CIS 511: Theory of Computation **Jan 17, 2017**

Lecture 2:

Professor Sampath Kannan *Zach Schutzman*

NB: *These notes are from CIS511 at Penn. The course followed Michael Sipser's Introduction to the Theory of Computation (3ed) text.*

2.1 More Finite Automata

Recall: every DFA recognizes some (regular) language, a language is regular if there exists some DFA recognizing it. NFAs also recognize regular languages (every DFA is also an NFA).

Question: are NFAs more powerful than DFAs? They are certainly a broader class of machines, given that the set of all DFAs is a propers subset of the set of all NFAs.

Theorem 2.1 *NFAs recognize exactly the class of regular languages.*

Proof: *(A rough sketch)*

Consider an NFA M without ϵ-transitions. We can imagine a string's traversal as maintaining a set of possible states you are in, and the NFA accepts if and only that set contains a final state at the conclusion of reading the string. Let's consider the set of states at each step.

Create a DFA M' with states corresponding to subsets of states of the NFA. Now, add transitions in M' corresponding to the set of possible transitions in M. Formally, let $S \subseteq Q$, then $\delta'(S, a) = \bigcup_{q \in S} \delta(q, a)$. These subsets S correspond to states in M'.

The start state of M', q_0' is the state corresponding to $\{q_0\}$. The final states of M', F' are the subsets of Q containing at least one element of F.

We can deal with ϵ-transitions by extending δ' to also include all states reachable from reading a and a following ϵ. Formally, define $E(q)$ as the set of states reachable from q without consuming an input character (i.e. 0 or more ϵ-transitions). Then make $\delta'(S, a) = \bigcup_{p \in E(q) \,:\, q \in S} \delta(p, a)$.

■