Proposed algorithm to create simple Finite Automata

Kavita Tewani

Computer Engineering Department, ITM Universe, Vadodara

Abstract - Theory of computation is been one of the challenging course from student perspective since it involves the abstract notions and mathematical backgrounds. However, there are so many applications of finite automata in real world and hence it becomes necessary to understand the working of automata. The finite automata is mainly of two types - a deterministic finite automaton (DFA) - also known as deterministic finite state machine - is a finite state machine that accepts/rejects finite strings of symbols and only produces a unique computation (or run) of the automaton for each input string. 'Deterministic' refers to the uniqueness of the computation. However there is no fixed algorithm to create a DFA. The paper proposes an algorithm to create a basic automaton.

Keywords - Theory of Computation, DFA, NFA, JFLAP, Basic DFA, Transition function

I. INTRODUCTION

A finite automaton is a 5-tuple \( \langle Q, \Sigma, q_0, \delta, A \rangle \) where \( Q \): set of states, \( \Sigma \): set of input symbols, \( q_0 \): initial or start state, \( A \): set of final states and \( \delta \): transition function (for DFA mapping is \( Q^* \Sigma \rightarrow Q \) and for NFA mapping is \( Q^* \Sigma \rightarrow 2^Q \)). The mapping shows that in DFA there is unique transition for every input symbols whereas in case of NFA, we have multiple choices for transition on a unique symbol. The design of both the automata is different; NFA is comparatively easy to design but DFA, as it has restriction for the transition, requires some rules to be followed. The paper proposes a simple algorithm to create the basic DFA where we have restriction on the start, end and substring of the string.

II. ALGORITHM

The algorithm is divided into three cases according to the restricted string and text in italics shows the execution of the algorithm with respect to the example.

1. If the condition string contains restriction on n symbols then number of states = n + 1.
2. Initially accept the individual symbol of the string one by one.
3. The first symbol starts with the initial state and the last symbol transit to the final state.
4. Let \( \Sigma = \{x_1, x_2\} \) denote the set of input symbols and i denote the first symbol of the restricted string and d denotes the first state.

\[ \text{Consider the string contains restriction on abb, i.e., on 3 symbols so number of states} = 4. \text{ Here } q_0 \text{ is the initial state and } q_3 \text{ is final state}\]

5. Let x be the symbol whose transition is defined and x’ is the symbol whose transition has to be defined.

\[ \text{In our example for state } q_0: x = a, x' = b. \]
6. Repeat the following steps until all the state has unique transition for all input symbol.

7. If the restriction is on the end of the string
   i) The symbol $x'$ will transit in a loop (i.e. on the same state for $d$).

   The symbol $b$ will transit in same state.

   ii) If $i=i+1$
       Then the transition for $x'$ will transit from $d+1$ state to state $d$.
       else
       The symbol $x'$ for state $d+1$ will transit in loop.

   iii) Now a symbol will transit to the state $d$ if at some state whole string is to be accepted, to the state $d+1$ if $n-1$ symbols are to be accepted, to state $d+2$ if $n-2$ symbols are to be accepted and so on. Generally it can be written as a symbol transit to state $d+j$ if $n-j$ symbols is to be accepted.

   iv) Initialize $d=0$, $k=2$
       while($d+k<n+1$)
       a) Read the symbols on the transition from state $d$ to $d+1$ and from $d+1$ to $d+2$ from $d+2$  ... ...$d+k$ (i.e. two symbols).
       In the reference example the symbols will be $ab$.
       b) Append the symbol $x'$ with the above string.
       i.e. $aba$
       c) Compare the string with the given restriction input string.
       On comparing we find that the string becomes $aba$ and according to restriction it should end with $abb$, therefore to fulfil the condition the string must append $bb$ (2 symbols).
       d) Find out the remaining symbols that have to be read to fulfil the condition.
       Remaining symbols=$bb$
       e) If the symbols to be read are $n-j$ then transit it to state $d+j$.
       Symbols to be read=2 (3-1) then transit to state $d+1$ (where $j=1$)
f) Increment the value of k. 
   Now k=3, which is less than 4. So by repeating above steps. We get

8. If the restriction is on the start of the string
   i) The symbol x' will go to a state (dead state), for all the states except for the last state.
ii) For the final state and dead state all the input symbols will go in a loop.

9. If the restriction is on the substring then follow step 6 except for the final state. For final state follow step 7(ii).
REFERENCES