CPSC 406 Assignment 1

Cameron Irving

February 5, 2024

1 Report

The section of reading primarily covered DFA's or Deterministic Finite Automata where for each input there is precisely one state that the automation can move to, similar to a doubly linked list data structure. A DFA consists of a set of states, a set of inputs, a transition function that receives an input and returns the new state, a starting state, and a final state which we abbreviate into the shorthanded form as

$$A = (Q, \sigma, \delta, q_0, F)$$

The DFA will only accept strings that we allow it to accept in the set of inputs and we can broaden the complexity of the DFA by adding more states until we have our own language. We can visualize DFA's by drawing them as transition diagrams or by placing the data into transition tables which is essentially the transition function set as a table showing the inputs and their respective outputs.

2 Exercise 2.2.2

Prove that for any state q and strings x and y:

$$\hat{\delta}(q, xy) = \hat{\delta}(\hat{\delta}(q, x), y)$$

To accomplish this, we will do an induction on y
Basis: If $y = \epsilon$ then $\hat{\delta}(q, x) = \hat{\delta}(\hat{\delta}(q, x), \epsilon)$. Treat $\hat{\delta}(q, x)$ as a state and call it s
Induction Hypothesis: For strings shorter than y, let $y = zl$ where l is the last value of y, then:

$$\hat{\delta}(\hat{\delta}(q, x), y)$$

By assumption of $y = zl$

$$\hat{\delta}(\hat{\delta}(q, x), zl)$$

By the definition of $\hat{\delta}$ and treating $\hat{\delta}(q, x)$ as a state we get

$$\delta(\hat{\delta}(\hat{\delta}(q, x), z)l)$$

By applying the Inductive Hypothesis we get

$$\delta(\hat{\delta}(q, xz), l)$$

By the definition of $\hat{\delta}$

$$\hat{\delta}(q, xzl)$$

And finally since $y = zl$

$$\hat{\delta}(q, xy)$$

$\blacksquare$

3 Question

Is there a point where the number of states in a DFA causes it to become inefficient to have to change states one by one or with modern computing is the "foolproof" design of a DFA more valuable?