

# Improving Rule Base Quality to Enhance Production Systems Performance

**Nabil Arman**

Department of Computer Science, Palestine Polytechnic University, Hebron, Palestine  
Email: narman@ppu.edu

Received September 1, 2012; revised October 3, 2012; accepted October 12, 2012

## ABSTRACT

Production systems have a special value since they are used in state-space searching algorithms and expert systems in addition to their use as a model for problem solving in artificial intelligence. Therefore, it is of high importance to consider different techniques to improve their performance. In this research, rule base is the component of the production system that we aim to focus on. This work therefore seeks to investigate this component and its relationship with other components and demonstrate how the improvement of its quality has a great impact on the performance of the production system as a whole. In this paper, the improvement of rule base quality is accomplished in two steps. The first step involves re-writing the rules having conjunctions of literals and producing a new set of equivalent rules in which long inference chains can be obtained easily. The second step involves augmenting the rule base with inference short-cut rules devised from the long inference chains. These inference short-cut rules have a great impact on the performance of the production system. Finally, simulations are performed on randomly generated rule bases with different sizes and goals to be proved. The simulations demonstrate that the suggested enhancements are very beneficial in improving the performance of production systems.

**Keywords:** Production System; Rule Base Quality; Inference Short-Cut Rules; Inference Chains

## 1. Introduction

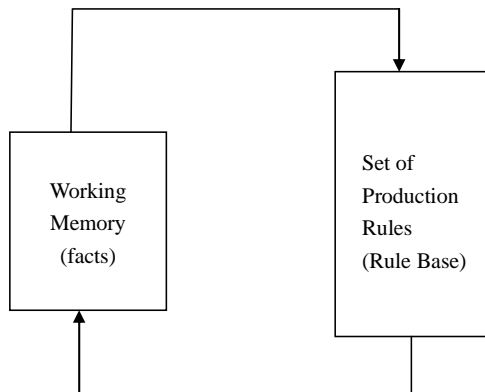
A production system, in the context of artificial intelligence, is a model of computation that has proved particularly important in different sub domains such as search algorithms implementation and modeling human problem-solving [1]. A schematic diagram of a production system is presented in **Figure 1**. As shown, a production system is defined by three major components, namely the set of production rules (a rule base), a working memory containing a description of the current state of the world in a reasoning process, and the recognize-act cycle that represents the control structure for a production system.

Production systems are intensively used in state-space searching algorithms and expert systems in addition to their use as a model for problem solving in artificial intelligence. Thus, improving their performance is an issue of great importance. The improvement can be focused on either improving the representation and access of facts in the working memory or improving the rule base quality. In this paper, rule base quality is the component that we focus on. This work therefore seeks to demonstrate how the enhancement of the rule base quality has a great impact on the performance of the production system as a whole.

In this paper, the enhancement of rule base quality is performed by firstly re-writing the production system rules having conjunctions of literals and producing a new set of equivalent rules in which long inference chains can be devised. Secondly, augmenting the rule base with inference short-cut rules obtained from the long inference chains. These inference short-cut rules improve the performance of the production system. Finally, several simulations are performed on a set of random rule bases with different sizes and goals to be proved. These simulations demonstrate that these enhancements are very useful in improving the performance of the production system.

## 2. Related Work

The performance of a production system has attracted a large amount of research efforts. Improving performance of production systems by restructuring facts, where the focus was on the set of facts in the production system, was presented in [2]. A genetic algorithm for production systems optimization was presented in [3]. The algorithm finds an ordering of condition elements in the rules of a production system that results in a (near) optimal production system with respect to execution time. Trying



**Figure 1. A production system.**

to find such an ordering can be complicated since there is often a large number of ways to order condition elements in the rules of a production system. In addition, using heuristics to order condition elements, in many cases, conflict with each other. The improvements presented in [2,3] are considered dynamic improvements that are employed during production system execution and thus incurred additional overhead that may slow down the production system execution. This affects the performance of a production system negatively.

The quality of rule bases has been a subject of a large amount of research efforts in the context of different application domains including information distribution systems, active databases, expert systems to name just a few [4,5]. In [4], the focus was on generating a set of rules that are fault-free and has a minimum cost in case the cost of applying different rules is known and of great importance. In [5], the focus was on dynamic rule bases where rules' status changes during the operation of the rule-based system. However, these improvements did not include the issue of the rule-based system execution per se.

Another technique for improving the quality of a rule base was presented in [6]. This technique used a number of heuristics to suggest certain corrections of the rule base faults found in the rule base. Again, this technique did not consider the issue of rule-based system execution.

A declarative verification approach for improving the quality of rule-based applications was presented in [7]. This approach presented a particular way to control and to improve the rule quality by means of rule verification using a declarative approach. Although the approach is flexible and easy to maintain, it did not take into account the issue of rule-based system execution.

A state-based knowledge representation approach, in which domain-specific knowledge is expressed by combinations of the relevant objects' states were presented in [8]. Using this technique, a method for detecting logical

inconsistencies was developed, which can deal with the demands of various domain-specific situations through reducing part of restrictions in existing methods. However, this approach did not affect the execution of the system.

In this paper, the focus is on the quality of the rule base rather than the ordering of the facts in the fact base or the ordering of the condition elements in the rules of a production system or the rule base verification. This can be performed before the production system starts execution as a preprocessing step and thus speeds up the production system execution without incurring additional overhead during the production system execution. Therefore, this is considered a static improvement that is performed before a production system starts execution.

### **3. Rule Base Quality Enhancement of a Production System Performance**

The performance of a production system depends largely on the rules in its rule base. The structure of rules in a production system and their interrelationships determine how the search space is explored. In large production systems, many rules are used and these rules will be tried in order based on a conflict resolution strategy. A number of conflict resolution strategies have been used including refraction, recency, priority, and specificity. In this paper, priority is used to order the rules for possible firing during the production system execution.

To improve the performance of a production system, one should consider enhancing the quality of its rule base by handling all possible problems and anomalies associated with rule bases including inconsistency, contradiction, circularity, and redundancy [9]. This should be handled before the production system starts execution and can be performed using well-known verification tools and techniques [5,9]. An inconsistency exists in the rule base of a production system if a condition of one rule is mutually exclusive to the consequent or action of such rule (or a chain of rules). A contradiction exists if two rules conclude different actions from the same condition. A redundancy exists if two rules conclude the same action from the same rule condition. A subsumption exists if two rules conclude the same action, but one has additional constraints in the rule condition, which may or may not be necessary (a specific kind of redundancy). A circularity exists if the rule base contains a cycle that makes the production system to enter an endless loop by keeping the insertion of the same facts to the fact base of the production system without making any progress towards the solution or goal of the problem.

In this paper, the focus is on the issues that affect the production system performance during its execution. A restructuring of the rules based on their actions/consequent makes it easier to apply the major improvement of the

quality of the rule base. Re-writing the rules having a conjunction of literals is of high importance since it facilitates the process of checking other types of anomalies and to apply other improvements. For example, a rule of the form:

$$p \rightarrow q_1 \wedge q_2 \wedge \dots \wedge q_n$$

can be re-written as a set of rules of the form:

$$p \rightarrow q_1, p \rightarrow q_2, \dots, p \rightarrow q_n$$

where the conjunction of literals  $q_1 \wedge q_2 \wedge \dots \wedge q_n$  is used to devise the set of new rules obtained by having the left-hand side  $p$  of the original rule to be the left-hand side of all new rules and taking each literal  $q_i$  as the right-hand side of the new rules. This makes it easier to detect long chains in the rule base of the production system. In large rule bases, long chains may result when one rule triggers a second rule, and the second rule triggers a third rule, and so on. In this case, a number of new rules representing a form of inference short-cuts can be added to the production system rule base. The original rule base of the production system is augmented with these inference short cut rules. These rules can be assigned a higher priority to guarantee their applications first during the inference process of the production system. For example, a rule base of the production system having a set of rules of the form:

$$P_1 \rightarrow P_2, P_2 \rightarrow P_3, \dots, P_j \rightarrow P_{j+1}, \dots, P_{n-1} \rightarrow P_n$$

can be augmented with a number of inference short-cut rules depending on the selected value of  $j$  representing the inference chain length. For example, if  $j$  is selected with a value of 3, then the rule base is augmented with the rules:

$$P_1 \rightarrow P_4, P_2 \rightarrow P_5, \dots, P_j \rightarrow P_{j+3}, \dots, P_{n-3} \rightarrow P_n$$

Again, these inference short-cut rules are assigned higher priorities than rules with the same conditions to guarantee their applications first and thus rule base ordering can be based on that.

Based on the improvements presented above, our approach can be presented in the following algorithm:

*Rule Base Quality Enhancement (Rule Base (RB), Augmented Rule Base (ARB), Inference Chain Length (J))*

*{Use a Verification Tool to Handle Anomalies (Inconsistencies, Contradictions, Redundancies, and Circularities) in RB*

*Re-write rules having actions/consequents consisting of conjunction of literals in the modified RB*

*Augment the modified RB with a number of inference short-cut rules based on the value of J and devise ARB}*

#### 4. Results and Performance Evaluation

To determine the performance of our approach, several simulations were performed for random rule bases with 50, 100, 150, and 200 rules and proving/answering a set

of 30 randomly generated goals. The same 30 randomly generated goals were answered using the original rule base without the enhancements suggested. The time taken to prove these goals was determined for both techniques and the times were plotted for different sizes of the rule bases as presented in **Figure 2**. As shown in **Figure 2**, our approach for proving the goals outperforms the traditional approach.

In addition, a rule base of 200 rules was tested for varying number of inference chain length of values 2, 3, 4 and 5 and proving/answering a set of 30 randomly generated goals. The same 30 randomly generated goals were answered using the original rule base without introducing the inference short-cut rules. The time taken to prove these goals was determined for both techniques and the times were plotted for different sizes of the rule bases as presented in **Figure 3**. As shown in **Figure 3**, our approach for proving the goals outperforms the traditional approach since our approach benefits from the inference short cuts and the re-ordering of the rule bases based on new inference short-cut rules' priorities.

#### 5. Discussion and Conclusion

In this paper, we proposed an approach for enhancing the

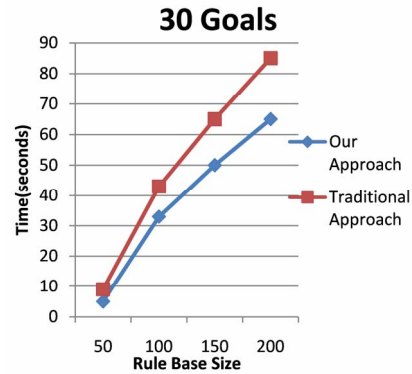


Figure 2. Performance evaluation of our approach using rule base size.

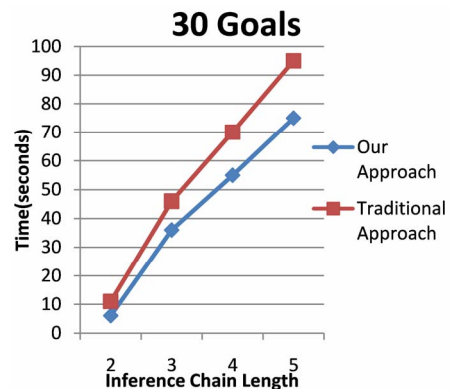


Figure 3. Performance evaluation of our approach using inference chain length.

quality of a rule base component of a production system which is of high importance to improve the performance of a production system. Production systems have a special value since they are used in state-space searching algorithms and expert systems in addition to their use as a model for problem solving in artificial intelligence. The simulations demonstrated that the improvement of rule base quality, in terms of augmenting the rule base with inference short-cut rules obtained after replacing rules with conjunctions of literals by a new set of equivalent rules in which long inference chains can be obtained easily, has a great impact on the performance of the production system as a whole.

## 6. Acknowledgements

The author thanks the reviewers for suggestions to improve the paper.

## REFERENCES

- [1] G. Luger, "Artificial Intelligence: Structures and Strategies for Complex Problem Solving," 6th Edition, Addison-Wesley, Boston, 2009, pp. 200-201.
- [2] W. Mustafa, "Improving Performance of Production Systems by Restructuring Facts," *Abhath Al-Yarmouk Journal of Natural Sciences and Engineering*, Vol. 11, No. 1A, 2002, pp. 105-120.
- [3] W. Mustafa, "Optimization of Production Systems Using Genetic Algorithms," *International Journal of Computational Intelligence and Applications*, Vol. 3, No. 3, 2003, pp. 233-248. [doi:10.1142/S1469026803000987](https://doi.org/10.1142/S1469026803000987)
- [4] N. Arman, "Generating Minimum-Cost Fault-Free Rule Bases Using Minimum Spanning Trees," *International Journal of Computing and Information Sciences*, Vol. 4, No. 3, 2006, pp. 114-118.
- [5] N. Arman, "Fault Detection in Dynamic Rule Bases Using Spanning Trees and Disjoint Sets," *The International Arab Journal of Information Technology*, Vol. 4, No. 1, 2007, pp. 67-72.
- [6] N. Arman, D. Richards and D. Rine, "Structural and Syntactic Fault Correction Algorithms in Rule-Based Systems," *International Journal of Computing and Information Sciences*, Vol. 2, No. 1, 2004, pp. 1-12.
- [7] S. Lukichev, "Improving the Quality of Rule-Based Applications Using the Declarative Verification Approach," *International Journal of Knowledge Engineering and Data Mining*, Vol. 1, No. 3, 2011, pp. 254-272. [doi:10.1504/IJKEDM.2011.037646](https://doi.org/10.1504/IJKEDM.2011.037646)
- [8] J. Ma, G. Zhang and J. Lu, "A State-Based Knowledge Representation Approach for Information Logical Inconsistency Detection in Warning Systems," *Knowledge-Based Systems*, Vol. 23, No. 2, 2010, pp. 254-272. [doi:10.1016/j.knosys.2009.05.010](https://doi.org/10.1016/j.knosys.2009.05.010)
- [9] N. Arman, D. Rine and D. Richards, "General Fault Detection Algorithms in Constrained Rule-Based Information Distribution Systems," *Yarmouk University Research Journal, Pure Science and Engineering Series*, Vol. 11, No. 2, 2003, pp. 190-203.