Reduction Processing Time in Rule-Based Expert System Using Rete Algorithm

Tin Ma Ma, Myint Myint Yee
University of Computer Studies, Yangon
tinmama1992@gmail.com

Abstract

Nowadays, many Rule-Based Expert Systems have been proposed and almost all of them assume "IF-THEN" rules. The process of matching rules is performed by the data-driven approach, Forward Chaining method, in the Rule-Based Expert System. As the rules and data grow, the time complexity of Forward Chaining method is exponential. Because of this exponential complexity, the system will perform quite slowly. So reducing the processing time plays the important role in Forward Chaining method. The Rete algorithm provides for reducing the processing time. This algorithm drops the time complexity of Forward Chaining to linear rather than exponential. It can improve the performance of the system. This paper applies a Rule-Based Expert System for helping to choose the specific doctors according to the diseases. By applying this system in android phones, the users can be easily found the specific doctors in time.

1. Introduction

Everywhere you go, people are using mobile phones to keep in touch with family and friends, to find a nearby restaurant, or to check the latest news headlines, or to know information about physicians, hospitals and clinics. Their phones have become an indispensable part of their lives with applications that bind them close to each other and the world around them. Nowadays, there are so many physicians, hospitals and clinics. When people suffer from diseases, they need to find the information about the physicians concerned with their diseases. The expert system is used to show information about physicians, hospitals, diseases and schedule of physicians. An expert system is a set of programs that manipulate encoded knowledge to solve problems in a specialized domain that normally requires human expertise. If-then rules are one of the most common forms of knowledge representation used in expert system. Systems employing such rules as the major representation paradigm are called rule-based expert systems. The rule-based expert system uses a simple technique: it starts with a rule-base, which contains all of the appropriate knowledge encoded into If-Then rules, and a working memory, which may or may not initially contain any data, assertions or initially known information. In the Rule-Based Expert System, the Forward Chaining method checks the rule condition against the data in working memory but the processing time of Forward Chaining method is expensive. To reduce the processing time, this paper uses Rete algorithm. The Rete algorithm reduces the time complexity of Forward Chaining method by reducing the number of comparisons between rule conditions and data in the working memory.

In this paper, we propose the user to choose the suitable doctors for his/her disease in which hospital on what time of the day easily and describe how to apply the Rule-Based Expert System that uses the knowledge about the doctors, hospitals, diseases and schedules in mobile phones.

The rest of this paper is organized as follows. Section 2 describes the related work. In Section 3, we discuss the background theories that are used. In Section 4, system design is presented. In Section 5, case study is described. In Section 6, the experimental results are shown. Finally, Section 7 concludes the paper.

2. Related Work

Tomoya Kawakami, Naotaka Fujita, Tomoki Yoshihisa proposed a rule-based Home Energy Management System (HEMS) using Rete Algorithm [7]. In this paper, rules for managing energy are processed by smart taps in network and the loads for processing rules and collecting data are distributed to smart taps. The number of processes and collecting data are reduced by processing rule-based on the Rete algorithm. They evaluated the proposed system by simulation. In the simulation environment, rules are proposed by a smart tap that relates to the action part of each rule, they implemented by proposed system as HEMS using smart taps. M. Veera Narayana, A. Sunil Kumar, B. Suneel Kumar, N. Samkishao, B. Jogeswara Rao focus on the Rete Algorithm to support disjunctive relationship using 'or' connectivity and use the shadow proxy mechanism to update objects in the working memory[3]. In this paper, the system works on the mechanism of rule-based system and the Rete Algorithm. This system is a web based application for online users with JSP as front end and MYSQL as backend. Christoph F. Eick proposed "Historical Rete Networks to support the Debugging of Forward Chaining Rule-based Programs" [1]. It presents how a forward chaining program run's "historical" details can be stored in its Rete inference network, used to match rule conditions to working memory. This system can be done without seriously affections the network's run time performance. It can be used during debugging, and a debugging tool, that incorporates these techniques.

3. Background Theory

In this section, we describe Expert System, Rule-Based Expert System, Methods of the Rule-Based Expert System and Rete Algorithm.

3.1 Expert System

An expert system is a computer system that emulates the decision-making ability of a human expert, i.e. it acts in all respects as a human expert. Expert systems have emerged from early work in problem solving, mainly because of the importance of domain-specific knowledge. The expert knowledge must be obtained from specialists or other sources of expertise, such as texts, journal articles, and data bases. Expert system receives data from the user and provides expertise in return. The user interacts with the system through a user interface, constructed by using menus, natural language or any other style of interaction. The rules collected from the domain experts are encoded in the form of knowledge base. The inference engine may infer conclusions from the knowledge base and the data supplied by the user [3] [4]. An expert system based on a collection of rules that a human expert would follow in dealing with a problem and it implements production rule is known as rule-based expert system.

3.2 Rule-Based Expert System

The Rule-Based Expert System uses a simple technique: it starts with a rule-base, which contains all of the appropriate knowledge encoded into If-Then rules, and a working memory, which may or may not initially contain any data, assertions or initially known information. The system examines all the rule conditions (IF) and determines a subset, the conflict set, of the rules whose conditions are satisfied based on the working memory. Of this conflict set, one of those rules is triggered (fired). When the rule is fired, any actions specified in its THEN clause are carried out. These actions can modify the working memory, the rule-base itself, or do just about anything else the system programmer decides to include. This loop of firing rules and performing actions continues until one of two conditions is met: there are no more rules whose conditions are satisfied or a rule is fired whose action specifies the program should terminate [2].

3. 2.1 Methods of Rule-Based Expert System

There are two methods of Rule-Based Expert System, Forward Chaining and Backward Chaining.

Forward Chaining: An inference engine using forward chaining searches the inference rules until it finds one where the IF clause is known to be true. When found it can conclude, or infer, the THEN clause, resulting in the addition of new information to its dataset. In other words, it starts with some facts and applies rules to find all possible conclusions. Therefore, it is also known as Data Driven Approach [5] [6].

Backward Chaining: An inference engine using backward chaining would search the inference rules until it finds one which has a THEN clause that matches a desired goal. If the IF clause of that inference rule is not known to be true, then it is added to the list of goals (in order for goal to be confirmed it must also provide data that confirms this new rule). In other words, this approach starts with the desired conclusion and works backward to find supporting facts. Therefore, it is also known as Goal-Driven Approach [6].

Of the two methods available, forward-chaining or backward-chaining, the one to use is determined by the problem itself. If the 'average' rule has more conditions than conclusions, that is the typical hypothesis or goal (the conclusions) can lead to many more questions (the conditions), forward-chaining is favored. If the opposite holds true and the average rules have more conclusions than conditions such that each fact may fan out into a large number of new facts or actions, backward-chaining is ideal [2].

.

4. System Design

The Proposed System is written in Android Development Tools (ADT) which is a plugin for Eclipse Integrated Development Environment (IDE). The Proposed System is applied in all kinds of mobile phones with android version 4.2.2. This also uses android programming language and SQLite Database. The Proposed System uses the data set of 786 doctors in 15 hospitals. The information of 786 doctors in 15 hospitals is manually collected by the information center of hospitals and pamphlets of those hospitals.

4.1 System Flow of Proposed System

In Figure 1, the system flow diagram of Proposed System is presented. Firstly, the Proposed System accepts the user input. And then checks the rete-list which stores a list of rules matched or partially matched by the working memory, is empty or not. If the rete-list is empty, this system uses the Forward Chaining Algorithm to match rule. If not, the Proposed System uses the Rete Algorithm to match rule. When the Forward Chaining Algorithm is used, this Forward Chaining Algorithm checks the rule match or not. If the rule matches, the Proposed System adds the matched rule to the rete-list and shows the relevant information to the user. If not, ask the user to question again. When the Rete Algorithm is used, the Rete Algorithm checks the rule match or not. If the rule matches, the Proposed System shows the relevant information to the user. If not, the Proposed System asks the user to question again.

4.2 Algorithms Use in Proposed System

The Main Algorithm of Proposed System (shown in figure 2) uses two algorithms, Forward Chaining Algorithm (shown in figure 4) and Rete Algorithm (shown in figure 3). The Main Algorithm checks the rete-list which stores a list of rules matched or partially matched by the working memory, empty or not. At first, the rete-list is empty; the Main Algorithm calls the Forward Chaining Algorithm. Next time, the Main Algorithm calls the Rete Algorithm.

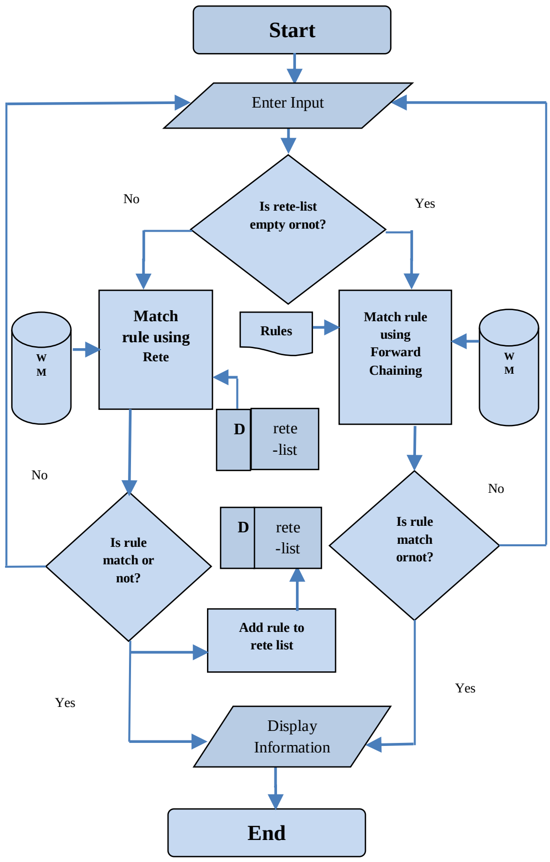

Figure 1. System Flow Diagram

```
Algorithm: Main
Input: Knowledge Base KB, input
Let rete-list: List of rule matched or partially matched, initially null
Output: output
Begin
    If rete-list is not empty then
        If Rete() returns true then
                display output;
        Else
            If ForwardChaining() return true then
                    display output;
            Else
                    display "Not Match";
            Endif
        Endif
    Else
            If Forwardchaining() return true then
                    display output;
            Else
                    display "Not Match";
            Endif
    Endif
End
```

Figure 2. Main Algorithm

Initially, the Forward Chaining Algorithm assigns three list that is agenda, clause and count. Agenda list stores the symbol (i.e, doctor, hospital, specialist, emergency, day and start-period). Clause list stores the clause, which is the logical formula of a particular rule-like form (i.e doctor&hospital=>R1; doctor&specialist=>R2;doctor&specialist&hospital=>R3;doctor&hospitalday=>R4;doctor&hospital&day&start-period=>R5;doctor&specialist&start-period=>R6;emergency=>R7;). Count list stores the number of premises which is the part of the rule after

IF as like doctor and hospital, doctor and specialist, etc.

The Forward Chaining Algorithm is processed by the following steps:

Step1: This algorithm processes until the agenda list is empty.

Step2: The premise p is assigned the first item of the agenda list.

Step3: The following processes do until the premise p does not appear in each clause of the clause list.

> **Step3.1:** This algorithm decrements the count list.
>
> **Step3.2**: If the count list is equal to zero, the Forward Chaining Algorithm checks a clause is matched with the user input by the working memory. If it is matched with the user input, the matched clause is added to the rete-list, returns Boolean true and this algorithm terminates. If not, the head of a clause is added to the agenda list.

The Forward Chaining Algorithm processes the above steps until the agenda list is empty. When the agenda list is empty, this Forward Chaining Algorithm returns Boolean false and terminates.

The Rete Algorithm is processed by the following steps:

Step1: The Rete Algorithm matched the rule with the user request in the rete-list.

Step2: If the rule matched with the user request, returns Boolean true. If not, returns Boolean false.

If the Forward Chaining Algorithm returns Boolean true, the Main algorithm displays the output as well as the Rete Chaining Algorithm returns Boolean true.

```
Algorithm: Rete
Input: Knowledge Base KB, input
Let rete-list: List of rule matched or partially matched,
Output: Boolean true or false
Begin
        Search input in rete-list
        If input exists in rete-list then
                return true;
        Else
                return false;
        Endif
End
```

Figure 3. Rete Algorithm

```
Algorithm: ForwardChaining
Input: Knowledge Base KB, input
Let agenda: a list of symbol
    clause: a list of clause
    count: a list of integer, number of premises
Output: Boolean true or false
Begin
    While agenda is not empty do
    {
      p =agenda.remove(0);
      for each clause c in whose premise p appears do
      {
        decrement count[c];
        If count[c] = 0 then do
            If HEAD[c]= ask then
                rete-list=c;
                return true;
            Endif
            agenda.add(Head[c]);
        Endif
      }
      End for
    }
    End while
    return false;
End
```

Figure 4. Forward Chaining Algorithm

4.3 Defined Rules for Proposed System

The following rules which used in inferencing must define in the knowledge base:

1: IF doctor and hospital
THENRule 1;
2: IF doctor and specialist
THEN Rule 2;
3: IF doctor and specialist and hospital
THEN Rule 3;
4: IF doctor and hospital and day
THEN Rule 4;
5: IF doctor and hospital and day and start-period
THEN Rule 5;
6: IF doctor and specialist and start-period
THEN Rule 6;
7: IF emergency
THEN Rule 7;

One of the rules is fire, the proposed system display as follows:

IF Rule 1 Then
Display detailed information of doctor according to the hospital
IF Rule 2 Then
Display detailed information of doctor according to the specialist
IF Rule 3 Then
Display detailed information of doctor according to the specialist and the hospital
IF Rule 4 Then
Display detailed information of doctor according to the day
IF Rule 5 Then
Display detailed information of doctor according to the day and start-period
IF Rule 6 Then
Display detailed information of doctor according to the start-period
IF Rule 7 Then
Display detailed information of emergency department

5. Case Study

For example, the user wants to search "Surgeon Moe Moe Tin is a doctor of Bahosi hospital".

Firstly the Proposed System savesthe defined rules (described in section 4.3) in knowledge base as a form (doctor&hospital=>R1;doctor&specialist=> R2;doctor&specialist&hospital=>R3;doctor&hospital day=>R4;doctor&hospital&day&start-period=>R5; doctor&specialist&start-eriod=>R6;emergency=>R7; doctor;hospital;specialist;day;start-period; emergency;). Secondly, the Proposed System accepts the user input that is doctor, specialist and hospital. The Proposed System checks the rete-list. At first, the rete-list is empty; the Proposed System uses the Forward Chaining Algorithm.

The Forward Chaining Algorithm processes until the agenda list is empty. First, this algorithm removes the first item of agenda list and assigns this value to the premise p (i.e p=doctor). And then this algorithm searches the premise p in each clause. The premise p appears in a clause (doctor&hospital=>R1) and then decrements the count list. The count list is equal to zero but the rule is not matched with the request. The head of clause (R1) is added to the agenda list. The Forward Chaining Algorithm processes the above processes until the agenda list is empty or the head of clause and the user input is matched by the working memory. The Forward Chaining Algorithm returns Boolean true because of the third rule. When the Forward Chaining Algorithm returns Boolean true, the Proposed System adds the third rule to the rete-list and displays the detailed information of Surgeon Moe Moe Tin (shown in figure 5).

Next Time, the user want to search "Surgeon Moe Moe Tin is a doctor of Bahosi hospital" again. This time, the rete-list is not empty; the Proposed System uses the Rete Algorithm. The Rete algorithm searches the user input in the rete-list. The user input matches with the rule by the working memory in the rete-list and then returns Boolean true.

When the Rete Algorithm returns Boolean true, the Proposed System displays the detailed information of Surgeon Moe Moe Tin (shown in figure 5). If the Rete Algorithm returns Boolean false, the Proposed System uses the Forward Chaining Algorithm to match the rule again.

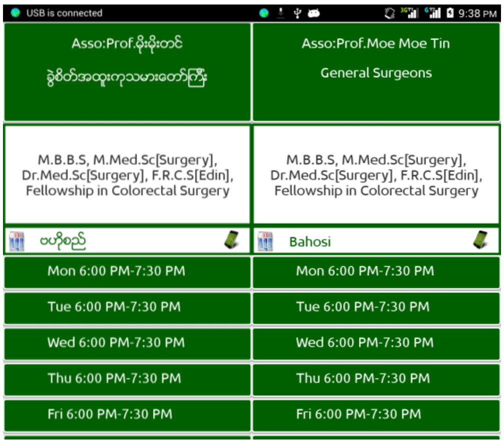

Figure 5. Display Result when Third Rule is fired

6. Experimental Results

The experiment for the Proposed System uses 786 data in Doctor Table, 15 data in Hospital Table, 31 data in Specialist Table, 20 data in Emergency Table and 4644 data in Schedule Table. The Proposed System runs 20 times of each matching process and shows the average milliseconds of the processing time. In this section, some of the experimental results are shown.

We illustrated the processing times of the Forward Chaining Algorithm and the Main Algorithm that is matched the Rule 1 in figure 6.

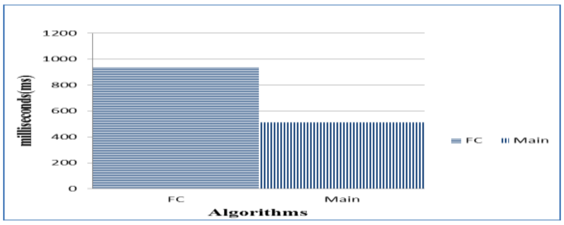

Figure 6. Processing times of Forward Chaining Algorithm and Main Algorithm matched with the Rule 1

In Figure 7, the processing times of the Forward Chaining Algorithm and the Main Algorithm that is matched with the Rule 1, 2, 3 and 4 are illustrated.

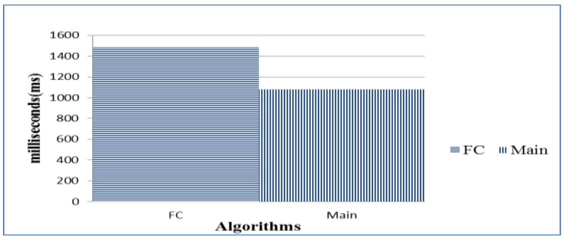

Figure 7. Processing times of Forward Chaining Algorithm and Main Algorithm matched with the Rule 1, Rule 2, Rule 3 and Rule 4

In Figure 8, the processing times of the Forward Chaining Method and the Main Algorithm that is matched with the Rule 1, 2, 3 ,4, 5, 6, 7 and 8 are illustrated.

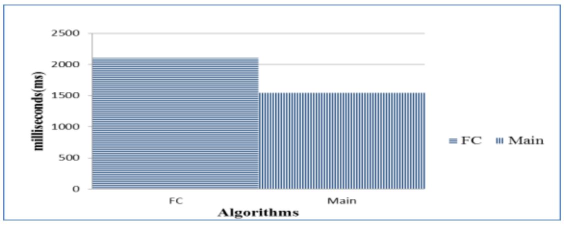

Figure 8. Processing times of Forward Chaining Algorithm and Main Algorithm matched with the Rule 1, Rule 2, Rule 3, Rule 4, Rule 5, Rule 6, Rule 7 and Rule 8

According to the results, the processing time of the Main Algorithm is faster than the processing time of the Forward Chaining Algorithm. Therefore, the processing time of Forward Chaining Algorithm can be reduced by applying the Main Algorithm in Proposed System.

7. Conclusion

This Proposed System provides to know the knowledge of doctors and hospitals as an expert. In this system, it decides how to make rules by using Rule-Based Expert System. This also uses the Rete Algorithm to reduce the time complexity of the Forward Chaining Algorithm which is an algorithm of Rule-Based Expert System. Therefore, by applying the Proposed System in mobile phones, the user can search the doctors who will be in which hospital on what time of the day easily.

References

[1] Christoph F. Eick proposed “Historical Rete Networks to support the Debugging of Forward Chaining Rule-based Programs”, Journal on Artificial Intelligence Tools 2(1), pp.47-70, Apirl 1993.

[2] https://cis.k.hosei.ac.jp/~rhuang/Miccl/AI-2/L9-src/ai-depot_decisionTree1.pdf

[3] M. Veera Narayana, A. Sunil Kumar B. Suneel Kumar, N.Samkishan, B. Jogeswara Rao, “Implementation of RETE Algorithm Using Lemon Expert”, International Journal of Emerging Research in Management&Technology, ISSN: 2278-9359 (Volume-2, Issue-7), July 2013.

[4] M.S. Prasad Babu, N.Thirupathi Rao, “Implementation of Parallel Optimized ABC Algorithm with SMA Technique for Garlic Expert Advisory System”, International Journal of Computer Science, Engineering and Technology (IJCSET), Volume 1, Issue 3, October 2010.

[5] R. C. Chakraborty, “Expert Systems”, Artificial Intelligence, June 01, 2010.

[6] Tilotma Sharma, Navneet Tiwari and Deepali Kelkar, “Study of Difference between Forward and Backward Reasoning”, International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com(ISSN 2250-2459, Volume 2, Issue 10, October 2012).

[7] Tomoya Kawakami, Naotaka Fujita, Tomoki Yoshihisa, Masahiko Tsukamato, “An Evaluation and Implementation of Rule-Based Home Energy Management System Using Rete Algorithm”, 20 July 2014.