Mathematical Methods for Computer Science I

Fall 2016

Outline 8

2.2. **Nondeterministic Finite Automata.**

Definition: A **non-deterministic finite automaton** $N = (Q, \Sigma, \delta, q_0, F)$ has the same definition as a DFA, *except* that its transition function $\delta : Q \times \Sigma \to 2^Q$ takes as input a state and a string and returns a *set of states.* (2^Q: set of all subsets of Q)

Definition: The **language accepted** or **recognized by the NFA** $N = (Q, \Sigma, \delta, q_0, F)$ is the set $L(N) = \{w \in \Sigma^* \mid \delta(q_0, w)$ *contains* a state in $F\}$.

Theorem. *Let L be a language accepted by an NFA. Then there exists a DFA that accepts L.*

2.3. **NFA with ε-transition: ε-NFA.**

Definition: An **ε-NFA** $= (Q, \Sigma, \delta, q_0, F)$ has the same definition as a NFA, *except* that $\delta : Q \times (\Sigma \cup \{\varepsilon\})^* \to 2^Q$.

Definition: The **closure** of a state q, denoted by $\mathcal{C}(q)$ is the set of all states (vertices) p such that there is a path from q to p where edges are all labelled ε (note that $q \in \mathcal{C}(q)$).

Definition: The **language accepted by an ε-NFA** $E = (Q, \Sigma, \delta, q_0, F)$ is
$L(E) = \{w \mid \hat{\delta}(q_0, w)$ contains a state in $F\}$.

Theorem. *Let L be a language accepted by an ε-NFA. Then there exists a NFA that accepts L.*