CPSC 421: Introduction to Theory of Computing	**Winter Term 1 2018-19**

Lecture 4: September 12

Lecturer: Nicholas Harvey *Scribes: Kaitian Xie*

Language accepted by DFAs = Language accepted by NFAs = Regular Languages

Important properties of Regular Languages i.e. closure properties. These are much easier to prove using NFAs.

Theorem 4.1 *(1.25 and 1.45 in text) If A and B are regular, so is $A \cup B$ $\{x : x \in A$ or $x \in B\}$.*

Proof: The slide shows there is an NFA that accepts $A \cup B$.

By theorem 1.39, $A \cup B$ is a regular language. ■

Theorem 4.2 *Let A and B be regular. Then so are:*

- *Concatenation:* $A \circ B = \{x \circ y : x \in A, y \in B\}$
- *Star:* $A^* = \{x_1 \circ x_2 \circ \cdots \circ x_k : \text{ each } x_i \in A \text{ and } k \geq 0\}$
- *Complement:* $\Sigma^* \setminus A = \{x : x \notin A\}$

Note: *Σ is in $A*$ $\rightarrow$ start state must be an accepting state*

Proof: (Theorem 1.39 and 1.40)

Main claim: Let $M = (Q, \Sigma, \delta, q_0, F)$ be an NFA. Let L ba a language accepted by M. We can construct a DFA $M' = (Q', \Sigma, \delta', q_0, F)$ that also accepts L (so M and M' are equivalent).

First we need to define ϵ-closure. For any set $S \subseteq Q$, let $E(S)$ be the set of all states in Q that can be reached by following any number of ϵ-transitions.

Back to proof:

- The states $Q' = 2^Q$ $\{S : S \subseteq Q\}$
- Accepting state $F' = \{S \subseteq Q :$ any state in S is an accepting state $\} = \{S \subseteq Q : S \cap F \neq \emptyset\}$

Start state: $q_0' = E(\{q_0\})$

Transition function δ':

If NFA could be in states S, next input symbol is a, what states could it be in next?

- First, it could follow any ϵ-transition, so could move to any state in $E(S)$.
- Next, unite $E(S) = \{S_1, \cdots, S_n\}$ could move to any state in $\delta(S_1, a) \cup \delta(S_2, a) \cup \cdots = \cup_{S \in E(S)} \delta(S, a)$.
- Again, it can follow any ϵ-transitions

Final definition: $\delta'(S, a) = E(\cup_{S \in E(S)} \delta(S, a))$. ■