International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 4 Issue 6, June 2015

A Survey on Deterministic Finite Automata Compression Techniques

Ms. Utkarsha P. Pisolkar, Asst. Prof. Shivaji R. Lahane

Abstract— **Signatures based regular expression pattern matching is a very major process in many network security applications, such as intrusion detection and prevention system. Finite Automata is used in pattern matching process to represent the patterns. A memory efficient pattern matching process is the need of today's network security applications. There are many techniques available which build the pattern matching process memory efficient which reduces the size of Deterministic finite automata. Deterministic finite automata compression can be done by decreasing number of state, decreasing number of transitions, decreasing size of character set and by decreasing number of bits which represent state. Preceding works on all these techniques are discussed in this paper.**

Index Terms— **Deterministic finite automata, Intrusion detection system, Pattern matching, Regular expression**

I. Introduction

A security of computer system is a very important from malicious activities like viruses. Many security attacks interrupt the functionality of computing infrastructure. To guarantee the protection of computer system from attacks, a lot of security appliances such as intrusion detection and prevention system use variety of security techniques. For a computer system, one may use host base intrusion detection system such as antivirus software. For a network, one may use Network Intrusion detection System. Intrusion Detection System (IDS) inspects all the incoming data and detects the security threat. Signature based pattern matching is most widely used technique. Most security threats like viruses have some sort of signatures. The signature patterns are match up to detect any security threat. Signature based Pattern matching has become the important element of all network security applications to compare these signature patterns [1].

Pattern matching entails examination of a given input data for the presence of some signature pattern. More multifaceted signature patterns are written by regular expressions. Regular expressions are easily represented by finite automata [2]. Finite Automata has two types as Deterministic Finite Automata (DFA) and Non-deterministic Finite Automata (NFA). Regular expressions are easily converted into NFA. NFA is easily converted into DFA. This final DFA has large size of transition table. Therefore to reduce the size of DFA many DFA compression techniques have been proposed. Prior work on all these techniques is discussed in this paper.

Rest of the paper is organized as follows. Section one provides introduction, DFA compression techniques are discussed in section two. The section three presents comparison between DFA compression techniques used till today. A conclusion is given in section four.

II. DFA compression techniques

To make a signature based pattern matching process memory efficient, many DFA compression techniques have been proposed. Techniques broadly classified into decreasing number of transitions, decreasing number of states, decreasing size of character set and decreasing number of bits which represent state.

A. Transition Condensing

In transition condensing technique, the total numbers of transitions are decreased for every state. The D^2FA and CD^2FA have used transition condensing technique.

The Delayed Input DFA (D^2FA) has decreased the space requirements by lessening the number of distinct transitions between states [3]. Default transitions were used to minimize memory requirement of transition function. In transition function of DFA lots of states may have related sets of outgoing transitions; unnecessary transitions were removed by a simply one default transition [3]. For two states u and v which have common transitions to the similar set of states, {S}, for some set of input characters, {X}. These transitions were removed from one state u and launched a default transition from u to v which followed for all the characters in {X}. So, u maintained distinctive next states for those transitions which are not similar to u and u used the default transition to v for the similar transitions. A D^2FA is created by changing a DFA by substituting number of transitions of the DFA with a lone default transition. D^2FA had reduced space requirement of DFA storage. But the disadvantage is that when a default transition is followed, the main memory must be accessed once more to retrieve the transitions of the referred stat and therefore it introduced additional main memory accesses per byte [3].

Content Addressed Delayed Input DFA (CD^2FA) has constructed on the delayed input DFA (D^2FA) to remove disadvantage of it [4]. In this approach state numbers are substituted by content label. These content labels compactly hold information that to be stored in the table entry for the state and successive states of a D^2FA are referred by CD^2FA by using content label information [4]. The precedent default

***Ms. Utkarsha P. Pisolkar**, PG student, Computer Engineering Department, G.E.S.'s R.H. Sapat College of Engineering, Management Studies and Research, Nashik, Savitribai Phule Pune University, India.*

***Mr. Shivaji R. Lahane**, Asst. Prof., Computer Engineering Department, G.E.S.'s R.H. Sapat College of Engineering, Management Studies and Research, Nashik, Savitribai Phule Pune University, India.*

transitions are omitted by content label. In the earlier of state traversal, selected information is available. This change makes the CD^{2}FA to evade any default traversal. Therefore, it had removed the disadvantages of D^{2}FA technique [4].

B. State Condensing

In state condensing technique, the numbers of DFA states are reduced. Hybrid DFA-NFA [5], HFA [6], XFA [7] are based on state condensing technique.

The hybrid DFA-NFA state condensing solution based on the study that DFAs are not good with large sets of regular expression and NFAs lessen the memory storage problem but show the way to a huge memory bandwidth requirement [5]. The cause is that several NFA states may be lively in analogous and every input character may set in motion many transitions. A hybrid DFA-NFA solution is planned which took jointly the advantages of both automata: when creating the automaton, any states that contributing to state blast remains an NFA encoding, while the others are changed into DFA states [5].

The History-based Finite Automaton (HFA) approach has studies three main boundaries of the traditional DFAs based NIDS [6]. First, DFAs have not take benefit of the truth that normal input streams rarely match more than few initial symbols of any signature patterns. Second, the DFAs are very unproductive in many partially matching signature patterns and blow up in size. Third, DFAs are unable of maintaining path of the incidences of some sub-regular expressions. Therefore, the H-FA is projected to resolve all of these limitations by inserting a little information to remain the transition history [6]. It has decreased the number of states. More information is memorized by this approach through laying up them in a tiny and speedy cache which called as history buffer [6]. But limitation of this approach is that it has it has number of restricted transitions for every character, which has outputted in a huge size of transition table and a dawdling inspection speed [6].

The extended character set (XFA) is one more state condensing technique [7]. A XFA has utilized number of automata transformations to eradicate restricted transitions which is limitation of HFA. XFA is restricted to single supplementary state for every regular expression, which is unsuitable for tricky regular expressions [7].

The Delta Finite Automata has condensed the number of states and transitions [8]. This approach is based on the phenomenon that the majority neighboring states contribute to several familiar transitions. Therefore, this approach has stocked up only the dissimilarities among them [8]. The main characteristic of the delta finite automata is that it has requested lone a state transition for every input character. A new state programming design called Char-State reduction based on input characters are projected which has shown the relationship of lots of states with only some input characters [8]. This reduction design has included into the delta finite automata algorithm, which has outputted in a additional DFA memory size diminution [8].

C. Character set Condensing

In character set condensing technique, character set is attempted to condense. The alphabet reduction approach has converted the set of characters in an alphabet to a minor set of clustered characters that make the similar transitions for a considerable quantity of states in the automaton [9]. A number of alphabet compression tables have used as a trivial method for condensing the memory requirements of DFAs. This technique has used an approach to separate the states of a DFA and calculating a divergent alphabet compression table for every partition [9].

D. Bit Condensing

The History based Encoding, eXecution and Addressing (HEXA) approach has used bit condensing technique. It compresses the number of bits required to represent each state and takes the advantage of implicit information to reduce the information that must be stored explicitly [10]. The approach demonstrates that binary tries used for IP route lookup can be implemented using two bytes per stored prefix. In DFA, multiple paths leading to each node are stored in history [10]. By augmenting the history with some additional discriminating information, this approach ensures that each node is mapped to a distinct storage location using appropriate hashing techniques [10].

III. Comparison of DFA Compression Techniques

The Comparison of DFA compression techniques used till today is shown in Table I.

Table I. Comparison of DFA compression techniques

Techniques	**Objectives**
D^{2}FA	D^{2}FA has reduced space requirements by reducing the number of distinct transitions between states. But it includes the traversal of several states when processing a single input character [3].
CD^{2}FA	CD^{2}FA has built upon the delayed input DFA, whose state numbers are replaced by content label. The content label is used to omit past default transitions [4].
Hybrid DFA-NFA	It is state reduction solution that combines the advantages of DFA and NFA. But limitation of this technique is that it has required many automata transformations [5].
HFA	H-FA has reduced number of states by adding some information to keep the transition history. But, it has number of conditional transitions per character, which resulted in a large transition table and a slow inspection speed [6].
XFA	A XFA has used number of automata transformations to avoid conditional transitions in H-FA [7].

Delta Finite Automata	Delta Finite Automata has reduced the number of states and transitions and it is based on the study that most adjacent states share several common transitions, so it stored only the differences between them [8].
Alphabet Compression	It has mapped the set of characters in an alphabet to a smaller set of clustered characters that label the same transitions for a substantial amount of states in the automaton [9].
HEXA	A History based Encoding, eXecution and Addressing (HEXA) approach has compressed the number of bits required to represent each state and takes the advantage of implicit information to reduce the information that must be stored explicitly [10].

IV. Conclusion

Signature based Pattern matching is a main process used in intrusion detection and prevention system. It matches up to signature patterns to identify security threats. It is very significant to decrease the memory space requirement of DFA which represents signature patterns of security threat. The Size DFA can be reduced by techniques like transition condensing, state condensing; character set condensing and bit condensing. Transition condensing technique trims down number of transitions for each state in DFA. State condensing technique trims down the number of states in DFA. In character set condensing, alphabet set of signature patterns is trims down to minor size. Bit condensing technique trims down the number of bits required to represent every state in DFA. At the end, comparison of DFA compression technique is discussed. These techniques are able to be applied as only or in combination of two or more depending on their advantages and disadvantages.

Acknowledgment

We are glad to express our sentiments of gratitude to all who rendered their valuable guidance to us. We would like to express our appreciation and thanks to Prof. Dr. P. C. Kulkarni, Principal, G. E. S. R. H. Sapat College of Engg., Nashik. We are also thankful to Prof. N. V. Alone, Head of Department, Computer Engg., G. E. S. R. H. Sapat College of Engg., Nashik. We thank the anonymous reviewers for their comments.

REFERENCES

[1] AnatBremler-Barr, D.Hay, Y. Koral, "CompactDFA:Scalable pattern matching Using Longest Prefix Match Solutions," in IEEE/ACM Transaction on networking,vol-22,No.2,April 2014.

[2] A.V. Aho and M.J. Corasick. "Efficient String Matching: An Aid to Bibliographic Search." Communications of the ACM, 18(6):333–340, 1975.

[3] S. Kumar, S. Dharmapurikar, F. Yu, P. Crowley, and J. Turner, "Algorithms to accelerate multiple regular expressions matching for deep packet inspection", in Proc. of ACM SIGCOMM , pages 339-350. ACM, 2006.

[4] S. Kumar, J. Turner, J. Williams, "Advanced algorithms for fast and scalable deep packet inspection", in Proc. ACM/IEEE Symp. Archit. Netw. Commun. Syst. (ANCS), pages 81-92. ACM, 2006.

[5] M. Becchi, P. Crowley, "A hybrid finite automaton for practical deep packet inspection", in Proc. Conf. Emerging Netw. Exp. Technol.(CoNEXT), pages 1-12, 2007.

[6] S. Kumar, B. Chandrasekaran, J. Turner, G. Varghese, "Curing regular expressions matching algorithms from insomnia, amnesia, and acalculia", in Proc. ACM/IEEE Symp. Archit. Netw. Commun. Syst. (ANCS), pages 155-164. ACM, 2007.

[7] R. Smith, C. Estan, and S. Jha, "Xfa: Faster signature matching with extended automata", in IEEE Symposium on Security and Privacy, May 2008.

[8] D.Ficara, S.Giordano, G. Procissi, F.Vitucci, G.Antichi, A.D. Pietro, "An Improved DFA for Fast Regular Expression Matching" ACM SIGCOMM Computer Communication Review, Volume 38, Number 5, October 2008.

[9] S. Kong, R. Smith, and C. Estan, "Efficient signature matching with multiple alphabet compression tables," in Proc. Int. Conf. Security Privacy Commun. Netw. (Securecomm), 2008.

[10] S. Kumar, J. Turner, P. Crowley, and M. Mitzenmacher, "HEXA: Compact data structures for faster packet processing", in Proc. IEEE Conf. Comput. Commun. (INFOCOM), 2009.

Ms. Utkarsha P. Pisolkar received her B.E. degree in Computer Engineering from N. D. M. V. P. S.'s K. B. T. College of Engineering, Nashik, Maharashtra, India, in 2012. At present, she is pursuing her M.E. in Computer Engineering from Gokhale Education Society's R. H. Sapat College of Engineering, Nashik. Her research interests include Network & Computer Security, Pattern Matching Algorithms and Theory of Computation

Mr. Shivaji R. Lahane is currently working as an Assistant Professor in the Computer Engineering Department of Gokhale Education Society's R. H. Sapat College of Engineering, Management Studies and Research, Nashik (INDIA). He received his Masters Degree in Computer Science and Engineering from P.R.M.I.T and R. Badnera, Amravati University (INDIA) in 2010 & Bachelors Degree in Computer Science and Engineering from Anuradha College of Engineering Chikhali, Buldhana, and Amravati University (INDIA) in 2005. He has taught Computer related subjects at undergraduate & postgraduate level. His areas of interest are Data Communication, Network & Information Security, Algorithms and Advanced Computer Network.

.