A Structural Colour Simplification in Well-Formed Coloured Nets

Giovanni Chiola *and Giuliana Franceschinis †
Dip. di Informatica, Università degli Studi di Torino
corso Svizzera 185, 10149 Torino, Italy
email: chiola@di.unito.it

Abstract

Well-Formed coloured Petri nets (WN) have been recently defined as a new formalism that allows the automatic identification of model symmetries by means of the construction of the symbolic reachability graph (SRG). Here we define a sufficient structural condition for the simplification of the colour structure of the net that can exploit symmetries not detectable with the SRG. The simplified WNs may have reduced SRG and in any case are easier to analyze by means of the SRG technique.

1 Introduction

Coloured Petri nets (CPNs) [1] have been proposed in the literature as a good modelling tool for the study of complex distributed systems. Initially, coloured nets were proposed as a kind of shortened notation to describe large and symmetric Petri net models. The common behaviour of several entities constituting a large system is described by the Petri net structure, while different entities are identified by different "coloured" tokens that can flow through the net.

Subsequently, the idea of *parametrization* has been suggested by Genrich [2]. A single net model can represent many different systems that differ only in the total number of components that behave similarly (i.e., that differ only in the cardinality of the colour sets). The whole family of such similar systems is modelled and studied using a single coloured Petri net, provided that the analysis results are parametric with respect to the colour sets. In [3] an example of parametric proof of a distributed algorithm is provided using coloured Petri net techniques.

One of the most powerful analysis techniques for bounded (coloured) Petri net systems is the analysis of the *reachability graph* (RG), i.e., the complete enumeration of the state space. In this case parametric analysis independent of the cardinality of the colour sets is of course impossible: a CPN model must be instantiated with particular choices of colour sets and initial markings in order to compute the RG. However, one can still take advantage of the coloured net formalism for the analysis: the symmetry properties of coloured models can be exploited in order to reduce the total number of states that should be analyzed to check for the properties of the model [4].

Well-formed coloured Petri nets (WNs) have been introduced in [5] together with a totally algorithmic procedure for the construction of a "symbolic reachability graph" (SRG) that exploits symmetries to reduce the size of the state space representation. The same reduction in state space can also be exploited for performance evaluation based on Markovian analysis. In [6] Stochastic WNs (SWNs) have been introduced to allow this kind of improved analysis.

In this paper the notion of *redundant colour component* is defined; it is based on the observation that some symmetries in the RG of WNs are due to the presence of redundancy in the colour specification i.e., some colour components exist that do not need so many distinguishable elements for a correct specification of the model in terms of possible transition firing sequences. Some sufficient structural conditions for the reduction of the cardinality of a colour component at the net level (i.e., independently of the initial marking) are given. Notice that when the number of distinct elements of a basic colour set becomes one, the corresponding colour component can be cancelled from the net. The RG aggregation obtained through the above decolourization procedure in general is not detected (hence not exploited) by the SRG generation algorithm (and vice-versa). These two RG aggregation methods are then complementary and our conjecture is that combining them it is possible to exploit all the behavioural symmetries of the model implicit in its structure.

The concept of colour redundancy and colour simplification has already been introduced in [7]. In this paper we extend the results in [7]. In fact now the partial decolourization of color classes is introduced besides the total decolourization. Moreover less restrictive sufficient conditions for colour simplification are defined.

*Part of this work has been done while G. Chiola was visiting researcher at the Laboratoire MASI of the Université Pierre et Marie Curie, Paris (6), with the financial support of a NATO-CNR annual research grant.

†This work has been partially supported by the CNR project "Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo"

TH0386-3/91/0000/0144/$01.00 © 1991 IEEE

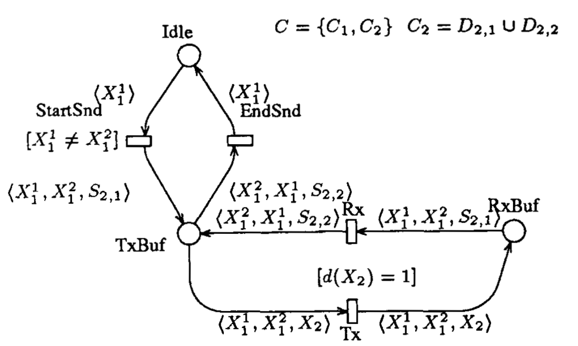


Figure 1: A message exchange protocol

The presence of redundancies in the model description is quite natural in CPNs, and derives from the methodology followed in the model construction. Once the basic elements that flow in the model structure are defined, two extreme choices may be considered "natural" in the definition of the basic colours that identify such elements: either each basic element has its own identity or no distinction is made between basic elements (like in the case of P/T nets). This is because a-priori it would be too difficult to conceive a representation that is both minimal and correct. The best approach would probably be that of considering all basic elements undistinguishable during the model construction process until the necessity arises to actually make some distinction (for example to identify higher and lower priority processes). Anyway the results presented in this paper allow an easy construction of a WN model without consideration for the problem of minimality of the representation. Unnecessary distinctions among basic colours can be identified and eliminated automatically, thus producing an equivalent simplified model that is easier to analyze.

2 WN Definition

Well-formed Nets are a formalism with the same modelling power of CPNs from the point of view of representation conciseness. The main difference between WNs and CPNs is that in WNs the functions are defined in a structured way, by composition of few basic functions. This structured definition allows one to directly infer some algebraic properties of functions or some behavioural symmetries in the net.

In the following subsections we first introduce the basic ingredients for building WNs and then we give their formal description. The concepts will be illustrated on the example net of Figure 1 that models a skeleton of a packet switching communication protocol. The modeled system consists of a set of sites that can exchange information by sending messages. The message exchange protocol between any two sites is very simple: each site eventually transmits some data messages split into packets and waits for the corresponding acknowledgement. A site that has transmitted some data packets but has not yet received an acknowledgement cannot start another data message transmission to any site.

We then discuss the problems that predicates on arcs could introduce in the task of discovering and eliminating colour redundancies at a purely structural level. Finally we discuss the need for some WN simplification rule to eliminate fictitious synchronizations in the net structure.

2.1 WN basic ingredients

In this section we introduce very briefly, due to space constraints, the basic ingredients needed to define a WN. See [5] for a more extended description.

Object classes A colour class, denoted C_i, is a set of objects of the same type. In the net of Figure 1 there are two object classes: the sites (C_1) and the packets (C_2). Classes can be ordered, in this case a "successor" function $\oplus$ is associated to them. When objects belonging to the same class have different behaviours it is convenient to partition the class in *static subclasses*, denoted $D_{i,q}$, of homogeneously behaving objects. In our example here are two kinds of packets: data transfer or acknowledge, therefore C_2 is partitioned into two static subclasses $D_{1,1}$ and $D_{1,2}$.

Colour domains Place and transition colour domains are made of Cartesian products of object classes. A given class C_i can occur several times, say e_i times, in the product so that if $I = \{1, \ldots, n\}$ is the set of possible color class indexes, a colour domain can be characterized by a vector $J \in Bag(I)$ of n integers $\{e_1, \ldots, e_n\}$, representing the occurence of each class in the domain definition.

The colour domain of a place or transition r is denoted $C(r)$ or $C_{J(r)}$. In our example $J(Idle) = \{1, 0\}$, $J(Tbuf) = \{2, 1\}$ then $C(Idle) = C_1$ and $C(Tbuf) = C_1 \times C_1 \times C_2$ (or $C(Tbuf) = (C_1)^2 \times C_2$).

Standard predicates The predicates used in the function definition or associated to a transition are [1]: $X_i^j = X_i^k$, $X_i^j = \oplus X_i^k$, $d(X_i^j) = q$, $d(X_i^j) = d(X_i^k)$, and combinations of these predicates using $\wedge$, $\vee$, and $\neg$. The first two predicates are used to compare objects of the same class while the third and fourth check the objects membership to a given static subclass.

Colour functions WN standard function are built up using only three basic kind of functions: the *Identity functions* denoted by X_i^j, if several X_i^j appear next to the same transition with the same superscript j, then they must return the same object; the *Successor functions* $\oplus X_i^j$, applicable only to ordered classes; the *Diffusion functions* $S_{i,q}$ thet have either the effect of synchronizing all the objects in $D_{i,q}$ if used on input or inhibition arcs, or of diffusing these objects if used on output arcs.

[1] Function d gives for each colour element $c_i \in C_i$, the index q of the static colour domain $D_{i,q}$ to which it belongs.

A linear combination homogeneously indexed basic functions (i.e., functions that return bags of objects belonging to the same class C_i) is still a basic function; the general form of such basic functions is:

$$\sum_{q=1}^{n_i} \alpha_{i,q}.S_{i,q} + \sum_{k=1}^{e_i} (\beta_{i,k}.X_i^k + \gamma_{i,k}. \oplus X_i^k) \quad (1)$$

The general form of a function labelling a generic arc between place p and transition t is a weighted sum of possibly guarded terms. Each term is a tuple of basic functions (1). The arity of the function tuples is equal to arity of the Cartesian product defining $C(p)$. if we denote $f_{i,j}$ the element in a function tuple corresponding to the j_{th} occurrence of class C_i in $C(p)$, the formal definition of a standard colour functions in WNs is:

$$\sum \delta.[pred] \bigotimes_{i,j} f_{i,j} \quad (2)$$

where δ is a positive integer, and $[pred]$ stands for a standard predicate.
The value of the colour function "$[pred]f$" is given by: $[pred]f(c) \;=\;$ If $pred(c)$ then $f(c)$ else 0.

In our example net all the functions have only one term. The tuple elements of the function on the arc connecting *Startsnd* to *Tbuf* are: $f_{1,1} = X_1^1$, $f_{1,2} = X_1^2$, $f_{2,1} = S_{2,1}$.

2.2 WN formal definition

Definition 1 (WN) *A well formed net* $\mathcal{N} = \langle P, T, C, J, W^-, W^+, W^H, \Phi, \pi \rangle$ *is made of:*

P, T *the finite sets of places and transitions;*

C *the family of object classes;* C_i *is partitioned in static subclasses* $C_i = \bigcup_{q=1}^{n_i} D_{i,q}$*;*

$J : P \cup T \rightarrow Bag(I)$, $C(r) = C_{J(r)}$ *denotes the domain of node r;*

W^-, W^+, W^H : $W^-(p,t), W^+(p,t), W^H(p,t) \in [C_{J(t)} \rightarrow Bag(C_{J(p)})]$ *the input, output, and inhibition functions are standard functions of the form shown in (2); the set of input, inhibition and output places of a given transition t will be denoted* $^\bullet t$, $^\circ t$ *and* $t^\bullet$ *respectively;*

$\Phi(t) : C_{J(t)} \rightarrow \{True, False\}$ *is a standard predicate associated with the transition t. By default we will assume* $\forall t \in T$ *the standard predicate* $\Phi(t) = True$*;*

$\pi : T \rightarrow I\!N$ *the priority function. By default we assume* $\forall t \in T$ *the value* $\pi(t) = 0$*.*

A WN can represent a whole family of models that differ only for the actual cardinality of the basic colour sets and for the initial marking. Each individual model obtained by instantiating a WN with a given set of parameters is called a WN *system*.

Definition 2 (WN system) *A well formed net system* $\Sigma = (\mathcal{N}, CS, M_0)$ *is made of:*

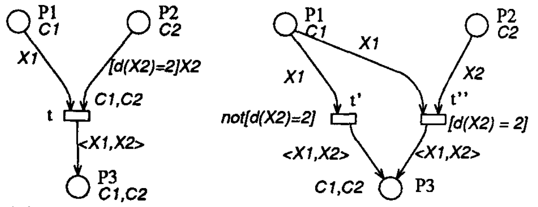


Figure 2: An example of function predicates elimination.

$\mathcal{N}$ *a well-formed net;*

CS *a colour set instantiation function that assigns to each static subclass* $D_{i,q}$ *a particular set of colour elements* $CS(D_{i,q})$*;*

M_0 $M_0(p) \in Bag(C(p))$ *is the initial marking of p.*

2.3 Function versus transition predicates

It is important to observe that predicates can have different effects depending on whether they label transitions or arc functions. Predicates on transitions are used to add some further condition on the transition enabling. Predicates in the function definitions are instead used to obtain variant functions, i.e., functions that depend on the particular colour instantiation of the corresponding transition. It may happen that an arc completely disappears for a particular instantiation of the corresponding transition (when all the predicates in the sum in (2) evaluate to "false"). It is clear that in general this situation could prevent the application of any structure-based algorithm.

To overcome this problem it is possible to perform a transformation on the net structure that makes explicit all possible forms that a variant function can assume. This transformation requires that each transition that has guarded functions on its arcs be replicated as many times as the possible different definitions of variant function on the arcs. The example depicted in Figure 2 shows this kind of transformation. There is only one function predicate: $[d(X2) = 2]$. When the predicate is false, the arc disappears. When this predicate is true, the function on the arc connecting P_2 to t becomes simply $X2$. Hence two copies of t are needed that account for the two situations: t' with an associated predicate $[\neg\ d(X_2) = 2]$ and t'' with an associated predicate $[d(X2) = 2]$.

In [8] a completely structural general transformation procedure is described that eliminates all the function predicates from a WN.

2.4 WN simplification

The structural colour redundancy detection and elimination algorithm that is described in Section 3.2 is based on the detection of possible colour synchronization patterns on the net structure. It may happen that WN models contain redundant structures that do not add information to the model itself, that have no influence on the state space size, and that introduce fictitious synchronization patterns in the model.

In [7] and [8] a simplification procedure is defined that allows one to remove some of these redundancies. In the sequel of the paper we assume that the WN models treated are simplified before any further consideration.

3 Colour redundancy elimination

In this section we consider only nets that do not have arc function predicates.

3.1 Decolourization of the RG

The formal definition of RG decolourization is given over a set of colour components selected through a *projection support* for a given *colour shrinking function.* The concept of redundant colour component is also introduced.
Let $C_J = \bigotimes C_i^{e_i}$ be the Cartesian product of basic colour domains that define the colour of some place or transition.

Definition 3 (Projection support) *is a function from colours to components*

$$ps : C \rightarrow 2^{(I\!N \times I\!N)}$$

where $C = \{C_J : J \in Bag(I)\}$ *is the set of all possible colours that can be built on the set C of basic colour domains, the components are indicated by pairs* ⟨colour domain index, occurrence index⟩ *and* $ps(C_J) \subseteq \{\langle i,j\rangle : (1 \leq j \leq e_i)\}$.

Definition 4 (Colour shrinking function) *is a function from colour elements in the basic colour classes C_i to colour elements in the shrinked colour classes C'_i (C'_i is the image of C_i through s) such that $|C'_i| \leq |C_i|$ and C'_i has a number of static subclasses less than or equal to the one of C_i and*

$$\forall c_1, c_2 \in C_i, \quad d(c_1) = d(c_2) \quad \Rightarrow \quad d(s(c_1)) = d(s(c_2))$$

The function s establishes a correspondence between a static subclass $D_{i,q}$ in C_i and a static subclass $D'_{i,q}$ in the shrinked colour class C'_i. The shrinked static subclasses can be either of cardinality one or of the same cardinality of the originating static subclass $D_{i,q}$.

Observe that the definition of s does not require that this relation among static subclasses be injective.

Notice that the above definition allows one to have a colour class in which only some static subclasses are decolourized while others remain unchanged. Moreover it may happen that two or more static subclasses in the original model collapse into a single static subclass after the decolourization. In this case the new static subclass has cardinality one.

A further comment is needed for ordered classes. In fact it makes no sense to decolourize only some static subclasses of an ordered colour class because after the decolourization of a static subclass $D_{i,q}$ (of cardinality greater than one) we would be unable to compute correctly the function $\oplus X_i$ on the elements of the decolourized subclass. Thus, an ordered subclass can be either completely decolourized or not even partially decolourized.

Definition 5 *The decolourization through a colour shrinking function s of the components of a colour tuple $c \in C_J$ identified by a projection support ps is the function*

$$\mathcal{D}_{ps,s}(\bigotimes_{i,k} c_{i,k}) = \bigotimes_{i,k} c'_{i,k}$$

where $c'_{i,k} =$ if $\langle i,k\rangle \in ps$ then $s(c_{i,k})$ else $c_{i,k}$ *i.e., the decoulorization function $\mathcal{D}_{ps,s}$ applied to a tuple $c \in C_J$ gives another tuple of the same arity where each component $c_{i,k}$ selected in ps is substituted by $s(c_{i,k})$.*

Notice that when a colour class C_i that contains a single static subclass is shrinked so that $|C'_i| = 1$, the corresponding components in the tuple could be cancelled by the decolourization (in this case the arity of the tuple would decrease).

The decolourization of a marking needs a projection support for each place and a unique colour shrinking function. We denote by ps_p the support corresponding to place p. Let $\mathcal{F}$ be the family of place projection supports:

$$\mathcal{F} = \{ps_p : p \in P\}$$

Definition 6 *The decolourization of a marking M through the colour shrinking function s on the components selected by the family $\mathcal{F}$ of place supports, denoted by $\mathcal{D}_{\mathcal{F},s}$, is a vector of $|P|$ functions; we denote by $\mathcal{D}_{ps_p,s}$ the p−th component of $D_{\mathcal{F},s}$. $\mathcal{D}_{\mathcal{F},s}(M)$ is computed by applying the p−th function to the corresponding place marking* [2]*:*

$$\forall p \in P,\ \forall c \in C(p)\ \ \mathcal{D}_{ps_p,s}(M)(p, \mathcal{D}_{ps_p,s}(c)) = M(p,c)$$

The subsequent step is the decolourization of the RG of a WN system $\Sigma = (\mathcal{N}, CS, M_0)$. It consists of the decolourization of its markings and of the transition instances labelling the arcs between markings. For this task the union of two projection support families is required: one for the places in the net and another one for the transitions.

Before defining the decolourization of RG(Σ) notice that such decolourization must be applied to place and transition components in a consistent way, taking into account the correlation between transition and place colour components induced by the functions.

Definition 7 *A family of projection supports is* consistent *with a WN iff the following conditions are satisfied*

[2] $M(p,c)$ denotes the multiplicity of the colour c in $M(p)$

1. $\forall t \in T$, *if* $\langle i,j\rangle \in ps_t(C(t))$ *and at least one predicate associated to the transition compares component* $\langle i,j\rangle$ *with another component* $\langle i,j'\rangle$, *then* $\langle i,j'\rangle \in ps_t$

2. $\forall t \in T$, $\forall p \in {}^\bullet t \cup t^\bullet \cup {}^\circ t$, $\forall \langle i,j\rangle \in ps_t(C(t)) : X_i^j$ *or* $\oplus X_i^j$ *occurs in component* $\langle i,k\rangle$ *of the function associated to the arc connecting p and t,* $\langle i,k\rangle \in ps_p(C(p))$.

3. $\forall p \in P$, $\forall t \in {}^\bullet p \cup p^\bullet \cup {}^\circ p$, $\forall \langle i,k\rangle \in ps_p(C(p)) : X_i^j$ *or* $\oplus X_i^j$ *occurs in component* $\langle i,k\rangle$ *of the function associated to the arc connecting p and t,* $\langle i,j\rangle \in ps_t(C(t))$.

Definition 8 (RG decolourization.) *Let* Σ *be a WN system and* $RG(\Sigma)$ *its reachability graph. let* $\mathcal{F}'$ *be a family of place and transition projection support functions*

$$\mathcal{F}' = \{ps_p : p \in P\} \cup \{ps_t : t \in T\}$$

and s a colour shrinking function. The decolourization of $RG(\Sigma)$ *onto* $\mathcal{F}'$ *through the shrinking function s denoted by* $\mathcal{D}_{\mathcal{F}',s}(RG(\Sigma))$ *is such that:*

- *the markings of* $\mathcal{D}_{\mathcal{F}',s}(RG(\Sigma))$ *are obtained decolourizing the markings of* $RG(\Sigma)$ *with respect to* $\mathcal{F}'$ *and s, and then aggregating the identical decolourized markings;*
- *an arc* $\langle \mathcal{D}_{\mathcal{F},s}(M_i), \mathcal{D}_{\mathcal{F},s}(M_j), t(\mathcal{D}_{ps_t,s}(c)\rangle$ *exists in* $\mathcal{D}_{\mathcal{F}',s}(RG(\Sigma))$ *iff an arc* $\langle M_i, M_j, t(c)\rangle$ *exists in* $RG(\Sigma)$*; identical decolourized arcs are aggregated into a single arc.*

Definition 9 (Redundant colour component) *A family* $\mathcal{F}'$ *of place and transition projection supports and a colour shrinking function s identify* a redundant colour component *for a given system* Σ *if the decolourization of* $RG(\Sigma)$ *onto* $\mathcal{F}'$ *through s satisfies the following* lumpability *conditions:*

$$\begin{array}{l}\forall\ M_i, M_j \in RG(\Sigma)\ :\ \ \mathcal{D}_{\mathcal{F}',s}(M_i) = \mathcal{D}_{\mathcal{F}',s}(M_j)\\ \quad \textit{if } M_i[t,c\rangle M_k \textit{ then}\\ \quad \exists c' \in C(t)\ :\ \ M_j[t,c'\rangle M_l\ \wedge\ \mathcal{D}_{ps_t,s}(c) = \mathcal{D}_{ps_t,s}(c')\\ \quad \wedge\ \mathcal{D}_{\mathcal{F}',s}(M_k) = \mathcal{D}_{\mathcal{F}',s}(M_l)\end{array}$$

The consequence of this property is that when a redundant colour component is eliminated from a system, the projected *RG* is reduced in size but still contains all the relevant information about the behaviour of the modelled system (in terms of possible transitions sequences).

We illustrate this concept by means of a simple example. In Figure 3 a WN system is shown together with its *RG*. Figure 4.a depicts the decolourized *RG* onto the family of projection supports and using the colour shrinking function s defined hereafter:
$ps_{P1} = \emptyset$, $ps_{P2} = \{\langle 2,1\rangle\}$, $ps_{P3} = ps_{t1} = ps_{t2} = \{\langle 2,1\rangle\}$
$\forall c \in C_1 : s(c) = c$; $\forall c \in C_2 : s(c) = \epsilon$, where ϵ denotes the neutral colour.
They make all elements of C_2 undistinguishable so that all C_2 components selected by the projection in places and transitions can be cancelled. Observe

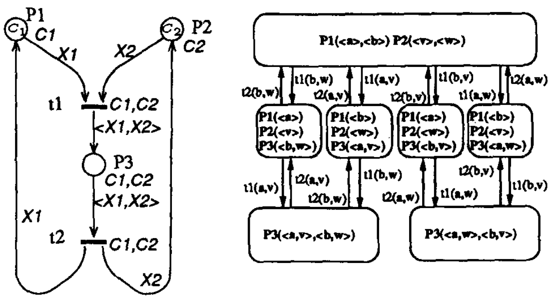
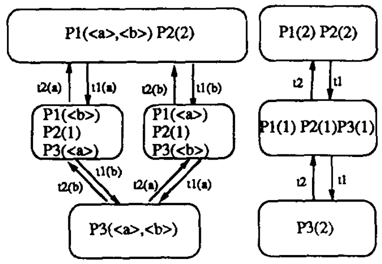


Figure 3: A simple WN system Σ_1 and its *RG*



(a) (b)
Figure 4: Decolourized *RG*s of Σ_1

that the *strong lumpability* conditions of Definition 9 are satisfied, so that the whole colour domain C_2 (i.e., every occurrence of C_2 in the place and transition colours) is redundant in this system. The *RG* can be further aggregated, resulting in the RG of Figure 4.b, by recognizing that C_1 is also a redundant basic colour domain.

3.2 Structural colour simplification

In the previous section we have defined the concept of redundant colour component based on the *RG* of a system. Since the aim of colour redundancy elimination is to reduce the size of the RG, our goal is to find some criteria that allow the detection of redundant colour components directly on the projected net structure, independently of the initial marking. In this section we define the projection of WN nets w.r.t. a family of projection supports, give some sufficient conditions on the projected net structure for the detection of redundant colour components and define a procedure that transform the net eliminating the redundant colour.

The family of projection supports used must satisfy the consistency constraints previously defined in order to obtain syntactically correct projected nets. Furthermore the procedures are intended to work on *minimal* projection support families

Definition 10 *A consistent projection support family is minimal iff no other consistent support family is* *included in it. (We say that $\mathcal{F} \subset \mathcal{F}'$ if $\forall x \in P \cup T$, $ps_x \subseteq ps'_x$ and $\exists x \in P \cup T : ps_x \subset ps'_x$.)*

Observe that a minimal consistent support family will surely contain only components belonging to the same colour class.

The decolourization procedure can then be iteratively applied using different minimal projection supports. The application order however is not irrelevant i.e., if in a given net there are two minimal colour components that satisfy the decolourization conditions, it could happen that after having eliminated the first one, the second is no more redundant in the reduced net. In this paper we do not give any decolourization order criteria.

It is important to observe that given a colour class C_i, there may exist many minimal consistent projection support families containing disjoint sets of components of C_i. In this case it may happen that C_i can be shrinked according to a given function s for one of these projection support family, but the same kind of reduction do not apply for the others. The solution to this apparent problem is in the definition of consistent projection support family; in fact since the components belonging to different consistent projection support families do not share objects (i.e. colour elements contained in the components selected by the first family cannot be "mixed" with those contained in the components selected by the second family) then we can use a different copy of C_i for each family and consider them as if they were different colour classes.

The procedure that checks whether a given projection support family $\mathcal{F}$ is structurally redundant works on the *projection over* $\mathcal{F}$ of the net $\mathcal{N}$. The definition of projection of $\mathcal{N}$ over $\mathcal{F}$ follows:

Definition 11 *The projection of a place (transition) colour domain $C_J = \bigotimes_{i \in I} C_i^{e_i}$ with respect to a projection support ps is:*

$$\mathcal{P}_{ps}(C_J) = \bigotimes_{i \in I} C_i^{e'_i}$$

where [3] $e'_i = \sum_{j=1}^{e_i} \Gamma_{ps(C_J)}(\langle i,j \rangle)$

Definition 12 *The projection a WN $\mathcal{N}$ onto a family $\mathcal{F}$ of place and transition projection supports, denoted by $\mathcal{N}' = \mathcal{P}_{\mathcal{F}}(\mathcal{N})$, is a net such that:*
$P' = P$; $\forall p \in P'$, $C'(p) = \mathcal{P}_{ps_p}(C(p))$
$T' = T$; $\forall t \in T'$, $C'(t) = \mathcal{P}_{ps_t}(C(t))$
$\forall t \in T$, only the predicates that make reference to components selected by ps_t are kept in $\mathcal{N}'$.
$\forall p \in P$, $\forall t \in T$, if $W^+(p,t)$ (or W^- or W^H) is defined as $\sum \delta \bigotimes_{i,k} W^+_{i,k}$ the corresponding projected arc function is:
$\mathcal{P}_{ps_p}(W^+(p,t)) = \sum \eta\delta \bigotimes_{i,j:\langle i,j \rangle \in ps_p(C(p))} W^+_{i,k}$ *where the constant η is computed on the excluded components of each element as follows:*

$$\eta = \prod_{i,j:\langle i,j \rangle \notin ps_p(C(p))} |W^+_{i,j}|$$

[3] $\Gamma_{ps(C_J)}$ is the characteristic function of the set $ps(C_J)$

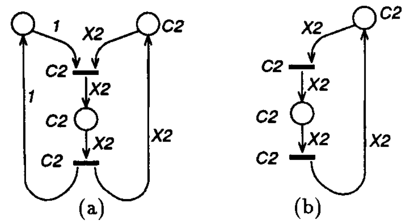


Figure 5: Projection and cancellation of neutral subnet

Examples of cardinality computed on functions: $|kS_i| = k|C_i|$; $|S_{i,q}| = |D_{i,q}|$; $|X_i + \oplus X_i| = 2$; $|k(S_i - X_i)| = k(|C_i| - 1)$. Observe that the cardinality $|W^+_{i,j}|$ could not be computed in case of functions with predicates. For this reason we focused only on nets without function predicates.

Now let us describe a procedure that decides whether a given minimal projection support family identifies a redundant colour component. The procedure builds the appropriate shrinking function to be used in conjunction with the family of projection supports in order to obtain the decolourized net.

3.3 A sufficient condition for colour shrinking

The following procedure accepts a WN and a minimal consistent family $\mathcal{F}$ of projection supports as input and decides if the colour components selected by the projection supports are redundant. The procedure is made of two steps:

1. Projection of the net on $\mathcal{F}$ and cancellation of the neutral subnet; let $\mathcal{N}'$ be the net resulting from this step.

2. Redundancy check on $\mathcal{N}'$.

The first step should be clear. We illustrate it on the net and projection support of Figure 3. The projection gives the net in Figure 5.a. The resulting net after the cancellation of the neutral part is depicted in Figure 5.b.
The second step checks if the net $\mathcal{N}'$ resulting from the first step satisfies the properties listed below. The colour redundancy test is performed for each static subclass $D_{i,q}$ separately.

Definition 13 (Colour Redundancy)
Redundancy check for static subclass $D_{i,q}$. For each transition t in $\mathcal{N}'$ the following properties must be satisfied:

1. *the set of functions that label the input and inhibitor arcs of t can only be functions of type X_i^j, $\oplus X_i^j$ and linear combinations of the two. If the class is not ordered and the place connected to the arc is colour-safe*

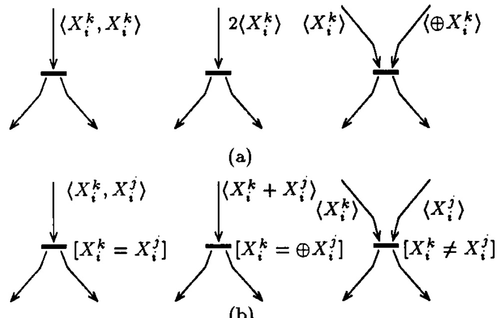


Figure 6: Some examples of function combinations that prevent decolouring

[4] *this condition can be relaxed and diffusion functions (S_i or $S_{i,q}$) are allowed.*

2. *Let X_i^j be a function that can select one element in $D_{i,q}$ (i.e. the transition predicates do not imply that $d(X_i^j) \neq q$). No more than one occurrence of either X_i^j or $\oplus X_i^j$ is allowed in the labelling functions of all input and inhibitor arcs of t. Examples of forbidden situations are depicted in Figure 6.a.*

3. *If two functions X_i^j and X_i^k that can both select an element in $D_{i,q}$ occur in the input or inhibitor arc functions of t, the transition predicates do not imply $[X_i^j = X_i^k]$ or $[X_i^j = \oplus X_i^k]$ or $[\oplus X_i^j = X_i^k]$ or their negation. Examples of forbidden situations are depicted in Figure 6.b.*

In the particular case of ordered classes, the above requirements should be verified for all static subclasses in which a colour domain is partitioned. Furthermore, predicates that refer to static subclasses should not occur on the selected transitions.[5]

If these properties are satisfied, then either the colour components under study are never used in a synchronization or into a conflict resolution (due to the conditions on the functions combination on input and inhibitor arcs), or the synchronizations can be solved just by counting the number of tokens in the place (due to the colour-safeness property of the place). Hence the static subclass $D_{i,q}$ can be decolourized by a function s that maps each element $c \in D_{i,q}$ onto the same element $c'_{i,q}$, the unique element of the shrinked subclass $D'_{i,q}$. Otherwise $D'_{i,q}$ is set equal to $D_{i,q}$ and the shrinking function s is the identity for the elements in $D_{i,q}$.

A further reduction in a colour class cardinality could be achieved by "merging" some decolourized static subclasses when possible. The presence of the predicate $[d(X_i^j) = d(X_i^k)]$ or its negation on any transition of the projected net prevents this operation. Provided that this kind of predicate does not occur, all static subclasses $D_{i,q}$ that satisfy the decolourization conditions and such that the predicate $[d(X_i^j) = q]$ does not occur on any transition in the projected net can be merged into a unique static subclass $D'_{i,q'}$ of cardinality one. Let Q be the set of mergeable static subclass indexes. The shrinking function s must be modified as follows:

$$\forall c \in \cup_{q \in Q} D_{i,q} : s(c) = c_{i,q'}$$

The conditions should be tested for each subclass $D_{i,q}$, if the conditions are satisfied at least for one of them, then the net $\mathcal{N}$ can be transformed using the following procedure. The decolourized net is denoted $\mathcal{D}_{\mathcal{F}',s}(\mathcal{N})$.

Definition 14 (Net decolourization) *with respect to a minimal consistent family $\mathcal{F}'$ of projection supports and a shrinking function s:*

- *The newly defined, shrinked colour class C'_i is added to the set of colour classes.*
- *For each place and transition colour components selected by the corresponding projection support the colour class C_i is substituted by the new class C'_i.*
- *Function components selected by the projection support of the corresponding place are transformed as follows:*

  X_i^j remains unchanged

  $S_{i,q}$ is substituted by $\frac{|D_{i,q}|}{|D'_{i,q}|} S_{i,q}$

  S_i is substituted by $\sum_q \frac{|D_{i,q}|}{|D'_{i,q}|} S_{i,q}$
- *Transition predicates that refer to completely decolourized classes are substituted by "true." Predicates referring to classes that are only partially decolourized remain unchanged, unless the predicate is of the form $X_i^j \neq X_i^k$ (that relate functions on input or inhibitor arcs —say X_i^j— to functions on output arcs, otherwise the decolourization would not be allowed). In that case it should be transformed taking into account the cardinality of the static subclasses in the original system. If all the static subclasses in the original net have cardinality greater than one then the predicate is changed to true otherwise it becomes*[6]

$$\Big(\bigvee_{\substack{\forall q: |D_{i,q}|>1 \,\wedge\, D_{i,q} \\ \text{has been shrinked}}} [d(X_i^j) = q] \Big) \vee [X_i^j \neq X_i^k] \quad (3)$$

[4]A place is colour-safe iff its marking never contain two copies of the same tuple.

[5]The reason for this further requirement is due to the rule of total decolourization of ordered classes that forces an ordered class to be either completely decolourized or remain unchanged.

[6]The reason for using the above predicate is that different objects may belong to the same static subclass provided that it has a cardinality greater than one. When a static subclass of cardinality greater than one is shrinked, different colour elements in the original class could become equal in the decolourized class.

Since the predicate in Equation (3) depends on the cardinality of the original static subclasses $D_{i,q}$, it is not completely parametric. The possible solution to this problem could be to consider for each static subclass that is involved in the predicate the two cases $|D_{i,q}| = 1$ and $|D_{i,q}| > 1$ and build the resulting reduced nets for all the possible cases. When the net is instantiated, it is possible to choose the correct reduced net among those generated by the reduction algorithm.

If the possibility of merging static subclasses is exploited a further transformation is needed on the reduced net predicates. In fact predicates of type $[d(X_i^j) = q]$ must be adjusted taking into account the new static subclasses indexes. Moreover, predicates of type $[X_i^j \neq X_i^k]$ must be transformed in the following boolean expression (where Q is the set of indexes of the merged subclasses)

$$[X_i^j \neq X_i^k] \vee (\bigvee_{\forall q \in Q} [d(X_i^j) = q]) \vee (\bigvee_{\substack{\forall q \notin Q : |D_{i,q}| > 1 \wedge \\ D_{i,q} \text{ shrinked}}} [d(X_i^j) = q])$$

instead of the one in Equation (3).

We have already mentioned that when a class is completely decolourized (i.e. $|C_i'| = 1$) the corresponding components can be "removed" from the net. When performing this cancellation however, the coefficients of the eliminated function component must be taken into account: the cardinality of each eliminated function component becomes a factor multiplying the function tuple.

It is possible to prove the following statement:

Proposition 1 *Let $\mathcal{F}$ be a family of minimal projection supports, consistent with net $\mathcal{N}$, that satisfies the above properties. Let s be the corresponding colour shrinking function. Then the RG of every system $\Sigma = (\mathcal{N}, CS, M_0)$ satisfies the lumpability conditions and*

$$RG(\mathcal{D}_{\mathcal{F},s}(\mathcal{N}), \mathcal{D}_{\mathcal{F},s}(M_0)) = \mathcal{D}_{\mathcal{F},s}(RG(\mathcal{N}, CS, M_0)) \quad (4)$$

3.4 SRG aggregations versus decolourization

Let us discuss the differences between the aggregations obtained through the *SRG* generation [5] and those obtained by the decolourization procedure proposed in this paper.

The intuitive motivation for the aggregations achievable through decolourization and not through the SRG is the fact that the SRG aggregations are performed step by step, and preserve the minimal information necessary to solve every type of synchronization and/or conflict resolution that could be found in the future evolution of the net. The SRG construction algorithm has no knowledge of the global structure of the net, so that it cannot forecast whether certain pieces of information contained in the current state will ever be required to take decisions in future states.

On the contrary, the decolourization procedure considers whole subnets through which colour elements may flow and checks whether a colour specification is needed or not to take a decision about the enabling of any transition in the subnet. Therefore, in some sense, it looks ahead to the possible future needs of colour information. In [7] a case is shown in which the decolourization aggregation is stronger than the corresponding SRG aggregation.

4 Some examples

Let us now apply the decolourization method to two example models.

4.1 A multiprocessor system

The net in Figure 7 models a multiprocessor system. The WN is a variation of a WN model already studied in [9,10,6]. It is composed of n processors, each one owning a local memory. All processors can access their local memories using local busses; a processor can also access the local memory of another processor using a global bus. The global bus is unique and is shared by all processors. Hence each processor can be in one of five states: active (i.e. performing a computation), waiting for access to the local memory, accessing its local memory, waiting for access to an external memory, or accessing an external memory. The external memory requests have preemptive priority over the local memory access.

We need only one class of objects: the class P of processors. Some transition colour domains are defined as the Cartesian product $P \bigotimes P$. For graphical reasons in the figure we have used x and y instead of X_1^1 and X_1^2 on the input and output arcs of these transitions.

The five process states listed above are represented by the places as follows: "active" state is represented by *Active*; "waiting for and performing a local memory access" states are represented by *OwnMemAcc*; "waiting for an external memory access" state is represented by *Queue* ; "performing an external memory access" state is represented by *Preempt*, *NPY* or *NPP* depending on the preemptive effect that the access has or could have on a local memory access.
Any transition connecting two of the above places represent the transition between the corresponding states.

This net is covered by a unique, minimal consistent projection support family that includes all colour components. It is easy to verify that all decolourization requirements are satisfied: indeed no synchronization depends on the identity of the tokens since there is no relation between functions x and y.

In this case the unique colour set is not partitioned into static subclasses and it can be completely decolourized. All colour annotations on places, transitions, and arcs can be cancelled, so that the decolourized net is a P/T net. All arcs in the decolourized

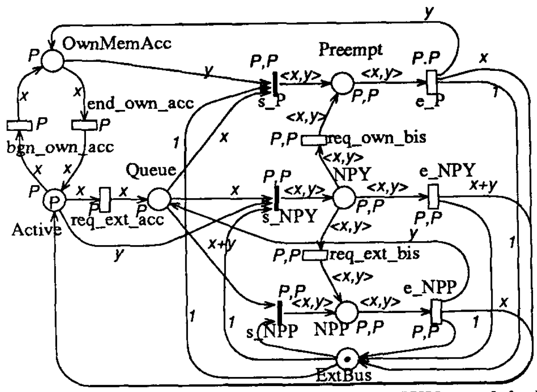


Figure 7: A multiprocessor system WN model that can be completely decolourized

net have weight one except the arcs from *Queue* to *s_NPP* and from *e_NPY* to *Active* that have weight two.

4.2 A message exchange system

We apply now the decolourization procedure to the net in Figure 1 introduced in Section 2. From the decolourization point of view this example is slightly more complicated than the previous one because of the presence of predicates and of static subclasses.

Let's describe in some detail the meaning of the net nodes. Place *Idle* represent idle sites. Places *TxBuf* and *RxBuf* represent the transmitter and receiver packet buffers respectively. The former can contain both data and acknowledgement packets. Transition *StartSnd* represents the generation of a set of data packets composing a message. The predicate labelling this transition expresses the constraint that a site cannot send messages to itself. Transition *Tx* represent the transfer of a packet from the transmitter buffer to the receiver buffer. Transition *Rx* represents both the receipt of a message and the transfer of an acknowledgement packet to the sender site buffer while *EndSnd* represents the reception of the acknowledgement packet to the site.

There are two minimal consistent projection support families in this case: the first one contains the complete sites class; the second one contains the complete packets class.

Concerning the packets class, the two static subclasses have to be considered separately. The projected net satisfies the first decolourization requirement for neither of the two static subclasses: in fact there are diffusion functions S on the input arcs of transitions *Rx* and *EndSnd*. Nevertheless, if places *RxBuf* and *TxBuf* were colour-safe, the decolourization would still be possible.

This possibility depends on the initial marking as well as on the net structure. Assuming that the initial marking of place *Idle* was the whole sites class and

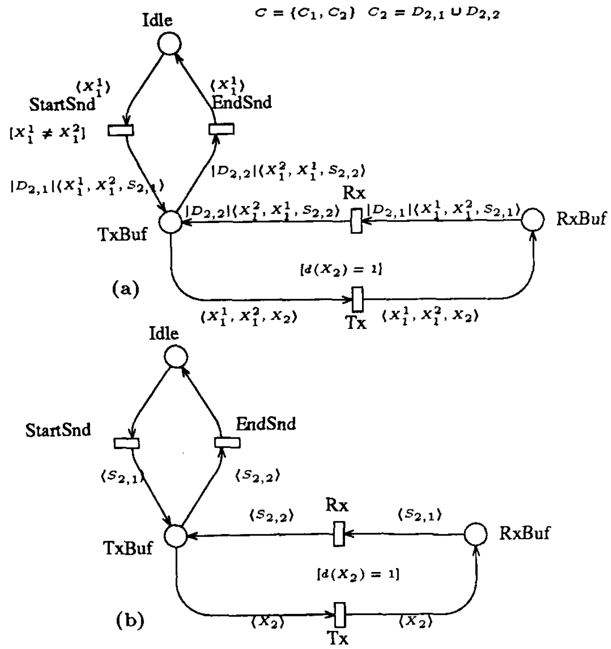


Figure 8: Decolourized models of the message exchange protocol

that the other places were initially empty, the colour-safeness condition would be satisfied (the statement could be proven by using the P- invariants, but this proof is out of the scope of this paper). Under the additional condition that the net is colour-safe the two static classes can be decolourized but they cannot be merged into a unique class because of the predicate $[d(X_2) = 1]$ on transition *Tx*. The decolourized net is shown in Figure 8.a.

The net obtained by projection on the sites class projection support family does not satisfy the first decolourization condition unless $|D_{2,1}| = |D_{2,2}| = 1$. In fact the projected function on the arc from *RxBuf* to *Rx* is $|D_{2,1}|\langle X_1^1, X_1^2\rangle$ and similarly the projected function on the arc from *TxBuf* to *EndSnd* is $|D_{2,2}|\langle X_1^2, X_1^1\rangle$. Of course if $|D_{2,1}| = |D_{2,2}| = 1$ then the packets class is already minimal; furthermore the sites class can be completely decolourized because the synchronizations on transitions *Tx*,*Rx*, and *EndSnd* do not depend on the colour of the two elements X_1^1 and X_1^2. From a semantic point of view, the condition $|D_{2,2}| = 1$ is quite natural to enforce since there is only one type of acknowledmenet packets. Instead the condition $|D_{2,1}| = 1$ means that no splitting of messages into smaller data packet fragments is performed and it is usually not verified. The decolourized net obtained in this case is depicted in Figure 8.b. Notice that in this case the predicate associated with transition *StartSnd* has been removed because the sites class has been completely decolourized.

5 Conclusions

The possibility of defining redundant WNs is important from a modelling point of view since it simplifies the task of setting up a model. However, the presence of redundancies in the marking description may hamper the practical possibility of model analysis. Hence the development of algorithmic model simplification techniques is crucial for a practical utilization of a CPN formalism.

In this paper the concept of colour structure simplification has been introduced for WNs. The simplification technique implies a reduction of the size of the reachability graph while preserving the behavioural properties of the net. This work significantly extends the initial results on colour simplification presented in [7]. The reduction is obtained by exploiting some colour symmetries of the net.

Unlike the symbolic reachability graph technique [5], the symmetries are detected and eliminated at the net structural level, so that the simplication obtained is parametric on (independent of) the cardinality of the basic colour sets. The decolourization and the SRG techniques are independent and complementary, so that they may be combined to optimize the analysis of WNs.

From the analysis of the second example of decolourization presented in Figure 8 it is evident that additional parametric information is usually needed besides the structure of the WN. In [11] the concept of model family has been formally introduced in the framework of P/T nets. The next step in this research, in addition to the experimental implementation of the techniques proposed in this paper, will be the development of an analogous concept of WN families containing a specification of the instantiation constraints for the colour sets and the initial marking. The formal specification of these additional information deriving from the semantics of the family of systems to be modelled will allow one to take information into account in the decolourization procedure.

Acknowledgements

The authors are grateful to the anonymous referees for their detailed comments and suggestions on the draft of this paper.

References

[1] K. Jensen. Coloured Petri nets and the invariant method. *Theoretical Computer Science*, 14:317–336, 1981.

[2] H.J. Genrich. Predicate/transition nets. In W. Brawer, W. Reisig, and G. Rozenberg, editors, *Advances on Petri Nets '86 - Part I*, volume 254 of *LNCS*, pages 207–247. Springer Verlag, Bad Honnef, West Germany, February 1987.

[3] G. Balbo, G. Chiola, S.C. Bruell, and P. Chen. An example of modelling and evaluation of a concurrent program using coloured stochastic Petri nets: Lamport's fast mutual exclusion algorithm. *IEEE Transactions on Parallel and Distributed Systems*, 1991. accepted for publication.

[4] P. Huber, A.M. Jensen, L.O. Jepsen, and K. Jensen. Towards reachability trees for high-level Petri nets. In G. Rozenberg, editor, *Advances on Petri Nets '84*, volume 188 of *LNCS*, pages 215–233. Springer Verlag, 1984.

[5] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. On well-formed coloured nets and their symbolic reachability graph. In *Proc. 11^{th} International Conference on Application and Theory of Petri Nets*, Paris, France, June 1990. Reprinted in *High-Level Petri Nets. Theory and Application*, K. Jensen and G. Rozenberg (editors), Springer Verlag, 1991.

[6] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. Stochastic well-formed coloured nets and multiprocessor modelling applications. Technical Report 90/41, Université Paris 6, 4 Place Jussieu, 75252 Paris Cedex 05, France, October 1990. IBP Tech. Report, (submitted for publication). Also in *High-Level Petri Nets. Theory and Application*, K. Jensen and G. Rozenberg (editors), Springer Verlag, 1991.

[7] G. Chiola and G. Franceschinis. A condition for behaviour-preserving colour simplification in Well-Formed coloured nets. In *Proc. ISCAS'91*, Singapore, June 1991. Special session on Net-based analysis and Design of Concurrent Distributed Event Systems.

[8] G. Chiola and G. Franceschinis. A structural colour simplification in well-formed coloured nets. Technical report, Dip. di Informatica, Univ. di Torino, 1991.

[9] M. Ajmone Marsan and G. Chiola. Construction of generalized stochastic Petri net models of bus oriented multiprocessor systems by stepwise refinements. In *Proc. 2^{nd} Int. Conf. on Modeling Techniques and Tools for Performance Analysis*, Sophia Antipolis, France, June 1985. ACM.

[10] G. Chiola and G. Franceschinis. Colored GSPN models and automatic symmetry detection. In *Proc. 3rd Intern. Workshop on Petri Nets and Performance Models*, Kyoto, Japan, December 1989. IEEE-CS Press.

[11] G. Chiola, S. Donatelli, and G. Franceschinis. On parametric P/T nets and their modelling power. In *Proc. 12^{th} International Conference on Application and Theory of Petri Nets*, Aarhus, Denmark, June 1991.