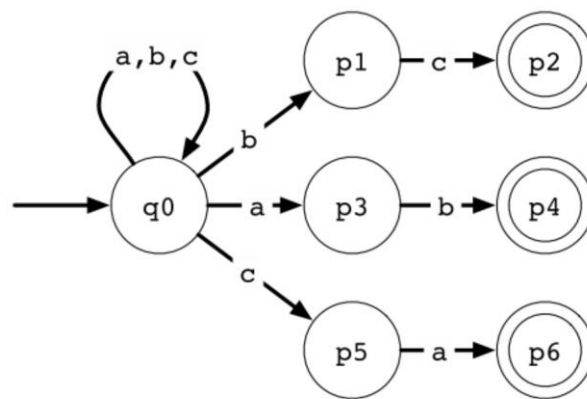


Exercise 3 – Non-Deterministic Finite Automata (NFAs)

It is usually easier to define an NFA for recognizing a given regular language than to define a DFA. In many cases, the NFA is then converted to a DFA and it is the DFA which is implemented in software.

Suppose that for a given application, the NFA below is given as part of the specification of the input strings (over $\Sigma=\{a, b, c\}$) to be accepted in a text box.



- Describe (in plain text) the language accepted by this NFA.
- Represent this automaton using the formal notation.
- Explain the possible reason why there are 3 different final states (p2, p4 and p6) instead of just one.
- Convert the NFA into a DFA accepting the same language using the subset construction method. Present both the transition table and the diagram of the resulting DFA.