x)\in B$.

ראינו גם את משפט הרדוקציה:
אם $A\le_m B$ ו-$B\in\mathrm{R}$, אזי $A\in\mathrm{R}$.

כעת, נוכיח את הטענה הבאה:
טענה
יהי $L_1,L_2\in\Sigma^*$. אם $L_1\le_m L_2$ אזי:

1. אם $L_1\notin\mathrm{RE}$ אזי $L_2\notin\mathrm{RE}$.
2. אם $L_1\notin\mathrm{coRE}$ אזי $L_2\notin\mathrm{coRE}$.
3. אם $L_1\notin\mathrm{R}$ אזי $L_2\notin\mathrm{R}$.

הוכחה
נוכיח את הטענות לפי הסדר, מלבד האחרונה שאותה הוכחנו בכיתה.

1. תהי $f:\Sigma^*\to\Sigma^*$ רדוקצייה מ-L_1 ל-L_2 ונניח כי יש מ"ט M שמזהה את L_2. נראה כעת כי $L_1\in\mathrm{RE}$. נגדיר מ"ט N (שתזהה את L_1) שפועלת באופן הבא: בהינתן קלט x של N:
   (א) N מחשבת את $y=f(x)$.
   (ב) N מסמלצת את ריצת M על y ועונה כמוה. (אם M תקבל אזי N תקבל, אם M דוחה אז N דוחה, אם M לא עוצרת, אזי N לא עוצרת).
2. נניח כי $L_1\le_m L_2$ ואנו יודעים כי $\overline{L_1}\le_m\overline{L_2}$. אם נניח כי $L_2\in\mathrm{coRE}$ אזי $L_2\in\mathrm{RE}$. ולכן אם $\overline{L_1}\in\mathrm{RE}$ אזי $L_1\in\mathrm{coRE}$.

ניתוח שיוך $\mathrm{ALL_{TM}}$
ניקח את השפה $\mathrm{ALL_{TM}}=\{\langle M\rangle\mid L(M)=\Sigma^*\}$ - כל המכונות ששפתן היא Σ^*. מהו השיוך שלה?

תחילה, נעשה רדוקציה $A_{\mathrm{TM}}\le\mathrm{ALL_{TM}}$ ונראה כי $\mathrm{ALL_{TM}}\notin\mathrm{coRE}$.

<u>בנייה</u>
עלינו להוכיח כי יש רדוקציה כזו - כלומר עלינו להראות כי M מקבלת את w אם ורק אם $L(K)=\Sigma^*$.
בהינתן קלט $\langle M,w\rangle$ של הרדוקציה f, f פולטת $\langle K\rangle$, כאשר K פועלת באופן הבא.
בהינתן קלט x של K, K מתעלמת מ-x, מסמלצת את ריצת M על w ועונה כמוה (אם קיבלה את w, K תקבל את x וכן הלאה).

<u>נכונות</u>

- אם $\langle M,w\rangle \in A_{\text{TM}}$ אזי M מקבלת את w. במקרה זה, K תקבל כל קלט x ולכן $L\left(K\right)=\Sigma^*$ ולכן $\langle K\rangle \in \text{ALL}_{\text{TM}}$.
- אם $\langle M,w\rangle \notin A_{\text{TM}}$ אזי M אינה מקבלת את w. כלומר, M דוחה את w או אינה עוצרת על w. בשני המקרים, K אינה מקבלת כל קלט x ולכן $L\left(K\right)=\emptyset$, כלומר $\langle K\rangle \notin \text{ALL}_{\text{TM}}$.

<u>חישוב</u>

ברור כי הרדוקציה חשיבה, כי אפשר לממש את K כמכונה אוניברסלית, המקודדת בתוכה את התיאור של M ו-w.

כעת, נראה כי $\overline{A_{\text{TM}}} \leq_m \text{ALL}_{\text{TM}}$ ולכן נקבל כי $\text{ALL}_{\text{TM}} \notin \text{RE}$.

נרצה להראות כי M אינה מקבלת את w, אם ורק אם $L\left(K\right)=\Sigma^*$.

<u>בנייה</u>

בהינתן קלט $\langle M,w\rangle$ של הרדוקציה, הרדוקציה פולטת $\langle K\rangle$, כאשר K פועלת באופן הבא.

בהינתן קלט x של K, אזי K מסמלצת את ריצת M על w למשך $|x|$ צעדים:

1. אם M קיבלה את w במשך ה-$|x|$ צעדים, אזי K תדחה.
2. אם M קיבלה את w במשך ה-$|x|$צעדים, K תקבל.

<u>נכונות</u>

- אם $\langle M,w\rangle \in \overline{A_{\text{TM}}}$ אזי M אינה מקבלת את w, ולכל x, M לא מקבלת את w תוך $|x|$ צעדים, ולכן K מקבלת כל x, כלומר $\langle K\rangle \in \text{ALL}_{\text{TM}}$.
- אם $\langle M,w\rangle \notin \overline{\text{ALL}_{\text{TM}}}$, אזי M מקבלת את w, ויהי t מספר הצעדים שבו M מקבלת את w. נשים לב כי לכל x עם $|x| \geq t$ מתקיים כי M מקבלת את w תוך $|x|$ צעדים, ולכן לכל x עם $|x| \geq t$ יתקיים כי K דוחה את x. כלומר, $L\left(K\right) \neq \Sigma^*$ ולכן $\langle K\rangle \notin \text{ALL}_{\text{TM}}$.

ניתוח אפיון $\text{REPEAT}_{\text{TM}}$

ניקח את $L=\text{REPEAT}_{\text{TM}}=\{\langle M,w\rangle \mid$ M אינה עוצרת על w וריצת M וחוזתר לפחות פעמיים על אותה קונפ' $\}$

נרצה להוכיח כי $\text{REPEAT}_{\text{TM}} \in \text{RE}$.

מכונה K המזהה את $\text{REPEAT}_{\text{TM}}$ עובדת באופן הבא.

<u>בנייה</u>

בהניתן קלט x של K:

1. K בודקת אם $x=\langle M,w\rangle$. אם כן, K תמשיך ל-2. אחרת, K דוחה.
2. עבור $1 \leq i$, K מחשבת את הקונפ' ה-i-ית בריצת M על w, C_i:
   (א) אם C_i מקבלת או דוחה, K דוחה.
   (ב) אם C_i הופיעה באיטרציה קודמת, K מקבלת.

<u>נכונות</u>

אם K מקבלת קלט x, אזי יש $x=\langle M,w\rangle$, ויש איזושהי איטרציה $i=j$ ששם גילינו כי M חזרה על C_j לפחות פעמיים, כלומר בפרט $\langle M,w\rangle \in L$.

אם $\langle M,w\rangle \notin \text{REPEAT}_{\text{TM}}$ אזי M מקבלת או דוחה את w, או שאין קונפיגורציה חוזרת. כלומר, K תיתקע ולכן $\langle M,w\rangle \notin L(K)$.

כעת, נראה כי $\text{REPEAT}_{\text{TM}} \notin \text{R}$, ונראה זאת אם נראה כי $\text{REPEAT}_{\text{TM}} \notin \text{coRE}$, באמצעות רדוקציה לשפה שלא ב-coRE.
נוכל לעשות זאת באמצעות רדוקציה ל-HALT_{TM}, כלומר $\text{HALT} \leq_m \text{REPEAT}_{\text{TM}}$.
ניקח $\langle M,w\rangle \to \langle N,w'\rangle$ ונרצה כי M עוצרת על w אם ורק אם N' עוצרת על קונפ' בריצתה על w.

<u>בנייה</u>
בהינתן קלט $\langle M,w\rangle$ של הרדוקציה, הרדוקציה פולטת $\langle N,x\rangle$ כאשר N פועלת באופן הבא:
בהינתן קלט w' של N, N מתעלמת מ-w', מסמלצת את M על w, תוך ניהול מונה הסופר את מספר צעדי הסמלוץ.
אם N עוצרת על w, N עוברת למצב $q_{\text{סתם}}$ ו-N נשארת תמיד ב-$q_{\text{סתם}}$ ומזיזה את הראש שלה שמאלה.

<u>נכונות</u>
אם $\langle M,w\rangle \in \text{HALT}_{\text{TM}}$ אזי M עוצרת על w, ולכן כש-N רצה על w, היא מתישהו תגיע למצב $q_{\text{סתם}}$ ולכן בסופו של דבר N תגיע לקצה השמאלי שלה ותישאר תקועה באותה קונפ', כלומר $\langle N,w\rangle \in \text{REPEAT}_{\text{TM}}$.
אם $\langle M,w\rangle \notin \text{HALT}_{\text{TM}}$, כלומר M אינה עוצרת על w, נתבונן בריצת N על w', ומכיוון ש-M לא עוצרת על w, אזי גם N לא עוצרת. אך מכיוון ש-N מעלה את מונה הצעדים ב-1 אחרי סמלוץ של צעד של M, אזי N לא חוזרת על קונפ' בריצתה על w.

ניתוח אפיון $\text{USELESS}_{\text{TM}}$
$\text{USELESS}_{\text{TM}} = \{\langle M\rangle \mid$ קיים מצב $q \notin \{q_{\text{acc}}, q_{\text{rej}}\}$ כך שלכל קלט w הריצה של M אינה עוברת ב-$q\}$.
נרצה להוכיח כי $\text{USELESS}_{\text{TM}} \notin \text{coRE} \setminus \text{RE}$.
נבצע רדוקציה עם $\overline{A_{\text{TM}}}$ כך ש-$\overline{A_{\text{TM}}} \leq_m \text{USELESS}_{\text{TM}}$ ונראה כי $\text{USELESS}_{\text{TM}} \notin \text{RE}$.

<u>בנייה</u>
בהינתן קלט $\langle M,w\rangle$, הרדוקציה פולטת $\langle K\rangle$ כאשר K פועלת באופן הבא:
בהינתן קלט x של K, K מסמלצת את ריצת M על w.
אם M מקבלת את w, K עוברת ל-$q_{\text{מיוחד}}$ ומשם היא מבקרת בכל המצבים הלא מקבלים והלא דוחים שלה, ואחרי זה היא עוצרת.
אם M אינה מקבלת את w, K דוחה (ולכן לא עברה ב-$q_{\text{מיוחד}}$).

<u>נכונות</u>
אם $\langle M,w\rangle \in \overline{A_{\text{TM}}}$ אזי M אינה מקבלת את w. כלומר, K אינה עוברת ב-$q_{\text{מיוחד}}$ לכל קלט x.
כלומר, יש מצב שהוא לא ישים, ובפרט $\langle K\rangle \in \text{USELESS}_{\text{TM}}$.
אם $\langle M,w\rangle \notin \overline{A_{\text{TM}}}$ אזי M תסמלץ את w, כלומר לכל קלט x של K, K עוברת ב-$q_{\text{מיוחד}}$ ולכן עוברת בכל המצבים שלה.
כלומר $\langle K\rangle \notin \text{USELESS}_{\text{TM}}$.

תרגול מס' 8:

יום שני

29.11.21

(בדבר)

הגדרה

תכונה היא אוסף של מ"ט.

הגדרה

תכונה P תיקרא סימנטית אם לכל מ"ט M_1, M_2, אם $L(M_1) = L(M_2)$ אזי $M_1 \in P$ אם"ם $M_2 \in P$.

דוגמאות

$P = \{M \mid L(M) \neq \emptyset\}$ היא תכונה סימנטית.

$P = \{M \mid$ יש ל-M 3 מצבים$\}$ היא איננה סימטרית.

הגדרה

תכונה P תיקרא "לא טריוויאלית" אם היא לא כל המכונות בעולם, וגם איננה הקבוצה הריקה.

אבחנה

P היא סימנטית לא טרוויאלית אם ורק אם $\overline{P}$ היא סימנטית לא טרוויאלית.

משפט רייס

תהי P תכונה סימנטית לא טרוויאלית, אזי $L_P = \{\langle M \rangle \mid M \in P\} \notin \mathrm{R}$.

דוגמה

$\mathrm{ALL_{TM}}$ היא תכונה סימטרית לא טרוויאלית ואכן היא איננה כריעה.

למה

תהי P תכונה סימטרית לא טרוויאלית ונניח כי $T_\emptyset \notin P$ (כאשר $L(T_\emptyset) = \emptyset$) אזי $A_{\mathrm{TM}} \leq_m L_P$.

הוכחת הלמה

בהינתן P תכונה סימטרית לא טרוויאלית, כך ש-$T_\emptyset \notin P$, נרצה להראות רדוקציה מ-A_{TM} ל-L_P.

נתבונן ב-$H \in P$.

<u>בנייה</u>

בהינתן קלט $\langle M, w \rangle$, הרדוקציה פולטת $\langle T \rangle$, כאשר T פועלת באופן הבא. בהינתן קלט x של של T, T מסמלצת את ריצת M על w.

אם M לא עצרה, T לא עצרה, אם M דוחה, אזי T דוחה. אם M מקבלת את w, אזי T מסמלצת את H על x ועונה כמו H.

כעת, עולה כי T מקבלת או דוחה או לא עוצרת על x, אם H מקבלת או דוחה או לא עוצרת על x, בהתאמה.

<u>נכונות</u>

אם M מקבלת את w, אזי לכל x, T מקבלת את x אם"ם H מקבלת את x. כלומר $L(T) = L(H)$.

מכיוון ש-P סימנטית ו-$H \in P$, נקבל כי $T \in P$ ולכן $\langle T \rangle \in L_p$.

אם M אינה מקבלת את w, במקרה זה T אינה מקבלת כל קלט x. כלומר $L(T) = \emptyset$, ולכן $T \notin P$. כלומר $\langle T \rangle \notin L_P$.

הוכחת המשפט

תהי P תכונה סימנטית לא טרוויאלית. ישנן שתי אפשרויות:

1. $T_\emptyset \notin P$ ולכן לפי הלמה $A_{\mathrm{TM}} \leq_m L_P$, כלומר $L_p \notin \mathrm{coRE}$.

2. $T_{\emptyset} \in P$ ולכן $T_{\emptyset} \notin \overline{P}$. מהאבחנה שראינו מקודם עולה כי $\overline{P}$ היא גם תכונה סימנטית לא טרוויאלית ולכן יש רדוקציה $A_{\text{TM}} \leq_m \overline{L_p}$, כלומר $\overline{A_{\text{TM}}} \leq_m L_p$ ולכן $L_P \notin \text{RE}$.

2.3 מכונות טיורינג אי דטרמיניסטיות

הגדרה

מכונת טיורינג אי דטרמיניסטית היא שביעייה:

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

כאשר δ מוגדרת על ידי:

$$\delta : Q \setminus \{q_{\text{rej}}, q_{\text{acc}}\} \times \Gamma \to 2^{Q \times \Gamma \times \{L,R\}} \setminus \{\emptyset\}$$

דוגמא

$\delta(q, \sigma) = \{\langle q_1, \sigma, L \rangle, \langle q_2, \sigma_2, R \rangle\}$.

למשל אם $C = ur_1q\sigma v$ אזי ייתכן כי $uq_1\sigma_1 v$ או כי $u\sigma_1\sigma_2 q_2 v$.

נסתכל על מכונה אי דטרמיניסטית M וקלט w של M.

הגדרה

בהינתן M ו-w, נאמר כי M מקבלת את w, אם קיימת ריצה מקבלת של M על w. M היא מכונה מכריעה אם לכל קלט w, כל הריצות של M על w עוצרות.

כלומר, $L(M) = \{w \in \Sigma^* \mid$ קיימת ריצה מקבלת של M על $w\}$.

דוגמה

$L = \{\langle n \rangle \mid$ אינו ראשוני $n\}$.

מ"ט אי דטרמינטסית M המכריעה את L עובדת באופן הבא:

בהינתן $\langle n \rangle$, M מנחשת מספר בינארי P, ביט אחרי ביט. לאחר מכן, בודקת אם P/n, אם היא מקבלת, אם לא היא דוחה.

כיצד היא מנחשת?

נכתוב תאים ריקים באופן דטרמיניסטי, לכל תא ריק כזה ננחש שנכתוב 0 או 1 בכל תא. עץ הקונפיגורציות (שנגדיר אותו בקרוב) יכיל עבור כל מספר $1 \ldots n$ את הריצה שלו, והאם מתקבלת או לא.

הגדרה

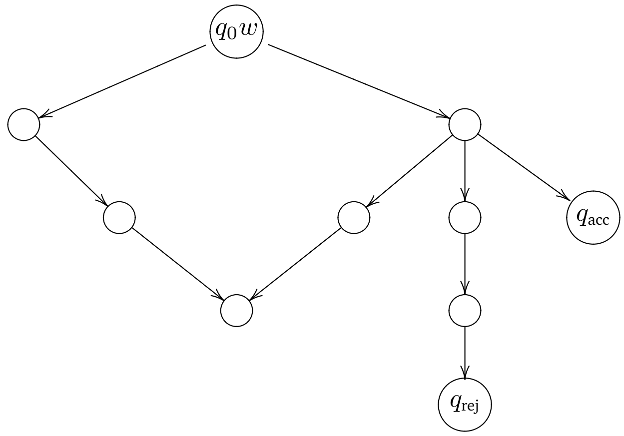
תהי M מכונה אי דטרמינסטית, ויהי קלט w עבור M.

נגדיר את עץ הריצות של M ל- w על ידי $T_{M,w} = \langle V, E \rangle$.

נסמן ב-C את קבוצת הקונפיגורציות של M.

$V \subseteq C \times (\mathbb{N} \cup \{0\})$ וגם $E \subseteq \bigcup_{i \geq 0} (C \times \{i\}) \times (C \times \{i+1\})$ כך ש-$(\langle C, i \rangle, \langle d, i+1 \rangle) \in E$ אם ורק אם d היא קונפיגורציה עוקבת של C (כל קשת בין רמות ויורדת).

נניח כי כל הקודקודים מ-V ישיגים מהשורש.



אבחנות

1. קיים קבוע K (שתלוי ב-$\langle M\rangle$ ולא תלוי ב-w) שחוסם את דרגת הפיצול המקסימלית של העץ.
2. M היא מכריעה אם ורק אם לכל w, $T_{M,w}$ סופי.

טענה

לכל מ"ט אי דטרמיניסטית N מכריעה יש מ"ט דטרמיניסטית D מכריעה עם $L(D)=L(N)$.

הוכחה

נוכל להתייחס לכל קודקוד בעץ שהינו קונפיגורציה, בתור כתובת.
כיצד D פועלת? בהינתן קלט x, לכל איטרציה i:

1. D רושמת את כל המילים $u\in\Sigma_k^*$ באורך i בסדר ליקסוגרפי.
2. D בודקת האם אחת מהמילים האלו היא כתובת של קונפיגורציה מקבלת ב-$T_{M,x}$. אם כן, D מקבלת.
3. אם כל הכתובת שחישבנו לא חוקיות, D דוחה.

נבחין זמן הריצה הוא לכל היותר $O\left(K^t\right)\cdot O\left(t^2\right)$ כאשר t הוא גובה העץ. אמנם, מתקיים כי:

$$O\left(t^2k^t\right)=2^{\log\left(t^2\right)+\log\left(k^t\right)}=2^{\log(t)+t\log(k)}=2^{O(t)}$$

אם יש קונפיגורציה לא בהכרח מכריעה, זה גם יעבוד.

תרגול מס' 9:

יום שני

06.12.21

(בדר)

3 סיבוכיות

3.1 רדוקציות פולינומאליות

3.1.1 הגדרות והקדמה

ראינו בהרצאה את ההגדרה של מחלקת NP.

הגדרה

המחלקה P היא מחלקת השפות שניתנות להכרעה בזמן פולינומיאלי, כלומר $\mathrm{P} = \bigcup\limits_k \mathrm{TIME}\left(n^k\right)$.

הגדרה

המחלקה NP היא מחלקת השפות שניתנות להכרעה בזמן פולינומיאלי עם מ"ט א"ד, כלומר $\mathrm{P} = \bigcup\limits_k \mathrm{NTIME}\left(n^k\right)$.

כמו כן, ראינו בהרצאה את ההגדרה השקולה, שקיים מוודא פולינומיאלי V עבור שפה L:

הגדרה

נאמר כי קיים מוודא פולינומיאלי ל-L אם:

$$L = \{w \mid \text{קיים } c \text{ פולינומיאלי ב-} w \text{ כך ש-} V \text{ מקבלת בזמן פולינומיאלי את } \langle w, c\rangle\}$$

הגדרה

יהיו $K, L \subseteq \Sigma^*$. רדוקציה פולינומית מ-K ל-L היא רדוקציית מיפוי מ-K ל-L הניתנת לחישוב על ידי מ"ט בזמן פולינומי. אם יש רדוקציה כזו מ-K ל-L, נסמן $K \leq_p L$.

ראינו בהרצאה כי עבור $K, L \subseteq \Sigma^*$,מתקיים כי $K \leq_P L$ אם:

1. $K \in \mathrm{P} \Rightarrow L \in \mathrm{P}$.
2. $K \in \mathrm{NP} \Rightarrow L \in \mathrm{NP}$.
3. $K \in \mathrm{coNP} \Rightarrow L \in \mathrm{coNP}$.

כמו כן, ראינו כי רדוקציות פולינומית הן **טרנזיטיביות**, כלומר לכל L_1, L_2, L_3 מתקיים כי $L_1 \leq_p L_2 \wedge L_2 \leq_p L_3$ גורר כי $L_1 \leq_p L_3$.

הגדרה

שפה L היא NP- קשה אם לכל $K \in \mathrm{NP}$ מתקיים כי $K \leq_p L$. (אם L היא NP- קשה ו-J $L \leq_p J$ אזי J היא NP-קשה).

הגדרה

נאמר כי שפה L היא NP-שלימה אם היא NP-קשה וגם $L \in \mathrm{NP}$.

טענה

אם L היא NP- שלימה, וגם $L \in \text{P}$, אזי $\text{P} = \text{NP}$.

הוכחה

נראה כי $\text{NP} \subseteq \text{P}$. תהי $K \in \text{NP}$. מכיוון ש-L היא NP- קשה, אזי $K \leq_P L$.
מכיוון ש-$L \in \text{P}$, נקבל ממשפט הרדוקציה כי $K \in \text{P}$.

3.1.2 רדוקציה פולינומיאלית לכיסוי בקודקודים

הגדרה

יהי $G = (V, E)$ גרף פשוט לא מכוון. קליקה ב-G היא תת קבוצה $C \subseteq V$ כך שלכל $x \neq y \in C$, יתקיים כי $\{x, y\} \in E$.

הגדרה

כיסוי קודקודים ב-G הוא תת קבוצה $C \subseteq V$ כך שלכל קשת $\{x, y\} \in E$, מתקיים כי $x \in C$ או $y \in C$.

נגדיר: $\text{CLIQUE} = \{\langle G, K \rangle \mid$ יש ב-G קליקה בגודל $K\}$, שנוכיח בהרצאה כי היא NP-שלימה.

כמו כן, נגדיר:

$\text{VC} = \{\langle G, K \rangle \mid$ יש ב-G כיסוי קודקודים בגודל לכל היותר $k\}$

(נוכל להשתמש ב'לכל היותר' ולא ב'בדיוק', כי ביטוי אחד מוכל בשני).

נוכיח כעת כי $\text{VC} \in \text{NP}$. מוודא פולינומי V עבור VC מקבל כקלט $\langle G, K \rangle \$S$ ו-V בודק:

1. $K = |S|$.
2. S היא כיסוי קודקודים: נעבור על כל צלע $\{x, y\} \in E$, ולכל צלע כזו נעבור על כל S בשביל לבדוק האם $x \in S$ או $y \in S$.

אם אין בדיקה עוברת, V דוחה.

טענה

VC היא NP-קשה.

הוכחה

נראה כי $\text{CLIQUE} \leq_P \text{VC}$, כלומר נראה כי יש ב-$G'$ כיסוי בגודל K' אם ורק אם יש ב-G קליקה בגודל K.

<u>בנייה</u>

בהינתן קלט $\langle G = (V, E), K \rangle$, הרדוקציה פולטת $\left\langle \overline{G} = \left(V, \overline{E}\right), n - K \right\rangle$ כאשר $n = |V|$.

<u>נכונות</u>

נניח שיש ב-G קליקה C בגודל K. נראה כי $\overline{C}$ היא כיסוי קודקודים ב-$\overline{G}$ בגודל $n - K$.
נניח בשלילה כי $\overline{C}$ אינה כיסוי ב-$\overline{G}$. כלומר, יש קשת $\{x, y\} \in \overline{E}$ כך ש-$x, y \notin \overline{C}$. אזי $x, y \in C$. כיוון ש-C קליקה, עולה כי $\{x, y\} \in E$ וזאת סתירה.

נניח ש-S היא כיסוי ב-$\overline{G}$ בגודל $n - K$, ונראה כי $\overline{S}$ היא קליקה ב-G בגודל K.
ראשית, ברור כי $|S| = |V \setminus S| = n - (n - K) = K$.

נניח בשלילה כי S אינה קליקה. כלומר, יש $x, y \in \overline{S}$ כך ש-$\{x, y\} \notin \overline{E}$. מכיוון ש-$S$ היא כיסוי ב-$\overline{G}$, נקבל כי $x \in S$ או $y \in S$.

<u>חישוב בזמן פולינומי</u>

נבחין כי המעבר הראשון (ההעתקה למשלים) אורך $O(|G|)$, והסריקה של המכונה (מעבר על כל זוגות הקודקודים ועל כל הקשתות) לוקחת $O\left(|V|^2 \cdot |E|\right)$- לכן הראינו כי כל הבנייה של המכונה אורכת בזמן פולינומיאלי.

3.1.3 רדוקציה פולינומילאית לקבוצה שולטת

הגדרה

נאמר כי קבוצה היא קבוצה שולטת בגרף C, אם כל הקודקודים נמצאים במרחק לכל היותר 1, מקודקוד זה.

נגדיר כעת את הקבוצה הבאה: $\text{DS} = \{\langle G, K\rangle \mid K$ יש קבוצה שולטת בגודל לכל היותר$\}$.

מוודא עבור השפה מעל $\langle G, K\rangle \$S$ בודק אם:

1. $|S| = K$.
2. S היא קבוצה שולטת בגודל K: נעבור על כל קודקוד v ונבדוק אם $v \in S$ (על ידי מעבר על S), ואם לא, נבדוק אם יש $u \in S$ וקשת $e \in E$ כך ש-$e = \{v, u\}$.

טענה

DS היא NP- קשה.

הוכחה

נראה כי DSVC $\leq_P$, כלומר, נראה כי יש ב-G' קבוצה שולטת בגודל K' אם ורק אם יש כיסוי קודקודים בגודל K.

<u>בנייה</u>

בהינתן קלט $\langle G = (V, E), K\rangle$, הרדוקציה פולטת $\langle G' = (V', E'), K'\rangle$, כאשר:

- $K' = K + f$, כש-f הוא מספר הקודקודים המבודדים ב-E.
- $V = V \cup \{v_e\}_{e \in E}$ ו-$E' = E \cup \{\{v_e, u\}, \{v_e, v\} \mid e = (u, v) \in E\}$.

<u>נכונות</u>

נניח כי יש ב-G כיסוי C בגודל K. נסמן ב-F את קבוצת הקודקודים המבודדים ב-G ונראה כי הקבוצה $S = C \cup F$ היא קבוצה שולטת ב-G' בגודל לכל היותר K', ונקבל:

$$|S| = |C \cup F| \leq |C| + |F| = K + f = K'$$

נראה כי $S = C \cup F$ היא קבוצה שולטת ב-G'. יהיה $x \in V$, ונחלק למקרים:

- אם $x = v_e \in V' \setminus V$. במקרה זה, קיימת $e = \{u, v\} \in E$ ו-$\{v, v_e\}, \{u, v_e\} \in E'$. מכיוון ש-$C$ כיסוי קודקודים ב-G, אזי $u \in C$ או ש-$v \in C$, ומכיון ש-v_e מחובר ל-u ו-v ב-G', אזי הוא במרחק לכל היותר 1 מ-$C \cup F$.

- x הוא מבודד ב-G', ולכן x מבודד ב-G. כלומר $x \in F$ ולכן $x \in S$. כלומר, הוא במרחק לכל היותר 1 מ-S.

- $x \in V \setminus F$, ולכן יש בהכרח שכן y של x ב-G. מכיוון ש-C כיסוי ב-G, אזי $x \in C$ או $y \in C$. בשני המקרים, x במרחק לכל היותר 1 מ-S.

נניח שב-G' יש קבוצה שולטת DS בגודל K' ונניח בה"כ כי DS אינה מכילה קודקודים חדשים. נתבונן בקבוצה $D = DS \backslash F$ ואז יתקיים:

$$|D| = |DS \backslash F| = K + f - f = K$$

כעת, נראה כי D היא כיסוי ב-G. תהי $e \in E = \{x, y\}$. נתבונן בקודקוד החדש ב-G' שנגדירו בתור v_e. נבחין כי v_e במרחק לכל היותר 1 מ-DS ומכיוון שהוא לא מבודד, אזי הוא במרחק לכל היותר 1 מ-D. מההנחה ש-$v_e \notin D$, נקבל כי v_e היא במרחק בדיוק 1 מ-D. כלומר, השכנים היחידים של v_e הם x ו-y ולכן $x \in D$ או $y \in D$ ולכן D היא כיסוי קודקודים ב-G.

<u>חישוב בזמן פולינומי</u>
ספירת הקודקודים המבודדים לוקחת $O\left(|V|^2\right)$, הוספת קודקודים לוקחת $O\left(|E|\right) = O\left(|V|^2\right)$ והוספת צלעות לוקחת $O\left(|V|^2\right)$ ולכן מדובר בזמן פולינומיאלי בגודל הגרף.

הגדרה
מסלול המילטוני בגרף הוא מסלול שעובר בכל הקודקודים, בדיוק פעם אחת.

תרגול מס' 10:

יום שני

13.12.21

(בדר)

3.1.4 הוכחת D-ST-HAMPATH כ-NP-קשה

ראינו כי D-ST-HAMPATH$=\{\langle G, s, t\rangle$ כך שבגרף המכוון יש מסלול המילטון מ-s ל-$t\}$, שייכת ל-NP.
ניתן להגדיר גם את D-HAMPATH $= \{\langle G\rangle \mid$ יש בגרף המכוון G מסלול המילטוניו$\}$ או את D-HAMCYCLE $= \{\langle G\rangle \mid$ יש בגרף המכוון G מעגל המילטוניו$\}$.
כך גם ניתן להגדיר את U-HAMPATH ואת U-ST-HAMPATH וגם את U-HAMCYCLE שמביא גרסה לגרפים לא מכוונים.

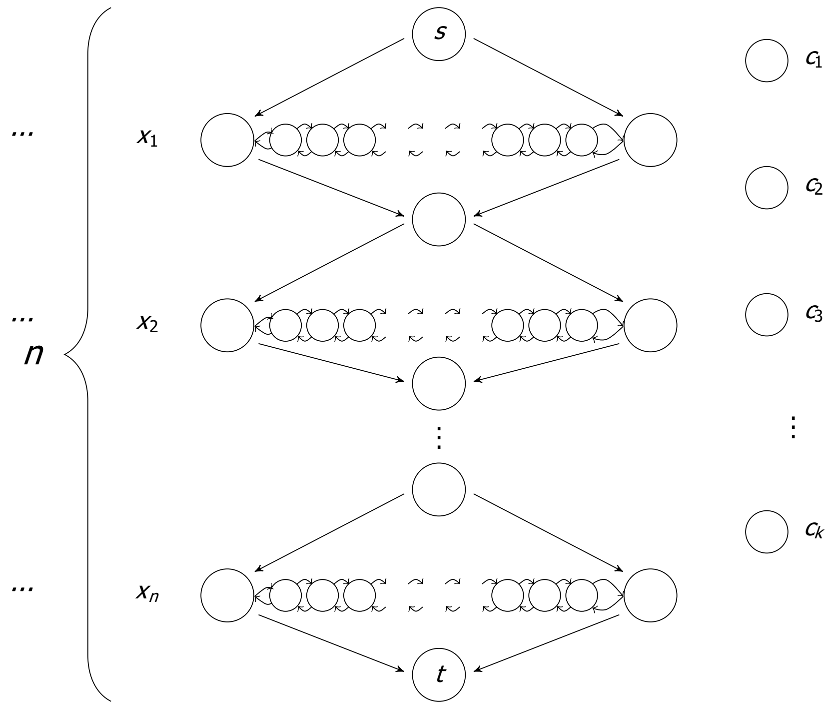
נראה כעת כי D-ST-HAMPATH היא NP-קשה. נעשה זאת באמצעות רדוקציה מ-3-SAT: כלומר, נראה כי 3-SAT $\leq_p$ D-ST-HAMPATH, על ידי כך שנראה כי נוסחה u ספיקה אם ורק אם יש בגרף המכוון G מסלול המילטוני מ-s ל-t.
תחילה, נגדיר מספר סימונים:

- נסמן ב-n את מספר המשתנים ב-φ, כך ש-$x_1, \ldots, x_n$ הם המשתנים

- נסמן ב-k את מספר הפסוקיות ב-φ, כך ש-$c_1, c_2 \ldots, c_k$ הן הפסוקיות.

<u>בנייה</u>
עבור קלט G, ניצור מבנה יהלום, לכל קודקוד x[12]:

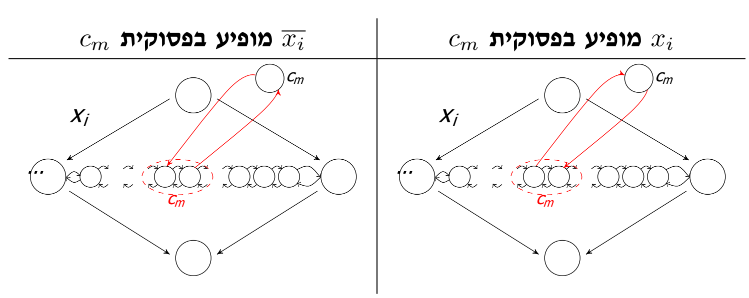
[12] התמונות בחלק זה בסיכום מתוך הסיכום של מאור מזרחי, סמסטר ב' 2021.



נשים לב לכמה נקודות חשובות בגרף בצורה זו שיצרנו אותו:

- הקודקודונים בשורות הארוכות, מתאימים לכל אחת מהפסוקיות, כך שלכל פסוקית נשים שני קודקודנים - למשתנה ושלילתו. מעבר לזה, נוסיף קודקודנים שהמטרה שלהם להפריד בין כל אחת מהפסוקיות.
- מתאפשר מעבר דו כיווני לקודקודוני x_i. כלומר, ניתן להגיע ל'צעד הבא', הן באמצעות צד ימין והן באמצעות צד שמאל.
- הקודקודים המבודדים מתאימים לכל אחת מהפסוקיות.
- טיול בצורת zigzag הוא הליכה דרך צד שמאל, ואז צד ימין, ואז צד שמאל וכו'.
- טיול בצורת zagzig הוא הליכה דרך צד ימין, ואז צד שמאל, ואז צד ימין וכו'.

כעת, נבחין כי אם x_i מופיע בפסוקית ה-c_m, נחבר את **צד שמאל** של הקודקודונים המתאימים לפסוקית, ל-c_m, 'ונחזור' לצד ימין שלהם.
אם x_i מופיע בפסוקית ה-c_m, נחבר את **צד ימין** של הקודקודונים המתאימים לפסוקית, ל-c_m, 'ונחזור' לצד שמאל שלהם.
אם גם x_i לא מופיע בפסוקית ה-c_m וגם $\overline{x_i}$ לא מופיע בפסוקית ה-c_m, נמשיך כרגיל עד הפסוקית הבאה.



אבחנה

עבור משתנה x ועבור פסוקית c המכילה את x, אם ההשמה $x = \mathbb{T}$ מספקת את c, אז אם נטייל בצורת zigzag ביהלום של x או אם ההשמה $x = \mathbb{F}$, ואם נטייל בצורת zagzig ביהלום של x, נוכל לצאת מהיהלום לבקר בקודקוד הפסוקית c ולעבור לקודקוד הבא בשרשרת, ולהמשיך את הטיול בצורה דומה.
למה? נחשוב רגע אינטואיטיבית. אם ההשמה של x היא $\mathbb{T}$, אז נרצה כי אם x מופיע בפסוקית c, נוכל להגיע ל-c ולהתקדם הלאה. זאת נוכל לעשות כיוון שהכיוון הוא 'כלפי ימינה'.

אבחנה 2

אם יש מסלול המטייל ביהלום המתאים למשתנה x בצורת zigzag ומסלול זה קופץ מהיהלום, נוגע בקודקוד c וחוזר לקודקוד הבא בשרשרת, אזי $x = \mathbb{T}$ מספקת את c.

אבחנה נוספת (של המסכם)

אנחנו לא בהכרח חייבים שכל המשתנים יקבלו ערך $\mathbb{T}$. כלומר, ברור שייתכן לפעמים כי חלק מהמשתנים או צירופם עם שלילתם יקבלו ערך $\mathbb{F}$, הרי מספיק לנו שבכל פסוקית יהיה ערך $\mathbb{T}$ אחד.

נניח כי $S : \{x_1, \ldots, x_n\} \to \{\mathbb{F}, \mathbb{T}\}$ היא השמה מספקת של φ ונתאר מסלול המילטוני מ-s ל-t:

- נתחיל ב-s.
- עבור $1 \leq i$:
  - אם $S(x_i) = \mathbb{T}$ נטייל ביהלום ה-i בצורת zigzag ואחרת, נטייל בצורת zagzig.
  - עבור כל פסוקית c, אם לא ביקרנו ב-c, וגם $S(x_i)$ מספקת את c, נעבור (בזכות אבחנה 1) לבקר ב-c ונחזור להמשיך את הטיול שלנו ביהלום באותה צורה.

בצורה כזאת, יצרנו למעשה **מסלול המילטוני** כי עברנו בין כל הקודקודים ולכן ביקרנו ב-C_m פעם אחת, ובכל הקודקודים פעם אחת, וסיימנו[13].

בכיוון השני, נניח כי יש מסלול המילטוני מ-s ל-t, P.

הגדרה

נאמר כי P הוא נחמד אם הוא עובר על היהלומים לפי הסדר ועובר בכל יהלום בצורת zigzag או zagzig, ואם הוא יוצא מיהלום ומבקר בקודקוד c אזי הוא חוזר מיד לקודקוד הבא בשרשרת.

אבחנה

אם מישהו לא נחמד, מתישהו הוא עושה את האקט הלא נחמד הראשון שלו.

עלינו להראות כי P נחמד.
נניח בשלילה כי P לא נחמד ונתבונן בפעם הראשונה את האקט הלא נחמד הראשון שלו. כלומר, עובר ליהלום אחר, ולא הקודקוד הבא בשרשרת.
במקרה זה, אנחנו מבקרים ב-a ואחרי זה ב-c. בגלל ש-P הוא המילטוני, מתישהו נבקר ב-b. מכיוון שביקרנו ב-c ו-a הדרך היחידה שבה נוכל לבקר ב-b היא דרך השכן d, אבל במקרה זה נתקענו ב-b, בפרט לא נוכל להגיע ל-t.

נתבונן כעת ב-U-ST-HAMPATH - כל הגרפים הלא מכוונים, שיש מסלול המילטוני מ-s ל-t.

[13] האופציה לעבור לא מחייבת אותנו לעבור במסלול זה.

נבצע רדוקציה $\text{U-ST-HAMPATH} \leq_p \text{D-ST-HAMPATH}$.

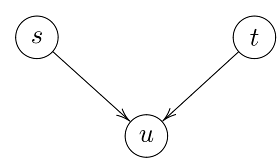
כלומר, עלינו להראות כי יש בגרף המכוון מסלול המילטוני מ-s ל-t אם"ם יש בגרף הלא מכוון מסלול המילון מ-s' ל-t'.

הורדת הכיוונים לא תעזור, כיוון שאנו מוסיפים כיוונים שלא היו בגרף המקורי.

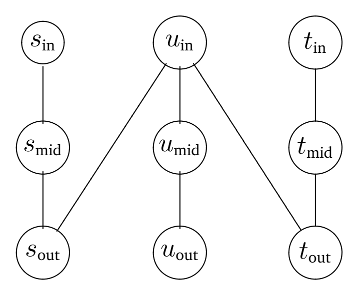
<u>בנייה</u>

בהינתן גרף מכוון $G=(V,E)$ וקודקודי s,t, הרדוקציה פולטת את הגרף הלא מכוון $G'=\left(V',E'\right)$ ו-$s_{\text{in}},t_{\text{out}}$ כאשר $V'=\{v_{\text{in}},v_{\text{mid}},v_{out} \mid v\in V\}$ ו-$E'=\{\{v_{\text{ in}},v_{\text{ mid}}\},\{v_{\text{mid}},v_{\text{out}}\} \mid \in V\text{v}\}\cup\{\{u_{\text{out}},v_{\text{in}}\} \mid (u,v)\in E\}$.

דוגמה



יעבור ל:



<u>נכונות</u>

נניח כי $P=s,u^1,u^2,\dots,u^k,t$ הוא מסלול המילטוני ב-G.

נתבונן במסלול הבא: $s_{\text{ in}},s_{\text{mid}},s_{\text{out}},u^1_{\text{in}},u^2_{\text{mid}},u^3_{\text{out}},u^2_{\text{in}}\dots u^k_{\text{in}}\dots t_{\text{mid}},t_{\text{out}}$ ומדובר במסלול המילטוני ב-G'.

נניח כי יש מסלול המילטוני P מ-s_{in} ל-t_{out}. נראה כי P לא יכול להשתמש במעברי $\text{in}\to\text{out}$. נניח בשלילה כי הוא לא עושה זאת, ונתבונן בפעם הראשונה שהוא עושה זאת $(v_{\text{in}},h_{\text{out}})$. כעת, עבור v_{mid} ישגם שתי אופציות:

- אם v_{mid} מופיע ב-P לפני v_{in} אזי ביקרנו ב-v_{mid} דרך v_{out}, ולכן הגענו ומכיוון ש-P מתחיל מ-in של קודקוד כלשהו, נקבל כי הגענו ל-v_{out} דרך קודקוד אחר, ששונה מ-v_{mid}, כלומר הגענו מ-in של קודקוד כלשהו, בסתירה למינימליות

- אם v_{mid} מופיע ב-P אחרי v_{out}, אזי במקרה זה הגענו ל-v_{mid} דרך v_{out} וכבר ביקרנו ב-v_{in} ולכן אנו תקועים ב-v_{mid}, כי ביקרנו בכל שכניו. בפרט, P לא מגיע ל-t_{out}.

אם כך, P לא כולל מעברי $\text{in}\to\text{out}$ ולכן בפרט המסלול הוא מהצורה $s_{\text{ in}},s_{\text{mid}},s_{\text{out}},u^1_{\text{in}},u^2_{\text{mid}},u^3_{\text{out}},u^2_{\text{in}}\dots u^k_{\text{in}}\dots t_{\text{mid}},t_{\text{out}}$ ולכן בפרט המסלול ההמילטוני הוא $s,u^1,u^2,\dots,u^k,t$, כי אנו עוברים בכל הקודקודים ועוברים פעם בדיוק פעם אחת.

חישוב

מדובר בזמן פולינומיאלי, כי רק מכפילים את הקודקודים בתלות ב-V ומוסיפים קשתות כתלות ב-E ו-V.

3.2 קשרים בין מחלקות

הגדרה

נגדיר את co-NP להיות $\text{co-NP} = \left\{L \mid \overline{L} \in \text{NP}\right\}$.

הגדרה

L היא co-NP-קשה אם לכל $K \in \text{co-NP}$ מתקיים כי $K \leq_p L$ ואם בנוסף L ב-co-NP אזי נאמר ש-L היא co-NP-שלימה.

טענה

L היא NP-קשה אם ורק אם $\overline{L}$ היא co-NP קשה.

הוכחה

נניח כי L היא NP-קשה ותהי $K \in \text{co-NP}$, אזי $\overline{K} \in \text{NP}$.

לכן עולה כי $\overline{K} \leq_p L$ ולכן $K \leq_p \overline{L}$.

הכיוון השני דומה.

דוגמה

נתבונן בשפה הבאה:

$$\overline{\text{SAT}} = \{\langle u\rangle \mid \text{אין השמה מספקת של } u\}$$

כלומר, כל ההשמות של u אינן מספקות.

מדובר בשפה co-NP שלימה.

השפה $\overline{\text{SAT}}$ שייכת אם כן ל-co-NP. בנוסף, נבחין כי $\text{CONTRATICTION} = \{\langle\varphi\rangle \mid \text{אין ל- השמה מספקת}\varphi\}$ שקולה ל-$\overline{\text{SAT}}$ כי כל נוסחה אפשר להמיר ל-CNF.

תרחישים אפשריים:

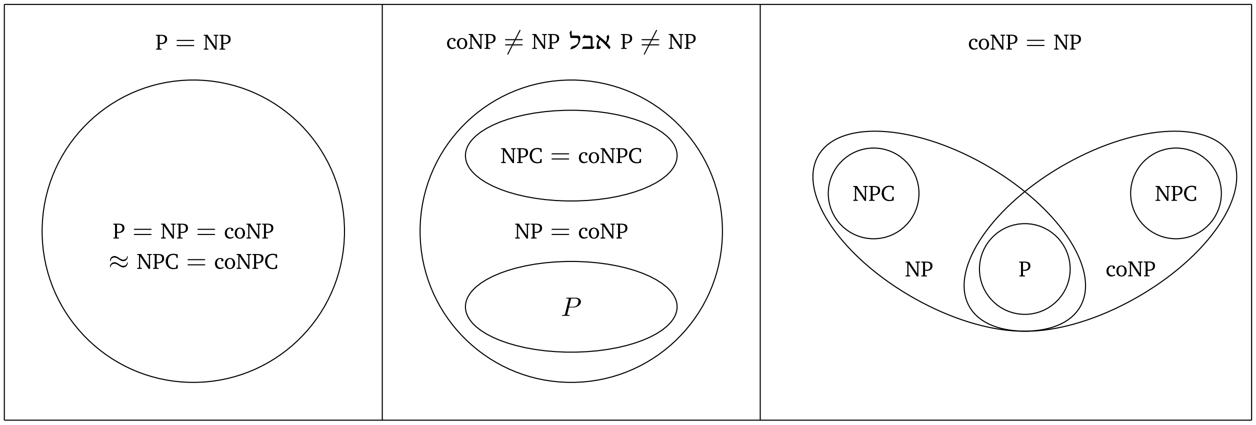
תרגול מס' 11:

יום שני

20.12.21

(בדר)

1. NP=P. במקרה זה, coNP=NP וגם 'כמעט' שוות ל-NPC ולכן NPC=coNP.
2. NP≠P. במקרה זה, ייתכנו שתי אפשרויות:
   (א) coNP=NP ולכן NPC=coNP.
   (ב) coNP≠NP ולכן NPC≠coNP והן אפילו זרות לחלוטין.



3.2.1 שאלות ממבחנים לגבי P או coNP

תהי $A = \{\langle\varphi\rangle \mid$ φ היא נוסחת CNF כך שקיימת השמה מספקת ל-φ עם בדיוק 10 משתנים שהם $\mathbb{T}$ והשאר $\mathbb{F}\}$.
נרצה להוכיח כי שפה זו ב-P.
אלגוריתם פולינומי עבור L עובד באופן הבא:
בהינתן נוסחה בוליאנית φ מעל המשתנים $x_1, x_2 \ldots, x_n$, נבדוק אם φ היא מצורת CNF. אם לא, נדחה, אחרת נמשיך.
נעבור על כל תתי הקבוצות $S \subseteq \{1, \ldots, n\}$ בגודל 10, ועבור קבוצה S כנ"ל, נתבונן בהשמה:

$$\sigma_s(x_i) = \begin{cases} \mathbb{T} & i \in S \\ \mathbb{F} & i \notin S \end{cases}$$

נבדוק (בזמן פולינומיאלי) ש-σ_S מספקת את φ , ואם כן נקבל. אחרת, אם כל ההשמות עבור כל הקודקודים S אינן מספקות, נדחה.

<u>נכונות</u>

האלגוריתם מקבל אם ורק אם יש תת קבוצה S שההשמה σ_S מספקת את φ, אם ורק אם יש השמה מספקת שנותנת ערך $\mathbb{T}$ בדיוק ל-10 משתנים.

<u>חישוב</u>

עלינו לשים לב כי אנו עוברים מספיק פולינומיאלי של השמות, כלומר:

$$\binom{10}{n} = \frac{n(n-1)\ldots(n-9)}{10!} \leq n^{10}$$

נוסחה מאוזנת

הגדרה
נאמר שנוסחה בוליאנית φ מצורת CNF מאוזנת אם יש השמה σ כך ש:

- σ מספקת את φ.
- σ נותנת ללפחות $\frac{1}{3}$ משתנים ערך $\mathbb{T}$.
- σ נותנת ללפחות $\frac{1}{3}$ משתנים ערך $\mathbb{F}$.

נגדיר את השפה $L = \{\varphi \mid$ מאוזנת CNF היא נוסחה מצורת $\varphi\}$.
נרצה להוכיח כי בעיה זו היא NP-שלימה.
תחילה, עלינו להוכיח שבעיה זו היא ב-NP.

<u>בנייה</u>
בהינתן קלט φ עם n משתנים, הרדוקציה מייצרת φ' כאשר φ' מוגדרת על ידי:

$$\varphi' = \varphi \wedge \underbrace{\left(\bigwedge_{i\in[n]} x_i \vee x_i \vee x_i\right)}_{\varphi_{\mathbb{T}}} \wedge \underbrace{\left(\bigwedge_{i\in[n]} \overline{y_i} \vee \overline{y_i} \vee \overline{y_i}\right)}_{\varphi_{\mathbb{F}}}$$

כאשר $x_1, \ldots, x_n$ ו-$y_1, \ldots, y_n$ הם משתנים חדשים.

<u>נכונות</u>
אם קיימת השמה מספקת ל-φ, אזי נוכל להרחיב אותה להשמה σ המספקת את φ' ע"י זה שניתן לשים ערך $\mathbb{T}$ לכל x_i וערך $\mathbb{F}$ לכל y_i. נשים לב כי σ מספקת את φ', כי ההשמה המקורית מספקת את φ וגם ההשמה $x_i = \mathbb{T}$ מספקת את $\varphi_{\mathbb{T}}$ וההשמה $y_i = \mathbb{F}$ מספקת את $\varphi_{\mathbb{F}}$. נבחין כי φ מאוזנת כי σ נותנת $\mathbb{T}$ ללפחות n משתנים ו-$\mathbb{F}$ ללפחות n משתנים.

אם יש השמה σ שמספקת את φ' ונותנת ערך $\mathbb{T}$ ללפחות $\frac{1}{3}$ משתנים וערך $\mathbb{F}$ ללפחות $\frac{1}{3}$ משתנים, אזי מכיוון שהיא מסופקת היא מספקת את הביטוי שיש ב-φ ומכיוון שב-φ לא מופיעים משתנים חדשים, נקבל כי הצמדים של σ על המשתנים המקוריים מספקים את φ.