אוטומטים 2 - הרצאה 3

סיכום: אורן דנון   מרצה: ד"ר פסקין צ'רניאבסקי ענת

20 במרץ 2018

1 מבוא: בניית אוטומט מינימלי

בשיעור הקודם התחלנו לתאר אלגוריתם למינימליזציה של DFA (אלגוריתם שאפשר לתכנת). האלגוריתם כמו שתיארנו עד כה מחלק את קבוצת המצבים ב Q לקבוצות $S_1,\dots S_k$. יתקיים (לא נוכיח), שמכל מצב ב S_i, הקבוצה שמגיעים אליהן ע"י קריאת אותיות σ כלשהן בהתאמה אינו תלוי במצב המסוים שבחרנו מהקבוצה. נגדיר את δ באוטומט המצומצם בין המצבים ב A הן הקבוצות $S_1,S_2,\dots,S_k$ ונגדיר $\delta(S_i,\sigma)=S_j$, כאשר S_j שייך לקבוצה של $\delta(q_i,\sigma)$ עבור $q_i\in S_i$ שנבחר שרירותית. מסתבר גם שהקבוצות $S_1,S_2,...,S_k$ מקיימות שכל המצבים בתוך כל קבוצה הם או כולם ב F או כולם ב $Q\setminus F$.
נגדיר את F' להיות אוסף הקבוצות המכילות מצב מקבל.
נציין כי הוכחת הנכונות של האלגוריתם מסתמכת על טענות ומושגים שראינו בהוכחת משפט נרוד, נדלג עליה מקוצר זמן.
דוגמה:

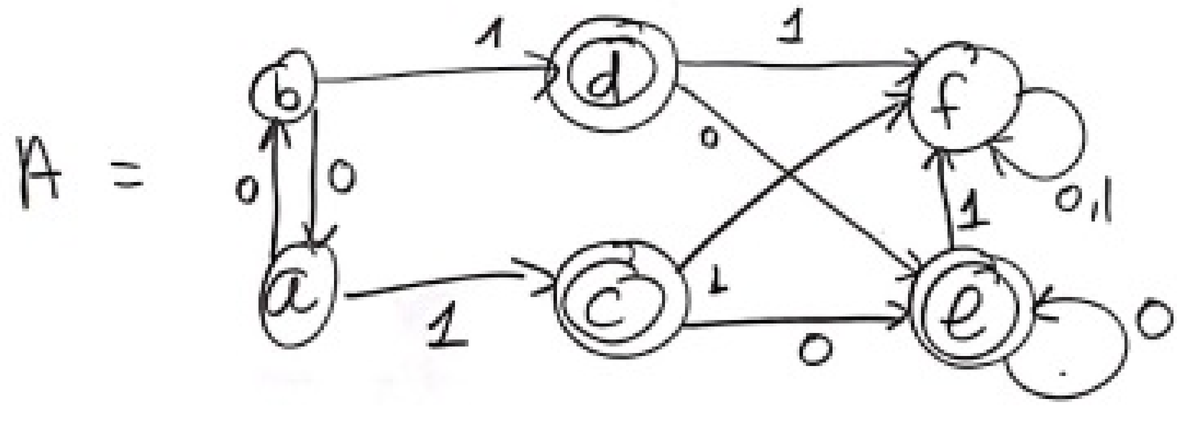

Figure 1: Automata

- אתחול נגדיר קבוצות: $F = S_1^0 = \{e,c,d\}$, $S_2^0 = Q\setminus F = \{a,b,f\}$ ונשים לב שהחלוקה מגדירה יחס שקילות על מצבים עם מחלקות שקילות. כל המצבים שאינם ניתנים להפרדה ע"י z באורך 0

$$\forall_{x\in L(q),y\in L(p)},\ \ xz\in L \Leftrightarrow yz\in L,\ \ \forall |Z|\le 0$$

- אטרציה 1: ננסה לפצל כל קבוצה לתתי קבוצות כך שמצבים בתתי קבוצות שונות נתנים להפרדה ע"י אות אחת (במובן שהאות מובילה אותם לקבוצות שונות בחלוקה הקודמת)
הערה: כמו באיטרציה ה־0, אלא שבאיטרציה ה־ i מדובר ב־ z כך ש $|z| \le i$.
בתוך S_1 אין מה להפריד. בתוך S_2 לא ניתן להפריד את a,b, אבל f מופרד משניהם ע"י $\sigma = 1$. כי $\delta(f,\sigma)\in S_2$ אבל $\delta(a,1),\ \delta(b,1)\in S_1$ קיבלנו חלוקה:

$$S_1^1 = \{e,c,d\},\ S_2^1 = \{a,b\},\ S_3^1 = \{f\}$$

- איטרציה 2: לא נוספו הפרדות חדשות, ולכן האלגוריתם מסתיים. שלב שתיים נבנה את האוטומט:

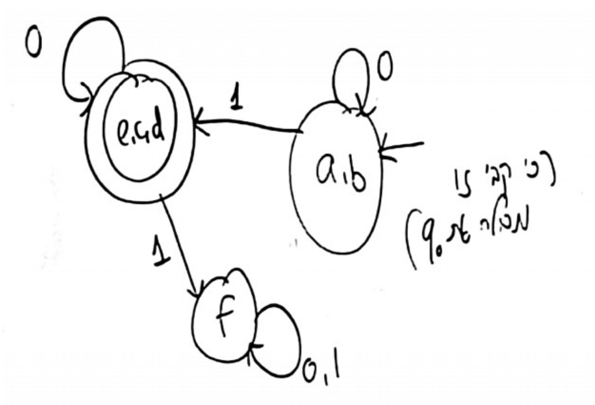

Figure 2: Min Automata

2 דקדוקים (נושא חדש)

בסדרת הקורסים על מודליים חישוביים אנחנו עוסקים בשפות פורמליות. כאשר שפה פורמלית היא פשוט תת־קבוצה של Σ^* עבור Σ א"ב סופי כלשהו. אנחנו מתעניניים גם בשאלה האם ניתן לבדוק שייכות לשפה L באמצעות מודל חישובי כלשהו. במקרה הכי כללי, בקורס חישוביות, נתעניין האם אפשר לזהות את השפה באמצעות תכנת מחשב ($java$ עם זכרון לא חסום מראש).ישנן כל מיני דרכים להגדיר שפה. לדוגמה,

כבר בבית ספר הגדרנו שפות בשפה "חצי חופשית" באמצעות תורת הקבוצות. לדוגמה: $L = \{x \in \{0,1\}^* : \ x \ is \ a \ prime\}$ ־ זו שפה שאפשר לכתוב לה תכנית מחשב. אבל למשל

$$L = \Big\{ x \text{ תוכנת מחשב עבור מחשב עם זיכרון לא חסום } \text{ עוצרת(לא נתקעת) על כל קלט אפשרי } x \mid \Big\}$$

לשפה זו לא קיימת תוכנת מחשב (תראו בקורס בחישוביות)

דרך נוספת שראינו להגדיר שפות היא באמצעות "תוכנה". בקורס אוטומטים 1 הגדרנו שפה של אוטומט באמצעות הגדרת ה־ DFA המקבל אותה.
בקורס אוטומטים 1 ראינו גם דרך להגדיר שפה באמצעות תהליך רקורסיבי סינטקטי: ביטויים רגולים. לדוגמה $(0+1)^*11(0+1)^*$,הוא ביטוי רגולי המגדיר את השפה

$$L = \{x \in \{0,1\}^* \,|x \text{ contains } 11\}$$

בקורס הזה נראה הגדרה סינטקטית קצת דומה לביטויים רגולריים, של שפות. והגדרה זו תהיה באמצעות דקדוק שכתוב. דקדוק שכתוב יגדיר תהליך רקורסיבי ליצירת מילים. והגדרה של הדקדוק היא קבוצת המילים שאפשר ליצור באמצעות התהליך
דוגמה: נרצה לבנות דקדוק עבור מספרים עשורניים עם פסיקים המפרידים בין כל 3 ספרות (נרשה אפסים מובילים) לדוגמה:

- $1,986$ חוקי
- $22,349,150$ חוקי
- 01 חוקי
- $111,$ לא חוקי
- $1,12,123$ לא חוקי

נבנה את הדקדוק בצורה רקורסיבית נגדיר משתנה N_1 שתפקידו "לייצר" ספרה

1. $N_1 \to 0|1|2|...|9$ (עשר כללי שכתוב המאפשרים להחליף את N_1 בספרה).
2. נגדיר משתנה N_2 שתפקידו לייצר זוג ספרות $N_2 \to N_1N_1$
3. נגדיר משתנה N_3 שתפקידו לייצר מספריים תלת ספרתיים. $N_3 \to N_1N_2N_3$ ניתן לגזור את המספר 701 (חוקי) באמצעות הסדרה:

$$N_3 \overset{\text{החלפנו את } N_3 \text{ בצד ימין של הכלל השלישי}}{\Longrightarrow} N_1N_1N_1 \overset{\text{שכתבנו את } N_1 \text{ האמצעי לאפס לפי הכלל הראשון}}{\Longrightarrow} N_10N_1$$

$$\Rightarrow N_107 \Rightarrow 107$$

יש כאן $3! = 6$ סדרות גזירה אפשריות, ראינו אחת מהן למילה 107 מ ־ N_3 107 . הסדרות זהות עד כדי סדר פיתוח המשתנים. אפשר "לסכם" את כולן בעץ הגזירה הבא:

Figure 3: Tree

4. נגדיר קטגוריה נוספת של N_4 של מספרים בן 1 או 2 או 3 ספרות :

$$N_4 \to N_1|N_2|N_3$$

5. נגדיר משתנה נוסף L שתפקידו לגזור רצף של שלשות המופרדות בפסיקים: $L \to N_3|N_3, L$. לדוגמה, כדי לגזור מספר באורך 21 נציע סדרת גזירה שמתחילה כך:

$$L \overset{L\to N_3,L}{\Longrightarrow} N_3, L \Rightarrow N_3, N_3, L \overset{L\to N_3}{\Longrightarrow} N_3, N_3, N_3, N_3 \Rightarrow ...$$

6. נגדיר משתנה שמגדיר מספר: $N_3 \to N_3|N_3, L$. נקבע את N להיות המשתנה ההתחלתי בדקדוק.

דוגמה: כדי לגזור $12,345,186$ נשתמש בעץ הגזירה הבא:

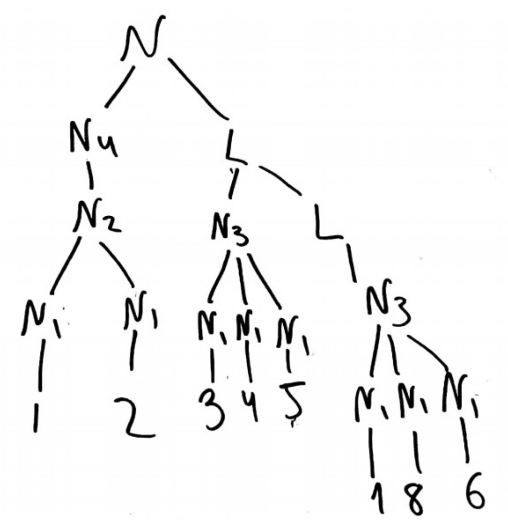

Figure 4: Tree for 12,345,186

דוגמה: נבנה דקדוק (פשוט) עבור שפה לא רגולרית ניקח את

$$L=\left\{a^nb^n|\ n\in\mathbb{N}^+\right\}$$

נגדיר דקדוק $G:\ S\to aSb|ab$ נוכיח שהדקדוק אכן גוזר את השפה שהתכוונו (כזכור, בצורה לא פורמלית הגדרנו את $L(G)$ להיות קבוצת המילים שניתן לגזור בדקדוק) נוכיח שני כיוונים:
$L\subseteq L(G)\Leftarrow$ כלומר נראה שכל $\forall_{n\geq 1}a^nb^n\in L(G)$. נוכיח באינדוקציה על n.
בסיס: עבור $n=1$ אכן $S\Rightarrow ab$ היא סדרת גזירה חוקית עבור ab.
צעד: נניח שהטענה נכונה עבור n ונראה עבור $n+1$ ננסה לבנות סדרת גזירה:

$$S\Rightarrow aSb\overset{I.H.\Rightarrow S\to a^nb^n}{\Rightarrow^*}aa^nb^nb=a^{n+1}b^{n+1}$$

$L(G)\subseteq L\Rightarrow$ כדי שהאינדוקציה תעבוד נוכיח טענה חזקה יותר:
טענה: אם אלפא נגזרת מ־S כלומר $S\Rightarrow^*\alpha$ כאשר אלפא רצף של משתנים ואותיות "רגילות". אזי אלפא מאחת מהצורות הבאות:

- $a^nb^n, n\geq 1$
- $a^nSb^n,\ \ n\geq 1$

נשים לב שהטענה $L(G)\subseteq L$ נובעת מיידית מהטענה החזקה. מדוע? כי לפי הטענה, כל מילה טרמינלית (= מורכבת רק מאותיות ולא ממשתני דקדוק) שניתן לגזור מ ־ S היא מהצורה $a^nb^n,\ \ n\geq 1$ ולכן שייכת ל L נותר להוכיח את הטענה: נעשה זאת באמצעות אינדוקציה על אורך סדרת הגזירה:
בסיס: אורך גזירה אחד לפי מבנה הדקדוק ניתן לגזור רק $S\Rightarrow ab$, ו־a,b אכן ב־L.
צעד: נניח עבור אורך גזירה n ונוכיח עבור $n+1$. תהי α מילה שקיימת עבורה סדרת גזירה באורך $n+1$ (המילה לאו דווקא טרמינלית). כלומר מתקיים:

$$S\Rightarrow^n\beta\Rightarrow\alpha$$

שימו לב שעבור β קיימת סדרת גזירה ב n צעדים, מהנחת האינדוקציה. $\beta=a^mSb^m,\ \ m\geq 1$, שימו לב ש β לא מהצורה a^mb^m, כי זו מילה שלא ניתן להמשיך לגזור (ואנחנו גזרנו מ β את α). ממבנה הדקדוק ישנם שני אפשרויות ל־ α או $\alpha=a^{m+1}b^{m+1}$ אם גזרנו $S\to ab$ או $\alpha=a^{m+1}Sb^{m+1}$ אם גזרנו מ ־ S את aSb בשני המקרים α היא מהצורה כנדרש.
$\square$