International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 1

A Novel Algorithm for Reduction of Non-Deterministic Finite Automata

G. Mutyalamma, K. Komali, G. Pushpa

Asst.Prof Department of Computer Science and Engineering,
Dadi Institute of Engineering & Technology, Anakapalle, A.P, India

ABSTRACT

In automata theory a minimization is the task of transforming a given finite state machine into an equivalent automation that has a minimum number of states. Here, the reduction of Deterministic Finite Automata(DFA) is very simple whereas Nondeterministic Finite Automata(NFA) is complex because which has maximium number of possible paths to reach new states. So a minimal NFA is a primal problem in automata theory. We consider the problem of *approximating* a minimal NFA or a minimal regular expression. There are several approaches to NFA minimization either without approximation guarantees or running in at least exponential time. Here this paper introducing the new NFA reduction algorithm for the minimization of NFA. This algorithm will reduce number of state transitions of Nondeterministic Finite Automata. NFA reduction algorithm also resolves the complexity of Kameda- Weiner algorithm. This paper shown empirically that this algorithm is effective in largely reducing the memory requirement of NFA minimization algorithm. Reducing the size of NFA by using NFA Reduction Algorithm has been shown to reduce importantly the search time.

Keywords: *Non Deterministic Finite Automata (NFA), Simplest Automation Matrix, Simplified Functional Matrix (SFM), Transition table, Transition function.*

I. INTRODUCTION

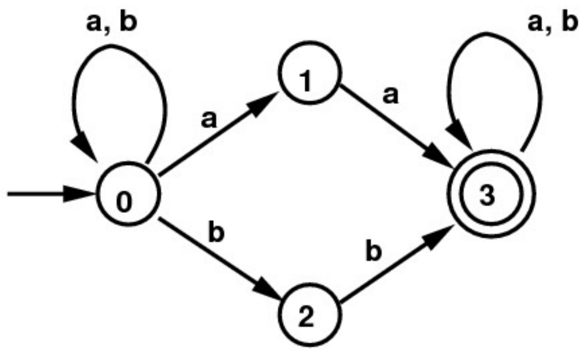
By NFA reduction algorithms, we mean algorithms which from a given NFA produce a smaller equivalent NFA w.r.t. the number of states. Actually, the main algorithms given in this paper also reduce the number of transitions, so that there is no ambiguity about which complexity measure is considered as being reduced. Among automata which recognize a given regular language *L*, the problem of computing one or every minimal NFA w.r.t. the number of states has been shown to be *NP*-hard in [3]. Indeed, known algorithms like in [9,13,4] are quite not practicable. Our present aim is to provide reduction algorithms which have a polynomial complexity w.r.t. the initial number of states in the given NFA. Such algorithms may be useful, for instance, to prevent or moderate the blow up during the determinization of the NFA, or to speed up either its straightforward simulation, or its imulation based on a partial determinization. Among the most basic objects of formal language theory are regular languages and their acceptance devices, finite automata and regular expressions. Regular expressions describe lexical tokens for syntactic specifications, textual patterns in text manipulation systems and they are the basis of standard utilities such as scanner generators, editors or programming languages (perl, awk, php). Internally regular expressions are converted to (nondeterministic) finite automata and the succinctness of this representation crucially determines the running time of the applied algorithms. Contrary to the problem of minimizing dfa’s, which is efficiently possible, it is well known that nfa or regular expression minimization is computationally hard, namely PSPACE-complete [11]. Jiang and Ravikumar [8] show moreover that the minimization problem for nfa’s or regular expressions remains PSPACE-complete, even when specifying the regular language by a dfa.

The state minimization of deterministic finite automata (DFAs) is well-known but the state minimization of nondeterministic finite automata (NFAs) is more complicated.

A nondeterministic finite automaton (or NFA) can be defined as a 5-tuple (Q, $\sum$, T, q_0, F) Where,
-Q is a finite set of states.
-$\sum$ is the input alphabet
-q_0 is the initial state

Where q_0€Q
-F is the final state

Where F€Q i.e, set of final states
-T: $Q\times\sum \rightarrow 2^Q$ is the transition function



Nondeterministic automata allow more possible transitions with same input for each state to new state. NFA also allows epsilon as transition which means transition with epsilon does not carry any character as input to any new state. You can observe in this above example.

***Transition table for NFA**:*

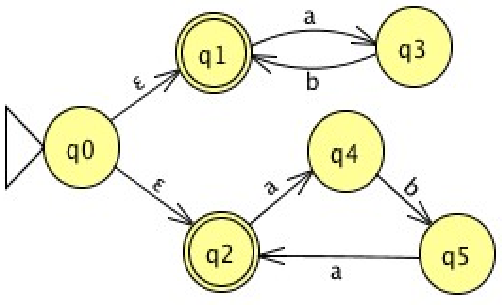
Staes/inputs	a	b
0	{0,1}	{0,1}
1	{3}	Ø
2	Ø	{3}
3	{3}	{3}

Finite automata used to recognize and define the regular languages. Finite automata can be classified into two types- FA without output and finite automata without output. DFA,NFA,EPSILON-NFA related to FA without output. Now question is among these which is powerful language? Every language is powerful when we prove equivalence of each automata. If we convert DFA to NFA or NFA to DFA or NFA to epsilon-NFA or epsilon-NFA to NFA all languages are powerful.

DFA→NFA
NFA→DFA
E-NFA→NFA
NFA→E-NFA

If any automata conversion is possible both are equivalent and powerful. Minimal DFA(with minimum no of states) construction is very familiar to us where as NFA is difficult to get minimal NFA because it has 2^Q states.

NFA's structure for regular expression with epsilon (ab)∪(aba)*.*



T(q0,E)→{q1,q2}	T(q0,{a,b})→ Ø
T(q1,a)→{q3}	T(q1,b)→ Ø
T(q2,a)→{q4}	T(q2,b)→ Ø
T(q3,a)→ Ø	T(q3,b)→{q1}
T(q4,a)→ Ø	T(q4,b)→{q5}
T(q5,a)→ {q2}	T(q5,b)→ Ø

M=({q0, q1, q2, q3, q4, q5},{a,b},T,q0,{q1,q2})

The rearward word (w) of a word (w) {w=x_1, x_2.... x_n} will be w= x_n, x_{n-1}… x_i and if we have a language L then the rearward language of a language (L) will be L = {w|w€i} for an automaton Z= (Q, Σ, Δ, I, F) the rearward automaton will be Z'= (Q,Σ,Δ,I,F)

For a granted language L if DFA distinguishes the language and it also has the minimal possible no. of states then this type of automata is known as Orthodox Automata and for rearward language L if DFA recognize the rearward language L and it also has the minimal possible number of states then this type of automata is known as Rearward Orthodox Automata. The NFA state minimization problem can be defined as follows: find an automaton which is equivalent to given NFA with minimum possible number of states. Note that solution of this problem may not be unique

II. RELATED WORK

The paper by Himanshu Pandey, V. K Singh, Amit Pandey [14] on " **A New NFA Reduction Algorithm for State Minimization Problem** give a more efficient algorithm for constructing the same equivalence, together with results from a computer implementation". We are inspired by the work of V. K Singh [1]. We are inspired by the work of Guangming Xing [1], shown that no auxiliary states can be removed without violating the describing properties of Thompson NFA in his paper "Minimized Thompson NFA". In this paper 'Reducing the Size of NFAs by Using Equivalences and Preorders' Lucian llie, Roberto Solis-Oba, Sheng yu clubbed the concept of Equivalences and Preorders for minimization of NFA. This article explains the minimization of nfa's or regular expressions either for given nfa's or regular Expressions[16].

We consider the problem of approximating a minimal nfa or a minimal regular expression. There are several approaches to nfa minimization [2,5,6,10] either without approximation guarantees or running in at least exponential time. This article explains why such guarantees cannot be expected for efficient algorithms. The classical reduction algorithms for DFAs, e.g. in [7], are based on an equivalence relation over the set of states *Q* which converges step by step toward the coarsest equivalence relation such that all states contained in one equivalence class have the same right language.

The concept of using Hash Table for minimization of DFA is very useful concept for creating a minimal DFA. This concept is given by Vishal Garg , Anu in 2013. Yi. Liu, Taghi M. Khoshgoftaar [12] in DFA Minimizing State Machines Using Hash- Tables. We are inspired by the work of C. Hsiang Chan, R. Paigeb [15] overcome drawbacks of both methods with a O(r) time O(s) space algorithm to construct an O(s) space representation of McNaughton and Yamada's NFA. Given any set V of NFA

III. IMPLEMENTATION

By the help of NFA reduction algorithm we have to minimize the non Deterministic finite automata. If we have a NFA with following condition:

-Initial state has a self loop and incoming or outgoing transition.

For that type of condition we will use NFA Reduction Algorithm.

Steps of NFA Reduction Algorithm:

Step1: Firstly we have to draw a transition table of given NFA (Which satisfy the following condition).

Step2: Then we construct Simplest Automaton Matrix with the help of transition table.

Step3: Now we draw a transition graph of following NFA with opposite transition.

Step4: Next we create a transition table of rearward NFA and construct the Rearward Automaton Matrix.

Step5: With the help of simplest automaton and rearward automaton matrix, we will form a Simplified Functional Matrix (SFM).

Step6: The elements of the SFM is defined by the following formula

$$\mathbf{x \cap y = \acute{\emptyset} \rightarrow 0}$$
$$\mathbf{x \cap y = \acute{\emptyset} \rightarrow 1}$$

Step7: Now we apply these steps on the SFM table

(i) $(x_i \cap y_i) \rightarrow U$
(ii) $(x_i \grave{U} y_i) - (x_i \cap y_i) \rightarrow V$
(iii) $j: U \rightarrow V$

NFA reduction algorithm exploits unique features of the minimization NFA to achieve high throughput.
We have taken a transition graph of NFA

Regular expression is: (a+b)*a(a+b)b

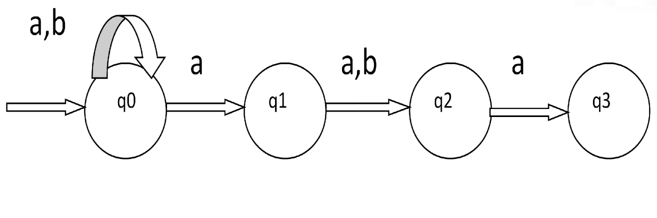


Table 1: Transition Table of given Diagram

states	I/P=a	I/P=b
q0	{q0,q1}	{q0}
q1	{q2}	{q2}
q2	Ø	{q3}
q3	Ø	Ø

Table 2: Simplest Automation Matrix

states	I/P=a	I/P=b
q0	{q0,q1}	{q0}
{q0,q1}	{q0,q1,q2}	{q0,q2}
{q0,q2}	{q0,q1}	{q0,q3}
{q0,q3}	{q0,q1}	{q0}
{q0,q1,q2}	{q0,q1,q2}	{q0,q2,q3}

Diagram of NFA with opposite transition

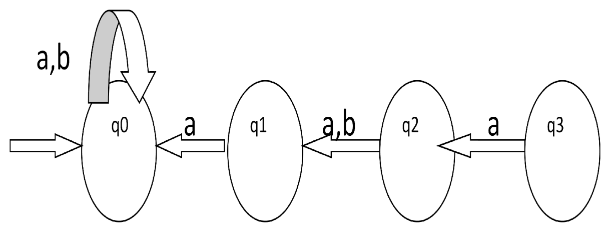


Table 3: Transition table of rearward NFA

states	I/P=a	I/P=b
q0	{q0}	{q0}
q1	{q0}	Ø
q2	{q1}	{q1}
q3	{q2}	Ø

Table 4: Rearward Automation matrix

states	I/P=a	I/P=b
q0	{q0}	{q0}

For creating SFM table we will take the state column of simplest automation matrix and for SFM rows we will take the state column of rearward automation matrix.

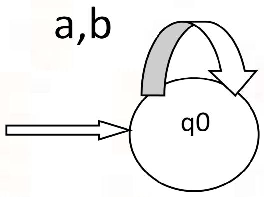
Table 5: Simplified Functional Matrix(SFM)

X$_i$/y$_i$	{q0}
q0	{q0}
{q0,q1}	{q0}
{q0,q2}	{q0}
{q0,q3}	{q0}
{q0,q1,q2}	{q0}

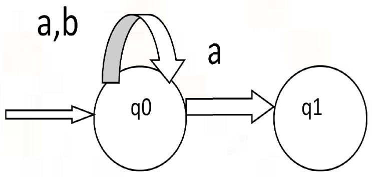
6. The elements of the SFM is defined by the following formula
$x\cap y= Ø\rightarrow 0$
$x\cap y= Ø\rightarrow 1$

7. Now we apply these steps on the SFM table
Steps are:
(i) $(x_i\cap y_i) \rightarrow U$
(ii) (x$_i$Ùy$_i$) - $(x_i\cap y_i)\rightarrow V$
(iii) $j:U\rightarrow V$

[1] $(x_i \cap y_i) \rightarrow (q0\cap q0) \rightarrow q0=U$
(x$_i$Ùy$_i$) - $(x_i\cap y_i)\rightarrow$(q0Ùq0)-$(q0\cap q0) \rightarrow q0=V$
$U\rightarrow V$ means $q0\rightarrow q0$
Now we check that transition ($1\rightarrow1$) is exist or not in the original NFA. We can see that transition ($1\rightarrow1$) is
exist, so the transition graph will be,

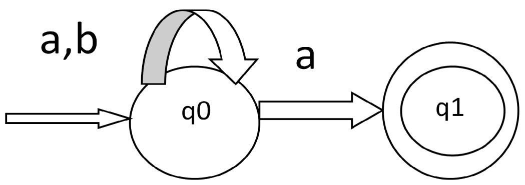


[2] $(x_i\cap y_i) \rightarrow \{(q0,q1) \cap(q0)\} \rightarrow q0=U$
[(x$_i$Ùy$_i$)- $(x_i\cap y_i)$] →{[(q0,q1)Ù(q0)]-[$(q0,q1)\cap q0$] $\rightarrow q1=V$
$U\rightarrow V$ means $q0\rightarrow q1$
Now we check that transition ($q0\rightarrow q1$) is exist or not in in the original NFA. We can see that transition ($q0\rightarrow q1$) is exist, so the transition graph will be,



[3] $(x_i\cap y_i) \rightarrow\{(q0,q2) \cap(q0)\} \rightarrow q0=U$
[(x$_i$Ùy$_i$)- $(x_i\cap y_i)$] →{[(q0,q2)U(q0)]-[$(q0,q2)\cap q0$]} $\rightarrow q2=V$
$U\rightarrow V$ means $q0\rightarrow q2$
But there is no transaction between state 1 and state 3 in given NFA. So the minimal NFA will be same as above.
[4] $(x_i\cap y_i) \rightarrow\{(q0,q3) \cap(q0)\} \rightarrow q0=U$
[(x$_i$Ùy$_i$)- $(x_i\cap y_i)$] →{[(q0,q3)U(q0)]-[$(q0,q3)\cap q0$]} $\rightarrow q3=V$
$U\rightarrow V$ means $q0\rightarrow q3$
But there is no transaction between state 1 and state 3 in given NFA. So the minimal NFA will be same as above.

[5]. $(x_i \cap y_i) \rightarrow \{(q0,q1,q2) \cap (q0)\} \rightarrow q0=U$
$[(x_i\grave{U}y_i)-(x_i\cap y_i)] \rightarrow \{[(q0,q1,q2)U(q0)]-[(q0,q1,q2)\cap(q0)]\} \rightarrow$
$(q1,q2)=V$
$U \rightarrow V$ means $q0 \rightarrow (q1,q2)$
We have a transition b/w state q0&q1 but there is no transition between state q0 and state q2 in original NFA. So the final minimal NFA will be:



Algorithm: NFA Reduction Algorithm

Require NFA X
1. Construct Simplest Automaton Matrix and Rearward Automaton Matrix.
2. Construct Simplified Functional Matrix (SFM).
3. Find all the exits combination with the help of these function,
   $(x_i \cap y_i) \rightarrow U$
   $(x_i\grave{U}y_i) - (x_i \cap y_i) \rightarrow V$
   $j{:}U \rightarrow V$
4. Find minimum legitimate cover(s) of SFM and construct minimum state NFA(s).
   Ensure: Minimum state NFA(s) equivalent to *A*

IV. CONCLUSION

We would like to mention first that these reduction techniques work well with some slightly different definitions of automata, like labeled transition systems with finite recognition, which are automata without final states. Recognized words are words that do not cause a deadlock. Our opinion is that the most interesting algorithm is the one based on the first-order approximation, which has a reasonable worst case complexity, and should prove to be very fast in practice.

To our knowledge, this algorithm is the most efficient one in this level of performance. Higher-order approximations may be useful if one really cares about the succinctness of the NFA. In the present paper we have considered new Reduction algorithms for NFA state minimization problem which is known to be computationally hard. These algorithms are widely used in combinatorial optimization. The essential feature of the proposed algorithm is that the most time consuming part of the exact algorithm is replaced with fast local search function. Numerical experiments have shown that such type of concept is much less time consuming and allows obtaining acceptable results. In the future we plan to concentrate on the other time consuming algorithm.

REFERENCES

1. Lucian llie, Roberto Solis-Oba, Sheng yu: 2005,"Reducing the Size of NFAs by Using Equivalences and Preorders", Lecture Notes in Computer Science, Volume 3537, 2005, pp 310-321 in Springer.
2. J.-M. Champarnaud, F. Coulon, Nfa reduction algorithms by means of regular inequalities, Theoret. Comput. Sci. 327 (3) (2004) 241–253.
3. *Université de Rouen, LIFAR* 2003 "NFA reduction algorithms by means of regular inequalities" Theoretical Computer Science 327 (2004) 241 – 253
4. J.-M. Champarnaud, F. Coulon, Theoretical study and implementation of the canonical automaton, Fund. Inform. 55 (2003) 23–38.
5. L. Ilie, G. Navarro, S. Yu, On nfa reductions, in: J. Karhumäki, H.A. Maurer, G. Paun, G. Rozenberg (Eds.), Theory Is Forever, in: Lecture Notes in Comput. Sci., vol. 3113, Springer-Verlag, 2004, pp. 112–124.
6. L. Ilie, S. Yu, Follow automata, Inform. and Comput. 186 (1) (2003) 140–162.
7. J.E. Hopcroft, An *n* log *n* algorithm for minimizing the states in a finite automaton, in: Z. Kohavi (Ed.), The Theory of Machines and Computations, Academic Press, NewYork, 1971, pp. 189–196.
8. T. Jiang, B. Ravikumar, Minimal nfa problems are hard, SIAM J. Comput. 22 (6) (1993) 1117–1141.
9. T. Kameda, P.Weiner, On the state minimization of nondeterministic finite automata, IEEE Trans. Comput. C(19) (1970) 617–627.
10. O. Matz, A. Potthoff, Computing small nondeterministic finite automata, in: Proc. of theWorkshop on Tools and Algorithms for the Construction and Analysis of Systems, Dpt. of CS., Univ. of Aarhus, 1995, pp. 74–88.
11. A.R. Meyer, L.J. Stockmeyer, The equivalence problem for regular expressions with squaring requires exponential space, in: Proc. 13th Ann.

IEEE Symp. on Switching and Automata Theory, 1972, pp. 125–129.

12. Wojciech Wieczorek : 2012, "Induction of Non-Deterministic Finite Automata on Supercomputers", JMLR: Workshop and Conference Proceedings 21:237{242, 2012 The 11th ICGI.

13. H. Sengoku, Minimization of nondeterministic finite automata, Master's Thesis, Kyoto University, 1992.

14. V. K Singh, Himanshu Pandey, Amit Pandey 2015 " **A New NFA Reduction Algorithm for State Minimization Problem"** *International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868*

15. M. Vázquez de Parga, P. García, Damián López, 2013 "A polynomial double reversal minimization algorithm for deterministic finite automata", Theoretical Computer Science 487 (2013) 17–22.

16. Gregor Gramlich, Georg Schnitger 2004 "Minimizing nfa's and regular expressions" Journal of Computer and System Sciences 73 (2007) 908–923