언폴딩에 기반한 유한페트리넷의 실용적 도달가능성 분석
(Pragmatic Reachability Analysis of Bounded Petri Nets based on Unfoldings)

김 의 석 † 이 정 근 †† 이 동 익 †††
(Euiseok Kim) (Jeong-Gun Lee) (Dong-Ik Lee)

요 약 본 논문에서는 유한페트리넷의 도달가능성에 대한 효율적인 분석방법을 제안한다. 일반적으로 도달가능성 분석은 유한페트리넷의 동적 성질에 관한 연구에 있어서 가장 근본이 되는 분석중의 하나이다. 그러나 일반적인 유한페트리넷의 경우, 도달가능성 분석은 분석시간과 요구되는 메모리 양의 측면에서 지수승의 복잡도를 요구하게 되며 많은 경우 상태공간폭발 문제를 야기할 수 있다. 이에 본 논문에서는 유한페트리넷의 분석을 위한 중간모델인 언폴딩(unfolding)상에서 장소(place)간의 구조관계 분석을 통하여 시간 및 메모리의 측면에서 효율적으로 도달가능성 분석을 수행하는 방법을 제안한다. 제안된 방법은 구조적 제약 없이 모든 종류의 유한페트리넷에 적용 가능하다.

Abstract This paper suggests an efficient reachability analysis method of bounded petri nets. Reachability analysis is a fundamental basis for studying the dynamic properties of any discrete event systems. However, it takes at least exponential execution time and memory space to verify in general petri nets. That is, state space explosion problem may occur. In this paper, we attack problems of previous approaches - state space explosion and restrictions to applicable petri net classes - by formulating the reachability problem as set operation over structural relations among places on an unfolding.

1. Introduction

Petri Nets(PNs) are widely recognized as a powerful model for communication protocols, concurrent and parallel programs, flexible manufacturing/industrial control systems and especially discrete event systems characterized by asynchronous and concurrent evolution[1]. For correct modeling and analysis of PNs, describing discrete event systems, deadlock-freeness, liveness, safeness, boundedness and reachability may be checked. In particular, reachability analysis is one of the most fundamental problems among analysis problems of discrete event systems. However, PN based reachability analysis is an intractable problem due to exponential execution time and memory space in the general case. Previous approaches to reachability analysis may be classified into following three groups; (1) the reachability graph(RG) method (2) the matrix equation approach, and (3) methods using structural characteristics such as SM-components, handles etc.[2]. The RG method is the most explicit method but it may suffer from state space explosion, i.e. the number of states of the RG may grow exponentially with the number of transitions due to the inherent concurrency of PNs. The matrix equation approach and structural

· 본 연구는 1998년도 한국학술진흥재단 과학기술기초중점연구(1998-016-E00058) 및 교육부 BK21사업에 의한 지원으로 수행되었음
† 학생회원 : 광주과학기술원 정보통신공학과
uskim@gegurikjist.ac.kr
†† 비 회 원 : 광주과학기술원 정보통신공학과
eulia@geguri.kjist.ac.kr
††† 종신회원 : 광주과학기술원 정보통신공학과 교수
dilee@kjist.ac.kr
논문접수 : 1999년 4월 15일
심사완료 : 2000년 4월 6일

characteristic based method have severe restrictions in applicable PN classes.

Recently binary decision diagrams(BDDs) and unfoldings have been proposed for representing state space of PNs efficiently[3][4]. However, since BDDs can not be derived from k-bounded PNs without knowledge of PNs' upperbound k, the BDD based method still has restriction in applicable PN classes. On the other hand unfolding, which was introduced by McMillan for the first time, has no restriction to its applicability other than unboundedness[4]. Moreover, since it is based on partial order semantics, there is no state space explosion problem inherently. Miyamoto et. al[5][6] made the first attempt to perform reachability analysis on unfoldings through characterizing it into a maximal complete subgraph division and achieved noticeable experimental results. However, in the worst case, when a given marking is not in a PN, their method should explore all the state space and hence it requires tremendous time. In this paper, we attack problems of previous approaches - state space explosion and restrictions of applicable PN classes - by formulating the reachability problem as set operation over structural relations among places on unfoldings. Since the suggested method in this paper decides a reachability of a given marking through a sequence of set computations among places in the marking, it is always performed without exploring the whole space. Therefore, although our approach still takes exponential time complexity, it can carry out reachability analysis quickly compared to previous approaches for most PNs. Moreover, it can be adopted for submarking reachability analysis. Therefore, it is very beneficial in the pragmatic viewpoint and we show its usefulness with experimental results.

2. Basic Definitions

In this section we introduce PNs and unfoldings, which are used as intermediate models for PN analysis.

2.1 Petri Nets

Definition 2.1 (Petri Net) A Petri Net is a 4-tuple $\Sigma=(P,T,F,M_0)$, where P is the set of places, T is the set of transitions, $F\subseteq(P\times T)\cup(T\times P)$, such that $dom(F)\cup range(F)=P\cup T$, M_0 is the initial marking represented by a $|P|\times 1$ column vector. By firing of a series of transitions, σ, a marking M changes into M', denoted by $M[\sigma>M'$. The set of reachable markings from M is denoted by $[M>$.

We use dot notations to represent the preset and postset of a place or a transition as follows;

(i) $\cdot t(t\cdot)$ is the set of input(output) places of t.

(ii) $\cdot p(p\cdot)$ is the set of input(output) transitions of p.

2.2 Unfoldings

In prior to defining unfoldings, we need to introduce OCNs on which unfoldings are based. OCNs, which are kind of acyclic PNs, can be derived from PNs by Algorithm 2.1.

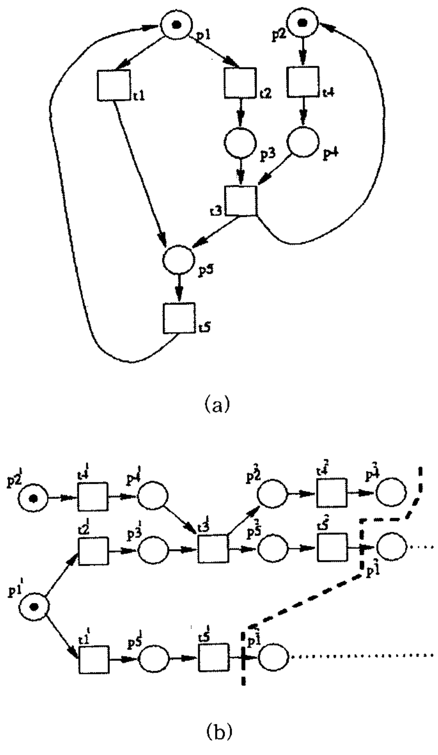


Fig. 1 (a) Petri Net (b) Unfolding

Algorithm 2.1(Generation of an OCN)

[Step1] Copy every place p_i such that $M_0(p_i)\geq 1$ into the OCN.

[Step2] Choose a transition t_i from the PN.

[Step3] *For each place in* $\cdot t_i$, *find a copy in the OCN.* If a copy can not be found, go back to Step2. Do not choose the same subset more than once for a given t_i.

[Step4] If any pair of chosen places is not in concurrent relation, go back to Step2.

[Step5] Make a copy of t_i in the OCN. Call it t_i^j. Draw an arc from each place which was found in Step3 to t_i^j.

[Step6] For each place in $t_i \cdot$, make a copy in the OCN and draw an arc from t_i^j to it.

[Step7] Repeat Step2-Step6 as many as possible.

In an OCN, P', T' and M' denote sets of places, transitions and markings corresponding to P, T and M in a PN respectively. In an OCN, a node in a PN can occur several times and j-th appearances of node p_i and t_i are denoted as p_i^j and t_i^j(j=1, 2, 3...) respectively. OCNs, which are based on partial order semantics and acyclic, are greatly helpful to investigate behavior of concurrent systems. An OCN represents acyclically all possible processes occurring in a given PN. Thus an OCN may expand infinitely, even though a given PN is bounded. This means that, at least in bounded cases, every possible process appears repeatedly in an OCN. It was proved that global states can also be defined in an OCN and a complete prefix, called *unfolding of an OCN* preserves the state space of a PN[4][7]. Hence an unfolding is very useful to analyse a PN without experiencing state space explosion. Fig. 1 shows a PN and a corresponding unfolding. For the lack of space, a formal definition of an unfolding is omitted here. Please refer to [4] and [7] for the related definitions.

Definition 2.2 (Ordering Relations) Let $\Sigma=(P,T,F,M_0)$ be an acyclic PN or an OCN and x_1, $x_2 \in P \cup T$.

(i) x_1 precedes x_2, denoted by $x_1 \Rightarrow x_2$, if(x_1, x_2) belongs to the reflexive transitive closure of F, i.e. there is a directed path in Σ from x_1 to x_2. x is in precedence relation with itself.

(ii) x_1 and x_2 are in conflict, denoted by $x_1 \# x_2$, if there exist distinct transitions t_1, $t_2 \in T$ such that $\cdot t_1 \cap \cdot t_2 \neq \emptyset$, and $t_1 \Rightarrow x_1$, and $t_2 \Rightarrow x_2$.

(iii) x_1 and x_2 are concurrent, denoted by $x_1 \| x_2$, if they are neither in precedence nor in conflict.

3. Reachability Characterization and a Pragmatic Solution on Unfoldings

In this section, we introduce a reachability problem and explain in detail how to solve it efficiently on unfoldings with the suggested approach.

3.1 A Reachability Problem in Petri Nets

As previously mentioned, reachability analysis of the PN is among the most important analyses of a system modeled by the PN. A sequence of fired transitions results in a sequence of markings. A firing or occurrence sequence is denoted by $\sigma = M_0 t_0 M_1 t_1 M_2 t_2 \ldots t_n M_{(n+1)}$ or simply $\sigma = t_0\, t_1\, t_2 \ldots t_n$. A reachability problem in PNs is as follows;

Problem 3.1 For a given marking M of a PN, is a marking M reachable from an initial marking M_0? or is there σ such that $M_0[\sigma\rangle M$?

For example, a marking $M=\{p_2, p_5\}$ of the PN in Fig. 1 is reachable from the initial marking $M_0=\{p_1, p_2\}$ through a firing sequence $\sigma_1 = t_2\, t_4\, t_3$, i.e., $M_0[\sigma_1 \rangle M$.

3.2 Solving a Reachability Problem Using Unfolding Based Method

Definition 3.1 For any place p_i' in an unfolding, the union of p_i' and places in the concurrent relation with p_i' is defined as $Y_{p_i'}$. That is, $Y_{p_i'}=\{p_i'\} \cup \{p_{ik}' \mid p_{ik}'$ and p_i' are in the concurrent relation, $k=1..n\}$.

For example, for a place p_1^1 of the unfolding in Fig. 1(b), $Y_{p_1^1}$ is $\{p_1^1, p_2^1, p_4^1\}$. Since structural relations between any two places on an unfolding can be obtained during building it, $Y_{p_i'}$ is easy to obtain.

Proposition 3.2 The set of places in the maximal *concurrent relation on an unfolding corresponds to a* marking of PN or its unfolding[8].

[Proof] Trivial ■

In this paper, sink transitions are not considered, since they are not suitable for a marking concept in Proposition 3.2. However, although there exist sink transitions, we can apply the same reachability analysis procedure suggested in this paper by replacing sink transitions with dummy places. Therefore, we assume that PNs have no sink transitions for the convenience of explanation.

Proposition 3.3 For any marking $M'=\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ in an unfolding, $Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap ... \cap Y_{p_{ik}^{jk}} = M'$. Moreover, for a multiset of places in the unfolding, $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$, if $Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap ... \cap Y_{p_{ik}^{jk}} = \{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$, there exists a marking $M'=\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ in the unfolding.

[Proof] Firstly we prove the first argument. Assume that there exists a place p_{ia}^{ja} which belongs to $Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap ... \cap Y_{p_{ik}^{jk}}$ but does not belongs to $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$. Since p_{ia}^{ja} is in $Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap ... \cap Y_{p_{ik}^{jk}}$, it is concurrent with all p_{ib}^{jb}(a≠b, a, b=1..k). Therefore, all the places in $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ are not in the maximal concurrent relation. This implies that $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ is not a marking from Proposition 3.2. This is a contradiction to the assumption and hence $Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap ... \cap Y_{p_{ik}^{jk}} \subset \{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$. Next, we assume that a place p_{ia}^{ja} in $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ does not belong to $Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap ... \cap Y_{p_{ik}^{jk}}$. According to Proposition 3.2, p_{ia}^{ja} is in concurrent relation with all the places in $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ except for itself. Therefore, p_{ia}^{ja} belongs to all $Y_{p_{im}^{jm}}$(m=1..k)s and hence is also in $Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap ... \cap Y_{p_{ik}^{jk}}$. Therefore, $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\} \subset Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap ... \cap Y_{p_{ik}^{jk}}$. From the above, $Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap ... \cap Y_{p_{ik}^{jk}} = \{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$.

We now prove the second argument. Notice that $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ cannot have the same places in this case although it is a multiset. Assume that $Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap ... \cap Y_{p_{ik}^{jk}} = \{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ is satisfied but $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ is not a marking in the unfolding. According to Proposition 3.2, since a marking of an unfolding is a set of places in the maximal concurrent relation, only the following two cases can be considered;

(case 1) There exist two places, p_{ia}^{ja} and p_{ib}^{jb}, which are not in the concurrent relation, in $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$.

(case 2) Any two places in $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ are concurrent with each other but there exists a place p which is concurrent with all the places in $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ but does not belong to $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$. In the case 1, since p_{ia}^{ja} and p_{ib}^{jb} are places in $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$, $Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap ... \cap Y_{p_{ik}^{jk}} = \{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ implies that $Y_{p_{ia}^{ja}}$ and $Y_{p_{ib}^{jb}}$ contain p_{ib}^{jb} and p_{ia}^{ja} respectively and p_{ia}^{ja} and p_{ib}^{jb} are in the concurrent relation with each other. Therefore, case 1 is impossible. In case 2, since p is concurrent with all the places in $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$, p belongs to all the $Y_{p_{ia}^{ja}}$'s(a=1..k). Therefore, it belongs to $Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap ... \cap Y_{p_{ik}^{jk}} = \{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ and the case 2 is also impossible. Hence, the assumptions are contradictory. Therefore, if $Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap ... \cap Y_{p_{ik}^{jk}} = \{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$, the marking $M'=\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ exists in the unfolding. ■

As mentioned in Proposition 3.2, a marking M in an unfolding is a set of places in the maximal concurrent relation. That is, there is no set of places where all places are concurrent with others and includes M. Proposition 3.3 gives a way to check concurrency and maximality in the same time through set operations of $Y_{p_{ia}^{ja}}$'s. For example, for a marking $M'=\{p_3^1, p_4^1\}$ of an unfolding in Fig. 1(b), $Y_{p_3^1} \cap Y_{p_4^1} = \{p_2^1, p_3^1, p_4^1\} \cap \{p_1^1, p_3^1, p_4^1, p_5^1\} = \{p_3^1, p_4^1\}$ is satisfied. Moreover, for the set of places, $\{p_1^1, p_4^1\}$, $Y_{p_1^1} \cap Y_{p_4^1} = \{p_1^1, p_2^1, p_4^1\} \cap \{p_1^1, p_3^1, p_4^1, p_5^1\} = \{p_1^1, p_4^1\}$ is satisfied. Therefore, the marking $\{p_1^1, p_4^1\}$ exists in the unfolding.

Theorem 3.4 In a PN with an initial marking M_0, a destination marking $M=\{p_{i1}, p_{i2}, ..., p_{ik}\}$ is reachable from M_0 iff a multiset of places, $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ such that $Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap ... \cap Y_{p_{ik}^{jk}} = \{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ exists in the corresponding unfolding.

[Proof] If M is reachable from an initial marking M_0, there exists a marking M' reachable from the initial marking M_0' in the unfolding, where M_0' and M' correspond to M_0 and M respectively. We assume that a multiset of places, $\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ such that $Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap ... \cap Y_{p_{ik}^{jk}} = \{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ exists in the corresponding unfolding. If $Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap ... \cap Y_{p_{ik}^{jk}} = \{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$ is satisfied, a marking $M'=\{p_{i1}^{j1}, p_{i2}^{j2}, ..., p_{ik}^{jk}\}$

exists in the unfolding according to Proposition 3.3. For M', M' is reachable from an initial marking M'_0 and this implies that the marking M is reachable from the initial marking M_0. ■

The following two examples explain how Theorem 3.4 can be applied to reachability analysis. For a set of places, $\{p_2, p_5\}$, of a PN in Fig. 1, since $Y_{p_2^2} \cap Y_{p_5^2} = \{p_2^2, p_5^2\} \cap \{p_2^2, p_4^2, p_5^2\} = \{p_2^2, p_5^2\}$, $\{p_2, p_5\}$ is reachable from the initial marking. As the second example, we consider whether $\{p_3, p_5\}$ is reachable or not. For all the pairs of places corresponding to p_3 and p_5, since $Y_{p_3^1} \cap Y_{p_5^1} = \{p_2^1, p_3^1, p_4^1\} \cap \{p_2^1, p_4^1, p_5^1\} = \{p_2^1, p_4^1\}$ and $Y_{p_3^1} \cap Y_{p_5^2} = \{p_2^1, p_3^1, p_4^1\} \cap \{p_2^2, p_4^2, p_5^2\} = \emptyset$, we can conclude that $\{p_3, p_5\}$ is not reachable from the initial marking by Theorem 3.4.

An algorithm to decide the reachability by using Proposition 3.3 and Theorem 3.4 is given below;

Algorithm 3.1 (Reachability analysis algorithm based on unfoldings)

[variable]

$M = \{p_{i1}, p_{i2} \ldots, p_{ik}\}$: a target marking

$P_{ia} = \{p_{ia}^{jn} \mid n=1, 2, \ldots\}$: a set of places in an unfolding corresponding to a place p_{ia}, a = 1..k, in a PN.

[Step1] Select each place from P_{i1}, P_{i2}, ... P_{ik} respectively and make a multiset $P_I = \{p_{i1}^{j1}, p_{i2}^{j2}, \ldots, p_{ik}^{jk}\}$ of places. If P_I was made previously, throw it away and build a new multiset. If we cannot get a new P_I, a given marking M is not reachable.

[Step2] If $Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap \ldots \cap Y_{p_{ik}^{jk}} = P_I$, the given marking M is reachable. Otherwise, drop P_I and return to Step1.

The time complexity of Algorithm 3.1 is $O(|P_{i1}||P_{i2}| \ldots |P_{i|M|}|)$. Although it is still exponential, if the size of $|P_{ia}|$ is small, the time complexity does not matter. The experimental results in section 4 shows that the size of $|P_{ia}|$ is between 1 and 2 in most cases. Therefore, the suggested method can decide reachability rapidly for most PNs without state space explosion. Moreover, although the sizes of $|P_{ia}|$s are a little big, heuristics based on submarking reachability analysis and structural relations among places can reduce reachability analysis time efficiently. Since heuristics are not main topics of this paper, we do not explain them in detail.

Corollary 3.5 For any multiset of places, $S = \{p_{i1}, p_{i2}, \ldots, p_{ik}\}$ in a PN, a marking M, where all places in S have tokens in the same time, is reachable from an initial marking M_0.⇔The multiset of places, $\{p_{i1}^{j1}, p_{i2}^{j2}, \ldots, p_{ik}^{jk}\}$, such that $\{p_{i1}^{j1}, p_{i2}^{j2}, \ldots, p_{ik}^{jk}\} \subseteq Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap \ldots \cap Y_{p_{ik}^{jk}}$, exists in a corresponding unfolding.

[Proof ⇒] Without losing generality, we assume that a marking M' corresponding to M in an unfolding includes $\{p_{i1}^{j1}, p_{i2}^{j2}, \ldots, p_{ik}^{jk}\}$. According to proposition 3.2, all the places in $\{p_{i1}^{j1}, p_{i2}^{j2}, \ldots, p_{ik}^{jk}\}$ are in concurrent relations with others. Therefore, any place p_{ia}^{ja} in $\{p_{i1}^{j1}, p_{i2}^{j2}, \ldots, p_{ik}^{jk}\}$ belongs to all $Y_{p_{il}^{jl}}$(l=1..k)s and ⇒ is satisfied. ■

[Proof ⇐] We assume that any two places p_{ia}^{ja} and p_{ib}^{jb}, which are not concurrent with each other, exist in $\{p_{i1}^{j1}, p_{i2}^{j2}, \ldots, p_{ik}^{jk}\}$. Since $\{p_{i1}^{j1}, p_{i2}^{j2}, \ldots, p_{ik}^{jk}\} \subseteq Y_{p_{i1}^{j1}} \cap Y_{p_{i2}^{j2}} \cap \ldots \cap Y_{p_{ik}^{jk}}$ is not satisfied under the assumption, the assumption is wrong. That is, any two places in $\{p_{i1}^{j1}, p_{i2}^{j2}, \ldots, p_{ik}^{jk}\}$ are in concurrent relation. This fact implies that there exists a marking M' where all places in $\{p_{i1}^{j1}, p_{i2}^{j2}, \ldots, p_{ik}^{jk}\}$ have a token simultaneously. A marking M corresponding to M' is reachable according to the definition of unfoldings and all places in S have a token under the marking M. Hence ⇐ is satisfied. ■

Corollary 3.5 presents a way to decide submarking reachability through a minor modification of Theorem 3.4 and hence we can decide both reachability and submarking reachability simultaneously according to Corollary 3.5.

4. Experimental Results

Here, we claim that the suggested method is very beneficial in the following two aspects;

- to solve a reachability problem of any bounded PNs.
- to be competitive or superior to previous

Table 1 Size Comparison between Petri Nets and Unfoldings

Example	PN (pl+tr)	Unfolding (pl+tr)	Unfolding / PN	Example	PN (pl+tr)	Unfolding (pl+tr)	Unfolding / PN
sendr-done	17	17	1	hybridf	42	75	1.79
hazard	22	22	1	wrdatab	57	81	1.42
fair-arb-sg	33	47	1.424	nbe10b	54	102	1.88
half	19	27	1.42	master-read	68	128	1.88
pla	18	24	1.33	pe-send-ifc	114	124	1.08
qr42	30	30	1	mp-forward-p	42	45	1.07
sbuf-read-ctl	35	38	1.09	rcv-setup	29	29	1
fair-arb-sg-jo	52	77	1.48	trimos-send	48	100	2.08
nowick	37	37	1	alloc-outboun	43	45	1.04
rpdft	44	44	1	slave-j25.fc	131	155	1.18
atod	31	31	1	low-lat-new	41	130	3.17
chu133	31	31	1	low-lat-good	37	122	3.29
vbe5b	28	28	1	low-lat-true	45	157	3.49
vbe5c	24	27	1.125	low-lat-unsaf	69	235	3.41
chu150	30	30	1	nak-good	66	420	6.36
ram-read-sbu	50	53	1.06				
nak-pa	44	44	1	Average		1.638	
roberto	36	36	1	Standard Deviation		1.139	

approaches in the aspects of execution time and memory space.

Since the suggested method is based on unfoldings, which are free of any structural restrictions except unboundedness, the first advantage is inherent. Therefore, in this section we show that the suggested method has the second advantage through comparisons with previous methods. Table 1 shows the average and standard deviation of unfoldings' relative sizes to PNs' sizes for various PN benchmarks. As discussed in the last section, the suggested method has time complexity of $O(|P_{i1}||P_{i2}|\ldots|P_{iM}|)$. Although it still has an exponential time complexity, if the sizes of $|P_{ia}|$'s are small, the exponential time complexity does not matter in the pragmatic aspect. As shown in the Table 1, the size of an unfolding is 1.638 times bigger than that of a PN on the average and the standard deviation is 1.139. These values imply that the relative sizes of unfoldings are small and relatively regular for most PNs. Moreover, most worst cases can be avoided through submarking reachability analysis and heuristics based on structural relations among places. Therefore, the time complexity $O(|P_{i1}||P_{i2}|\ldots|P_{iM}|)$ results in short execution time for most PNs.

Table 2 shows the execution time and the model sizes needed in RG, BDD based method and the suggested method respectively. Simulation is performed on SUN sparc20 with 128MB main memory. Data for RG and BDD based methods show the execution time required in building RGs or BDDs and data for the suggested method represents unfolding derivation time and reachability analysis time for any 100 markings. In this simulation, we select the RG based method and the BDD based method as competitors because the former is most traditional approach and the latter is the representative approach suggested in order to overcome state

Table 2 RG based method v.s. BDD based method v.s. Unfolding based method(suggested)

Example	PN (pl+tr)	States of RG	Execution Time(sec.)	Nodes of BDD	Execution Time(sec.)	Unfolding (pl+tr)	Execution Time(sec.)
amul-sh3	57	168	0.1	369	0.97	81	0.08+0.16
amul-sh4	76	960	0.5	604	4.80	141	0.27+0.16
amul-sh5	95	6,080	4.1	1730	30.56	180	0.35+0.23
amul-sh6	114	29,760	28.6	2499	193.48	228	0.77+1.04
amulet3	81	513	0.3	514	3.57	113	0.19+0.08
amulet4	75	741	0.5	1931	8.48	238	4.22+0.08
amulet5	135	39,150	38.8	2138	275.22	252	0.83+0.42
amulet6	162	283,095	fail	3534	1121.09	334	1.67+1.53
dme10	81	11,264	8.3	271	2.79	81	0.11+0.1
dme20	161	2.2×10^{7}	fail	553	46.65	161	0.42+0.12
dme30	241	3.3×10^{10}	fail	833	265.88	241	1.17+0.1
dme40	321	4.5×10^{13}	fail	1113	879.14	321	2.53+0.12

space explosion. Benchmarks used in simulation are scalable PNs where state space enlarges very rapidly with sizes of PNs and they show difference in efficiency among the RG, the BDD based method and the suggested method, explicitly. RG based method shows similar or superior performance in PNs with small state spaces compared to BDD based method or the suggested method. However, as the Table 2 presents, since an execution time and a required memory space increase exponentially, it may be impossible to build RGs themselves let alone perform reachability analysis. This phenomenon means that the RG based approach is suitable only for small PNs with small state spaces but for PNs generating large state space it is inefficient and inappropriate. The BDD based method, which was invented to attack state space explosion problem, shows off its robustness and efficiency carrying out reachability analysis for PNs where the RG based method failed. However, as previously mentioned, since BDDs can not be obtained from k-bounded PNs without knowledge of PNs' upperbound k, BDD based method still has a restriction of applicable PNs. Moreover, it may still suffer from state space explosion for several PNs generating large state space. Compared to the former two methods, ours demonstrates both efficiency and robustness performing reachability analysis successively for all benchmarks containing PNs generating tremendous state space. As previously mentioned, unfolding based reachability analysis was performed by Miyamoto et. al for the first time. Table 3 presents comparisons between them and the approach in this paper and each simulation is performed for any 10 markings. For all the benchmark groups in Table 3, the suggested approach shows off its faster speed than [5]. Since the method in [5] should explore whole state space for non-reachable markings, reachability analysis time increases sharply with the size of unfolding. Compared to them, since suggested method decides reachability through set operation of structural relations among places in a target marking, it can decide reachability fast without searching whole state space. However, our approach still has a weak point. As explained in section 3, the suggested method requires exponential time complexity, $O(|P_{i1}||P_{i2}|\ldots|P_{|M|}|)$, and it may result in large execution time as the sizes of marking and unfolding increases. Simulation data for the last benchmark group in Table 3 show this weak point of the suggested algorithm. Several items corresponding to the suggested method have two values, one is small value and the other is large value

Table 3 [5] v.s. method suggested in this paper

Example	Execution Time(sec.) [5]	Execution Time(sec.) suggested method	Example	Execution Time(sec.) [5]	Execution Time(sec.) suggested method
dme10	0.03	0.01	dme20	0.13	0.01
dme30	1.63	0.01	dme40	6.21	0.02
simp-ph10	0.82	0.01	simp-ph20	18.29	0.02
simp-ph40	300.94	0.02	simp-ph50	445.05	0.02
15-5pipe	0.16	0.11	30-10pipe	29.96	0.5
45-15pipe	3444.3	0.42/101.44	60-20pipe	fail	2.14/fail

or fail, which means that execution time exceeds 1 hours. For most simulations, only small execution time is required but there are some cases requiring large execution time. In order to improve the utilization of the suggested method, the methods presented in [7] and [9] can be adopted for reducing the relative size of an unfolding and smarter heuristic method to reduce search space efficiently will be needed. Consequently, in spite of this weak point, we conclude that the suggested reachability analysis method based on unfoldings can achieve better performance than other existing methods through several experimental results.

5. Conclusions and Future Work

In this paper, we suggest a reachability analysis method working on unfoldings. we attack several problems of previous approaches - state space explosion and restrictions in applicable PN classes - by formulating the reachability problem as set operation of structural relations among places on an unfolding. Experimental results show that speedup in execution time, low memory requirement and no class restriction are achieved through the suggested method. Nowadays we are focusing our effort on upgrading our reachability analysis program and deriving smaller unfoldings.

References

[1] T. Murata, "Petri Nets: Properties, Analysis and Application," In *Proceedings of the IEEE*, Vol. 77, No. 4, 1989.

[2] D. I. Lee, S. Kumagai and S. Kodama, "Handles and Reachability Analysis of Free Choice Nets," *Lecture Notes in Computer Science*, vol.935, pp.298-315, Jun. 1995.

[3] E. Pastor, O. Roig, j. Cortadella and R. M. Badia, "Petri Net Analysis Using Boolean Manipulation," *Lecture Notes in Computer Science*, vol.815, pp.416-435, Jun. 1994.

[4] K. I. McMillan, "Symbolic Model Checking," *Kluwer Academic Publishers, Boston*, 1993.

[5] T. Miyamoto and S. Kumagai, "A Graph Theoretic Approach to Reachability Problem with Petri Net Unfoldings," *IEICE Transactions on Fundamentals*, vol.E79-A, pp.1809-1816, Nov. 1996.

[6] T. Miyamoto, "Methods and Application of Petri Net Unfoldings for Discrete Event System Verification," *PhD thesis*, Osaka University, 1997

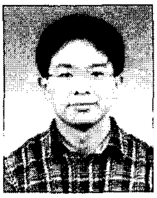
[7] A. Kondratyev, M. Kishinevsky, A. Taubin and S. Ten, "Analysis of petri nets by ordering relations in reduced unfoldings," *Technical Report TR:95-2-002, The Univ. of Aizu*, Jun. 1995.

[8] A. Semenov, A. Yakovlev, E. Pastor, M. A. Pena, J. Cortadella, L. Lavagno "Partial Order Based Approach to Synthesis of Speed-Independent Circuits," In *Proceedings of ASYNC'97*, pp.254-265, Apr. 1997.

[9] C. H. Hwang and D. I. Lee, "A Concurrency Characteristic in Petri Net Unfolding," *IEICE Transaction on Fundamentals*, vol.E81-A, pp.532-539, Apr. 1998.

김 의 석

1995년 연세대학교 전산과학과 학사. 1997년 광주과학기술원 정보통신공학과 석사. 1997년 ~ 현재 광주과학기술원 정보통신공학과 박사과정. 관심분야는 병행시스템(Concurrent Systems) 해석 및 설계, Petri Nets 이론, 비동기 회로 설계 및 CAD 등



이 정 근

1996년 한림대학교 전자계산학과 학사. 1998년 광주과학기술원 정보통신공학과 석사. 1998년 ~ 현재 광주과학기술원 정보통신공학과 박사과정. 관심분야는 비동기 회로, 병렬 및 분산 계산, formal methods 등

이 동 익

1989년 오오사카 대학 전자공학과 석사. 1993년 오오사카 대학 전자공학과 박사. 1993년 ~ 1994년 일리노이 대학 컴퓨터 공학과 방문연구원. 1990년 ~ 1995년 오오사카 대학 전자공학과 문부교관. 1995년 ~ 현재 광주과학기술원 정보통신공학과 부교수. 관심분야는 병행시스템(Concurrent Systems) 해석 및 설계, Petri Nets 이론, 이동 에이전트 시스템, 보안시스템, 비동기 회로 설계 및 CAD 등