Proceedings of the Korean Nuclear Society Autumn Meeting
Taegu, Korea, October 1997

Knowledge Base Verification Based on Enhanced Colored Petri Net

Jong Hyun Kim and Poong Hyun Seong

Korea Advanced Institute of Science and Technology
Department of Nuclear Engineering
373-1 Kusong-dong, Yusong-gu
Taejon, Korea 305-701

Abstract

Verification is a process aimed at demonstrating whether a system meets it's specified requirements. As expert systems are used in various applications, the knowledge base verification of systems takes an important position. The conventional Petri net approach that has been studied recently in order to verify the knowledge base is found that it is inadequate to verify the knowledge base of large and complex system, such as alarm processing system of nuclear power plant. Thus, we propose an improved method that models the knowledge base as enhanced colored Petri net. In this study, we analyze the reachability and the error characteristics of the knowledge base and apply the method to verification of simple knowledge base.

I. Introduction

Verification involves the determination of whether the system is functioning or not. Verification is concerned with building the system right [1]. In the case of the knowledge base system, verification is concerned with ensuring that there is no rule set that can induce logical problems. Thus, verification can be defined as the demonstration of the consistency, completeness, and correctness of the knowledge base. As the expert system is applied to more organizations, it becomes more important to verify the knowledge base of the expert system. In recent years, many researchers studied the Petri net-based verification methods. Among these methods extended Petri net was adapted to model the knowledge base. The problem of knowledge base verification was transformed to that of reachability of specific state [2]. Petri net has been used to check the inconsistency of the knowledge base [3,4]. In an automated tool, called PREPARE, anomalies that appear in a knowledge base were defined as terms of Petri net and these terms were identified by using syntactic pattern recognition [5]. Even though Petri net have proved to be useful modeling tool for various practical applications, its representation becomes unmanageable when the system to be modeled is large and complex. In order to reduce this troubles Jensen have proposed colored Petri net. Colored Petri net is a kind of high level Petri net that is designed to model a large and complex system by coloring tokens of places. As an example of the verification with colored

Petri net, there has been an attempt to use the color to represent the negative and positive relationship of the knowledge base [6]. However it used the color in ineffectively way and was the incomplete colored Petri net application.

Colored Petri net has several weak points to model the knowledge base such that it loses the information after firing transitions. This study provides an improved method that models the knowledge base as enhanced colored Petri net (ECPN) which combines colored Petri net with Extended Petri net. Section II provides the formal definitions of colored Petri net and ECPN. The errors in knowledge base are defined in Section III and the error detectability is described with simple examples in Section IV.

II. Modeling the knowledge base with enhanced colored Petri net

We assume that rules in a knowledge base have the following format:

$$B_1, B_2, \cdots, B_n \Leftarrow A_1, A_2, \cdots, A_n$$

where A_n is a condition of rules, B_n is a conclusion, and "$\Leftarrow$" corresponds to "if" of rules.

When these rules are modeled with ECPN, the input places of a transition in ECPN represent the condition A_1, A_2, $\cdots$, A_n and the output place represents the conclusion B. "$\Leftarrow$" is replaced by the transition. Therefore, an elementary Petri net would represent the rule set of the form $A \Rightarrow B, C$ as shown in Figure 1. Figure 1 shows that the information of the condition, that is, the token of the input place, is lost after transition is fired. To settle this defect, ECPN has the shape as shown in Figure 2. As ECPN has a feedback arc, input place continuously possesses the tokens after firing transitions. A place S in Figure 2 plays a role of preventing a transition from being fired infinitely. Without the place S, Colored Petri net that has feedback arcs is enabled forever. In addition, place I is defined in this study. This place makes it simple to detect conflict rules. As shown in Figure 2, the conflict rules that can not coexist can be handled and detected with ease. One can notice that rules are the conflict rules when place I has two or more tokens.

The definitions of ECPN can be described as follows.

1) Definition 1: A enhanced colored Petri net is a tuple, ECPN (Σ, P, T, A, N, C, G, E, I)

(i) Σ is a finite set of non-empty types, called color sets.

(ii) P is a finite set of places.

(iii) T is a finite set of transitions.

(iv) A is a finite set of arcs such that:

- $P \cap T = P \cap A = T \cap A = \varnothing$

(v) N is a node function. It is defined from A into $P \times T \cup T \times P$.

(vi) C is a colored function. It is defined from P into Σ.

(vii) G is a guard function. It is defined from T into expression such that:

- $\forall t \in T$: [$Type(G(t)) = B \wedge Type(Var(G(t))) \subseteq \Sigma$]

(viii) E is an arc expression function. It is defined from A into expression such that :

- $\forall a \in A$: [$Type(E(a)) = C(p(a))_{MS} \wedge Type(Var(E(a))) \subseteq \Sigma$]

, where p(a) is place of N(a).

(ix) I is an initialization function. It is defined from P into closed expression such that:

- $\forall p \in P$: [$Type(I(p)) = C(p)_{MS}$]

Type(v), Type(expr), and Var(expr) denote the type of a variable, the type of an expression, and the set of variables in an expression respectively. Guard functions mean Boolean expressions. They define conditions that must be true for an activity to occur.

2) Definition 2: A step Y is enabled in a marking M iff the following property is satisfied:

$\forall p \in P$:

$$\sum_{(t,b)\in Y} E(p,t) < b > \leq M(p)$$

When $(t_1, b_1), (t_2, b_2) \in Y$ and $(t_1, b_1) \neq (t_2, b_2)$, (t_1, b_1) and (t_2, b_2) are concurrently enabled.

3) Definition 3: When a step Y is enabled in a marking M_1 it may occur, changing the marking M_1 to another marking M_2, defined by

$\forall p \in P$:

$$M_2 = M_1(p) - \sum_{(t,b)\in Y} E(p,t) < b > + \sum_{(t,b)\in Y} E(t,p) < b >$$

The first sum represents the removed tokens while the second represents the added tokens.

As defined above, a marking M in ECPN is given by

$$M = [\ M^C, M^S, M^I\ ]$$

Marking M^C can be divided into three ones with their natures – external, inferred and goal markings. Thus, M^C is given by

$$M^C = [\ M^{CE}, M^{CI}, M^{CG}\ ]$$

By dividing the marking M^C , ECPN is able to check the knowledge base anomalies, especially incompleteness, easily and effectively.

III. Knowledge verification with ECPN

The error types of knowledge base can be formulated as transition sequence problems [2]. The transition sequence problems of redundancy, circularity, conflict, dead end, unreferenced condition, unreachable conclusion, missing, and isolation are formulated by following propositions.

- Redundant rules: There is an initial marking M_0 that enables two nontrivial transition sequence, T_j and T_k . $T_t \cap T_k = \varnothing$. If the rule set R is redundant, then exists a marking $M^{Ci}_2$and i, s.t. $M_1 = \delta(M_0, T_j)$,$M_2 = \delta(M_1, T_k)$ and $c \in C(p)$, $c \in M^{Ci}_0 \cap M^{Ci}_1 \cap M^{Ci}_2$, $\#(c\ M^{Ci}_0)=0$, $\#(c, M^{Ci}_1)=1$, $\#(c, M^{Ci}_2)>1$.

- Circularity: If a rule set R has circular rules, then exists an initial marking M_0, with $\#(c, M^{ci}_0) = 1$, $c \in C(p)$, that minimally enables a transition sequence T_j, s.t. $M_1 = \delta(M_0, T_j)$, with $\#(c, M^{Ci}_1)>1$, $c \in C(p)$.

- Conflicting rule: There is a marking M_0 that minimally enables a nontrivial sequence of T_j. If the rule set R has conflicting rules, then for nontrivial sequence if integrity transition T_{ii} exist M^{Ii}_1 and i, s.t $M_1 = \delta(M_0, T_j + T_{ii})$, $c \in C(p)$, $c \in M_0 \cap M_1 \cap M_{Ii1}$ and $\#(c, M^{Ii}_1)>1$.

- Dead end: If a rule set R has a dead end, then exists an initial marking M_0, s.t. $c \in C(p)$, $c \in M^{CE}_0 \cap M^{CI}_0 \cap M^{CG}_0$, $\#(c, M^{CE}_0) \neq 0$, $\#(c, M^{CI}_0) \neq 0$, $\#(c, M^{CG}_0) = 0$, and $\forall T$ where $M_1 = \delta(M_0, T)$, $\#(c, M^{CG}_1) = 0$, $c \in C(p)$, $c \in M^{CG}_1$, and M^{CI}_1 with no output arc of color c.

- Unreachable goal: If a rule set R contains an unreachable goal, for all initial marking M_0 where $c \in C(p)$, $c \in M^{CE}_0 \cap M^{CI}_0 \cap M^{CG}_0$, $\forall M_0$, with $\#(c, M^{CE}_0) \neq 0$, $\#(c, M^{CI}_0) = 0$, $\#(c, M^{CG}_0) = 0$, and $\forall T$, $M_1 = \delta(M_0, T)$, exists k, s.t. $\#(c, M^{Ck}_1) = 0$ for $p_{Ck} = \{P_{CG}\}$.

- Unreferenced condition: If a rule set contains an unreferenced condition, then exists a marking M^{CI}_0 that $c \in C(p)$, $c \in M^{CI}_0$, $\#(c, M^{CI}_0) \neq 0$, and has no input arc of color c.

- Isolated rules : Unreferenced condition + Dead end

- Missing rules : Unreferenced condition + Unreachable conclusion + Dead end

IV. Simple applications

Simple examples are shown in Figure 3 and 4. Figure 3 contains two anomalies which are redundancy and circularity. Figure 3 (a) shows the initial marking. Figure 3 (b) shows the marking after the transition sequences are fired. It consists of two transition sequences, the process p that shows the redundant rule and the process q that represents the circular rule. We assume that two transition sequences of process p are interchangeable. The errors of these types can be detected with the propositions defined in the previous section. Figure 4 shows a few anomalies such as dead end, unreferenced condition, unreachable conclusion, missing rule, and isolated rule. Figure 4 (a) shows the initial marking and Figure 4 (b) shows the marking after the transition sequences are fired. If we assume that process p is the conflict rule, this error can be detected by checking 'the number of token of place I > 1'. The left side of the process q in the Figure 4 (b) shows the example of an unreachable goal. The right side of the process q shows a dead end, a missing rule and an isolated rule. One can know that these anomalies are detected with the propositions.

IV. Conclusions

In this study, we have described the formal approach of the knowledge base verification by using ECPN. It is found that the enhanced colored Petri net is suitable for the knowledge base verification of large systems. It also has the benefits such as making it easy to check anomalies and not losing the information as shown in Section III. Most of knowledge base problems were transformed as the transition sequence problems. Most of knowledge base anomalies can be detected by the presented method. The next step of the work would be to develop an algorithm for automating the knowledge base verification.

References

[1] R. M. O'Keefe, O. Balci, and E. P. Smith, "Validating Expert System Performance," IEEE Expert, vol. 2, No. 4, 81-89, 1987 Kurt Jensen, "Colored Petri Nets," Springer-Verlag, 1992.

[2] D.L. Nazareth, "Investigating the Applicability of Petri Nets for Rule-based System Verification," IEEE Trans. on Knowledge and Data Engineering, Vol. 4, No. 3, 402-415, June 1993.

[3] NGA KWOK LIU, "Formal Verification of Some Potential Contradictions in Knowledge Base Using a High Level Net Approach," Applied Intelligence, Vol. 6, 325-343, 1996.

[4] Tadao Murata, "A Petri Net Model for Reasoning in the Presence of Inconsistency," IEEE Trans. on Knowledge and Data Engineering, Vol. 3, No. 3, 281-292, June 1993.

[5] Du Zhang and Doan Nguyen, "A Tool for Knowledge Base Verification," IEEE Trans. on Knowledge and Data Engineering, Vol. 6, No. 6, 983-990, December 1994

[6] Chih-Hung Wu and Shie-Jue Lee, "Knowledge Validation with an Enhanced High-level Petri Net Model," Proceeding of the 11th conference on Artificial Intelligence for Applications, 126-132, 1996.

[7] Il won kwon, "A Method of Knowledge Base Verification and Validation for Nuclear Power Plants Expert Systems," KAIST, Korea, MA, 1996.

[8] Kurt Jensen, "Colored Petri Nets," Springer-Verlag, 1992.

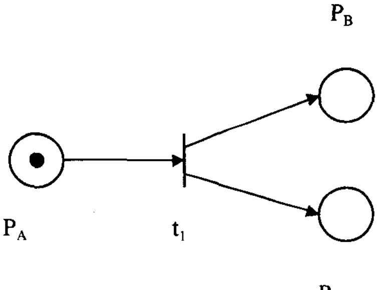


Figure 1. Petri net model

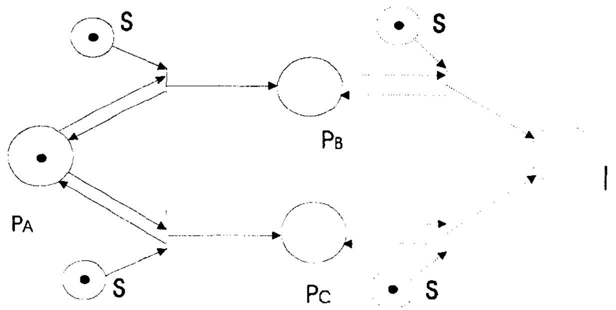
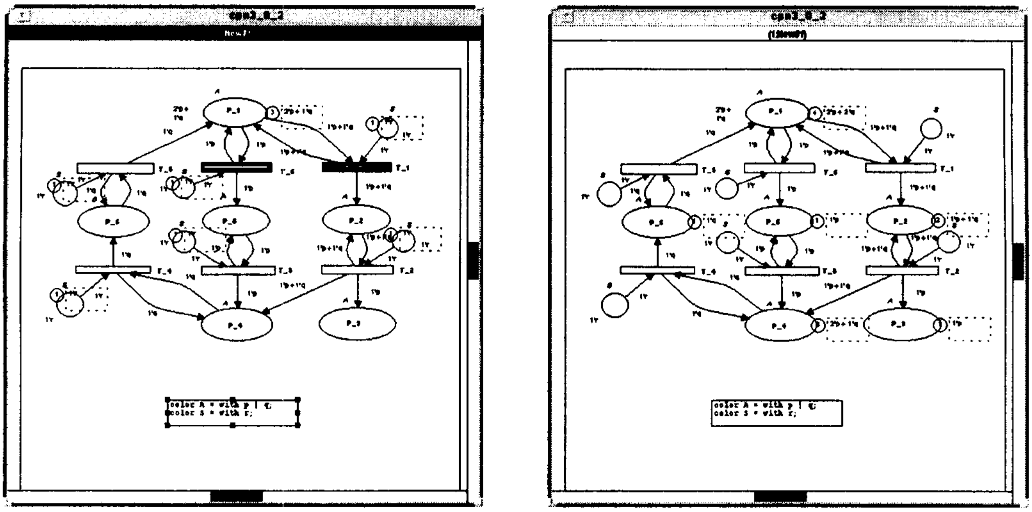
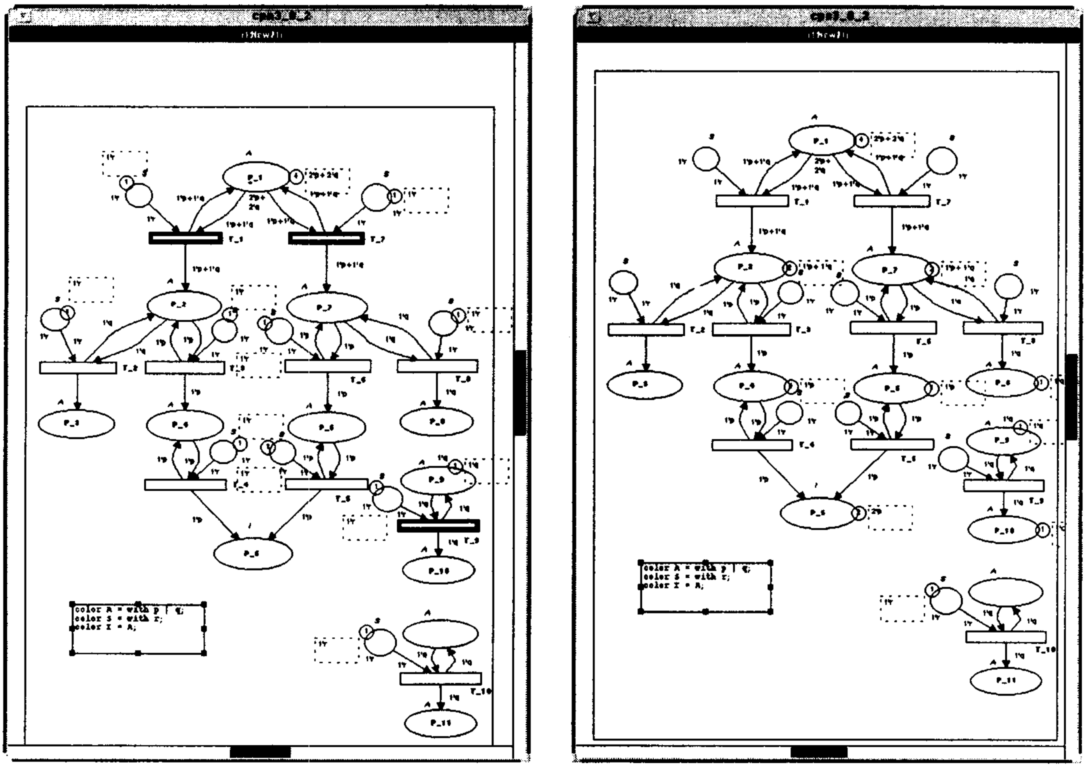


Figure 2. Enhanced colored Petri net model



(a) Before transition sequences (b) After transition sequences

Figure 3. Redundant and circular rules



(a) Before transition sequences (b) After transition sequences

Figure 4. Conflict rule, dead end, unrefereced condition, and unreachable goal