2005 International Conference on Control and Automation (ICCA2005)
June 27-29, 2005, Budapest, Hungary

Study on Knowledge Base Verification Based on Petri Nets

Qingfeng Wu*, Changle Zhou, Jinlin Wu, Chaonan Wang

* Institute of Artificial Intelligence, Xiamen University

Computer Science Department, Xiamen University, Fujian, P.R. China, 361005

E-mail: pollysecond@sina.com

Abstract—**The comparison of rule pairs is usually involved in traditional approaches to verify knowledge base. The efficiency of these approaches is low when used in the verification of large-scale knowledge base because of the comparison. An alternative method of detecting logical errors in knowledge base is presented in this paper. This is achieved by analyzing the reachability and the transition sequence of Petri nets which is the established model of rule base.**

I. Introduction

With the technology development of Expert System and problems to be solved becoming more and more complicated the number of rules in knowledge base increased ramatically and the structure also become more complicated. In addition to this, the introduction of new rules into knowledge base and the modification of existed rules can affect the other rules in the base. Therefore, the verification and maintenance of knowledge base is the key part of the knowledge system and determined the validity of the whole system [1].

The comparison of rule pairs is usually involved in traditional approaches to verify the knowledge base, for example in reference [2], van Melle and etc. checked the redundancy and conflict by comparing every rule pair in theknowledge base. But in practice the scale of knowledge base becomes larger and larger so this method cannot meet the need of efficiency. There are some other methods such as decision table, graphics and so on. In reference [3], the knowledge verification tool, KVB is introduced. The strategy of KVB to verify is verifying locally first and then verifying globally. But this tool is not so effective when the scale of the knowledge base is large and has high degree of relationship between rules.

The method based on Petri Nets can avoid the rule pair's comparison by analyzing the relation between rules with the reachability and transition sequence of Petri Nets. For example in reference [4], the author extended the token and directed arc and established the knowledge base model. In reference [5] the method to find mistakes is presented. But when the Petri net model is used in verification, its running rules should be extended. In reference [6], the model cannot express the negation logic, and the method in reference [7] involves the search of a tree, so when the reasoning takes many steps and the base has high relative degree the workload of verification is high. A model based on colored Petri Net is presented in this paper and the mistakes of the knowledge base can be detected by analyzing the reachability and transition sequence of the model.

II. Object of Knowledge Base Verification

In AI systems production rules is a kind of generally adopted knowledge expression method, so the verification of knowledge especially for production rules is discussed in this paper. First the object of knowledge verification is keeping the system correct, integrate and consistent, so the following phenomena should be detected in rule base:

(1) Redundancy. If the same result can be obtained with the same premise when a rule is deleted from the knowledge base, there is redundant rule. The following are some detailed situation:

a. Rule equivalence.

If $P \wedge Q \rightarrow R, Q \wedge P \rightarrow R$ these two rules are totally equivalent.

b. Transmit redundancy.

Manuscript received December 31, 2004. This work was supported by National Science Foundation of P.R. China (60275023)

Qingfeng Wu is with the Computer Science Department, Xiamen University, China (phone: 86-0592-2182334; fax: 86-0592-2186546; e-mail: pollysecond@sina.com)

Changle Zhou is with the Computer Science Department, Xiamen University, China (e-mail: dozero@xmu.edu.cn)

Jinlin Wu is with the Computer Science Department, Xiamen University, China (e-mail: jlwu@xmu.edu.cn)

Chaonan Wang was with the Computer Science Department, Harbin University of Science and Technology
(e-mail: wang_chaonan@hotmail.com)

If $P \to Q, Q \to R, P \to R$, the third rule is redundant. It is not the same with the multiple reference routes. For example

$P \to Q, Q \to S, P \to R, R \to S$. This is multiple reference route and not redundant. In knowledge verification the existence of multiple reference routes is not a mistake.

c. Hypotactic rules.

When the conclusions of rule R_1 and R_2 are the same, and R_1 has more restrictions than R_2, then R_1 is the hypotactic rule of R_2. For example $R_1 : P \wedge Q \to R, R_2 : P \to R$.

(2) Conflict rules. If a rule in the knowledge base supports not only the affirmation of some conclusion but also the negation of the same conclusion, then there is conflict rule in the knowledge base.

a. Conflict rules.

For example, $P \to Q, P \to \neg Q$.

b. Conflict rules chain. Two groups of rules get conflict conclusion with the same premise. For example $P \to Q, Q \to R$ and $P \to S, S \to T, T \to \neg R$ are two conflict rule chains.

(3) Circularity. If a group rules can form circularity, then they are circulate rule chain. $P \to Q, Q \to R, R \to P$.

(4) Dead ends. If the conclusion of a rule can neither match the premise of any other rules, nor the final conclusion, this conclusion is called dead end. This can be classified as redundancy and will not lead to system crash, but it can lower the efficiency.

(5) Unreachable goals. If there is no reference path in knowledge base to reach the system target conclusion with any initial conditions, this conclusion is an unreachable goal.

III. PETRI NETS DEFINITION

The first step to verify knowledge is to precisely formalize the original knowledge. There are many ways to formalize knowledge, for example logic system, production system, linguistic network, frame system, object-oriented and so on.Adoption of formalization methods will affect the knowledge verification. Petri net is a proper tool for describing and studying systems that are characterized as being asynchronous and concurrent. It has rich ways to describe systems and technique to analyze their actions [9]. In addition formalized knowledge with all above methods can be easily transferred into those with Petri Nets. So Petri Net is adopted in this paper as the tool to model knowledge base so the verification can be relatively uniform.

First a kind of colored Petri Net is defined to express knowledge and then knowledge base model based on Petri Net is built up.

Definition 1 Colored Petri net is a six-tuple.

$\sum = \{P, T, F, C, I_-, I_+\}$.

1. P is the place set and T is the transition set. $F \subseteq P \times T \, Y \, T \times P$ is flow relation set.

2. D is a given color collection, and $Power(D)$ is the power of collection D.

$C : P \, Y \, T \to Power(D)$ means for any $p \in P, C(p)$ is the collection of possible colors in the place P. For $t \in T, C(t)$ is the collection of possible colors in transition T.

3. I_- and I_+ are negative and positive functions of $P \times T$ respectively. For any $(p,t) \in P \times T$:

$I_-(p,t) \in [C(t)_{MS} \to C(p)_{MS}]_L$, and the condition of $I_-(p,t) = 0$ is $(p,t) \notin F$. $I_+(p,t) \in [C(t)_{MS} \to C(p)_{MS}]_L$, and the abundant essential condition of $I_+(p,t) = 0$ is $(t,p) \notin F$.

$C(t)_{MS}$ is the multi-set of $C(t)$, $[C(t)_{MS} \to C(p)_{MS}]_L$ is the collection of linear function from $C(t)_{MS}$ to $C(p)_{MS}$.

Definition 2 Suppose $p \in P, t \in T$, ${}^*t = \{p | (p,t) \in F\}$ and $t^* = \{p | (t,p) \in F\}$ are the input and output place collection respectively.

Definition 3 Define $M : P \to D_{MS}$ as the mark of colored Petri net system $\sum$,

$\forall p \in P : M(p) \in C(p)_{MS}$ should be satisfied. A multi-set of the color set $C(p)$ is assigned to every place p and this describes the distribution of tokens in the system.

Definition 4 If $\forall t \in T : X(t) \in C(p)_{MS}$, then $X : T \to D_{MS}$ is a step of $\sum$.

Definition 5 Supposing X is the next step of the system of $\sum$ with the mark of M, the subsequence step M' is $\forall p \in P$:

$$M'(p) = M(p) + \sum_{t \in T} I_+(p,t)(X(t)) - \sum_{t \in T} I_-(p,t)(X(t))$$

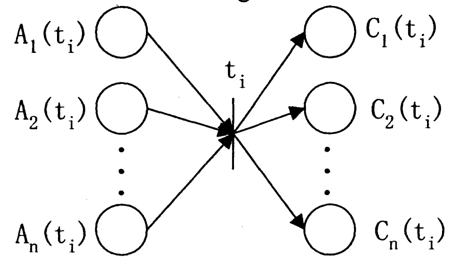
represented by $M[X > M'$.

IV. Formalization of Rule Base with Petri Nets

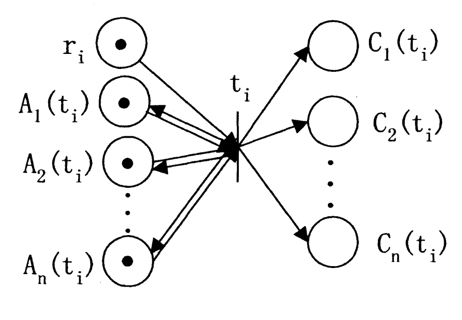
All the rules in the knowledge base can be represented by

$R_i : A_1(r_i) \wedge A_2(r_i) \ldots \wedge A_n(r_i) \rightarrow C_1(r_i) \wedge C_2(r_i) \ldots \wedge C_m(r_i)$

The mapping rules between knowledge base and the model which is defined above is as following: transition represents the execution of a rule; places are the premise and conclusion of rules; the logical relation of premise and conclusion is represented by directed arcs. For rule R_i, supposing t_i is its transition, then the premise $A_1(r_i) \wedge A_2(r_i) \ldots \wedge A_n(r_i)$ is ${}^{*}t$ and the conclusion $C_1(r_i) \wedge C_2(r_i) \ldots \wedge C_m(r_i)$ is t^{*}. Graphic representation of rule 1 is as figure 1.



Fg.1 Graphic of Rule 1



Fg.2 Extended Petri Net Model

Tokens in places represent if the clause of premise and conclusion is true. The negative and positive function on $P \times T$, which are I_- and I_+, determine the number of tokens, which is consumed and produced when the rule is executed. Places' sharing between different rules means the logic relationship between them. After transition is fired, tokens are released to its output place, this means the conclusion clause become true when the rule is executed.

But according to the running rules of Petri net, the premise status of rule as shown in figure 1 becomes uncertain because the token in place is consumed after the rule is executed. This is not to be expected. The initial facts and the facts produced in the process of reference should be reserved and can be used by multi rules. This problem can be solved by adding reverse arc between transitions of t_i and $p \in {}^{*}t_i$, that is let the $p \in {}^{*}t_i$ to be the adjoint places. But this can lead to new problem. Because the facts are reserved, transition t_i can be fired heaps of times. Place r_i is added as shown in figure 2. After transition t_i is fired the token in r_i is consumed, so at the next time transition t_i cannot be fired because it lacks of token. In this way, the rule can be used only once and the known facts can be reserved.

Apparently places in the above Petri net model can be classified into two categories, $P = \{P_C, P_R\}$, $r_i \in P_R$. For the convinces of analysis, P_C can be divided into three sub-sets, $P_C = \{P_{CE}, P_{CI}, P_{CG}\}$. P_{CE} is the collection of clauses which can obtain information through users' input and system database. P_{CI} is the collection of clauses which are produced in the reference process and P_{CG} is the collection of clauses which are the system conclusion. Definite color set D={b,w,f}, and the possible tokens in place are b,w and f. Color w means the clause or the conclusion represented by the place is true; color b means the clause or the conclusion represented by the place is false. Tokens in P_R has the color of f which means if the rule has ever been fired. The initial mark $M_0, M_0(P_R) = [1f]$, means there are no rules fired initially.

$M_0(P_c) \neq xB + yW$, x, y are positive integers and not equal to zero, which means there are no conflict facts initially.

V. Knowledge Verification and Example

According to the above mapping method, rules are formalized in the form of Petri net and the rules are connected by sharing places so a Petri net model of knowledge base is built up. Constructer mistakes can be detected by analyzing the reachibility and transition sequence of the Petri net.

(1) For the minimal initial mark M_0 that can fire transition sequence T_j, if there is redundancy in the knowledge base, there will be another transition sequence T_k and i, $T_j \cap T_k = \varnothing$, $M_0[T_j > M'$, $M'[T_k > M''$, which makes $M_0(p_i) = 0$, $M'(p_i) = 1b + 1f$ or $1w + 1f$, $M''(p_i) = 2b + 2f$ or $2w + 2f$, and j, k can be exchanged. As referred in the discussion of verification object, transmitted redundancy and multi-reference route are different and it is not a redundant mistake. An additional constraint should be included that there is at least one transition sequence which has only one

transition.

(2) For the minimal initial mark M_0 that can fire transition sequence T_j, if there is conflict in the rule set, then there should be another transition sequence T_k and i, makes $M_0(p_i)=0$,

$M'(p_i)=1b+1f$ or $1w+1f$,

$M''(p_i)=1b+1w+2f$.

(3) If there is circularity in rule set, then for the minimal initial mark M_0 that can fire the transition sequence T_j, and

$\exists i\ M_0(p_i)=1b+xf$ or $1w+xf, x\in\{0,1\}$

$M_0[T_j>M'$, makes $M'(p_i)=2b+2f$ or $2w+2f$.

(4) If there is dead end in the rule base, there should exist some initial mark M_0, $M_0(P_{CE})\neq[0]$, $M_0(P_{CG})=[0]$, makes $\forall T, M_0[T>M', M'(P_{CG})=[0]$.

(5) If there is unreachable goal in the rule base, any initial mark that satisfy $M_0(P_{CE})\neq[0]$, $M_0(P_{CI})=[0]$, $M_0(P_{CG})=[0]$, and $\forall T, M_0[T>M'$, $\exists k, p_k\in\{P_{CG}\}, M'(p_k)=0$.

Knowledge base verification is explained with knowledge base R as the example. The graphic of knowledge base R based on Petri net is shown in figure 3.

(1) Redundancy. For initial mark $M_0=[1w,0,1w,0,...,0]$ and transition sequence $T_j=\{t_2\}$, $M_0[T_j>M'$, $M'[T_k>M''$, $M_0(p_2)=0$, $M'(p_2)=1w+1f$, $M''(p_2)=2w+2f$, so there is redundancy in rule R_1, R_2, especially R_2 is the hypotactic rule of R_1. In another situation with the initial mark of $M_0=[1w,0,...,0]$ and transition sequence $T_j=\{t_1,t_3\}$ $T_k=\{t_4\}$ $M_0(p_4)=0$, $M'(p_4)=1w+1f$, $M''(p_4)=2w+2f$, rule R_1, R_3, R_4 are circulate redundancy.

(2) Conflict. With the initial mark of $M_0=[1w,0,...,0]$ and transition sequence $T_j=\{t_1,t_9\}$, $T_k=\{t_3,t_8\}$, $M_0[T_j>M'$, $M'[T_k>M''$. $M_0(p_8)=0$, $M'(p_2)=1b+1f$, $M''(p_2)=1w+1b+2f$, rule R_3, R_8, R_9 are conflict.

(3) Circularity. With the initial mark of $M_0=[0,0,0,0,1w,0,...,0]$, $M_0(p_5)=1w$ and the following transition sequence $T_j=\{t_5,t_6,t_7\}$, $M_0[T_j>M'$, $M'(p_5)=2w+f$. Rule R_5, R_6, R_7 form circularity.

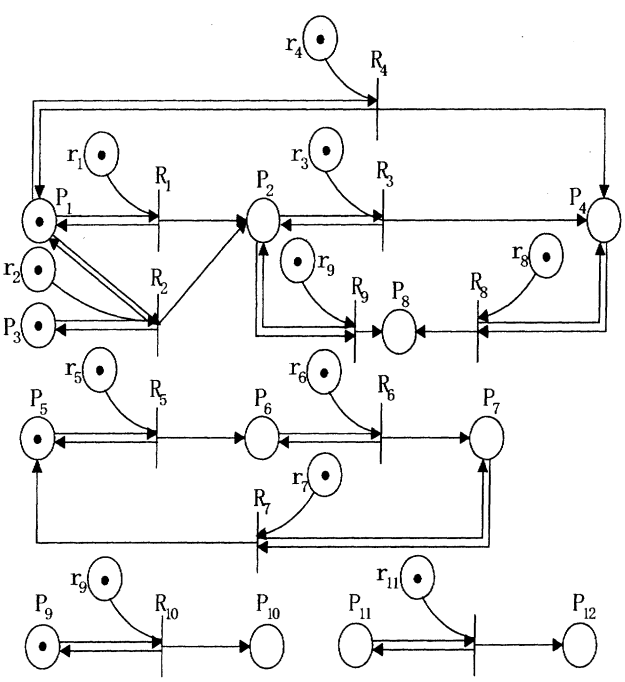
(4) Dead ends. With the initial mark $M_0=[0,0,0,0,0,0,0,0,1w,0,0,0]$, only transition t_{10} can be fired, and the result is $M'=[0,0,0,0,0,0,0,0,1w,1w,0,0]$. There is no other transition can be fired and the final conclusion is not reachable, that is

$\forall T, M_0[T>M', M'(P_{CG})=[0]$.

(5) Unreachable goal. With the initial mark, $M_0=[1w,0,1w,0,1w,0,0,0,1w,0,0,0]$ rule R_{11} will never be fired, that is $\forall T, M_0[T>M', M'(p_{12})=0$, so the conclusion of p_{12} is not reachable.

VI. Conclusion

The method presented in this paper to verify knowledge base turns the detection of logic mistake among knowledge to analysis of the reachiability and transition sequence of Petri net. Knowledge model based on Petri net is built up first which can reserve facts and express negation logic compared to ordinary Petri nets. The method to detect mistakes is presented at last.



$$R_1 : P_1 \rightarrow P_2$$
$$R_2 : P_1 \wedge P_3 \rightarrow P_2$$
$$R_3 : P_2 \rightarrow P_4$$
$$R_4 : P_1 \rightarrow P_4$$
$$R_5 : P_5 \rightarrow P_6$$
$$R_6 : P_6 \rightarrow P_7$$
$$R_7 : P_7 \rightarrow P_5$$
$$R_8 : P_4 \rightarrow P_8$$
$$R_9 : P_2 \rightarrow \neg P_8$$
$$R_{10} : P_9 \rightarrow P_{10}$$
$$R_{11} : P_{11} \rightarrow P_{12}$$

Fg.3 Knowledge Base Example

REFERENCES

[1] Wei-Tek Tsia, Rama Vishnuvajjala and Du Zhang, "Verification and Validation of Knowledge-based System", *IEEE Trans Knowledge and Data Eng,* 11(1), pp202-212, 1999.

[2] Van Melle W J, "System aids in constructuring computer consultation programs", *MI:UMI Research Press,* 1981,70-80

[3] Jie Zhang, Suzhen Zhang, Weisun Jiang, "Expert System Knowledge Base Verification", *Proceeding of East China University of Science and Technology,* 22(5), pp583-589, 1996.

[4] Liu Xiaoxia, " A Study on the Maintenance of a Rule Base Using Numerical Petri Nets", *[J] Mini-Micro Systems,* 18(7), pp74-77, 1997

[5] Liu Xiaoxia, "Knowledge Base Maintenance Systems Using Petri Nets", *[J] Mini-Micro Systems,* 23(7), pp859-861

[6] Derek L. Nazareth, "Investigating the Applicability of Petri Nets for Rule-Based System Verification", *IEEE Trans, Knowledge and Data Engineering,* 4(3), pp402-415, 1993.

[7] Jiang Hao, Luo Junzhou and Fang Ningsheng, "A Knowledge Verification Method Based on Color Petri *Net*", *Journal of Southeast University,* 30(1), pp77-82, 2000

[8] Mengshoel, O.J. and Delab, S. " Knowledge Validation: Principles and Practice", *IEEE Expert,* vol.8, pp62-68, 1993

[9] Chongyi Yuan. Theory of Petri Net. BeiJing. Publishing House of Electronic Industry, 1998