Kiril Solovey

מודלים חישוביים – סיכום

כריעות

<u>טענה:</u> לא כל הפונקציות חשיבות.

- מספר התוכניות הוא בן מניה.
- כל תוכנית מגדירה פונקציה מספרית אחת לכל היותר.
- לכן מספר האלגוריתמים הוא בן מניה בעוד שמספר הפונקציות המספריות אינו בן מניה.

<u>דוגמאות לתוכניות לא חשיבות</u>

$$bb(n) = max\{[p(\,)]|\, p \in L_0. |p| \leq n\}$$
$$mm(n) = min\{k|\, k \in \mathbb{N}, \forall p \in L_0, |p| \leq n \rightarrow [p(\,)] \neq k\}$$
$$minProg(n) = min\{|p|\,|\, p \in L_0, [p(\,)] \geq n\}$$
$$halt(x) = \{if\ x\ is\ a\ program\ that\ halts\ return\ T, else\ F\}$$

רדוקציה

נאמר שיש רדוקציה מפונ' f לפונ' g ונסמן $f \leq g$ אם באמצעות תכנית שמחשבת את g ניתן לכתוב תכנית שמחשבת את f. אם קיימת רדוקציה מ-f ל-g אז:

- אם f לא חשיבה, אז גם g לא חשיבה.
- אם g חשיבה, אז גם f חשיבה.

בעיית ההכרעה

נאמר ששפה היא **כריעה** \ **שבעיית ההכרעה היא כריעה** אמ"מ קיימת תכנית p שמקיימת:

1. אם $x \in L$ אז $p(x) = T$
2. אם $x \notin L$ אז $p(x) = F$

R

זוהי מחלקת השפות הכריעות.
מחלקה זו סגורה תחת פעולת **המשלים**.

רדוקציית מיפוי

נאמר שיש פונ' מיפוי משפה A לשפה B אם קיימת פונ' חשיבה f כך ש- $x \in A$ אמ"מ $f(x) \in B$. נסמן: $A \underset{m}{\leq} B$.

אז מתקיים:

- אם B היא RE, אז כך גם A.
- אם A אינה RE, אז גם B אינה RE.
- אם A אינה $Co\text{-}RE$, אז גם B אינה $Co\text{-}RE$.
- $A \underset{m}{\leq} B \Leftrightarrow \overline{A} \underset{m}{\leq} \overline{B}$.

<u>בעיות סופיות</u>
כל בעיה עם מספר סופי של בדיקות היא כריעה.
למשל: האם גיל קופטש יהיה חבר כנסת בכנסת ה-19?

משפט רייס

P היא בעיה סמנטית אם $f \equiv g \leftarrow P(f) \equiv P(g)$

השפה L אינה כריעה לפי רייס אם:

1. L היא שפה של תוכניות ($<p,x>$ אינה תוכנית!)
2. L היא בעיה לא טריויאלית
3. L היא בעיה סמנטית

<u>דוגמאות לבעיות לא כריעות לפי רייס:</u>

- $Pconst \coloneqq \{<p>|p\ halts\ on \leq i\ inputs\}$
- $Fin = \{<p>|p\ halts\ in\ a\ finite\ \#\ of\ inputs\}$

<u>שקילות סמנטית</u>

- אם S שפה סמנטית אז אם $a,b \in S$ לא בהכרח נובע כי a,b שקולות סמנטית.
- מצד שני, אם a,b שקולות סמנטית, אז שניהן בהכרח באותה שפה סמנטית

<u>סוגי בעיות סמנטיות:</u>

- type A: $P(\perp) = false$
- type B: $P(\perp) = true$

<u>משפט הרקורסיה</u>
לכל תוכנית $f: Scheme \rightarrow Scheme$ קיימת תוכנית c כך ש-$f(c) \equiv c$.

RE

RE היא מחלקת השפות שניתנות להכרעה למחצה, כלומר נאמר שתוכנית p מקבלת\מכריעה למחצה שפה L:

- אם $x \in L$ אז p מקבלת את x. $p(x) = T$.
- אם $x \notin L$ אז p דוחה את x או לא עוצרת. $p(x) = F \backslash \perp$.

<u>תכונות:</u>

- $L \notin RE$ לא בהכרח גורר ש-$L \notin Co-RE$
- $R = RE \bigcap Co-RE$
- $L \in RE \Leftrightarrow \bar{L} \in CO-RE$
- $L \in R \Leftrightarrow L, \bar{L} \in RE$
- לכל בעיה שהיא או משלימתה ב-RE ניתן לעשות רדוקציית מיפוי מ-$halt^*$ (כל התוכניות שמתבדרות לכל קלט)
- לכל בעיה ב-RE יש רדוקציית מיפוי ל-$halt_0$ (תוכניות ללא קלט שעוצרות)

שפות חשובות:

שפה	קלט	תאור	קבוצה	הסבר
$halt$	$<p,x>$	p עוצרת על הקלט x	$RE \setminus R$	השפה לא כריעה כי אחרת ניתן לפתור איתה את bb. בנוסף ב-RE כי ניתן להריץ את p על x בקלטים ששייכים לשפה ולחכות שיעצרו.
$halt_\varepsilon$	$<p>$	p עוצרת על הקלט הריק	$RE \setminus R$	ניתן להראות כי היא ב-RE בדומה לשפה הקודמת. כדי להראות שלא ב-$coRE$ מראים רדוקציה

				מהשפה $halt$.
Fin	$< p >$	התוכנית p עוצרת על מספר סופי של קלטים	$\notin RE$	לא ב-R ולא ב-REע"י רדוקציה מ-$\overline{halt}$
$halt^*$	$< p >$	התוכנית p עוצרת על כל קלט	$\notin RE, coRE$	רדוקציה מ-$halt$ ומ-$\overline{halt}$
$empty$	$< p >$	התוכנית p דוחה כל קלט או מתבדרת	$coRE \setminus R$	לא ב-RE לפי רדוקצייה מ-$\overline{halt}$. ב-$coRE$ כי ניתן לעבור על כל הקלטים עד שנתקלים בקלט שעליו התוכנית עוצרת.

STEPPER

זו תוכנית המקבלת p תוכנית, x קלט לתוכנית, k מספר טבעי.
תכונית זו מחזירה T אם p עצר על הקלט x בלכל היותר k צעדים, מחזירה F אחרת.
תוכנית זו היא כמובן <u>כריעה</u>

אנומרביליות

שפה L בעלת אנומרטור אם קיימת תוכנית $f: \mathbb{N} \to L$ שהיא על.

<u>טענה:</u> לכל שפה כריעה למחצה קיים אנומרטור.
<u>טענה:</u> לשפה קיים אנומרטור מונוטוני אמ"מ היא כריעה.
<u>טענה:</u> לכל שפה כריעה למחצה אינסופית קיימת תת שפה אינסופית כריעה. (ניתן ליצור אנומרטור מונוטוני)

סיכום שיטות הוכחה

בעיה L היא <u>כריעה</u> אם:
- מציגים אלגוריתם הפותר את הבעיה
- הבעיה היא סופית
- מראים רדוקציה לבעיה כריעה אחרת
- הבעיה והמשלים שלה כריעים למחצה
- מראים אנומרטור מונוטוני לשפה

כדי להוכיח שבעיה L <u>אינה כריעה</u> ניתן להשתמש ב:
- בשיטת הליכסון, למשל במקרה ה-BB הראנו כי הבעיה סותרת את עצמה
- רדוקציה (לאו דווקא מיפוי) מבעיה לא כריעה אחרת
- משפט $RICE$
- הצגת הוכחה הדומה להוכחת משפט $RICE$

בעיה L <u>כריעה למחצה</u> אם:
- מציגים אלגוריתם המכריעה למחצה את השפה
- מציגים רדוקציית מיפוי לבעיה כריעה למחצה
- אם קיים אנומרטור לשפה

המשלים של L <u>אינה כריעה למחצה</u> אם:
- משתמשים בשיטת הליכסון
- רדוקציית מיפוי מבעיה לא כריעה למחצה

- מראים ש-$\bar{L} \in RE \setminus R$

תובנות ממבחנים

- אם ידוע כי $L \in R$ והיא סמנטית, אזי היא טריויאלית (לפי רייס)
- אומנם קיימת מכונה שמכריעה את Σ^* ולכן היא ב-R, אך נשים לב שלא ניתן לייצר למכונה הזו קלט שלא בשפה. באופן סימטרי עבור השפה $\emptyset$ לא ניתן לייצר קלט שמתקבל בשפה.
- אם קיימת שפה L כך שמתקיים $\emptyset \leq_m L$ אז L היא בהכרח לא השפה המלאה. באותו אופן אם $\Sigma^* \leq_m L$ אז L בהכרח לא השפה הריקה.
- שפת כל המכונות שעוצרות שלמה ב-RE

מודלים חישוביים

היסטוריה חישובית חוקית - סדרת קונפיגורציות סופית:

1. קונפ' התחלה C_0 (במכונת מונים זה $\{a_1, 0, \dots, 0,1 | a_1 \in \mathbb{N}\}$)
2. $\forall i\ (C_i, C_{i+1})$ הוא מעבר חוקי.
3. C_n הוא קונפ' סיום.

Checker(p,x,c) : תכנית שמחזירה T אמ"מ c היא היסטוריה חישובית חוקית של ריצת p על x.

מכונת מונים

<u>מכונת מונים עם k רגיסטרים:</u> $CM = (\Lambda, R)$

Λ: סידרה סופית של הוראות $l_1, \dots, l_m$. $l_i \in \{inc\ j, dec\ j, if\ j = 0\ go\ to\ l_s, halt | 1 \leq j \leq k, l_s \in \Lambda\}$.

R: $r_1, \dots, r_k$ – סדרת רגיסטרים לא חסומים.

<u>קונפיגורציה:</u> במכונת מונים $\{(a_1, \dots, a_k, j) | \forall i\ a_i \in \mathbb{N}, l_j \in \Lambda\}$.

בהנתן קונפיגורציה ומכונת מונים ניתן להמשיך את החישוב.

<u>תכונות:</u>

- מכונת מונים עם 2 מונים לא יכולה לחשב כל דבר, למשל את הפונ': $n \to n^2$.
- ניתן לסמלץ מכונה עם 4 מונים ע"י מכונה עם 2 מונים: מונה אחד ישמור מספר מהצורה $2^i 3^j 5^k 7^l$ אשר ייצג את מצבי ארבעת המונים במכונה הראשונה.
- אם קיימת מכונת מונים עם 3 מונים המחשבת *f(n)* אז קיימת מכונת מונים עם 2 מונים המחשבת $2^{f(n)}$.
- כל מה שניתן לעשות עם מכונת n מונים, ניתן לעשות עם מכונת 3 מונים.
- בעיית העצירה במכונת מונים עם מונה אחד אינה כריעה.
- איך בודקים שמכונת מונים עם 2 סרטים מאפסת את אחד המונים במהלך הריצה על קלט ספציפי? מריצים את המכונה m פעמים (מספר שורות הקוד) – בהכרח אחד מהמצבים יחזור על עצמו (עקרון דיר החזירים). לכן אם נריץ את המכונה m פעמים * מספר האיברים ברגיסטר המקסימלי בתחילת הריצה, אם לא נגיע ל-*0*, לעולם לא נגיע.

מכונת טיורינג

$$M = (Q, \Sigma, \Gamma, q_0, q_a, q_r, \delta)$$

Q: קב' סופית של מצבים
Σ: א"ב של שפה (רווח לא שייך לא"ב)
Γ: א"ב סרט סופי ($\Sigma \subseteq \Gamma$, הפעם רווח שייך ל-א"ב)
q_0: מצב התחלה
q_a: קבוצת מצבים מקבלים
q_r: קבוצת מצבי דחייה
δ: פונקציית המעבר

<u>מספר הקונפיגורציות במכונת טיורינג המשתמשת ב-N תאים:</u>

$$\underbrace{|\Gamma|^N}_{\text{תוכן הסרטן}} \times \underbrace{N}_{\text{מיקום הראש}} \times \underbrace{|Q|}_{\text{מצב המכונה}}$$

<u>כל הגרסאות הבאות שקולות למכונת טורינג רגילה:</u>
- התחלת הסרט מסומנת
- ניתן להשאר במקום אחרי כתיבה
- מותר לכתוב פעם אחת בלבד על הסרט
- מספר סרטים
- סרט דו כיווני
- גרסה לא דטרמיניסטית

<u>סיבוכיות במכונות טורינג שונות:</u>
אם מכונת טורינג עם מספר סרטים מכריעה בעיה L בזמן $t(n)$ אז קיימת מכונת טורינג עם סרט יחיד המכריעה את הבעיה בזמן $O\left(t^2(n)\right)$.

שקילות מודלים

ניתן להראות כי המודלים הבאים שקולים:
- מכונת טורינג
- מכונת מונים עם 2 מונים
- RAM
- $Scheme$

מקרים בהם בעיית העצירה כריעה

- אורך התוכנית והקלט מוגבלים
- קיים חסם למספר הצעדים שעושה תוכנית שעוצרת.
- יש חסם על מספר הקונפיגורציות

סיבוכיות זמן

<u>הגדרה:</u> עבור פונ' $T: \mathbb{N} \to \mathbb{N}$ נגדיר את $DTIME\left(T(n)\right)$ בתור אוסף הבעיות שניתן להכריע בזמן $O\left(T(n)\right)$, כאשר n הוא אורך הקלט.

$$t_1(n) = O\left(t_2(n)\right) \to DTIME\left(t_1(n)\right) \subseteq DTIME\left(t_2(n)\right)$$

<u>היררכיית זמן:</u>

$$t_1(n) = O\left(\frac{t_2(n)}{log\, n}\right) \Rightarrow DTIME\left(t_1(n)\right) \subsetneq DTIME\left(t_2(n)\right)$$

המחלקה P

$$P = \bigcup_{c\geq 1} DTIME(n^c)$$

המחלקה NP

NP – בעיות שקלות לוידוא

<u>הגדרה</u>

שפה L היא ב-NP אם קיים פולינום $p: \mathbb{N} \to \mathbb{N}$ ומ"ט שרצה על בזמן פולינומיאלי (באורך הקלט) כך שלכל x:

$$x \in L \Leftrightarrow \exists \underbrace{w}_{x \text{ עדות עבור}} \in \Sigma^{p(|x|)}\ s.t.\ \underbrace{M(x,w) = T}_{\text{מוודא}}$$

<u>טענה:</u>

$$P \subseteq NP \subseteq EXP = \bigcup_{c\geq 1} DTIME\left(2^{n^c}\right)$$

רדוקציית מיפוי פולינומיאלית

נסמן $A \underset{p}{\leq} B$ אם קיימת פונ' חשיבה <u>ופול'</u> f כך ש-B $f(x) \in B \Leftrightarrow x \in A$.

<u>טענה:</u>

תהי $A \underset{p}{\leq} B$, אזי: אם $B \in P(NP)$ אז $A \in P(NP)$

אם $A \notin P(NP)$ אז $B \notin P(NP)$

NP-Hard

$$NPHard = \{L | \forall L` \in NP, L` \underset{p}{\leq} L\}$$

NP-Complete

$$NPC = \{L \mid L \in NPHard\ \ \&\ \ L \in NP\}$$

<u>טענה:</u>

אם $A \in NPC$, $B \in NP$ ו- $A \underset{p}{\leq} B$ אז $B \in NPC$

אם $A \underset{p}{\leq} B$, $A \in NPHard$ אז $B \in NPHard$

דוגמאות לבעיות

<u>בעיות ב-P:</u>

- בדיקת ראשוניות של מספר
- $2SAT$: פסוקים מהצורה $(x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge \ldots \wedge (x_{n-1} \vee x_n)$
- מסלול אוילר

<u>בעיות ב-NP:</u>

- $3SAT$
- SAT
- *Graph Colorability*
- *Independent Set*

<u>בעיות ב-$coNP$ ו-NP:</u>

- $factoring$: מקבלים שני מספרים וצריך לומר האם יש לשניהם מחלק משותף.

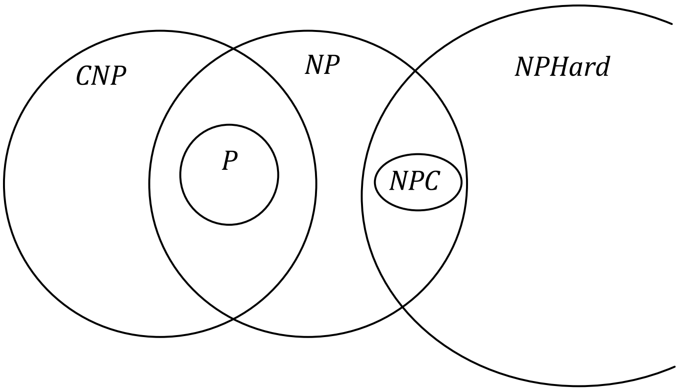
<u>בעיות ב-NPC:</u>

- $3SAT,SAT$
- $Hamilton\ Path$
- $Clique$
- IS - קיימת קבוצת קודקודים בגודל k בגרף G כך שאין קשת המחברת בין 2 מהקודקודים הללו
- $VC=\{<G,k>\mid$ האם ב-G קיים VC בגודל $k\}$
- $Subset\ Sum$, בהינתן קבוצה של מספרים, האם קיימת תת קבוצה שסכום איבריה הוא t.

<u>בעיות ב-$NPHard$ וגם ב-$coNPHard$ $(L\in NPHard \Leftrightarrow \bar{L}\in coNPHard)$:</u>

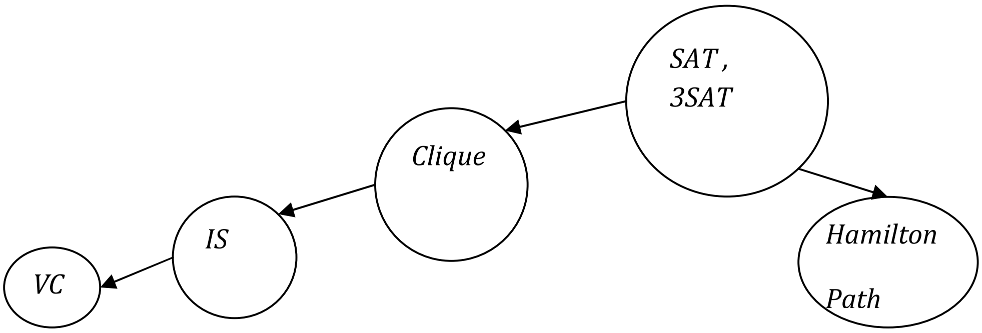
- $IS\oplus Clique=\{<G,k>|k\leq there's\ clique\ XOR\ IS\}$
- $Halt$

תאור הקבוצות:



בעיות ב-NPC

(החצים מציינים רדוקציות מיפוי פול")



סיבוכיות מקום

מודל לחישוב סיבוכיות מקום

סרט הקלט – קריאה. ראש נע לשני הצדדים.
סרט עבודה – קריאה וכתיבה. ראש נע לשני הצדדים.
סרט פלט – כתיבה. נע רק ימינה.

חישוב המקום הוא ע"פ כמות המקום בו השתמשנו בסרט העבודה.

$$\underbrace{|\Sigma|^N}_{\text{סרט הקלט}} \times \underbrace{|\Gamma|^S}_{\text{סרט העבודה}} \times \underbrace{N}_{\text{מיקום הראש בקלט}} \times \underbrace{S}_{\text{מיקום הראש בעבודה}} \times \underbrace{Q}_{\text{מצב נוכחי}}$$

PSpace

עבור פונ' $S: \mathbb{N} \to \mathbb{N}$, נאמר ש- $SPACE(S(N))$ היא אוסף השפות המוכרעות במקום $O(S(n))$.
נגדיר $PSpace = \cup_{c=1} Space(n^c)$

<u>בעיות ב-$PSPACE$:</u>

- $QSAT$

בעיית העצירה של מכונת טיורינג עם מקום מוגבל היא כריעה, מאחר ודרוש מקום פול' על מנת לבדוק האם יש קונפיגורציה של המכונה החוזרת על עצמה.

שפות רגולריות \ אוטומטים

<u>טענות:</u>

- $A \in DFA$ או $A \in NFA \Leftarrow x \in L(A)$ כריעה.
- כל שפה <u>סופית</u> היא רגולרית
- $L(A) = L(B)$ <u>כריעה</u>.

אוטומט סופי דטרמיניסטי - *DFA*

$$A = \left(\underbrace{Q}_{\text{קבוצת מצבים סופית}}, \underbrace{\Sigma}_{\text{א"ב קלט וסופי}}, \underbrace{\delta}_{\text{פונקציית מעבר}}, \underbrace{q_0}_{\text{מצב התחלה}}, \underbrace{F}_{\text{תת קבוצה של מצבים מקבלים}} \right)$$

$$\delta: Q \times \Sigma \to Q$$

<u>חישוב</u>

1. מתחילים במצב q_0
2. בכל צעד נקרא אות מהקלט $w \in \Sigma^*$ ונשתמש ב-δ לקביעת המצב הבא.
3. נקבל את w אמ"מ בסיום הריצה על הקלט נהיה ב-$q \in F$.

<u>הבדלים ממכונת טורינג</u>

1. אין ראש שנע שמאלה וימינה על הקלט.
2. אין כתיבה.
3. אין מצבי עצירה.

נגדיר $L(A)=\{w|A\ accepts\ w\}$

אוטומט סופי לא דטרמיניסטי – *NFA*

$$A=\left(\underset{\text{קבוצת מצבים סופית}}{Q},\underset{\text{א"ב קלט וסופי}}{\Sigma},\underset{\text{פונקציית מעבר}}{\delta},\underset{\text{מצב התחלה}}{q_0},\underset{\text{תת קבוצה של מצבים מקבלים}}{F}\right)$$

$$\delta: Q\times \Sigma\cup\{\varepsilon\}\rightarrow p(Q)$$

חישוב מקבל

רצף $\tilde{w_i}\in\Sigma\cup\{\varepsilon\}$, $w=\tilde{w_1}\tilde{w_2}\dots\tilde{w_m}$ וקבוצת מצבים $q_0`,\dots,q_m`$.

1. $q_0`=q_0$
2. $q_{j+1}\in\delta\left(q_j,\tilde{w_{j+1}}\right)$
3. $q_m`\in F$

נגדיר שאוטומט N מקבל מילה w אם קיים חישוב מקבל עבור w ב-N.

<u>משפט</u>: מחלקת השפות המתקבלות ע"י NFA ו-DFA זהות - השפות הרגולריות.

הפיכת *NFA* ל-*DFA*:

אוטומט לא דטרמיניסטי: $N=(Q,\Sigma,\delta,q_0,F)$

אוטומט דטרמיניסטי השקול ל-N: $M=(Q`,\Sigma,\delta`,q`_0,F`)$.

- $Q`=P(Q)$: ב-M יהיה מצב לכל קבוצת מצבים ב-Q.
- $q`_0=\{q_0\}$
- $F`=\{q\in Q`|q\ includes\ q`\ from\ F\}$: מצב מקבל ב-$M$ הוא קבוצת מצבים (מצב יחיד ב-M) המכיל מצב מקבל מ-N.
- $\forall R\in Q`\ \forall a\in\Sigma.\ \ \delta`(R,a)=\{q\in Q|\exists r\in R: q\in\delta(r,a)\}$: R הינה קבוצת מצבים של Q ב-N וגם מצב ב-M. מ-$\delta`$ נקבל קבוצה של מצבים מ-N (או מצב מ-M) של כל המצבים (ב-N) שניתן להגיע מהם דרך a ממצב כלשהו השייך ל-R.
- כדי לטפל גם במעברי ε, לכל מצב R ב-M מגדירים $E(R)=\{q|q\ can\ be\ reached\ from\ R\ by\ traversing\ 0\ or\ more\ \varepsilon\ arrows\}$.
  עתה מחליפים את $\delta(r,a)$ ב-$E\big(\delta(r,a)\big)$. כלומר ההגדרה החדשה ל-$\delta`$ היא:
  $$\forall R\in Q`\ \forall a\in\Sigma.\ \ \delta`(R,a)=\big\{q\in Q\big|\exists r\in R: q\in E\big(\delta(r,a)\big)\big\}$$
  בנוסף נשנה את המצב ההתחלתי $q`_0=E(\{q_0\})$.

למת הניפוח (לשפות רגולריות)

L שפה רגולרית $\Leftarrow$ קיים קבוע p כך שלכל $w\in L$ ו-$|w|\geq p$ ניתן לפרק את w ל $w=xyz$ באופן הבא:

- $|xy|\leq p$
- $|y|>0$
- $\forall i\geq 0.xy^iz\in L$

סגירות שפות רגולריות לפעולות

- חיתוך
- איחוד
- שרשור
- משלים
- היפוך (reverse)
- *Kleene Star*
- $DropChar(L)=\{xy|x,y\in\Sigma^*,a\in\Sigma,xay\in L\}$

- $\max(L) = \{w \in L | \forall u \in \Sigma^*, wu \in L\}$
- $Rot(L) = \{yx | xy \in L\}$

בעיות כריעות בשפות רגולריות

- שייכות
- *Emptiness*
- *Fullness*
- *Subset*
- *Equivalence*

מינימליזציה של אוטומט

- ראשית מוחקים מצבים לא נחוצים.
- שנית מגדירים שפת המשך של מצב q, $L(q)$ = כל המילים שמתקבלות ממצב זה. נאחד מצבים עם שפות המשך שווה.

ביטוי רגולרי

לכל ביטוי רגולרי ניתן לבנות אוטומט סופי. גם ההפך נכון.

ביטוי רגולרי	משמעות	דוגמה
$+$	איחוד קבוצות	$a + b \equiv \{a\} \cup \{b\}$
$*$	*Kleene Star*	$a^* = \{\varepsilon, a, aa, aaa, \dots\}$
$\cdot$	שרשור	$(a+b)^* \equiv \{a, b, ab, aa, bb, ba \dots\}$

פעולת התרגום מאוטומט לביטוי רגולרי מתבצעת ע"י שימוש ב-$L(q, r, S)$ - שפת כל המילים w שמתחילות במצב q, מסתיימות במצב r ועוברות רק דרך המצבים שבקבוצה S.

סיכום שיטות הוכחה

<u>רגורליריות:</u>

- בניית *DFA*, *NFA*
- בניית ביטוי רגולרי
- שפה שמתקבלת משפות רגולריות ע"י פעולות משמרות רגולריות.

<u>אי רגולריות:</u>

- סתירה ללמת הניפוח
- קבלת שפה לא רגולרית מפעולות משמרות רגולריות

תובנות ממבחנים

- מעבר מ-DFA ל-NFA נעשה בזמן פולינומיאלי.

שפות חסרות הקשר

דקדוק חסר הקשר

$$G=\Big(\underbrace{V}_{\text{קבוצה סופית של משתנים}},\underbrace{\Sigma}_{\text{קבוצה סופית של טרמינלים}},\underbrace{R}_{\text{אוסף חוקים}},\underbrace{S}_{S\in V}\Big)$$
$$R: V\to (V\cup\Sigma)^*$$

<u>גזירה</u>: מתחילים במשתנה S ובכל שלב בוחרים משתנה כלשהו A וכלל גזירה $A\to\beta$ ומחליפים את A ב- β. מסיימים כשאין יותר משתנים.

<u>שפה חסרת הקשר</u>: $L(G)=\{w\mid$ ניתן ב-G לגזור את w מ-$S\}$.

למת הניפוח (לשפות חסרות הקשר)

תהי L שפה חסרת הקשר. אזי קיים קבוע $p>0$ כך שלכל מילה בשפה $w\in L, |w|\geq p$ מתקיימת חלוקה $w=uvxyz$ כך ש:

1. $|vy|\geq 0$
2. $|vxy|\leq p$
3. לכל $i\in\mathbb{N}$ מתקיים $uv^ixy^iz\in L$.

<u>הערה</u>: יש שפות לא חסרות הקשר שלא ניתן להוכיח כי הן אכן לא חסרות הקשר ע"י למת הניפוח.
כדי להראות שלמת הניפוח לא מקריסה ח"ה מראים שלכל מילה בשפה w (ארוכה מספיק) ניתן למצוא חלוקה שתקיים את למת הניפוח.

פעולות השומרות על סגירות של שפות ח"ה

- *Reverse*
- חיתוך עם שפה רגולרית
- איחוד
- שרשור
- הצבה חסרת הקשר – לוקחים שפה קיימת ומחליפים כל טרמינל בה לסימן התחלה של דקדוק ח"ה אחר.
- *Kleene Star*
- $pref(L)=\{w\mid \exists u. wu\in L\}$

דקדוק חסר הקשר ליניארי

זהו דקדוק שבו כלל גזירה הוא מהצורה $A\to x$ או $A\to xBy$ כאשר $x,y\in\Sigma^*, A,B\in V$.

דקדוק חסר הקשר הוא <u>ליניארי ימני</u> אם כל כלל גזירה הוא מהצורה $A\to a$ או $A\to aB$ או $S\to\varepsilon$ כאשר $a\in\Sigma, A,B\in V$.

<u>טענה</u>: משפחת השפות הנוצרות על ידי דקדוקים ליניאריים ימניים היא משפחת השפות הרגולריות.

צורת חומסקי לדקדוק חסר הקשר

בצורת חומסקי, כל כלל גזירה הוא מהצורה הבאה בלבד:

כלומר כל משתנה הופך לטרמינל או לשרשור שני משתנים.

<u>טענה</u>: כל דקדוק חסר הקשר ניתן להפוך לצורת חומסקי. זה נעשה באופן הבא:

- הוספת מצב התחלה חדש S_0
- נפטרים מכל כללי הגזירה מהצורה $A \to \varepsilon$
- נפטרים מכל כללי הגזירה של משתנה ההופך למשתנה אחר. למשל $A \to B$.
- מוסיפים כללי גזירה כך שהדקדוק החדש יצור אותה שפה כמו המקור.
- כלל גזירה היוצר מחרוזת הופך לרצף של כללי גזירה, למשל:
$$A \to u_1 u_2 \dots u_k \Longrightarrow A \to u_1 N_1,\ \ N_{i-1} \to u_i N_{i+1}$$

בעיות הכרעה בדקדוקים חסרי הקשר

- שייכות, $x \in L(G)$ – **כריעה**
- קבוצה ריקה, $L(G) = \emptyset$ - **כריעה**
- קבוצה מלאה, $L(G) = \Sigma^*$ - **לא כריעה**

שיטות הוכחה לח"ה

<u>שפה חסרת הקשר:</u>

- מראים דקדוק ח"ה לשפה
- השפה היא רגולרית
- השפה מתקבלת מהפעלת פעולות השומרות ח"ה על שפות ח"ה

<u>שפה אינה חסרת הקשר:</u>

- לא מתקיימת למת הניפוח
- אם מפעולות שומרות ח"ה ניתן לקבל שפה שהיא לא חסרת הקשר

הערה חשובה

לא קיים דקדוק חסר הקשר המייצר את כל החישובים של מכונה מסויימת. אך המשלים לכל המחרוזות שמייצגות חישוב הוא חסר הקשר.

תובנות ממבחנים

- ניתן לבדוק כי דקדוק כלשהו הוא סופי.
- ידוע כי קיים דקדוק חסר הקשר ל-Σ^*, אין צורך להראות בניה שלו במידה ורוצים להשתמש בו.

תאור כל מחלקות השפות לפי סיבוכיות