Proc. 3rd IEEE Int. Workshop on Petri Nets and Performance Models
PNPM89, Kyoto, December 1989.

AUTOMATED CONSTRUCTION OF COMPOUND MARKOV CHAINS FROM GENERALIZED STOCHASTIC HIGH-LEVEL PETRI NETS

Juan A. Carrasco

Departament d'Enginyeria Electrònica
Universitat Politécnica de Catalunya
Diagonal 647, 08028-Barcelona, SPAIN

Abstract

A new type of Petri nets: Generalized Stochastic High-Level Petri nets (GSHLPN's), collecting the qualities of GSPN's and SHLPN's, is presented. The automated construction of compound continuous-time Markov chains (CTMC's) from GSHLPN's is also considered. A formalism for the description of compound markings allowing a symbolic firing of the net to obtain a compound CTMC with correct state grouping is derived. The construction of the compound CTMC requires an algorithm to test the equivalence of compound markings. It is shown that, in the general case and for bounded number of rotation groups, the problem is polynomially equivalent to GRAPH ISOMORPHISM, a problem whose classification in the NP world is currently open.

1. INTRODUCTION

The success of Petri nets (PN's) to represent and analyze the qualitative behaviour of complex systems with synchronization and concurrency has motivated the introduction of stochastic Petri nets (SPN's) for the quantitative analysis of such systems [1], [2]. SPN's are PN's with exponentially distributed firing times associated to transitions. Bounded SPN's are isomorphic to finite continuous-time Markov chains (CTMC's). Therefore, SPN's do not provide more modeling power than CTMC's, but they are indeed an appropriate specification methodology for Markov models having CTMC's of large size. Since the introduction of SPN's, there have been a number of interesting developments. Some of them extend the modeling power of SPN's. Among these, we have the extended stochastic Petri nets [3] and the stochastic activity networks [4], both of which allow arbitrary firing time distributions. Other efforts are oriented to the development of more expressive types of Petri nets with the same modeling power as SPN's.

Generalized stochastic Petri nets (GSPN's) [5] extend SPN's by introducing inhibitor arcs and immediate transitions. This class of nets has recently be reextended [6] by allowing probabilities and priorities to be added to immediate transitions. GSPN's not only are more expressive than SPN's, but also support the specification of models with smaller CTMC's when some activities are much faster than others so that the first can be modeled by immediate transitions firing in zero time. Efficient algorithms for the generation of CTMC's from GSPN's have recently been presented [7].

Some systems are composed of sets of processing elements with identical behaviour. These systems have been called homogeneous. Since the size of the CTMC is the limiting factor for the numerical analysis of the model, it is highly desirable to take advantage of the symmetries exhibited by those systems to reduce the size of the CTMC. Although SPN's provide some support, they fail to exploit the symmetries in the behaviour of the system when, as it is often the case, processes have relationships. Since the tokens in a SPN do not convey any information, except the place where they are, the relationships can only be expressed by individualizing the processes. This not only yields CTMC's of undue size, but also enlarges the Petri net, making more arduous the specification of the model. In order to overcome this problem Lin and Marinescu [8], [9] have introduced stochastic high-level Petri nets (SHLPN's). SHLPN's are high-level Petri nets (HLPN's) with constant, but possibly marking dependent, firing rates attached to transitions. HLPN's are an extension of standard PN's in which tokens have colours, arcs have expressions with variables yielding bags of colours, and transitions have predicates (guards) on the variables used in the expressions of the surrounding arcs. HLPN's were proposed by Jensen [10] to summarize the qualities of predicate transition nets and coloured Petri nets. Similar to SHLPN's, stochastic coloured Petri nets, were proposed by Zenie [11]. SHLPN's have been shown to be a powerful methodology for the specification of models of homogeneous systems [8], [9], [12]. They allow the specification of complex homogeneous systems in a more succinct and readable way than standard PN's and yield CTMC's of much smaller size.

Although some work has been done towards the automated analysis of HLPN's [13], it is apparent that some problems have not yet received a satisfactory solution. These problems are identified for SHLPN's in [8], [9]: derivation of compound markings so that compound CTMC's have a correct state grouping, computation of transition rates between compound

TH0288-1/89/0000/0093$01.00 © 1989 IEEE

markings, and efficient testing of the equivalence of compound markings.

The contents of this paper is as follows. Section 2 introduces a new type of stochastic Petri nets: Generalized Stochastic High-Level Petri Nets (GSHLPN's), combining the qualities of SHLPN's and GSPN's. Section 3 presents a methodology to derive a symbolism for the description of compound markings which allows the symbolic firing of the net to obtain compound CTMC's with correct state grouping. It is apparent that the determination of the equivalence of compound markings is a difficult combinatorial problem. In section 4 we analyze the complexity of the problem from the point of view of the NP-completeness theory. Finally, conclusions and suggested directions for future work are presented in Section 5.

2. GSHLPN's

Generalized stochastic high-level Petri nets are obtained from SHLPN's by the introduction of immediate transitions with priorities, inhibitor arcs, and cases. This results in a better framework for the modeling of conflict and contention than SHLPN's. We first give a formal definition of GSHLPN's. The modeling power of the nets is illustrated next with an example.

2.1. Definitions

A generalized stochastic high-level Petri net is a 9-tuple:

$$GSHLPN = (S, P, T, I(), TO(), CO(), H(), M_0) ,$$

where:

S is the set of token types. Each token type $s \in S$ has a list of attributes with finite domains which, without lost of generality, are assumed to be of the form $D = \{1, ..., |D|\}$. A token of type s is denoted by $<s, a_1, ..., a_k>$, where a_i is the value of the ith attribute of s.

P is the set of places.

T is the set of transitions. As in GSPN's, there are two types of transitions: timed transitions and immediate transitions. Each timed transition t has associated a, possibly marking dependent*, firing rate $\lambda(t)$ and each immediate transition t a priority level $\pi(t)$. A transition t may have associated a predicate $P(t)$, i.e., a logical expression over the free variable set of t (defined in the next paragraph). A transition may have cases or not. The set of cases associated to a transition t is denoted by $C(t)$. Each case $c \in C(t)$ has a, possibly marking dependent*, probability $q(c)$.

* this dependency is restricted to be expressed in terms of the number of tokens of given type in each place, i.e., ignoring token attributes.

$I()$ is the input function which assigns sets of places to transitions. Each pair (p, t) with $p \in I(t)$ is called an input arc and p is said to be an input place of t. Each input arc (p, t) has associated a set of symbolic input tokens $s_I(p, t)$. The values of the attributes of the symbolic input tokens of a transition are specified by independent free variables constituting the *free variable set* of the transition.

$TO()$ is the transition output function which assigns sets of places to transitions without cases. Each pair (t, p) with $p \in TO(t)$ is called a transition output arc and p is said to be an output place of t.

$CO()$ is the case output function which assigns sets of places to cases. Each pair (c, p) with $p \in CO(c)$ is called a case output arc and p is said to be an output place of c.

$H()$ is the inhibitor function which assigns sets of places to transitions. Each pair (t, p) with $p \in H(t)$ is called an inhibitor arc and p is said to be an inhibitor place of t.

M_0 is the initial compound marking.

Transition output arcs, case output arcs and inhibitor arcs have associated bags of symbolic tokens $b_{TO}(t, p)$, $b_{CO}(c, p)$, and $b_H(t, p)$, respectively. The attributes of the tokens contained in these bags are either specified by expressions on the free variable set of t ($c \in C(t)$ for $b_{CO}(c, p)$) or left unspecified.

An *atomic token* is a token with all its attributes bound to values of their respective domains. An *atomic marking* m of the net is an assignment of bags of atomic tokens to places of the net. A *compound marking* M is an assignment of bags of symbolic tokens to places, where the attributes of the tokens are specified by expressions over a set of free variables taking values in the respective attribute domains. A compound marking M represents the subset of atomic markings $S(M)$ obtained by binding the free variables with values of the attribute domains. Two compound markings are *equivalent* if they represent the same subset of atomic markings.

In the graphical representation of the net, places are represented by circles, timed transitions by rectangles, immediate transitions by thin bars, and cases by ovals. Input arcs have an arrow in the transition side, transition and case output arcs have an arrow in the place side, and inhibitor arcs have a small circle in the transition side. Cases are connected to their transitions by lines.

The facts that in GSHLPN's tokens have attributes and transitions have predicates make necessary a more detailed definition of the enabling and firing rules than for GSPN's. The rules will be defined in terms of atomic markings. A *firing set* F of a transition t in an atomic marking m is a selection for each $p \in I(t)$ of a subset of tokens $F(p) \subset m(p)$ binding $s_I(p, t)$ satisfying the predicate $P(t)$ and such that for no $p \in H(t)$ there exists a subset of $m(p)$ binding $b_H(p, t)$ with the values imposed

by F to the variables in the free variable set of t. Note that $F(p)$ is a subset, and replicated elements in $m(p)$ are seen as different for the determination of the firing sets. Consider, for instance, the transition and atomic marking shown in Figure 1. The firing sets of the transition are:

$$\begin{array}{l} F_1\colon\ F_1(P2)=\{\langle B, 1, 2, 3\rangle_1\},\ F_1(P3)=\{\langle C, 2\rangle, \langle C, 3\rangle\} \\ F_2\colon\ F_2(P2)=\{\langle B, 1, 2, 3\rangle_2\},\ F_2(P3)=\{\langle C, 2\rangle, \langle C, 3\rangle\} \\ F_3\colon\ F_3(P2)=\{\langle B, 1, 3, 4\rangle\},\ F_3(P3)=\{\langle C, 3\rangle, \langle C, 4\rangle\} \\ F_4\colon\ F_4(P2)=\{\langle B, 2, 4, 5\rangle\},\ F_4(P3)=\{\langle C, 4\rangle, \langle C, 5\rangle\} \end{array} \quad (1)$$

If the atomic token in place $P1$ were $\langle A, 1\rangle$ instead of $\langle A, 6\rangle$, the only firing set of the transition would be F_4. Two firing sets F_i, F_j of a transition t are *conflicting* if for some $p \in I(t)$, $F_i(p) \cap F_j(p) \neq \emptyset$. A *firing distribution* is a maximum cardinality set of non-conflicting firing sets. The firing distributions for the transition $T1$ of Figure 1 are:

$$\begin{array}{l} FD_1 = \{F_1, F_4\} \\ FD_2 = \{F_2, F_4\} \end{array} \quad (2)$$

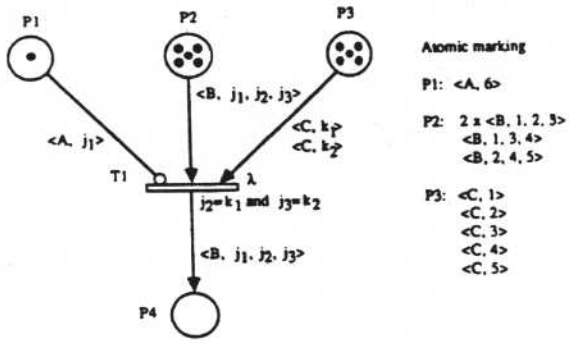


Figure 1. Example to illustrate enabling and firing rules in GSHLPN's.

A transition t (immediate or timed) is *firable* in an atomic marking m if t has some firing set in m. Firable timed transitions are *enabled* only if no immediate transition is firable. When there are firable immediate transitions only those with the highest priority are enabled. Only enabled transitions actually fire. Immediate transitions fire instantaneously in each firing set with the same probability. Timed transitions fire in each firing set belonging to some firing distribution with rates which are computed by assigning the same probability to each firing distribution and a rate $\lambda(t)$ to each firing set of the distribution. The firing of a transition t in a firing set F removes from each input place p the set of tokens $F(p)$. Cases have the same semantics as in stochastic activity networks [4]: if t has cases, one is sampled according to the probabilities $q(c)$. Output tokens are added to each transition (case) output place p according with the bags of symbolic tokens $b_{TO}(t, p)$ $(b_{CO}(c, p))$. Specified attributes are computed using the expressions defining them with the values for the free variables bound by F. For unspecified attributes, a value from the domain of the attribute is taken at random. For instance, the transition $T1$ in Figure 1 has the firing distributions (2) and a probability 1/2 is assigned to each of them. Given a firing distribution, the transition fires in each firing set of the distribution with rate λ. Then, $T1$ fires in F_1 and F_2 with rate $\lambda/2$ and in F_4 with rate λ. If $T1$ were immediate it would fire in each firing set (1) with probability 1/4. The firing of $T1$ in the firing set F_4 removes $\langle B, 2, 4, 5\rangle$ from $P2$, $\langle C, 4\rangle$ and $\langle C, 5\rangle$ from $P3$, and adds $\langle B, 2, 4, 5\rangle$ to $P4$.

The original definition of GSPN's [5] allowed nets with reachable markings enabling several immediate transitions and left to the user the responsability of defining the random switches required to disambiguate the model. Recognizing the practical problems of the approach, GSPN's were redefined in [6]. The new definition imposes some structural restrictions to the nets and allows priorities for immediate transitions. Weights are associated to immediate transitions as required in a computer-aided process to disambiguate the net. In GSHLPN's probabilistic selection is modelled by cases and the net has to be defined so that no reachable marking enables immediate transitions in the same immediate subnet of the corresponding priority. Immediate subnets of a given priority are [7] the connected components obtained by removing in the original net timed transitions and immediate transitions of lower priority. This approach has the advantage of allowing a highly efficient construction of reachability graphs with reduced number of vanishing states [7].

The firing policies adopted for immediate and timed transitions have been chosen to provide a useful default. Immediate transitions are commonly used as contention points and the firing rule adopted for this type of transitions assumes a random policy for the resolution of conflicts among contending processes. Timed transitions are commonly used as resource usage points and the firing rule adopted for this type of transitions implements a random policy within the constraint of optimal use of resources. The modeling of more complex contention and resource usage policies would require the use of subnets with combinations of immediate and timed transitions.

2.2. An example

In order to illustrate the expressive power of GSHLPN's we give an example in Figure 2. In this figure $\oplus$ represents the addition module the size of the domain of the involved attributes. The model is a slightly modified version of one described in [8], [9] and describes a shared memory multiprocessor system in which a set of identical processors with their own local memory access a set of common memory modules through a set of buses. A token-passing protocol is used in each bus for access control. The token is circulated among processors and its reception gives the processor the right to use the bus for accessing a common memory module. The execution of a processor alternates between the private and the common domains. In the private domain the

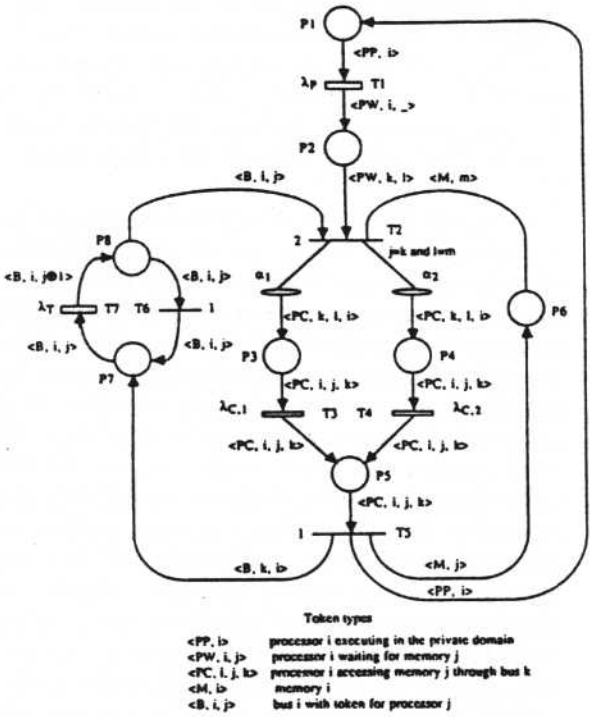


Figure 2. GSHLPN for a shared memory multiprocessor system with a token passing protocol.

processor accesses its local memory. In the common domain, it accesses a given common memory module taken at random from the set. The duration of an execution in the private domain is exponentially distributed with mean $1/\lambda_P$. The duration of an execution in the common domain has an hyperexponential distribution with parameters α_1, $\lambda_{C,1}$, and α_2, $\lambda_{C,2}$. The token pass delay has an exponential distribution with mean $1/\lambda_T$.

We follow the formal definition and give to each attribute of the sets of input tokens associated to a given transition a unique free variable, even if they are required to be equal by the transition predicate (see transition *T2*). An alternative would be to use the same free variable for the attributes required to be equal by the predicate. Unspecified attributes of output and inhibitor tokens are represented by an underscore. In order to model the contention for tokens and memory modules, we use a immediate transition *T2*, which is given priority over the immediate transition *T6* modeling the start of a token pass. Thus, if a processor waiting for an access to a memory module receives a token, it captures the token and starts the execution in the common domain if the memory module is free (transition *T2*), and sends the token to the next processor if the memory module is busy (transition *T6*). The execution in the common domain is modeled by introducing cases in the immediate transition *T2* with probabilities equal to the selection parameters of the hyperexponential distribution, and followed by places and timed transitions with the rates associated to each mode of the distribution.

GSHLPN's have over GSPN's the same advantages than SHLPN's have over SPN's: compactness and reduced size of the state-level model if a compound CTMC is constructed instead of the atomic CTMC. The introduction of immediate transitions makes also possible an economical modeling of processes which, compared to others, complete in a negligeable amount of time. For instance, using SHLPN's instead of GSHLPN's, the previous example would have required the use of timed transitions with very high rates for the modeling of the contention of processors for busses and memories, and the decission about whether to pass the token to the next processor. This not only would increase the size of the compound CTMC but also would stress numerical methods during its solution.

3. A SYMBOLISM FOR COMPOUND MARKINGS

Assume that the atomic reachability graph obtained by firing the GSHLPN from a given atomic marking of $S(M_0)$ enabling only timed transitions is finite, strongly connected, and contains all the atomic markings in $S(M_0)$. Under these conditions an ergodic atomic CTMC $(\Omega, X(t))$ isomorphic to the GSHLPN can be obtained in two steps as follows. In the first step, the atomic reachability graph G is constructed by firing the net from any atomic marking $m_0 \in S(M_0)$, using the rules described in the previous section. G has two types of nodes: tangible markings (enabling only timed transitions) and vanishing markings (enabling only immediate transitions), and associates rates to arcs coming out of tangible markings and probabilities to arcs coming out of vanishing markings. In the second step, vanishing markings are reduced as shown in [5] to obtain $(\Omega, X(t))$.

We are however interested in the construction of a compound CTMC $(\Omega', X'(t))$ whose states be tangible compound markings. The compound CTMC has to be a correct grouping of $(\Omega, X(t))$, i.e., has to satisfy the following requisites [14]:

1) the sets of atomic markings associated to the compound markings in Ω' are a partition of Ω

2) for all $M_1, M_2 \in \Omega'$, $M_1 \neq M_2$ and all $m_1 \in M_1$:

$$\lambda_{m_1, M_2} \equiv \sum_{m_2 \in S(M_2)} \lambda_{m_1, m_2} = \lambda'_{M_1, M_2}$$

The construction of the compound CTMC can be done in analogy with the construction of the atomic CTMC, substituting the atomic firing of transitions by a symbolic firing working with compound markings, and merging equivalent compound markings instead of identical atomic markings. The requisites 1) and 2) are satisfied if the symbolism for the representation of compound markings and the symbolic firing are defined such that:

Property 1 *two compound markings M_1, M_2 are either disjoint $(S(M_1) \cap S(M_2) = \emptyset)$ or equivalent $(S(M_1) = S(M_2))$.*

Property 2 *all the atomic markings represented by a compound marking enable the same transitions.*

Property 3 *symbolic firings are isomorphic to atomic firings. More explicitely, the symbolic firing of a transition t in a compound marking M yields compound markings* $M_1, M_2, ..., M_n$ *(some could be equivalent) with rates* λ_{M, M_i} *if t is timed, or probabilities* p_{M, M_i} *if t is immediate, such that the firing of t in any* $m \in S(M)$ *yields atomic markings* $m_1, m_2, ..., m_n$ *with* $m_i \in S(M_i)$ *and with rates* $\lambda_{m, m_i} = \lambda_{M, M_i}$ *if t is timed, or probabilities* $p_{m, m_i} = p_{M, M_i}$ *if t is immediate.*

If properties 1, 2, and 3 are satisfied then the compound CTMC is a correct grouping of the atomic CTMC. A formal proof can easily be derived by induction on the elementary processes involved in the construction of the compound CTMC from the GSHLPN, however it is rather long and is not given here. In this section we define a symbolism for the description of compound markings of a GSHLPN and a symbolic firing satisfying properties 1, 2, and 3, and thus yielding compound CTMC's with a correct state grouping. The symbolism is based on a classification of the attributes of the token types of the net into groups. The classification also considers the place where token types can be held.

A *located token type* of a net is a pair *(s, p)*, where *s* is a token type and *p* is a place of the net which can hold tokens of type *s*. Let *PH(s)* be the set of places which, assumming that the GSHLPN is not overspecified, can hold tokens of type *s*. *PH(s)* can be found by examination of the arc labels: $p \in PH(s)$ if some input or inhibitor arc *(p, t)* has a token of type *s*, or some transition (case) output arc *(t, p)* *((c, p))* has a token of type *s*. The net of Figure 2 has the following located token types:

(PP, P1),
(PW, P2),
(PC, P3), (PC, P4), (PC, P5),
(M, P6),
(B, P7), (B, P8)

Although in the net of the example each place can only hold tokens of one type, it is clear that in general a place may hold tokens of several types. An attribute of a located token type will be called *located attribute* and the notation *att(s, p, i)* will be used to indicate the ith attribute of the located token type *(s, p)*. Before defining the classification of located attributes, it is necessary to select a particular syntax for predicates and expressions used in inhibitor and output token attributes. The syntax chosen is appropriate to capture permutation and cyclic symmetries in the values of the token attributes.

It is assumed that the predicate of a transition *t* is a logical expression with clauses of the types:

$$i = j \tag{3}$$
$$i \neq j \tag{4}$$
$$A_1 i_1 \oplus ... \oplus A_n i_n = K \tag{5}$$
$$A_1 i_1 \oplus ... \oplus A_n i_n \neq K \tag{6}$$

and that expressions for attributes of inhibitor and output tokens of a transition *t* have one of the forms:

$$i \tag{7}$$
$$A_1 i_1 \oplus ... \oplus A_n i_n \oplus K \tag{8}$$

where, i, j, and i_k belong to the free variable set of *t*, the attributes of the free variables in the same clause, and, for expressions, the attributes of the free variables and the attribute receiving the expression, have the same domain D, K and A_k are non-negative integers smaller to $|D|$, and the summation is taken modulus $|D|$. Note that $i = j \oplus K$, $i \neq j \oplus K$ are clauses of, respectively, the types (5), (6), and $i \oplus K$ is an expression of type (8).

Two located attributes are *connected* if the value of one of them in some inhibitor or output token is defined by an expression using a free variable of the other. For instance, *att(PW, P2, 1)* and *att(PP, P1, 1)* of the net of Figure 2 are connected. Consider the equivalence relation *link* defined as the reflexive and transitive (connect is symmetric by definition) closure of connect. We define the following relation from the equivalence relation link: two located attributes *a, b* are *related* if they have free variables appearing in a clause of some predicate or are linked to located attributes satisfying that condition. Note that the relation is symmetric and transitive, but not necessarily reflexive. A located attribute is *irrelevant* if it is not related to any other located attribute. Since the relation is symmetric and transitive, related attributes are classified into subsets, with all the attributes in a subset being related to each other. Such subsets are called *groups*. A group *G* is a *rotation group* if free variables for several located attributes in *G* appear together in a clause of types (5), (6) or an expression of type (8). A *permutation group* is a group which is not a rotation group.

Irrelevant attributes indicate overspecification of token types. They can be omitted in the corresponding located token types and we will assume they are. A net with irrelevant located attributes removed will be called *normalized*. For the net of Figure 2 it is found that all the located attributes corresponding to the first attribute of the token type *B* and the third attribute of the token type *PC* are irrelevant. In this case, the normalization of the net is accomplished by simply removing those attributes from those token types and the fact that these attributes can be removed indicates that the identity of the busses is irrelevant for the modeling of the multiprocessor system. In the general case, it may be necessary to define new token types, since token types

have by definition a fixed numbers of token attributes and the same attribute could be irrelevant in some places and relevant in others. The relevant attributes of the net of Figure 2 are classified into one rotation group R_1 and one permutation group P_1:

$$R_1 = \{att(PP, P1, 1), att(PW, P2, 1), att(PC, P3, 1), att(PC, P4, 1), att(PC, P5, 1), att(B, P7, 2), att(B, P8, 2)\}$$

$$P_1 = \{att(PW, P2, 2), att(PC, P3, 2), att(PC, P4,2), att(PC, P5, 2), att(M, P6, 1)\}$$

The located attributes of R_1 are processor identifiers and the located attributes of P_1 are memory identifiers. Even if attributes in the same group have different semantics, the domain requirements imposed for clauses and expressions guarantee that all the attributes in a group will have the same domain.

Let R_i and P_i be, respectively, the rotation and permutation groups of the net and denote by RD_i and PD_i the domains of the located attributes in the respective groups. We associate to each rotation group R_i a free variable r_i taking values in RD_i , and to each permutation group P_i , a set of distinguished free variables p_{ij} , $j = 1, ..., |PD_i|$, taking *different* values in PD_i. A *compound marking* M is an assignment of bags of symbolic tokens to places of the net where the located attributes of each group R_i are specified by expressions $r_i \oplus K$ (K constant, $0 \le K \le |RD_i|-1$), and the located attributes of each group P_i by distinguished free variables p_{ij}, $j = 1, ..., U_i(M)$, $U_i(M) \le |PD_i|$. The set $S(M)$ of atomic markings represented by M is obtained by assigning values to the free variables from their respective domains (different values for the distinguished free variables associated to the same permutation group).

Assume that the multiprocessor system taken as example has 4 processors, 2 memories and 2 busses. Then $|RD_1|=4$ and $|PD_1|=2$. Let r be the free variable associated to R_1 and p_1, p_2, p_3, p_4 the distinguished free variables associated to P_1. A reachable compound marking of the normalized net is:

$$\begin{aligned} &P1: \langle PP, r\rangle, \langle PP, r \oplus 1\rangle \\ &P2: \langle PW, r \oplus 2, p_1\rangle \\ &P3: \langle PC, r \oplus 3, p_1\rangle \\ &P6: \langle M, p_2\rangle \\ &P7: \langle B, r\rangle , \end{aligned} \qquad (9)$$

which describes the state of the system in which two consecutive (according to the cyclic ordering defined by the token passing protocol) processors are executing in the private domain, the next processor is waiting for a memory currently used, the remaining processor is executing in the common domain in the mode with mean duration $1/\lambda_{C,1}$, accesing the memory for which the latter is waiting, and the token of the free bus is being passed to the second processor executing in the private domain. We show next that the compound markings thus defined satisfy Property 1.

Proposition 1 *Two compound markings M_1, M_2 of a GSHLPN are either disjoint or equivalent.*

Proof

From the definition of compound markings it is clear that the elements of $S(M)$ can be generated from any atomic marking $m \in S(M)$ by combinations of rotations $\rho_i(x) = x \oplus K$ ($0 \le K \le |RD_i| - 1$) of the values of located attributes in each rotation group R_i and substitutions of the $U_i(M)$ different values of the located attributes of each permutation group P_i by different values of PD_i. Assume $m \in S(M_1) \cap S(M_2)$. This imposes $U_i(M_1) = U_i(M_2)$ for all the permutation groups P_i. Then, all the elements in $S(M_1)$ and $S(M_2)$ can be obtained by combinations of the same mappings of the attribute values of m and $S(M_1) = S(M_2)$ ∴

The symbolic firing of a GSHLPN is defined in complete isomorphism with its atomic firing but using the free variables supporting the description of compound markings, instead of values of the attribute domains, for the determination of the firing sets and firing distributions, and for the evaluation of the predicates and specified attributes of output and inhibitor tokens. For unspecified attributes, each of the expressions $r_i, r_i \oplus 1, ..., r_i \oplus (|RD_i|-1)$ is taken with the same probability if the located attribute belongs to the rotation group R_i, and each of the variables $p_{ij}, j = 1, ..., |PD_i|$ is taken with the same probability if the located attribute belongs to the permutation group P_i *.

For instance, the transition $T1$ of the GSHLPN of Figure 2 is enabled in the compound marking (9) and has one firing distribution comprising the firing sets $\{\langle PP, r\rangle\}$ and $\{\langle PP, r \oplus 1\rangle\}$. The transition fires in each firing set with rate λ_P. Since the second attribute of the output token of the transition is unspecified, there are two modes for each firing set, each associated to a memory module of the system. Each mode has a probability 1/2. Thus, the firing of $T1$ yields 4 compound markings, each with rate $\lambda_P/2$. The compound markings are:

$$\begin{aligned} M_1 \quad &P1: \langle PP, r\rangle \\ &P2: \langle PW, r \oplus 1, p_1\rangle, \langle PW, r \oplus 2, p_1\rangle \\ &P3: \langle PC, r \oplus 3, p_1\rangle \\ &P6: \langle M, p_2\rangle \\ &P7: \langle B, r\rangle \end{aligned}$$

* Although this is not the most efficient approach, it shows clearly the isomorphism (Property 3). A more efficient and equivalent method is to take any of the U_i free variables of the permutation group P_i remaining in the compound marking with probability $1/|PD_i|$, and, if $U_i < |PD_i|$, a different free variable of the group with probability $(|PD_i| - U_i)/|PD_i|$.

M_2 P1: $\langle PP, r \rangle$
P2: $\langle PW, r \oplus 1, p_2 \rangle, \langle PW, r \oplus 2, p_1 \rangle$
P3: $\langle PC, r \oplus 3, p_1 \rangle$
P6: $\langle M, p_2 \rangle$
P7: $\langle B, r \rangle$

M_3 P1: $\langle PP, r \oplus 1 \rangle$
P2: $\langle PW, r, p_1 \rangle, \langle PW, r \oplus 2, p_1 \rangle$
P3: $\langle PC, r \oplus 3, p_1 \rangle$
P6: $\langle M, p_2 \rangle$
P7: $\langle B, r \rangle$

M_4 P1: $\langle PP, r \oplus 1 \rangle$
P2: $\langle PW, r, p_2 \rangle, \langle PW, r \oplus 2, p_1 \rangle$
P3: $\langle PC, r \oplus 3, p_1 \rangle$
P6: $\langle M, p_2 \rangle$
P7: $\langle B, r \rangle$

Given the definition of the symbolic firing of a transition, it should be clear that properties 2 and 3 are satisfied if the symbolism for compound markings is coherent with the predicates and specified attributes of output and inhibitor tokens. This is stated by the following proposition:

Proposition 2 *The symbolism for compound markings satisfies the following properties:*

a) located attributes with free variables in the same clause belong to the same group G, and if the clause is of type (5) or (6) G is a rotation group.

b) the located attributes with free variable in an expression (7) or (8) and the located attribute whose value is defined by the expression belong to the same group G, and if the expression is of type (8) G is a rotation group.

Proof

To prove a), consider first clauses of types (3), (4), let a, b be the located attributes of, respectively, i, j, and assume $a \neq b$ ($a=b$ is a trivial case). Then, by definition, a and b are related and belong to the same group. Parallel arguments show that the located attributes with free variable in a clause of type (5) or (6) are related and belong to the same group G. In addition, since G has located attributes with free variables appearing together in a clause of type (5) or (6), G is a rotation group.

To prove b), consider an expression of type (7), let a be the located attribute of i, b the located attribute whose value is defined by the expression, and assume $a \neq b$. By definition, a and b are connected and, since link is the closure of the connect relation, linked. Since a is not irrelevant (irrelevant attributes are suppressed), a is related to some other located attribute c. But, being a and b linked, b is also related to c. Therefore, a and b belong to the same group. It can be shown similarly that the located attributes with free variable in an expression of type (8) and the located attribute whose values is defined by the expression belong to the same group G. In addition, since G has located attributes with free variables appearing together in an expression of type (8), G is a rotation group $\therefore$

Therefore, we can state the result looked for in this section:

Theorem 1 *The compound CTMC obtained by firing symbolically a GSHLPN with the compound markings as defined in this section is a correct state grouping of the atomic CTMC of the GSHLPN.*

To end this section we justify why located attributes and not just attributes of token types have been used to define the symbolism for compound markings. The reason is that disregarding the place would produce in some cases less compact compound markings. Consider, for instance, the GSHLPN shown in Figure 3, where all the attributes have the same domain D, with $|D|=2$. Disregarding the place would force compound markings with all the attributes in a unique rotation group. Considering, as proposed, located attributes two rotation groups are obtained:

$$R_1 = \{att(B, P2, 1), att(C, P3, 1)\}$$
$$R_2 = \{att(B, P4, 1), att(D, P5, 1)\}$$

This makes that, in general, each of the last compound marking contain several of the former. For instance, the compound marking:

P2: $\langle B, r_1 \rangle$
P3: $\langle C, r_1 \oplus 1 \rangle$
P4: $\langle B, r_2 \rangle$
P5: $\langle B, r_2 \oplus 1 \rangle$

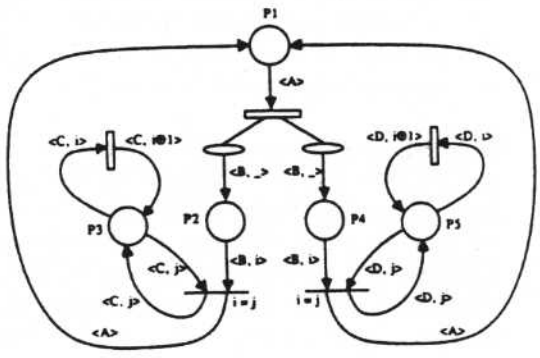


Figure 3. A GSHLPN to illustrate the advantages of using located attributes.

represents the same set of atomic markings that the following compound markings together:

P2: <B, r>	P2: <B, r>
P3: <C, r ⊕1>	P3: <C, r ⊕1>
P4: <B, r >	P4: <B, r ⊕1>
P5: <D, r ⊕1>	P5: <D, r >

4. COMPOUND MARKING EQUIVALENCE

The construction of the compound CTMC requires the determination of the equivalence of compound markings. Two compound markings M_1, M_2 are equivalent if the subsets $S(M_1), S(M_2)$ coincide. By renaming free variables of permutation groups we obviously get equivalent compound markings. Therefore, without lost of generality, it will be assummed that the set of distinguished free variables of each permutation group used in any compound marking M has the form $\{p_{ij}, 1 \le j \le U_i(M)\}$. If $U_i(M_1) \neq U_i(M_2)$ for some P_i, the compound markings are clearly not equivalent. Assuming $U_i(M_1) = U_i(M_2)$ for all P_i, the markings are equivalent if and only if there exists a rotation $\rho_i(x) = x \oplus K_i$ for the located attributes of each rotation group and a permutation of each set of distinguished variables p_{ij} associated to a permutation group P_i mapping M_1 into M_2. Since the construction of large compound CTMC's will involve a large number of equivalence tests, an efficient algorithm should be found to carry them out. However, although heuristics which will work well in most cases are easily found, our attempts to derive an algorithm with a polynomial complexity were unsuccessful. Therefore, we decided to study the complexity of the problem from the point of view of the NP-completeness theory (see, for instance [15]). We present in this sections the results of the analysis.

Let RG be the number of rotation groups and PG the number of permutation groups of the GSHLPN. We first show that for bounded RG, the problem is polynomially equivalent to GRAPH ISOMORPHISM, a well-known problem whose classification in the NP world is still unknown. We then show that if token types have at most one permutation attribute the problem is in P (a polynomial deterministic algorithm exists).

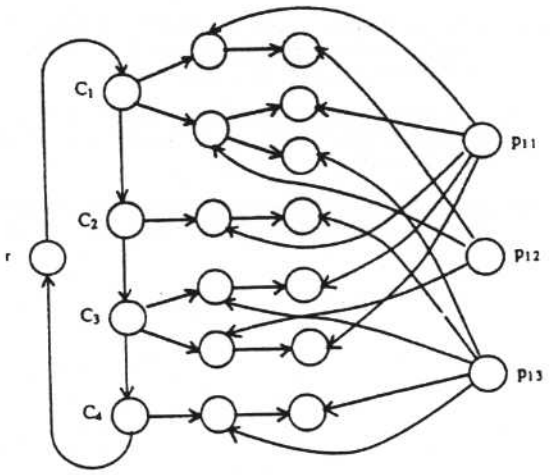
The problem is obviously in NP. In addition it is known [15] that DIRECTED GRAPH ISOMORPHISM is polynomially equivalent to GRAPH ISOMORPHISM. Therefore, it is enough to find Karnap reductions, in both directions, between COMPOUND MARKING EQUIVALENCE and DIRECTED GRAPH ISOMORPHISM. Before describing the reductions we give some additional definitions. A *classification* of a compound marking M is a partitioning of the tokens in M in classes holding tokens in the same place, of the same token type, with the same multiplicity, and with the same expressions for each located attribute belonging to a rotation group. A *grouping* of a compound marking M is a partitioning of the classes of M into groups holding classes associated to the same place, token type and multiplicity.

Reduction from COMPOUND MARKING EQUIVALENCE to DIRECTED GRAPH ISOMORPHISM

Compute the classes and groups of M_1 and M_2. If it is not possible to define a one-to-one mapping between corresponding groups (groups with the same place, token type and multiplicity), or the numbers of classes in some pair of corresponding groups are different, the markings are not equivalent. Note that the mapping of groups is unique if exists. Take classes $C_i^1, 1 \le i \le k$ of M_1 from different groups $G_i^1, 1 \le i \le k$ such that all the variables r_i instantiated in M_1 appear in some C_i^1 (note that this can always be done with $k \le RG$). Consider the groups G_i^2 mapped to G_i^1, and all the selections of classes $C_i^2, 1 \le i \le k$ of M_2 with $C_i^2 \in G_i^2$ and such that $|C_i^2| = |C_i^1|$. If there not exists any selection with $|C_i^2| = |C_i^1|, 1 \le i \le k$, the markings are not equivalent. For a given selection, each pair (C_i^1, C_i^2) requires a rotation ρ_j for each located attribute of R_j in the token type of the classes. For instance, if the tokens in C_i^1 are of the form $<s, ..., r_1, r_2 \oplus 1>$ and those in C_i^2 are of the form $<s, ..., r_1 \oplus 1, r_2 \oplus 3>$, the rotations $\rho_1(x) = x \oplus 1$ and $\rho_2(x) = x \oplus 2$ are required. The required rotations may be incompatible. If the required rotations are incompatible for all the selections, the markings are not equivalent.

Consider the selections requiring compatible rotations. By construction, there is a rotation for each rotation group with located attributes instantiated in the compound markings, mapping the rotation attributes of the tokens in C_i^1 into those of C_i^2, but this may not be the case for the remaining classes. The sets of rotations mapping rotation attributes for all the classes constitute an *arrangement of compatible rotations*. If no arrangement of compatible rotations exists for any selection $C_i^2, 1 \le i \le k$, the markings are not equivalent. Since tokens in classes have unique expressions for their rotation attributes, an arrangement defines a one-to-one mapping of the classes of the compound markings. The arrangement maps M_1 into M_2 except for the variables p_{ij}. It is enough to check whether a permutation exists for each set of variables $p_{ij}, 1 \le j \le U_i$ mapping each class of M_1 into the corresponding class of M_2. We assume any ordering of the classes of M_1 and consider the ordering of M_2 induced by the one-to-one mapping between classes. Then, for each compound marking, we construct a directed graph "writing" the compound marking except for permutations of the variables p_{ij} as follows.

Let M be the compound marking, CS the set of classes, with ordering $C_1, ..., C_m$, and VS the set of variables p_{ij} instantiated in M. The directed graph D associated to M contains a node r, and the subsets CS and VS. D also contains the arcs required to build the circuit $(r, C_1, ..., C_m)$, and rooted at each node C_i a tree constructed to represent the set of tokens in C_i, with each node of the tree receiving an arc from some node in VS. The construction is illustrated in Figure 4. It should be clear that there



Class	Tokens	Mulupliciy
C_1	<...., p_{11}, p_{12}>	2
	<...., p_{12}, p_{11}>	2
	<...., p_{12}, p_{13}>	2
C_2	<...., p_{11}, p_{13}>	1
C_3	<...., p_{13}, p_{11}>	1
	<...., p_{12}, p_{11}>	1
C_4	<...., p_{13}, p_{13}>	1

Figure 4. Example illustrating the construction of directed graphs in the reduction of COMPOUND MARKING EQUIVALENCE to DIRECTED GRAPH ISOMORPHISM.

is a one-to-one mapping between the set of leaves of the tree rooted at C_i and the set of tokens C_i. If the directed graphs D_1 and D_2 obtained for some arrangement of compatible rotations are isomorphic, the markings are equivalent. Otherwise, they are not ∴

It can easily be seen that the reduction is correct. Note that $(r, C_1, ..., C_m)$ is the only cycle of the directed graphs, r the only node in the cycle with output degree 1, and p_{ij} the only nodes without input arcs. Therefore, r, C_1, ..., C_m, the trees rooted at nodes C_i, and the set of nodes $\{p_{ij}\}$ are identified in the directed graphs and any permutation of nodes mapping the graphs has to be restricted to $\{p_{ij}\}$. Let N be the number of tokens in the compound markings. Since at most RG classes C_i^1 will have to be taken and the number of tokens in a class is bounded by N, the number of sets of rotations is bounded by N^{RG} and so is the number of arrangements of compatible rotations. Therefore, the number of times that DIRECTED GRAPH ISOMORPHISM is invoked is bounded by N^{RG}. By construction, the size of the graph is linearly related with the size of the compound markings. It is a trivial task to check that all the other operations involved are polynomial in the size of the compound markings. Therefore, the reduction is polynomial for bounded RG.

Reduction from DIRECTED GRAPH ISOMORPHISM to COMPOUND MARKING EQUIVALENCE

This reduction is trivial. For each directed graph D a compound marking M is built having tokens of the same type, in the same place and with multiplicity 1. The marking uses distinguished free variables of one permutation group. Tokens have two attributes and a distinguished variable i is used for each node of the directed graph. There is a token $<A, i, j>$ for each arc (i,j) of D. The compound markings thus obtained are checked for equivalence. The directed graphs are isomorphic if the compound markings are equivalent ∴

The last reduction shows that is enough to allow token types with more than one permutation attribute to have a polynomial equivalence between the problems. In the case in which token types have only one permutation attribute, the invocations of the DIRECTED GRAPH ISOMORPHISM PROBLEM done in the first reduction are trivial and can be solved in polynomial time. Therefore, in this case, COMPOUND MARKING EQUIVALENCE is polynomial. This allows to state the following theorems:

Theorem 2 *COMPOUND MARKING EQUIVALENCE with bounded number of rotation groups is polynomially equivalent to GRAPH ISOMORPHISM if the compound markings have token types with more than one permutation attribute.*

Theorem 3 *COMPOUND MARKING EQUIVALENCE with bounded number of rotation groups is polynomial if compound markings have token types with at most one permutation attribute.*

5. CONCLUSIONS

We have presented generalized stochastic high-level Petri nets, a new type of nets obtained by the combination of the qualities of stochastic high-level Petri nets and generalized stochastic Petri nets and have discussed their semantics, paying special attention to the firing policies, which become more complicated than for standard stochastic Petri nets due to the existence of token types with attributes. After illustrating the expressive power of the nets with an example we have considered the automated construction of compound CTMC's from GSHLPN's.

We have given properties of symbolisms for compound markings which guarantee that the compound CTMC obtained by symbolically firing the net will have a correct state grouping. For a particular, yet highly expressive, syntax, a symbolism for compound markings with those properties has been derived. The procedure is amenable of automation. The construction of the compound CTMC requires an algorithm to test the equivalence of compound markings. We have shown that, in the general case

and for bounded number of rotation groups, the EQUIVALENCE COMPOUND MARKING problem is polynomially equivalent to the well-known GRAPH ISOMORPHISM problem. The classification of the last problem is currently open, but there is strong evidence that it cannot be NP-complete (see, [16] for details). Although the existence of a polynomial algorithm to test the equivalence of compound markings would constitute an important asset for GSHLPN's (and SHLPN's as well), its inexistence does not rule out their application, since it seems apparent that heuristics which will be efficient in most cases can be developed. We are currently working in that direction.

REFERENCES

[1] M. K. Molloy, *Performance Analysis using Stochastic Petri Nets*, IEEE Trans. on Computers, vol. C-31, no. 9, pp. 913-917, September 1982.

[2] S. Natkin, *Les reseaux de Petri Stochastiques et leur application a l'evaluation des systèmes informatiques*, Thèse de Docteur Ingénieur, CNAM, Paris, 1980.

[3] J. B. Dugan, G. Ciardo, A. Bobbio, and K. S. Trivedi, *The design of a unified package for the solution of stochastic Petri net models*, in Proc. Int. Workshop on Timed Petri Nets, Torino, Italy, pp. 6-13, July 1985.

[4] J. F. Meyer, A. Movaghar, and W. H. Sanders, *Stochastic activity networks: structure, behaviour and applications*, in Proc. Int. Workshop on Timed Petri Nets, Torino, Italy, pp. 80-87, July 1985.

[5] M. A. Marsan, G. Conte, and G. Balbo, *A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems*, ACM Trans. on Computer Systems, vol. 2, no. 1, pp. 93-122, May 1984.

[6] M. A. Marsan, G. Balbo, G. Chiola, and G. Conte, *Generalized Stochastic Petri nets revisited: Random switches and priorities*, in Proc. Int. Workshop on Petri Nets and Performance Models, 1987, pp. 44-53.

[7] G. Balbo, G. Chiola, G. Franceschinis, and G. Molinar Roet, *On the efficient construction of the tangible reachability graph of generalized stochastic Petri nets*, in Proc. Int. Workshop on Petri Nets and Performance Models, 1987, pp. 136-145.

[8] Ch. Lin and D. C. Marinescu, *On Stochastic High-Level Petri Nets*, in Proc. Int. Workshop on Petri Nets and Performance Models, 1987, pp. 34-43.

[9] Ch. Lin and D. C. Marinescu, *Stochastic High-Level Petri Nets and Applications*, IEEE Trans. on Computers, vol. 37, no. 7, pp. 815-825, July 1988.

[10] K. Jensen, *High-level Petri Nets*, Informatik-Fachberichte, vol. 66, pp. 166-180, 1983.

[11] A. Zenie, *Coloured stochastic Petri nets*, in Proc. Int. Workshop on Timed Petri Nets, Torino, Italy, pp. 262-271, July 1985.

[12] D. C. Marinescu, *Domain oriented analysis of PDE splitting algorithms*, Journal of Information Science, vol. 43, pp. 3-24, 1987.

[13] P. Huber, A. M. Jensen, L. O. Jepsen, and K. Jensen, *Towards Reachability trees for high-level Petri nets*, Lecture Notes in Computer Science, vol. 188, pp. 215-133, Springer-Verlag, 1985.

[14] I. A. Papazoglou, E. P. Gyftopoulos, *Markov Processes for Reliability Analysis of Large Systems*, IEEE Trans. on Reliability, vol. R-26, no. 3, pp. 232-237, August 1977.

[15] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., New York, 1983.

[16] D. S. Johnson, *The NP-Completeness Column: An Outgoing Guide*, Journal of Algorithms, vol. 9, no. 3, Sept. 1988, pp. 426-444.