세미죠인을 기반으로 한 패트리 넽의 형식적 정의

이 종 근[†]

요 약

본 연구에서는 패트리 넽에서의 함수적 축소방법을 제시하였다. 이 방법의 기본 개념은 하나의 관계에서 다른 관계에로의 사상함수를 트랜지션의 관계에 적용하는 것이다. 본 연구에서는 먼저 패트리 넽을 관계형 함수로 표현한 후에, 공통의 플레이스를 제거하고 두 개의 트랜지션이 하나의 새로운 트랜지션으로 축소되는 CF-join과 공통의 플레이스를 겹침으로 하나의 트랜지션화하는 CE-join, 그리고 공통의 플레이스를 제거함으로 별개의 간소화된 트랜지션으로 나뉘는 EQ-join을 제시하였다. 축소된 패트리 넽은 기존의 넽이 갖는 성질들 - 생존성과 보존성등을 그대로 보존한다.

A formal Definition of Semi-Join Based Reduction Method of Petri Nets

Jongkun Lee[†]

ABSTRACT

A functional reduction method of Petri nets is proposed. The method is based on interpretation of relations and transitions with functions which map one series of a relation to another. In particular, we propose CF−join which combines two transitions to new one after reduction of the common places, CE−join which superpose two transitions to one after superposition of the common places, and EQ−join which reduces the common places, after the Petri nets were explained by a relational scheme. A reduced net can be obtained without changing the properties such as liveness and boundness.

1. Introduction

Petri nets are graph models usefuls for analyzing and modeling the system which has concurrent characteristics, like parallel system, communication protocol and etc. The main advantage of Petri nets consists of the possibility of proving formally some desirable properties such as liveness,boundness or reversibility[5]. However, recently, these systems have become large and complex, which makes it difficult to analyze systems.

Most of the reduction methods[1, 2, 4, 7, 8, 12, 14, 15] for solvong complex problems of Petri nets focus on developing rules for combining several nets with some very specific rules of refinement or compositions, without changing certain properties. This approach has produced synthesis technique which allows to build impressive example. It has been found that no characterization of the expressive power of the technique is given[5]. One of possible solution to this problem is the formal definition of which rules should be used to obtain all and only the well behaved model of a certain class. Actually, the transitions can be agglomerated into new one, with an equivalent interface,after the reduction of common places between them. In all the classes defined in literatures, it is likely that the operation of the reduction has a similar result

[†] 종신회원 : 창원대학교 전산과 교수

논문접수 : 1994년 3월 4일, 심사완료 : 1994년 5월 1일

to the composition of two functions and to the join of two relations in therelational database[21, 22]. If Petri nets are explained by a relational from like a relational database, we could use composition or join concepts for combining nets.

In this study, we propose a functional formalism which is based on an interpretation of the query and relations as functions which map one column of a relation to another column. The basic idea is that a relation instance maps from a set of output places of transition in Petri nets to the corresponding set of input places in adjacent transition. Then we use this functional formalism to select two joinable transitions and to agglomerate two transitions into one by the reduction of common places. Ordinary Petri nets[11] are considered in this paper.

This paper is organized as follows. In section 2, the functional view of relations for the reduction process is introduced, in section3, some definitions of the Petri nets are given. In section 4, the reduction rules which combine two transitions into one by reduction of common place and which reduce the common places are proposed. In section 5, the properties of the Petri nets and a reduction algorithm are given. We demonstrate an application of these rules to a version of the ECMA protocol and we also compare our reduction rules to other approaches.

2. Relation and Transitions as Functions

2.1 Functions defined by a relation

Let V be a finite set of attributes {A1,A2,···, An}. Each attribute Ak has a finite domain dom (Ak)={p1,···,pm}. R(V), the scheme over V, is the set of relations on A1, A2,···,An. "r"(also called an instance) is an element of R(V), iff⊂r A1xA2x···xAn. An element t of r is a tuple t=(a1,···an) with ai∈dom(Ai), t[I] denotes the projection of t on I, subset of V, t∈[1,n].

Definition 2.1:Functional dependency on a scheme R

Let I ,J be two subsets of V. Then there is a functional dependency I−>J on R(V), iff for each r of R(V),∀t1,t2∈r such that $t_1[I]=t_2[I]$ $t_1[J]=t_2[J]$.

A function t from R[I] to R[J] is defined by each instance r of R(V) with I−>J as follows: for each t∈r t[J] = f(t[I]).

"f" is uniquely defined since two tuples that are equal upon I, are also equal upon J. This function is not necessarily a total one.

If I={A'_1,···,A'_n}, f is not defined on all values of A'_1 x···x A'_n. f is only defined on those values belonging to r[I].

Definition 2.2 : Relational function(RF, for short)

Let us denote R(V): I−>J, the functions defined by all instances r, maps from R[I] to R [J], where the scheme R(V) obeys a functional dependency I−>J. These functions can be characterized by couples (x,y) (such that y=f (x)) defined as follows:

$$\{(x,y) \mid t\in r, r\in R(V), t[I]=x, t[J]=y\},$$

also we can show that as follows:

$$\{x,y\}=(_{AI\ \ AJ})(\sigma_{AI=X}(R(V))),$$

where is a projection formular form and σ is a selection formular form in R.

If I(resp J) is a singleton (set with one element i(resp j)), we shall use the notation R (V):i−>j instead of $R(V)_{(i)->(j)}$, when no confusion arises.

Examples:

Let R(V):1−>2,with V={A1A2}, then we get:

A1	A2
p1 p1 p2	p3 p4 p5

R(V):1−>2(p1)={(p1,p1),(p3,p4)}

R(V):1−>2(p2)={p2,p5},

A relational tableau has a (fixed) number of columns, called attributes and a number of rows, called tuples. The domain of the ith column of the tableau, corresponding to an attribute Ai.

To simplify our works, we consider all relations which have only two attributes, i.e. R(V), with V={A1A2}.

For any one tuple v_1 in V, if it has multivalues, such that $|dom(v_1[A_i])| > 1, (1 \leq i \leq 2)$, then it called the partial tuples, and if $\forall$ $|dom(v_1[A_i])| = 1$, then called the total tuple.

Example:

Let R(V), with V=(A1A2},for any one tuple v1,v2 in V then:

1) partial tuple:

if $v1[A_1]=\{p1,p2,p3\}$, $v_1[A_2]=\{p1,p2\}$

2) total tuple:

if $v_2[A_1]=\{p1\}$, $v_2[A_2]=\{p2\}$

	A1	A2
v1	p1p2p3	p1p2
v2	p1	p2

2.2 Composition of relational function

Let f:A−>B and g:C−>D be two functions. If the image domain of f is a subset of the domain of g, then the composition g⊙f, of the two functions can be formed where g⊙f: A−> D is defined by g f=g(f(x)) for all x∈A. The image domain of g⊙f is thus a subset of the image domain of g[10, 18, 19]. We give next definition corresponding to the idea of the composition of RFs.

Definition 2.3 : Composition of two RFs

Let S(V)$:_{1->2(x)}$, with V={A1A2}, T(W):1−>2, with W={B1B2} be two RFs. For any two tuples t_1 in V, and t_2 in W, then a composition (S(V):1−>2⊙T(W):1−>2) is defined as:

- dom(S(V)$:_{1->2(2)}$⊙T(W)$:_{1->2}$: $\{x \in dom(t_1[A_2]) \mid (dom(t_2[B_1]) \subseteq dom(t_1[A_2])) \vee (dom(t_2[B_1]) \supseteq dom(t_1[A_2]))\}$
- (S(V):1−>2(x)T(W):1−>2=T(S(V)): $(dom(t_1[A_1]) \cup dom(t_2[B_1])) - (dom(t_1[A_2])\ dom(t_2[B_1]))$
  → $dom(t_1[A_2]) \cup dom(t_1[B_2])) - (dom(t_1[A_2])\ dom(t_2[B_1]))$
  $(x: (dom(t_1[A_1]) \cup dom(t_2[B_1])) - (dom(t_2[A_2]) \cap dom(t_2[B_1])))$.

Composition of two RFs are classified into the following three kinds:

− If all RFs have the total tuples, (S(V)$:_{1->2(X)}$ T(W)$:_{1-2}$,has a total tuple.

− If $(dom(t_2[B_1]) \subseteq (dom(t_2[A_2]))$, two RFs S (V)$:_{1->2}$ and (S(V)$:_{1->2}$⊙T(W)$:_{1->2}$) have same domains.

− If $(dom(t_2[B_2]) \supset (dom(t_1\{A_2]))$, the range of S(V)$:_{1->2(x)}$ is included in the range of (S(V)$:_{1->2(x)}$⊙T(W)$:_{(1->2)}$, such that $dom(t_1[A_2]) \cup (dom(t_1[A_2]) \cup dom(t_1[B_2])) - (dom(\bar{t}_1[A_2]) \cup dom(t_2[B_1])))$.

Example:

Let S(V):1−>2,with V={A1A2},T(W):1−>2, with W={A1A2} and R(X):1−>2, with X={A1A2}, then:

1) S(V):1−>2(x):(p1p2)−>(p3), T(W):1 −>2:(p3) −>(p4)
   (S(V):1−>2(x)⊙T(W):1−>2) : (p1p2) −>(p4)
2) S(V):1−>2(x):(p1p2)−>(p3p4), T(W):

1->2:(p3) ->(p5)

(S(V):1->2(x)⊙T(W):1->2) : (p1p2) ->(p4p5)

3) S(V):1->2(x):(p1p2)->(p3), T(W):1 ->2:(p3p5) ->(p4)

(S(V):1 -> 2(x) ⊙ T(W):1 -> 2) : (p1p2p5)->(p4)

2.3 Join of RF

The join is an operation that takes two relations and concatenates each tuples of the second relation with those tuples of the first relation that satisfy a specified condition. The condition is defined over the attributes of the two relations[20]. There are several kinds of oin operators: equi-join and natural-join and etc., for any total tuple in the general relation models. But in this paper, we consider two cases of tuples such as total tuple and partial tuple which are introduced in previous clause. Hence, in this section, for join of two RFs in the relations, we propose a new join operator the EE-join(note them by $><^{EE}$).

The EE-join operator is applied to two relations, R(V) with A1A2 and S(W) with A1A2, via a common attribute where $[R.A_2] = [S.A_1]$. The EE-join of these relations makes a new relation, T(X) with Y1Y2,whose tuples, t X, have new two attributes(Y1 : the domain of the tuples and Y2:the range of them):

$$t[Y1]=(r[R.A_1] \cup s[S.A_1])-(r[R.A_2] \cap s[S.A_1])$$

$$t[Y2]=(r[R.A_2] \cup s[S.A_2])-(r[R.A_2] \cap s[S.A_1]),$$

are formed by concatenating those tuples of R(V) and S(W), $r \in R(V)$, $s \in S(W)$, whose $[R.A_2]= [S.A_1]-$ components are $((r[R.A_2] \subseteq s[S.A_1]) \vee (r[R.A_2] \subseteq s[S.A_1])$, i.e. $R(V)[[R.A_2]=[S.A_1]]\ S(W)=T(X)$, where,

$$T(X)=\{t \mid \exists r \in V, s \in W, t[Y1] = (r[R.A_1] \cup s[S.A_1])-(r[R.A_2] \cap s[S.A1]) \wedge t[Y2]=(r[R.A_2] \cup s[S.A_2])-(r[R.A_2] \cap s[S.A_1])\}.$$

We shall use the notation of EE-join like as R(V):i->j><^{EE}S(W):i->j.

Example : Fig. 1 shows an example EE-join of relational function derived from the relation instances R(V) and S(W).

R(V)

A1	A2
p1 p2	p3
p1	p4
p2p3	p5p6

S(W)

A2	A3
p3 p4	p5 p6
p4	p5
p5	p1p7

R(V) ><EE S(W)

Y1	Y2
p1 p2 p4	p5 p6
p1	p5
p2p3	p1p7p6

(Fig. 1) An instance of EE-Join trom R(V) and S(W) relation

Thus, EE-join of previous RFs defines the following forms:

(i) R(V):$_{1->2}$(p1p2)><^{EE}S(W):$_{1->2}$(p3p4), we get:
Y1 = {p1p2p4}, Y2={p5p6},

(ii) R(V):$_{1->2}$2(p1)><^{EE}S(W):$_{1->2}$(p4), weget:
Y1={p1}, Y2={p5},

(iii) R(V):$_{1->2}$(p2p3)><^{EE}S(W):$_{1->2}$(p5), we get:
Y1={p1p3}, Y2={p1p6p7}.

2.4 Semi-join as function composition

It is important to point out that the composition of two functions f and g derived from two relations R and S, respectively may be defined using a relation which is a join of R and S. Gardarin demonstrated that a function composition is almost equal to a semi-join in [6].

Lemma 2.5 : Semi−join[21]

The following relational algebra operations:

1) selection of a relation S(V), with $V=\{A_1A_2\}$, on the $dom(A_1)$,

2) EE−join of the result with relation, T(W), with $W=\{B_1B_2\}$,

3) projection of the result on Y and Y', can be represented as a composition of two RFs derived from S(V) and T(W).

Therefore, the following relation is obtained:$S(V) \odot^{EE}(W)$.

Proof:Let S(V) with $V=\{A_1A_2\}$ and T(W) with $W=\{B_1B_2\}$ be two relations. These two relations EE−semi−joinable then for any two tuples t1 in S(V),t2 in T(w),if $t_1[A_2]\ t_2[B_1]$, where $\theta$is a set of constraints of the form $x\theta y$; $\theta \in \{=,>,<,\geq\ ,\leq\}$. Then we have:

$E=(_{\{Y1,\ Y2\},})(\sigma_{A1=(x)}(S(V))><^{EE}T(W))$, where

$Y1=dom(t_1)\cup dom(t_2)-(t_1[A_2]\cup t_2[B_1])$,

$Y2=ran(t_1)\ ran(t_2)-(t_1[A_2]\cap t_2[B_1])$.

Let S(V):1−>2 with $V=\{A_1A_2\}$ and T(W):1 −>2 with $W=\{B_1B_2\}$ be two RFs in two relations S(V) and T(W), repectively, then let S (V):1−>2(x)⊙T(W):1−>2 be a composition of two RFs:

E=T(W):1−>2(x)⊙S(V):1−>2

$=(dom(t_1)\ dom(t_2)-(t_1[A_2]\cap t_2[B_1]))$

$->(ran(t_1)\ ran(t_2)-(t_1[A_2]\cap t_2[B_1]))$

$(x:(dom(t_1)\ dom(t_2)-((t_1[A_2]\ t_2[B_1])))$.

Lemma 2.6 : the composition of two RFs T(W): 1−>2 [with $W=\{B_1B_2\}$] S(V):1−>2(x)[with $V=\{A_1A_2\}$] can be expressed as a unique RF using the EE−join of the two relations defining the RFs S(V):1−>2 [with $V=\{A_1A_2\}$] and T (W):1−>2 [with $W=\{B_1B_2\}$].

Proof : Let E=T(W):1−>2⊙S(V):1−>2(x), then we have:

$E=(dom(t_1)\cup dom(t_2)-(t_1[A_2]\cap t_2[B_1]))$

$\rightarrow(ran(t_1)\cup ran(t_2)-(t_1[A_2]\cap t_2[B_1]))$

$(x:(dom(t_1)\cup dom(t_2)-((t_1[A_2]\cap t2[B1])))$.

Let S(V) with $V=\{A_1A_2\}$ and T(W) with $W=\{B_1B_2\}$ be two relations. If $t_1[A_2]\ t_2[B_1]$,where$\theta\exists \{=,>,<,\geq\ ,\geq\}$, for any two tuples t_1 in S(V) and t_2 in T(W),then these two relations are semi−joinable. Then new relation $T'(X)=T(W)><^{EE}S(V)$,with $X=\{Y1,Y2\}$ is:

$Y1=dom(t_1)\cup dom(t_2)-(t_1[A_2]\cap t_2[B_1])$,

$Y2=ran(t_1)\ ran(t_2)-(t1[A_2]\cap t_2[B_1])$.

In this new relation,we can consider one RF:

T'(X):1−>2(x),with Y1Y2, where

$X=(dom(t_1)\ dom(t_2)-((t_1[A_2]\ t_2[B_1])))$.

3. The Notations of Petri net

In this section, we use the common notations of Petri net as in [3,5,9,11] and only recall the most important ones.

3.1 Notations of Petri nets

Definition 3.1 : PN=(P,T,E,S), a Petri net, is a 4−tuple, where $P=(p_1,\cdots,p_n)$ is a finite set of symbols, each symbol is called a places, $|P| \neq 0$, $T=(t_1,\cdots,t_m)$ is a finite set of transitions, $|T| \neq 0$, and E=PxT−>N is a input function, where is the set of positive integers, and S=TxP −>N is a output function.

Definition 3.2 : [21] The transition has 4 modes based on the number of input and output places.

1) The transition t is called T mode: it has only one input place and output place, i.e., $|{}^{0}t| = |t^{0}| =1$,

2) The transition t is called J mode : it has several input places and only one output place , i.e., $|{}^{0}t|>1$, $|t^{0}| =1$,

3) The transition t is called F mode : it has only one input place and several output

places , i.e., $|{}^{o}t| = 1$, $|t^{o}| > 1$,

4) The transition t is called X mode : it has several input places and several output places , i.e., $|{}^{o}t| > 1$, $|t^{o}| > 1$.

Definition 3.3 : A marked net, <PN,Mo> is a Petri net, PN, with an initial marking Mo. A transition $t \in T$ is enabled at a marking M iff $M(p) \geq E(p,t)$ for every $p \in P$. If t is enabled at M, then transition t may be fired yielding a new marking M' given by the equation:$M'(p) = M(p) - E(p,t) + S(t,p)$ for all $p \in P$. M(t>M' denotes that M' is reached from M by firing t.

Definition 3.4 : A finite sequence of transition, s =t1,···,tn, is a finite firing sequence of <PN,Mo >, iff there exists a sequence of marking M0 t1 M1 t2,···,tn Mn such that $\forall i, 1 \leq i \leq n$:Mi(ti>Mi+1. Marking Mi+1 is said to be reachable from Mo by firing s : Mo(s>Mi+1. An infinite sequence of transitions, s=t1,··· is an infinite firing sequence of <PN,Mo> iff there exists a sequence M0t1M1t2,··· such that $\forall i, 1 \leq i$:Mi−1(ti >Mi.

Definition 3.5:Let R(PN,Mo) be the set of all marking reachable from Mo.

Let L(PN,Mo) be the set of all firing sequences and their suffixes in <PN,Mo>:L(PN,Mo)={s | M(s>and $M \in R(PN,Mo)$}.

Definition 3.6 : A marked net <PN,Mo> is bounded iff there exists a finite $K \in N$ such that for all $pi \in P$ and for all reachable marking M,M $\in R(PN,Mo)$, $M(pi) \leq k$(if k=1 then the marked net is called Condition/event net or Safe net).

A transition t is live in <PN,Mo> iff for all marking $M \in R(PN,Mo)$, there exists $M' \in R(PN, M)$ such that M' enables t.

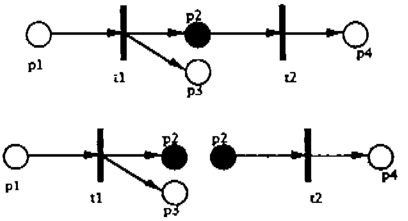
The marked net <PN,Mo> is live iff $\forall t \in T$ are live.

Definition 3.7 : [13] Let <PN,Mo)> be a marked net. Pn is decomposible via a subset Po of P iff there exists {t1,t2} a subset of T, such as $({}^{0}(t1) \cup (t1)^{0}) \cup ({}^{0}(t2) \cup (t2)^{0}) = Po$.

Example:

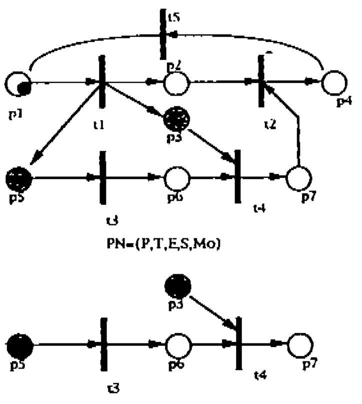
Let N=<PN,Mo> be a marked net, where P={p1,p2,p3,p4}, T={t1,t2}, then

$({}^{0}(t1) \cup (t1)^{0}) \cap ({}^{0}(t2) \cup (t2)^{0}) = Po = \{p2\}$.



Definition 3.8 : [21] Let <PN_{RP},M_{RP}> and <PN,M> be two marked nets. PN_{RP} is a subnet of PN and is denoted $PN \sqsupseteq PN_{RP}$ if and only if, $P \sqsupseteq P_{RP}$, $T \sqsupseteq T_{RP}$,$E_{rp} = E \cap (P_{RP} x T_{RP})$,$S_{RP} = S \cap (T_{RP} x P_{RP})$ and M M_{RP}. a place $p \subseteq P_{RP}$ is a input door (ID)(resp output door OD)of RP iff there exists $t \in {}^{0}p$(resp $t \in p^{0}$) such that $t \in T_{RP}$.

Example:



○ :Output door(OD) ● :Input door(ID)

RP=(PRP,TRP,ERP,SRP,MRP)

Definition 3.9 : [13] Let $PN_1=<PN_{RP1}, M_{RP1}>$, $PN_2=<PN_{RP2},M_{RP2}>$ be two marked nets. PN1,PN2 are composable via Po iff $\cap PN_{RP1}$ PN_{RP2} =Po and $T_{RP1} \cap T_{RP2} = \phi$

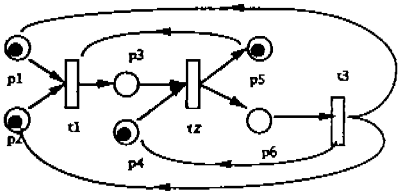
Definition 3.10 : [13] Let $PN_1=<PN_{RP1},M_{RP1}>$, $PN_2=<PN_{RP2},M_{RP2}>$ be two marked nets, composable via Po. The net PN=<PN,M> obtained by the composition PN_1,PN_2 via Po is defined by: $P=PN_{RP1} \cap PN_{RP2}$=Po and $T=T_{RP1} \cap T_{RP2}$.

3.2 Petri Nets like as logic program

The structure of Petri nets(PN) displays the interrelationships between rules and predicates as specified by the logic program.[21]

We concentrate on logic programs consists of function-free Horn rules $r_1,r_2,\cdots,rm$, and each rule r(also called a transition) is of the form $B_1, \cdots,B_q:=A_1,\cdots,A_p$, where A_i,B_j are atomic formulas,:= denotes implication,$p \geq 0$ and $q \geq 0$. This clause means that if A_1 and $\cdots$ and A_p then B_1 or$\cdots$or B_q. The right-hand side and the left-hand side are each called the condition and the cinclusion. A Horn clause is one that is either q=1 or q=0, where $A_i,1 \leq i \leq p$,are nonnegated.

Example : Suppose we have the following PN:



The logic program corresponding to the above net will be as follows:

T={t1,t2,t3},

P={p1,p2,p3,p4,p5,p6},

t1=([p3],{t1[1]≥t3[2]}),

t2=([p5,p6],{t1[2]≥t2[1]}),

t3=([p1,p2,p4],{t2[2]≥t3[1]})

E(t1)={p1,p2,p5},

E(t2)={p3,p4}, E(t3)={p6},

S(t1)={p3}, S(t2)={p5,p6},

S(t3)={p1,p2,p4},

and the logic program of this net is:

t1:(p3) ← (p1,p2,p5),

t2: (p5,p6) ← (p3,p4),

t3: (p1,p2,p4) ← (p6).

In addition, the tableau of this net is:

	A1	A2
t1	p1p2p5	p3
t2	p3p4	p5p6
t3	p6	p1p2p4

4. The Composition of Transition in the Petri Nets

In this section, we propose a set of reduction rules which combines two transitions into one or superposes two transitions over the other, or reduces the common places based on EE-join ofRFs which is produced in the previous section. And a set of reduction rules are called "Functional reduction". Functional reduction of two RFs are classified into the following three kinds:

(1) Strong composition: combines two relations to new one after reduction of the common places,

(2) Superposition:superpose two relations to one after superposition of the common places,

(3) Equivalence: reduces the common places.

Let $<PN_{RI'},M_{RI'}>$ be a subnet of <PN,Mo>=< P,T,E,S,Mo>, $PN_{RI'}$ is decomposed as T1(V) and T2(W). Let T1(V):1->2 with V=

$\{A_1A_2\}$, T2(W):1−>2 with W=$\{A_1A_2\}$ be two transitions. For simplify our research, Ordinary Petri Nets are considered here, and a marked net <P N,Mo> is replaced by <PN“, Mo”>.

4.2.1 Strong omposition(CF)

Definition 4.1 : For any tuples $t1_1$ in V, and $t2_1$ in W, if $t1_1[T1.A_2]$ $t2_1[T2.A_1]$ and the following conditions are verified:

(1) ∀ $t1_1$,$t2_1$ T", pki∈ P, (j≥1),

- dom($t2_1$) ran($t1_1$)={pki},≑θ
- (dom(t2)⊆ran(t1))∨ran(t1)⊆ dom(t2))

(2) ID=dom($t1_1$), OD=ran($t2_1$),

(3) dom($t2_1$)∩ran($t1_1$)∩ran($t2_1$)≑(ran($t1_1$)∩dom($t2_1$))

then a strong composition made a new RF of transition T(X):1−>2, with X={Y1Y2}, t∊X:

- t"[Y1] = dom($t1_1$)+dom($t2_1$)−pki
- t"[Y2] = ran(t2)+ran(t11)−pki

so a reduced net PN" is as follows:

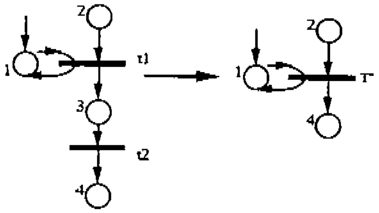
(1) P"=P−pki

(2) T"=T−$t1_1$−$t2_1$+t"

(3) Mo"(p")=Mo(p)+ M(pki)+ S(pki,$t1_1$)−E(pki,t21).

Example :

(1)



T1(V)

A1	A2
p1 p2	p3p1

T2(W)

A1	A2
p3	p4

T1(V) θCF T2(W)

Y1	Y2
p1 p2	p1p4

By the Strong composition CF, we get:

P"={p1,p2,p4}, T"={t"}

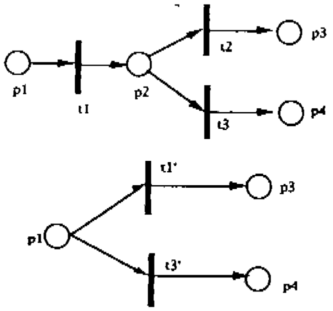
t"[Y1]=p1+p2+p3-p3=p1+p2,

t"[Y2]=p4+p1+p3-p3=p1+p4

so,t":{p1,p2} → {p1,p4}

In this net, t1 is X mode, t2 is T mode and an agglomerated transition t' is X mode

(2)



T1(V)

A1	A2
p1	p2

T2(W)

A1	A2
p2 p2	p3 p4

T1(V) θCF T2(W)

Y1	Y2
p1 p1	p3 p4

By the strong composition CF, we get:

P"={p1,p3,p4}.

T"={t"1,t"3}.

t"1[Y1]=p1+p2-p2=p1,

t"1[Y2]=p3+p2-p2=p3

t"3[Y1]=p1+p2-p2=p1,

t"3[Y2]=p4+p2-p2=p4

M"(p)=0.

So, t"1:{p1}→{p3}, t"3={p1}→ {p4}.

4.2.2 Superposition(CE)

Definition 4.2:For any two tuples $t1_1$ in V,and $t2_1$ in W, if $t1_1[T1.A_1]\theta t2_1[T_2.A_1]$ and if the following conditions are verified:

(1) Let∀$t1_1$,$t2_1$ T" , pk,pk'∈P, (1≤n,m)

- (dom($t1_1$)∩dom($t2_1$)={pkn})≑θ∧(ran($t1_1$)∩ran($t2_1$)={pk'm}≑θ),

(where≥n 1, m≤1)

(2) $ID=dom(t1_1)$, $OD=ran(t2_i)$,

A new RF:T(X):1→2, with X={Y1Y2},where t"∈X, was derived from a superposition:

- $t\alpha"[Y1]=dom(t1_1)+dom(t2_1)-k_n(Pk_n)$
- $t\ "[Y2]=ran(t1_1)+ran(t2_1)-k_m(Pk_m)$

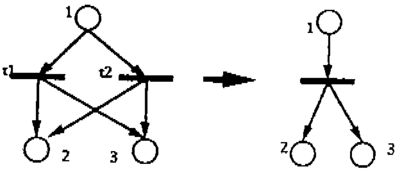
so reduced net PN" is as follows:

(1) P"=P

(2) $T"=T-t1_1-t2_1+t"$

(3) Mo"(p")=Mo(p)

Example :



T1(V)

A1	A2
p1	p2p3

T2(W)

A1	A2
p1	p2p3

T1(V) ΘCE T2(W)

Y1	Y2
p1	p2p3

By the superposition of composition CE, we get:

P"={p1,p2}, T"={t"}

t"[Y1]=p1+p1-p1=p1,

t"[Y2]=p2+p3+p2+p3-p2-p3=p2+p3

so,t":{p1} -> {p2,p3}

4.2.4 Epuivalence(EQ)

Definition 4.3 : For any two tuples $t1_1$ in V, and $t2_1$ in W, if $t11[T1.A2]\theta t2_1[T2.A_1]$ and if the following conditions are verified:

(1) Let $\forall t1_1,t2_1 \in T$, pk_j P,

$(dom(t1_1) \cap ran(t2_1)) = (dom(t2_1) \cap ran(t1_1)) = \{pkj\} \neq \theta$

(2) $S(pk_j,t1_1)-E(pk_j,t2_1)=h_j \geq 0$, $(j \geq 1)$

(3) $M(pk_j)- \Sigma k'_j E(pk_j,t_i1) \geq 0$

Two new RFs of transition $T3(X1):_{1->1}$ with X={Y1Y2}, where t"∈X and T4(X2):1→2 with X={Y1Y2}, where t"∈X were derived from a equivalence:

$t"1[Y_1]=dom(t1_1)-(M(pk_j)-\Sigma E(pk_j,t2_1))pkj$, $(1 \leq_j)$

$t"1[Y_2]=\ ran(t1_1)-(M(pk_j)-\Sigma E(pk_j,t2_1))pk_j$

$t"2[Y_1]=dom(t2_1)-(M(pk_j)-\Sigma E(pk_j,t1_1))pk_j$, $(1 \leq j)$

$t"2[Y_2]=\ ran(t2_1)-(M(pk_j)-\Sigma E(pk_j,t1_1))pk_j$

so the reduced net PN" is as follows:

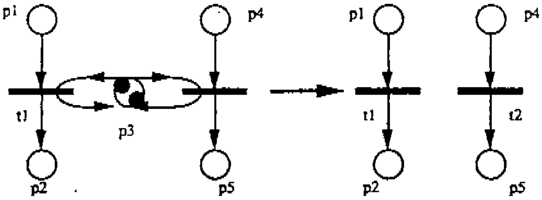
(1) P"=P, $(h_j \leq\ 0)$

$P-pk_j$,(otherwise, j≥1)

(2) $T"=T-t_i+t"_i$,(where i≥2)

(3) Mo"(p")=Mo(p)

Example :



T1(V)

A1	A2
p1p3	p2p3

T2(W)

A1	A2
p4p3	p3p5

T1(V) ΘEQ T2(W)

T3(X1)

Y1	Y2
p1	p2

T4(X2)

Y1	Y2
p4	p5

By the equivalence EQ, for any tuples t1 in V, and t2 in W, we get:

P"={p1,p2,p4,p5},

T"={t',t"}

t'[Y1]=p1+p3-p3=p1,

t'[Y2]=p2+p3-p3=p2

$t^{*}[Y1]=p4+p3-p3=p4$,
$t^{*}[Y2]=p5+p3-p3=p5$

5. Properties of the Petri Nets and Reduction algorithm

In this section, we consider properties of the Petri nets and reduction algorithm which are proposed in section 4, and comparison with other approaches.

5.1 Properties of the Petri nets

Where pratical applications are concerned, it is necessary to avoid the analysis of a large complex Petri net. It is convenient to proceed by the reduced net (PN*) and reducible module (RP) if they are homogeneous with the original net (PN).

About the properties of the Petri nets, the liveness and safety are defined
by the firing sequence of transitions. If a reduction dose not change any
firing sequence, the liveness is not changed. For prove this property, we concerned as follows three theorems:

Theorem 5.1: A Petri net is live if and only if its reduced net is live.

Proof:The liveness of Petri nets is defined by the firing sequence of transitions and the number of tokens received and produced by transitions, so if a reduction dose not change any firing sequence and the number of tokens, the liveness is not changed.[7] Therefore, we prove this theorem by checking whether a reduction changes the firing sequence and number of tokens for each reduction rule.

Strong Composition(CF):In the firing sequence of transition, a reduction of CF by an agglomerated transition is a replacement of the subfiring sequence by an agglomerated transition, based on the EE-join of RF.

Therefore, the reduction dose not change the firing sequence. Also, the number of tokens received and produced by the agglomerated transition is equal to that number of the RP.

Superposition(CE):A reduction of the CE eliminates nondeterministic situation. Transitions having the same input and output place through same weights of arcs and merged in an agglomerated transition. The agglomerated transition has the same input and output place; and the weight arcs is not changed. Therefore, the reduction dose not change the firing sequence and the number of token entering ID and leaving OD.

Equivalence(EQ):The reduction rule of EQ dose not delete any transition or replace any transition by an agglomerated transition. Therefore, the firing sequence is not changed through the reduction. In a RP, the number of tokens in reduced common place. The transition can be fired without considering the state of reduced common places. Therefore, the reduction dose not change the number of tokens entering and leaving the RP.

Theorem 5.2:A Petri net is bounded if and only if its reduced net is bounded.

Proof:The boundedness of Petri net is determined by the number of tokens which flow in the net. If the number of tokens is not changed, the initial property of boundedness is preserved. In the Theorem 5.1, we have seen that all of the reduction rules do not change the number of tokens which flow in the net.

Theorem 5.3:A Petri net is properly terminating if and only if its reduced net is properly terminating.

Proof:In the same way as the earlier theorems, the property of proper termination is preserved because the reduction dose not change the number or direction of flow of tokens in the net.

5.2 Reduction algorithm

After the results contained in section 4, we propose a reduction algorithms for functional reduction, and prove that it can reduce any Petri nets to an elementary one.

Reduction Algorithms

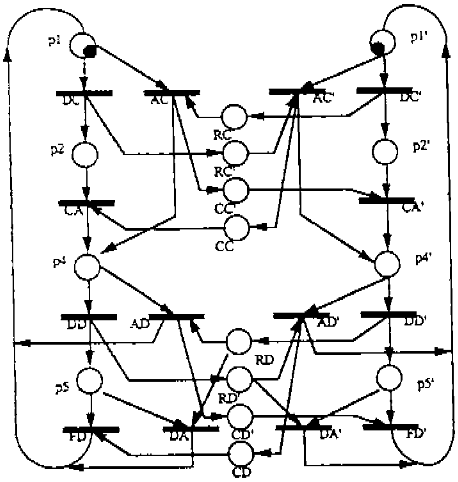
```
 1st step/*search RP in the nets*/
 begin
   input:= <PN,M0>, a Petri net
   i:=0; fini='F';<PNi,Mi>=<PN,M0>;
   repeat
     begin
       if PNi is CF reducible then T:=3
         else if PNi is CE reducible then T:=2
         else if PNi is EQ reducible then T:=1
             else fini:='T';
 end-begin,if fini='F' then do 2nd step;
 i:=i+1;
 until fini = 'T';
 ⟨PN*, M*⟩=⟨PNi, Mi⟩; output ⟨PN*, M*⟩
end.

2nd step/* reduction step*/
begin
  case T:[1,2,3]
    1:let   <PNi+1,Mi+1>   be   the   result
                applying EQ to <PNi,Mi>;
    1:let   <PNi+1,Mi+1>   be   the   result
                applying CE to <PNi,Mi>;
    1:let   <PNi+1,Mi+1>   be   the   result
                applying CF to <PNi,Mi>;
end.
```

6. Illustractive example and comparations with other approachs

6.1 Illustartive example

In this section, we informally introduce functional reduction through example. The example net which models aconnection and disconnection model of ECMA protocol in which is presented in [2,7].



The relational table of this net and reduced net by functional reduction as follows:

Tr.	input place	output place
DC	P1	P2RC'
AC	P1RC	P4CC'
CA	P2CC	P4
AD	P4	P5RD
DD	P4RD	P1CD'
FD	P5CD	P1
DA	P5RD	P1
DC'	P1'	P2'RC
AC'	P1'RC'	P4'CC
CA'	P2CC'	P4'
AD'	P4'RD'	P1'CD
DD'	P4'	P5'RD
FD'	P5'CD'	P1'
DA'	P5'RD'	P1'

(a) Relational table of example net

Tr.	input place	output place
DC	P1	P2RC'
AC	P1RC	P5RD'CC'
CA	P2CC	P5RD'
AD		
DD	P2CCRD	P1CD'
FD	P5CD	P1
DA	P5RD	P1
DC'	P1'	P2'RC
AC'	P1'RC'	P5'RDCC
CA'	P2CC'	P5'RD
AD'	P2CC'RD'	P1'CD
DD'		
FD'	P5'CD'	P1'
DA'	P5'RD'	P1'

(b) Relational table of reduced net

The relational table of reduced net which is reduced by the Berthelot method is:

Tr.	input place	output place
DC		
AC	P1P1'	P4P4'
CA		
AD	P4RD	P1CD'
DD	P4	P5RD'
FD	P5CD	P1
DA	P5RD	P1
DC'		
AC'		
CA'		
AD'	P4'RD'	P1'CD
DD'	P4'	P5'RD
FD'	P5'CD'	P1'
DA'	P5'RD'	P1'

And, we can summarize a table which is explained number of transition and places of the nets(Original example net, reduced net whuich is reduced by Berthelot method, and by us) like as follows:

	ORIGINAL NET	BERTHELOT METHOD	PROPOSED METHOD
T	14	9	12
P	16	10	14

(P: NO. OF PLACES, T: NO. OF TRANSITION)

This means that proposed method has not strong power than Berthelot method but its an usual reduction method for analysis or verification the protocol.

6.2 Comparisons with other approaches

Our reduction has weaker reduction power than one of Berthelot[2], and also the reduction rules proposed(CF,CE,EQ) are very similar to some ones of [2], but CF comprehend some rules of [2]. Lee[7] has four rules:(1)RSN-1:this rule consists of RSN-1P and RSN-1T,(2)RSN-2P,(3) RSN-3P and (4)RSN-4T. In these four rules, RSN-1P,RSN-2P and RSN-3P can be covered by the join of CF and CE. And rules RSN-1T and RSN-4T can be covered by the join of EQ. Ramamoorthy[12] has two rules:decision free SWBM and non-decision free SWBM. The CF cover all the SWBM's topologies. In addition, Tu [16] proposed 10 rules which have 4 rules of Berthelot. Rule1-7 and 10 are covered by CF,CE and EQ.

7. Conclusion

In this paper,we have suggested a framework for transformating a large class of Petri nets into simply reduced nets. The concerned class includes all reduction rules which may be transformed into functional expressions, that is what we call the functional reduction class. We claims that the approach is easy to implement for transformation and shows a simpler reduction procedure. The advantages are as follows:

1) It is easy to find the reducible module by the relational tableau of transitions.
2) The reduction rules is only one type (composition) and is automatically showed the results places(input,output places) with the number of arcs.
3) The agglomerated transition mode can be expected.
4) A Petri nets explained by the formal definition using relational scheme in relational database.

Acknowledgements

We are sinceraly grateful of the 3 anonymous referees for their remarks improved the paper.

References

[1] C.Andre, F. Boeri, J. Marin, Synthese et Realisation des Systemes logiques a Evolution Simultanee, RAIRO, 10, pp. 67-86, 1976

[2] G.Berthelot, Transformations de reseau de Petri, TSI, 4 (1), pp. 91-101, 1985

[3] G.W.Brams, Reseaux de Petri theorie et Pratique, Marsson, Paris, pp. 9-22, 1982

[4] G. Chehaibar, Validation of Phase-executed protocols Modelled with Coloured Petri nets, Proc. application and theory of Petrinets, pp. 84-103, 1990

[5] Esparza, M. Silva, Top-down Synthesis of Live and Bounded Free Choice nets, Proc. application and theory of Petri nets, pp. 63-83, 1990

[6] G. Gardarin, C. Maindreville, Evaluation des Programmes Logiques Recursifs par des

Fonctions Recurrents, TR-141, MASI, Univ. P.et M. Curie, 1986

[7] K. H. Lee, J. Favrel, Hierarchical Reduction Method for analysis and Decomposition of Petri nets, IEEE Tr. on system, man and Cyb., smc-15(2), pp. 272-280, 1985

[8] G. Memmi, J. Vautherin, Analyzing nets by the Invariant Methd, LNCS No.254,Berlins, S-V., pp. 300-336, 1987

[9] G. J Nutts, Evaluation Nets for Computer Systems Performance Anaysis, Proc. Fall Joint Computer Conf., pp. 279-286,1982

[10] A.Ola, G. Ozsoyoglu, A Family of Incomplete Relational Database Models, Proc.15th Inter. Conference on VLDB, pp. 23-31, 1989

[11] J. L. Peterson, Petri Net Theory and the Modeling of System, Prentice-Hall, NJ, 1981

[12] C. V. Ramamoorthy, Y. Yaw, W. T. Tasi, A Petri Net Reduction Algorithms for Protocol Analysis, Proc. Communnications Architecture & Protocols, pp. 157-166, 1986

[13] Y.Souissi, On liveness preservation by composition of nets via a set of places,Proc. application and theory of Petri nets,pp. 104-122, 1990

[14] I.Suzuki, T.Murata, A Method for Stepwise Refinement and Abstraction of Petri Nets, Journal of Computer and System Science, 27, pp. 51-76, 1983

[15] R. Valette, Analysis of Petri Nets by Stepwise Refinements, Journal of Computer and System Sciene, 18, pp. 35-46, 1979

[16] S. TU, S. M. Shatz, T. Murata, Applying Petri Net Reduction to Support Ada-Tasking Deadlock Detection, Proc. Distributed Computer Systems, pp. 96-103, 1990

[17] G. Berthelot, R. Terrat, Petri nets Theory for Correctness of Protocols, IEEE Trans. on Communication Vol. COM30, No.12,pp. 2497-2505, 1982

[18] C.Delobel, R. G. Casey, Decomposition of a diatbase and Theory of Boolean swiching functions,IBM Journal of Research and Development, pp. 374-386, 1973

[19] J. D. Ulm]an, Principles of Databaseseand Knowledge-base Systems, Vol. I. Computer Sci. Press, Rockville,Maryland, 1988

[20] M. T. Ozsu, P. Valduriez, Principales of Distributed Database Systems, Prentice-Hall, N. Y, 1991

[21] J. K. Lee, On the Semi-join for composition of Transition in Petri Nets, Proc. of JTC-CSCC'93 ,pp.306-311, 1993

[22] J. K. Lee, Functional Reduction of the General Petri Nets, Research Bull. of inst. of Basic Sci., Changwon Nat. Univ.,vol. 4, pp. 141-156, 1933

이 종 근

1974년 숭실대학교 전산학과 졸업(학사)
1978년 고려대학교 경영대학원 경영학전공(경영학 석사)
1986년 숭실대학교 대학원 전산과 졸업(공학 석사)
1992년 USTL(프)전산과 박사 과정 수료
1994~77년 육군본부 중앙전산소 근무
1977~83년 (주)대우근무
1983~현재 국립창원대학교 전산과 교수
관심분야 : Petri Net 이론 연구, 성능분석 및 평가, 소프트웨어 복잡도분석, Protocol검증