PGSS: Computer Science Notes

Etash Jhanji

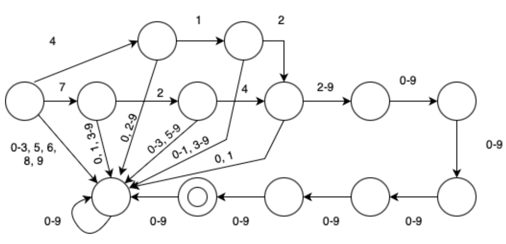
1 Deterministic Finite Automata (DFAs)

- A state machine which given any natural language input will always return the same output where a natural language is any language which is accepted by a DFA.

- Any language can be used as long as every letter is defines (e.g. 0-1 (binary), a-z (english))

Note Because DFAa are <u>deterministic</u> you must define a transition state for any character in the alphabet for every state.

- Sink state: where an input repeatedly returns itself into a reject state and cannot get out of it

- 

2 Pigeonhole Principle

If there exists a number n of pigeonholes and m pigeons is greater where $m > n$ there must be more than one pigeon sharing a hole.

2.1 Pumping Lemma for Regular Languages

If A is a regular language, then there must exist some DFA m which recognizes A.

Let p be the number of states in M.

Note By definition the number of states must be finite.

Consider any String $s \in A$ with length $\geq p$, there must be at least 1 state which is visited more than once when processing s because of the *Pigeonhole Principle*.

Taking the path that corresponds to processing s and dividing it into 3 parts, let x be the portion which we cross before the first repeat, y be the portion which repeats, and z be the portion of the path which takes us to an accept state.

M will accept any string of the form xy^iz for $i > 0$

2.2 Even number of 0's and 1's

Assume machine M with p states recognizes the language with even numbers of 0's and 1's

Thus $s = 0^P 1^P$ defines the string

By the pumping lemma there must be an $xy^i z$ where $|y| \geq 0$ and $|xy| \leq p$.

Let i be arbitrarily defined as 2, $i = 2$ thus $xy^i z = xy^2 z$

We use the entire definition to recognize the portion fo 0's from the input and cannot define the portion for 1's thus contradicting the definition of a DFA.

It is not possible to define a DFA which can recognize an even number of 0's and 1's.