ECE374 SP23 HW3

Contributors

Zhirong Chen (zhirong4)

Ziyuan Chen (ziyuanc3)

Problem 4

An all-NFA M is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ that accepts $x \in \Sigma^*$ if **every** possible state that M could be in after reading input x is a state from F, in contrast to an ordinary NFA that accepts a string if **some** state among these possible states is an accept state.

Prove that all-NFAs recognize the class of regular languages.

Solution

Proof. We prove that a language is regular *iff* it can be represented by an all-NFA.

If. Any DFA represents a regular language. Then it suffices to show the transformation from an all-NFA to a DFA.

Given a typical NFA $N = (Q, \Sigma, \delta, s, F)$, we can transform it into a DFA

$$D = (Q', \Sigma, \delta', s', F')$$

where

- $Q' = P(Q)$
- $\delta'(X, a) = \bigcup_{q \in X} \delta(q, a), \forall X \subseteq Q, a \in \Sigma$
- $s' = \epsilon\text{-reach}(s) = \delta^*(s, \epsilon)$
- $F' = \{X \subseteq Q \mid X \cap F \neq \emptyset\}$

The four equations above also apply to the transformation of an all-NFA, *except* for F' which must be redefined as

$$F' = \{X \subseteq Q \mid X \subseteq F \text{ and } X \neq \emptyset\}$$

X must be (1) non-empty and (2) a subset of F. This corresponds to the definition of an all-NFA that **every** possible ending state should be an accept state.

We have proven that any all-NFA represents a regular language.

Only if. Any regular language can be represented by a DFA.

An interesting observation is that a DFA ***itself is*** an all-NFA: a string x is accepted if "every possible" state a DFA could be in after reading x is an accept state. There is in fact only one such state -- the machine is deterministic!

We have proven that any regular language can be represented by an all-NFA. **Q.E.D.**