HW1 Problem 1

Justin Law

February 4, 2019

1 Proving All-NFAs accept the class of regular languages

Theorem 1 We can show that all-NFAs accept the class of regular languages, since they are just as computationally powerful as DFAs.
Proof We will show that an all-NFA can be simulated by a DFA, and vice versa.
Lemma 1 For every all-NFA A, there exists a DFA B such that $L(B) = L(A)$.
Proof We use the subset construction to show that an all-NFA $A = (Q_A, \Sigma, \delta_A, q_{0A}, F_A)$ can be simulated with a DFA $B = (Q_B, \Sigma, \delta_B, q_{0B}, F_B)$. Define B as follows:

1. $Q_B = 2^{Q_A}$
2. $q_{0B} = ECLOSE(q_{0A} \in Q_B$
3. $F_B = \{S | S \in Q_B, S \subseteq F_A\}$
4. Let $a \in \Sigma, R = \bigcup\limits_{s \in S} \delta_A(s, a)$ in: $\delta_B(S, a) = \bigcup\limits_{r \in R} ECLOSE(r) \in Q_B$

This setup is exactly the same as in the proof for the theorem stating normal NFAs can be simulated by DFAs, save for item 3. That is, we changed it so every state in A corresponding to S must be in F_A in order for B to accept. Since item 4 is identical, it has already been shown that the extended transition functions are equivalent via induction. However, we include it here for thoroughness.
By induction, we can show $\hat{\delta}_A(q_0, w) = \hat{\delta}_B(q_0, w)$ for a string $w \in L(A)$.

- Base Case: $\hat{\delta}_A(q_0, \epsilon) = \hat{\delta}_B(q_0, \epsilon) = q_0$
- Inductive Step: Let $w = xa, a \in \Sigma$. Assume $\hat{\delta}_A(q_0, x) = \hat{\delta}_B(q_0, x) = \{p_1, p_2 \cdots p_k\}$. By induction, $\hat{\delta}_A(q_0, w) = \bigcup\limits_{i=1}^{k} \delta_A(p_i, a)$. By subset construction, $\delta_B(\{p_1, p_2 \cdots p_k\}, a) = \bigcup\limits_{i=1}^{k} \delta_A(p_i, a)$. So, $\hat{\delta}_B(q_0, w) = \delta_B(\hat{\delta}_B(q_0, x), a) = \delta_B(\{p_1, p_2 \cdots p_k\}, a) = \bigcup\limits_{i=1}^{k} \delta_A(p_i, a)$. Thus, $\hat{\delta}_A(q_0, w) = \hat{\delta}_B(q_0, w)$.

Since both A and B accept a string w if $\hat{\delta}_A(q_{0A}, w)$ or $\hat{\delta}_B(q_{0B}, w)$ contain only states in F_A, we prove that $L(B) = L(A)$.

$\square$

Lemma 2 Trivially, for every DFA B, there exists an all-NFA A such that $L(A) = L(B)$.

Proof Since B is deterministic by definition, there is only one possible outcome given a string w. In order for B to accept w, $\hat{\delta}_B(q_{0B}, w) \in F_B$. When that happens, we can say every outcome is in F_B, so B is already an all-NFA, and we can construct A identically to B.

$\square$

Theorem 1 All-NFA-s accept the class of regular languages.

Proof By lemmas 1 and 2, we have shown that all-NFAs and DFAs are equally powerful. Since we know DFAs recognize the class of regular languages, so to must all-NFAs. This completes the proof.

$\square$