

אוטומטים – תרגול

DFA

הגדרה פורמלית:

DFA זה חמישייה $\langle Q, q_0, \Sigma, \delta, F \rangle$ כך ש:

1. Q - קבוצה סופית של מצבים.
2. Σ - קבוצה סופית של סימני א"ב (א"ב – הגדרה – קבוצה לא ריקה, ספית של סימנים)
3. $q_0 \in Q$ - מצב התחלתי.
4. $F \subseteq Q$ - קבוצת מצבים סופיים (מקבלים).
5. $\delta: Q \times \Sigma \rightarrow Q$ - פונקצית המעבר.

שפה:

Language L is the subset of Σ^*

$$\Sigma^0 = \epsilon$$

$$\Sigma^1 = \Sigma$$

$$\Sigma^2 = \Sigma \times \Sigma$$

...

$$\Sigma^k = \Sigma^{k-1} \times \Sigma^1$$

$$\Sigma^* = \bigcup_{i=0}^{\infty} \Sigma^i$$

שפה ש – DFA מקבל: אוסף כל המסלולים ממצב התחלתי לאחד מהמצבים הסופיים.

דוגמאות:

1. נתון $\Sigma = \{0,1\}$ בנה אוטומט עבור Σ^*
2. בנה אוטומט מעל א"ב בינארי המקבל את כל המחרוזות המכילות את הרצף 01.
3. בנה אוטומט מעל א"ב בינארי המקבל את כל המחרוזות שלא מכילות את הרצף 01.
4. בנה אוטומט מעל א"ב בינארי המקבל את שפת כל המילים המכילות מספר זוגי של אפסים.
5. בנה אוטומט המקבל את כל המילים הבינאריות שבהם הספרה לפני האחרונה היא אפס.
6. בנה אוטומט עבור השפה: $L = \{w \mid w \text{ is binary number divided by } 3\}$
7. בנה אוטומט מעל א"ב בינארי המקבל את כל המילים שערכן מתחלק ב-6.

שפה רגולארית

שפה L נקראת **רגולארית** אם קיים DFA המזהה אותה.

כל שפה סופית היא רגולרית.

יהיו L ו- W שפות רגולאריות אזי LFW , $L \cap W$, \bar{L} שפות רגולאריות.

NFA

הגדרה פורמאלית:

NFA זה חמישייה $\langle Q, q_0, \Sigma, \delta, F \rangle$ כך ש:

1. Q - קבוצה סופית של מצבים.
2. Σ - קבוצה סופית של סימני א"ב (א"ב - הגדרה - קבוצה לא ריקה, סופית של סימנים)
3. $q_0 \in Q$ - מצב התחלתי.
4. $F \subseteq Q$ - קבוצת מצבים סופיים (מקבלים).
5. $\delta: Q \times \Sigma \rightarrow P(Q)$ - פונקצית המעבר.

דוגמאות:

1. צייר NFA המזהה את אוסף כל המילים הבינאריות שערכן מתחלק ב-2 או ב-3.
2. בנה NFA המקבל את כל המילים הבינאריות שבהם ספרה לפני אחרונה היא 0.

משפטים:

DFA הוא מקרה פרטי של NFA

לכל NFA ניתן לבנות DFA שמזהה בדיוק את אותה השפה.

עד עכשיו ראינו שפות רגולאריות.
האם קיימות שפות לא רגולאריות?
נסתכל בשפה:

$$L = \{0^n 1^n\}$$