International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 8– No.3, February 2015 – www.ijais.org

A New NFA Reduction Algorithm for State Minimization Problem

Himanshu Pandey
Department of Computer Science,
BBD University,
Lucknow, India

V. K Singh, Ph.D
HOD- Department of Information Technology
BBDNITM
Lucknow, India

Amit Pandey
B.Tech-Computer Science,
SVNIET
Lucknow, India

ABSTRACT

The problem of creating a minimal NFA is a primal (fundamental) problem. Reducing the size of NFA by using NFA Reduction Algorithm has been shown to reduce importantly the search time. This paper innovate a new NFA reduction algorithm for the state minimization of NFA. The analysis of the proposed algorithm is given and also demonstrates the results of the numerical experiments. This paper conceives the problem of reducing the number of state and transition of Non Deterministic Finite Automata. Numerical experiments show that NFA reduction algorithm produces a minimal automation in all most condition. NFA reduction algorithm also resolves the complexity of Kameda-Weiner algorithm. This paper shown empirically that these algorithm are effective in largely reducing the memory requirement of NFA minimization algorithm.

Keywords

Non Deterministic Finite Automata (NFA), Simplest Automation Matrix, Rearward Automaton Matrix, Simplified Functional Matrix (SFM).

1. INTRODUCTION

NFA Reduction algorithms exploit unique features of the minimized NFA to achieve high throughput. In this paper, inquire pertinence of NFA reduction techniques – a conventional algorithm to reduce the number of transitions and states in NFA. The paper presents NFA reduction techniques, with a different reduction time complexity and power. Our focus and main contribution is a study of NFA reduction techniques for creating minimal NFA. In particular,

- Surveyed different variants of NFA reduction techniques.
- Evaluated Kameda Weiner algorithm for NFA Reduction and provide new algorithm for NFA minimization.
- Modified the Kameda Weiner algorithm for NFA reduction to reduce its complexity.

The state minimization of deterministic finite automata (DFAs) is well-known but the state minimization of nondeterministic finite automata (NFAs) is more complicated. Finite automata (FA) are widely used in various fields and peculiarly reorganization of formal Languages. We provide some necessary definitions.

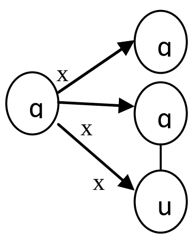
A nondeterministic finite automaton (or NFA) can be defined as a 5-tuple ($Q, \Sigma,, T, q_D, F$) Where,

-Q is a finite set of states.

-Σ is the input alphabet

-T: $Q\times(\Sigma \cup \lambda) \rightarrow Q$ is the transition function.

$$\delta(q, x) = \{q_1, q_2, \ldots, q_k\}.$$



(Resulting states with following one transition with symbol x)

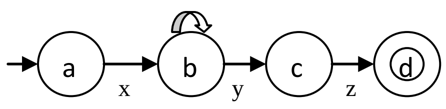
-$q_D \in Q$ is the starting state.

-$F \subset Q$ represents a set of final states.

Finite automata used to recognize and define the regular languages. Two automata are called equivalent if they recognize one and the same language. For each NFA the equivalent DFA may be constructed using the powerset construction process (each state of such DFA is a subset of states of the original NFA). NFAs recognize regular languages, and find that any string is in the language it represents. Consider the following regular language over the alphabet Σ = {a, b} (represented by the regular expression

aa*b

This language can be represented by the following NFA.



The rearward word $\overline{(w)}$ of a word (w) $\{w=x_1, x_{2\ldots} x_n\}$ will be $\overline{w}= x_n, x_{n-1\ldots} x_i$ and if we have a language L then the rearward language of a language (L) will be $\overline{L}$ = {w|wЄi} for an automaton Z= (Q,∑,Δ,I,F) the rearward automaton will be,

$\overline{Z}$= (Q,∑,$\overline{\Delta}$,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.

2. RELATED WORK

The paper by Michael Albert and Steve Linton [6] on "A Practical Algorithm for Reducing Non-Deterministic Finite State Automata" give a more efficient algorithm for constructing the same equivalence, together with results from a computer implementation". 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. Hermann Gruber1 and Markus Holzer2 [8] proposed the investigation of computational complexity of the nondeterministic finite automaton (NFA) minimization problem for finite and unary regular languages, They show this on his paper "Computational Complexity of NFA Minimization for Finite and Unary Languages". The paper 'Local Search Heuristics for NFA State Minimization Problem* by Andrey V. Tsyganov [9] introduce new heuristic methods for the state minimization of nondeterministic finite automata. These methods are based on the classical Kameda-Weiner algorithm joined with local search heuristics. We have used this concept for minimization of NFA. 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. . Henrik Bj¨orklunda, Wim Martens [13] have shown that no such significant extensions exist, under the assumption that PTIME 6= NP. and also proved that minimization is NP-hard for all finite automata classes that contain the NFAs that accept strings of length three. In this paper 'NFA Reduction for Regular Expressions Matching Using FPGA' Vlastimil Koˇsaˇr, Martin ˇZ ´adn´ık, Jan Koˇrenek [11] proposed to accelerate regular expression matching via mapping of a nondeterministic finite automaton into a circuit implemented in an FPGA. These algorithms exploit unique features of the FPGA to achieve high throughput. Paper by Manuel Vázquez de Parga, Pedro García, Damián López [14] proposed a polynomial-time deterministic finite automaton minimization algorithm directly derived from Brzozowski's double reversal algorithm. We take into account the framework by Brzozowski and Tamm, to propose an atomization algorithm that allows us to achieve polynomial time complexity. We are inspired by the work of Jean Vuillemin, Nicolas Gamaon present a cubic time algorithm to reduce a xor-automaton A 2 NXA to a minimal formM = MXA(A) which accepts ¤©(M) = ¤©(A), within the least possible number of states. It is a finite strong normal form SNF: ¤© (A) = ¤© (A0), MXA (A) = MXA (A0) and automata equivalence is efficiently decided through reduction to the SNF. Wojciech Wieczorek clubbed a Supercomputers with nfa, treated the induction of NFAs based on finite languages. That is constituted by the following task: given two disjoint finite sets S+; S- of words and an integer k > 0, build a k-state NFA that accepts the language S+ and does not accept any word from the set S□. We are inspired by the work of C. Hsiang Chan, R. Paigeb [10] 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 states, our representation can be used to compute the set U of states one transition away from the states in V in optimal time 0(1 V I+ 1 U I). McNaughton and Yamada's NFA requires O (1 VI x I UI) time in the worst case.

3. OUR FRAMEWORK

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:

1. Firstly we have drawn a transition table of given NFA (Which satisfy the following condition).
2. Then we construct Simplest Automaton Matrix with the help of transition table.
3. Now we draw a transition graph of following NFA with opposite transition.
4. Next we create a transition table of rearward NFA and construct the Rearward Automaton Matrix.
5. With the help of simplest automaton and rearward automaton matrix, we will form a Simplified Functional Matrix (SFM).

Then, for each nonempty states p_i of $B\left(i=1, \ , m\right)$ $C \ j$ □1, ,
and q_j of □ n□ the element r_{ij} of the SFM is defined by the following formula

$$r_{ij} = \begin{cases} 0, & p_i \cap q_j = \varnothing, \\ 1, & p \cap q_j \neq \varnothing. \end{cases}$$

Let X be a subset of rows and Y a subset of columns of the SFM.

6. Then we compare the values of SFM by this function:

   (i) **$(x_i \cap y_i) \rightarrow U$**

   (ii) **(ii) $(x_i$Ù$y_i) - (x_i \cap y_i) \rightarrow V$**

   (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.

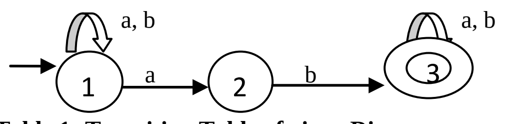


Table 1: Transition Table of given Diagram

State	Input a	Input b
1	1,2	1

2	Ǿ	3
3	3	3

Table 2: Simplest Automation Matrix

State	Input a	Input b
1	1,2	1
1,2	1,2	1,3
1,3	1,2,3	1,3
1,2,3	1,2,3	1,3

Diagram of NFA with opposite transition

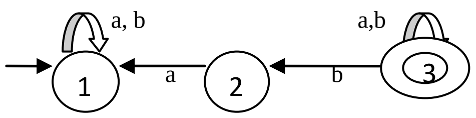


Table 3: Transition table of rearward NFA

State	Input a	Input b
1	1	1
2	1	Ǿ
3	3	2,3

Table 4: Rearward Automaton Matrix

State	Input a	Input b
1	1	1

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	{1}
{1}	1
{1,2}	1
{1,3}	1
{1,2,3}	1

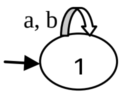
The elements of the SFM is defined by the following formula

$$\begin{cases} x \cap y = \acute{\emptyset} \rightarrow 0 \\ x \cap y = \acute{\emptyset} \rightarrow 1 \end{cases}$$

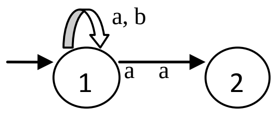
7. Now we apply these steps on the SFM table

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

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



[2]. $(x_i \cap y_i) \rightarrow \{(1,2) \cap (1)\} \rightarrow 1 = U$
$[(x_i \grave{U} y_i) - (x_i \cap y_i)] \rightarrow \{[(1,2) \grave{U} (1)] - [(1,2) \cap 1] \rightarrow 2 = V$
$U \rightarrow V$ means $1 \rightarrow 2$
Now we check that transition $(1 \rightarrow 2)$ is exist or not in in the original NFA. We can see that transition $(1 \rightarrow 2)$ is exist, so the transition graph will be,

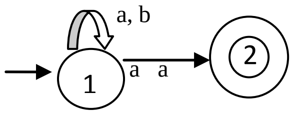


[3]. $(x_i \cap y_i) \rightarrow \{(1,3) \cap (1)\} \rightarrow = U$
$[(x_i \grave{U} y_i) - (x_i \cap y_i)] \rightarrow \{[(1,3)U(1)] - [(1,3) \cap 1]\} \rightarrow 3 = V$
$U \rightarrow V$ means $1 \rightarrow 3$

But there is no transaction between state 1 and state 3 in given NFA. So the minimal NFA will be same as above.

[4]. $(x_i \cap y_i) \rightarrow \{(1,2,3) \cap (1)\} \rightarrow 1 = U$
$[(x_i \grave{U} y_i) - (x_i \cap y_i)] \rightarrow \{[(1,2,3)U(1)] - [(1,2,3) \cap (1)]\} \rightarrow (2,3) = V$

$U \rightarrow V$ means $1 \rightarrow (2,3)$
We have a transition b/w state 1&2 but there is no transition between state 1 and state 3 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*

4. CONCLUSION

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 part of the Ka-meda-Weiner algorithm, *i.e.* the exhaustive search for grids of the RAM.

5. 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] M. Albert and Steve Linton: July 2014. "A Practical Algorithm for Reducing Non-Deterministic Finite State Automata", Elsevier Science.

[3] Andrey V. Tsyganov: September 2012, "Local Search Heuristics for NFA State Minimization Problem", *Int. J. Communications, Network and System Sciences*, 2012, 5, 638-643.

[4] H. Gruber and M. Holzer: "Computational Complexity of NFA Minimization for Finite and Unary Languages",Institut f¨ur Informatik, Ludwig-Maximilians-Universit¨at M¨unchen, Oettingenstraße 67, D-80538 M¨unchen, Germany.

[5] Henrik Bj¨orklunda, Wim Martens: April 2011, "The Tractability Frontier for NFA Minimization", International Colloquium on Automata, Languages and Programming 2008.

[6] Yonghua Zhou Yuliu Chen, "The QFD-based Decision-making Approach for Strategic BPR" Beijing 100084, P. R. China.

[7] Guangming Xing, August 20-22, 2007 "Minimized Thompson NFA", Western Kentucky University, Bowling Green, KY 42101.

[8] V. Garg, Anu: June 2013, "DFA Minimizing State Machines Using Hash- Tables" ,International Journal of Engineering Trends and Technology (IJETT) - Volume4 Issue6- June 2013.

[9] Vlastimil Koˇsaˇr, Martin ˇZ ´adn´ık, Jan Koˇrenek, 2007 "NFA Reduction for Regular Expressions Matching Using FPGA", MSM 0021630528, the IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070 and the grant BUT FIT-S-11-1.

[10] 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.

[11] J. Vuillemin, N. Gama: Dec 2009, "Efficient Equivalence and Minimization for Non Deterministic Xor Automata", [Research Report] 2010, pp.25. <inria-00487031>.

[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] M. Mohri, F. Pereira and M. Riley, "AT&T General-Purpose Finite-State Machine Software Tools," 1997. http://www.research.att.com/sw/tools/fsm

[14] S. Lombardy, R. Poss, Y. Régis-Gianas and J. Sa-karovitch, "Introducing VAUCANSON," In: O. H. Ibarra and Z. Dang, Eds., *Implementation and Application of Automata*, CIAA 2003, Santa Barbara, 16-18 July 2003, pp. 96-107.

[15] S. H. Rodger, "JFLAP: An Interactive Formal Languages and Automata Package," Jones and Bartlett Publishers, Inc., USA, 2006.

[16] T. Kameda and P. Weiner, "On the State Minimization of Nondeterministic Finite Automata," *IEEE Transactions on Computers*, Vol. C-19, No. 7, 1970, pp. 617-627. doi:10.1109/T-C.1970.222994

[17] J. Hromkovic, "Algorithmics for Hard Problems—Intro-duction to Combinatorial Optimization, Randomization, Approximation, and Heuristics," Springer, Berlin, 2001.

[18] F. Glover and G. A. Kohenberger, "Handbook of Meta-heuristics," Kluwer Academic Publishers, Boston, 2003.

[19] V. Kell, A. Maier, A. Potthoff, W. Thomas and U. Wer-muth, "AMORE: A System for Computing Automata, Monoids and Regular Expressions," Proceedings of the 6th Annual Symposium on Theoretical Aspects of Com-puter Science on STACS 89, Springer-Verlag, New York.