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 it main components. One of mentioned component is inference engine. It inferences the knowledge of expert system and gives answer to user. Since the idea of ternary grid issued in 2004, there is only several developed method, technique or engine working on ternary grid knowledge model. In 2010 an inference method of expert system based on ternary grid was developed. The process of inference implements iterative forward chaining. The disadvantage of mentioned method is low efficiency because of the high number of iterative process. In order to improve the efficiency, a new method of forward chaining in ternary grid has been developed. It implements recursive process. This paper describes the development of inference engine of expert system that can work in ternary grid knowledge model. The strategy to inference knowledge uses forward chaining with recursive process. The design result is implemented in the form of software. The result of experiment shows that the inference process works properly and more efficient in comparison to 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 manipulate encoded knowledge to solve problems in a specialized domain that normally 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, journal, articles, and database [9]. 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 task which requires 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 which would otherwise be lost when an expert system is no longer available, they can allow an expert to concentrate on more difficult aspect of the task, they can enforce consistency, and they can perform dangerous tasks which would otherwise be carried out by humans.

One of known and very popular expert system type is production rule. Production rule 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 system which is 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:

> *IF (mathematic score >= 60%) AND (physic score > 58%)*
> *THEN student passed the examination*

The statement of the rule above means that a student can pass the examination if he/she has got mathematic score more than or equal 60% and physic score more than 58%.

Production system or production rule provides appropriate structures for performing and describing search process. 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 rule are selected for matching of antecedents by comparing the facts in the database. It also specifies how

Manuscript received August 5, 2011
Manuscript revised August 20, 2011

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.

	Fj
Ri	0/1/2

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).

	$F1$	$F2$	...	Fj
$R1$	a_{11}	a_{12}		a_{1j}
...				
Ri	a_{i1}	a_{i2}		a_{ij}

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:

$$Ri1 = \{ j \mid a_{ij} = 1 \} \quad (1)$$

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

$$Ri2 = \{ j \mid a_{ij} = 2 \} \quad (2)$$

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

$$Fj1 = \{ i \mid a_{ij} = 1 \} \quad (3)$$

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

$$Fj2 = \{ i \mid a_{ij} = 2 \} \quad (4)$$

The developed forward chaining method will be implemented in inference engine of expert system based on Ternary Grid. Inference engine of expert system is 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:

$$n = \sum_{i} \sum_{j} x_{ij} \qquad (5)$$

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

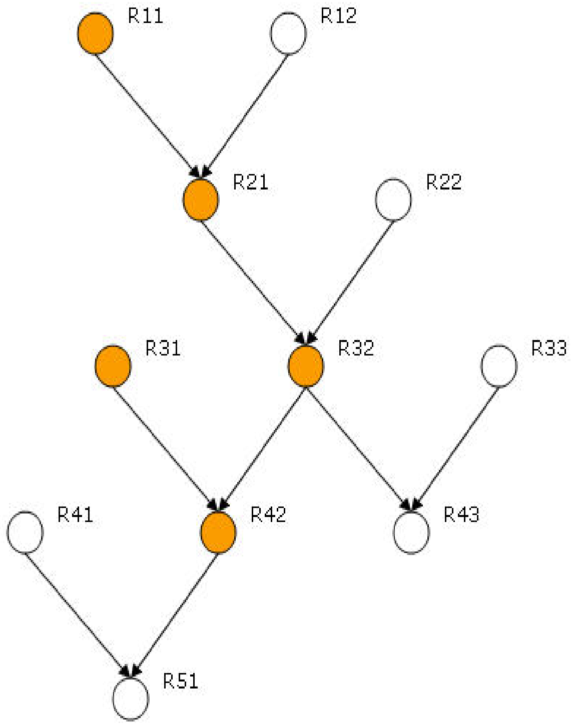

Fig. 3. Backward chaining illustration

Fact R11, R21, R31, R32 and R42 are already known. The inference process begins from fact R11 and move forward to it conclusion parts. From the fig. 3 above it can be seen that only one rule is applied, in which fact R42 as the conclusion part and fact R21 respectively R22 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:

```
Input known facts
Mark Rootnode
Searchnode = Rootnode
Searchbegin = 1

Do while there is applied rule or the queue is not
empty
    Search step
    Start search from rule Searchbegin according to a
    rule number x, which is applicable in the
    Searchnode and leads to an unknown Nextnode
    If there is applied rule then
        Continue search
        Store Nextnode in Mark
        Searchbegin = 1
    Else if the queue is not empty
        Change Searchnode
        Searchnode = Endnode
        of the first edge of the queue
        Searchbegin = 1
        Advance the queue

Result output
```

3. Result

The same data as [6] is used in this experiment.

IF <tutorial score >= 60> AND <thesis draft is finished> THEN <tutorial result passed>; IF < tutorial result passed> AND <presentation score greater than or equals 60> AND <comprehensive test passed> THEN <final project passes>; IF <final project passes> AND <thesis revision on time> THEN < graduation on time>; IF < final project passes > AND <thesis revision not on time > THEN <graduation postponed>

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 Fk. The inference engine searches all rules that are possible to be executed and stores them in set of rules Rx which.

$$R_x = \{ p \mid p \to q,\ p \in F_k, p \in F, F_k \subset F \} \qquad (6)$$

The inference engine determines then rules that are able to be applied and stores in the following set of rule *Ryn*.

$$R_{y_n} \subset R_x \qquad (7)$$

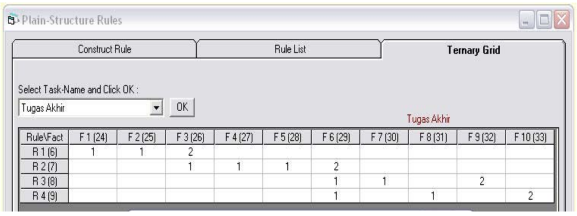

Fig. 4. given rules in ternary grid

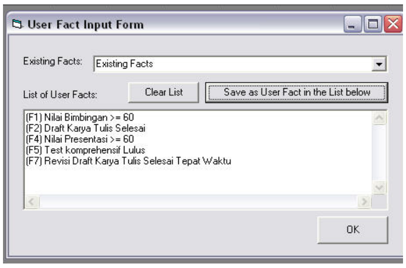

Fig. 5. known facts

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

$$R_z = \bigcup_n R_{y_n} \quad (8)$$

Fig. 6. Result of inference process

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

Tabel 13. Experiment data

Number of rule	Number of fact	Number of Loop
10	35	75
20	70	170
30	100	347
50	180	756

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

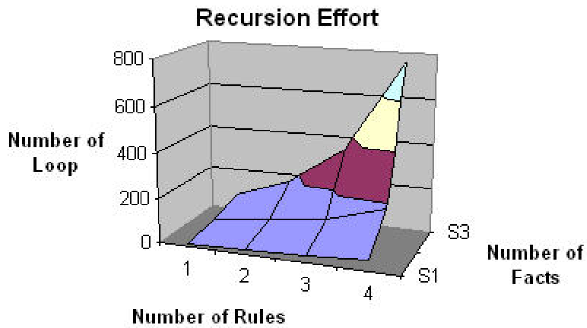

Fig. 6. Recursion effort

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 LITABMAS DIKTI) that provides me with competence research grant (Hibah Kompetensi.

References

[1] Erdani, Y., Hunger, A., Werner, S., Mertens, S., 2004. Ternary Grid as a Potentially New Technique for Knowledge Elicitation/Acquisition, 2nd IEEE Conference on Intelligent System, vol I: pp. 312-315. ISBN 0-7803-8278-1, Varna - Bulgaria, June 22-24, 2004. This paper has been accepted also by SEKE 04 (Software Engineering and Knowledge Engineering) conference in Banff, Canada.

[2] Erdani, Y., Hunger, A., Werner, S., 2005. Improving the Knowledge Performance using Ternary Grid Knowledge Acquisition and Model, WSEAS Transactions (Journals) on Information Science and Application, Issue 2, Volume 2, February 2005. ISSN 1790-0832

[3] Erdani, Yuliadi, 2005. Acquisition of Human Expert Knowledge for Rule-based Knowledge-based Systems using Ternary Grid, Verlag – Dissertation.de, Berlin, 2005, ISBN 3-89825-000-8

[4] Yuliadi Erdani. 2008. Konsep Inferensi pada Model Pengetahuan berbasis Ternary Grid, Prosiding Seminar Nasional Informatika – Universitas Pembangunan Nasional “Veteran”, 24 Mei 2008, Yogyakarta, ISSN 1979-2328

[5] Yuliadi Erdani. 2009. Konsep Pengembangan Expert System Shell berbasis Ternary”, Prosiding Seminar Nasional Industrial Service (SNIS 2009), Universitas Sultan Ageng Tirtayasa, Fakultas Teknik, Jurusan Teknik Industri, Pusdiklat PT. Krakatau Steel, 29 April 2009. ISBN 978-979-19280-0-7

[6] Yuliadi Erdani, 2010. Pengembangan Metode Inferensi untuk Sistem Pakar berbasis Ternary Grid, Jurnal P&PT Vol. VIII, No. 2, Desember 2010, ISSN 0854-5766

[7] David J.C. McKay. 2003. Information Theory, Inference, and Learning Algorithms. Cambridge University Press.

[8] Hunger, A., Werner, S., CONGA: A Course Online/ Offline Information and Guidance System to support an International Degree Course, Proceeding of ICCE 99 (Chiba-Japan, 1999, ISBN 1 58603 027 2, Page 577-583)

[9] Joseph C. Giarratano, Gary Riley. 2005. Expert Systems, Principles and Programming, ISBN 0-534-38447-1

[10] Krishnamoorthy, C. S., Rajeev, S., Artificial Intelligence and Expert Systems for Engineer, CRC Press, Boca Raton – Florid, 1996, ISBN 0-8493-9125-3

[11] Lunze, Prof. Dr.-Ing. Jan. 1994. Künstliche Intelligenz für Ingenieure, Band 1: Methodische Grundlagen und Softwaretechnologie, R. Oldenbourg Verlag, München Wien, ISBN 3-486-22287-2

[12] A. Newell and H.A. Simon. 1972. Human problem solving (Prentice Hall, Englewood Cliffs, NJ, 1972).

[13] Patterson, Dan W. 1990. Artificial Intelligence and Expert System, Prentice Hall, Englewood Cliffs, New Jersey, ISBN 0-13-477100-1

[14] Sascha Mertens, Marius Rosu, Yuliadi Erdani. 2004. An intelligent dialogue for online rule based expert systems, Proc. of the 2004 International Conference on Intelligent User Interfaces, January 13-16, 2004, Funchal, Madeira, Portugal. ACM 2004, ISBN 1-58113-815-6

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.