Developing Backward Chaining Algorithm of Inference Engine in Ternary Grid Expert System

Yuliadi Erdani
Politeknik Manufaktur Bandung
Bandung, Indonesia

***Abstract*— The inference engine is one of main components of expert system that influences the performance of expert system. The task of inference engine is to give answers and reasons to users by inference the knowledge of expert system. Since the idea of ternary grid issued in 2004, there is only several developed method, technique or engine working on ternary grid knowledge model. The in 2010 developed inference engine is less efficient because it works based on iterative process. The in 2011 developed inference engine works statically and quite expensive to compute. In order to improve the previous inference methods, a new inference engine has been developed. It works based on backward chaining process in ternary grid expert system.**

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 backward 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, dynamically and more efficient to compute in comparison to the previous developed methods.

Keywords- expert systems; ternary grid; inference engine; backward chaining.

I. INTRODUCTION

There is no official definition for the term of expert system but there are some descriptions for it created by people working in the field of expert system. With the term of expert system we refer to a computer system or a program, into which several procedures of artificial intelligence are integrated. Expert system can be understood as vehicles for Artificial Intelligence (AI) techniques.

Expert system is also applied artificial intelligence. Expert systems are programs for storing and processing knowledge of a special area, that why they are able to answer to questions and solve problems, with which experts normally deal [6]. In the current situation, expert system is an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution.

The expert knowledge must be obtained from specialist or other sources of expertise, such as texts, journal, articles, and database [8]. 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.

Some task that can be performed by expert system are difficult tasks to be specified, the task that 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 [11]. An example of applied expert system is web-based consultation system [1]. Benefit of expert systems is the ability to preserve valuable knowledge which would otherwise be lost when an expert system is no longer available. Expert system also can allow an expert to concentrate on more difficult aspect of the task. It 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 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 *there is a flame* ***THEN*** *there is fire*

The statement of the rule above means that fire is caused by a flame. If anything happens with a flame, it will lead to fire production. It is the idea of production system. The production system or production rule provides appropriate structures for performing and describing search process. A production system has four basic components as enumerated below [9]: 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 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.

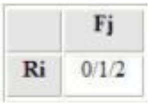
Building or construction of rules can easily be done in most rule-based expert system. Knowledge expert or knowledge engineer does not have to do any work specifying rules and how they are linked to each other. Sometime the knowledge expert or knowledge engineer can reference rules or facts that have 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.

The mentioned problems above have been solved with so called Ternary Grid [1][4][5]. Since the ternary grid can solve some problem concerning knowledge bottleneck, 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 interprets and evaluates the facts in the knowledge base in order to provide an answer. 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

The developed inference machine of expert system can work in ternary grid knowledge model. The strategy to find solution previously uses forward chaining with iterative approach [12]. As another alternative solution, the inference machine can be implemented by using backward chaining method. The emphasis of this paper is to describe the backward chaining method that is implemented in the inference engine of ternary grid expert system. The backward chaining method should bring more benefit than developed forward chaining method, e.g. dynamic answers of inference engine, efficient computation effort, etc.

II. Methods

Before talking the inference engine, we must first regard the knowledge representation. The Ternary Grid represents the production rule in the following structure (Fig. 1):



	Fj
Ri	0/1/2

Figure 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).

In the beginning of the development, the Ternary Grid was only used for knowledge acquisition system. The basic feature of the system architecture is to organize the independent and sequential obtaining process of the factual knowledge and the elicitation process of judgmental knowledge using Ternary Grid. The overall systems architecture is presented in terms of collection of functions providing effective acquisition, processing, transferring and flexible transformation of knowledge. This section gives an overview of the system that shows the design approach of the system and the concept of acquisition process.

The Ternary Grid acquisition system has task to organize the knowledge base, to obtain the factual knowledge, to elicit the judgmental knowledge, and to transfer the knowledge into knowledge base. The system was able to improve the performance of expert system knowledge [5]. Meanwhile the Ternary Grid has been used for other part related the expert system, such as knowledge representation, knowledge based system, inference engine, etc.

Even the Ternary Grid has been applied in an inference engine, but the approach of existing inference engine uses only forward chaining method. The backward chaining method has not been used in any inference engine. The developed backward 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 [9].

Backward chaining is a strategy of inference process which is the opposite of forward chaining. The strategy of backward chaining is started from a goal and ended with a fact that leads to the goal. Backward chaining method is also called as goal driven strategy of inference engine. In other literature, the backward chaining is a chaining process that begins with the last element in the chain and proceeds to the first element. This is often a very effective way of developing complex sequences of behavior

There are two search algorithms, which are normally used by backward chaining method, i.e. depth-first and breath-first search algorithm. Both algorithms search data in a tree structure. Depth-first search algorithm searches a data in a tree as deep as possible before backtracking. Breath-first search algorithm searches a data in neighbor nodes before it moves deeper to the bottom of the tree. Using depth-first search algorithm, the process of backward chaining can be illustrated as follows:

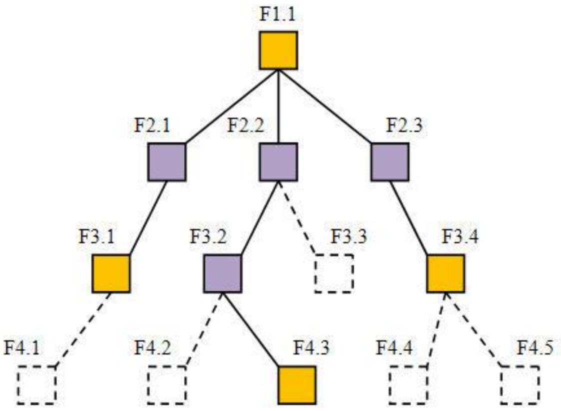


Figure 2. Backward chaining illustration

Figure 3 shows the illustration of backward chaining process of inference engine of expert system. The known facts are F1.1, F2.1, F3.2 and F3.4 in the beginning of process. The inference process begins from fact F1.1. That fact F1.1 is called as goal. The inference process moves then backward to other facts behind goal.

It is so called condition part of rule. The inference engine tries to applied fact F2.1, F2.2 and F2.3, etc. The developed backward chaining uses depth-first search algorithm. The following steps describe the inference process of expert system using developed backward chaining algorithm as follows:

Goal: F1.1

Fetch: F2.1 → unknown

Fetch: F3.1 → known

Rule (F3.1, F2.1) → fired

Rule (F2.1, F1.1) → fired

Fetch: F4.1 → unknown

Fetch: F2.2 → unknown

Fetch: F3.2 → unknown

Fetch: F4.2 → unknown

Fetch: F4.3 → known

Rule (F4.3, F3.2) → fired

Rule (F3.2, F2.2) → fired

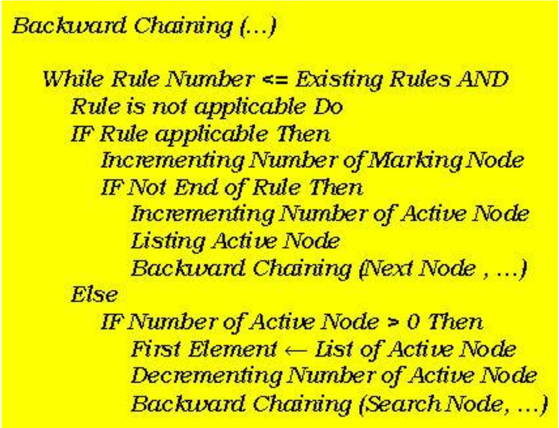
Rule (F2.2, F1.1) → fired

…

Etc.

The inference process continues until all possible facts have been asked (tested) to be fired. The developed algorithm searches all data deeper into the bottom of tree structure in the knowledge base of expert system.

The designed and implemented backward chaining algorithm is explained in the following algorithm:



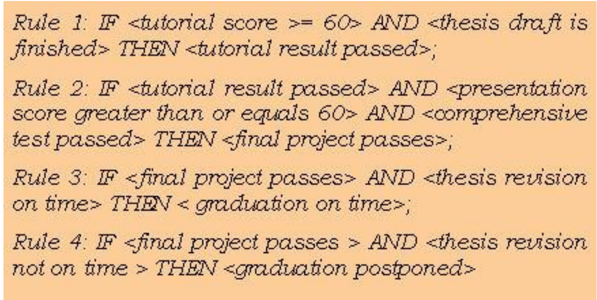
```
Backward Chaining (…)

  While Rule Number <= Existing Rules AND
    Rule is not applicable Do
    IF Rule applicable Then
      Incrementing Number of Marking Node
      IF Not End of Rule Then
        Incrementing Number of Active Node
        Listing Active Node
        Backward Chaining (Next Node , …)
    Else
      IF Number of Active Node > 0 Then
        First Element ← List of Active Node
        Decrementing Number of Active Node
        Backward Chaining (Search Node, …)
```

The process continues as far as the number of rules is less than the number of existing rules and there is rule that is not applicable. If a rule is applicable (fired) then the program search the next facts that lead to applicable rule until there is no fact found anymore. If a rule is not applicable (fired) then the program search other possible rule in other paths. The process continues until all possible facts have been tested or fetched.

III. RESULTS

The same data as [12] [13] is used in this experiment.



Rule 1: IF <tutorial score >= 60> AND <thesis draft is finished> THEN <tutorial result passed>;

Rule 2: IF <tutorial result passed> AND <presentation score greater than or equals 60> AND <comprehensive test passed> THEN <final project passes>;

Rule 3: IF <final project passes> AND <thesis revision on time> THEN < graduation on time>;

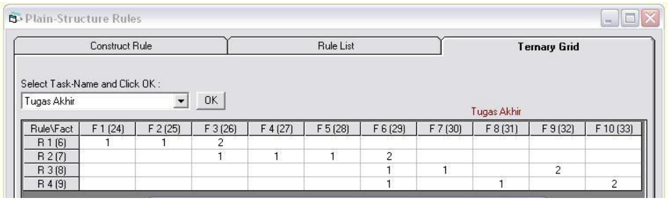
Rule 4: IF <final project passes > AND <thesis revision not on time > THEN <graduation postponed>

According to Ternary Grid acquisition technique [5], the mentioned rules are inputted into ternary grid knowledge base as it is shown in figure 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 backward 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:

$$R_x = \left\{ p \,\middle|\, p \to q,\ p \in F_k, p \in F, F_k \subset F \right\} \qquad (5)$$

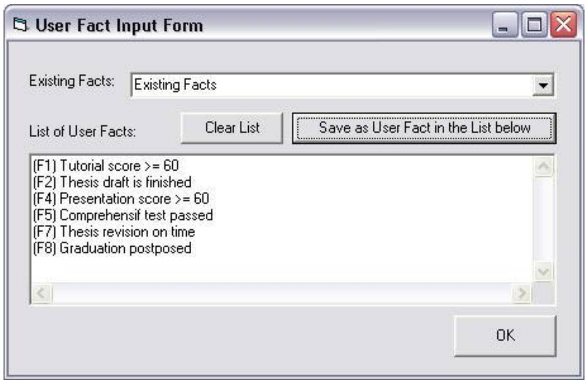
The inference engine determines then rules that are able to be applied and stores in the following set of rule Ryn.

$$R_{y_n} \subset R_x \qquad (6)$$



Rule\Fact	F 1 (24)	F 2 (25)	F 3 (26)	F 4 (27)	F 5 (28)	F 6 (29)	F 7 (30)	F 8 (31)	F 9 (32)	F 10 (33)
R 1 (6)	1	1	2							
R 2 (7)			1	1	1	2				
R 3 (8)						1	1		2	
R 4 (9)						1		1		2

Figure 3. Given facts and rules in ternary grid



(F1) Tutorial score >= 60
(F2) Thesis draft is finished
(F4) Presentation score >= 60
(F5) Comprehensif test passed
(F7) Thesis revision on time
(F8) Graduation postposed

Figure 4. Kknown facts

The application does then the inference task by processing all facts that are given before. The result of inference process can be shown in figure 6.

Inconsistent rules can be detected and eliminated using the following processes:

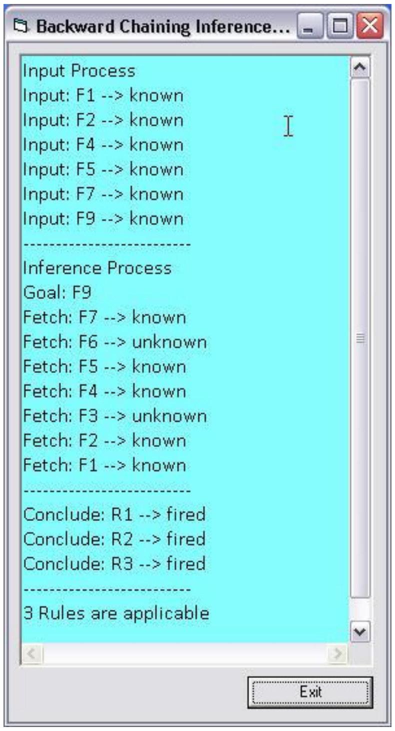
- Find rows, in which value 3 appears:

$$B = \left\{ i \mid a_{ij} = 3 \right\} \qquad (7)$$

- Remove row duplication

$$C = \bigcup_{b \in B} b \qquad (8)$$

The result of inference process shows the effectiveness of the developed algorithm. In comparison to the method of [7] [12] and [13], the developed inference method can work directly in ternary grid without having to be converted to rule-based format. In comparison to [12] and [13], the developed method work more dynamic and efficient to compute. The implemented recursive approach in inference process reduced the number of required iteration. The result of inference process can also show other facts that weren't known before. These all facts could lead to give more conclusions that will bring more information to expert system.



```
Input Process
Input: F1 --> known
Input: F2 --> known
Input: F4 --> known
Input: F5 --> known
Input: F7 --> known
Input: F9 --> known
-------------------------
Inference Process
Goal: F9
Fetch: F7 --> known
Fetch: F6 --> unknown
Fetch: F5 --> known
Fetch: F4 --> known
Fetch: F3 --> unknown
Fetch: F2 --> known
Fetch: F1 --> known
-------------------------
Conclude: R1 --> fired
Conclude: R2 --> fired
Conclude: R3 --> fired
-------------------------
3 Rules are applicable
```

Figure 5. Result of inference process using backward chaining algorithm

The following data are taken from several conducted experiments

TABLE I. EXPERIMENT DATA

Number of rule	Number of fact	Number of Iteration
10	35	62
20	70	134
30	100	295
50	180	673

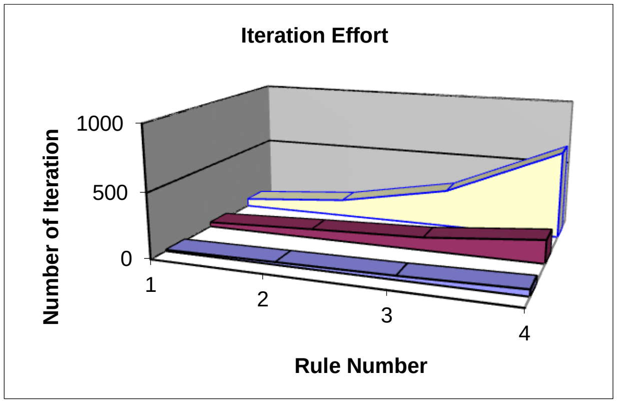


Figure 6. Recursion effort

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

IV. CONCLUSION

The developed inference engine using backward chaining method in ternary grid works properly. It can determine all applied rules that lead to the goal fact. In comparison to the previous work using iterative approach and forward chaining algorithm, the developed method works more dynamic and more efficient to compute. The inference process could also detect and lead to other rules that previously unknown and brought new solutions. Referring to some literatures concerned expert systems; the developed method is novel and will give contribution in developing inference method of expert systems.

REFERENCES

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

[2] Erdani, Y., Hunger, A., Werner, Stefan., Mertens, S. Web-Based Consultation System with Expert System, IASTED CST 2003 – International conference (International Association of Science and Technology for Development – Computer Science and Technology), May 19-21, 2003, Cancun, Mexico, 2003, ISBN – 0-88986-349-0, page 61-64

[3] Erdani, Y., Hunger, A., Werner, S., Mertens, S. 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.

[4] Erdani, Y., Hunger, A., Werner, S., 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

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

[6] Horn, Christian; Kerner, Immo O. : Lehr- und Übungsbuch Informatik, Band 3. Fachbuchverlag Leipzig, im Carl Hanser Verlag, München Wien, 1997. ISBN 3-446-18699-9

[7] 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)

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

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

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

[11] Sascha Mertens, Marius Rosu, Yuliadi Erdani. 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

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

[13] Yuliadi Erdani, Developing Recursive Forward Chaining Method in Ternary Grid Expert Systems, IJCSNS (International Journal of Computer Science and Network Security), Vol. 11 No. 8, August 2011, ISSN 1738-7906

AUTHORS PROFILE

Dr. Ing., Yuliadi Erdani, M.Sc., Dipl. EL. 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.