CS 205

Lecture # 5

13/01/21

**Prop :** Let $L_n = \{ w \in \{0,1\}^* \mid n^{th}$ character from end of $w$ is $1 \}$.
Then $L_n$ is regular and <u>any</u> DFA recognizing (i.e. accepting) $L_n$ must have at least $2^n$ states.

**Proof :** (Already shown that $L_n$ is regular.)

By contradiction. Assume that there is a DFA $M$ with less than $2^n$ states that accepts $L_n$.
The no. of strings of length $n$ is $2^n$. Therefore, there must be strings $w_0, w_1$ where $w_0 \neq w_1$, of length $n$ s.t. $\delta(q_0, w_0) = \delta(q_0, w_1)$.
Let $i$ be the first position where $w_0$ and $w_1$ differ.

Wlog, $w_0$ has $0$ in the $i^{th}$ position and $w_1$ has $1$.

$$w_0 0^{i-1} = \cdots \boxed{0} \overbrace{\cdots}^{n-i} \Big| \overbrace{\cdots}^{0^{i-1}} \notin L_n$$

$$w_1 0^{i-1} = \cdots \boxed{1} \cdots \Big| \cdots \in L_n$$

$i^{th}$ posn.

same state for both strings

We have $w_0 0^{i-1} \notin L_n$ and $w_1 0^{i-1} \in L_n$ and $\delta(q_0, w_0) = \delta(q_0, w_1)$.

$$\boxed{\underset{\sim}{\text{Fact}} \quad \delta(q, uv) = \delta(\delta(q, u), v)}$$ By induction on $|v|$

$$\begin{aligned} \delta(q_0, w_0 0^{i-1}) &= \delta(\underset{\sim}{\delta(q_0, w_0)}, 0^{i-1}) \\ &= \delta(\delta(q_0, w_1), 0^{i-1}) \\ &= \delta(q_0, w_1 0^{i-1}) \end{aligned}$$

Contradiction, since LHS $\notin F$ and RHS $\in F$.

## <u>Nondeterministic Finite Automata</u> (NFA)

Singular : automaton
plural : automata

Nondeterminism : given a state and an input symbol there are <u>several</u> possibilities for the next state.

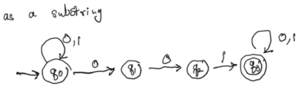
<u>Intuition</u> (1) Automaton "forks" or "branches" into different runs on the same input string

run – the sequence of states visited by the m/c in processing the input string

(2) Automaton "guesses" the right next state which will lead to the input string being accepted.

<u>Ex</u> $L$ = set of all strings containing 001 as a substring

NFA




$\boxed{w = 1001} \in L$

$r_1 = q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_0 \xrightarrow{0} q_1 \xrightarrow{1} \times$

$\hat{\delta}(q_0, 0) = \{q_0, q_1\}$

$\hat{\delta}(q_0, 001) = \{q_0, q_3\}$

<u>Formal Defn</u>

An NFA $M = (Q, \Sigma, \delta, q_0, F)$ has

$Q$ : finite set of states

$\Sigma$ : alphabet

$\delta : Q \times \Sigma \longrightarrow 2^Q$ (powerset of $Q$)

$F \subseteq Q$ : set of final or accepting states

We extend $\delta$ to $\hat{\delta} : Q \times \Sigma^* \to 2^Q$ as follows:

$\hat{\delta}(q, \epsilon) = \{q\}$ ← set of states

$$\hat{\delta}(q, ua) = \bigcup_{r \in \hat{\delta}(q, u)} \delta(r, a) \qquad u \in \Sigma^*,\ a \in \Sigma$$

$$= \{ p \mid p \in \delta(r, a) \text{ for some } r \in \hat{\delta}(q, u) \}$$

---

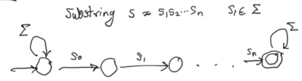
NFA $M$ <u>accepts</u> $w$ if $\hat{\delta}(q_0, w) \cap F \neq \emptyset$.

$$L(M) = \{ w \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset \}$$

language accepted / recognized by NFA $M$.

---

<u>Pattern Recognition</u> NFA recognizing all strings with substring $s = s_1 s_2 \cdots s_n$, $s_i \in \Sigma$




<u>KMP algorithm</u> / Boyer-Moore algorithm

---

Quiz on Monday : 18/01
30 + 5 mins
9:15 – 9:50
MS Forms

Everything so far is included.

---