הרצאה 7

1 תזכורות מהרצאה 6

יחס שקילות ייקרא אינווריאנטי מימין אם לכל $z \in \Sigma^*$ מתקיים: $R(x,y) \Leftrightarrow R(xz,yz)$.

עבור שפה $L \subseteq \Sigma^*$, היחס R_L מוגדר: $R_L(x,y) \Leftrightarrow \forall z \in \Sigma^* : xz \in L \leftrightarrow yz \in L$.

משפט האפיון של R_L: לכל $L \subseteq \Sigma^*$ מתקיים:

R_L אינווריאנטי מימין, ומעדן את L. ואם יחס R אינווריאנטי מימין ומעדן את L, הוא מעדן את R_L.

משפט נרוד: שלושת התנאים הבאים שקולים:

- השפה L רגולרית,
- קיים יחס שקילות R שהוא אינווריאנטי מימין ומעדן את L, וגם $index(R) < \infty$,
- $index(R_L) < \infty$.

1.1 שימוש במשפט נרוד, כדי להראות ששפה אינה רגולרית

נגדיר קבוצת מילים אינסופית, ונראה שלכל שתי מילים בקבוצה ניתן למצוא סיפא מפרידה.

זה יראה שכל שתי מילים יושבות במחלקת שקילות נפרדת של R_L. ולכן צריכות להיות אינסוף מחלקות שקילות.

לפי משפט נרוד, נקבל שהשפה לא רגולרית.

האם זה תקף תמיד? האם בכל שפה שבה נוכל למצוא אינסוף זוגות מילים עם סיפא מפרידה, זה מראה שהשפה לא רגולרית?

נשקול את השפה: $L = \{w \in \Sigma^* : |w| \equiv 0(mod\ 2)\}$. מילים באורך זוגי.

קיימות רק שתי מחלקות שקילות. אבל לכל מילה, אפשר למצוא אינסוף מילים שלא שקולות לה.

מה ההבדל?

בדוגמה הראשונה הראנו באופן כללי על שתי מילים, וזה מוכיח ש**כל** שתי מילים בקבוצה יהיו במחלקת שקילות נפרדת.

בדוגמה השנייה קבענו את אחת המילים. לא הראנו שזה עובד לכל שתי מילים שנבחר.

2 מינימליזציה של אס"ד

כיצד נבנה אוטומט עם מספר מינימלי של מצבים?

ניעזר בטענה, שההוכחה שלה מסתמכת על כמה נקודות שראינו בהוכחה של משפט נרוד:

2.1 טענה:

אם A אס"ד המקבל את L, אזי $index(R_L) \le |Q_A|$ הוא מספר מצבי האוטומט A_L (האוטומט שראינו בהוכחת משפט נירוד).

הוכחה: לפי ההגדרה, $L = L(A)$ ולכן R_A מעדן את L. ממשפט האפיון של R_L מתקיים כי R_A מעדן גם את R_L. ולכן $index(R_L) \le index(R_A)$. מתקיים $|Q_{A_L}| = index(R_L) \le index(R_A) \le |Q_A|$, ובסה"כ: $|Q_{A_L}| \le |Q_A|$, כנדרש.

קיבלנו כי R_L הינה עם כמות מחלקות שקילות מינימלית, ו-A_L הוא האוטומט בעל כמות המצבים המינימלית.

2.2 אלגוריתם לבנייה

1. נמצא את כל מחלקות השקילות של השפה.
2. בניית A_L:
   2.1. לכל מחלקת שקילות, נקצה מצב באוטומט.
   2.2. נסדר את δ לפי המחלקות.

בצורה זו, נקבל אוטומט בעל כמות מצבים מינימלית.

הבעיה: אי אפשר לתכנת את זה. מחשב (עדיין) לא יודע לפענח מהן מחלקות השקילות.

לכן, נציג את האלגוריתם של *Hopcroft*, שאותו אפשר לתכנת: סדר הפעולות שלנו: נציג את האלגוריתם, אינטואיציה למה הוא נכון, ודוגמת הרצה.

2.3 אלגוריתם *Hopcroft*

<u>קלט</u>: אס"ד $A = (Q', \Sigma, q_0', \delta', F')$ כלשהו.

<u>פלט</u>: אס"ד לאותה שפה, עם כמות מינימלית של מצבים. (יש רק אחד כזה, עד כדי איזומורפיזם).

השלבים:

1. נמחק את המצבים שאינם ישיגים מ-q_0'. נסמן ב A את האוטומט המתקבל. (מציאת המצבים ע"י *BFS*).
2. נגדיר יחס: $R \leftarrow \phi$. זהו יחס סימטרי. (בינארי על Q. כלומר הוא מתייחס לזוגות – זוג מסויים מקיים או לא מקיים את היחס).
   2.1. נעדכן: $R \leftarrow \{\{p, q\} : q \in F \wedge p \notin F\}$. כל הזוגות שאחד מהם מקבל והשני לא.
   2.2. שלב S:
   2.2.1. נחשב $S \leftarrow \{\{p, q\} \notin R : \exists\sigma \in \Sigma : \{\delta(q, \sigma), \delta(p, \sigma)\} \in R\}$.
   נבדוק זוגות שעוד לא ב-R: (כלומר כרגע, שניהם מתקבלים או שניהם לא). נבדוק אם יש תו בא"ב שמפריד ביניהם.
   2.2.2. אם $S \neq \phi$, נעדכן את $R \leftarrow R \cup S$ ונחזור ל 2.2.1. כאשר S ריקה, אין זוגות חדשים, נעבור לשלב הבא:
3. אם $\bar{R}$ (המשלים) אינו יחס שקילות על Q, נחזיר *FAIL*. (זה לא אמור לקרות, אבל זה *failsafe* בשביל נכונות האלגוריתם).
4. נבנה DFA מצומצם $\tilde{A} = (\tilde{Q}, \Sigma, \widetilde{q_0}, \tilde{\delta}, \tilde{F})$ באופן הבא:

   $\tilde{Q}$ = קבוצת מחלקות השקילות של $\bar{R}$.

   $\widetilde{q_0}$ = מחלקת השקילות של q_0 ב $\bar{R}$. נסמן ב $[q_0]_{\bar{R}}$.

   $\tilde{F} = \{\tilde{q} : \tilde{q} \cap F \neq \phi\}$.

   $\forall\tilde{q} \in \tilde{Q}, \forall\sigma \in \Sigma : \tilde{\delta}(\tilde{q}, \sigma) = [\delta(q, \sigma)]_{\bar{R}}$

2.4 אינטואיציה לאלגוריתם

בהינתן A (ללא מצבים לא ישיגים), מתקיים:

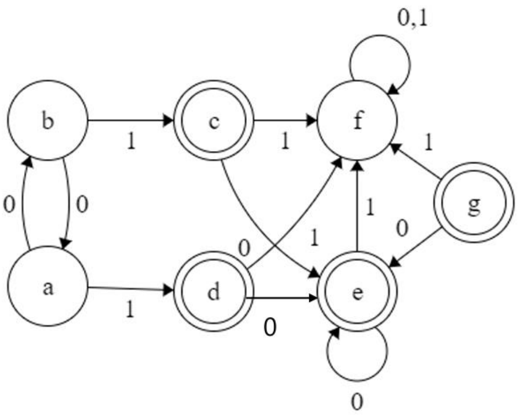
קבוצת מחלקות השקילות של R_A הן בדיוק $S = \{L_q : q \in R\}$.

R_A מעדן את R_L, ו-R_L מעדן את L.

האלגוריתם יקבל את המצבים p כך שה L_q-ים הם תתי קבוצות של אותה מחלקת שקילות ב-R_L יחד, ובין כל השאר היחס R יפריד.

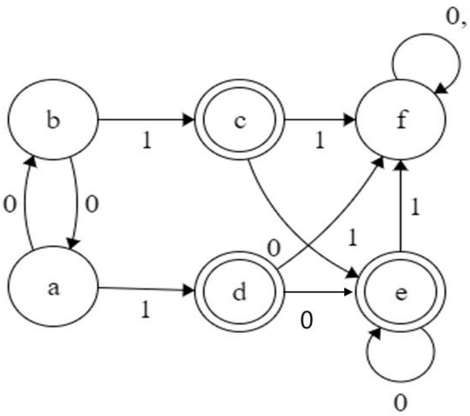
2.5 דוגמת הרצה 1

נתון האוטומט הבא: (שימו לב לחיצים שיוצאים מ- c, d – התחתון שיוצא מ-c זה 0, העליון שיוצא מ-d זה 1).



שלב 1: סילוק מצבים לא ישיגים.

כאן, המצב היחיד שלא ישיג הוא g. נוציא אותו ונקבל את האוטומט הבא:



שלב 2: נאתחל יחס $R = \phi$.

נכניס את כל הזוגות שאחד מהם ב-F והשני לא:

$$R = \{\{a,c\},\{b,c\},\{f,c\},\{a,d\},\{b,d\},\{f,d\},\{a,e\},\{b,e\},\{f,e\}\}$$

שלב 2.2.1: נחשב את S, קבוצת כל הזוגות שאינם ב-R, אך שעל ידי קריאת אות כן יהיו בו.

$\{a,f\}$ ייכנס ל-S כי האות 1 מפרידה ביניהם: $\delta(a,1) \in F, \delta(f,1) \notin F$.

כנ"ל $\{b,f\}$.

כלומר, קיבלנו ש

$$R = \{\{a,c\},\{b,c\},\{f,c\},\{a,d\},\{b,d\},\{f,d\},\{a,e\},\{b,e\},\{f,e\},\{a,f\},\{b,f\}\}$$

באיטרציה השנייה, S תהיה ריקה. האלגוריתם יעצור.

מה האוטומט המתקבל? איך מפענחים מ-R מהן מחלקות השקילות?

האלגוריתם אמר להסתכל על $\bar{R}$. נעשה את זה: איזה זוגות לא ב-R?

$$\{a,b\} \notin R, \qquad \{\{c,d\},\{d,e\},\{c,e\}\} \nsubseteq R$$

נתבונן היטב: כל המצבים נמצאים ביחס עם f, כלומר, R מפריד בין כל המצבים לבין f. ולכן f הוא מחלקת שקילות בפני עצמו. וגם אלו שציינו למעלה הם מחלקות שקילות.

טבלת עזר: נבנה טבלת "הצלבות" בין המצבים, כדי לראות מי לא ב-R:

טבלה התחלתית: (שימו לב שהיא סימטרית).

	a	b	c	d	e	f
a	X					
b	X	X				
c	X	X	X			
d	X	X	X	X		
e	X	X	X	X	X	
f	X	X	X	X	X	X

אחרי השלב הראשון:

	a	b	c	d	e	f
a	X		✓	✓	✓	✓
b	X	X	✓	✓	✓	✓
c	X	X	X			✓
d	X	X	X	X		✓
e	X	X	X	X	X	✓
f	X	X	X	X	X	X

ובסוף האלגוריתם:

	a	b	c	d	e	f
a	X		V	V	V	V
b	X	X	V	V	V	V
c	X	X	X			V
d	X	X	X	X		V
e	X	X	X	X	X	V
f	X	X	X	X	X	X

ולכן $\bar{R} = \{\{a,b\},\{c,d\},\{c,e\},\{d,e\},\{f\}\}$. ($f$ כי היא מופרדת מכולם – השורה ועמודה שלה מלאות).

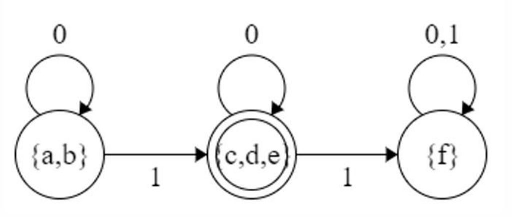
ונקבל את מחלקות השקילות הבאות: $\{a,b\},\{c,d,e\},\{f\}$.

המצב הראשון היה a ולכן $\{a,b\}$ הינה מחלקת השקילות של q_0 החדש.

להן הולכים משם? לפי האוטומט המקורי, עבור 0 אנחנו נשארים במקום. 1 מעביר ל-c או d, כלומר ל- $\{c,d,e\}$.

מכל המצבים משם, 0 משאיר אותנו בקבוצת המצבים הזאת. עבור 1 היה לנו $\tilde{\delta}(c,1) = \delta(d,1) = \delta(e,1) = f$.

מצב f היה ונשאר בור. קיבלנו את האוטומט הבא:



ועכשיו קל לראות מה השפה: $L = \{w \in \{0,1\}^* : \#_1(w) = 1\}$.

3 **הוכחת האלגוריתם** (רק טענה 0 הייתה השנה)

צריך להוכיח שהוא עוצר, וצריך להוכיח שהוא מחשב נכון.

בהקשר של ייצור אוטומט שקול, "מחשב נכון" הכוונה שבניית האוטומט נכונה – $\tilde{\delta}$ מוגדרת היטב, $\tilde{F}$ מקבל רק מילים רלוונטיות, האלגוריתם לא מחזיר *FAIL*, ולבסוף ששפת האוטומט שמתקבלת אכן זהה לשפת האוטומט המקורי.

נשתמש בכמה טענות עזר:

טענה 0: האלגוריתם תמיד עוצר.

הוכחת 0: האלגוריתם עוצר כאשר S ריקה. היות ויש $T = \binom{|Q|}{2}$ זוגות אפשריים, האלגוריתם יעצור לאחר לכל היותר $T+1$ איטרציות. (אפשר לשפר את זה, אבל לצורך עצירת האלגוריתם זה חסם מספיק טוב).

שאר ההוכחה לא הייתה בחומר השנה!

טענה 1: אינווריאנטה:

לכל $i \in \mathbb{N}$ מתקיים: בהינתן $q \neq q \in Q$ כלשהו, המחרוזת הקצרה ביותר z כך שקיימים $x \in L_p, y \in L_q$ כך ש $xz \in L(A) \wedge yz \notin L(A)$ היא באורך i אמ"מ $\{p,q\}$ מצטרף אל R בסוף האיטרציה ה i.

מסקנה 1: (מטענה 1): האלגוריתם אינו מחזיר *FAIL*. כלומר, $\bar{R}$ שמתקבל בסוף האלגוריתם הוא אכן יחס שקילות.

הוכחה: $(p,q) \in \bar{R} \Leftrightarrow L_p, L_q$ הן תתי קבוצות (לא ריקות) של אותה מחלקת שקילות של $R_L(A)$.

מסקנה 2: $index(R_L(A)) = index(\bar{R})$. אם נצייר את הוכחת המסקנה הקודמת נקבל בקלות את המסקנה הזו. $\bar{R}$ הפריד בין שפות מצבים בדיוק כמו ש $R_L(A)$ אמור להפריד.

טענה 2: אם $\tilde{q} \in \tilde{F}$, אזי $\tilde{q} \subseteq F$.

הוכחה: מההגדרה של $\tilde{F}$ נובע כי אם $\tilde{q} \in \tilde{F}$ אזי קיים $q \in \tilde{q}$ כך ש $q \in F$. נב"ש שקיים $p \in \tilde{q}$ אבל $p \notin F$. מכאן, p,q ניתנים להפרדה ע"י ϵ, כלומר לפי טענה 1, הם צריכים להשתייך למחלקות שקילות שונות של $\bar{R}$ (ולהיכנס ל R כבר באיטרציה 0). בסתירה לכך ש $p,q \in \tilde{q}$, באותה מחלקת שקילות של $\bar{R}$.

טענה 3: $\tilde{\delta}$ מוגדרת היטב. (מוגדרת באותה צורה לכל מועמד $q \in \tilde{q}$ שאנו בוחרים).

הוכחה: הגדרנו $\tilde{\delta}(\tilde{q},\sigma)=[\delta(q,\sigma)]_{\bar{R}}$, כך ש $q\in\tilde{q}$. נראה שאם $p,q\in\tilde{q}$ אזי $[\delta(p,\sigma)]_{\bar{R}}=[\delta(q,\sigma)]_{\bar{R}}$.

נב"ש שזה לא נכון – נסמן $\delta(p,\sigma)=p'$, $\delta(q,\sigma)=q'$. ונניח ש $[q']_{\bar{R}}\neq[p']_{\bar{R}}$. ההנחה הזו שקולה לכך ש $(p',q')\notin\bar{R}\Leftrightarrow(p',q')\in R$. לכן, מטענה 1, קיים z המפריד בין p',q'. אבל אז σz מפריד בין p לבין q בסתירה לכך ש $(p,q)\in\bar{R}$.

טענה 4: $L(A)=L(\tilde{A})$. לשם כך נוכיח את טענת עזר 5:

טענה 5: $\forall w\in\Sigma^*:\hat{\tilde{\delta}}(\widetilde{q_0},w)=\left[\hat{\delta}(q_0,w)\right]_{\bar{R}}$.

הוכחה באינדוקציה על $|w|$:

בסיס: נבחר $|w|=0$, כלומר $w=\epsilon$. הטענה מתקיימת לפי הגדרת $\widetilde{q_0}$.

צעד: נניח שהטענה נכונה עבור $|w|<n$ כלשהו, ונוכיח עבור $|w|=n$:

נסמן $w=x\sigma$ כך ש $|x|=n-1$, לכן מתקיים:

$$\hat{\tilde{\delta}}(\widetilde{q_0},x\sigma)=\tilde{\delta}\left(\hat{\tilde{\delta}}(\widetilde{q_0},x),\sigma\right)=^{\text{א}}\tilde{\delta}\left(\left[\hat{\tilde{\delta}}(q_0,x)\right]_{\bar{R}},\sigma\right)=^{\text{ב}}[\delta(p,\sigma)]_{\bar{R}}$$

א – מהגדרת $\hat{\delta}$, והנחת האינדוקציה. ב – מהגדרת $\tilde{\delta}$

(כאשר $p\in\left[\hat{\delta}(q_0,x)\right]_{\bar{R}}$)

הוכחת טענה 4:

$$\begin{aligned}&w\in L(\tilde{A})\Leftrightarrow\\&\hat{\tilde{\delta}}(\widetilde{q_0},w)\in\tilde{F}\Leftrightarrow\\&\forall q\in\hat{\tilde{\delta}}(\widetilde{q_0},w):q\in F\Leftrightarrow\\&\forall q\in\left[\hat{\delta}(q_0,w)\right]_{\bar{R}}:q\in F\Leftrightarrow\\&\hat{\delta}(q_0,w)\in F\Leftrightarrow w\in L(A)\end{aligned}$$

לפי הסדר: מהגדרת $\tilde{F}$, מטענה 2, מטענה 5, מטענה 2, הגדרת F.

כעת נוכיח את טענה 1: נזכיר אותה:

אינווריאנטה: לכל $i\in\mathbb{N}$ מתקיים: בהינתן $q\neq q\in Q$ כלשהו, המחרוזת הקצרה ביותר z כך שקיימים $x\in L_p,y\in L_q$ כך ש $xz\in L(A)\wedge yz\notin L(A)$ היא באורך i אמ"מ $\{p,q\}$ מצטרף אל R בסוף האיטרציה ה i.

הוכחה באינדוקציה על i:

בסיס: (שלב 1 של האלגוריתם, הגדרת R)

צעד: נניח שהטענה נכונה עבור i כלשהו, ונוכיח עבור $i+1$.

נניח ש z שאורכה $i+1$ היא המחרוזת הקצרה ביותר המפרידה בין זוג $p,q\in Q$ כלשהו. נסמן $z=x\sigma$ ונתבונן ב $q'=\delta(q,\sigma),p'=\delta(p,\sigma)$. לפי ההנחה, w הינה המחרוזת הקצרה ביותר המפרידה בין p',q'. מהנ"א, נקבל כי p',q' הופרדו באיטרציה ה-i של האלגוריתם.

לכן, אם p,q לא הופרדו עוד קודם לכן, הם יופרדו ע"י בדיקת התו σ באיטרציה ה-i. ואכן, הם לא הופרדו קודם לכן שכן אחרת הייתה מחרוזת z' המפרידה ביניהם (ואין כזו, בגלל הנ"א).

זו הוכחה בכיוון אחד. הכיוון השני בדרך כלל במטלה.

4 מבוא לדקדוקים

עד כה הכרנו שתי דרכים להגדיר שפות: $L = \{\dots\}$, או אוטומט שפת – $L(A)$. נלמד דרך שלישית – דקדוק (שכתוב).

מהו דקדוק? קבוצת משתנים V, קבוצת טרמינלים T, וקבוצה של כללי שכתוב P. כמו כן, יש משתנה התחלתי S. (כל הקבוצות סופיות).

לדוגמה: $S \to BA, B \to a, A \to bab|cab$. אלו הכללים שרשומים ב P. בנוסף, ניתן לראות ש $T = \{a,b,c\}, V = \{S,A,B\}$.

4.1 דקדוק כמגדיר שפה

קבוצת המילים המתקבלות מהדקדוק מוגדרות כאוסף של כל המילים שניתן לקבל כאשר מתחילים במחרוזת S ומבצעים סדרת צעדים בהם משכתבים תת מחרוזת כלשהי, באמצעות אחד מכללי השכתוב. עוצרים כאשר מתקבלת מחרוזת מעל T^*. בדוגמה לעיל נקבל:

$$S \to BA \to aA \to \begin{cases} abab \\ acab \end{cases}$$

כאשר החץ הראשון מכלל הגזירה $S \to BA$, השני מהכלל $B \to a$, השלישי מהכלל של A.

כדי להוכיח נכונות בנייה של דקדוק (כלומר להוכיח כי $L = L(G)$) נצטרך להראות הכלה דו כיוונית (בד"כ באינדוקציה על אורך הגזירה או אורך המילה).

פורמלית:

דקדוק הוא רביעייה $G = (T,V,P,S)$ כאשר:

V קבוצה סופית ולא ריקה של משתנים.

T קבוצה סופית ולא ריקה של טרמינלים (אותיות), הזרה ל-V.

P קבוצה של כללי שכתוב (או כללי גזירה) כך שכל כלל ב-P הוא מהצורה $\alpha \to \beta$ כאשר:
$\alpha \in (V \cup T)^* \cdot V \cdot (V \cup T)^*$, $\beta \in (V \cup T)^*$. (ב-α יש לפחות משתנה אחד, ועוד משתנים וטרמינלים. ב-β לא חייב להיות – זה יכול להיות כל ערבוב של משתנים וטרמינלים).

לדוגמה: הרשום להלן הינו דקדוק - $V = \{S\},\ T = \{a,b\},\ P = \{S \to aSb|ab\}$.

4.2 גזירה

יהי $G = (T,V,P,S)$ דקדוק, ויהיו $\gamma_1, \gamma_2 \in (V \cup T)^*$. נאמר כי γ_2 נגזר ישירות מ γ_1 ונסמן:

$\gamma_1 \Rightarrow_G \gamma_2$ (γ_1 גוזר ישירות בדקדוק G את γ_2),

אם ורק אם:

קיימים $\gamma, \chi, \alpha, \beta \in (V \cup T)^*$ כך ש: $\alpha \to \beta \in P$ וגם $\gamma_1 = \gamma\alpha\chi$, $\gamma_2 = \gamma\beta\chi$.

במלים אחרות: אם ניתן להגיע מ γ_1 אל γ_2 ע"י הפעלה בודדת של כלל שכתוב כלשהו מ P.

לדוגמה, בדקדוק לעיל: $S \Rightarrow ab$.

4.3 עוד קצת סימונים

נסמן $\alpha \Rightarrow_G^* \beta$ אם ניתן להגיע מ-α אל β במספר כלשהו של צעדים (כולל 0 צעדים). בפרט, $\alpha \Rightarrow_G^* \alpha$ לכל α ע"י הפעלת 0 כללים.

נסמן $\alpha \Rightarrow_G^i \beta$ אם ניתן להגיע מ-α אל β ב-i שלבים. לדוגמה, $\alpha \Rightarrow \beta \Leftrightarrow \alpha \Rightarrow^1 \beta$. בד"כ נשמיט את ה G כאשר ברור על איזה דקדוק מדובר.

4.4 שפה של דקדוק

הגדרה: בהינתן דקדוק G, השפה של $L(G)$ מוגדרת כך:

$$L(G) = \{x \in T^* : S \Rightarrow^* x\}$$

כלומר, קבוצת המילים מעל T^*, שאפשר להגיע אליהם מהמשתנה ההתחלתי על ידי כללי הגזירה.

לדוגמה: נתון הדקדוק הבא: $V = \{S\}, T = \{a, b\}, P = \{S \to aSb|\epsilon\}$.

מהי שפת הדקדוק? נעשה כמה צעדים כדי לקבל תחושה:

$$S \to aSb \to aaSbb \to aaaSbbb \to aaaaSbbbb \to a^nSb^n \to a^nb^n$$

זה דקדוק עבור השפה $L = \{a^nb^n : n \geq 0\}$. כי בכל שלב אפשר להוסיף רק a בהתחלה ו-b בסוף.

רואים שלדקדוק יש כוח שלא היה לשפות רגולריות. בעזרת דקדוק די פשוט קיבלנו שפה לא רגולרית.

4.5 דוגמה לבניית דקדוק עבור שפה

נבנה דקדוק לשפת המספרים העשרוניים.

הטרמינלים יהיו $T = \{0,1, \dots ,9\} \cup \{,\}$.

נייצר את המשתנים:

צריך משתנה עזר שמשכתב את $\{0,1, \dots ,9\}$: נקרא לו N_1. הכלל הוא: $N_1 \to 0|1| \dots |9$.

נרשום משתנה עבור קטגוריית מספר דו ספרתי: N_2. ונכניס ל-P את כלל הגזירה הבא: $N_2 \to N_1N_1$.

כנ"ל לגבי תלת ספרתי: $N_3 \to N_1N_1N_1$.

עבור מספרים של יותר מ-3 ספרות, צריך דרך להכניס פסיק: $S \to S, N_3$.

אם יש לנו S, אפשר להוסיף מימין מספר תלת ספרתי אחרי פסיק. ובגלל שה-S תמיד משמאל, המבנה של שלוש ספרות בין כל פסיק יישאר. לדוגמה:

$$S \to S, N_3 \to S, N_3, N_3 \to S, 123{,}456$$

כאשר המעבר האחרון נובע מכללי הגזירה שקבענו בהתחלה. ונרצה גם דרך להפוך את ה-S למספר. הוא יכול להיות מספר חד, דו, או תלת ספרתי: $S \to S, N_3|N_1|N_2|N_3$. ועכשיו נוכל לסיים את השכתוב:

$$S, 123{,}456 \to 78{,}123{,}456$$