Developing Recursive Forward Chaining Method in Ternary Grid Expert Systems

Yuliadi Erdani

Politeknik Manufaktur Negeri Bandung, Indonesia

Summary
The performance of expert systems is determined by the performance of its main components. One of the mentioned components is the inference engine. It inferences the knowledge of expert systems and gives answers to users. Since the idea of ternary grid was issued in 2004, there is only a few developed methods, techniques, or engines working on ternary grid knowledge models. In 2010, an inference method of expert systems based on ternary grid was developed. The process of inference implements iterative forward chaining. The disadvantage of this method is low efficiency because of the high number of iterative processes. In order to improve the efficiency, a new method of forward chaining in ternary grid has been developed. It implements recursive processing. This paper describes the development of the inference engine of expert systems that can work in ternary grid knowledge models. The strategy to infer knowledge uses forward chaining with recursive processes. The design result is implemented in the form of software. The result of the experiment shows that the inference process works properly and more efficiently than the previous developed iterative forward chaining.

Key words: expert system, inference method, recursive method, ternary grid

1. Introduction
An expert system is a set of programs that manipulates encoded knowledge to solve problems in a specialized domain that requires human expertise. An expert system’s knowledge is obtained from expert sources and coded in a form suitable for the system to use in its inference or reasoning processes. The expert knowledge must be obtained from specialist or other sources of expertise, such as texts, journals, articles, and databases. This type of knowledge usually requires much training and experience in some specialized field such as medicine, geology, system configuration, or engineering design. Once a sufficient body of expert knowledge has been acquired, it must be encoded in some form, loaded into a knowledge base, then tested, and refined continually throughout the life of the system. Expert systems can perform some tasks that require expertise. Such tasks often have one or more of the following characteristics: the task may be difficult to specify, the task may have incomplete or uncertain data, there may not always be an optimum solution, the task cannot be solved in a step-by-step manner, and solutions are often obtained by using accumulated experience [9] [5] [14].

Expert systems can bring the following benefits [13]: they can preserve valuable knowledge that would otherwise be lost when an expert system is no longer available, they can allow an expert to concentrate on more difficult aspects of the task, they can enforce consistency, and they can perform dangerous tasks that would otherwise be carried out by humans.

One of known and very popular expert system types is the production rule. Production rules are simple but powerful forms of knowledge representation providing the flexibility of combining declarative and procedural representation for using them in a unified form. The term production rule came from production systems which were developed by [12]. A production system is a model of cognitive processing, consisting of a collection of rules (called production rules, or just productions). Each rule has two parts: a condition part and an action (conclusion) part. The meaning of the rule is that when the condition holds true, then the action is taken. A typical production rule is given below:

\[
\text{IF} \ (\text{mathematic score} \geq 60\%) \ \text{AND} \ (\text{physics score} > 58\%)
\]

\[
\text{THEN} \ \text{student passed the examination}
\]

The statement of the rule above means that a student can pass the examination if he/she has got a mathematical score more than or equal to 60% and a physics score more than 58%. Production systems or production rules provide appropriate structures for performing and describing search processes. A production system has four basic components as enumerated below [10]: A set of rules following the classical IF-THEN construct. If the conditions on the left-hand side are satisfied, the rule is fired, resulting in the performance of action on the right-hand side of the rule. A database of current facts established during the process of inference. A control strategy which specifies the order in which the rules are selected for matching of antecedents by comparing the facts in the database. It also specifies how
to resolve conflicts in selection of rule or selection of facts, and a rule firing module. An expert system that impalements production rule is known as rule-based expert system. In most rule-based expert system, building of rules can easily be done. Knowledge engineer does not have to do any work specifying rules and how they are linked to each other. Sometime the knowledge engineer can reference rules or facts that have not yet been created. It seems to be a simple and an instant work. The problem due to the performance of the knowledge will not occur until the number of rules is getting higher. Some problem may appear in the form of inconsistent rules, unreachable rules, redundant rule and closed rule chain of rules. In 2004 the solution to those problems were issued. It known as Ternary Grid [1][2][3]. Since the idea of ternary grid issued in 2004, there is no any developed inference method, technique or machine working on ternary grid knowledge model. As consequence of it, all ternary grid knowledge must be converted into production system format, so that the knowledge can be processed by rule-based inference machine to deliver solution. The inference engine of an expert system takes over the processing of the rules, which is called with rule-based expert systems also easily rule interpreters [7]. By the numerous methods of problem solution, which can be implemented in a rule interpreter, only the representatives of the concatenation strategies are to be treated here: forward chaining and backward chaining. Ternary grid knowledge model is an alternative solution to solve the mentioned problem. The developed inference machine of expert system can work in ternary grid knowledge model. The strategy to find solution uses forward chaining with iterative approach [6]. Due to the efficiency of algorithm, the recursive approach should be implemented in forward chaining process, which will be explained in this paper.

2. Method

The organisation of production rule can be easily represented in a Ternary Grid that has the following structure in Fig. 1.

![Ternary Grid basic structure](image)

Fig. 1. Ternary Grid basic structure

Ri: Rule i (i is the number of rule)
Fj: Fact j or logical term (j is the number of fact)

\[ i = \{1,2,3,...,I\} \]
\[ j = \{1,2,3,...,J\} \]
\[ J > I + 1 \]

The Value of every grid box is 0, 1 or 2
0 = unused, is represented by empty grid box.
1 = Fact Fm belongs to the condition part of rule Rn (LHS= Left Hand Side).
2 = Fact Fm is part of the conclusion part of Rn (RHS = Right-Hand Side).

![Ternary Grid as matrix](image)

Fig. 2. Ternary Grid as matrix

Ternary Grid can be considered as matrix. It is shown in the Fig. 2.

\[ a_{ij} = \{0,1,2\} \]

Value 0 is represented by empty matrix cell. The following value sets are needed for knowledge optimisation process:

The set of condition parts in row i is determined as follows:

\[ Ri1 = \{ j \mid a_{ij} = 1 \} \] (1)

The set of conclusion parts in row i is determined as follows:

\[ Ri2 = \{ j \mid a_{ij} = 2 \} \] (2)

The set of condition parts in column j is determined as follows:

\[ Fj1 = \{ i \mid a_{ij} = 1 \} \] (3)

The set of conclusion parts in column j is determined as follows:

\[ Fj2 = \{ i \mid a_{ij} = 2 \} \] (4)
The developed forward chaining method will be implemented in inference engine of expert system based on Ternary Grid. Inference engine of expert system is a computer program that answers questions from user. It processes all information from the knowledge base by firing rules and facts [11].

In the previous approach where iterative process was implemented, the complexity can be described as follows:

\[ n = \sum_i \sum_j x_{ij} \]  

Using recursive approach, the process of forward chaining can be illustrated as follows:

\[ R_{11}, R_{21}, R_{31}, R_{32}, R_{42} \] are already known. The inference process begins from fact \( R_{11} \) and move forward to its conclusion parts. From the fig. 3 above it can be seen that only one rule is applied, in which fact \( R_{42} \) as the conclusion part and fact \( R_{21} \) respectively \( R_{22} \) as the condition part of the rule. The same way of inference process can be also used in Ternary Grid. The developed algorithm of forward chaining in Ternary Grid is described as follows:

\[ F_{\alpha p F_{p q p R}} \subset \alpha \in \in \rightarrow \alpha, \quad (6) \]

The same data as [6] is used in this experiment. According to Ternary Grid acquisition technique [3], the mentioned rules are inputted into ternary grid knowledge base as it is shown in fig. 4. Using the developed concept, the rule-based format must not be converted into ternary grid. The inference process of the expert system in ternary grid uses forward chaining with recursive approach. All fact inputs are stored in set of facts \( F_k \). The inference engine searches all rules that are possible to be executed and stores them in set of rules \( R_x \) which.

\[ R_{ij} = \{ p | p \rightarrow q, p \in F_{ij}, q \in F, R_{ij} \subseteq F \} \]  

(6)

The inference engine determines then rules that are able to be applied and stores in the following set of rule \( R_{yn} \).

\[ R_{yn} \subseteq R_x \]  

(7)
To avoid duplication, the following formula are implemented in the inference engine

\[ R_z = \bigcup_n R_n \]

(8)

The result of inference process shows the effectiveness of the developed algorithm. In comparison to the method of [8] and [4][6], the developed inference method can work directly in ternary grid without having to be converted to rule-based format. In comparison to [6], the developed method work more efficient. The implemented recursive approach in inference process reduced the number of required looping.

The following data are taken from several conducted experiments

<table>
<thead>
<tr>
<th>Number of rule</th>
<th>Number of fact</th>
<th>Number of Loop</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>35</td>
<td>75</td>
</tr>
<tr>
<td>20</td>
<td>70</td>
<td>170</td>
</tr>
<tr>
<td>30</td>
<td>100</td>
<td>347</td>
</tr>
<tr>
<td>50</td>
<td>180</td>
<td>756</td>
</tr>
</tbody>
</table>

Figure 6 show the effort of recursion that is influenced by increasing the number of facts and rules.

4. Conclusion

The developed inference engine using recursive forward chaining method in ternary grid works properly. It can determine all facts that lead to rules which are possible to be applied. In comparison to the previous work using iterative approach, the developed method can reduce the number of looping significantly and works therefore more efficient. Referring to some literatures concerned expert systems, the developed method is novel and will give contribution in developing inference method of expert systems

Acknowledgments

I would like to express my gratitude to all those who gave me the possibility to complete this paper. The special thank goes to DP2M-DIKTI (DIT LIT/ABMAS DIKTI) that provides me with competence research grant (Hibah Kompetensi).
References


Dr. Ing., Yuliadi Erdani, M.Sc., Dipl. Ing., HTL received the B.S. degree in Electrical Engineering from the Ingenieurschule Burgdorf (ISB) Switzerland in 1995 and the M.S. degree in Computer Science and Communication Engineering (CSCE) from the University of Duisburg, Germany in 2002. He received PhD degree (Dr.-Ing.,) in Informationstechnik from the same university i.e. University of Duisburg-Essen, Germany in 2005. He did a lot of work with expert systems. He has been continuing his research work in the area of expert systems from 2006 until now and every year he always got research fund/grant. He works now as lecture in a state Polytechnic (university of applied science) in Bandung, Indonesia.

Dr. Ing., Yuliadi Erdani, M.Sc., Dipl. Ing., HTL received the B.S. degree in Electrical Engineering from the Ingenieurschule Burgdorf (ISB) Switzerland in 1995 and the M.S. degree in Computer Science and Communication Engineering (CSCE) from the University of Duisburg, Germany in 2002. He received PhD degree (Dr.-Ing.,) in Informationstechnik from the same university i.e. University of Duisburg-Essen, Germany in 2005. He did a lot of work with expert systems. He has been continuing his research work in the area of expert systems from 2006 until now and every year he always got research fund/grant. He works now as lecture in a state Polytechnic (university of applied science) in Bandung, Indonesia.