A graph-based, Turing machine-based hybrid approach for data structuring

-Modding edges into a 3-tuple that also includes a transition letter

Null State = -1

Based on the mathematical definition, a DFA is pretty much a set of rules and information for telling if a word is in a given language

Goals

* Make *Theory of Formal Languages* a more interactive class by developing software students can use to actually simulate and create automata
* Make automata directly accessible for those who need to incorporate it in their projects/software
* Study the different computer implementations of formal languages, their pros and cons
  + Graph Theory, Tables, Linked Structures
* Study the applications of Automata outside of traditional uses
* Use Regular Expression notation to simplify the process
* An emphasis of ease of use over efficiency, due to the fact that we are dealing with very non-trivial material.

Uses (compare this to traditional/iterative methods)

* Searching for substrings that involve more complex patterns that just 1 sequence of letters
* Complex Data/input sanitization
  + It is easier to write:
    - If(word in language)
  + Than
    - IF(word has even parity AND has EXACTLY one substring of “101” AND terminates in 0 AND … )
* Otherwise, the more conditions (the stricter L(A) is), the more variables you have to keep track of when you iterate through a variable to check these conditions, and some of these conditions are very non-trivial to check.