A Rete Rule Reasoning Algorithm Based on the Audit Method Ontology

Gang Liu[1,a], Shaobin Huang[1,b], Duan Zhang[2,c] and Yudan Du[1,d]

[1]*Harbin Engineering University*
[2]*Harbin Resident Audit Office of CNAO*
[a]*liugang@hrbeu.edu.cn* ,[b]*huangshaobin@hrbeu.edu.cn,* [c]*zdbaggio@sina.com,*
[d]*duyudan@hrbeu.edu.cn*

Abstract

For the application background on a guidance of audit information resource, this paper constructs the ontology of audit methods which is suitable for medical insurance in audit field, and provides information basis for the ontology-based rule reasoning, through utilizing the collected relevant information of audit methods in actual work of medical insurance. Then, the paper mainly researches the rule reasoning algorithm—forward rule rapid matching Rete algorithm in expert system. Aiming at the existing problems in Rete algorithm, the paper introduces the improved method of active hash. Finally, the paper verifies that the improved Rete algorithm can improve implementation efficiency, and has practical significance in the intelligent retrieval system.

Keywords: *ontology-based rule reasoning, Rete, expert system, intelligent retrieval, audit method*

1. Introduction

With the development and progress of the field about computer technology, especially the Internet technology, people could share information resources through the Internet in the whole world, quickly, which makes the Internet information resources to increase constantly and become very rich. So rapidly and effectively access to information resources has become an important force to promote social progress. However, to find and obtain useful information isn't an easy thing due to the rapid growth of information in the large number of information resources. Therefore, how to quickly access the information resources in a particular area has become an important and meaningful issue in order to achieve the sharing and utilization of information resources.

The traditional information retrieval technology is based on the keyword matching, lacking semantic information. So it introduces the semantic and semantic processing in intelligent retrieval process. In the study of natural language, semantics is the people's know or feel for objective things. However, due to the weak ability of the computer in understanding and dealing with natural language, it can only deal with and understand the symbolic logic expressions of formalization. Therefore, it is using logic symbols to represent the concepts and logical relationships among each concept in natural language that converts into the semantic which the computer can understand and deal with. Through the formal description and an abstract representation of a concept, intelligent semantic retrieval, which understands the real query target of users, uses the semantic relations between the concepts of computer analysis to find knowledge that is hidden in the information resource and accord with the

ISSN: 1738-9968 IJHIT

intention of users. In the network era of rich information, the rapid and efficient query and access to the needed information requirements become more and more strong, therefore, to realize the intelligent retrieval becomes more realistic.

At present, the demand for a complete retrieval system of social insurance audit field is also more obvious. In the field of social insurance audit, audit information resources of various types (including the audit methods, the audit results, policies and regulations, etc.) in the audit plays an important role, in past auditors use these resources in the manual search ways in the audit process, and the efficiency is low. The professional information guidance system, which is researched and designed for social insurance audit field, by establishing the relationship between audit methods and domain ontology, regards the audit method as a service to realize the specific audit target, recommending related audit methods.

2. Background and description

To realize the semantic retrieval is without the understanding and treatment of semantics, and semantic analysis can’t be separated from knowledge. Therefore, building knowledge base is the necessary condition to realize semantic retrial. In recent years, the application of ontology in the information retrieval system is more and more wide. As the description of system on domain knowledge conceptualization, domain ontology defines the term in the field, describes the relationships among concepts, provides the logic rules for semantic reasoning, and can provide powerful support for the reasoning in the retrieval process [1]. In semantic retrieval system based on ontology knowledge base, semantic analysis and reasoning orients concept, relation and instance (metadata). The entire semantic retrieval process is to make the computer to recognize and understand the structure and metadata information of the domain ontology, and obtains explicit results even tacit knowledge process in the information base according to relevant logic rule [2].

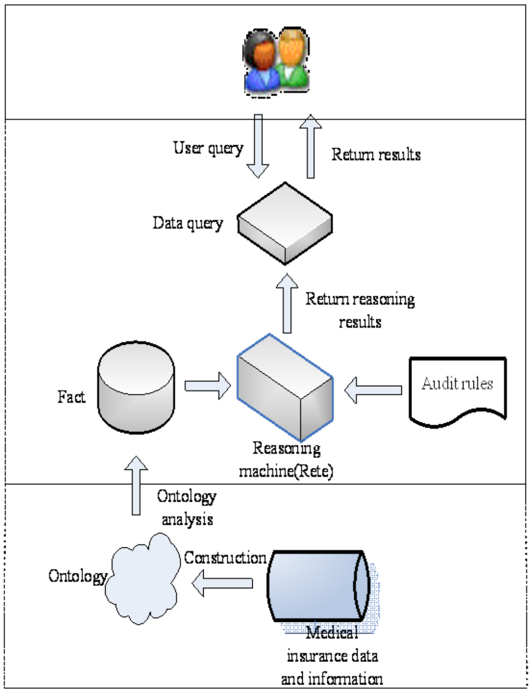


Figure 1. Framework on ontology-based guidance system of audit information

The system architecture on semantic retrieval is analyzed according to retrieval process, as shown in Figure 1. In the whole system, ontology is information organization framework of system. The metadata which is formed by analysis of audit information resources and on the basis of ontology annotation domain resources is stored in the database. Semantic reasoning is that retrieval system calls inference engine for reasoning according to the relevance of concept and inference rules by using the concept on ontology and instance (metadata)[3]. In view of the request on user's query, the results are searched in an ontology database. Because the metadata which is stored in the ontology is formed by reasoning, the matching results will contain implied conclusion of logical reasoning rules, which reveals the internal relationships among the things to realize intelligent retrieval.

3. The construction of audit method

Through the research and analysis of existed construction methods on domain ontology [4], this paper puts forward specific steps to build the audit method ontology, combining with the analysis which is for knowledge structure from the audit business and method in social insurance audit field [5], as shown in Figure 2.

The main steps in Figure 2 are described below.

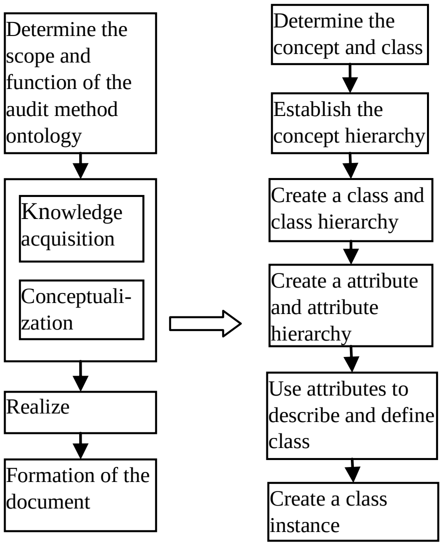

Determine the scope and function of the audit method ontology
Knowledge acquisition
Conceptuali-zation
Realize
Formation of the document
Determine the concept and class
Establish the concept hierarchy
Create a class and class hierarchy
Create a attribute and attribute hierarchy
Use attributes to describe and define class
Create a class instance

Figure 2. The construction of audit method ontology steps

(1)Determine the scope and function of the audit method ontology

Because there isn’t mature standard ontology in the social insurance audit field in use, in order to support the guidance system on audit information, a new method ontology on medical insurance audit needs to be constructed. The scope of the Medicare audit method ontology is the related knowledge of medical insurance auditing field, mainly including the region, the medical insurance fund, the insured person, the insurance unit, medical institutions, drugs, the nature of units, audit methods etc.. The related knowledge’s collection and acquisition in medical insurance audit field is mainly derived from audit experts and staff who actually participate in medical insurance audit [6].

The role of audit method ontology on medical insurance is mainly to manage a variety of knowledge on medical insurance audit field, especially audit method, providing the data and semantic support for service of audit guidance information, and then providing certain information foundation for the realization of the audit guidance system, which is an important part of the system [7].

(2)Determine the concepts and class of audit approach ontology

Ontology is an information organization and expression form which has a good kind of the hierarchical structure, and its concept has hierarchical relationships. The kind of hierarchical relationships well reflect the relationships between the concept and concept. It is convenient to describe the knowledge of class by using the inheritance relationships and relationships between parts and whole of concept [8].

In the actual medical insurance audit work, through the analysis and research of some auditing policies and regulations medical insurance, making actual investigation in some areas, listening to the actual business operations from the social security person, and collecting many information of medical insurance audit field, audit method ontology on medical insurance is constructed by extracting some concepts [9].

(3)Establish hierarchical structure among attributes and describe the relationship among the classes

After the analysis and construction of hierarchical relationships among classes on audit method ontology of medical insurance, it is necessary to set up properties of audit method ontology on medical insurance, including data property (Datatypeproperty) and object property (Objectproperty).

Data properties are used to express the knowledge of class about ontology, its domain is a kind of class in ontology, Range is the basic data types, i.e., any, float, int, string, date, datetime, etc.

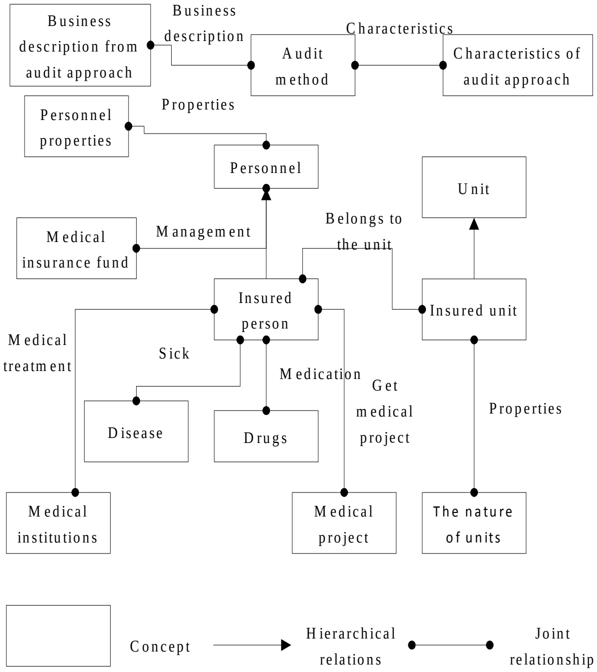


Figure 3. Conceptual relationship diagram of audit method ontology on medical insurance

Object properties are used to describe the relationships among classes. Its domain is a class of ontology, and its range is an instance of a class.

In the foundation on hierarchical relations b concept and class, according to the property relationships among concepts, conceptual relation diagram can be used to describe the overall of audit method ontology on medical insurance, as shown in Fig. 3.

4. The improved Rete algorithm based on active hash method

This section introduces the first fast matching algorithm-Rete algorithm based on the rule reasoning [10], in view of the existing problems in Rete algorithm, *e.g.*, because the fan-out of a node is too large, adding the fact process leads to low efficiency. It proposes Improved-method based on the active hash method, and gives the improved algorithm.

4.1. Rete algorithm

Rete algorithm is an efficient matching algorithm to realize the reasoning rules in the system based on rules [11]. The Rete algorithm firstly builds a Rete network, and then it provides an efficient implementation for expert system by the network for pattern matching.

The related concepts in Rete algorithm are as below:

(1) The fact: that is, the multiple relationships between the object and the object or between object and attribute. It generally uses a triple group to represent (id^ attr value), for example:

w1:(B1 ^on B2)
w2:(B1 ^on B3)
w3:(B2 ^left-of B3)

(2) Rule: The statement composed of conditional part and conclusion part is for reasoning, when existing facts meet the condition, rules are activated, getting rules corresponding conclusion. The general form of a rule is as below:

```
(name of the rule
LHS /* One or more conditions */
-->
RHS)
```

The LHS is part of the conditions, and RHS is the part of conclusions.

A rule example is as below：

```
(rule1
(B1 ^on B2)
(B2 ^left-of B3)
-->
(B1 ^on B3))
```

(3) Pattern: That is the condition part of the rules, generalization form of the known facts, and relationship of no instantiation. The form of pattern is as follows:

(^on)
(^left-of)
(^color red)

The compiled result of Rete algorithm is a Rete network consisting of the facts, rules and patterns, as shown in Fig. 4, including four nodes like the root node, type node, alpha node, beta node. Among them, the root node is the entrance of the rete network, not an actual node. Type node stores in the various types about facts, all the facts from the corresponding type node get into rete network.

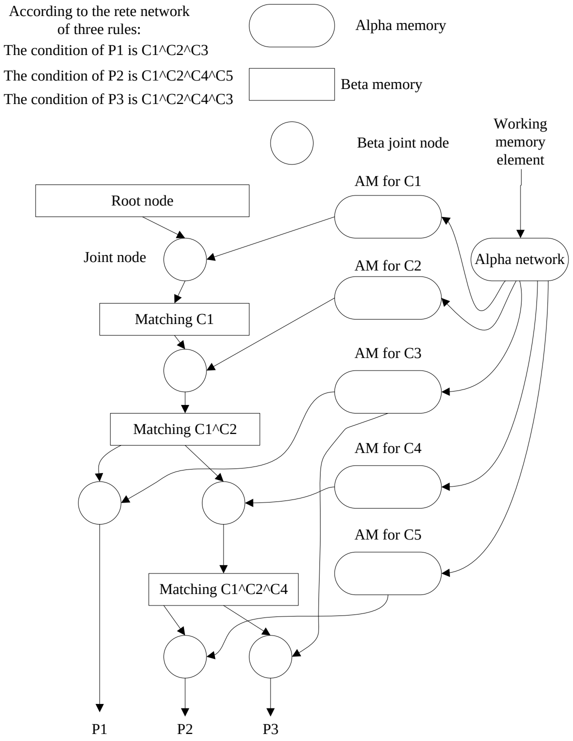


Figure 4. Rete network diagram

Figure 4 shows a Rete network which is constructed by three rules. It can be seen from the figure that working memory elements (namely the facts) into the Alpha network, according to the node types after discrimination, are into the corresponding Alpha memory (*i.e.* AM). According to certain sequence, alpha memory and other nodes do join operation. According to the conditions of each rule in the order, the first condition corresponding to the Alpha memory and a root node do are connected, and then generate a joint node which matches the first condition and beta memory. Each condition in rules and beta memory which is generated recently do joint operation, generating joint node that matches the condition and all conditions before, so on, until all conditions in rules are fulfilled, and the path after each rule matching is got finally.

4.2. Improved Rete algorithms

In Figure 4, the process how the facts enter corresponding the Alpha memory after it is into the Alpha network is transparent, however, it is an important process, as shown in Figure 5.

Figure 5 shows a fact will be added to the Alpha network, then the Alpha network sequentially detects attribute and value, but only detects the constants, facts will be added into the corresponding Alpha memory according to the detected constant.

This paper mainly studies and improves another drawback of Rete algorithm. It is obvious that adding fact has a big shortcoming [12], that is, if one of the attributes and values has a lot of constant forms, it means a node in the graph will spread out a lot of child nodes. If the most common method is used, *e.g.* using simple data-flow network, in turn, begins to detect these child nodes from the first child node, finds which is the corresponding child node of the current constant of fact, and then adds the facts to corresponding AM, which will be a lot of useless comparison operations, leading to time-consuming. For example, the color contrast in Figure 5, if some expert system contains many color constants, *e.g.* it has five kinds of color like red, maize, blue, green, white. when the corresponding color constant in fact is white, it does, in turn, the matching search from head, and then needs compare for 5 times, which the top four is invalid comparison operations, costing a lot of time, leading to a slow operation. In order to reduce the cost of comparison operation, this paper seeks an improved method.

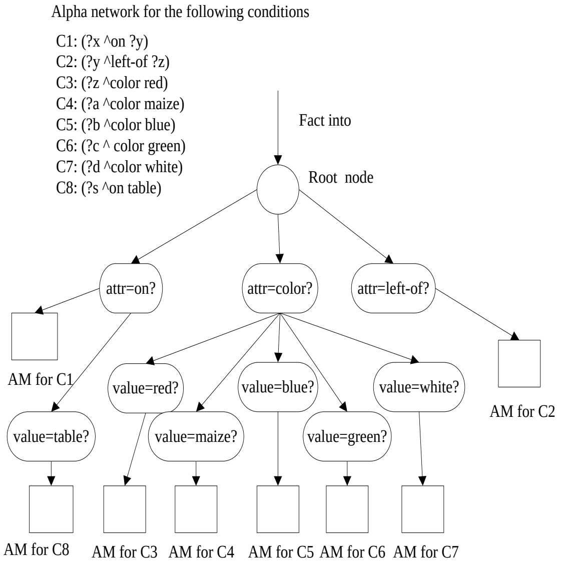


Figure 5. Facts are into the Alpha network process in Rete algorithm

4.3. Active hash method is introduced

In the existing problems in Rete algorithm, compared with traditional hash method, active hash method is an improvement to the traditional hash method. In the traditional hash methods, for frequent added constant in fact, the corresponding node of the constant is set to end of a linear list when the hash table is generated, and then it can be found by traversing a large number of nodes in the linear list, determining its position, which is bound to increase the average search length when hash table is successfully found. At same time, it leads to inefficiency, and the problem that the efficiency of adding fact in Rete algorithm is low doesn’t have a good solution.

In fact, many phenomenon and things in reality have a continuity truth, *i.e.* the thing doesn’t appear in a recent period of time, and it does in the near future, that possibility is also very small. The idea is also widely used in the page replacement of the operating system, namely Least Recently Used (LRU) replacement algorithm. Active hash method is put forward by the idea based on continuity principle. In active hash method, when a node is searched, the node moves forward to the linear list header, using the idea and operation can reduce the average search length in the process of the whole search on fact constant. The process of active hash method is as shown in Figure 6.

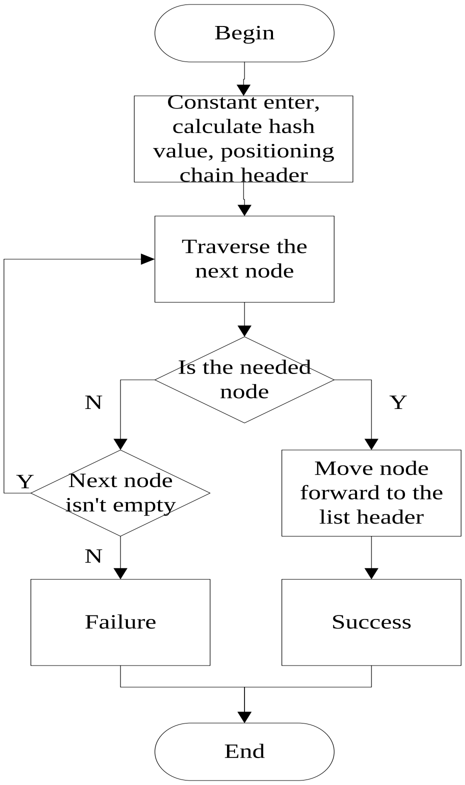

Begin
Constant enter, calculate hash value, positioning chain header
Traverse the next node
Is the needed node
N
Y
Y
Next node isn't empty
Move node forward to the list header
N
Failure
Success
End

Figure 6. The process of active hash

The specific steps of active hash method are as follow:

Step (1): According to the fact that enters the instance, in turn, testing attribute and value contains constants;

Step (2): For every constant, calculate hash value, to determine their own linked list, positioning chain header;

Step (3): To traverse the next node in the current linked list, check whether the constant with the constant contained in node is same, if so, perform step (4), otherwise, perform step (5);

Step (4): Move the current node forward to the list header in the current chain table, and then successfully add the fact into the Alpha network.

Step (5): Check whether the next node in current list is empty, if so, perform step (6), otherwise, backtracking perform the step (3);

Step (6): If the fact instances in Alpha network have no effect, abandon the fact instance. Return to perform the step (1) until all instances are into the Alpha network.

In the process of adding the fact in Rete algorithm, using active hash methods do hash of the node, compared to traditional hash methods, which reduces the average search length, and the efficiency was improved, but there are still some problems, considering a following situation. As shown in Figure 7.

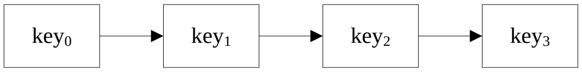


Figure 7. A situation of existing problem

Assume that the number of searching Key_3 is very few, and one is the search about Key_3. According to the active hash method, Key_3 will need to move forward to the linear list header after finding Key_3 successful. There is no search about Key_3 after a long period of time, however, the search is key_2, key_1, key_0. It is obvious that, in this case, the average search length to find these nodes are still large, however, the modification for the linear list header is too frequent, at same time there are some problems on efficiency and system load.

It is due to the existing problem, this paper finds a solution for improving active hash method, which makes it more suitable for constant hashed in Rete algorithm. The idea is that judging a node instead of moving directly it forward to the linear list header. If the node is always needed to find, it is moved forward to a linear list header, if not, do not move, it remains in the original location.

In this case, the problem becomes how to determine whether a node is often needed to search, and how to define that the measure becomes points to improve active hash method. The definition given in this paper is that using two times to find time intervals of the node as basis. A variable (*time*) is set at each node to record and store last time to find the node. Here uses "*search frequency g*" to represent a node frequency, and "*search frequency g*" of a node is the reciprocal of the current search time and the time interval of the last time to find, g = *1 / (time - timeold),* e.g. the larger is the time interval, the smaller the probability to be found recently is.

In addition, setting another variable average in the header node is as average of "*search frequency g*" on all nodes in corresponding linear list. When searching for a node, if according to formula on "*search frequency g*" to calculate the G isn't less than average value of "*search frequency g"* on all the nodes. It is considered that the probability of the node to be found is larger in the future, so it is moved forward to the linear list header, and updates the average value of all node "*search frequency g*" for the latest value (here needs set a variable at each node to support, that is execution times to find the node). Otherwise, it is considered that the node couldn't be searched in a long period of time, so node still maintains its original location.

In summary, the specific steps of the fact into the Alpha network based on improved active hash method of Rete algorithm are as followed:

Step (1): According to the fact that enters the instance, in turn, testing attribute and value contains constants;

Step (2): For every constant, calculate hash value, to determine their own linked list, positioning chain headers;

Step (3): To traverse the next node in the current linked list, check whether the constant with the constant which is contained in node is same, if so, perform step (4), otherwise, perform step (8);

Step (4): Calculate search frequency g of the current, e.g. the reciprocal of the current search time and the time interval of the last time to find, g = *1 / (time - timeold)*;

Step (5): The search frequency of current node isn't less than the average of all nodes "*search frequency g*" in current chain table, if so, perform step (6), otherwise, perform step (7);

Step (6): Move the current node forward to the list header in the current chain table;

Step (7): Update average, time, and num in current chain table, fact is successfully added into Alpha network;

Step (8): Check whether the node in the current chain table is empty, if so, perform Step (6), otherwise, backtracking perform the step (3);

Step (9): If the fact instances in Alpha network have no effect, abandon the fact instance. Return to perform the step (1) until all instances are into the Alpha network.

The progress of improved active hash method is shown in Fig.8.

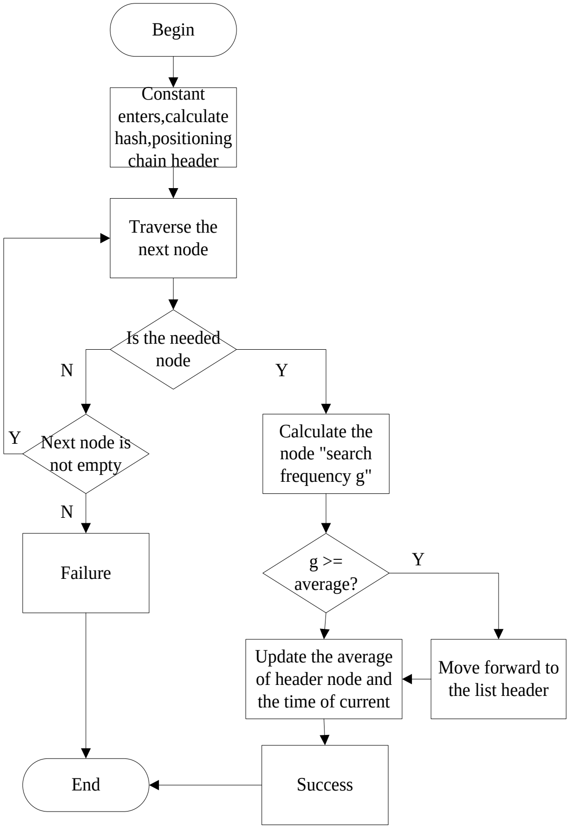

Begin
Constant enters,calculate hash,positioning chain header
Traverse the next node
Is the needed node
N
Y
Y
Next node is not empty
N
Calculate the node "search frequency g"
Failure
g >= average?
Y
Update the average of header node and the time of current
Move forward to the list header
End
Success

Figure 8. The process of improved active hash

LIST$_i$ is as a linear linked list in the hash table, which i represents the position of current linear linked list in the hash table. Assume that the capacity of hash table is N, Key denotes the corresponding Key-value of current constant. Every node in linear list contains a variable of time, which denotes that the number of the node has been found. In addition, each item in an array (*average*) represents the search frequency of the corresponding linear chain. The improved algorithm description is given below:

NotNull(i, j)： node LISTi[j] isn’t empty；

EQ(key, i, j)： Judge whether key and Key in LISTi[j] is the same；

TimeSpace(i, j) ： Obtain the time interval on the current time of current node LTSTi[j] and a search for the node last time;

GreaterThanAve(g, i)： Judge whether 1/g is greater than or equal to the average search frequency average[i] in the current linked list LISTi;

MoveFront(i, j)： Move the current node LIST[j] forward to the corresponding linked list header;

Update(i, j)： Update the average search frequency of current linked list LISTi, set the time in node LISTi[j] to the current time;

Input is the Key of constant form;

```
Procedure HashMethod(KeyType key)
begin
i = Hash(key);                // Obtain hash address
j = 1;                  // Traverse the next node of linked header for every linked list
while( NotNull(i, j)    // There is record in the position
&&  !EQ(key, i, j) )       // Key isn’t equal
```

```
j++;                // The next node of current linked list
end while;
if EQ(key, i, j)   then
      g = TimeSpace(i, j);
      if (GreaterThanAve(g, i))  then  // 1/g is greater than or equal to the average search frequency
                                          of current linked list
         MoveFront(i, j) ;
         Update(i, j);
      end if;
end if;
end
```

The fact instances of Alpha network in Rete is the algorithm input, and all effective fact instances entering into the corresponding Alpha memory in Alpha network are the results which are obtained after algorithm finished.

4.4. Algorithm analysis

In the social insurance audit, rules are set according to the policy, regulations and specific audit business. It uses the improved Rete algorithm based on active hash method, and it else uses social security audit data as a basis for matching rules, which activate audit rules that meet all conditions. It provide obtained conclusion as information for audit person.

(1)Complexity analysis

Assume that P is the number of condition pattern in a production rule, A is the number of Alpha memory in Alpha network.

The major space-spending when Rete algorithm run depends on how many beta memory is produced in Beta network, so the space complexity in Rete is represented by the number of the beta memory. The space complexity of improved Rete algorithm in the worst is $O(A^P)$.

The major time-spending when Rete algorithm run depends on how many times are compared between beta node in the Beta network and alpha node in Alpha network when they do joint operation. The consumed time that facts enter into the corresponding Alpha memory in Alpha network occupies a certain component in the time consumption of the whole algorithm. The time complexity of a matching rule in improved Rete algorithm in the worse time is $O(A^{2P-1})$.

Rete algorithm sacrifice space for time, and the rule matching algorithm based on rule reasoning in expert system has good applicability. This paper introduces improved Rete algorithm based on active hash method, which makes the execution time of algorithm to be further reduced.

(2)Comparison analysis

i) Experimental goal

In order to illustrate the validity of the improved Rete algorithm based on active hash method and it can improve the execution efficiency of algorithm, the different algorithms are realized in the same experiment environment, and it analyses and compares the performance of algorithm.

ii) Experimental data

The experimental data is a social security audit data in 2008 which is provided by Heilongjiang province audit department.

iii) Experimental program

In order to verify the efficiency of improved Rete algorithm, this paper conducted two sets of experiments. The first group sets the fixed number of rules; the other sets the fixed number of facts.

(3)Comparative analysis of the execution time

Two sets of experimental results are as shown in Fig. 9 and Fig. 10.

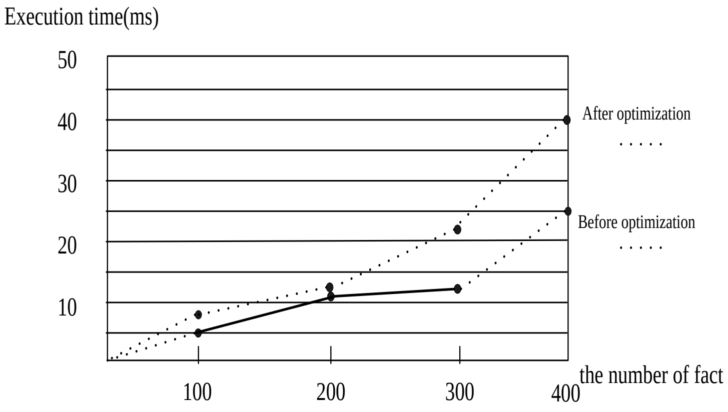


Figure 9. Execution time contrast of rule number fixed 10

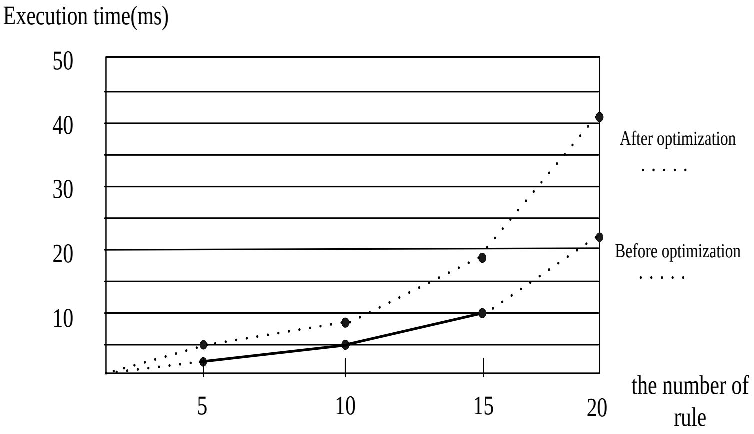


Figure 10. Execution time contrast of fact number fixed 100

The experimental results of first group are shown in Figure 9, and the number of rule is set at 10. The change is got by observing the execution time along with the number of fact increased. Form Figure 9, it is obvious that the execution time of algorithm before and after optimized algorithm increased along with the fact increased. When fact’s quantity is small, the efficiency of the optimized algorithm isn’t obviously improved. When the number of fact increases to 300 and 400, the efficiency of the optimized algorithm is obviously improved.

The experimental results of the second group are as shown in Figure 10. With reference to the first set of experiments, the number of fact in the second group is set at 100, and the change is got by observing the execution time along with the number of rule increased. Form Fig. 10, the execution time of algorithm before and after optimized algorithm increases along with the rule increased. At the same time, when rule’s quantity is small, the efficiency of the optimized algorithm isn’t obviously improved. When the number of rule increases to 15 and 20, the efficiency of the optimized algorithm is obviously improved.

It is seen from two sets of experiments, the efficiency of optimized algorithm is higher than the unimproved algorithm. When the number of facts and rules are large, the efficiency is obviously improved.

5. Conclusions

In intelligent retrieval system, the introduction of ontology provides the basic conditions which are necessary for the realization of the semantic reasoning. On the support of ontology database, the retrieval system understands the user's intent from the semantic level and the extraction of metadata which is based on ontology needs the retrieval of deep-level. The retrieval system seeks the relations which are contained in the concepts, explores the hidden information in the data, completes the deductive reasoning of metadata, and ultimately achieves the advanced and intelligent retrieval in the matching manner and all this is based on the inter-level association of concepts, associated attributes and related inference rules.

This paper studies rule inference algorithm- forward rules fast matching algorithm- Rete algorithm in expert system. There are some questions in the Rete algorithm, e.g. a node which its fan-out is too large leads to inefficient in the fact entering the process. In order to solve this problem we introduced the active hash method to improve. The improved Rete algorithm on efficiency in implementation is improved by validating analysis. The optimization of performance on reasoning algorithm and the perfection of the reasoning results are the focus of further research.

ACKNOWLEDGEMENTS

This work is sponsored by the Postdoctoral Science Funds of China under grant number 2013M541345, the Fundamental Research Funds for the Central Universities of China under grant number HEUCF100603 and HEUCFZ1212, the National Science & Technology Pillar Program under grant number 2012BAH08B02.

References

[1] N. Zhang and B. Li, “Research of Semantic Retrieve and the Key Technologies”, Computer Technology and Development, vol. 16, no. 11, (**2006**), pp. 22-25.

[2] B. Gounot, V. Donfack, V. Lasbleiz, J. Bourde, and A. Duvauferrier, “Regis, Creating an ontology driven rules base for an expert system for medical diagnosis”, Studies in health technology and informatics, vol. 169, (**2011**), pp. 714-718.

[3] Chen and Y. Jen, “Development of a method for ontology-based empirical knowledge representation and reasoning”, Decision Support System, vol. 50, no. 1, (**2010**), pp. 1-20.

[4] Y. Chi, “Ontology-based curriculum content sequencing system with semantic rules”, Expert System With Applications, vol. 36, no. 4, (**2009**), pp. 7838-7847.

[5] Vesin, B. Ivanovic, M. K. Milicevic, A. Budimac and Zoran, “Ontology-based semantic recommendation in programming tutoring system”, Expert System With Applications, vol. 39, no. 15, (**2012**), pp. 12229-12246.

[6] J. Qing and J. Zhou, “The Research on the Construction of Ontology-based Medical Image Knowledge Management System”, 2012 International Conference on Biological and Biomedical Sciences (ICBBS2012), (**2012**).

[7] K. Li and W. Gao, “Studies on Ontology-Based Irrigation Management Information Systems Modeling”, Proceedings of the Third IFIP International Conference on Computer and Computing Technologies in Agriculture (CCTA 2009), (**2009**).

[8] Cali, A. Gottlob, G. Pieris and Andreas, “Towards more expressive ontology languages: The query answering problem”, Artificial Intelligence, vol. 193, (**2012**), pp. 87-128.

[9] Esposito, M. D. Pietro and Giuseppe, “An ontology-based fuzzy decision support system for multiple sclerosis”, Engineering Applications of Artificial Intelligence, vol. 24, no. 8, (**2011**), pp. 1340-1354.

[10] W. Chen and J. Zhang, “A Survey on Computer-assisted Audit Techniquesc(CAATs)”, Computer Science, vol. 34, no. 10, (**2007**), pp. 290-294.

[11] Amailef, Khaled, Lu and Jie, “Ontology-Supported Case-Based Reasoning Approach for Intelligent m-Government Emergency Response Services”, Decision Support System, (**2013**).

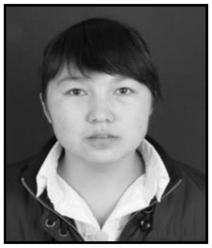
[12] Z. Hu, “Interference Finding Expert System Based on Case Reasoning and Rule Reasoning”, Computer Engineering, vol. 35, no. 18, (**2009**), pp.185-187.

Authors

Gang Liu (1976-), he is an associate professor with the College of Computer Science and Technology at Harbin Engineering University, China. He received his Ph.D. from Harbin Engineering University in 2008. His research interests include grid computing, distributed computing and simulation, and policy analysis.

Shaobin Huang (1965-), he is a professor with the College of Computer Science and Technology at Harbin Engineering University, China. He received his Ph.D. from Harbin Engineering University in 2004. His research interests include grid computing, distributed computing and simulation, and model checking.

Duan Zhang (1980-), he is at Harbin Resident Audit Office Of CNAO, China. He received his MSc from Northeast Forestry University, China, in 2010. His research interests include grid computing, information system auditing, on-line auditing and data model simulation.



Yudan Du (1989-), he is presently a MSc with the College of Computer Science and Technology, Harbin Engineering University, China. She received her Bachelor from Zhongyuan Usniversity of Technology, China, in 2012. Her research interests include ontology and rule-based reasoning.