Compositional construction and analysis of Petri net systems

Isabel C. Rojas M.

Doctor of Philosophy
University of Edinburgh
1997

Abstract

Most Petri net (PN) based modelling formalisms represent the system modelled as a flat net. This may not clearly reflect the elements that participate in the system and the way they communicate or interact. It can also be difficult to determine the model's behaviour or prove some of its properties. Viewing the model as a set of components that interact is more appropriate, especially for models of parallel and distributed systems.

In this thesis a compositional method for the construction and analysis of Well-formed nets (WNs) systems is presented. WNs allow a natural representation of complex distributed systems, maintaining the same expressive power as the unconstrained coloured net formalisms. The main motivation of this work has been to offer an appropriate method for the specification, design and analysis of parallel and distributed systems. The set of composition operations defined is based on the operators of Process Algebras (PA). Mimicking the PA operators allows us to benefit from the compositional nature of PA. The definition of the composition operations has taken into account the peculiarities and characteristics of the PN formalisms, such as synchronisation, state evolution and token flow. The models obtained by applying the compositional method proposed are termed *compositional WN* (*cWN*) systems.

To consolidate a framework for the compositional construction and analysis of *cWN*s, we study the construction of structural and state space information about a *cWN* using information about its sub-components. The matrix description of the model—known as the incidence matrix—is shown to be obtainable using the incidence matrices of its sub-components, together with the knowledge of the composition operations used. By studying the relation between the resulting incidence matrix and the incidence matrices of its sub-components, methods are proposed to obtain the semiflows of the *cWN* model, using the semiflows of its sub-components. New, higher-level semiflows are defined, based on the structured definition of colours and arc functions of WN models.

We show how the state space of a higher-level component can be built from the state spaces of its sub-components. This leads to the definition of a grouping of markings, termed a *composed marking*. It is proved that state space analysis over composed markings allows the verification of state space properties of the

complete system, such as reachability, absence of deadlock and liveness, using the reduced state space.

The concepts and propositions introduced are illustrated throughout the dissertation by the use of a series of examples. The methods proposed are applied over a model of a flexible manufacturing system, as a way to consolidate the understanding of the methodology.

As a step towards the definition of a methodology for the performance oriented compositional construction and analysis of Stochastic Well-formed net systems modelling parallel and distributed systems, we study the extension of the compositional operations and methods proposed to support the incorporation of time specifications of the system modelled.

Acknowledgements

First of all I want to express my most sincere gratitude to my supervisor Dr. Jane Hillston, for her constant support, advice and encouragement. I will also like to thank Dr. Rob Pooley for his guidance and support during my studies and my examiners, Dr. Giuliana Franceschinis and Dr. Murray Cole, for their useful comments and recommendations.

Thank you very much to the Venezuelan National Scientific Research Council (CONICIT) for their financial support during my entire studies.

My thanks also go to my family and friends who encouraged me during my studies.

Last but not least, I thank Thomas, my best result, for his patience, support, love and encouragement.

Declaration

I declare that this dissertation was composed by me and that the work contained therein is my own, except where explicitly stated otherwise in the text. Some of the work presented has been previously published in [Roj95, RM96].

(*Isabel C. Rojas M.*)

Table of Contents

Table of Notation

P	set of places
T	set of transitions
In	set of input arcs ($In \subseteq P \times T$)
O	set of output arcs ($O \subseteq T \times P$)
H	set of inhibitor arcs ($H \subseteq P \times T$)
W^-	set of input arc functions
W^+	set of output arc functions
W	incidence matrix
C	set of basic colour classes
C_i	colour class i
$D_{i,q}$	static subclass q of C_i
$\tilde{C}_i$	set of static subclasses of a basic colour class C_i
n_i	number of static subclasses of colour class C_i
Σ	finite set of types called *colour sets*
J	functions that defines the colour domains of places and transitions
$C(\tau)$	colour domain of an element $\tau \in P \cup T$
M	an "ordinary" marking
M_0	the initial marking
$E(M)$	set of enabled transitions in marking M

$E_{im}(M)$	set of enabled immediate transitions in marking M
$p^\bullet$	set of transitions for which p is input
$^\bullet p$	set of transitions for which p is output
$t^\bullet$	set of output places of transition t
$^\bullet t$	set of input places of transition t
$M[t\rangle M'$	M' is produced by the firing of t in M
$M[\sigma\rangle M'$	M' is reachable by the firing sequence σ from M
Δ	set of delays associated with the transitions of a timed PN
$\mathcal{F}$	set of exponentially distributed firing rates associated with each of the transitions of a PN system
θ	function that assigns rates to timed transitions and weights to immediate transitions
π	function that assigns priorities to the transitions
$Bag(A)$	denotes the set of finite multisets over a set A
$\Phi(t)$	standard predicate associated with a transition t
ES	set of entry places of a $cSWN$
FS	set of final places of a $cSWN$
ES'	set of entry places participating in a composition operation
FS'	set of final places participating in a composition operation
$L; R$	sequential composition of components L with component R
Γ	function that identifies the places to be fused under a sequential composition operation
$L + R$	choice composition between two cWNs (L and R)
$L + R/\{w(L), w(R)\}$	choice composition between two $cSWN$s (L and R)

$Choice(Q)$	set of places of component Q participating in a choice composition
$L\|R$	independent parallel composition between components L and R
$L\|_c R$	competing parallelism between components L and R
$S\|_c S$	internal competing parallelism within component S
Λ	one-to-one function that defines the pairs of places to be fused in a competing parallelism composition
$CL(S)$	closing operation on component S
Θ	function that identifies the pair of places to be fused in a closing operation
$L \sim R/\{t_1, t_2\}$	untimed synchronisation between components L and R
$L \diamond R/\{t_1, t_2\}$	patient communication between components L and R
$L \odot R/\{t_1, t_2\}$	impolite communication between components L and R
$\\|z\\|$	support of a vector z
$G = (\xi, \cdot)$	group of permutations on $C_1, \ldots, C_n$ with $\xi = \{s = \langle s_1, \ldots, s_h, s_{h+1}, \ldots, s_n \rangle\}$
$\bar{c}$	a symbolic colour
$\widehat{x^R}$	extension of a symbolic (P-) T-semiflows with respect to the (places) transitions of the sub-component R
$\widehat{x_c^R}$	choice extension of a symbolic (P-) T-semiflows with respect to the (places) transitions of the sub-component R
$\mathcal{M}$	a symbolic marking
$\mathcal{M}_0$	initial symbolic marking
Z_i^j	dynamic sub-class j of colour class C_i
X_i^j	the identity function
$\oplus^u X_i^j$	the u-successor function

$S_{i,q}$	the broadcast (synchronisation) function
$[t, \lambda, \mu]$	symbolic (or composed) transition instance
λ	set of functions that assign dynamic subclasses to the parameters of a transition
μ	set of functions that assign objects of the dynamic subclasses specified by λ to the parameters of a transition
$\widehat{\mathcal{M}}$	a composed marking
$\widehat{\mathcal{M}_0}$	initial composed marking
$\mathcal{E}$	marking component of a composed marking
$\widehat{\mathcal{M}^L_\mathcal{N}}$	composed sub-marking of the composed marking $\widehat{\mathcal{M}_\mathcal{N}}$ with respect to the left sub-component of $\mathcal{N}$
$\overline{\widehat{\mathcal{M}^R_\mathcal{N}}}$	complement of a composed sub-marking $\widehat{\mathcal{M}^L_\mathcal{N}}$
$\widehat{Z^j_i}$	composed dynamic sub-class j of colour class C_i
dyn_ss	set of dynamic subclasses in a composed dynamic subclass
$\widehat{C}_i$	set of composed dynamic sub-classes of colour class C_i

Chapter 1

Introduction

Petri nets (PNs) [Rei85, Pet81] are a very well established formalism for the modelling of concurrent systems. However, most PN models represent the system modelled as a flat net. This may not clearly reflect the elements that participate in the system and the way they communicate or interact. It can also be difficult to determine the model's behaviour or prove some of its properties. The model's structure can be very different from the functional specification of the system.

Viewing the model as a set of components that interact is more appropriate, especially for models of parallel and distributed systems. As stated in [KMF90], the underlying model of distributed systems is that of loosely coupled components running in parallel and communicating by message passing. Based on sub-models representing components of a system, a model can then be developed as the composition of sub-models. The structure of the resulting model reflects that of the system. Models of components can be developed by different modellers and, in principle, libraries of re-usable components can be formed.

Well-formed nets (WN) [CDFH91] allow a natural representation of complex distributed systems, maintaining the same expressive power as the unconstrained coloured PN formalisms. This dissertation investigates the benefits of the compositional construction of WN systems, to offer—as a consequence—an appropriate method for the specification, design and analysis of PN models of parallel and distributed systems. The main contribution of this dissertation is the study and proposal of methods that use knowledge of the properties of the sub-components of a WN system, and of the way in which they were composed, to build the structural and state space information about the WN system. This contributes to the efficient solution of large PN systems. The outline of the dissertation is as follows.

The general background material for this dissertation is presented in Chapters 2 and 3. Chapters 4, 5 and 6, each contain background material specific to the work presented in the chapter. Chapter 2 introduces WNs and general concepts of the PN theory which are necessary for the definition and understanding of the compositional methods proposed. Chapter 3 reviews the existing work in the area of the incorporation of compositionality into the PN formalism, describing the different approaches and motivations. Based on a set of criteria for the construction and analysis of models of parallel and distributed systems, the chapter concludes by discussing the advantages and disadvantages of the methods reviewed, determining what can be learned from each of these methods.

The criteria for the construction and analysis of models of parallel and distributed systems, introduced in Chapter 3, act as the basis for the definition of the compositional methods proposed. Chapter 4 defines the set of operations for the compositional construction of WN systems. The set of composition operations defined is based on the operators of Process Algebras (PA). Mimicking the PA operators allows us to benefit from the compositional nature of PA. The peculiarities and characteristics of the PN formalism, such as synchronisation, state evolution and token flow, are taken into account when defining the set of composition operations. The WN components obtained by using the compositional construction method defined are termed *Compositional WN*s (*cWN*s). As a guide to the modeller, the factors that must be considered when composing a system are discussed. To illustrate the use of the composition operations for the construction WN systems two examples are presented.

To consolidate a framework for the compositional construction and analysis of *cWN*, the construction of structural and state space information about a *cWN* using information about its sub-components is studied. Chapter 5 is devoted to the compositional construction of structural information, specifically *place* and *transition semiflows* of *cWN* models. The matrix description of the model—known as the *incidence matrix*—is shown to be obtainable using the incidence matrix of its sub-components, together with knowledge of the composition operation used to compose it. By studying the relation between the resulting incidence matrix and the incidence matrix of the sub-components, methods are proposed to obtain the semiflows of *cWN* models, using the semiflows of the sub-components. New, higher-level semiflows are defined, based on the structured definition of colours and arc functions of WN models, which exploit the presence of symmetries in the model. An example is presented to illustrate the method proposed.

Chapter 6 studies the compositional construction of the state space of *cWN* systems. State space analysis techniques are based on the graph that contains all possible evolutions of the PN system, known as the *reachability graph*. The construction of the state space of a component using the state space of its subcomponents leads to the definition of a new type of marking grouping termed *composed marking* and, as a consequence, a new type of transition firing termed *composed firing*. We study the relation between the *symbolic* reachability graph of a WN system and a *composed* reachability graph of a *cWN* system. It is proved that state space analysis over composed markings allows the verification of state space properties of the complete system, such as reachability, absence of deadlock and liveness, using the reduced state space. To illustrate the application of the method proposed a series of small examples is presented throughout the chapter.

To strengthen the understanding of the compositional methods proposed, in Chapter 7 we present an example to which we apply them. The example is a modification of the flexible manufacturing system presented in [CT91]. The system is constructed in a compositional manner using the composition operations defined and applying the guidelines presented in Chapter 4.

In Chapter 8 studies the extension of the compositional operations and methods proposed to support the incorporation of time specifications of the system modelled. *Compositional Stochastic WNs* (*cSWN*) are defined based on the definition of *cWN*s and using Stochastic Well-formed nets (SWNs) as the basic formalism. We study the necessary changes to the compositional methods proposed in order to support the definition of *cSWN*s.

Finally, Chapter 9 summarises the main results of this work. The methodology proposed is analysed according to the criteria presented in Chapter 3 for the compositional construction and analysis of PN models of parallel and distributed systems, and it is compared with the existing work in the area. The chapter concludes by proposing topics of further research in the area of compositional construction and analysis of PN systems and possible future developments of the compositional methodology proposed.

Chapter 2

Concepts of Petri net theory

2.1 Introduction

This chapter contains part of the background information for this dissertation. In the first section we introduce the basic concepts of the Petri net formalism; gradually, in subsequent sections, we augment the definition of basic Petri nets, to arrive at Well-formed nets. This is the Petri net class that forms the basis of the compositional Petri net methodology proposed in this dissertation.

2.2 Petri nets

Petri nets (PNs) [Rei85, Pet81] are a graph based mathematical formalism for the description of concurrent systems. A PN can be seen as directed bipartite graph whose set of nodes is divided into a set of *places* and a set of *transitions*. Arcs can go from places to transitions (input arcs) or transitions to places (output arcs). In general, transitions represent events, while places represent conditions, although there are some cases in which they are given different interpretations. Graphically places are represented by circles and transitions by rectangles.

Formally, a PN *model* is defined as a 4-tuple:

$$PN = \langle P, T, In, O \rangle$$

where:

P	is a finite set of places;
T	is a finite set of transitions, $P \cap T = \emptyset$;
$In \subseteq P \times T$	is the set of *input* arcs;
$O \subseteq T \times P$	is the set of *output* arcs;

In principle, places and transitions can be connected by more than one input or output arc. These multiple arcs are replaced by a single *weighted* arc, where the weight or *multiplicity*, corresponds to the original number of input (or output) arcs connecting p with t. The functions $W^- : In \to \mathbb{N}^+$ and $W^+ : O \to \mathbb{N}^+$, return the multiplicity of the input and output arcs, respectively. A place p is an *input* place of a transition t if there is an arc from p to t. Similarly, p is an *output* place of t if there is an arc from t to p. We will denote by ${}^\bullet t$ and $t^\bullet$ the sets of input and output places, respectively, of a transition t; and by ${}^\bullet p$ and $p^\bullet$ the set of input and output transitions of p, respectively.

The matrix description of a PN model is termed the *incidence matrix*. The incidence matrix, denoted W, is a matrix of dimension $|P| \times |T|$ whose entries $W(p,t)$ are defined as:

$$\forall p \in P,\ \forall t \in T,\ W(p,t) = W^+(p,t) - W^-(p,t)$$

The basic definition of a PN model can be extended by incorporating a third type of arc, called *inhibitor arcs*, which also connect places with transitions ($H \subseteq P \times T$). The function $W^h : H \to \mathbb{N}^+$ returns the multiplicity of an inhibitor arc. We will denote by ${}^h t$ the set of inhibiting places of a transition t. An inhibitor arc is represented by a line with a circled ending.

The state of the system is modelled in a PN by the distribution of *tokens* over the places of the PN. Tokens are indistinguishable markers that reside in places. Graphically they are represented as filled circles. The state of a PN (usually referred to as *marking*) is determined by a marking function $M : P \to \mathbb{N}^+$. $M(p)$ specifies the number of *tokens* contained in a place p. The *initial marking* (M_0) of the net represents the initial distribution of tokens in the places of the net. A PN *system* is given by a PN model plus an initial marking. Figure 2.1 shows a very simple example of a PN system. The multiplicity of the arcs has been omitted, assuming that they all have multiplicity 1.

The *dynamic* behaviour of a PN system is specified by the *enabling* and *firing* rules. A transition t is said to be *enabled* if each of its input places has at least as many tokens as the multiplicity of the input arc from that place to t, and if each inhibiting place p_h of the transition has less tokens than the multiplicity of the inhibitor arc from p_h to t. Formally, t is enabled in M if,

$$\forall p \in {}^\bullet t,\ W^-(p,t) \leq M(p) \quad \text{and} \quad \forall p \in {}^h t,\ W^h(p,t) > M(p)$$

The set of enabled transitions in a marking M is denoted by $E(M)$.

Enabled transitions can *fire*, removing from each input place as many tokens as the multiplicity of its corresponding input arc, and placing in each output place as many tokens as the multiplicity of its corresponding output arc. The firing of a transition t, from a marking M, producing a marking M', is denoted

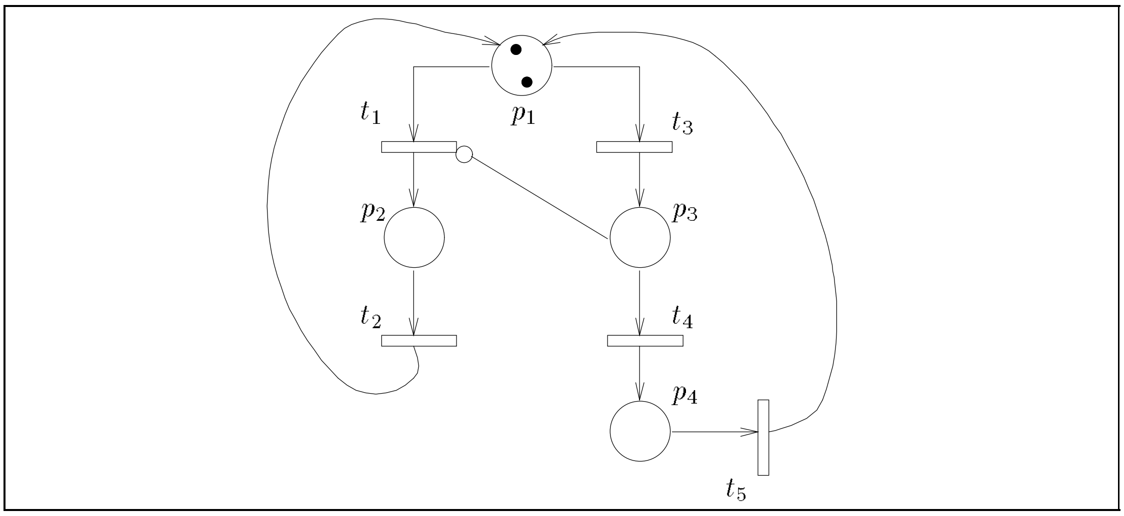


Figure 2.1: Example of a PN system with inhibitor arcs.

by $M[t\rangle M'$. A marking M'' is said to be *reachable* from M, if there exists a sequence σ of transition firings starting from M such that after firing σ we obtain the marking M'' ($M[\sigma\rangle M''$). The set of all reachable markings from the initial marking is called the *reachability set* (RS) of the system. From the RS we can obtain the *reachability graph* (RG) of a system. The RG is defined as a labelled directed multigraph, where its nodes are the markings in the RS and where arcs are labelled with transitions names. An arc from marking M to marking M' with label t indicates that $M[t\rangle M'$. Figure 2.2 presents the RG of the PN system introduced in Figure 2.1. The states are represented by 4 digit numbers where the i^{th} digit corresponds to the marking of place p_i.

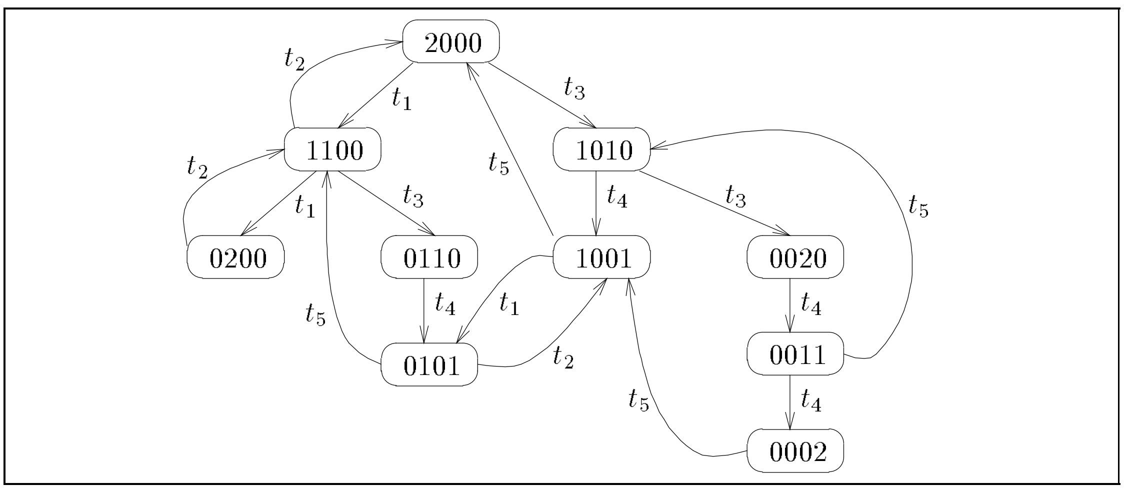


Figure 2.2: Reachability graph of the PN system introduced in Figure 2.1.

A PN system is said to contain a *deadlock* if one or more nodes in RG have no output arcs, i.e. if it is possible to reach one or more states where no transitions are enabled.

A transition t is said to be *live* if and only if for each marking M reachable from the initial marking M_0 there exists a marking M', reachable from M, such that $t \in E(M')$. A PN system is said to be *live* if all $t \in T$ are live in it [AMBC+95]. A transition that is not live is said to be *dead*. Given a dead transition it is possible to find a marking M reachable from M_0 such that t is not enabled in any of the markings reachable from M. A PN system is said to be *partially live* if it has no deadlock states but has some states with dead transitions.

A place p is said to be (k-)*bounded* in a PN system if and only if there exists a $k \in \mathbb{N}^+$ such that for all markings in RS it holds that $M(p) \leq k$. The PN system is said to be (k-)bounded if all its places are (k-)bounded. PN systems that are 1-bounded are said to be *safe*.

In Chapter 3 we will refer to several classes of Petri nets. Here we introduce some of these classes. A *Place/Transition net* (P/T net) [Rei86] is a basic PN in which places have a capacity. This capacity expresses the maximum number of tokens that each place can contain. Note that this is different from the bound of a place, a bound of a place could be less than its capacity. Arcs have weights which correspond to the concept of multiplicity in the basic PN formalism.

A *free choice net* is a PN in which it holds that:

$$\forall p \in P, \forall t \in T, \text{ if } W^-(p,t) + W^+(p,t) > 0 \text{ then } p^\bullet = \{t\} \text{ or } {}^\bullet t = \{p\}$$

i.e. if there is an arc between p and t then it holds that either t is the only output transition of p or that p is the only input place of t.

A *state machine*, or otherwise known as an *S-graph*, is a PN where every transition has exactly one input place and one output place:

$$\forall t \in T, \ |{}^\bullet t| = |t^\bullet| = 1$$

2.3 Coloured Petri nets

Coloured Petri nets (CP-nets) [Jen92] have been introduced to allow the modeller to make more manageable descriptions of large PN models, by folding equal subnets into each other. This folding consists of representing all instances of a process type by a single subnet and distinguishing the individual processes of this type by different colours. This allows the representation and study of more complex systems with symmetric characteristics. There are various definitions of CP-nets, which mainly differ in which elements are considered to be coloured (tokens, places, transitions and/or arcs) [Fin92, Buc92, Jen92] . Colours are mainly used to model two aspects [Buc92]: different behaviour patterns of entities in the system, and different parts of the system with similar or equivalent structure. Different entities (resources, data, etc.) are identified by different "coloured" tokens (or objects).

The *colour set* of a place is the set of *colours* that can be taken by the tokens in the place. Each token can take one colour, but there can be more than one token per colour. We require colour sets to be finite in order to ensure that a coloured net may be 'unfolded' into its equivalent PN. The set of possible colours of a transition (*colour set* of a transition) is determined by the colour sets of its input and output places and the arc functions between the transition and these places. Formally, a CP-net is a 7-tuple $\langle P, T, \Sigma, C, W^+, W^-, M_\circ \rangle$ where:

- P is a finite, non-empty set of places;
- T is a finite, non-empty set of transitions; $P \cap T = \emptyset$;
- Σ is a finite set of types called *colour sets*;
- C is a function from $P \cup T$ into colour sets, $C : P \cup T \rightarrow \Sigma$. We denote by $C(p)$ the colour set of a place p, and similarly $C(t)$ as the colour set of a transition t;
- W^-, W^+ are a set of functions $W^-(p,t), W^+(p,t) : C(t) \times C(p) \rightarrow \mathbb{N}^+$;
- M_0 is the initial marking.

The function M, over the set of places, defines a marking in the CP-net ($\forall p \in P,\ M(p) : C(p) \rightarrow \mathbb{N}^+$). $M(p)$ will determine, for each colour c in the colour set of p, the number of tokens of colour c in p.

The firing rule of CP-nets is defined as:

- A transition t is enabled for a marking M and a colour $c_t \in C(t)$, if and only if : $\forall p \in P, \forall c_p \in C(p) :\ M(p)(c_p) \geq W^-(p,t)(c_p, c_t)$;
- The firing of a transition t, with colour $c_t \in C(t)$, from a marking M produces a marking M' defined by:

  $$\forall p \in P, \forall c_p \in C(p) : M'(p)(c_p) = M(p)(c_p) - W^-(p,t)(c_p, c_t) + W^+(p,t)(c_p, c_t)$$

2.4 Well-formed coloured Petri nets

Well-formed coloured nets (WN)[CDFH90] are equivalent to CP-nets from an expressive power point of view [CDFH91]. Any CP-net can be translated into an equivalent WN, with the same underlying structure. In WN the expressions of the colour functions, and of composition of colour classes, are rewritten in a more explicit or parametric form, in terms of a set of basic constructs provided by the formalism.

In WN a token can be regarded as an instance of a data structure with a certain number of fields whose semantics depend on the place to which that token belongs. The definition of the "data type" associated with each place is called the *place colour domain*. The colours representing elements of the same type are grouped in a (basic) *colour class*. These classes form the sets of basic types. The colour domain of a place can be formed by the Cartesian product of basic colour classes. Elements within a colour class may be ordered. This ordering is assumed to be circular, so that a *successor* function applied to the last element returns the first one. When objects of the same class have different behaviour the class can be partitioned into *(static) subclasses*, each one representing a distinguished behaviour amongst the elements of the basic class.

Formally the family of colour classes of a WN is defined by:

$$C = \{C_1, \dots, C_h, C_{h+1}, \dots, C_n\}$$

where n is the number of basic classes, and it holds that $\forall i, j \in \{1, \dots, n\}$, $i \neq j \Rightarrow C_i \cap C_j = \emptyset$, and

- $\forall C_i,\ i \in \{1, \dots, h\}$, C_i is a non-ordered class,
- $\forall C_i,\ i \in \{h+1, \dots, n\}$, C_i is an ordered class, with $|C_i| > 1$. The k^{th} successor of an object $c \in C_i$ is denoted by $\oplus^k c$, and it holds that:

$$\forall c_j \in C_i,\ \bigcup_{k=1}^{|C_i|} \oplus^k c_j = C_i$$

The partition of a colour class into *static subclasses* is denoted by $C_i = \cup_{q=1}^{n_i} D_{i,q}$ where the $D_{i,q}$ are predefined and disjoint.

$$\forall\, i \in \{1\ \dots, n\},\ \forall\, q, r \in \{1, \dots n_i\},\ \ q \neq r \Rightarrow D_{i,q} \cap D_{i,r} = \emptyset$$

where n_i denotes the number of static subclasses of the colour class C_i.

Let $Ind = \{1, \ldots, n\}$ be the set of indexes of the colour classes and let $J = \bigotimes_{i=1}^{n} e_i$ be a n-tuple of integers belonging to $Bag(Ind)$[1]. $C_J = \bigotimes_{i=1}^{n}(C_i)^{e_i}$ is a *colour domain* formed by the Cartesian product of colour classes whose number of occurrences in C_J is given by the values e_i in J. If $J = \vec{0}$ then by definition $C_{\vec{0}} = \{\varepsilon\}$, where ε is called the *neutral colour*. A colour tuple $c_J \in C_J$ is denoted $\bigotimes_{i=1}^{n} \bigotimes_{j=1}^{e_i} c_i^j$.

The transitions in a WN can be considered as procedures with formal parameters. These parameters define the transition's colour domain; their declaration is part of the net description, and the type associated with each parameter must be a colour class. The colour domain of a transition t ($C(t)$) is constrained by the colour domains of its input, inhibiting and output places. A transition, whose formal parameters have been instantiated to actual values is called a *transition instance*, denoted $[t, c]$, where $c \in C(t)$ represents the assignment of actual values to the transition parameters. The enabling of a transition instance $[t, c]$ is determined by evaluating the transition's predicates and the arc expressions of all input and inhibiting places with respect to the assignment c. Many instances of the same transition could be concurrently enabled, given that they are considered as independent events.

Consider a place p and a transition t, such that $C(p) = C_J$ where $J = \bigotimes_{i=1}^{n} e_i$. An arc function between p and t will be a J-tuple, where an entry of the tuple will be a function from $C(t) \times C(p)$ to $\mathbb{N}^+$, formed by the linear combination of three basic functions, namely:

- The *projection* or *identity* function X_i^k. Placed in the j^{th} entry of the arc function between a place p and a transition t, it will return 1 $\forall (c_p, c_t) \in C(p) \times C(t)$ such that the j^{th} entry of c_p equals the k^{th} entry of c_t.

- The *successor* function $\oplus^u X_i^k$ to be used on ordered sets. It represents the u successor of the object selected by X_i^k, which means that it only makes sense when it is applied over a transition which also has a function X_i^k on one of its arcs. Placed in the j^{th} entry of the arc function between a place p and a transition t, it will return 1 $\forall (c_p, c_t) \in C(p) \times C(t)$ such that the j^{th} entry of c_p equals the the u successor of the k^{th} entry of c_t.

- The *diffusion* or *synchronisation* function $S_{i,q}$. When associated with an input arc it represents the synchronisation of all the elements of the static

[1]A *multiset* is a set that can contain several occurrences of the same element. $Bag(A)$ denotes the set of finite multisets over a set A.

subclass $D_{i,q}$. If it is on an output arc, it represents the broadcast of all the objects of the static subclass $D_{i,q}$. Placed in the j^{th} entry of the arc function between a place p and a transition t, it will return 1 $\forall c_p \in C(p)$ such that the j^{th} entry of c_p is a colour in $D_{i,q}$.

To restrict the firing possibilities of a transition, the idea of predicates guarding transitions is incorporated. These predicates influence the colour domain of the transition. *Standard predicates* can be formed by logical combinations of the three basic predicates, namely:

- $X_i^j \in D_{i,q}$; returns TRUE if the colour of the j^{th} occurrence of C_i in the colour instantiation of the transition belongs to $D_{i,q}$.
- $X_i^j = X_i^k$; returns TRUE if the colour of the j^{th} occurrence of C_i in the colour instantiation of the transition equals the colour of the k^{th} occurrence.
- $X_i^j = \oplus^u X_i^k$; returns TRUE if the j^{th} occurrence of C_i in the colour instantiation of the transition equals the u successor of the colour of the k^{th} occurrence.

Figure 2.3 presents a WN model with guarded transitions. We assume that there is only one basic colour class C_1, therefore, we can represent X_1^j by X_j.

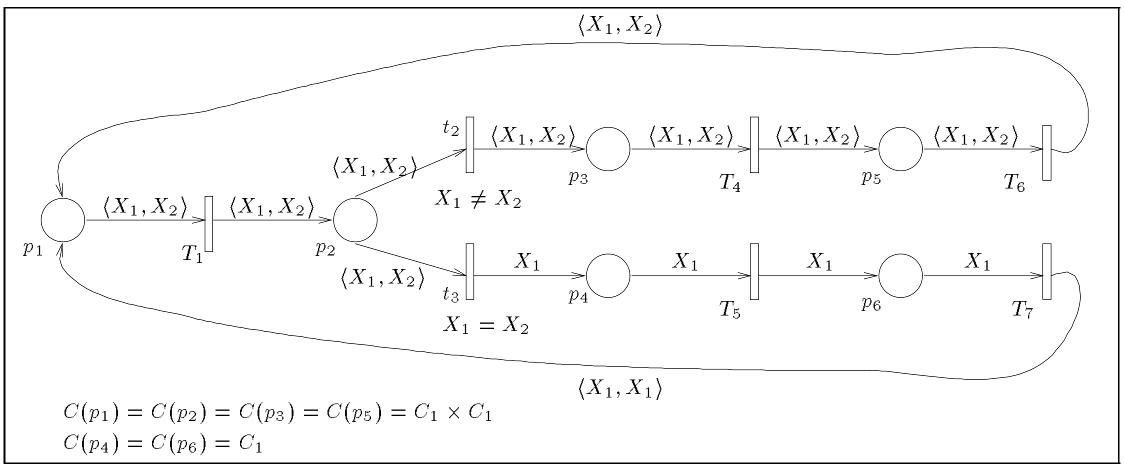


Figure 2.3: Example of a WN model.

The number of objects of a colour c_p selected by a transition instance $[t, c_t]$ from a place p is the Cartesian product of the evaluation of the entries of the arc function between p and t for the pair (c_p, c_t).

Let us now introduce the formal definition of WNs as given in [CDFH91].

Definition 2.1 *A Well-formed net system is a 9-tuple*

$$WN = \langle P, T, C, J, W^-, W^+, W^h, \Phi, M_0 \rangle$$

where:

- P *is a finite set of places;*
- T *is a finite set of transitions,* $P \cap T = \emptyset$*;*
- C *is the family of colour classes:* $C = \{C_1, ..., C_n\}$ *(we denote by* $I = \{1, ..., n\}$ *the ordered set of indexes) with* $C_i \cap C_j = \emptyset$ *for any* $C_i, C_j \in C$*; Any* $C_i \in C$ *may be partitioned into static subclasses,* $C_i = \bigcup_{q=1}^{n_i} D_{i,q}$*;*
- $J : P \cup T \to Bag(I)$*, where* $Bag(I)$ *is the multiset on* I*.* $C(\tau) = C_{J(\tau)}$ *denotes the colour domain of node* $\tau \in P \cup T$*;*
- $W^-, W^+, W^h : W^-(p,t), W^+(p,t), W^h(p,t) \in [C_{J(t)} \to Bag(C_{J(p)})]$ *the input, output, and inhibition functions arc expressions;*
- $\Phi(t) : C_{J(t)} \to \{\text{TRUE}, \mathit{FALSE}\}$ *is a standard predicate associated with a transition* t*. By default we will assume that* $\forall t \in T$*,* $\Phi(t) = $TRUE;
- $M_0 : M_0(p) \in Bag(C(p))$ is the initial marking.

The syntactic definition of WNs leads to new algorithms for the construction of the state space of a PN system, based on the concept of *symbolic marking* [CDFH91]. A symbolic marking, denoted $\mathcal{M}$, represents an equivalence class of an equivalence relation defined over the state space of the WN system. This equivalence relation is based on the idea of symmetry of objects of the basic colour classes. The *symbolic reachability graph* (SRG) of a WN is defined over symbolic markings. It consists of a symbolic representation of all possible states of a system and the possibility of transition from one to another. The symbolic markings together with a *symbolic firing rule* allow the construction of the SRG.

The use of symbolic markings introduces the concept of *dynamic subclasses*, which represent sets of objects that are not identified individually but are known to permute with each other in any firing instance to produce markings that belong to the same equivalence class. A dynamic subclass is characterised by its cardinality, and by the static subclass to which the represented objects belong. The concept of dynamic subclass affects both the symbolic marking representation and the symbolic firing. Using dynamic subclasses instead of variables in the marking representation allows a much more compact description of the marking itself. This will be presented in greater detail in Chapter 6.

We have introduced the concepts of PN theory that will form the working platform of this dissertation. In each individual chapter other concepts will be introduced, related to the area covered by the chapter. In the following chapter we present some of the existing work in the area of compositionality and PNs.

Chapter 3

Compositionality in Petri nets

3.1 Introduction

This chapter corresponds to the second part of the background material of this dissertation. Section 3.2 reviews the existing work in the area of compositionality in Petri nets, describing the different approaches and motivations. Based on this study in Section 3.3 we present our criteria for the definition of an adequate method for the construction and analysis of PN models of parallel and distributed systems, discussing the advantages and disadvantages of the works studied, and determining what can be learned from each of them.

Although PNs are widely recognised as a powerful formalism for the modelling of concurrent and parallel systems, the lack of built-in compositional constructs makes their use inappropriate or not suitable for the modelling of realistic (non-toy) systems. The incorporation of compositionality into PNs has been the topic of much research, dating from the early seventies. These studies have been motivated and approached from several angles, such as:

- the ability to build a system in a compositional or modular way, allowing at the same time the deduction and/or preservation of the properties of the components from the properties its sub-components;
- the representation and analysis of resource/program (hardware/software) systems;
- the refinement or synthesis of models;
- and the compositional analysis of properties of the model.

The methods proposed to incorporate compositional features into PN vary according to the underlying Petri net class employed (e.g. P/T nets, CP-nets, etc.), the characteristics of the (basic) components and the set of composition operations defined. These features are strongly influenced by the motivation and objectives for incorporating compositionality into PNs. Let us now study some of the existing work related to compositionality in PNs, according to their objectives and approaches.

3.2 Motivations for compositionality

3.2.1 Preservation of behavioural properties

The behavioural properties of PN systems are, in general, very hard to analyse due to the size of the net and of its state space. Working with restricted classes of nets can make this task easier [BDC92]. This has been the approach taken by many studies that define PN components and composition operations in such a way that the preservation of properties, such as liveness and boundedness, can be guaranteed when composing the components. There are two basic and complementary approaches to the solution of this problem [ES90]: (modular) composition of subnets, or transformation of subnets. Here we review some of the existing works in each of these areas.

Modular approach

Hack's work on State Machine Decomposable (SMD) nets [Hac72] was one of the first approaches to the incorporation of compositional notions into Petri nets. It emphasised decomposition of a given net in order to prove behavioural properties, identifying the relation between structural and behavioural properties of the net. He worked on liveness and boundedness properties of free choice nets. Although the approach is decompositional, by identifying how the system can be decomposed into subnets that are known to have certain properties and by knowing how these subnet are combined, it is possible to define the components and composition operations that preserve the properties of the system. This idea has been the basis of several other works based on the characterisation of the structures which must be avoided in order to preserve certain properties [ES90, ES91b, ES91a, Sou93, Sou91b, SM90, Sou91a, BG96], resulting in the definition of restrictions over the structure of the operands, of the operations or both. In [ES90, ES91b, ES91a], Esparza and Silva consider the composition of free choice net components by means of synchronisations that are resolved by the fusion of places and/or transitions. The set of fused elements is termed the synchronisation or communication *medium*. The types of composition are characterised according to their preservation of the liveness and boundedness properties.

Similarly, Souissi, in [Sou91b], and with Memmi, in [SM90], studies the definition of restrictions over the operations. The communication media are specific types of subnets, shared places [Sou91b]; in [SM90] the study also considers other, more complex, types of subnets (sequential processes and well-formed[1] net blocks). In [Sou91a] and [BG96], they work over FIFO nets (P/T nets in which some places behave as FIFO queues). The FIFO nets used in [BG96] are coloured FIFO nets. In [Sou91a], Souissi works on composition via transitions. Structurally sufficient conditions for liveness compositionality are proposed and expressed in terms of connectivity of non-merged queues with the merged transitions. In this way the author defines compatibility relations between the FIFO subnets being composed. In [BG96], Benalycherif and Girault relax the constraints imposed by Souissi in [Sou91a], by defining a new structural condition for liveness compositionality. This condition is based on the incorporation of a *non-constraining* relation between the subnets being composed. Intuitively, a component L is said to be non-constraining with respect to another R, if for every reachable marking M of the composed net, it holds that if there is a transition t of L enabled in the restriction M_L of M with respect to the places of L, t will remain enabled after the firing of a sequence of transitions of R that does not include transitions in the medium.

Sibertin's approach in [SB93] is somewhat different. The composition of Petri nets is presented within the framework of a client-server protocol. Composition is defined as an asynchronous communication, between a *server* net and a *client* net, employing a *use* function to associate services with demands. The composition is resolved at the level of the net by the fusion of places, and the resulting fused places are viewed as one way communication channels. The *use* function keeps track of places being fused, i.e. of the relation service-request, so that other clients can request the same service. Sibertin studies the possibility of composing while preserving the net's language and liveness properties. This analysis of preservation of properties is based on characteristics of the relation between the clients and the servers.

Transformation approach

An alternative approach, to the problem of analysing behavioural properties in a compositional manner, involves the notion of *net transformations*. The idea of this approach is that starting from a simple system, transformation rules permit the modification—abstraction and/or refinement—of the system, while at the

[1]In this context well-formed refers structural boundedness and liveness.

same time preserving its behavioural properties. This is the approach presented by Berthelot in [Ber86] and [Ber87]. The idea is to apply a set of transformations to the net, which are known to preserve a given property. Berthelot introduces a P/T net transformation, defined in behavioural terms, that is based on the addition of non-constraining subnets. This transformation is proposed to preserve liveness and/or boundedness of the initial net in the resulting net. Similarly, in [ES90, ES91b], Esparza and Silva propose a set of transformation techniques for free choice nets. Two types of transformation rules are defined: a *reduction* rule and a *synthesis* rule. These rules are based on the definition of *macroplaces*—which represent subnets—and *marking structurally implicit places* (MSIP)—places whose row in the incidence matrix of the subnet can be obtained from the linear combination of the rows of the other places. Reduction rules substitute subnets by macroplaces or eliminate MSIP. Synthesis rules substitute a macroplace by its corresponding subnet or add implicit places to a subnet.

Müller, in [M8̈5], introduces the concept of Constructible Petri nets. They are proposed to offer a synthesis approach to the design of systems. Unlike the two previous approaches, this work does not concentrate on the preservation of behavioural properties. Instead, the motivation of his work is based on the classical notion of transition refinement. The idea is that a transition which, at a certain level of abstraction, models an atomic event, can be substituted by a subnet. In this way a family of net classes can be built by the *refinement* of transitions starting from a distinguished set of *generating nets*. Müller studies the sufficient conditions for the preservation of liveness when refining a net; however, the refinement operations are not defined based on these conditions.

3.2.2 Compositional analysis

Related to the idea of composition preserving properties is the idea of modular or compositional analysis of PN systems. The works by Christensen and Petrucci [CP92], by Christensen and Hansen [CH93], and Chehaibar [Che91], are studies of modular analysis of coloured nets. The incorporation of colours into the underlying Petri net formalism implies that the composition operations must also consider the colours of places and transitions. Chehaibar introduces the concept of *reentrant nets*, intended to represent phase-protocols, which may correspond to the execution of a request by a server or to a procedure call. Reentrant nets are coloured nets with two distinguished disjoint sets of places, namely, *entry places*—which cannot have input transitions—and *final places*. These sets form the interface of the net. Since a phase may be iterated, the final places of a reentrant net may have outgoing arcs. Chehaibar works on both transformation of reentrant nets and modular construction of reentrant nets from smaller ones. The transformation operations are based on two equivalence notions, namely, interface equivalence and home space[2] equivalence. Composition can be done by sharing of interface places or composing reentrant nets in a ring. The resulting net of this last composition is not a reentrant net, but is deadlock free.

Christensen and Petrucci work on a more general class of coloured nets, termed Modular CP-nets, which consist of sets of formally related CP-nets, where each CP-net is called a module. Two types of relations between the modules are contemplated, a set of places sharing the same tokens, or transition sharing. The main objective for the introduction of Modular CP-nets is to provide a framework for the compositional analysis of coloured Petri net systems. Modular CP-nets are used for place invariant analysis in [CP92] and later, in [CH93], for modular state space analysis.

Recalde et al., in [RTS95], present Deterministically Synchronised Sequential Processes (DSSP), as a modular subclass of PNs suitable for the methodological construction of parallel and distributed systems that can be modelled as several agents cooperating by message passing. DSSP were originally defined, with a different name, in [Rei82]. Since then their definition has evolved by gradually relaxing the restrictions over how the components can communicate [SB88, Sou93]. A DSSP model represents the structure of the system modelled and it is built as a composition on sequential components. A DSSP is a P/T system formed by a group of live and safe state machines which communicate with each other through buffers. The analysis techniques employed take advantage of the structure of DSSP. Generalising the structure of DSSP, in [RTS96] Recalde et al. introduce $\{SC\}^*$ECS nets, as a PN class to model modular and hierarchical cooperating systems. Starting from $\{SC\}^*$ECSs built by composition of DSSP nets communicating through buffers, more complex $\{SC\}^*$ECS nets can be built by composing $\{SC\}^*$ECS nets in the same manner. The analysis techniques take advantage of the hierarchical and modular structure of the model to calculate the characteristics of the overall system, such as structural liveness and boundedness and the existence of home states.

[2]A set $HS \subseteq RS$ is said to form a (T-)*home space* if and only if for every $M \in RS$ there exists a firing sequence σ enabled in M such that when fired it reaches a marking in HS.

3.2.3 Construction of models

Compositionality is a fundamental concept in the methods proposed for the construction of (parallel) hardware-software models [Fer92], [BDC93] and [DF96]. These studies propose the construction of resource-application models by using a resource or hardware model, a process or application model and an interface or mapping to combine the previous two. The basic idea of these studies consists of clearly separating the model of the application from the pattern of resource usage.

In [BDC93], Botti et al. introduce Process/Resource boxes (P/R Boxes). P/R Boxes are based on Process Boxes (PB), defined in [HHB92]. A software model is built by composing PB, while the hardware model is obtained by composing Resource Boxes (RB). The usage of a resource in a certain process is represented by the definition of a mapping from resources to processors. RB are labelled nets, where places represent states of the resources and transitions represent the basic services offered by the resource to its environment. The P/R-Box methodology attempts to provide PN with modularity and compositionality, enhancing their effectiveness and (re)usability in modelling complex systems.

OBJSA nets [BBCDC95] emphasise the properties of *superposed automata* (SA) [DCDMPS81] with respect to the possibility of building system models by the composition of sequential—non-deterministic—components. The aim of the authors (Battiston et al.) is to overcome the difficulty of structuring nets in such a way that they reflect the structure of the system modelled. An SA net results from the combination, through transition superposition, of a set of state machine components, each of them representing a sequential component in the system modelled.

3.2.4 Compositional semantics for Petri nets

The works on the definition of compositional net semantics (e.g. [Val93],[HHB92], [BDH92] and [BB93]) also represent a modular approach to the construction and analysis of PN systems. In [Val93], Valmari proposes a compositional method based on the CSP [Hoa85] semantics. The paper discusses the theoretical and technical prerequisites for compositional state space generation methods. This leads to the definition of a compositional semantic for nets—based on labelled P/T nets—suitable for the compositional state space generation. In [BDH92], Best proposes the Box Calculus, based on CCS [Mil89]. The Box Calculus is intended to serve as a bridge between PN theory and concurrent programming applications, offering an algebraic structure for PNs. The Box calculus has been the basis of many other research works. In [BFF+95], Best et al. develop a high level net model, called M-nets, in terms of which the semantics of the concurrent language $B(PN)^2$ (a Basic Petri Net Programming Notation) can be formulated compositionally. In this approach transformation and composition are combined.

In [Kot78] Kotov describes a semantics of control structures and operations, in terms of structured nets. From *atomic* nets (corresponding to program statements), more complex nets are formed by applying net operations.

Broy and Streicher, in [BS92], propose a functional semantics for modelling high level PNs. They offer a model that combines PNs, as models for the representation of distributed systems, with functional models of parallel systems, providing, at the same time, a framework for modular construction of PN models and a modular semantic.

3.3 Criteria for a compositional method for the construction and analysis of PN models of parallel and distributed systems

It is broadly agreed that compositionality is a very desirable property for the construction and analysis of models of parallel and distributed systems. We have reviewed different studies on the incorporation of compositionality into PN. In this section we present the criteria that we will take into account for the definition of a compositional method for the construction and analysis of a performance-oriented PN systems modelling parallel and distributed systems. Based on these criteria we discuss the suitability of the methods studied for the construction and analysis of PN models of parallel and distributed systems. In this way it is possible to determine what can be learned from each method.

Criteria for the definition of an appropriate method for the construction and analysis of performance-oriented PN systems

Model-system relation : the model of the system and the way it is built should reflect the system's structure: its sub-systems or processes and the relations between them.

The definition of a basic component : we should be able to construct the model in a regular and progressive manner, starting from components that have common characteristics. In this way the modeller can identify what will constitute a basic component and can build the model with the use of composition operations.

The PN formalism : the formalism should support the representation of the different types of resources, different patterns of behaviour and symmetric structures, common in parallel and distributed systems.

Re-usability : being able to re-use (sub-)models and the information about their behaviour saves time and effort in the construction of more complex models and in the deduction of their properties.

Let us now analyse the methods reviewed according to each of these criteria.

3.3.1 Model-system relation

As stated earlier, the underlying model of parallel and distributed systems is that of loosely coupled components that communicate or synchronise [KMF90]. A compositional method for the construction of parallel and distributed systems should reflect the structure of the system modelled. This means that the components should represent identifiable functions or subsystems, and the composition operations should reflect the relation between these functions or subsystems [BDF95, DF96, Fer92].

The methods that present a compositional net semantics satisfy this criteria by definition. However, in general, the relation between the subnets composed is one of control flow. We want to be able to also represent data flow between components.

In the methods that are based on the separation of the hardware or resource and application models, for example Donatelli and Franceschinis' [DF96], Botti et al. [BDF95, BDC93] and Ferscha [Fer92], and on DSSP and $\{SC\}^*$ECS nets [RTS95, RTS96], there is a clear correspondence between components and system parts, and between composition operations or medium and the relationship between the system parts. This idea forms the basis of these methodologies. However, in the methods based on P/R boxes [DF96, BDF95, BDC93], the relations reflected are the resource requirements of a parallel and distributed software application over a certain hardware or resource configuration; the relation between processes or between resource nets is limited to synchronisation of activities. In the work on DSSP and $\{SC\}^*$ECS nets, the communication between system parts is always asynchronous, representing cooperating processes that communicate by message passing. Similarly, in Sibertin's client-server protocol [SB93] the system parts correspond to clients and servers that communicate in an asynchronous manner, requesting services or providing them. In general, in these methods the communication between the system's parts is limited to a certain type. This is not the case of Ferscha's work on PRM-nets, where processes can communicate in a synchronous and asynchronous manner by the use of several composition operations. However, the composition of components is mainly used for the aggregation of resource requirements.

In other works such as Souissi and Memmi's [SM90, Sou91b], Hermanns' et al. [HHMR97], Valmari's [Val93] or Benalycherif and Girault's [BG96], where the central idea is the preservation of properties, there is no direct correspondence between system parts and their relations, and model components and composition operations, respectively. Furthermore, given the definitions and restrictions over the composition medium it is not clear how to define composition operations that preserve the properties, while at the same time reflect the structure of the system modelled. The same situation arises in Donatelli's work on the compositional construction of models by transition fusion, with the aim of improving the quantitative analysis of the model [Don93, Don94], and in the Modular nets introduced by Christensen and Petrucci in [CH93], where place and transition fusions have no direct correspondence in the system modelled.

3.3.2 The definition of a basic component

The definition of a *basic component*, with a pre-defined net structure, offers regularity to the compositional method and can be used to deduce the properties of the model built. If all components can be built, starting from basic components, by applying successive composition operations, then the characteristics of a component can be deduced from the characteristics of the basic components and knowledge of the composition operations employed. As we have seen, many of the compositional methods studied follow this approach. The methods proposing compositional net semantics, such as [BDH92, Val93, HHB92], amongst others, use the idea of basic components. However, the concept of basic Boxes in the Box Calculus [BDH92], is introduced by examples of the simplest types of boxes, and not as regular building blocks.

For the construction of Regular nets [Kot78], Kotov introduces the concept of an atomic (regular) net. Regular nets are built from atomic nets by applying composition. Similarly, the methods proposed by Esparza and Silva [ES90, ES91b] consider the modelling processes as starting from an *atomic net*. By applying synthesis rules, it is possible to progressively build more complex systems.

Although the works on medium composition do not explicitly define any type of basic net for the composition, Benalycherif's work [BG96], on non-constraining relations, uses the idea of decomposition of the nets until reaching recognisable types of subnets. These types of subnets can be viewed as the basic components, all of which have the same structure.

In other work, such as the P/R Boxes based approaches [DF96, BDF95, BDC93] and the client-server protocol [SB93], although the structure of the system is reflected in the model, they do not offer a compositional method for the construction of the components representing the subsystems. The modelling of subsystems is done using conventional flat net models and their characteristics must be obtained by applying the traditional analysis methods. Donatelli and Franceschinis [DF96] allow the composition of components at the resource level, but this is limited to synchronisation operations. Similarly, in the case of DSSP [RTS95] and $\{SC\}^*$ECS [RTS96] the models representing the cooperating processes are built in the conventional way. However, the structure of the sub-components is restricted to state machines, which are known to have certain properties.

Admittedly, it could be argued that in these methods the concept of a basic component is much more general, not necessarily all having the same structure. For example, in the client-server protocol a client or a service net represent basic components, in DSSP and $\{SC\}^*$ECS the state machines represent basic components and in P/R Boxes the P and R boxes represent basic components.

The initial marking of components The state of a system is modelled in a Petri net by the distribution of tokens in the places of the net. Viewing the system as composed of subsystems, the state of the system should be composed or deduced from the state of the subsystems. In the same way, in a compositional PN system it should be possible to deduce the state of a component from the composition of the states of its subcomponents. Although not explicitly mentioned, in most of the approaches that consider place fusion composition, the operations are defined by characterising the resulting net, neglecting the definition of the (initial) marking of the resulting component with respect to the initial marking its subcomponents. For example, it is said that the initial state of a Petri Box is formed by assigning a marking of 1 to all initial or entry places. However, if this is done for basic Petri Boxes how is the initial marking of the composition of two Petri boxes defined with respect to the initial markings of the operands? This is not made clear and it is assumed that the initial marking is defined for a Petri Box representing a complete system. Christensen and Petrucci [CH93, CP95], deal with the problem of initial marking by enforcing that only places that have the same initial marking can be fused. This initial marking is then considered as common for the places in the fusion set. This is implicitly the approach used in [SM90], but the marking is not part of the composition operations.

3.3.3 The Petri net formalism

Colours can be used to model the different behaviour patterns of entities in the system, and different parts of the system with similar or equivalent structures. Symmetric processes tend to be common in parallel and distributed systems, making the presence of colour notions a useful feature. Colouring can refer to places, transitions or tokens. With the use of coloured tokens it is possible to model different elements (resources, data, etc.) that are transferred between the processes of the system. In [BDF95], Botti et al. combine (stochastic) WNs with P/R boxes, offering a compositional model that exploits the symmetries of system. However, the validation of properties of the system is performed once the system is built, based on the properties of the unfolded (stochatic) PN system.

To model parallel and distributed systems it is necessary to offer composition operations for the asynchronous and synchronous communication of components. There is a consensus that asynchronous communication is obtained by fusion of places between components. However, there are two basic methods for the modelling of synchronous communication: by transition fusion or by using labelled transitions. Labelled transitions are, in some cases, used as intermediate steps to identify the transitions to be fused [DF96, BDC93, BDF95]. The approaches that directly use transition fusion rely on the modeller's knowledge of which transitions are to be fused. This is overcome by the incorporation of labelled transitions, since the labels define the sets of transitions to be fused. In order to re-use a component with labelled transitions it might be necessary to rename its transitions. This, however, would not be necessary when working with non-labelled transitions.

3.3.4 Re-usability of components

Compositionality and modularity should imply re-usability. A complete system may constitute a sub-system within another system or a sub-system may form part of several systems. To be able to re-use a component its functionality must be known. Although re-usability is not explicitly mentioned in some of the compositional methods studied, all the methods that associate a functionality with the components implicitly allow the re-use of components. The definition of a compositional semantics for PNs implies that the the semantics of a component can be obtained from the semantics of its subcomponents. In this case the set of reachable states, and thus the behaviour, of each component cannot be influenced by the environment. However, if we consider a model where the states of a component can be influenced by its environment, then it is not possible to guarantee temporal or functional behaviour. Nevertheless, it would be desirable to be able to use information about the properties and behaviour of a component as a basis for the deduction of the properties and behaviour of the resulting compounds. Rather than preservation of properties, this approach implies the construction or deduction of the properties of a component from the properties of its parts.

Chapter 4

Operations for the compositional construction of Petri net systems

4.1 Introduction

In the previous chapter we reviewed some of the existing works in the area of compositionality in PNs, analysing them according to a set of criteria for the definition of a method for the compositional construction and analysis of PN models of parallel and distributed systems. Based on these criteria, in this chapter we define a set of operations for the compositional construction of PN models of parallel and distributed systems.

The set of composition operations defined is based on the operators of Process Algebras (PA). By mimicking the operators of PA, we benefit from the compositional nature of that formalism. The definition of composition operations has taken into account the peculiarities and characteristics of the PN formalism, such as synchronisation, state evolution and token flow. The composition operations are defined over Well-formed coloured nets (WN). The WN systems obtained by applying the composition operations are termed *composable WN* (*cWN*) systems.

As stated in [SB93], a modular approach improves the structure of models and favours the design of re-usable nets only if it is based upon a high level protocol which defines the communication between nets in terms of their functional abilities, and not solely in terms of their graph structure. In this sense the compositional operations defined represent different kinds of communication or scheduling between components.

The rest of this chapter is structured in the following manner. In Section 4.2 we describe the factors that were considered when defining the compositional method. First we review the characteristics of some of the existing process algebras. We then justify the selection of the WN formalism as the basis for the compositional method defined. The section concludes by determining what will be the nature of the composition operations to be defined, with respect to the type of information that would be transfered between components, i.e. data and/or control flow. In Section 4.3 we present our compositional method for the construction of *cWN* systems. We define the *basic component*, based on the idea of a basic element in stochastic PA as defined by Ribaudo [Rib95a], going on to define the set of composition operations of the compositional method proposed. For each operation we describe its abstract functionality (i.e. the relation between the components), its syntax and the characteristics of the component resulting from its application. We conclude this section by presenting an example to illustrate how the operations defined can be applied to build a *cWN* system. The example presented is the well known "dining philosophers" problem. The type of composition operation and the order in which composition operations are applied determine the characteristics of the model obtained. In Section 4.4 we discuss the factors that must be taken into account to construct a *cWN* system, offering a set of informal guidelines for the modeller. At the end of this section we present another, slightly more complex, example to which we apply the guidelines proposed. To conclude this chapter, in Section 4.5 we present the conclusions of the chapter and proposals for future work.

4.2 Defining a compositional method for the construction of Petri net systems

To offer a framework for the compositional construction of PN systems it is necessary to determine what will constitute a component and the ways in which components can communicate and/or synchronise. In order to take advantage of the compositional nature of Process Algebras (PA), our approach is to mimic the operators of PA, while at the same time considering the characteristics and peculiarities of the PN formalism. The intension is not to create a net semantics for process algebras or concurrent program languages as in [BDH92, HHB92, Kot78, Rib95b]. Here the idea is to use the general meaning of the process algebra constructs to offer compositional primitives for the construction of PN systems, where meaning refers to the functional interpretation of the operators. In [Rib95a] it is suggested that perhaps the best way to incorporate compositional primitives in PNs would be based on the operations of PA. A straightforward translation, however, would not take into account the characteristics of the PN formalism, such as synchronisation, state evolution and token flow.

Let us now briefly revise the PA and SPA formalisms.

4.2.1 Process Algebra

PA are abstract languages for the specification and behavioural analysis of concurrent systems. In PA a system is characterised by its active components and the interactions between them. Each component may be atomic or may itself be composed of components. Complex systems are built starting from the basic building blocks and applying the constructors of the algebra. Examples of PA include the Calculus of Communicating Systems (CCS)[Mil89], Communicating Sequential Processes (CSP)[Hoa85], the Algebra of Communicating Processes [BK85] and LOTOS [BB89]. The grammar of the language defines the ways in which the behaviour of a component may be built up from activities or an interaction of components. Operators are available for composition as well as mechanisms for abstraction which disregard internal details. Qualitative properties of a model are investigated by inspecting the transition diagram associated with the model.

The operators defined in the various PA differ, but in general focus on synchronisation, sharing, scheduling and communication between components. In CCS, the combinators of the language make it possible to construct an agent (the name given to an active component) which:

- has a designated first action (prefix),
- can behave as either of two agents (choice),
- can either "execute" two agents concurrently and independently or can make them synchronise over common actions (parallel composition),
- cannot perform any action of a predefined set of actions (restriction),
- behaves as another agent but with actions relabelled by a relabelling function (renaming),
- can have activities that are not visible externally (hiding),or
- has a recursive behaviour (recursion).

CSP additionally supports an external choice which allows the intervention of the environment.

4.2.2 Choosing the Petri net formalism

By incorporating the notion of colours, the possibility of associating information with tokens and of parameterising transition firing, makes it possible to represent in a very concise manner systems that would have required huge uncoloured nets to be described.

As studied in Chapter 3, WNs allow a natural representation of complex distributed systems, having the same expressive power as the unconstrained coloured net formalism. They permit the identification of model symmetries by means of the *symbolic reachability graph*, reducing the state space representation of the model. For these reasons we chose to work with WNs rather than with Coloured PNs [Jen92] as the basic modelling formalism.

4.2.3 Communication between components

Our initial work, on defining the composition operations for the construction of PN systems, was aimed at reproducing, as closely as possible, the semantics of the operators of PAs. This meant viewing the composition operations as control flow operators. As we have seen in other methods following this approach [Che91, HHB92] it is necessary to define basic components that have an initial state and final states. The final states are needed in other to be able to identify when a component has finished and therefore the one following can start. In PA the evolution of the model is determined by the semantics of the composition operators used to form the components, whereas in PNs the evolution of the model is determined by the enabling and firing rules. The enforcement of a final state for components, such that the transitions of subsequent components do not become enabled while the first component is still "active" turned out to be a cumbersome and difficult problem. It would either be necessary to have prior knowledge of the possible final states in order to be able to fix the multiplicity of the arc functions in the other component or to assign lower priority to the transitions of the other component. However, we must keep in mind that the set of enabled transitions is defined over the overall system. Therefore subsequent components would have to have transitions with lower priorities than those in the first component and so on. In most of the existing methods that define final states this problem is avoided by using safe nets. In other methods like reentrant nets [Che91], the problem is avoided by defining as the final state of a reentrant net a state where all final places are marked. Reentrant nets allow the first component to "execute" simultaneously with its successor but over different "iterations".

Our subsequent work has focussed on the implementation of composition operations that allow the asynchronous communication of components that can "execute" in parallel, and where the behaviour of one component can affect the behaviour of another component already "active". The tokens transfered between components can act both as control flow elements and hardware or software resources (memories, buses, data, etc.).

4.3 Compositional WN (cWN) systems

Before defining a set of composition operations for the construction of PN models, it is necessary to determine what will represent a component, and how it can communicate or synchronise with others. We need to define the medium by which communication can be made, i.e. places, arcs and/or transitions, and how much information is required about a component in order to be able communicate with it, i.e. define the interface of the component.

The medium selected will depend on the way we want the components to communicate, i.e. if two components need to communicate in an asynchronous manner, then the communication should be made using places; if the components need to synchronise, then the best mechanism is by transition fusion. The connection of nets by arcs corresponds to communication by "message sending" (arc from a transition to a place) and "message taking" (arc from a place to a transition) as discussed in [SB93]. We will not consider arc communication in this work, as it would mean modifying the set of input or output places of the transitions of a component. Another alternative is the definition of composition subnets, where the composition operation defines a subnet that connects the participating components. This approach is used in Donatelli and Franceschinis in [DF96] and by Ferscha in [Fer92] to compose resource nets with program or application nets, and by Souissi and Memmi in [SM90] where subnets can be composed by a sequential process or by what is termed a *well-formed block*.

In our approach the communication medium can be formed by places, transitions or predefined subnets. It is not desirable to have to know the whole net structure of a component in order to allow it to communicate with another. Certain parts of the net structure of a component should be visible only to the component itself. A component can be defined as a black-box with an associated interface by which it can communicate with other components.

We will define the interface of a component as:

- a set of *entry places* (ES), by which a component receives information from other components,

- a set of *final places* (FS), from which the component transfers information to others and
- a set of *synchronisable transitions* (ST), by which the component can synchronise with other components.

4.3.1 The basic component

Based on the idea of an action in PA and its representation in PN—as introduced by Ribaudo in [Rib95a]—as the basic construction component, we introduce the concept of a basic WN (*bWN*) component. From a *bWN* we can build more complex components by using composition operations.

Intuitively, a *bWN* is a WN in which there is only one transition (t). t has a set of input places (In) and a set of output places (O), either of which could be empty. A *bWN* will represent a function with its inputs and outputs. These inputs can represent data, resources needed to execute the function or control assignment. The input places cannot intersect with the output places. The set of entry places ES of a *bWN* will correspond to the set of input places ($ES = In$), and the set of final places will correspond to the set of output places ($FS = O$). Given that the input and output places cannot intersect, $ES \cap FS = \emptyset$ (see Figure 4.1).

This definition is similar to that of *atomic* nets in [Kot78] and [ES90, ES91b]. However, in these papers atomic nets are defined as nets with one entry place, one final net and one transition.

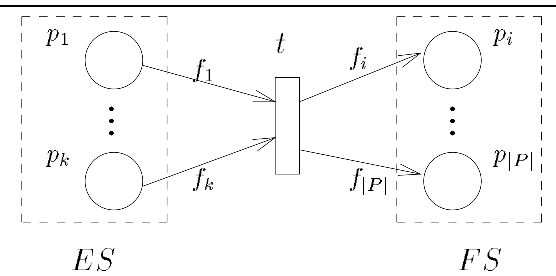


Figure 4.1: Basic WN component.

In order to keep the compositional model as simple as possible, we have restricted the type of arcs allowed in *bWN* to input and output arcs, i.e. inhibitor arcs are not allowed.

A *bWN* will have an initial parametric marking (MP) [AMBC+95], which represents a family of PNs with the same structure. The transition t of a *bWN* can be declared as synchronisable or not. Formally, a *bWN* is defined in the following way:

Definition 4.1 *A* **basic WN** (*bWN*) *is a* WN

$$bWN = \langle P, ES, FS, T, ST, C, J, W^-, W^+, \Phi, MP \rangle$$

where:

- $T = \{t\}$ *is the transition of the bWN;*
- $ST = \{t\}$ *if* t *is* synchronisable *otherwise* $ST = \emptyset$*;*
- $P = {}^\bullet t \cup t^\bullet$*; where* ${}^\bullet t$ *are the input places of transition* t *and* $t^\bullet$ *its output places;*
- $ES = {}^\bullet t$*;*
- $FS = t^\bullet$*;*
- C *is the family of basic colour classes from which colour domains are defined* ($C = \{C_1, ..., C_n\}$)*.*
- *We denote by* $I = 1, ..., n$ *the ordered set of indices with* $C_i \cap C_j = \emptyset$ *for any* $C_i, C_j \in C$*; the function* $J : P \cup T \to Bag(I)$*, defines colour domains;*
- $\forall p \in P,\ W^-(p,t), W^+(p,t) : C(t) \to Bag(C(p))$*, are the set of input and output arc functions of* t*, respectively;*
- $\Phi(t) : C_{J(t)} \to \{TRUE, FALSE\}$ *is a standard predicate associated with the transition* t*. By default* $\Phi(t) = TRUE$*;*
- MP *is the initial parametric marking of the places in* P*;*
  $\forall p \in FS:\ MP(p) = 0$*.*

We can define a transition predicate for t via the function Φ, which will be evaluated in every instantiation of the transition. Output places are considered to be the output of the function represented by the *bWN*, therefore, they all have initial marking zero.

Unlike the basic element in PA, a basic WN can execute the same action a finite number of times, i.e. transition t can fire a finite number of times, depending on its initial marking and the multiplicity of its input arcs. Although a *bWN* has one or more final states[1], this concept is not used to define the composition operations.

Having defined a *bWN* the next step in defining a compositional framework is the definition of the composition operations, to obtain what we will term *composable WNs*, taking *bWNs* as the fundamental elements.

Definition 4.2 *A* composable WN *(cWN) is either a bWN or a composition of cWNs.*

$$cWN ::= bWN \mid cWN * cWN \mid \circ cWN$$

where $*$ represents any binary composition operation and $\circ$ any unary operation.

4.3.2 Compositional operations

Let us now propose the set of composition operations that will form the basis of the compositional method for the construction of PN systems presented in this dissertation. Each operation represents a way in which the sub-systems can communicate. The composition operations are reflected at the level of the components by fusion of places, fusion of transition or sharing of a subnet.

Working with WNs adds another level of difficulty in the definition of composition operations. It is necessary to determine which places, according to the colour domain, can be fused and which cannot. We will only allow the fusion of places with equal colour domain.

In the examples presented to illustrate the operations, the elements composed are *bWN*; however, the definitions are given more generally in terms of *cWN*.

4.3.2.1 Sequential composition

To model the case when the output, or part of the output, of a function, or sub-system, forms part of the input of another sub-system, we include a sequential composition operation. The motivation of this operation is the Prefix operation in PA. The differences are that, in this case, the prefix of the sub-system or component is not necessarily a *bWN* (corresponding to the basic element) and that the participating components can "execute" in parallel. Unlike, other proposals of sequential composition, for example the join operation in [Kot78], not all output places of one component or all input places of the other have to participate in the composition. Our definition is similar to the sequential composition in reentrant nets [Che91], where the set of entry places (termed initial places in reentrant nets)

[1] A *bWN* can have several final states depending on the transition's instances that fire.

and the set of final places participating in the composition can be subsets of the entry set and of the final set, respectively, of the components being composed.

From the point of view of the interfaces of the participating components, sequential composition is resolved by the fusion of final places—of the component from which the information is extracted—with entry places—of the component into which the information goes.

Consider the sequential composition of two cWN, namely L and R, to form a cWN $\mathcal{N}$ (see Figure 4.2). The set of places participating in the operation is determined by the definition of a function $\Gamma : (FS'_L \subseteq FS_L) \to ES_R$, that associates final places of L with entry places of R. For a place $p_l \in FS'_L$ with $\Gamma(p_l) = p_r$ where $p_r \in ES_R$, it must hold that $C(p_l) = C(p_r)$ ($C(p_i)$ is the colour domain of p_i). This will mean that the type of information in p_l is the same as p_r. We will denote by ES'_R the set of places in ES_R forming the image of the function Γ. Places in FS'_L that are output places of a common transition cannot have the same image in ES'_R, otherwise we would be creating parallel arcs. As seen is Chapter 2, parallel arcs are represented in PNs by the multiplicity of the arc function. To maintain this representation we could either forbid the creation of parallel arcs or add the functions of the parallel arcs to create a unique arc representing a group of parallel arcs. In the compositions operations defined we have adopted the first solution to maintain the simplicity of the model and the idea that different output (entry) places of a transition represent different outputs (inputs) of the function represented by the transition.

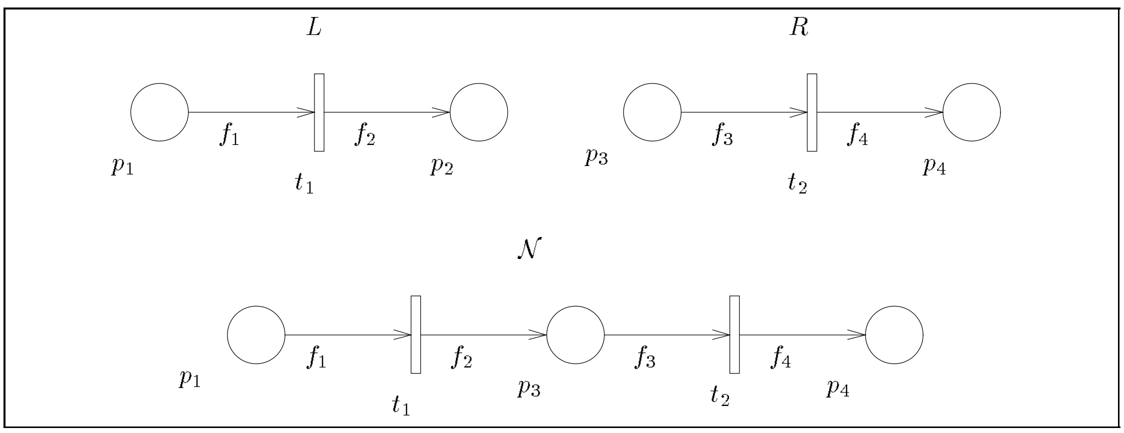


Figure 4.2: Sequential composition of WN components.

All places that are in the pre-image of a place $p_e \in ES'$ will be fused to p_e. However, the resulting fusion places will not be visible in the final component, i.e. they will not belong to either the entry set or the final set of the resulting component (see Figure 4.3).

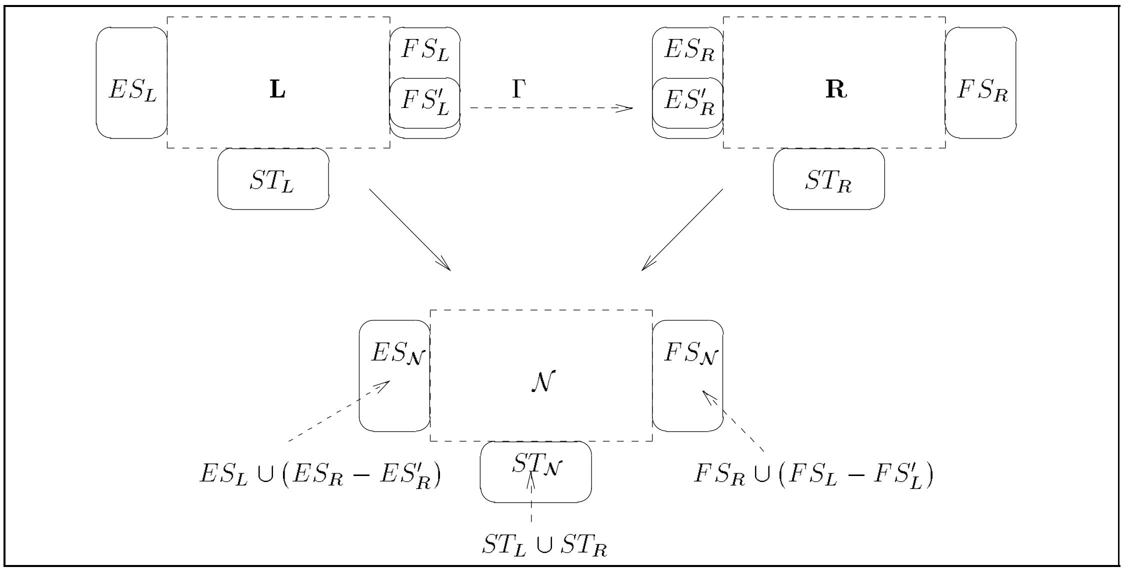


Figure 4.3: Construction of the interface of a *cWN* resulting from sequential composition.

Formally the *cWN* $\mathcal{N}$ resulting from the sequential composition of two *cWN*s L and R, denoted $(L\,;R)$, is a *cWN* defined as:

$$\mathcal{N} = \langle P_{\mathcal{N}}, ES_{\mathcal{N}}, FS_{\mathcal{N}}, T_{\mathcal{N}}, ST_{\mathcal{N}}, C_{\mathcal{N}}, J_{\mathcal{N}}, W^-_{\mathcal{N}}, W^+_{\mathcal{N}}, \Phi_{\mathcal{N}}, MP_{\mathcal{N}} \rangle$$

where,

- $P_{\mathcal{N}} = P_L \cup P_R - FS'_L$; $\forall p_i, p_j \in FS'_L$, if $\Gamma(p_i) = \Gamma(p_j)$ then ${}^\bullet p_i \cap {}^\bullet p_j = \emptyset$;
- $FS_{\mathcal{N}} = FS_L \cup FS_R - FS'_L$;
- $ES_{\mathcal{N}} = ES_L \cup ES_R - ES'_R$;
- $T_{\mathcal{N}} = T_L \cup T_R$;
- $ST_{\mathcal{N}} = ST_L \cup ST_R$;
- $C_{\mathcal{N}} = C_L = C_R$; the set of basic colour classes of L equals that of R;
- $\forall \tau \in P_{\mathcal{N}} \cup T_{\mathcal{N}}$,

$$C_{J_{\mathcal{N}}(\tau)} = C_{\mathcal{N}}(\tau) = \begin{cases} C_L(\tau) & \text{if } \tau \in P_L \cup T_L \\ C_R(\tau) & \text{if } \tau \in P_R \cup T_R \end{cases}$$

  Recall that all places in P_R will be places of $P_{\mathcal{N}}$.

- $W^-_{\mathcal{N}} = W^-_L \cup W^-_R$;

- $\forall t \in T_{\mathcal{N}}, \forall p \in P_{\mathcal{N}}$,

$$W^+_{\mathcal{N}}(p,t) = \begin{cases} W^+_L(p,t) & \text{if } p \in P_L - FS'_L \text{ and } t \in T_L \\ W^+_R(p,t) & \text{if } p \in P_R \text{ and } t \in T_R \\ \sum_{(p_k \in FS'_L:\, \Gamma(p_k)=p)} W^+_L(p_k,t) & \text{if } t \in T_L \text{ and } p \in ES'_R \\ 0 & \text{otherwise} \end{cases}$$

  Let us remember that places in FS'_L that are output to the same transition cannot have the same image in ES'_R;

- $\forall t \in T_{\mathcal{N}}$:

$$\Phi_{\mathcal{N}}(t) = \begin{cases} \Phi_L(t) & \text{if } t \in T_L \\ \Phi_R(t) & \text{if } t \in T_R \end{cases}$$

- $\forall p \in P_{\mathcal{N}}$,

$$MP_{\mathcal{N}}(p) = \begin{cases} MP_L(p) & \text{if } p \in P_L - FS'_L \\ MP_R(p) & otherwise \end{cases}$$

4.3.2.2 Choice composition

The choice composition operation represents a logical selection of the sub-component to which a given type of information should be transferred. It is a binary composition that works over the entry sets of two *cWN*s. The information is received via entry places of the participating *cWN*s. The choice composition is defined over two distinct components, to guarantee both that the set of places corresponding to each branch of the choice do not intersect and that they do not share common output transitions. If they intersected then the places in the intersection would always receive the information. If they shared common transitions then the operation would not represent a choice of the function or subsystem to which the information is given.

The structure of the resulting component is obtained by augmenting the net with (see Figure 4.4):

- a place (the *choice place*),
- two transitions (associated with each of the participating sub-components),
- a pair of arcs from the choice place into each of the transitions associated with the sub-components, and
- a set of arcs from each of these transitions to each place in the set of selected entry places of the corresponding sub-component.

The choice place (p_c) will be an entry place of the resulting component. The function on these arcs is the identity function defined over the colour domain of the choice place. It has been defined in this way to make the choice completely random, not influenced by the arc functions. Notice that the choice composition will broadcast the information to the entry places of the component to which the information is sent.

The choice operator only makes sense if all places participating in the choice have the same colour domain, i.e. receive the same information. The colour domain of the choice place will be that of the participating places. We will denote by $Choice(L)$, the subset of places of the ES of component L that participate in choice operation, and the dual for R. The places of $Choice(L)$ and $Choice(R)$, will not be visible in the environment of the resulting component. Figure 4.5 shows how the interface of the resulting component ($\mathcal{N}$) is defined, based on the interfaces of the participating sub-components (L and R).

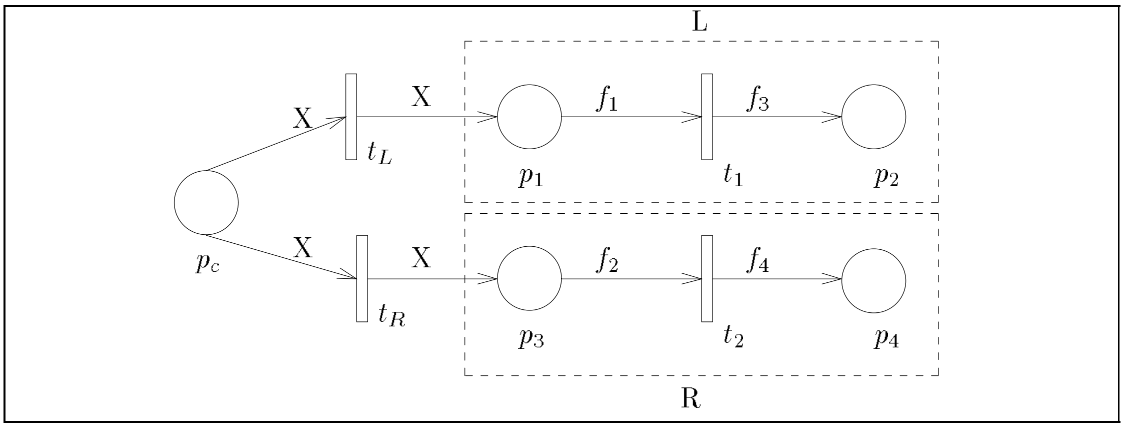


Figure 4.4: Choice composition of WN components.

Formally, the cWN $\mathcal{N}$ resulting from the choice composition of two cWNs L and R, denoted $L + R$, is a cWN defined as:

$$\mathcal{N} = \langle P_{\mathcal{N}}, ES_{\mathcal{N}}, FS_{\mathcal{N}}, T_{\mathcal{N}}, ST_{\mathcal{N}}, C_{\mathcal{N}}, J_{\mathcal{N}}, W^-_{\mathcal{N}}, W^+_{\mathcal{N}}, \Phi_{\mathcal{N}}, MP_{\mathcal{N}} \rangle$$

where,

- $P_{\mathcal{N}} = P_L \cup P_R \cup \{p_c\}$, with p_c the choice place.
- $ES_{\mathcal{N}} = ES_L \cup ES_R \cup \{p_c\} - Choice(L) - Choice(R)$; where $Choice(L)$ and $Choice(R)$ are the subsets of places of ES_L and ES_R, respectively, that participate in the choice operation;
- $FS_{\mathcal{N}} = FS_L \cup FS_R$;

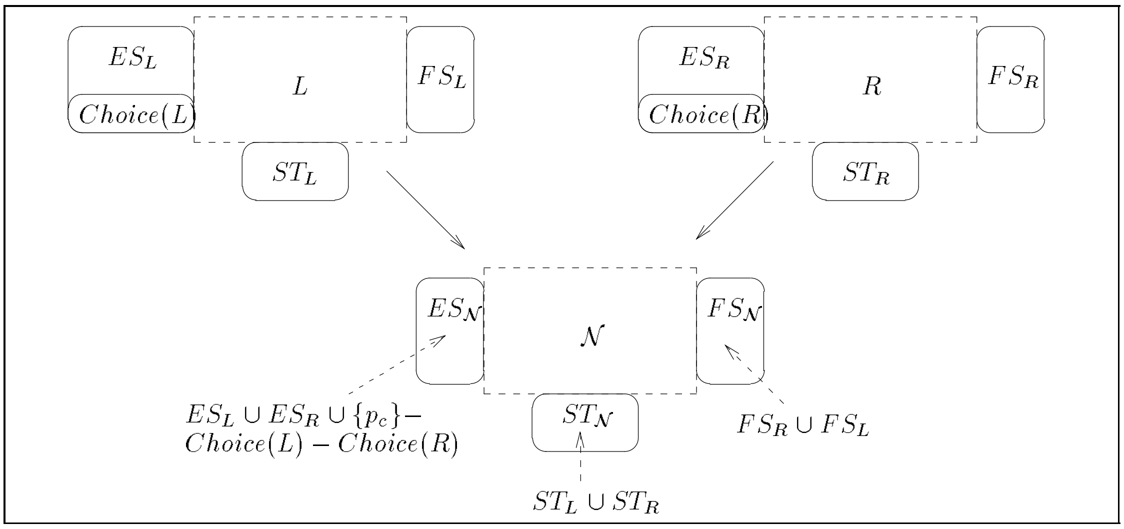


Figure 4.5: Construction of the interface of a *cWN* resulting from choice composition.

- $T_{\mathcal{N}} = T_L \cup T_R \cup \{t_L, t_R\}$, where t_L and t_R are the transitions, introduced by the choice operation, associated with the components L and R, respectively;

- $ST_{\mathcal{N}} = ST_L \cup ST_R$;

- $C_{\mathcal{N}} = C_L = C_R$;

- $C_{J_{\mathcal{N}}(p_c)} = C_{\mathcal{N}}(p_c) = C_{\mathcal{N}}(p_i)$ with p_i any place in $Choice(L) \cup Choice(R)$. Given that the arc functions of the transitions are identity functions defined according to the colour domain of p_c, $C_{\mathcal{N}}(t_L) = C_{\mathcal{N}}(t_R) = C_{\mathcal{N}}(p_c)$; all other elements in $P_{\mathcal{N}} \cup T_{\mathcal{N}}$ have the colour domains that they had in the sub-components.

- $\forall t \in T_{\mathcal{N}},\ \forall p \in P_{\mathcal{N}}$,

$$W^-_{\mathcal{N}}(p,t) = \begin{cases} X & \text{if } (t = t_L \text{ or } t = t_R) \text{ and } p = p_c \\ W^-_L(p,t) & \text{if } t \in T_L \text{ and } p \in P_L \\ W^-_R(p,t) & \text{if } t \in T_R \text{ and } p \in P_R \\ 0 & \text{otherwise} \end{cases}$$

  where X is the identity function defined over the colour domain of p_c.

- $\forall t \in T_{\mathcal{N}},\ \forall p \in P_{\mathcal{N}}$,

$$W^+_{\mathcal{N}}(p,t) = \begin{cases} X & \text{if } t = t_L \text{ and } p \in Choice(L) \\ X & \text{if } t = t_R \text{ and } p \in Choice(R) \\ W^+_L(p,t) & \text{if } t \in T_L \text{ and } p \in P_L \\ W^+_R(p,t) & \text{if } t \in T_R \text{ and } p \in P_R \\ 0 & \text{otherwise} \end{cases}$$

  where X is the identity function defined over the colour domain of p_c.

- $\forall t \in T_{\mathcal{N}}$:

$$\Phi_{\mathcal{N}}(t) = \begin{cases} \Phi_L(t) & \text{if } t \in T_L \\ \Phi_R(t) & \text{if } t \in T_R \end{cases}$$

- $\forall p \in P_{\mathcal{N}}$

$$MP_{\mathcal{N}}(p) = \begin{cases} MP_L(p) & \text{if } p \in P_L \\ MP_R(p) & \text{if } p \in P_R \\ 0 & \text{if } p = p_c \end{cases}$$

4.3.2.3 Independent parallel composition

Independent parallel composition is defined over two components that can "execute", or have an active token game, simultaneously and independently. Independent parallel composition implies no place or transition fusion. The participating sub-components do not share or interchange information. This operation corresponds to parallel composition in PA, without considering any type of co-operation or synchronisation between the components. The ES of the resulting component is obtained from the union of the ESs of the sub-components, and similarly for the FS and the ST (as shown in Figure 4.6).

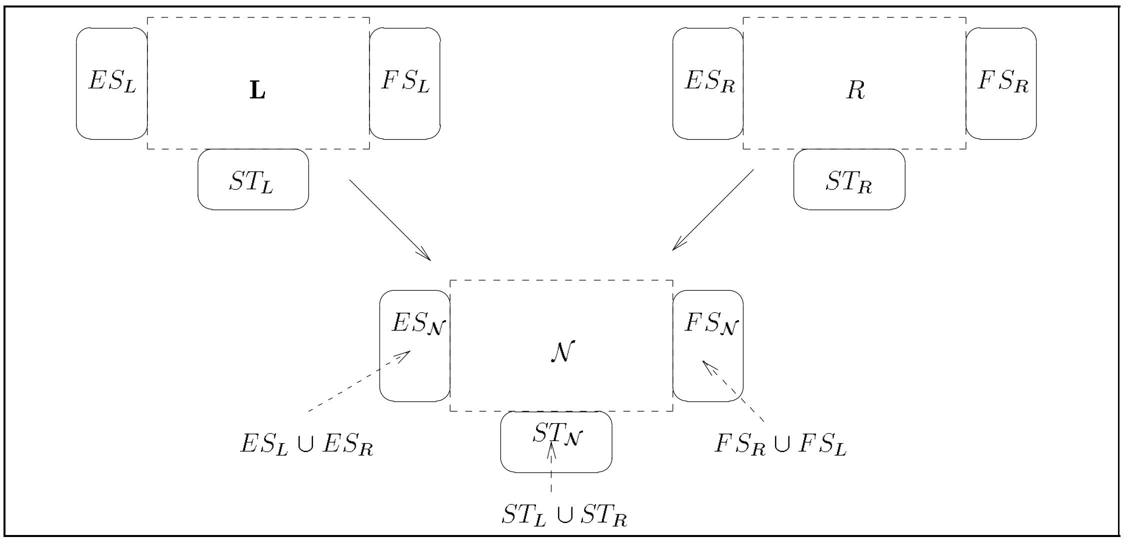


Figure 4.6: Interface of the cWN resulting from an independent parallel composition.

Formally the cWN $\mathcal{N}$ resulting from the independent parallel composition of two cWNs L and R, denoted $L|R$, is defined as:

$$\mathcal{N} = \langle P_{\mathcal{N}}, ES_{\mathcal{N}}, FS_{\mathcal{N}}, T_{\mathcal{N}}, ST_{\mathcal{N}}, C_{\mathcal{N}}, J_{\mathcal{N}}, W^-_{\mathcal{N}}, W^+_{\mathcal{N}}, \Phi_{\mathcal{N}}, MP_{\mathcal{N}} \rangle$$

where,

- $P_{\mathcal{N}} = P_L \cup P_R$;

- $ES_{\mathcal{N}} = ES_L \cup ES_R$;
- $S_{\mathcal{N}} = FS_L \cup FS_R$;
- $T_{\mathcal{N}} = T_L \cup T_R$;
- $ST_{\mathcal{N}} = ST_L \cup ST_R$;
- $C_{\mathcal{N}} = C_L = C_R$;
- $\forall \tau \in P_{\mathcal{N}} \cup T_{\mathcal{N}}$,

$$C_{J_{\mathcal{N}}(\tau)} = C_{\mathcal{N}}(\tau) = \begin{cases} C_L(\tau) & \text{if } \tau \in P_L \cup T_L \\ C_R(\tau) & \text{if } \tau \in P_R \cup T_R \end{cases}$$

- $W^-_{\mathcal{N}} = W^-_L \cup W^-_R$;
- $W^+_{\mathcal{N}} = W^+_L \cup W^+_R$;
- $\forall t \in T_{\mathcal{N}}$:

$$\Phi_{\mathcal{N}}(t) = \begin{cases} \Phi_L(t) & \text{if } t \in T_L \\ \Phi_R(t) & \text{if } t \in T_R \end{cases}$$

- $MP_{\mathcal{N}} = MP_L \cup MP_R$;

4.3.2.4 Competing parallelism composition

The motivation for competing parallelism composition is to model the situation where two functions, or two sub-systems, compete over common input information. In contrast to the choice composition, the decision of to which component the information is given is not completely random, in this case the arc functions emerging from the places with shared information can be different than the identity function. Competing parallelism can be applied over two *cWN* or internally, between the places of a single *cWN*.

Consider the competing parallel composition of two *cWN*s, L and R, to obtain a *cWN* $\mathcal{N}$. For each component it is necessary to define the set of entry places that will participate in the operation. This will be done by the definition of a function $\Lambda : (ES'_L \subseteq ES_L) \rightarrow ES_R$, which will determine which places of ES_L are to be fused with which places in ES_R. In order to maintain the number of input places for each transition of the participating components, the function Λ is defined as one-to-one, i.e. $|ES'_L| = |ES'_R|$ where ES'_R is the range of the function Λ. We can then represent Λ as a set of ordered pairs of places to be fused, with the first element belonging to L and the second to R. For each ordered pair the place belonging to L will be fused into the place of R. The fused place will inherit the arcs of the place in L. The interface of the resulting $cWN\ \mathcal{N}$, after applying the competing parallelism operation over the components L and R, is represented in Figure 4.7a.

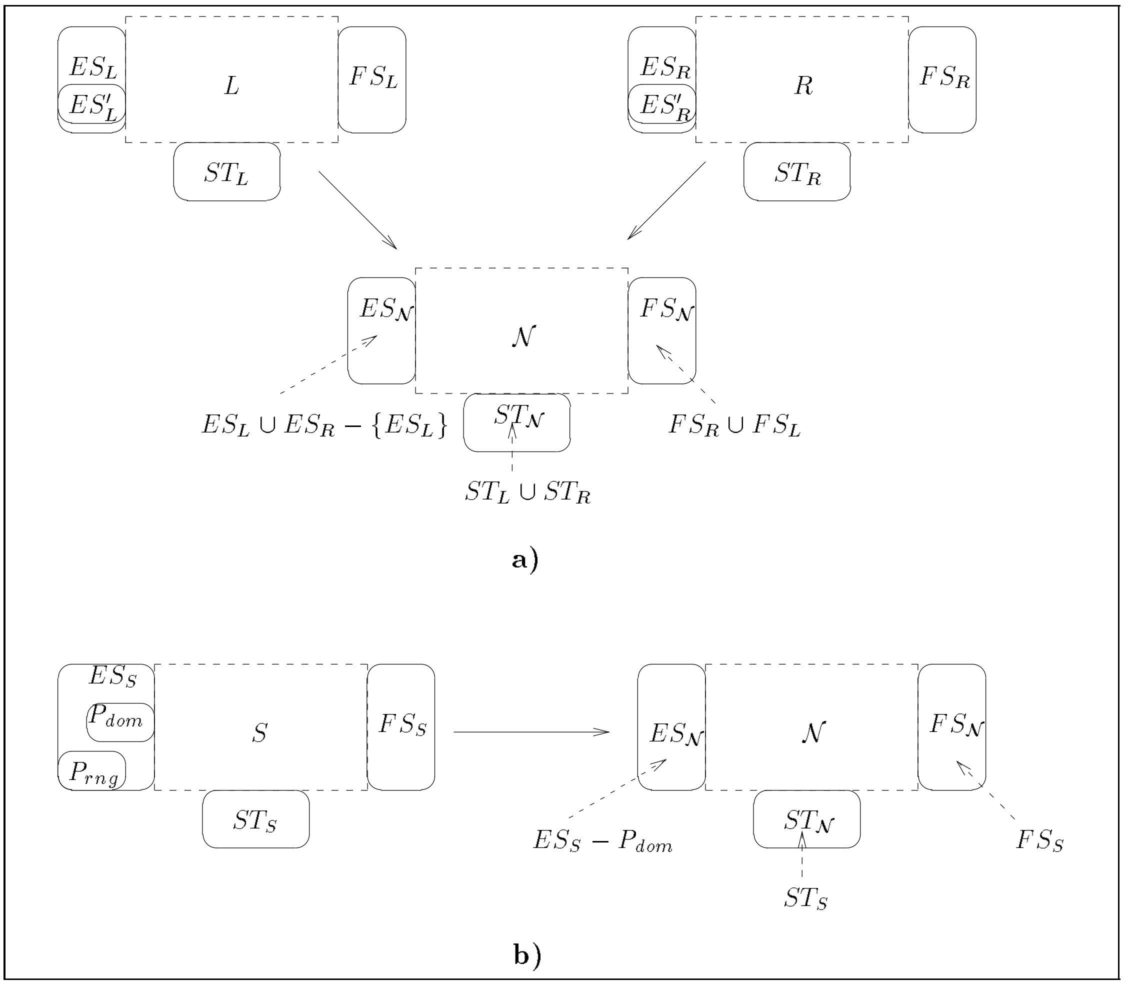


Figure 4.7: Interface of the cWN resulting from the competing parallelism composition between two cWNs (a) and within a cWN (b).

Formally the $cWN\ \mathcal{N}$ resulting from the competing parallel composition of two cWNs L and R, denoted $L|_c R$, is defined as:

$$\mathcal{N} = \langle P_{\mathcal{N}}, ES_{\mathcal{N}}, FS_{\mathcal{N}}, T_{\mathcal{N}}, ST_{\mathcal{N}}, C_{\mathcal{N}}, J_{\mathcal{N}}, W^-_{\mathcal{N}}, W^+_{\mathcal{N}}, \Phi_{\mathcal{N}}, MP_{\mathcal{N}} \rangle$$

where,

- $P_{\mathcal{N}} = P_L \cup P_R - ES'_L$, where ES'_L is the domain of the function Λ.

- $ES_{\mathcal{N}} = ES_L \cup ES_R - ES'_L$;

- $FS_{\mathcal{N}} = FS_L \cup FS_R$;

- $T_{\mathcal{N}} = T_L \cup T_R$; $ST_{\mathcal{N}} = ST_L \cup ST_R$;
- $C_{\mathcal{N}} = C_L = C_R$;
- $\forall \tau \in P_{\mathcal{N}} \cup T_{\mathcal{N}}$:

$$C_{J_{\mathcal{N}}(\tau)} = C_{\mathcal{N}}(\tau) = \begin{cases} C_R(\tau) & \text{if } \tau \in P_R \cup T_R \\ C_L(\tau) & \text{if } \tau \in P_L \cup T_L \end{cases}$$

- $\forall t \in T_{\mathcal{N}}, \forall p \in P_{\mathcal{N}}$,

$$W^-_{\mathcal{N}}(p,t) = \begin{cases} W^-_L(p,t) & \text{if } p \in P_L - ES'_L \text{ and } t \in T_L \\ W^-_L(p_k,t) & \text{if } p \in ES'_R \text{ and } \Lambda^{-1}(p) = p_k \text{ and } t \in T_L \\ W^-_R(p,t) & \text{if } p \in P_R \text{ and } t \in T_R \\ 0 & \text{otherwise} \end{cases}$$

- $\forall t \in T_{\mathcal{N}}, \forall p \in P_{\mathcal{N}}$,

$$W^+_{\mathcal{N}}(p,t) = \begin{cases} W^+_L(p,t) & \text{if } p \in P_L - ES'_L \text{ and } t \in T_L \\ W^+_L(p_k,t) & \text{if } p \in ES'_R \text{ and } \Lambda^{-1}(p) = p_k \text{ and } t \in T_L \\ W^+_R(p,t) & \text{if } p \in P_R \text{ and } t \in T_R \\ 0 & \text{otherwise} \end{cases}$$

- $\forall t \in T_{\mathcal{N}}$:

$$\Phi_{\mathcal{N}}(t) = \begin{cases} \Phi_L(t) & \text{if } t \in T_L \\ \Phi_R(t) & \text{if } t \in T_R \end{cases}$$

- $\forall p \in P_{\mathcal{N}}$,

$$MP_{\mathcal{N}}(p) = \begin{cases} MP_L(p_k) + MP_R(p) & p \in ES'_R \text{ with } \Lambda^{-1}(p) = p_k \\ MP_L(p) & \text{if } p \in P_L - ES'_L \\ MP_R(p) & \text{if } p \in P_R - ES'_R \end{cases}$$

For competing parallelism within a single component the function Λ is defined over the set of entry places of the *cWN* into itself. The range of the function Λ cannot intersect with its domain, i.e. a place is not fused with itself, and places with common input and/or output transitions cannot be fused. This last restriction is to prevent the creation of parallel arcs. The interface of the resulting *cWN* $\mathcal{N}$, after applying an internal competing parallelism to a *cWN* S, is represented in Figure 4.7b.

Formally, we define the *cWN* $\mathcal{N}$ resulting from the applying the competing parallelism composition over a single component S by:

$$\mathcal{N} = \langle P_{\mathcal{N}}, ES_{\mathcal{N}}, FS_{\mathcal{N}}, T_{\mathcal{N}}, ST_{\mathcal{N}}, C_{\mathcal{N}}, J_{\mathcal{N}}, W^-_{\mathcal{N}}, W^+_{\mathcal{N}}, \Phi_{\mathcal{N}}, MP_{\mathcal{N}} \rangle$$

where,

- $P_{\mathcal{N}} = P_S - P_{dom}$; where P_{dom} is the (set of places in) the domain of the function Λ;
- $ES_{\mathcal{N}} = ES_S - P_{dom}$;
- $FS_{\mathcal{N}} = FS_S$;
- $T_{\mathcal{N}} = T_S$;
- $ST_{\mathcal{N}} = ST_S$;
- $C_{\mathcal{N}} = C_S$;
- $\forall \tau \in T \cup P,\ J_{\mathcal{N}}(\tau) = J_S(\tau)$;
- $\forall t \in T_{\mathcal{N}}, \forall p \in P_{\mathcal{N}}$,

$$W^-_{\mathcal{N}}(p,t) = \begin{cases} W^-_S(p,t) & \text{if } p \in P_S - P_{dom} - P_{rng} \\ W^-_S(p,t) & \text{if } p \in P_{rng} \text{ and } t \in p^\bullet \\ W^-_S(p_k,t) & \text{if } p \in P_{rng} \text{ and } \Lambda^{-1}(p) = p_k \text{ and } t \in p_k^\bullet \\ 0 & \text{otherwise} \end{cases}$$

  Cases 2 and 3 are exclusive given that places that are fused cannot have common output transitions.

- $\forall t \in T_{\mathcal{N}}, \forall p \in P_{\mathcal{N}}$,

$$W^+_{\mathcal{N}}(p,t) = \begin{cases} W^+_S(p,t) & \text{if } p \in P_S - P_{dom} - P_{rng} \\ W^+_S(p,t) & \text{if } p \in P_{rng} \text{ and } t \in {}^\bullet p \\ W^+_S(p_k,t) & \text{if } p \in P_{rng} \text{ and } \Lambda^{-1}(p) = p_k \text{ and } t \in {}^\bullet p_k \\ 0 & \text{otherwise} \end{cases}$$

  Cases 2 and 3 are exclusive given that places that are fused cannot have common input transitions.

- $\forall t \in T_{\mathcal{N}},\ \Phi_{\mathcal{N}}(t) = \Phi_S(t)$;
- The initial marking MP_S is defined $\forall p \in P_{\mathcal{N}}$ as

$$MP_S(p) = \begin{cases} MP_S(p) + MP_S(p_k) & \text{if } p \in P_{rng} \text{ with } \Lambda^{-1}(p) = p_k \\ MP_S(p) & \text{otherwise} \end{cases}$$

Relation between the different types of parallelism

Consider competing parallelism defined over two components, L and R, with ES'_L and ES'_R the places of L and R, respectively, that participate in the composition, and a pair of places (p_e, p_f), such that $p_e \in ES'_L$ and $\Lambda(p_e) = p_f$ (therefore $p_f \in ES'_R$). This operation can be redefined as first applying the competing parallelism operation over L and R with $ES'_L = \{p_e\}$ and $ES'_R = \{p_f\}$, followed by the internal competing parallelism operation with P_{dom} equal to the original $ES'_L - \{p_e\}$, and P_{rng} equal to the original $ES'_R - \{p_f\}$. It can also be represented by an independent parallel composition of L and R, followed by an internal competing parallelism operation with $P_{dom} = ES'_L$ and $P_{rng} = ES'_R$.

4.3.2.5 Closing operation

When we consider systems with non-terminating behaviour it is necessary for a model—or sub-components within a model—to be able to feed information back to its input. This structure is supported by the introduction of the closing operator CL. This operation is related to the recursion operator in PA.

The closing operation considers the fusion of a final place of a component with an entry place of the same component. The resulting fusion place will be part of the entry set of the component, inheriting the arcs of both the entry and the final place involved in the composition. Multiple final places output to different transitions can be fused with a single entry place, by the iterative application of the operation. This is possible because the closing operation preserves the entry place condition (see Figure 4.8). However, we cannot fuse multiple entry places with a single final place. In cases where we needed this, we would first have to apply competing parallel composition to the entry places involved and then apply the closing operation.

The definition of a function Θ determines the pair of places (p_f, p_e) to be fused, where $p_f \in FS$ and $p_e \in ES$. The final place p_f will be fused with p_e, which will behave both as p_f and p_e. Let us remember that entry places of cWN can be output places of transitions of the cWN. The condition $C(p_e) = C(p_f)$ must hold in order to guarantee that there will be no tokens passed into the place p_e which are not defined in its colour domain.

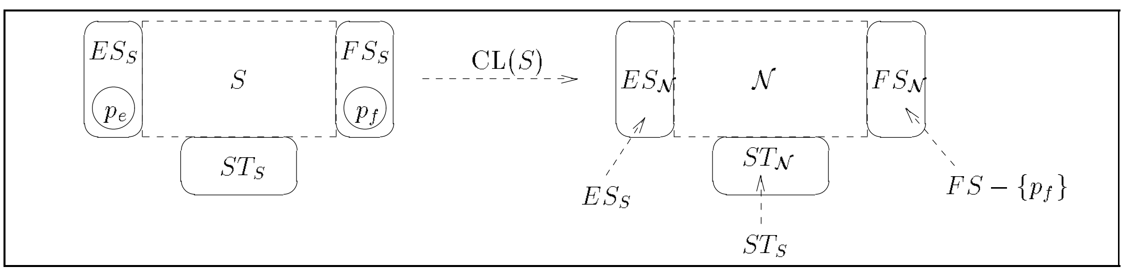


Figure 4.8: Interface of a component resulting from a closing operation.

Formally the cWN $\mathcal{N}$ resulting from applying the closing operation over a

cWN S (denoted $CL(S)$), is defined as:

$$\mathcal{N} = \langle P_{\mathcal{N}}, ES_{\mathcal{N}}, FS_{\mathcal{N}}, T_{\mathcal{N}}, ST_{\mathcal{N}}, C_{\mathcal{N}}, J_{\mathcal{N}}, W^-_{\mathcal{N}}, W^+_{\mathcal{N}}, \Phi_{\mathcal{N}}, MP_{\mathcal{N}} \rangle$$

where,

- $P_{\mathcal{N}} = P_S - \{p_f\}$;
- $ES_{\mathcal{N}} = ES_S$;
- $FS_{\mathcal{N}} = FS_S - \{p_f\}$;
- $T_{\mathcal{N}} = T_S$;
- $C_{\mathcal{N}} = C_S$;
- $\forall \tau \in T \cup P,\ J_{\mathcal{N}}(\tau) = J_S(\tau)$;
- $\forall t \in T_{\mathcal{N}}, \forall p \in P_{\mathcal{N}},\ \ W^-_{\mathcal{N}}(p,t) = W^-_S(p,t)$
- $\forall t \in T_{\mathcal{N}}, \forall p \in P_{\mathcal{N}}$,

  $$W^+_{\mathcal{N}}(p,t) = \begin{cases} W^+_S(p_f,t) & \text{if } p = p_e \text{ and } t \in {}^\bullet p_f \\ W^+_S(p,t) & \text{otherwise} \end{cases}$$

  Recall that p_e and p_f cannot have common input transitions.
- $\forall t \in T_{\mathcal{N}},\ \Phi_{\mathcal{N}}(t) = \Phi_S(t)$;
- $\forall p \in P_{\mathcal{N}},\ MP_{\mathcal{N}}(p) = MP_S(p)$;
  Recall that the initial marking of a final place is zero;

4.3.2.6 Synchronisation

One of the main advantages that PNs for the modelling of concurrent system is the diversity of synchronisations that they can represent. We will consider the synchronisation of pairs (t_1, t_2) of synchronisable transitions, possibly from different components. The synchronisation results in the fusion of the two transitions into a transition t_{syn} (see Figure 4.9). When they both belong to the same component they must be different transitions ($t_1 \neq t_2$) and their set of input and output places must not intersect.

Sub-components will not detect the changes made as a consequence of a synchronisation. They still supply the same information to a transition and receive the same type of information from it, once it has fired. The functionality of the subnets involved does not change.

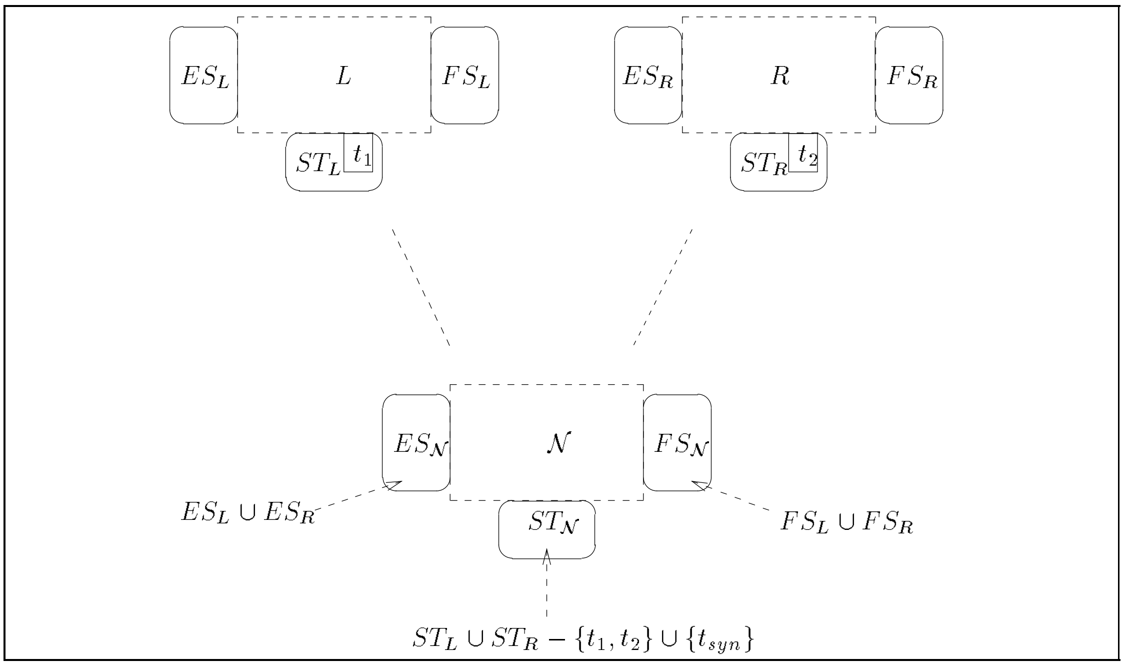


Figure 4.9: Interface of the component resulting from the synchronisation of two components, L and R.

The information about the colour domain, variables and predicates of the transitions participating in the synchronisation is required to define the colour domain and predicates, respectively, of the resulting transition t_{syn}. The colour domain of t_{syn} will be formed by the Cartesian product of the colour domains of the participating transitions. A transition with no variables is considered to have a neutral colour domain. The Cartesian product of a neutral colour domain with another C_J, is the colour domain C_J.

The predicates defined over the variables of the transitions being synchronised are inherited by the resulting t_{syn} transition. New predicates can be defined on this transition, relating variables of the synchronising transitions. The transition t_{syn} will be visible to the environment, replacing t_1 and t_2.

Formally, the *cWN* $\mathcal{N}$ resulting from applying a single transition synchronising operation over two *cWN*s L and R, denoted by $L \sim R/\{t_1, t_2\}$ is a *cWN* defined as:

$$\mathcal{N} = \langle P_{\mathcal{N}}, ES_{\mathcal{N}}, FS_{\mathcal{N}}, T_{\mathcal{N}}, ST_{\mathcal{N}}, C_{\mathcal{N}}, J_{\mathcal{N}}, W_{\mathcal{N}}^{-}, W_{\mathcal{N}}^{+}, \Phi_{\mathcal{N}}, MP_{\mathcal{N}} \rangle$$

where,

- $P_{\mathcal{N}} = P_L \cup P_R$;
- $ES_{\mathcal{N}} = ES_L \cup ES_R$;
- $FS_{\mathcal{N}} = FS_L \cup ES_R$;

- $T_{\mathcal{N}} = T_L \cup T_R - \{t_1, t_2\} \cup \{t_{syn}\}$; where t_1 and t_2 are the synchronising transitions of the *cWN*s L and R, respectively;

- $ST_{\mathcal{N}} = ST_L \cup ST_R - \{t_1, t_2\} \cup \{t_{syn}\}$;

- $C_{\mathcal{N}} = C_L = C_R$;

- $\forall t \in T_{\mathcal{N}}$,

$$C_{J_{\mathcal{N}}(t)} = C_{\mathcal{N}}(t) = \begin{cases} C_L(t) & \text{if } t \in T_L \\ C_R(t) & \text{if } t \in T_R \\ C_L(t_1) \times C_R(t_2) & \text{if } t = t_{syn} \end{cases}$$

  $\forall p \in P_{\mathcal{N}}$,

$$C_{J_{\mathcal{N}}(p)} = C_{\mathcal{N}}(p) = \begin{cases} C_L(p) & \text{if } p \in P_L \\ C_R(p) & \text{if } p \in P_R \end{cases}$$

- $\forall t \in T_{\mathcal{N}},\ \forall p \in P_{\mathcal{N}}$,

$$W^-_{\mathcal{N}}(p,t) = \begin{cases} W^-_L(p,t) & \text{if } p \in P_L \text{ and } t \in T_L \\ W^-_R(p,t) & \text{if } p \in P_R \text{ and } t \in T_R \\ W^-_L(p,t_1) & \text{if } p \in P_L \text{ and } t = t_{syn} \\ W^-_R(p,t_2) & \text{if } p \in P_R \text{ and } t = t_{syn} \\ 0 & \text{otherwise} \end{cases}$$

- $\forall t \in T_{\mathcal{N}},\ \forall p \in P_{\mathcal{N}}$,

$$W^+_{\mathcal{N}}(p,t) = \begin{cases} W^+_L(p,t) & \text{if } p \in P_L \text{ and } t \in T_L \\ W^+_R(p,t) & \text{if } p \in P_R \text{ and } t \in T_R \\ W^+_L(p,t_1) & \text{if } p \in P_L \text{ and } t = t_{syn} \\ W^+_R(p,t_2) & \text{if } p \in P_R \text{ and } t = t_{syn} \\ 0 & \text{otherwise} \end{cases}$$

- $\forall t \in T_{\mathcal{N}}$:

$$\Phi_{\mathcal{N}}(t) = \begin{cases} \Phi_L(t) & \text{if } t \in T_L \\ \Phi_R(t) & \text{if } t \in T_R \\ \Phi_L(t_1) \cup \Phi_R(t_2) \cup \text{new\_pred} & \text{if } t = t_{syn} \end{cases}$$

  where new_pred is the set of new predicates defined over the variables of t_{syn};

- $\forall p \in P_{\mathcal{N}}$,

$$MP_{\mathcal{N}}(p) = \begin{cases} MP_L(p) & \text{if } p \in P_L \\ MP_R(p) & \text{if } p \in P_R \end{cases}$$

In the case of the synchronisation being applied over a single *cWN* S, the resulting *cWN* $\mathcal{N}$, is defined as:

$$\mathcal{N} = \langle P_{\mathcal{N}}, ES_{\mathcal{N}}, FS_{\mathcal{N}}, T_{\mathcal{N}}, ST_{\mathcal{N}}, C_{\mathcal{N}}, J_{\mathcal{N}}, W^-_{\mathcal{N}}, W^+_{\mathcal{N}}, \Phi_{\mathcal{N}}, MP_{\mathcal{N}} \rangle$$

where,

- $P_{\mathcal{N}} = P_S$;
- $ES_{\mathcal{N}} = ES_S$;
- $FS_{\mathcal{N}} = FS_S$;
- $T_{\mathcal{N}} = T_S - \{t_1, t_2\} \cup \{t_{syn}\}$; where t_1 and t_2 are the synchronising transitions ($t_1 \neq t_2$) and $((^\bullet t_1 \cap {}^\bullet t_2 = \emptyset) \wedge (t_1^\bullet \cap t_2^\bullet = \emptyset))$;
- $ST_{\mathcal{N}} = ST_S - \{t_1, t_2\} \cup \{t_{syn}\}$;
- $C_{\mathcal{N}} = C_S$;
- $\forall t \in T_{\mathcal{N}}$,

$$C_{J_{\mathcal{N}}(t)} = C_{\mathcal{N}}(t) = \begin{cases} C_S(t) & \text{if } t \in t \neq t_{syn} \\ C_S(t_1) \times C_S(t_2) & \text{if } t = t_{syn} \end{cases}$$

  $\forall p \in P_{\mathcal{N}},\ C_{J_{\mathcal{N}}(p)} = C_{\mathcal{N}}(p) = C_S(p)$;

- $\forall t \in T_{\mathcal{N}},\ \forall p \in P_{\mathcal{N}}$,

$$W^-_{\mathcal{N}}(p,t) = \begin{cases} W^-_S(p,t) & \text{if } t \neq t_{syn} \\ W^-_S(p,t_1) & \text{if } t = t_{syn} \text{ and } p \in {}^\bullet t_1 \\ W^-_S(p,t_2) & \text{if } t = t_{syn} \text{ and } p \in {}^\bullet t_2 \\ 0 & \text{otherwise} \end{cases}$$

- $\forall t \in T_{\mathcal{N}},\ \forall p \in P_{\mathcal{N}}$,

$$W^+_{\mathcal{N}}(p,t) = \begin{cases} W^+_S(p,t) & \text{if } t \neq t_{syn} \\ W^+_S(p,t_1) & \text{if } t = t_{syn} \text{ and } p \in t_1^\bullet \\ W^+_S(p,t_2) & \text{if } t = t_{syn} \text{ and } p \in t_2^\bullet \\ 0 & \text{otherwise} \end{cases}$$

- $\forall t \in T_{\mathcal{N}}$:

$$\Phi_{\mathcal{N}}(t) = \begin{cases} \Phi_S(t) & \text{if } t \neq t_{syn} \\ \Phi_S(t_1) \cup \Phi_S(t_2) \cup \text{new\_pred} & \text{if } t = t_{syn} \end{cases}$$

  where new_pred is the set of new predicates defined over the variables of t_{syn};

- $\forall p \in P_{\mathcal{N}},\ MP_{\mathcal{N}}(p) = MP_S(p)$;

We have defined the set of compositional operations that have been included in the methodology for the compositional construction and analysis of *cWN* systems proposed in this dissertation. They have been summarised in Table 4.1.

Composition operations		
Sequential	$L;R$	Fuses final places of L with entry places of R. It requires the definition of a function $\Gamma: FS'_L \rightarrow ES_R$, where $FS'_L \subseteq FS_L$, to determine which places will be fused. The places fused are not included in the environment of the resulting component.
Choice	$L+R$	It combines two components by using a subnet formed by a *choice* place and two transitions output to the choice place. The places participating in the choice composition are not visible in the environment of the resulting component.
Independent Parallelism	$L \vert R$	Composes two components by union of their places and transitions. The operation does not involve any place or transition fusion. The environment of the resulting component is the union of the environments of the participating components.
Competing Parallelism	$L \vert_c R$	Composes two components by fusion of entry places. It requires the definition of a one-to-one function, Λ , that determines the pairs of places (one from each component) to be fused. The resulting fusion places will be in the entry set of the resulting component. It can also be applied to fuse entry places within a component. In this case the entry places cannot have common input or output transitions.
Closing	$CL(S)$	An internal operation that fuses a final place with an input place. The pair of places to be fused is defined by a function Θ. The resulting fusion place will belong to the set of entry places of the resulting component.
Synchronisation	$L \sim R/\{t_1, t_2\}$	Fuses two transitions. It can be applied over different components or internally, within a component. The transitions fused cannot have common input or output places. The resulting fusion transition will be in the environment of the resulting component.

Table 4.1: Summary of the syntax and semantics of the composition operations.

4.3.3 Using the composition operations for the construction of a cWN system

Let us now study a small example to illustrate how the compositional operations defined can be applied over the sub-components of a model. The example chosen is the well-known problem of the "Dining Philosophers". There is a group of philosophers sitting around a table, on which there are as many forks as philosophers and a huge bowl of spaghetti. However, because the spaghetti is very tangled, a philosopher requires two forks to eat. Therefore, he will have to compete with his neighbours for the use of his forks. A philosopher thinks for a while, then eats and once he has finished eating he returns to thinking.

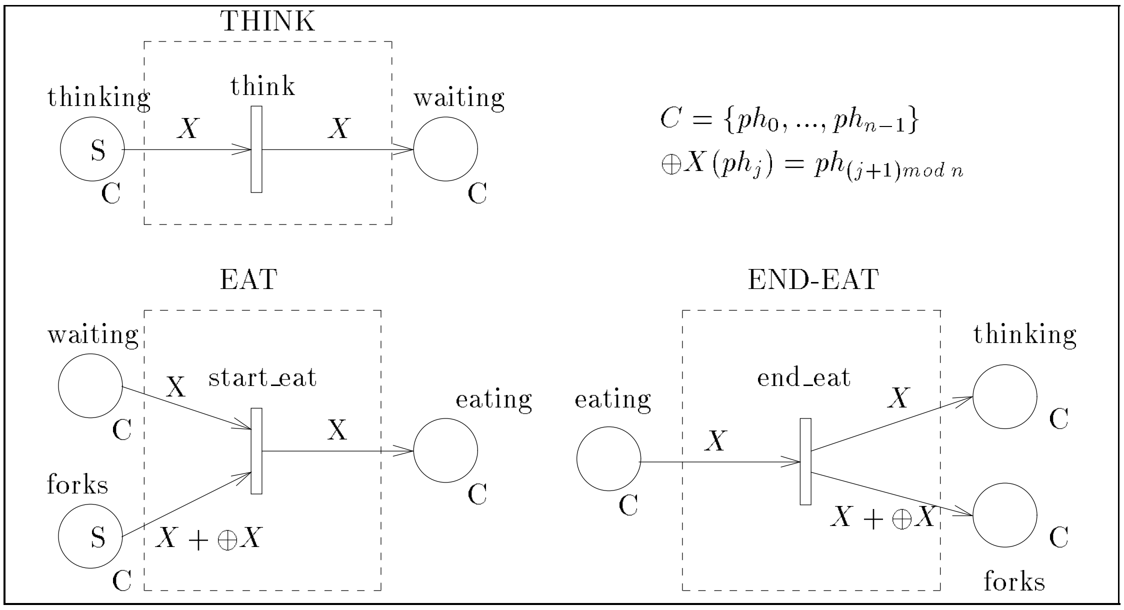


Figure 4.10: Basic components of the Dining Philosophers model

Before defining the basic components of the model we must determine the basic colour classes. The two types of elements that we must represent are philosophers and forks. As we have said, there is the same number of forks as philosophers. We will assume that philosophers and forks are numbered in a clockwise manner. When a philosopher is thinking his fork is considered to be free or usable by another philosopher. When a philosopher is eating he uses his fork and that of his left hand neighbour philosopher. This reflects a direct relation between the two sets of elements, allowing us to represent both philosophers and forks with a single ordered colour class $C = \{ph_o, \cdots, ph_{(n-1)}\}$, where n is the number of forks and philosophers.

In Figure 4.10 the set of basic components for the dining philosophers model is presented. Given that there is only one basic colour class with no static subclasses the indexes of the arc functions have been omitted. A philosopher will first think (component THINK), once hungry he will wait until he can eat (placing a token in place waiting). Component EAT models a philosopher eating, for which it requires a hungry philosopher and the availability of his forks; when he has both, the philosopher can start-eating. Component END-EAT receives a philosopher who was eating and wants to stop, this component sends the philosopher back to thinking and releases the forks. The marking S corresponds to the marking that has one coloured object for each colour in the colour class C.

Given the group of bWN components presented in Figure 4.10, the following steps can be followed in order to obtain the complete model of the problem. To identify instances of places that have the same name but which originally belong to different components we will employ a dot notation. The first element corresponds to the name of the component to which the place originally belongs, and the second element the name of the place (component names are in capital letters and place names in small letters).

Construction Steps:

1. Sequential composition of EAT with END-EAT (see component EAT_2 in Figure 4.11).

   EAT_2 = EAT;END-EAT with Γ(EAT.eating)= END-EAT.eating;

2. Closing operation over EAT_2 (see component EATING in Figure 4.11).

   EATING = CL(EAT_2) with Θ(END-EAT.forks)=EAT.forks;

3. Sequential composition of THINK with EATING (see Figure 4.12).

   THINK_EAT = THINK;EATING with
   Γ(THINK.waiting)= EATING.waiting;

4. Closing operation over THINK_EAT (see Figure 4.12).

   DINING_PHIL = CL(THINK_EAT) with
   Θ(EATING.thinking)= THINK.thinking.

We have shown how the composition operations can be used to build a cWN system. First we identified the colour domain of the whole system, then the basic functions, thus the basic components, and then the relation between the components. The way and order in which composition operations are applied may affect the outcome of the resulting cWN model. This will depend on the places and transitions in the interface of the intermediate components. In the

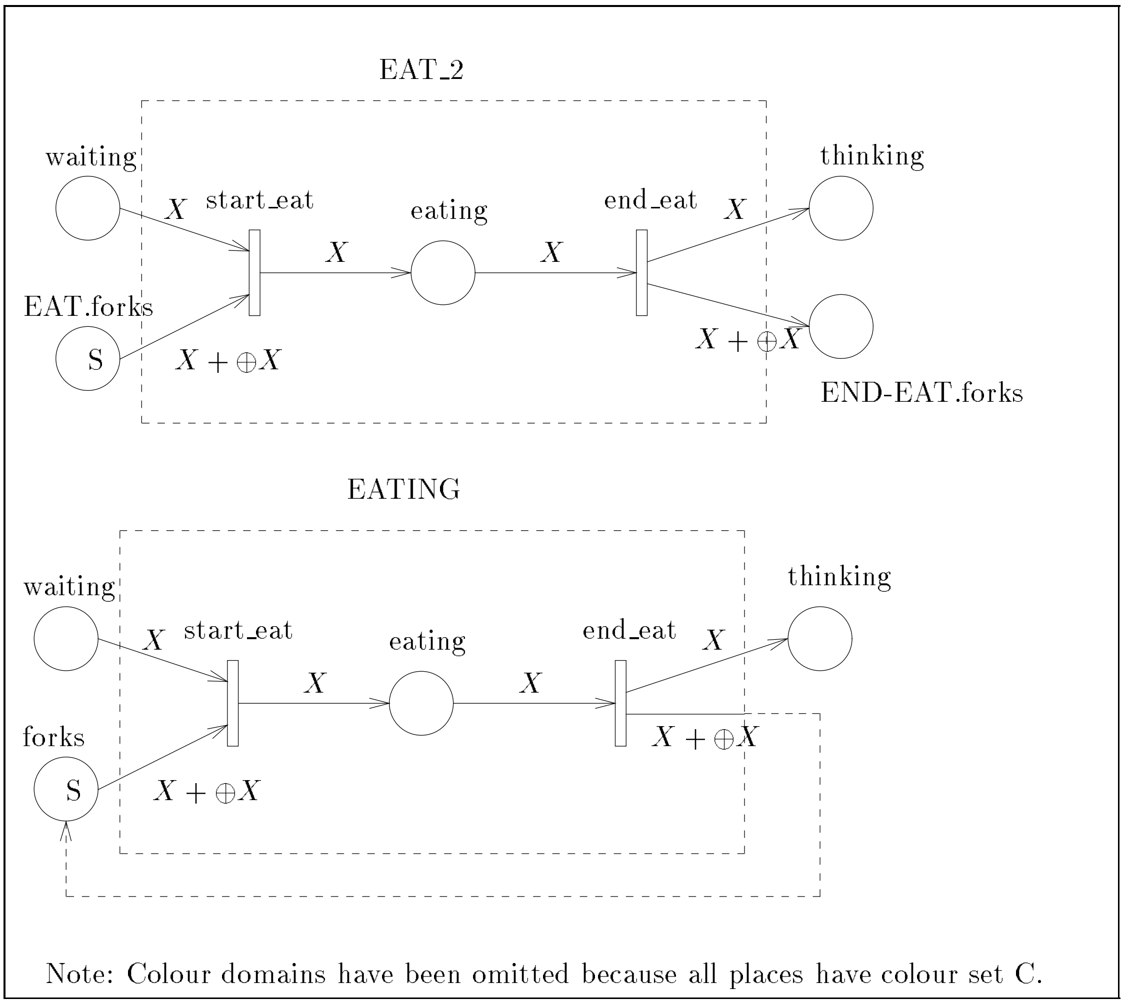


Figure 4.11: Applying the compositional operations to form the model of the Dining Philosophers problem (steps 1 and 2).

following section we discuss the factors that need to be considered when creating a *cWN* model. The aim of this analysis is to offer a set of guidelines to the modeller.

4.4 General guidelines for the construction of a *cWN* system

The aim of this section is to present a series of suggestions of what should be taken into account when using the composition operations defined to construct a *cWN* system. These guidelines are not intended to be taken as strict steps of the construction process.

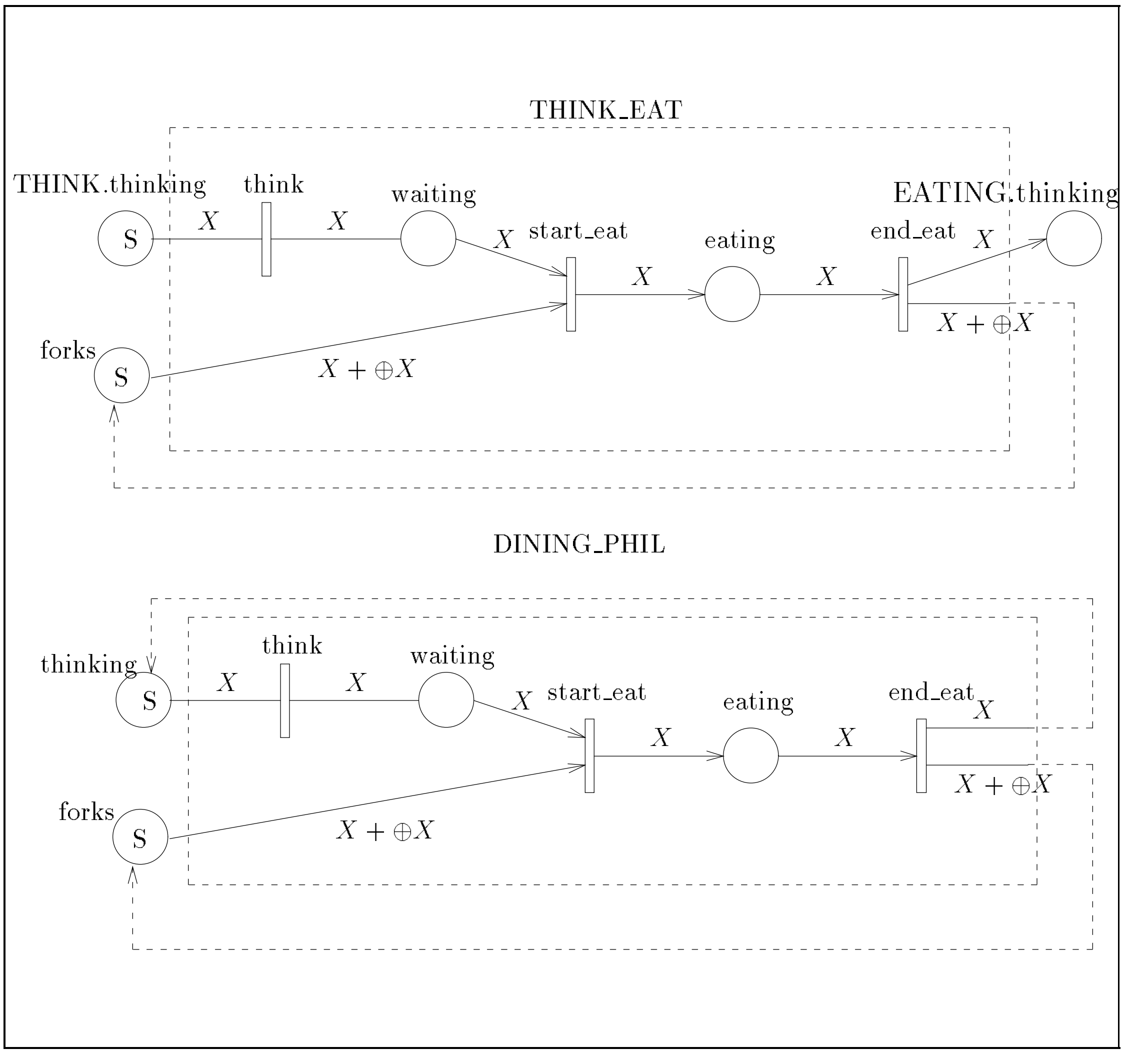


Figure 4.12: Applying the compositional operations to form the model of the Dining Philosophers problem (steps 3 and 4).

4.4.1 Identification of the basic colour classes

In *cWN*s colour classes are defined over the system as a whole. This restriction is imposed to avoid ambiguity, repetitions and mismatches. The fusion of places has been defined over places that have the same colour domain. Allowing the definition of colour classes at the level of components could require the definition of a function to associate colour classes of one component with colour classes of another. Two colour classes of different components that represent the same elements could be named or defined differently in the different components. Furthermore, what can be considered as a colour class at the level of a component can at the level of the whole system constitute a static sub-class. It would then be necessary to support the creation of new colour classes, of new static sub-classes, and redefinition of colour functions and of transition predicates, defined over the existing colour classes.

Given the system's specification, it should be possible to identify the type of entities participating in the system. In the example of the dining philosophers (Section 4.3.3) we could deduce, from the description of the problem, that philosophers and forks were the types of entities represented by tokens. By further analysis of these types we deduced that one colour class was sufficient.

The static colour sub-classes of the basic colour classes must also be defined from the beginning. The neutral colour domain is tacitly considered to be within the set of colour domains.

4.4.2 Selection of the composition operation

In order to build a model starting from the system's description we combine a top-down analysis with a compositional (bottom-up) construction method. Having defined the *bWN*s and/or identified (existing) *cWN*s that model the different parts of the system, it is necessary to establish how they should be composed.

Closing operation : If we need to feed information back from places in FS into places in ES of a component then a closing operation is required. The closing operation only fuses a pair of places at a time, therefore if we want to fuse more than two places, we will have to perform as many closing operations as the number of places in FS that need to be fused. We cannot fuse a single final place with several entry places, because when a final place participates in the closing operation the fused place is not included in the resulting FS.

Choice and competing parallelism between two components : If there exists a pair of *cWN*s, L and R, with common information requirements, for which they have to compete, then there are two options for their composition. If we want the assignment to be completely random and possibly that more than one place in each component obtains that information, then we would use the choice composition. Otherwise, we apply the competing parallelism operation, where the functions of the out-going arcs of the places participating in composition can influence outcome of the competition.

Competing parallelism within a single component: Consider a *cWN* resulting from applying a composition operation between two components. If within the set of entry places there are places that represent the same information and state, and thus have the same meaning in the model, then we can apply competing parallelism over this set of places.

Sequential composition : If the information offered by a place or a set of places in the FS_L of a cWN, L, is required by a place or a set of places in the ES_R of a component R, then we should define a sequential composition over L and R.

Independent parallel composition : In general, independent parallel composition is used when it is necessary to group components that do not interact, i.e. components that have independent token games. However, it can also be used to model another, more complicated, relation. If there is a pair of cWNs, L and R, for which:

- there is a third cWN, S, which needs to be sequentially composed with both L and R, i.e. $L; S$ and $R; S$, and
- L and R do not require information from each other and do not share common input information,

then L and R must first be composed by independent parallel composition, and then the resulting component can be sequentially composed with S.

Synchronisation : Synchronisation between two transitions is applied when it is necessary to represent a communication process between the functions modelled by the transitions.

4.4.3 Precedence relation between the operations

The order in which the composition operations are applied can strongly influence the way in which the components of a system can be made to interact. By analysing the interface of the resulting component after the application of each composition operation, we aim to offer a guide to the modeller on how to join a set of components to obtain the results desired. These guidelines are presented in terms of the following ordering of the operations—from strongest to weakest—according to their definition of the interface of the resulting component, restricting the application of other composition operations, and the information required from the subcomponents onto which an operation is applied. This is not a strict ordering, nor is it unique. For example, the closing operation may precede an internal parallel composition involving a common entry place p_e, or they could be applied in the reverse order, obtaining the same results.

Closing : This is an internal operation that only requires information from the component to which it is applied. Its influence on the interface of the component is that the final place will not exist as such, however, its information is still accessible in the resulting fusion place, which will be an entry place of the resulting component.

Internal competing parallelism : This is also an internal operation. It requires information only about the entry places of the component. This information is preserved in the resulting component through the resulting fusion places.

Internal synchronisation : An internal operation that requires information about the synchronisable transitions of the component. The resulting t_{syn} is also visible and acts as both of the synchronising transitions, although new predicates can be defined.

Choice (pre-selection) : This operation is defined over two components, requiring information about their entry sets. After the composition the participating entry places are not part of the interface of the resulting component. A choice composition can be followed by a closing operation, an internal competing parallelism or an internal synchronisation over the resulting component.

Competing parallelism between two components : This is a binary operation that requires information about the entry sets of the participating components. All information about the entry places involved in the operation is inherited by the resulting fusion places. The set of final places and synchronisable transitions of the resulting component are formed by the union of the corresponding sets of the participating components.

Sequential : This is a binary operation that requires information about the entry set of one component and the final set of the other. After the composition the resulting fusion places will not form part of the interface of the resulting component.

Synchronisation between two components : This operation requires information about the synchronisable transitions of the participating components. The entry and final sets of the resulting component will be the union of the corresponding sets of the participating components. The transition t_{syn} resulting from the synchronisation will belong to the interface of the resulting component and will act as both of the synchronising transitions.

Independent Parallelism : This is a binary operation where all the information of the subnets is preserved. This operation does not require information about the entry set, final set or the set of synchronisable transitions. The application of an independent parallel composition prevents the application of other binary operation. However, given that it preserves the interfaces of the participating components the fusion of places or transitions can be still be done by internal compositions. The choice operation has no internal equivalent, hence it is necessary to apply it before an independent parallel composition.

The identification of which composition operation should be applied, and a precedence relation between the composition operations, are intended to guide the modeller in the construction process. To illustrate the use of these guidelines, in the following section we apply them to a slightly more difficult example than the "Dining Philosophers" problem.

4.4.4 Model of a multiprocessor architecture

In this section we apply the guidelines described to a modification of the example of a Multiprocessor system introduced in [AMBC82]. There is a common memory (CM) distributed into modules local to each processor. It is assumed that memories are logically divided into private and common areas (PM_i and CM_i). Each processor i is connected to its own local memory module by a *local bus* (LB_i). A processor i accesses a non-local CM_j module using its own LB_i, the global bus (GB), and the LB_j connected to the destination CM_j module. Figure 4.13 shows the structure of this architecture.

A processor that gains access to the GB has priority to use any reachable resource and may preempt other processors. Processors preempted while using their LB will become blocked; processors preempted while waiting for the GB maintain their state, but release their LB.

Each processor runs an interactive application that will receive a input via an I/O device and will then process it. The processing of the input may require one or more memory accesses, concluding by sending an output to the corresponding I/O device.

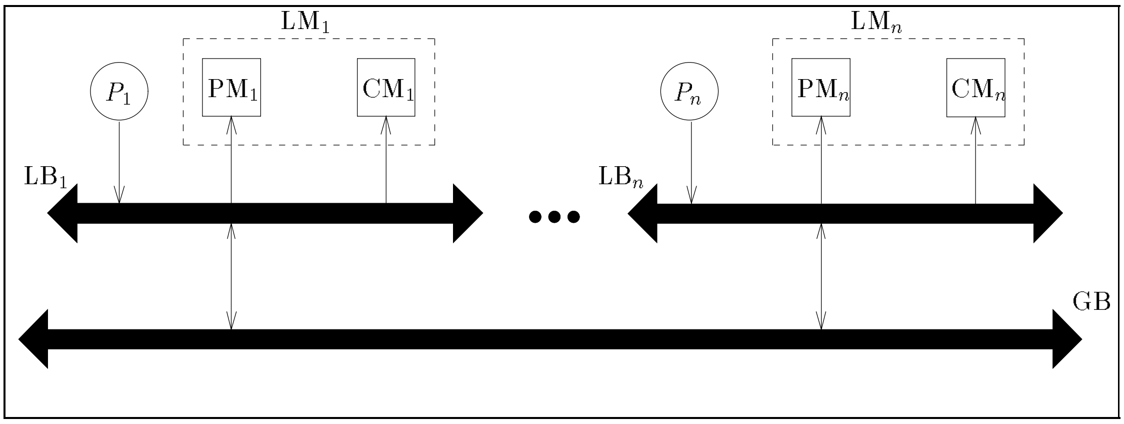


Figure 4.13: A multiprocessor architecture with distributed common memory modules, accessible through the LBs of the processors connected to each one.

Let us now analyse which basic colour classes should be employed in modelling the system. A private memory module can be identified by its associated processor. In the same way, the common memory area and the local bus associated with a processor can also be identified by the local processor. Each processor runs one process. Given these arguments the only colour class needed for this model will be the Processors colour class, with an identifier per processor. The variables in the model will be represented with the last letters of the alphabet, this has been considering that there is only one colour class with only itself as static sub-class and in this way indexes can be avoided.

Having defined the basic colour classes, we now identify the basic functions of the system. Let us start with the identification of the basic functions of the subsystem representing the multiprocessor architecture. Each basic function will in this case correspond to a *bWN*. Figure 4.14 on page 60 shows the *bWN* corresponding to each basic function. In Table 4.2 we give the correspondence between the names of the inputs and outputs of the basic functions and the names given to the places representing them in the *cWN*s.

Basic Functions of the multiprocessor architecture's subsystem

Generate Request (*bWN* A in Figure 4.14) Represents the activity of a processor before requesting access to either its local memory module or to a remote CM module.

Inputs : Active Processors

Outputs : Request Memory Access

Access LB (*bWN* B in Figure 4.14) This function represents the administration of the LB from the point of view the local processor. If the local pro-

PLACE	ABBREVIATION
Active Processors	AP
Request Memory Access	RMA
Free Local Buses	FLB
Memory Requests	MR
Busy Local Buses	BLB
Request PM	RPM
Request Local CM	RLCM
Request GB and remote CM	RGC
Global Bus	GB
GB and free LB Granted	GBGF
GB and busy LB Granted	GBGB
Finished memory access	FMA

Table 4.2: Correspondence between abbreviations used in the *cWN*s and the name inputs and outputs of the basic functions of the multiprocessor architecture.

cessor is requesting the use of its LB to access its local memory module, it verifies that the LB is free. If it is not free it will mean that it is in use by a remote processor which has priority over the use of the LB.

Inputs : Free LB, Requests memory access

Outputs : Busy LB (in use by local Processor), Memory request

Use PM (*bWN* C in Figure 4.14) Represents the time spent by the processor accessing its PM. To use its PM the processor must have gained access to its LB.

Inputs : Busy LB (by local processor), Request PM

Outputs : Free LB , processors that have finished memory access

Use local CM (*bWN* D in Figure 4.14) Represents the time spent by a processor using its local CM module. To use its local CM module the processor must have its LB assigned to it.

Inputs : Busy LB (by local processor), Request local CM

Outputs : Free LB, processors that have finished memory access

Access GB and LB (busy)(*bWN* E in Figure 4.14) This function represents the assignment of the global bus and of the remote local bus to a processor. This function assumes that the remote local bus is busy, in use by the local processor. If the access to the GB is granted before the remote local bus is

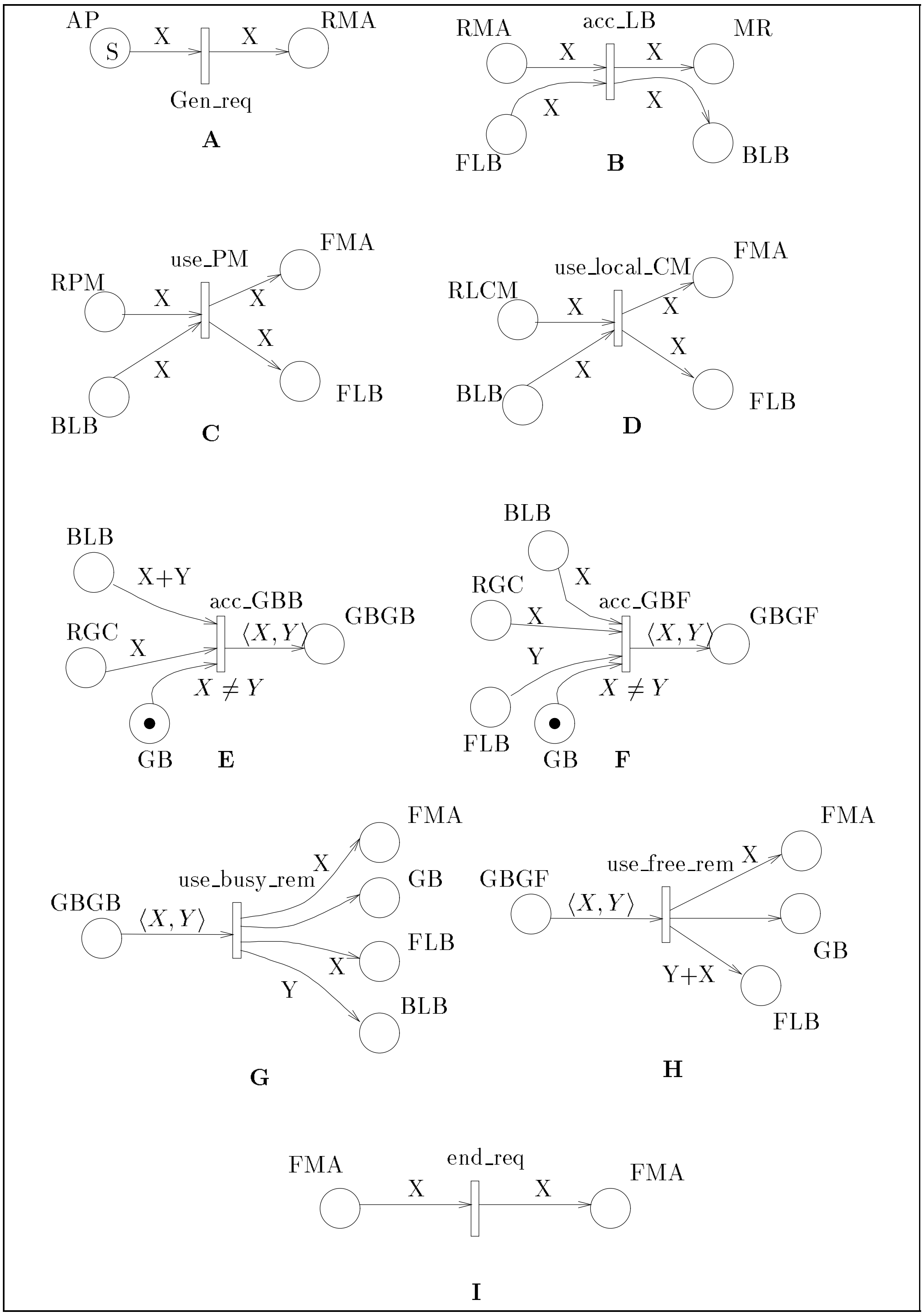


Figure 4.14: *bWN*s of the model of the multiprocessor architecture.

freed the function preempts the local processor, blocks it (maintaining its state) and uses the LB. As in the previous function processors that request access to a remote CM module "queue" for the access to the GB. To request the GB the processor must have gained access to its LB.

Inputs : Busy LB (in use by processor requesting GB), processor requesting access to GB, GB, LB of remote processor (busy)

Outputs : GB granted, busy remote LB granted

Access GB and LB (free)(*bWN* F in Figure 4.14) This function represents the assignment of the global bus and of the remote local bus to a processor. This function assumes that the remote local bus is free. Processors that request access to a remote CM module "queue" for the access to the GB. To request the GB the processor must have gained access to its LB.

Inputs : Busy LB (in use by processor requesting GB), processor requesting access to GB, GB, LB of remote processor (free)

Outputs : GB granted, free remote LB granted

Use Remote CM - LB busy (*bWN* G in Figure 4.14) Once a remote processor has gained access to the GB and to the LB of the remote site, it can work on the remote CM module. This function assumes that the LB of the remote site has been preempted.

Inputs : Identifier of the remote access (case of a busy LB) formed by the identifier of the processor accessing the remote site and the identifier of the remote site.

Outputs : Busy LB (of the remote site), GB, Free LB (local LB), processors that have finished memory access

Use Remote CM - LB Free (*bWN* H in Figure 4.14) This function behaves as the function above (Use Remote CM - LB busy) but it assumes that the LB of the remote site was free when that GB was assigned.

Inputs : Identifier of the remote access (case of a free LB) formed by the identifier of the processor accessing the remote site and the identifier of the remote site.

Outputs : Free LB (of both accessed and accessing module), FMA, GB

Finish memory access (*bWN* I in Figure 4.14) Once the memory request has been satisfied the processor returns to its active state.

PLACE	ABBREVIATION
I/O entry	I/O_E
Entry processing	EP
Memory access	MA
End memory access	EMA
Generate memory access	GMA
Further processing	Fproc

Table 4.3: Correspondence between abbreviations used in the *cWN*s and the name inputs and outputs of the basic functions of the I/O application.

Inputs : processor that has finished memory access

Outputs : Active processors

The basic functions of the interactive processes running on each processor can be identified as follows.

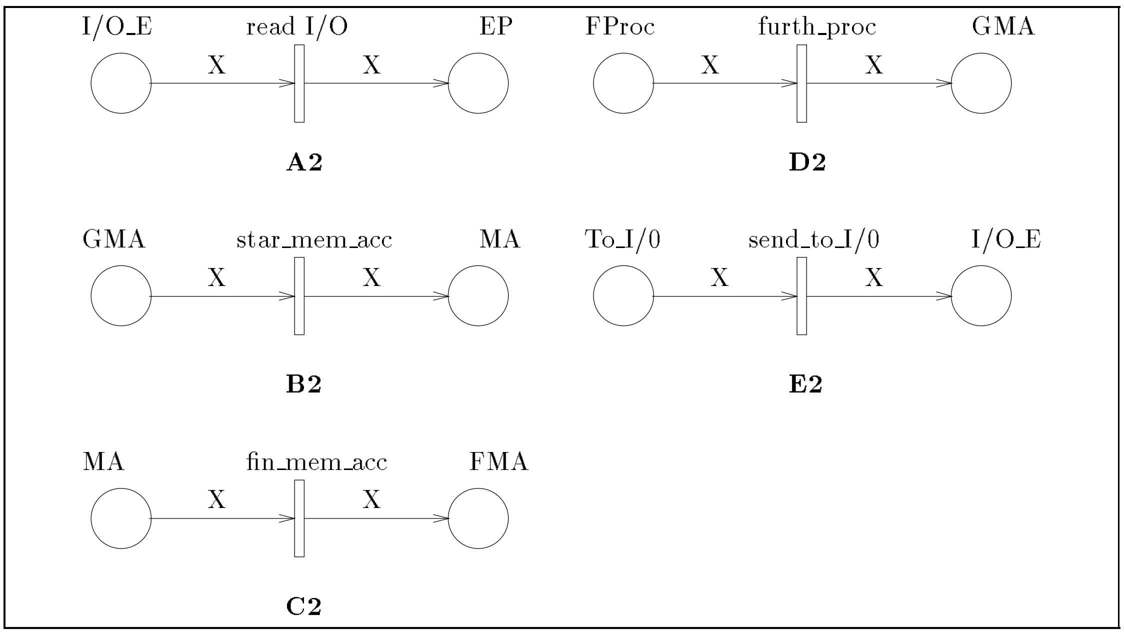


Figure 4.15: Basic functions of the I/O application running in each processor.

Basic function of the interactive processes

Read from I/O device (*bWN* A2 in Figure 4.15) Reads information from the input device

Inputs :Identifier of the process reading the information.

Outputs : Identifier of the process requesting memory access.

Start memory access (*bWN* B2 in Figure 4.15) Processes the information read and generates a memory access.

Inputs :Identifier of the process requesting memory access.

Outputs : Identifier of the process whose memory request is been attended to.

Finish memory access (*bWN* C2 in Figure 4.15) Processing after memory access.

Inputs : Identifier of the process whose memory request is been attended to.

Outputs : Identifier of the process that has concluded its memory access.

Generate new request (*bWN* D2 in Figure 4.15) If further access to memory is necessary it determines which these should be.

Inputs : Identifier of the process that needs further memory accesses.

Outputs : Identifier of the process requesting memory access.

Send output to I/0 device (*bWN* E2 in Figure 4.15) Once all the necessary processing of the input has been done it sends the appropriate output to the I/O device.

Inputs : Identifier of the process that has concluded processing its input.

Outputs : Identifier of the process reading the information.

Based on the *bWN*s presented in Figure 4.14, we execute the following steps to obtain the model of the multiprocessor system in a compositional manner. To distinguish two places which originally have the same name, we use a dot notation, prefixing to the name of each of the places the name of the component to which it belongs.

Following the guideline we identify that we need to perform a choice operation over the *bWN*s C and D (see Figure 4.16). Choice(C) will be formed by C.RPM and Choice(D)= {D.RLCM}. The interface of the resulting component, CD, will be:

ST:	{use_PM, use_local_CM}
ES:	{C.BLB, D.BLB, p_c}
FS:	{C.FLB, D.FLB, C.FMA, D.FMA}

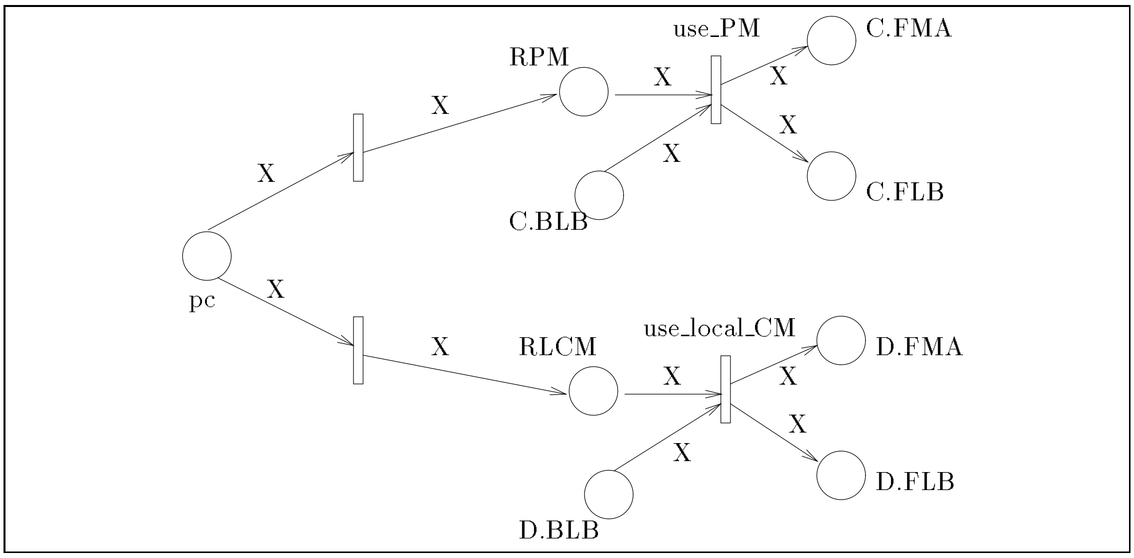


Figure 4.16: *cWN* CD, resulting from the choice composition of the *cWN*s C and D.

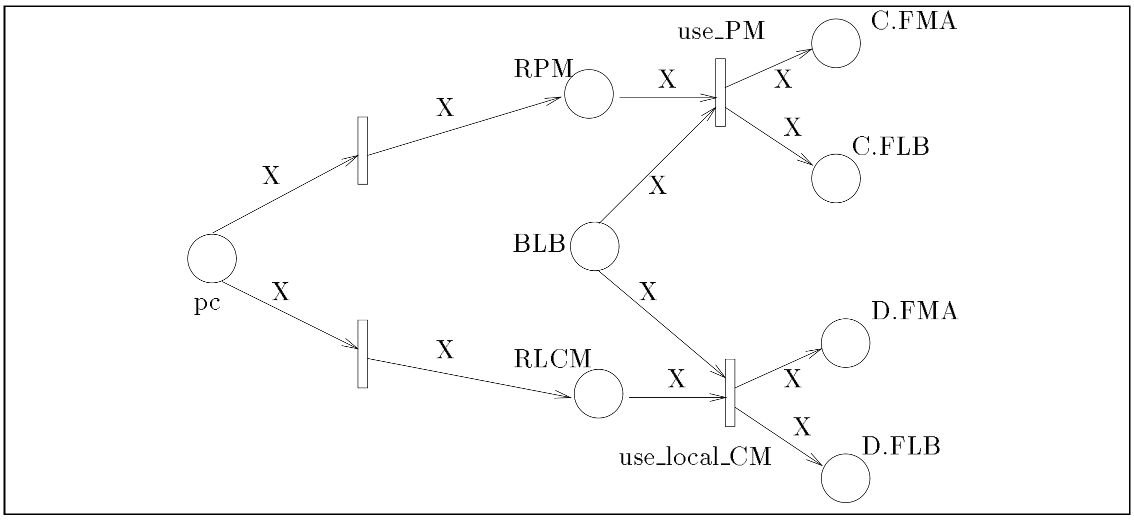


Figure 4.17: *cWN* CD′ resulting from the competing parallelism over the places C.BLB and D.BLB of component CD.

The places C.BLB and D.BLB in the *cWN* CD represent the same information. They will both contain the LBs that are in use. For this reason we apply an internal competing parallelism operation over the component, obtaining a component CD′ as shown in Figure 4.17. The interface of the *cWN* CD′ is given by:

ST :	{use_PM, use_local_CM}
ES:	{BLB, p_c}
FS:	{C.FMA, D.FMA, C.FLB, D.FLB }

There are no other internal compositions detected, therefore we continue to

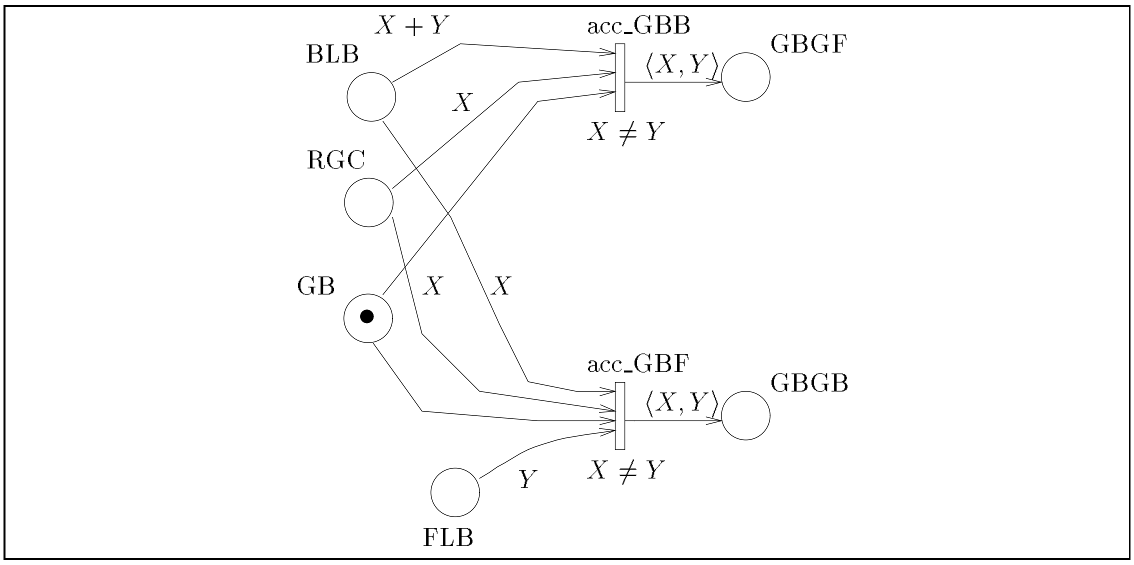


Figure 4.18: Competing parallelism of the *cWN*s E and F generate *cWN* EF.

apply the competing parallelism over the components E and F, with:

$$\Lambda(\text{E.GB}) = \text{F.GB}$$
$$\Lambda(\text{E.BLB}) = \text{F.BLB}$$
$$\Lambda(\text{E.RGC}) = \text{F.RGC}$$

We will denote the resulting *cWN* as EF (see Figure 4.18). The interface of EF is given by:

ST : {acc_GBB, acc_GBF}
ES: {GB, RGC, BLB, FLB}
FS: {GBGF, GBGB}

It is now possible to sequentially compose the *cWN*s EF with the component *G* and then with component *H* forming component EtoH(see Figure 4.19). The interface of the component EtoH, resulting from the sequential composition described, will be given by:

ST: {acc_GBB, acc_GBF, use_busy_rem, use_free_rem }
ES: { RGC, EF.BLB, EF.FLB, EF.GB}
FS: {G.GB, H.GB, G.BLB, G.FLB, H.FLB, G.FMA, H.FMA}

We can then apply the appropriate closing operations over the component EtoH (we have left the intermediate operations out) to obtain a component rem_acc as represented in Figure 4.20. The interface of this component will then be:

ST: {acc_GBF, acc_GBB use_busy_rem, use_free_rem }
ES: { RGC, BLB, FLB, GB}
FS: {G.FMA, H.FMA }

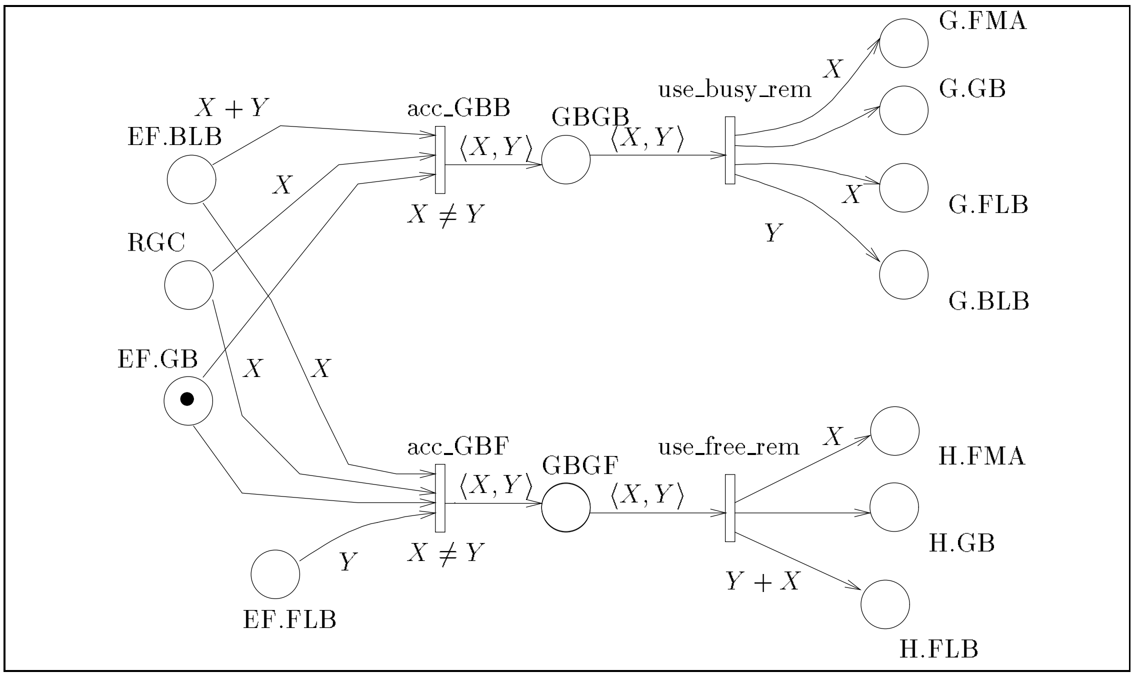


Figure 4.19: *cWN* EtoH obtained from the sequential composition of EF with the *cWN*s G and H.

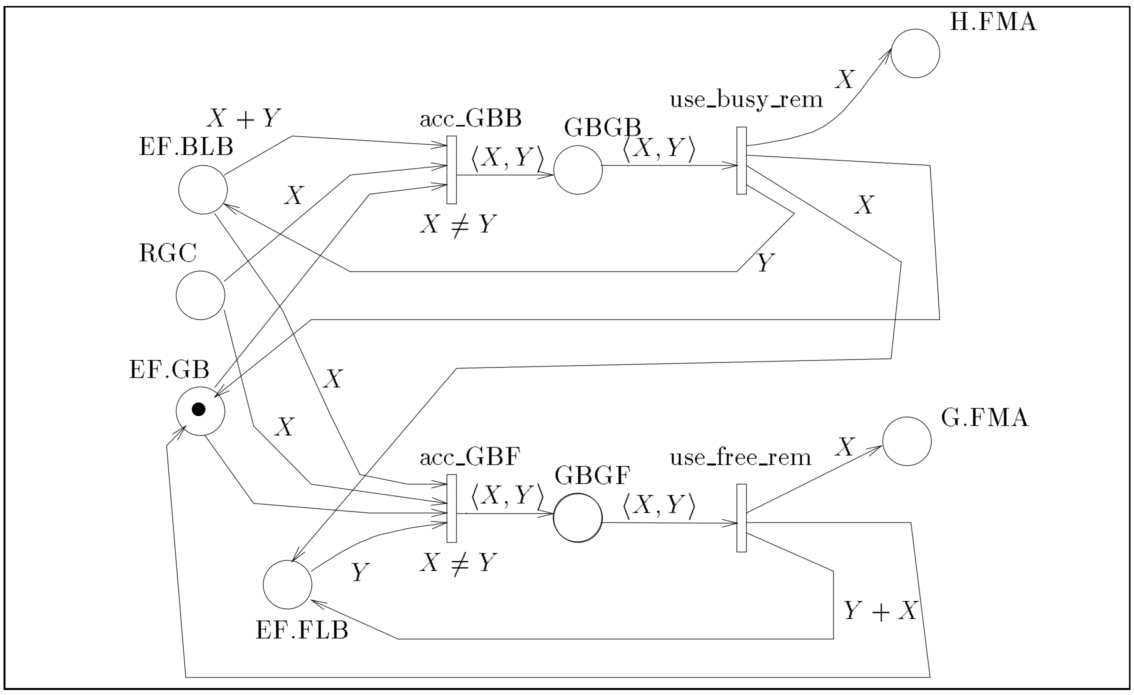


Figure 4.20: Component rem_acc resulting from recursively applying the closing operation over the *cWN* EtoH.

It is still necessary to compose components EtoH and CD′ by using a choice composition. This choice represents the random selection of whether a processor requests a local memory access or a remote memory access. The interface of the resulting component CtoH (see Figure 4.21) will be given by:

ST: {use_PM, use_local_CM, acc_GBB, ac_GBF, use_busy_rem, use_free_rem }
ES: { CD.BLB, CtoH.p_c2, EtoH.BLB, EtoH.FLB, GB}
FS: {G.FMA, H.FMA, C.FMA, D.FMA, C.FLB, D.FLB }

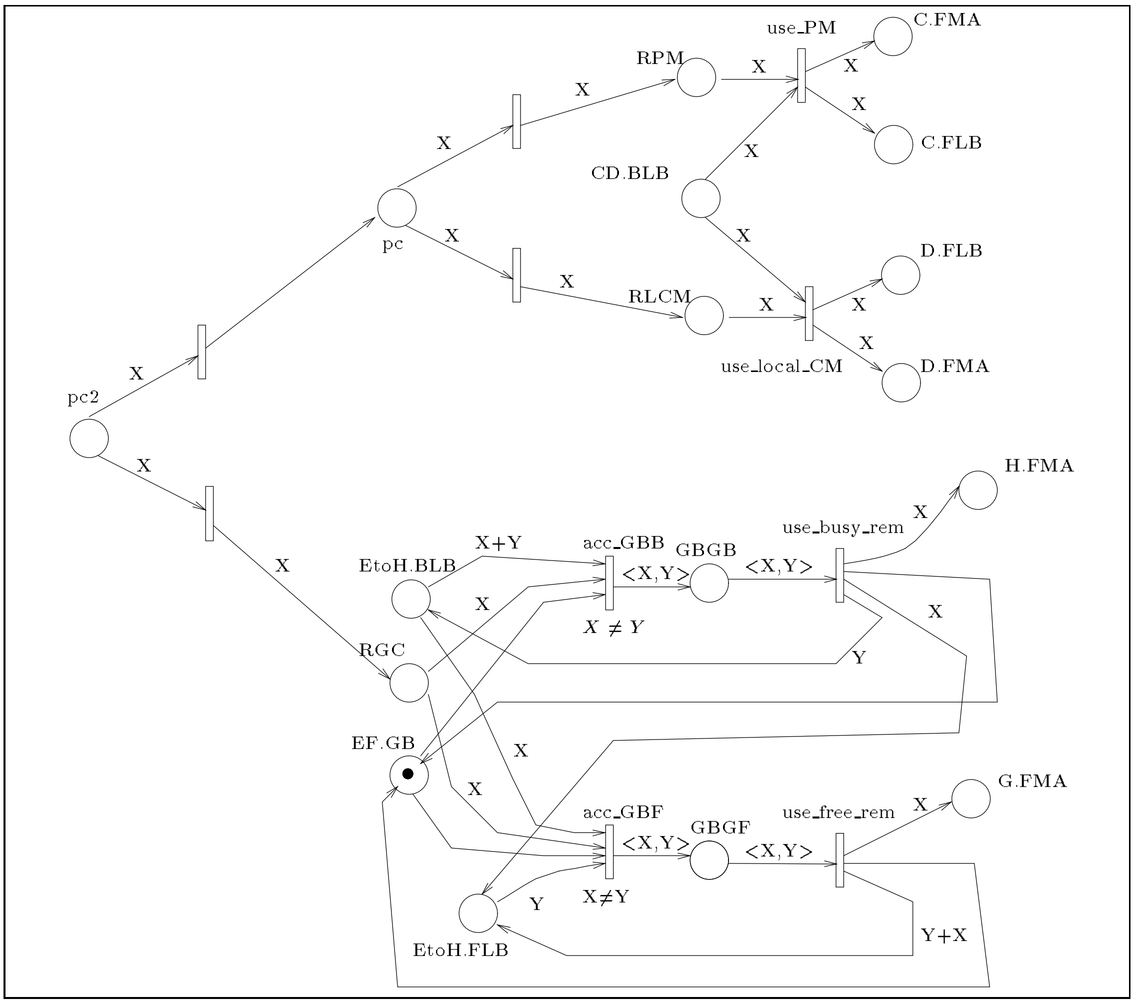


Figure 4.21: Choice composition of the *cWN*s CD′ and EtoH.

The places EtoH.BLB and CD.BLB in CtoH represent the same information, therefore it is necessary to fuse them generating the component CtoH′ with interface:

ST: {use_PM, use_local_CM, acc_GBB, ac_GBF, use_busy_rem, use_free_rem }
ES: { BLB, CtoH.p_c, EtoH.FLB, GB}
FS: {G.FMA, H.FMA, C.FMA, D.FMA, C.FLB, D.FLB }

We still need to sequentially compose the *bWN*s A and B, by the fusion of their corresponding RMA places. Given that any memory access requires the use of the processor's LB, after generating a memory access the processor tries to

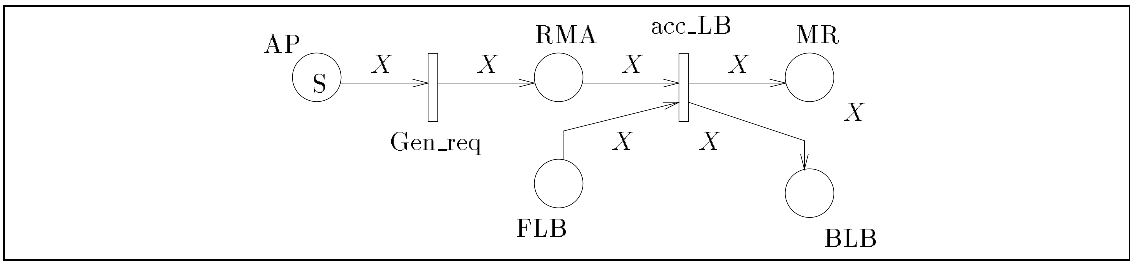


Figure 4.22: *cWN* obtained from the sequential composition of the *bWN*s A and B.

take possession of its LB. The component AB (see Figure 4.22) resulting from this composition will have as interface:

ST:	{req_mem_acc, acc_LB }
ES:	{ AP, FLB}
FS:	{BLB, MR }

It is then necessary to sequentially compose the *cWN*s AB and CtoH by fusing the place MR with CtoH.p_c2 and AB.BLB with CtoH′.BLB. The resulting component must then be sequentially composed to the *bWN* I by defining:
Γ(C.FMA)=Γ(D.FMA)=Γ(H.FMA)=Γ(G.FMa)= I.FMA
To obtain the final model of the multiprocessor architecture it is necessary to appropriately apply the closing operations over the resulting component fusing the AP places and internal competing parallelism to fuse the places AB.FLB and EtoH.FLB. The final model of the multiprocessor system is presented in Figure 4.23. The interface of this final component is given by:

ST:	{req_mem_acc, acc_LB,use_PM, use_local_CM, acc_GBF, ac_GBB, use_busy_rem, use_free_rem}
ES:	{ AP, FLB, GB}
FS:	∅

The names of the choice places have been modified to identify what they represent. The place FLB′ and BLB′ correspond to repeated occurrences of the place FLB and BLB, respectively. This has been done only to avoid crossing too many arcs in the net, making it easier to understand. The initial marking of the model will be M_0, defined as:

$$\forall p \in P,\ M_0(p) = \begin{cases} S & \text{if} p = \text{AP} \\ 0 & \text{otherwise} \end{cases}$$

where S corresponds to the set processors.

We will not go over the construction steps of the subsystem representing the interactive applications, considering that it is relatively straightforward and because of space reasons. Figure 4.24 shows the subsystem modelling the I/O

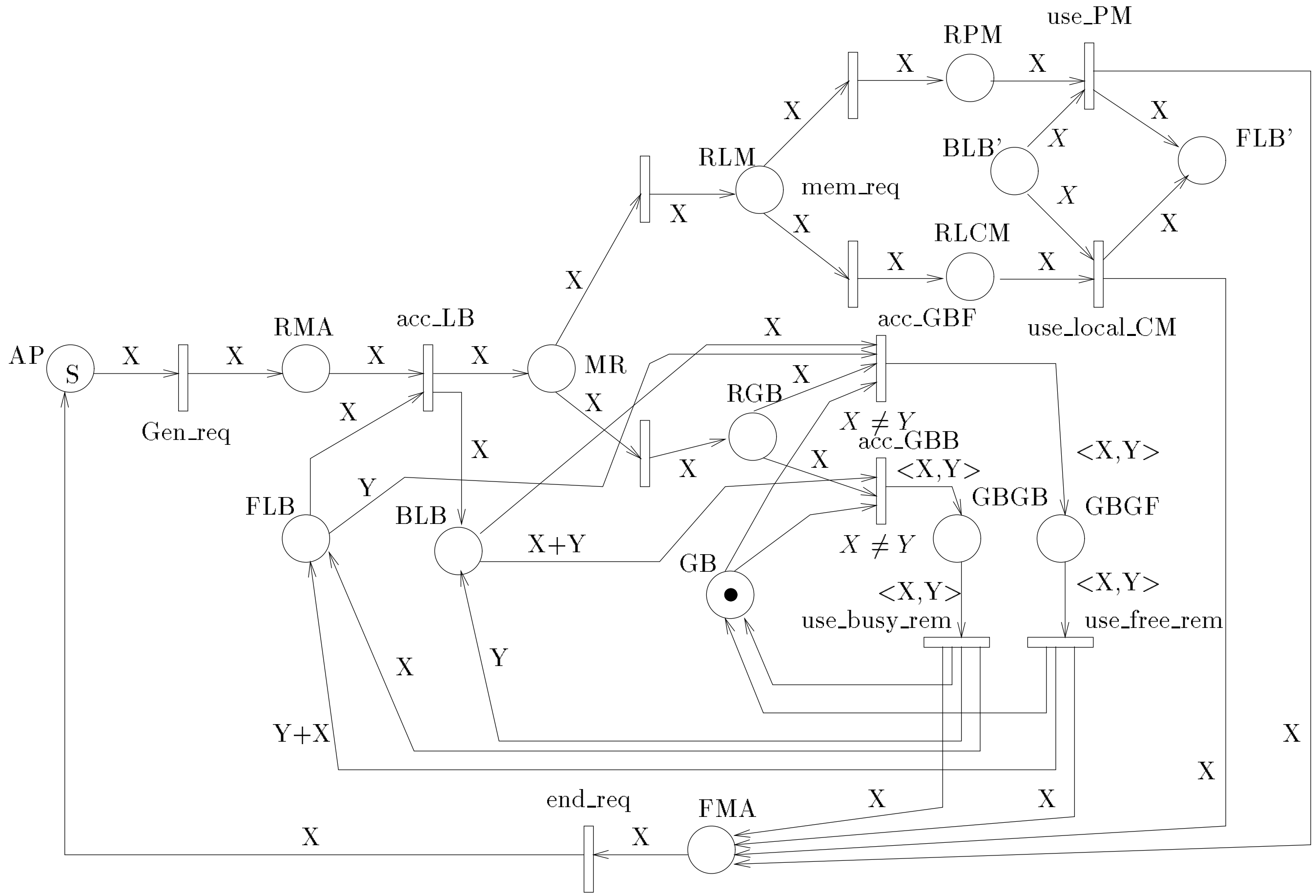


Figure 4.23: cWN of the multiprocessor architecture described.

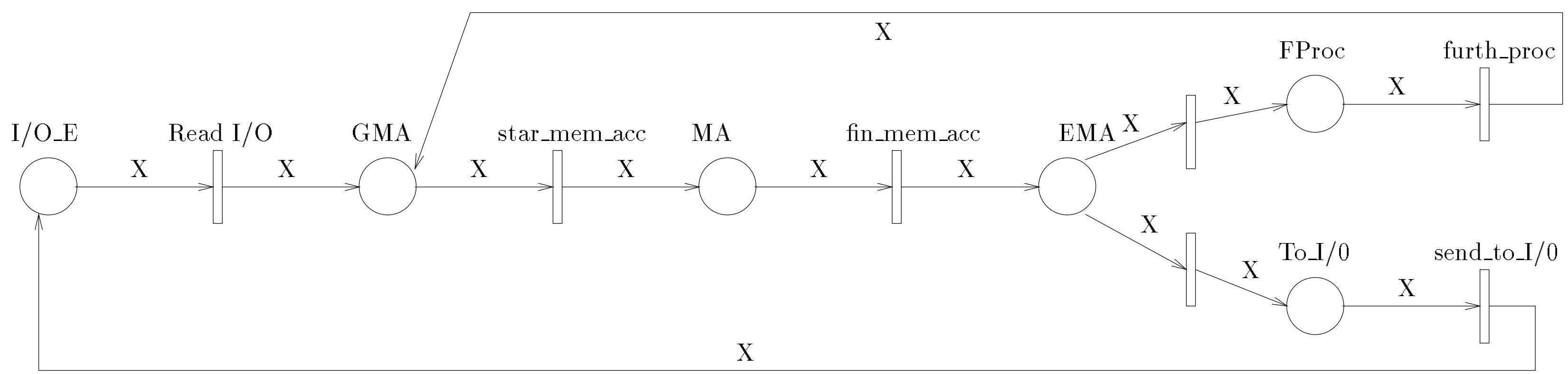


Figure 4.24: cWN of the I/O application running on each processor.

processes running in each of the processors of the multiprocessor architecture. To obtain the overall system it is necessary to synchronise the transition *Gen_req* of the subsystem representing the multiprocessor architecture (we will refer to the subsystem as subsystem 1) with the transition *start_mem_acc* of the subsystem representing the applications (we will refer to the subsystem as subsystem 2); and the transitions *End_req* of subsystem 1 and *fin_mem_acc* of subsystem 2. The first synchronisation of the subsystems is performed by applying a synchronisation operation over two components. We can then apply to the resulting component an internal synchronisation, with t_1=*End_req* and t_2 =*fin_mem_acc*. The *cWN* representing the overall system (see Figure 4.25) will have the following interface:

ST: {req_mem_acc, acc_LB,use_PM, use_local_CM, acc_GBF, ac_GBB, use_busy_rem, use_free_rem, read_I/O, send_to_I/O, furth_proc}
ES: { AP, FLB, GB, I/O_E, GMA, MA, FProc, To_I/O,}
FS: $\emptyset$

In a *cWN* there is no notion of firing priority of transitions, this implies that behaviours such us the preemption of remote accesses of LBs over accesses by the local processor cannot be appropriately represented. We have modelled it by "assigning" the remote local bus and the global bus simultaneously, using one transition.

To represent the states of a local bus (free or busy) we employed two places, one for each state. This lead to the need to divide the remote access of a CM module into two functions: one that needs to preempt because the LB is in use by its local processor, and the other that can use the remote free LB directly.

The choice operation (as it has been defined) only allows us to make choices between two components at a time (or between an even number of components when fusing choice places via a competing parallelism operation). Therefore, the selection of which type of request is generated (PM, local CM or remote CM), is made in two steps: first we distinguish between local or remote access, and then within local access we distinguish between access to the PM or the CM.

Although the model obtained can be considered as "naive" with respect to the number of transitions and places employed, it clearly reflects the functional behaviour of the system. The functions of the system are modelled as basic components. The combination of functions can be created by the adequate composition of the components reflecting these functions. Under a state-based construction approach, composition does not come about in a straightforward manner. Trying to see a component as representing a set of states or sub-states, and to then compose it with other components in order to form more complex states, does not

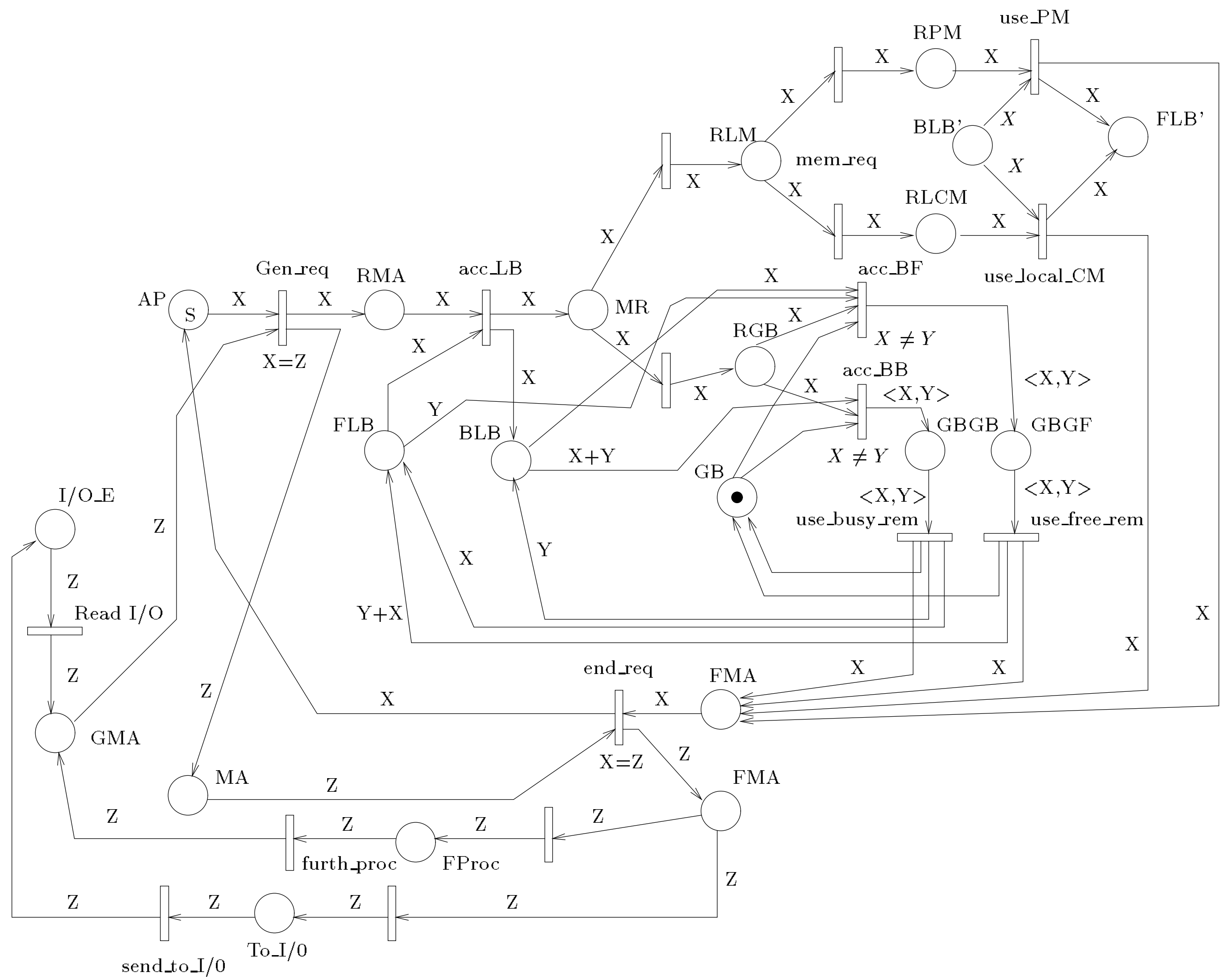


Figure 4.25: cWN of the multiprocessor system

logically or easily emerge from the analysis of a system. In general the creation of PN models depends very much on the abilities of the modeller. We propose a series of steps that guide the modeller in the construction process, in order to create a simple but understandable model.

4.5 Conclusions

In this chapter we have defined a set of operations for the compositional construction of what we have termed *Compositional WNs* (*cWN*s). The set of operations defined are based on the operators of PA, but also take into account the characteristics of the Petri net formalism. The idea is not to define a net semantics for process algebras, but to mimic the functionality of the process algebra operators to define composition operations for PNs.

We have presented a series of guidelines to help the modeller in the construction of the compositional model: from the definition of the basic WNs up to the determination of which composition operation to apply and when.

The set of composition operations has been defined in order to offer the basis for the compositional construction of WN systems, or PN systems, in general. Several restrictions have been imposed on the models that can be constructed using the operations proposed. This restrictions can be divided into *mandatory* and *convenient* constraints.

Mandatory refers to restrictions necessary in order that the definition of the composition of the operations and components can make sense:

- the fusion of places is only allowed over places with the same colour domain. This requires that the colour domain of the model has to be initially defined. To relax this condition it would be necessary to support mechanisms for the creation of new colour classes and/or static sub-classes of the existing colour classes, and to define what the fusion of places with tokens representing different objects would mean.

- the number of input and output places of a function should not be altered by the composition operation.

There are several convenient constraints, imposed to simplify the definition of the composition operations or of the resulting components.

- Inhibitor arcs are not allowed.

- No parallel arcs can be created. The creation of parallel arcs can be overcome by substituting the parallel arcs by a single arc with arc function equal to the sum of the individual arc functions.

The hiding operator in PA allows the definition of agents with activities that are only visible within the agent. The incorporation of such an operator in the method proposed would also be something desirable. It would be possible to convert synchronisable transitions into a non-synchronisable ones, eliminating them from the interface of their component, and to perform transition synchronisations not visible outside the component. We have only defined one level of synchronisable transitions. Internal synchronisation can only be performed between synchronisable transitions, which are part of the interface of the component. In this way it is not possible to reproduce the concept of hiding as such, where external synchronisation is not allowed but where internal synchronisation is still possible. The incorporation of abstraction and hiding operations into the method proposed, mimicking those of process algebras, would also represent a interesting extension to the set of operations defined. Another operation that has not been explicitly defined is that of renaming. We have implicitly used it (in a limited sense) when renaming the places with common names by prefixing the identifier of the component they originally belonged to.

The following chapters will investigate how the structural and state space information of a $cSWN$ can be obtained from the information of the lower level components, under each of the composition operations proposed.

Chapter 5

Structural Analysis of the Compositional WN models

5.1 Introduction

The structural analysis of PN models allows the investigation of properties that can be proved directly on the structure of the net, regardless of the initial marking. Any property proved structurally is valid for every possible PN system obtained from any instantiation of the initial parametric marking. Structural analysis can be investigated using either linear algebraic techniques or graph analysis techniques [AMBC+95]. We will work on the first type of techniques. Linear algebraic techniques work on a matrix description of the PN model, known as the *incidence matrix*. These methods lead to the computation of so-called *semiflows* (otherwise known as non-negative flows [CS90]) of places—*Place semiflows* (P-semiflows)—and of transitions—*Transition semiflows* (T-semiflows). P- and T-semiflows define invariants of the system. The basic idea behind place invariants is to construct equations which are satisfied for all reachable markings. The idea behind the invariant relations defined by T-semiflows is to identify firing sequences which if completed will return to the marking where they started from. The construction of place and transition invariants from P- and T- semiflows, respectively, requires the definition of an initial marking. We will limit our study to integer semiflows.

Under the framework of *cWN*s, the main idea is to use the information about the structural behaviour of sub-components to deduce the structural behaviour of the *cWN* resulting from their composition.

In [CP92], Christensen et al. propose a method for the compositional construction of place and transition invariants (P- and T- invariants) for CP-nets. This method is proposed under the framework of *Modular CP-nets*. They show that P-invariants of the modules can be combined to cover the total system if they have the same weights for the places that they share. The concept of modular CP-nets requires that all places that are fused must have the same initial marking. Furthermore, they must share the same marking: when a token is added to a place which belongs to a *fusion set* (the set of places being fused with each other) all places of the place fusion set will have the same token added. When a token is removed from a place which belongs to a place fusion set, all places of the set will have the same token removed.

Within the framework of *cWN*s it is not possible to guarantee that the initial marking of all places that will be fused is the same. It could even happen that the set of places to be fused is unknown. It is intended that the structural information of the sub-components should be obtained before the fusion of places or transitions takes place, without having to know which places or transitions will be fused (if any). For this reason it is not possible to apply the method proposed in [CP92] for the compositional deduction of structural behaviour, under the framework of *cWN*s. The method for the deduction of structural properties of the *cWN* that is proposed here, works at the level of P- and T- semiflows instead of at the level of invariants, allowing us to ignore the initial parametric marking of the *cWN*s being composed.

The rest of this chapter will be structured in the following way: in Section 5.2 we define P- and T- semiflows for WNs. The types of P- and T- semiflows defined reflect and exploit the characteristics of WNs, as CP-nets in which the symmetries between colour objects are implicitly represented. We define two types of higher level semiflows: *symbolic* P- and T- semiflows, and *static* P-semiflows. We will see why it is not possible to define static T-semiflows of *cWN*s. To conclude the section, we discuss the methods that have been proposed to obtain the *generative* and/or *significant* families of semiflows of CP-nets, pointing out which of these methods can be employed for the case of *cWN*s. Knowing that semiflows are defined based on the incidence matrix, in Section 5.3 we analyse how the incidence matrix of a higher level *cWN* can be built from the incidence matrices of its sub-components. This will be done for each compositional operation. Going on to study the compositional construction of symbolic and static P-semiflows, we begin Section 5.4 by proposing an algorithm to obtain the generative family of symbolic P-semiflows of a *bWN*. This algorithm can be adapted for the computation of the static P-semiflows of a *bWN*. We then analyse for each of the composition operations how we can construct symbolic and static P-semiflows of higher level *cWN*s from the symbolic and static P-semiflows of its sub-components. For most of the compositional operations the propositions that are introduced apply to general types of P-semiflows. This is not the case for the compositional operations involving place fusion. In this case, the method proposed is restricted to symbolic P-semiflows, and can be adjusted for the generation of static P-semiflows. We conclude Section 5.4 with an example, to apply the methods proposed. In Section 5.5 we cover the compositional construction of symbolic T-semiflows. As for the case of the P-semiflows, the methods proposed for most of the operations apply to general types of T-semiflows. However, for the case of synchronisation the method proposed only applies for symbolic T-semiflows. Finally, in Section 5.6 we present the conclusions of this chapter, and suggest future lines of work in this area.

5.2 Semiflows of WNs

The computation of semiflows of CP-nets is considered to be even more important than in ordinary Petri nets [CHP93]. This is because semiflows make it possible to verify behavioural properties without generating all the state space, which in the case of CP-nets tends to be huge.

Several types of positive flows for CP-nets have been introduced (see [GL83] and [VM84]). Many of the authors who have proposed algorithms for the calculation of flows of CP-nets have taken the most direct definition, viewing a positive flow of a coloured net as a set of positive flows of the unfolded Petri net [HC88]. However, this method is very expensive given that, in general, the size of the unfolded net is very large and the results cannot be folded or interpreted in the original CP-nets [HG87]. Furthermore, when the sizes of the colour domains are variable, it becomes impossible to unfold the net. The definition of semiflows for each of the types of CP-nets should exploit and reflect the characteristics of the type. Based on this idea we propose the definition of *symbolic semiflows* of WNs. The term symbolic flows has been used previously to represent a place flow that is defined as a function f from $Bag(C(p))$ to $Bag(D)$, such that $f \cdot W = 0$, where D is any finite set and W represents the incidence matrix of the net. The difference of the definition given in this work, is that the flows are defined at the level of *symbolic colours* and not of individual colours, assigning the same weight to all colours belonging to a symbolic colour. The invariants that can be obtained from symbolic semiflows reflect the characteristics of the system at the level of symbolic colours.

A higher level of P-semiflows can be defined for WNs to represent characteristics of coloured objects whose colour entries are defined over the same static sub-classes. This type of semiflow will be termed *static semiflows*. To formally define symbolic and static semiflows we will first introduce some necessary concepts.

Let us start by defining *symbolic colours* of a colour domain. The definition of a colour permutation will allow us to define equivalent colours (tuples) within a colour domain. As we have studied in Chapter 2, in a WN we have a set of colour classes $C = \{C_1, \ldots, C_h, C_{h+1}, \ldots, C_n\}$, such that for $i \in \{1, \ldots, h\}$, C_i is a non-ordered class, and for $i \in \{h+1, \ldots, n\}$, C_i is an ordered class. A colour class $C_i \in C$ can be partitioned into static subclasses $D_{i,q}$ such that $C_i = \cup_{q=1}^{n_i} D_{i,q}$.

Definition 5.1 (Colour Permutation (from [CDFH90])) *Consider the group of permutations $\mathcal{G} = (\xi, \cdot)$, with $\xi = \{s = \langle s_1, \cdots, s_h, s_{h+1}, \cdots, s_n \rangle\}$—where s will be also denoted as $\bigotimes_{i=1}^{n} s_i$—defined on $\bigotimes_{i=1}^{n} C_i$ such that:*

- $\forall i \in \{1, \ldots, h\}$, *$s_i$ is a permutation on C_i where $\forall D_{i,q}$, $s_i(D_{i,q}) = D_{i,q}$;*
- $\forall i \in \{h+1, \ldots, n\}$, *$s_i$ is a rotation on C_i where $\forall D_{i,q}$, $s_i(D_{i,q}) = D_{i,q}$. Note that this condition implies that if $n_i > 1$, i.e. the number of static sub-classes of C_i is greater than 1, then the only rotation allowed is the identity.*

Let $c \in C_J$, $s \in \xi$; then $s(c)$ is defined by:

$$s(\bigotimes_{i=1}^{n} \bigotimes_{j=1}^{e_i} c_i^j) = \bigotimes_{i=1}^{n} \bigotimes_{j=1}^{e_i} s_i(c_i^j)$$

Notice that for a colour domain C_J, $s(C_J) = C_J$. This follows from the fact that permutations and rotations are closed within the colour classes of a colour domain.

As a direct consequence of the previous definition we can define an equivalence relation over the colour tuples of a colour domain.

Definition 5.2 (Symbolic colour) *A* symbolic colour*, denoted $\bar{c}$, of a colour domain C_J is an equivalence class of the relation Eq defined over the colour tuples of C_J by:*

$$\forall c, c' \in C_J, \quad c \; Eq \; c' \Leftrightarrow \exists s \in \xi \text{ such that } c = s.c'$$

A symbolic colour will represent a set of colour tuples of a colour domain that have the same characteristics:

- the static sub-class associated with a colour entry i is the same for all colour tuples of the set.
- the relation between the colour entries (=,$\neq$ or u successor) is the same for all colour tuples of the set.

Before defining *symbolic semiflows*, based on the concept of *symbolic colours*, let us analyse the incidence matrix of a WN and study its characteristics.

5.2.1 The incidence matrix of a WN

In *WN* the arc functions are expressed in terms of the basic functions X_i^k, $\oplus^u X_i^k$ and $S_{i,q}$, as introduced in Chapter 2. Therefore, the entries of the incidence matrix of a *WN* can also be expressed in terms of these functions. Using the same approach as for state equivalence in WNs, as presented in [CDFH90], and given the definition of colour permutations, it is possible to define equivalence relations over the entries of the incidence matrix of a WN.

Proposition 5.1 (Equivalent entries of the incidence matrix)
Given an entry $W(p_i,t_j)(c_i,c_j)$ of the incidence matrix it holds that

$$\forall s \in \xi,\ W(p_i,t_j)(c_i,c_j) = W(p_i,t_j)(s(c_i),s(c_j))$$

Proof.- We know that the arc function $W(p_i,t_j)$ is an n-tuple [1], where each entry can be expressed in terms of the basic functions X_i^k, $\oplus^u X_i^k$ and $S_{i,q}$. An function X_i^k placed on the y entry of the arc function will return 1 only if the y^{th} entry of c_i equals the k^{th} entry c_i. If they are equal then it also holds that the s permutation of the y^{th} entry of c_i equals the s permutation of the k^{th} entry c_i. The same argument can be used for the successor function.

The function $S_{i,q}$ placed on the y^{th} entry of the tuple returns 1 for all colours of p_i whose y^{th} colour entry is a colour belonging to the static sub-class $D_{i,q}$. Given that permutations are closed within static sub-classes than all permutations of a colour satisfying this condition will also satisfy it. Therefore, the evaluation of the function $W(p_i,t_j)$ on the pair (c_i,c_j) is equal to its evaluation on the pair $(s(c_i),s(c_j))$, for all $s \in \xi$. ■

Having analysed how the symmetries between the behaviour of coloured objects in the net can be reflected in the structure of the incidence matrix of a WN,

[1] Where n will correspond to dimension of the colour domain of p_i

we now go on to define *symbolic semiflows* of an WN. The definition of this type of semiflow will exploit the symmetries represented in the incidence matrix.

5.2.2 P-semiflows of WNs

P-semiflows are non-negative left annullers of the incidence matrix. The set of P-semiflows is a vector space orthogonal to the space of rows of the incidence matrix. Therefore, P-semiflows can be generated from a basis of the space [CS90]. Associated with each P-semiflow is a P-invariant of the net, providing information on the behaviour of the net. A P-semiflow defines a weighted token count on the places of the net. In CP-nets the weights are assigned to coloured objects within the places. The weighted coloured object count defined by a P-semiflow will be invariant with respect to the firing of any transition instance $[t, c]$, with $t \in T, c \in C(t)$. No matter what the sequence of transition instances fires, the weighted coloured object count does not change, and remains the same for any marking reachable from the initial marking.

If, in a PN model all places are covered by P-semiflows then the net is *structurally bounded*. This means that for any reachable marking in any Petri net system obtained by the instantiation of the initial parametric marking, the maximum number of tokens in any place is finite.

Let us review some of the concepts related to semiflows (P- and T-) that will be used throughout the rest of this chapter.

- The *support* of a semiflow $y = \langle y_1, \dots y_k \rangle$ of dimension k is defined as :

  $$\|y\| = \langle y'_1, \dots y'_k \rangle$$

  where $i \in \{1, \dots, k\}$, $y'_i = 1$ if $y_i > 0$, or 0, otherwise.
- A semiflow is said to be *canonical* if and only if the greatest common divisor of its non-null elements is one.
- A semiflow y in the set G of P-semiflows (T-semiflows) of a PN is said to be *minimal* in G if and only if it is canonical and its support does not strictly contain the support of any other semiflow $z \in G$ ($\|z\| \not\subset \|y\|$) [Sil85].

All P-semiflows of a PN can be obtained as linear combinations of the set of minimal P-semiflows of the net. A set $F = \{f_1 \dots f_n\}$ of minimal P-semiflows of a PN is said to be *generative* if and only if $\forall g \in G,\ g \neq 0$ it holds that:

$$\exists \lambda_1, \dots \lambda_n \in \mathbb{Q}^+ \text{ such that } g = \sum_{i=1}^{n} \lambda_i \cdot f_i$$

Similarly we can define a generative set of T-semiflows.

To start analysing the types of P-semiflows of a *cWN*, let us define a P-semiflow of a *cWN* based on the unfolded incidence matrix of the WN.

Definition 5.3 (P-semiflow of a WN) *A P-semiflow v of a WN is a vector of cardinality $|P|$ of the form $v = \langle v_1, \cdots, v_{|P|}\rangle$, that satisfies:*

$$v \cdot W = 0$$

where $\forall i \in \{1, \ldots, |P|\}$, $v_i \in (\mathbb{N}^+)^{|C(p_i)|}$. Every component-vector v_i can be expressed as a tuple

$$\bigotimes_{r=1}^{|C(p_i)|} \delta_r$$

where $\delta_r = v_i(c_r) \in \mathbb{N}^+$.

In *WN*s we want to be able to maintain the parametrisation of the net for the construction of its semiflows. Having just parameters, it is not possible to know the cardinality of the colour domain of a place or a transition. Therefore, it is necessary to find a method that does not need the cardinality of the colour domains to obtain P-semiflows of the net, or that can leave the P-semiflows expressed in terms of the parameters of the model.

The calculation of P-semiflows that assign weights to each colour tuple of the colour domain of each place leads to the definition of P-invariants over coloured objects of particular colours. This does not reflect the symmetries in the behaviour of the coloured objects in the system. Moreover, the number of invariants obtained makes the deduction of overall properties of the system more difficult.

We will show that higher level P-semiflows, that work at the level of symbolic colours can be defined. This reduces the number of P-invariants that we need to handle, exploits the symmetries amongst the coloured objects, and allows us to determine the boundedness of the net at the level of symbolic colours.

5.2.2.1 Symbolic P-semiflows

When analysing the incidence matrix of a WN, we observed that the entries $W(p, \cdot)$ for colour tuples belonging to the same symbolic colour are equivalent. Given a P-semiflow v of a WN, all its permutations (defined as follows) can be proven to be P-semiflows of the WN.

Definition 5.4 (Permutation of a P-semiflow)
A permutation of a P-semiflow *v, denoted by $s \cdot v$, is defined as:*

$$s \cdot v = \langle s \cdot v_1, \ldots, s \cdot v_{|P|}\rangle$$

where $s \in \xi$. *For each* $i \in \{1, \ldots, |P|\}$

$$s \cdot v_i = \bigotimes_{r=1}^{|C(p_i)|} v_i(s(c_r)) = \bigotimes_{r=1}^{|C(p_i)|} v_i(\bigotimes_{k=1}^{n} \bigotimes_{j=1}^{e_k^i} s_k(c_{r,k}^j))$$

where e_k^i *is the number of occurrences of the colour class* k *in the colour domain of the place* p_i*, and* $c_{r,k}^j \in C_k$ *is the colour of the* j^{th} *occurrence of* C_k *in* c_r.

Proposition 5.2 *Given* v*, a P-semiflow of a WN* $\mathcal{N}$*, a permutation* $s \cdot v$ *is also a P-semiflow of* $\mathcal{N}$.

Proof.- By definition of vector-matrix multiplication

$$(s \cdot v) \cdot W = \bigotimes_{t \in T} (s \cdot v) \cdot W(\cdot, t)$$

For $(s \cdot v) \cdot W = 0$ it must hold that for every $t \in T$ and every instance $[t, c']$ with $c' \in C(t)$,

$$(s \cdot v) \cdot W(\cdot, t)(\cdot, c') = 0$$

So let us take any transition instance $[t, c']$ of a transition $t \in T$. It will then hold that:

$$\begin{aligned}(s \cdot v) \cdot W(\cdot, t)(\cdot, c') &= \sum_{p \in P} (s \cdot v_p) \cdot W(p, t)(\cdot, c') \\ &= \sum_{p \in P} \sum_{c \in C(p)} (s \cdot v_p(c)) \cdot W(p, t)(c, c')\end{aligned}$$

From the definition of equivalent entries of the incidence matrix (Proposition 5.1) we know that for any entry $W(p, t)(c, c')$ a permutation $s \cdot W(p, t)(c, c')$ is an entry of the incidence matrix with the same value. Substituting by the appropriate equivalent entry of W in the previous equation, we obtain:

$$\begin{aligned}(s \cdot v) \cdot W(\cdot, t)(\cdot, c') &= \sum_{p \in P} \sum_{c \in C(p)} (s \cdot v_p(c)) \cdot (s \cdot W(p, t)(c, c')) \\ &= \sum_{p \in P} \sum_{c \in C(p)} s \cdot (v_p(c) \cdot W(p, t)(c, c')) = s \cdot (v \cdot W(\cdot, t)(\cdot, c'))\end{aligned}$$

Since $v \cdot W(\cdot, t)(\cdot, c') = 0$ then $s \cdot (v \cdot W(\cdot, t)(\cdot, c')) = 0$, which allows us to conclude that $s \cdot v$ is also a P-semiflow of $\mathcal{N}$. ■

The normalised sum[2] of all the permutations of a P-semiflow forms a P-semiflow that assigns the same value to all the colour tuples belonging to a

[2]The normalised sum of a set of vectors V is the vector obtained by adding all vectors in V and dividing the resulting vector by the greatest common divisor (g.c.d.) of all its terms. The final vector will have terms with 1 as their g.c.d..

symbolic colour. P-semiflows obtained in this way can be represented in terms of the basic arc functions. This type of P-semiflow will be known as a *symbolic P-semiflow*.

With the use of the basic arc functions $S_{i,q}$, X_i^k and $\oplus^u X_i^k$, it is possible to assign any integer value to a symbolic colour of the colour domain of a place p. This can be done using a function $\widetilde{C}_p : C(p) \rightarrow \mathbb{N}^+$ defined in the following manner:

$$\widetilde{C}_p = \bigotimes_{i=1}^{h} \bigotimes_{j=1}^{e_i} \left(\sum_{q=1}^{n_i} \alpha_{i,q}^j \cdot S_{i,q} + \sum_{k \in \{1,\ldots,(j-1)\}} \beta_{i,j}^k \cdot X_i^k \right) \times$$
$$\bigotimes_{i=h+1}^{n} \bigotimes_{j=1}^{e_i} \left(\sum_{q=1}^{n_i} \alpha_{i,q}^j \cdot S_{i,q} + \sum_{k \in \{1,\ldots,(j-1)\}} \left(\beta_{i,j}^k \cdot X_i^k + \sum_{u \in \{1,\ldots,|C_i|\}} \gamma_{i,j,u}^k \cdot \oplus^u X_i^k \right) \right)$$

where $\alpha_{i,q}^j, \in \mathbb{N}^+$ and $\beta_{i,j}^k,\ \gamma_{i,j,u}^k \in \mathbb{Z}$.

The function $\alpha_{i,q}^j \cdot S_{i,q}$ applied over c_i^j, the j^{th} occurrence of C_i in the colour $c \in C(p)$, returns $\alpha_{i,q}^j$ if $c_i^j \in D_{i,q}$; $\beta_{i,j}^k \cdot X_i^k$ returns $\beta_{i,j}^k$ if c_i^j equals the colour of the k^{th} occurrence, and $\gamma_{i,j,u}^k \cdot \oplus^u X_i^k$ returns $\gamma_{i,j,u}^k$ if c_i^j equals the u successor of the colour of the k^{th} occurrence.

In the first occurrence of each colour class it is only possible to distinguish between the colours belonging to different static sub-classes. This is to represent that the first element of a symbolic colour can take any value within the static sub-class it belongs to.

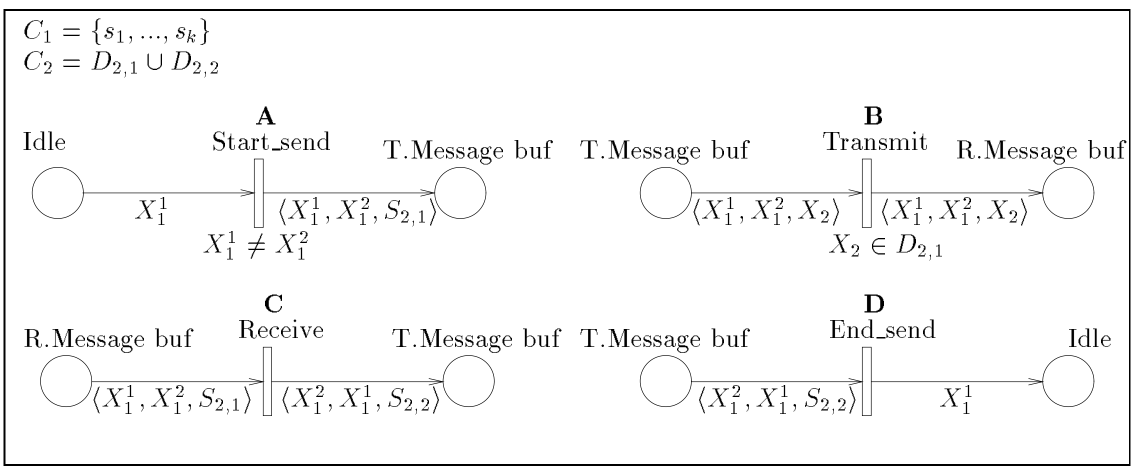


Figure 5.1: *bWN*s of the basic functions of the message communication system example

Table 5.1 presents the set of minimal symbolic P-semiflows (as defined above) of the *bWN*s of a message communication system (see Figure 5.1), taken from [CDF91], omitting the trivial P-semiflow $\vec{0}$. Notice that the symbolic P-semiflows are left expressed in terms of the cardinalities of the static sub-classes. For example, the tuple $\langle S_{1,1}, X_1^1, |D_{2,2}|S_{2,1}\rangle$ in the first symbolic P-semiflow of C, means that the symbolic P-semiflow assigns a weight of $|D_{2,2}|$ to all colour tuples of place *T.Message buf* whose first and second colour entries are equal, and whose third colour entry belongs to the static sub-class $S_{2,1}$.

Minimal Symbolic P-semiflows of **A**	
Idle	T.Message buf
$\langle \lvert D_{2,1}\rvert \cdot S_{1,1}\rangle$	$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1}\rangle$
0	$\langle S_{1,1}, X_1^1, S_{2,1}\rangle$
0	$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2}\rangle$
0	$\langle S_{1,1}, X_1^1, S_{2,2}\rangle$

Minimal Symbolic P-semiflows of **B**	
R.Message buf	T.Message buf
$\langle S_{1,1}, X_1^1, S_{2,1}\rangle$	$\langle S_{1,1}, X_1^1, S_{2,1}\rangle$
$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1}\rangle$	$\langle S_{1,1},, S_{1,1} - X_1^1, S_{2,1}\rangle$
$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2}\rangle$	0
$\langle S_{1,1}, X_1^1, S_{2,2}\rangle$	0
0	$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2}\rangle$
0	$\langle S_{1,1}, X_1^1, S_{2,2}\rangle$

Minimal Symbolic P-semiflows of **C**	
T.Message buf	R.Message buf
$\langle S_{1,1}, X_1^1, \lvert D_{2,2}\rvert \cdot S_{2,1}\rangle$	$\langle S_{1,1}, X_1^1, \lvert D_{2,1}\rvert \cdot S_{2,2}\rangle$
$\langle S_{1,1}, S_{1,1} - X_1^1, \lvert D_{2,2}\rvert \cdot S_{2,1}\rangle$	$\langle S_{1,1}, S_{1,1} - X_1^1, \lvert D_{2,1}\rvert \cdot S_{2,2}\rangle$
$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2}\rangle$	0
$\langle S_{1,1}, X_1^1, S_{2,2}\rangle$	0
0	$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1}\rangle$
0	$\langle S_{1,1}, X_1^1, S_{2,1}\rangle$

Minimal Symbolic P-semiflows of **D**	
T.Message buf	Idle
$\langle S_{1,1}, S_{1,1}, S_{2,2}\rangle$	$\langle \lvert D_{2,1}\rvert \cdot S_{1,1}\rangle$
$\langle S_{1,1}, X_1^1, S_{2,1}\rangle$	0
$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1}\rangle$	0

Table 5.1: Symbolic P-semiflows of the *bWN*s **A**,**B**,**C** and **D**, modelling the basic functions of a message communication system.

A symbolic P-semiflow can assign a non-zero weight to more than one symbolic colour of a given place. For example, in *bWN* **D**, the first symbolic P-semiflow assigns 1 to all symbolic colours of *T.Message buf* that have as third entry a colour belonging to the static sub-class $D_{2,2}$. This is caused by the fact that for all valid instances of the transition *End_send* the firing of the transition instance will place one coloured object in place *Idle*. However, the set of instances of *End_send* can be divided into those that extract coloured objects of the symbolic colour class $\langle S_{1,1}, X_1^1, S_{2,2}\rangle$ of *T.Message buf* and those that extract coloured objects of the symbolic colour $\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2}\rangle$. Thus, to be able to annul all entries associated with the transition it is necessary to cover all the symbolic colours of *T.Message buf* from which the transition *End_send* selects coloured objects.

An even higher level of P-semiflows can be defined for WNs. They define invariants over coloured objects with colours that satisfy the following condition: $\forall i \in \{1,\dots,n\}$, $\forall j \in \{1,\dots,e_i\}$, c_i^j (the entry of j^{th} occurrence of colour class i) belongs to a certain static sub-class of C_i. This type of P-semiflow will be called *static P-semiflows*, based on the concept of *static sub-domains*.

5.2.2.2 Static P-semiflows

Another way of grouping the colour tuples of a colour domain, is by considering to which static sub-class the colour entries of the colour tuples belong. Let us define a *static colour sub-domain*.

Definition 5.5 (Static colour sub-domain)
Given a colour domain $C_J = \bigotimes_{i=1}^{n}(C_i)^{e_i}$ a static sub-domain *C_J^g is defined as:*

$$C_J^g = \bigotimes_{i=1}^{n}\bigotimes_{j=1}^{e_i} \varrho_{j,i}^g = \bigotimes_{i=1}^{n}\bigotimes_{j=1}^{e_i}\sum_{q=1}^{n_i} \varrho_{j,i,q}^g D_{i,q}$$

where $\varrho_{j,i,q}^g \in \{0,1\}$ and $\sum_{q=1}^{n_i} \varrho_{j,i,q}^g = 1$. The value of $\varrho_{j,i,q}^g$ is determined by which static sub-class of C_i is to be considered for the j^{th} occurrence of colour class C_i in the static sub-domain C_J^g. Only one static sub-class can be considered for every occurrence of a colour class.

Notice that this definition is similar to that of the colour domain of transitions expressed in terms of static sub-classes as defined in Section 8.2.3 on page 208.

For example, given the basic colour classes C_1 and $C_2 = D_{2,1} \cup D_{2,2}$, and the colour domain $C_J = C_1 \times C_2 \times C_2$ the set of all possible static colour sub-domains of C_J would be:

$$C_1 \times D_{2,1} \times D_{2,1};\quad C_1 \times D_{2,1} \times D_{2,2};\quad C_1 \times D_{2,2} \times D_{2,1};\quad C_1 \times D_{2,2} \times D_{2,2}$$

A static sub-domain is formed by a set of symbolic colours.

We are interested in determining the relation between the behaviour of the coloured objects with colours in the static sub-domain. All valid transition instances will select the same number of coloured objects of a static sub-domain. Consider the *bWN* presented in Figure 5.2. Notice that regardless of the instantiation of X_1^1

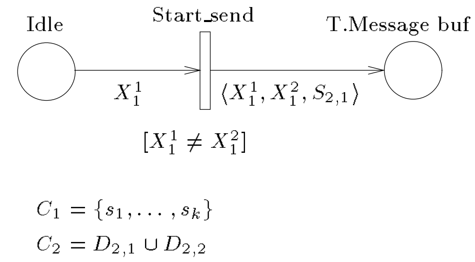


Figure 5.2: Example to show the number of coloured objects placed or extracted from a place by the firing of a transition instance.

the firing of the transition *Start_send* will always extract one coloured object from the place *Idle*. The colour domain of *Idle* has only one static sub-domain (namely itself). Therefore it holds that the number of coloured objects—with colours in the static sub-domain— extracted from *Idle* is equal for all valid instantiations of X_1^1. Given an instantiation of X_1^1, for every valid instantiation of X_1^2, the firing of the corresponding instance of *Start_send* will always place $|D_{2,1}|$ coloured objects in the place *T.Message buf*. The elements placed in *T.Message buf* will always belong to the static sub-domain $C_1 \times C_1 \times D_{2,1}$. For the other static sub-domain of *T.Message buf* (namely $C_1 \times C_1 \times D_{2,2}$) the number of coloured objects placed in *T.Message buf* will be zero for any valid instantiation of X_1^1 and X_1^2.

Proposition 5.3 *Suppose a place p and a transition t such that $p \in {}^\bullet t \cup t^\bullet$. The number of coloured objects, corresponding to a static sub-domain of a place p, that are selected by a transition instance $[t, c']$, is equal for all permutations of the transition instance.*

Proof.- From the definition of the incidence matrix we know that for an entry $W(p,t)(c,c')$ we can find an equivalent entry $s \cdot W(p,t)(c,c')$ with the same value, where $s \in \xi$. So, given a static sub-domain C_J^g, we then have that:

$$\sum_{c \in C_J^g} W(p,t)(c,c') = \sum_{c \in C_J^g} s \cdot W(p,t)(c,c')$$

From the definition of permutations of entries of the incidence matrix, and knowing that all permutations of a colour in a static sub-domain are colours of the same static sub-domain, we obtain:

$$\sum_{c \in C_J^g} s \cdot W(p,t)(c,c') = \sum_{c \in C_J^g} W(p,t)(s \cdot c, s \cdot c') = \sum_{c \in C_J^g} W(p,t)(c, s \cdot c')$$

∎

Given this proposition let us define a *static* P-semiflow.

Definition 5.6 (Static P-semiflow) *Consider $C = \{C_1, \ldots, C_n\}$ the set of basic colour classes of a WN $\mathcal{N}$, W the incidence matrix of $\mathcal{N}$, and the function $S_{i,q}$ over the colours of C_i, that returns 1 if the colour is in the static sub-class $D_{i,q}$ of C_i or 0 otherwise. A* static P-semiflow *of $\mathcal{N}$ is a vector $v = \langle v_1, \ldots, v_{|P|}\rangle$ where for $k \in \{1, \ldots, |P|\}$:*

$$v_k = \bigotimes_{i=1}^{n} \bigotimes_{j=1}^{e_i} \left(\sum_{q=1}^{n_i} \alpha^k_{i,j,q} \cdot S_{i,q} \right)$$

with $\alpha^k_{i,j,q} \in \mathbb{N}^+$.

The function $\alpha^k_{i,j,q} \cdot S_{i,q}$ returns $\alpha^k_{i,j,q}$ for all colours in $C(p_k)$ whose j^{th} entry of the colour class C_i is an colour in the static sub-class $D_{i,q,}$; otherwise it returns zero.

The P-invariants that can be deduced from static P-semiflows will establish that the weighted count of all objects with colours in the static sub-domain determined for each place will be constant regardless of the transition instances fired. The boundedness of the net will then be defined at the level of static sub-domains.

5.2.3 T-semiflows of WNs

T-semiflows are non-negative right annullers of the incidence matrix. The set of T-semiflows is a vector space orthogonal to the space of columns of the incidence matrix. Therefore, T-semiflows can be generated from a basis of the space. Associated with each T-semiflow is a T-invariant of the net, that defines the characteristics of the firing sequences that preserve the markings of the system. Leading to the formal definition of T-semiflows of CP-nets we will explain some concepts related to them.

Definition 5.7 (Transition count vector (adapted from [AMBC⁺95])) *Given a sequence of transition instances $\sigma = [t_1, c_1], \ldots, [t_k, c_k]$ where for $i \in \{1, \ldots, k\}$, $c_i \in C(t_i)$, a* transition count vector *x_σ of σ, is an integer vector whose i^{th} entry indicates how many times a transition instance $[t_i, c_i]$ appears in σ.*

A marking M' produced by firing the transition sequence σ from a marking M can be obtained using the following equation:

$$M' = M + W \cdot x_\sigma$$

Only the number of times that a transition instance fires is considered. The order in which the transition instances are fired is important for the definition of the transition sequence, and for checking whether a sequence can be fired, but it plays no role in the computation of the marking reached by the sequence [AMBC+95].

If $W \cdot x_\sigma = 0$, then $M' = M$, thus the firing sequence σ brings the CP-net back to the marking M.

Definition 5.8 (T-semiflow of a CP-net) *A T-semiflow of a CP-net with incidence matrix W, is a vector of cardinality $|T|$ of the form $x = \langle x_1, \ldots, x_{|T|}\rangle$, that satisfies:*

$$W \cdot x = 0$$

where $\forall i \in \{1, \ldots, |T|\}$, $x_i \in (\mathbb{N}^+)^{|C(t_i)|}$. Every component vector $x_i \in (\mathbb{N}^+)^{|C(t_i)|}$ can be expressed as a tuple $\bigotimes_{r=1}^{|C(t_i)|} \omega_r$, where $\omega_r = x_i(c_r) \in \mathbb{N}^+$, with $c_r \in C(t_i)$.

This identifies an invariant relation that states that by firing from marking M any sequence σ of transition instances, whose count vector is a T-semiflow, the marking obtained at the end of the sequence is equal to the starting one, provided that σ is fire-able from M. The covering of all transitions by T-semiflows is a necessary but not sufficient condition for liveness of bounded nets. If at least one transition is not included in a T-semiflow, while the net is covered by P-semiflows, the net is not live. Based on the unfolded representation of the incidence matrix of a *cWN*, a T-semiflow of a *cWN* is defined in the same way as a T-semiflow of a CP-net.

Given the concept of symbolic colours, we can define an equivalence relation at the level of transition instances. An equivalence class of this relation will be termed a *canonical transition instance*. This is based on the fact that the representation of symbolic colours in terms of the basic predicates, as described when defining symbolic P-semiflows, is canonical. The term symbolic transition instance has already being introduced in [CDFH91] to describe transition instances where object identifiers are replaced by variables.

Definition 5.9 (Canonical transition instance) *A* canonical transition instance *of a transition t with colour domain $C(t)$ is an equivalence class of the relation Eq_t, defined over the set of transition instances of t, as:*

$$\forall c, c' \in C(t), \;\; [t, c] \; Eq_t \; [t, c'] \Leftrightarrow \exists s \in \xi \textit{ such that } s \cdot [t, c] = [t, c']$$

A permutation of a transition instance is obtained by permuting the colour tuples of the instance.

From the equivalence relation defined over the entries of the incidence matrix (see Definition 5.1) we can observe that transition instances within a canonical transition instance have equivalent behaviours in the net. This equivalence is defined with respect to the type and number of coloured objects that they extract from, or place into, the input and output places of the transition.

Taking the same approach as for symbolic P-semiflows, it is possible to define T-semiflows that do not distinguish between transition instances belonging to the same canonical transition instance. This type of T-semiflow assigns weights to canonical transition instances. Symbolic T-semiflows will define invariants that establish that, by firing a sequence of canonical transition instances whose count corresponds to a symbolic T-semiflow we can return to the symbolic marking from which the sequence started. This is provided that the sequence is fire-able.

As for P-semiflows, let us prove that permutations of T-semiflows are also T-semiflows of the WN.

Proposition 5.4 *Given x, a T-semiflow of a cWN $\mathcal{N}$, a permutation $s \cdot x$ is also a T-semiflow of $\mathcal{N}$.*

Proof.-

$$W \cdot (s \cdot x) = \bigotimes_{p \in P} W(p, \cdot) \cdot (s \cdot x)$$

For $W \cdot (s \cdot x) = 0$ it must hold that $\forall p \in P$, $\forall c \in C(p)$,

$$W(p, \cdot)(c, \cdot) \cdot (s \cdot x) = 0$$

So let us take any $p \in P$ and colour $c \in C(p)$; it will hold that:

$$\begin{aligned} W(p, \cdot)(c, \cdot) \cdot (s \cdot x) &= \sum_{t \in T} W(p, t)(c, \cdot) \cdot (s \cdot x_t) \\ &= \sum_{t \in T} \sum_{c' \in C(t)} W(p, t)(c, c') \cdot (s \cdot x_t(c')) \end{aligned}$$

From the definition of equivalent entries of the incidence matrix W(Proposition 5.1), we know that for any entry $W(p, t)(c, c')$ a permutation $s \cdot W(p, t)(c, c')$ with $s \in \xi$, is an entry of the incidence matrix with the same value. Substituting in the previous equation, we obtain:

$$\begin{aligned} W(p, \cdot)(c, \cdot) \cdot (s \cdot x) &= \sum_{t \in T} \sum_{c' \in C(t)} (s \cdot W(p, t)(c, c')) \cdot (s \cdot x_t(c')) \\ &= \sum_{t \in T} \sum_{c' \in C(t)} s \cdot (W(p, t)(c, c') \cdot x_t(c')) = s \cdot (W(p, \cdot)(c, \cdot) \cdot x) \end{aligned}$$

Since $W(p,\cdot)(c,\cdot)\cdot x = 0$ then $s\cdot(W(p,\cdot)(c,\cdot)\cdot x) = 0$, which allows us to conclude that $s \cdot x$ is also a T-semiflow of $\mathcal{N}$. ■

Following the pattern of P-semiflows, the normalised sum of all the permutations of a T-semiflow forms a T-semiflow that assigns the same value to all the transition instances of a canonical transition instance. The resulting *symbolic T-semiflows* can be represented in terms of the basic arc functions, as for the case of symbolic P-semiflows. A minimal symbolic T-semiflow can cover more than one canonical transition instance of a transition. In Figure 5.3, we present an illustrative example to show a case where this happens. For it to be possible to return the marking of the place p_1 after the firing of a transition instance of t_1, transition t_2 must fire instances of both of its symbolic colours ($\langle S_{1,1}, X_1^1\rangle$ and $\langle S_{1,1}, S_{1,1} - X_1^1\rangle$).

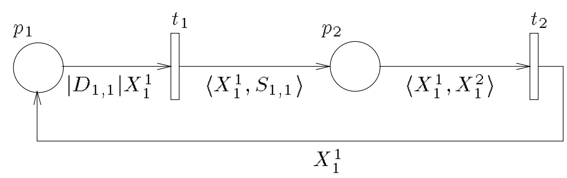


Figure 5.3: Illustrative example to present symbolic T-semiflows that cover more than one symbolic colour per transition.

Contrary to P-semiflows, it is not possible to define T-semiflows at the level of static sub-domains. For transition instances it does not always hold that, for colours belonging to a colour sub-domain the corresponding transition instances will all extract (place) the same number of coloured objects from (into) the input (output) places of the transition. Within the colour domain of the transition there can be valid and invalid instances, according to the transitions predicates. Invalid instances of a transition will have a zero column vector in the corresponding entries of the incidence matrix. Vectors that cover only invalid instances of the transitions of a *cWN* will be T-semiflows of the net. Given that these instances can never fire, it will always be the case that we stay in the same marking.

5.2.4 Obtaining semiflows of *cWN*s

The difficulty of finding semiflows for CP-nets comes down to the fact that the coefficients $W(p,t)$ of the incidence matrix are not simple integers, but colour functions [Cou90]. Several works have focussed on finding a general solution for some classes of coloured nets: Regular nets [HG87], Pr/T-systems without formulas in the transitions [Vau87], associative nets and ordered nets [HC88], commutative nets [Cou90], unary regular nets and unary P/T nets [CHP93], amongst others.

In [Cou90] Couvreur proposes a method to obtain the generative family of flows of a coloured net based on the use of the "generalised semi-inverse". This method is applicable in non-parametrised Petri nets. He also proposes solutions for calculating flows for coloured nets based on polynomial rings. The complexity of polynomial algebra does not allow both the computing of a generative family of flows and parametrisation in the general case of the complete structure.

In [HC88], Haddad and Couvreur propose three kinds of solutions that may be produced under certain restrictions:

1. With an arbitrary structure but with no parametrisation it is possible to obtain a generative family.

2. With an arbitrary structure and allowing parametrisation it is possible to obtain not a generative family, but a *significant* family.

3. By incorporating restrictions on structure it is possible to obtain a generative family, even when allowing parametrisation.

Here the term *significant* refers to a set that although it is not a generator set, i.e. it is not possible to obtain every semiflow of the net from the set, it allows us to deduce most of the structural information that can be deduced when obtaining the generative family of semiflows.

In *cWN*s we want to be able to maintain the parametrisation of the net for the construction of the semiflows, so the first option does not apply. Algorithms to obtain parametrised generative families of flows have been proposed but only for limited subclasses of CP-nets. Given the fact that *WN*s can represent any CP-net, then we can say that there is no limit in the structure of the CP-net. Although *cWN*s impose certain structural restrictions on the model—no inhibitor arcs are allowed and immediate transitions only appear as part of a choice composition—they do not impose any conditions on the colour domains that can be defined. This means that it is only possible to obtain *significant* families of flows (possibly not generative) of *cWN*s using these algorithms.

Added to the problem of finding semiflows for *cWN*s, it is necessary to offer the mechanisms for the composition of semiflows to form semiflows of higher level components.

In the following sections we propose a method for the compositional construction of symbolic P- and T-semiflows of *cWN*s. The objective of this work is not to offer an alternative method for the construction of semiflows of *WN*s, given that the method proposed is tightly coupled to the compositional operations defined and the general framework of *cWN*s.

In order to study how the P- and T- semiflows of sub-components can be used to form the semiflows of a higher level component, we will first study how the incidence matrix of *cWN*s can be formed from the incidence matrices of its sub-components.

5.3 Compositional construction of the incidence matrix of a *cWN*

The incidence matrix of a *cWN* can be obtained from the incidence matrix of its sub-component(s), knowing which composition operation is applied. Let us analyse how this is done.

5.3.1 Under independent parallel composition

Consider the composition of two *cWN*s, namely L and R, by a parallel composition operation. We will denote by W_L and W_R the incidence matrices of L and R, respectively. Given that $P_{\mathcal{N}} = P_L \cup P_R$ and $T_{\mathcal{N}} = T_L \cup T_R$, i.e. there is no place or transition fusion between the components, the incidence matrix of the resulting *cWN* $\mathcal{N}$, is defined by:

$$\forall p \in P_{\mathcal{N}},\ \forall t \in T_{\mathcal{N}},$$

$$W_{\mathcal{N}}(p,t) = \begin{cases} W_L(p,t) & \text{if } p \in P_L \wedge t \in T_L \\ W_R(p,t) & \text{if } p \in P_R \wedge t \in T_R \\ 0 & \text{otherwise} \end{cases}$$

with the corresponding block matrix representation being:

$$W_{\mathcal{N}} = \begin{pmatrix} W_L & 0 \\ 0 & W_R \end{pmatrix}$$

5.3.2 Under choice composition

In choice composition there is also no fusion of places or transitions of the *cWN*s composed. However, choice composition incorporates a new place (the *choice place*) and two new transitions, namely t_l and t_r. To apply a choice composition to two *cWN*s, namely L and R, it is necessary to define for each component the subset of entry places that will participate in the choice composition. These subsets are denoted $Choice(L)$ and $Choice(R)$. Recall that all the places that participate in the choice composition will have the same colour domain, which will also be the colour domain of the choice place. Consider the following representation of the incidence matrices W_L and W_R of the components L and R, respectively.

$$\begin{array}{ll} P_L - Choice(L) \\ Choice_L \end{array} \begin{pmatrix} W'_L \\ Ch_L \end{pmatrix} \qquad \begin{array}{ll} P_R - Choice(R) \\ Choice_R \end{array} \begin{pmatrix} W'_R \\ Ch_R \end{pmatrix}$$

The incidence matrix of the resulting cWN $\mathcal{N}$—generated from the choice composition of L and R— is then given by:

$$\begin{array}{r} \\ P_L - Choice(L) \\ P_R - Choice(R) \\ Choice(L) \\ Choice(R) \\ p_c \end{array} \begin{array}{c} \begin{array}{cccc} T_L & T_R & t_l & t_r \end{array} \\ \begin{pmatrix} W'_L & 0 & 0 & 0 \\ 0 & W'_R & 0 & 0 \\ Ch_L & 0 & w_l & 0 \\ 0 & Ch_R & 0 & w_r \\ 0 & 0 & -X & -X \end{pmatrix} \end{array}$$

where

- w_l is a column vector of dimension $Choice(L)$, where $\forall p \in Choice(L)$, $w_l(p) = X$, with X the identity function defined over the colour domain of p_c;
- w_r is a column vector of dimension $Choice(R)$, where $\forall p \in Choice(R)$, $w_r(p) = X$;

5.3.3 Under composition operations involving place fusion

At the level of the incidence matrix, the fusion of places is represented by the addition of the rows corresponding to the places being fused and placing the resulting vector as the row corresponding to the resulting fusion place. Recall that only places that have the same colour domain can be fused.

Given the incidence matrix of a cWN $\mathcal{N}$ with two distinguished places p_e and p_f:

$$\begin{array}{r} \\ P_{\mathcal{N}} - \{p_e, p_f\} \\ p_e \\ p_f \end{array} \begin{array}{c} T_{\mathcal{N}} \\ \begin{pmatrix} W'_{\mathcal{N}} \\ w_e \\ w_f \end{pmatrix} \end{array}$$

where

- $\forall p \in P_{\mathcal{N}} - \{p_e, p_f\}$, $\forall t \in T_{\mathcal{N}}$, $W'_{\mathcal{N}}(p,t) = W_{\mathcal{N}}(p,t)$ and
- $\forall t \in T_{\mathcal{N}}$, $w_e(t) = W_{\mathcal{N}}(p_e, t)$ and $w_f(t) = W_{\mathcal{N}}(p_f, t)$.

The resulting incidence matrix is given by:

$$\begin{array}{r c} & T_{\mathcal{N}} \\ P_{\mathcal{N}} - \{p_e, p_f\} & \multirow{2}{*}{$\begin{pmatrix} W'_{\mathcal{N}} \\ w_{ef} \end{pmatrix}$} \\ p_{ef} & \end{array}$$

where $w_{ef} = w_e + w_f$.

Binary composition operations that involve fusion of places of two components, say L and R, in order to create a third cWN $\mathcal{N}$, can be seen as a parallel composition, followed by the internal fusion of the corresponding places. In this case the set of transitions $T_{\mathcal{N}}$ can be divided into the disjoint sets T_L and T_R, and similarly the set of places $P_{\mathcal{N}}$ can be divided into $P_L - \{p_e\}$ and $P_R - \{p_f\}$ (assuming $p_e \in P_L$ and $p_f \in P_R$). The resulting incidence matrix will have the form:

$$\begin{array}{r c} & \begin{matrix} T_L & T_R \end{matrix} \\ \begin{matrix} P_L - \{p_e\} \\ P_R - \{p_f\} \\ p_{ef} \end{matrix} & \begin{pmatrix} W'_L & 0 \\ 0 & W'_R \\ w_e & w_f \end{pmatrix} \end{array}$$

where $\forall p \in P_L - \{p_e\}, \forall t \in T_L,\ W'_L(p,t) = W_L(p,t)$ (and similarly for W'_R), $\forall t \in T_L,\ w_e(t) = W_L(p_e, t)$ and $\forall t \in T_R,\ w_f(t) = W_R(p_f, t)$.

5.3.4 Under synchronisation

Unlike place fusion, the colour domain of the transition t_{lr} resulting from the synchronisation of two others, namely t_l and t_r, is formed by the product of the colour domains of t_l and t_r ($C(t_{lr}) = C(t_l) \times C(t_r)$). Given $\mathcal{S}$ a cWN, the entries of the incidence matrix of the cWN $\mathcal{N}$, resulting from the synchronisation of the transitions $t_l, t_r \in T_{\mathcal{S}}$, are given by:

$\forall p \in P_{\mathcal{N}}, \forall t \in T_{\mathcal{N}}, \forall c \in C(p), \forall c' \in C(t),$

$$W_{\mathcal{N}}(p,t)(c,c') = \begin{cases} W_{\mathcal{S}}(p,t_l)(c,c_l) & \text{if } (t = t_{lr} \ \wedge\ p \in {}^\bullet t_l \cup t_l^\bullet) \text{ where } c' = \langle c_l, c_r \rangle \\ W_{\mathcal{S}}(p,t_r)(c,c_r) & \text{if } (t = t_{lr} \ \wedge\ p \in {}^\bullet t_r \cup t_r^\bullet) \text{ where } c' = \langle c_l, c_r \rangle \\ 0 & \text{if } (t = t_{lr}) \ \wedge\ p \notin ({}^\bullet t_l \cup t_l^\bullet \cup {}^\bullet t_r \cup t_r^\bullet) \\ W_{\mathcal{S}}(p,t)(c,c') & \text{otherwise} \end{cases}$$

The synchronisation between two cWNs can be seen as a parallel composition of the cWNs followed by an internal synchronisation of its corresponding transitions.

5.4 Compositional construction of P-semiflows in cWNs

Having studied the compositional construction of the incidence matrix of a cWN for each of the composition operations, we now go on to study the construction of P-semiflows of a cWN using the set of P-semiflows of its sub-components.

Symbolic and static P-semiflows can be obtained for bWNs. These will act as the basis for the construction of P-semiflows of higher level cWNs.

5.4.1 P-semiflows of bWNs

As we saw when defining symbolic P-semiflows of WNs, the number of coloured objects of a symbolic colour $\bar{c}_p$ that a canonical transition instance $[t, \bar{c}]$ extracts from (places into) a place p, is equal for all transition instances of $[t, \bar{c}]$. The solution proposed for the deduction of symbolic P-semiflows of a bWN is based on this fact. By knowing how many coloured objects of a particular symbolic colour are extracted from an entry place, and how many coloured objects of a symbolic colour (of the final place) are placed into a final place by the canonical instances of the bWN's transition, it is possible to obtain a symbolic P-semiflow.

The algorithm presented is based on the Fourier-Motzkin method for solving systems of linear inequalities [CS90]. The basic idea is to eliminate variables from the system by adding to them all the inequalities resulting from positive linear combinations of pairs of inequalities. The algorithm is an adaptation and extension of Algorithm 1 in [CS90].

In a bWN the set of places can be partitioned into two sets, namely the entry set (ES) and the final set (FS). The algorithm will work on the set ES$\times$FS of pairs of places. For each pair $(p_e, p_f) \in$ ES$\times$FS it will obtain a matrix W_{ef} of cardinality $(|C(p_e)| + |C(p_f)|) \times |C(t)|$ (where t is the transition of the bWN) defined as:

$$\forall p \in \{p_e, p_f\}, \forall \bar{c_p} \in S_c(p), \forall \bar{c_t} \in S_c(t), \quad W_{ef}(p,t)(\bar{c_p}, \bar{c_t}) = W(p,t)(c_p, c_t)$$

where $S_c(p)$ and $S_c(t)$ are the sets of symbolic colours of the place p and the transition t, respectively, $c_p \in \bar{c_p}$ is a colour in the symbolic colour $\bar{c_p}$ and $c_t \in \bar{c_t}$ is a colour in the symbolic colour $\bar{c_t}$. Figure 5.4 shows the W_{ef} matrix corresponding to the bWN **C** of the example of the message communication system presented on page 83.

$$
\begin{array}{cl|cc}
 & & \multicolumn{2}{c}{t} \\
 & & \langle S_{1,1}, X_1^1 \rangle & \langle S_{1,1}, S_{1,1} - X_1^1 \rangle \\
\hline
 & \langle S_{1,1}, X_1^1, S_{2,1} \rangle & -|D_{2,1}| & 0 \\
p_e & \langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1} \rangle & 0 & -|D_{2,1}| \\
 & \langle S_{1,1}, X_1^1, S_{2,2} \rangle & 0 & 0 \\
 & \langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2} \rangle & 0 & 0 \\
 & & & \\
 & \langle S_{1,1}, X_1^1, S_{2,1} \rangle & 0 & 0 \\
p_f & \langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1} \rangle & 0 & 0 \\
 & \langle S_{1,1}, X_1^1, S_{2,2} \rangle & |D_{2,2}| & 0 \\
 & \langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2} \rangle & 0 & |D_{2,2}|
\end{array}
$$

Figure 5.4: Example of the W_{ef} built from the symbolic colour entries of the places *R. Message buf* and *T. Message buf* of the *bWN* **C** of Figure 5.1 on page 83, with $p_e =$ *R. Message buf* and $p_f =$ *T. Message buf*

Given an arc $\langle p, t \rangle$, it is possible to obtain the number of coloured objects selected (extracted or placed) by a canonical transition instance of t, for every symbolic colour of p. One approach would be to use "brute force", as said in [CDFH97], and calculate, for every symbolic colour of p, the number of colour objects that are selected by the canonical transition instance. This would mean that we would even consider symbolic colours whose coloured objects are not selected, due to the conditions defined over the transition's variables. As suggested in [CDFH97] an alternative, more "reasonable" way, is to deduce from the arc function and the transition's predicates which are the symbolic colours of p from which the canonical transition instance selects objects, and for these calculate the number of objects.

$$
\begin{array}{cl|cccc}
 & & \multicolumn{4}{c}{t} \\
 & & \langle S_{1,1}, X_1^1, S_{2,1} \rangle & \langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1} \rangle & \langle S_{1,1}, X_1^1, S_{2,2} \rangle & \langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2} \rangle \\
\hline
 & \langle S_{1,1}, X_1^1, S_{2,1} \rangle & -1 & 0 & 0 & 0 \\
p_e & \langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1} \rangle & 0 & -1 & 0 & 0 \\
 & \langle S_{1,1}, X_1^1, S_{2,2} \rangle & 0 & 0 & 0 & 0 \\
 & \langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2} \rangle & 0 & 0 & 0 & 0 \\
 & & & & & \\
 & \langle S_{1,1}, X_1^1, S_{2,1} \rangle & 1 & 0 & 0 & 0 \\
p_f & \langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1} \rangle & 0 & 1 & 0 & 0 \\
 & \langle S_{1,1}, X_1^1, S_{2,2} \rangle & 0 & 0 & 0 & 0 \\
 & \langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2} \rangle & 0 & 0 & 0 & 0
\end{array}
$$

Figure 5.5: W_{ef} of the *bWN* **B** of Figure 5.1

For example to obtain the W_{ef} of the *bWN* **B** of the message communication system the predicate $X_2 \in D_{2,1}$ already tells us that for the canonical transition instances of the transition *Transmit* for which X_2 is an element in $D_{2,2}$ the corresponding column in the W_{ef} will be $\vec{0}^T$. Figure 5.5 shows the W_{ef} of the *bWN* **B**, built using the symbolic colour entries of the places *R. Message buf* (p_e) and *T. Message buf* (p_f).

The basic functions $\beta_{i,j} \cdot X_i^j$ and $\gamma_{i,j}^u \cdot \oplus^u X_i^j$ in an arc function select for each static sub-class from which X_i^j can take its values, $\beta_{i,j}$ and $\gamma_{i,j}^u$ objects, respectively. The basic function $\alpha_{i,q} \cdot S_{i,q}$ in the j^{th} entry of the arc function, will select $\alpha_{i,q}$ times each coloured object whose j^{th} entry is a colour belonging to the static sub-class $S_{i,q}$.

In order to be able to obtain a generative family of symbolic P-semiflows, we must define what constitutes a minimal symbolic P-semiflow. This concept emerges directly from the general definition of minimal semiflows.

Definition 5.10 (Minimal symbolic P-semiflow) *A symbolic P-semiflow v of a cWN $\mathcal{N}$, is* minimal *if and only if it is canonical and it satisfies the condition $\forall z$ symbolic P-semiflow of $\mathcal{N}$, with $z \neq v$,*

$$\|z\| \not\subset \|v\|$$

Proposition 5.5 *Algorithm 5.1 obtains the set of minimal symbolic P-semiflows of a bWN.*

$\mathcal{J}$ denotes the set of minimal symbolic P-semiflows of the *bWN*. $\mathcal{J}_{ef}$ denotes a matrix whose rows are vectors that annul the AUX matrix. Each row of the resulting $\mathcal{J}_{ef}$s, represents a positive linear combination of rows of the corresponding matrix W_{ef}. At the end of the iteration that works over the matrix $[\mathcal{J}_{ef}|AUX]$, the rows of the matrix AUX are null. The vectors v defined from the rows of each $\mathcal{J}_{ef}$ are symbolic P-semiflows that only cover the entries of the places p_e and p_f. This is done for every possible pair of an entry place with a final place. Within the symbolic P-semiflows that cover the same places, the minimality is guaranteed by introducing in $\mathcal{J}$ only symbolic P-semiflows whose support is not equal or does not contain the support of any other symbolic P-semiflow in $\mathcal{J}$.

At the level of static P-semiflows, we want to be able to calculate, for a given colour sub-domain, the number of its coloured objects extracted from, or placed into, its associated place by a certain canonical transition instance. This can be done in the same way as for symbolic P-semiflows. In this case from a given canonical transition instance we obtain the set of static sub-domains for which the canonical transition instance selects coloured objects, and the number of coloured objects selected. The algorithm to obtain static P-semiflows of a bWN can be obtained by appropriately modifying the algorithm proposed for symbolic P-semiflows.

Algorithm 5.1.- (Obtaining the minimal symbolic P-semiflows of a $bSWN$**)**

> **/* Given the incidence matrix W, the set of entry places ES and the set of final places FS, it obtains the set of minimal symbolic P-semiflows $\mathcal{J}$ */**
>
> {
>
> - Let $\mathcal{J} = \emptyset$
> - Obtain the set $S_c(t)$ of symbolic colours of t
> - <u>For all</u> $p_e \in ES$ <u>do</u>
>   - Obtain the set $S_c(p_e)$ of symbolic colours of p_e
>   - <u>For all</u> $p_f \in FS$ <u>do</u>
>   - Obtain the set $S_c(p_f)$ of symbolic colours of p_f
>   - Obtain W_{ef} the reduced symbolic incidence matrix for p_e and p_f
>
>   /* *this can be done by using the information of the arc functions between t and p_e, between t and p_f and of the transition predicates of t.* */
>   - Let $AUX = W_{ef}$ and $\mathcal{J}_{ef} = I_{ef}$
>
>   /**where I_{ef} is an identity matrix of dimension $|S_c(p_e)| + |S_c(p_f)|$.* */
>   - <u>For</u> $i = 1$ to $|S_c(t)|$ <u>do</u>
>   - Add to the matrix $[\mathcal{J}_{ef}|AUX]$ all the rows which are linear combinations of pairs of rows of $[\mathcal{J}_{ef}|AUX]$ and which annul the i^{th} column of AUX.
>   - Eliminate from $[\mathcal{J}_{ef}|AUX]$ the rows with a non-zero i^{th} column.
>
>   <u>end do</u>
>   - <u>For every</u> row j of $\mathcal{J}_{ef}$ <u>do</u>
>   - obtain j' the canonical vector of j
>   - Obtain v defined as:
>
> $$\forall p \in P,\ \forall \bar{c} \subset C(p)\ v(p, \bar{c}) = \begin{cases} j'(p, \bar{c}) & \text{if } p \in \{p_e, p_f\} \\ 0 & \text{otherwise} \end{cases}$$
>   - If $\forall z \in \mathcal{J}$, $||v|| \not\supseteq ||z||$ then let $\mathcal{J} = \mathcal{J} \cup \{v\}$
>
>   <u>end do</u>
>
>   <u>end do</u>
>
>   <u>end do</u>
>
> }

Let us consider the application of the algorithm to the bWN **B** of the message communication system. We have already presented the W_{ef} of its only entry place with its only output place (see Figure 5.5). The matrix $[\mathcal{J}_{ef}|AUX]$ produced by the algorithm will have the form:

$$
\begin{array}{cccccccc|cccc}
 &&&&&&&& \langle S_{1,1},X_1^1,S_{2,1}\rangle & \langle S_{1,1},S_{1,1}-X_1^1,S_{2,1}\rangle & \langle S_{1,1},X_1^1,S_{2,2}\rangle & \langle S_{1,1},S_{1,1}-X_1^1,S_{2,2}\rangle \\
1&0&0&0&0&0&0&0&-1&0&0&0\\
0&1&0&0&0&0&0&0&0&-1&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0\\
\\
0&0&0&0&1&0&0&0&1&0&0&0\\
0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0
\end{array}
$$

The third, fourth and last two rows of the right hand side of the matrix are already null vectors therefore their corresponding left hand-side rows define symbolic P-semiflows. Each column entry of the left hand side matrix ($\mathcal{J}_{ef}$) corresponds to a symbolic colour of p_e or p_f. The third row (for example) defines the symbolic P-semiflow:

$$[\langle S_{1,1}, X_1^1, S_{2,2}\rangle, 0]$$

In the first iteration i the algorithm will add the 1^{st} and 5^{th} rows producing a row whose first column in AUX is 0. We then obtain the matrix:

$$
\begin{array}{cccccccc|cccc}
 &&&&&&&& \langle S_{1,1},X_1^1,S_{2,1}\rangle & \langle S_{1,1},S_{1,1}-X_1^1,S_{2,1}\rangle & \langle S_{1,1},X_1^1,S_{2,2}\rangle & \langle S_{1,1},S_{1,1}-X_1^1,S_{2,2}\rangle \\
1&0&0&0&1&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&-1&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0\\
\\
0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0
\end{array}
$$

The following iteration will work over the second column of AUX. The algorithm would then add the 2^{nd} and 5^{th} rows producing:

$$
\begin{array}{cccccccc|cccc}
 &&&&&&&& \langle S_{1,1},X_1^1,S_{2,1}\rangle & \langle S_{1,1},S_{1,1}-X_1^1,S_{2,1}\rangle & \langle S_{1,1},X_1^1,S_{2,2}\rangle & \langle S_{1,1},S_{1,1}-X_1^1,S_{2,2}\rangle \\
1&0&0&0&1&0&0&0&0&0&0&0\\
0&1&0&0&0&1&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0\\
\\
0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0
\end{array}
$$

Given that AUX is now a null matrix in the right hand side we are left with the vectors that define a base for the set of symbolic P-semiflows of bWN **B**.

5.4.2 Compositional construction of symbolic and static P-semiflows

Given the generative families of symbolic (static) P-semiflows of lower level *cWN*s (not necessarily *bWN*s), we propose a method for the compositional combination of symbolic (static) P-semiflows to form symbolic (static) P-semiflows of higher level *cWN*s. As for the case of the incidence matrix, we will study the different types of composition operations that have been defined for the construction of *cWN*s. Many of the results obtained are applicable to general types of P-semiflows, therefore we will only use the word symbolic when the results are limited to this type of P-semiflow. The results on symbolic P-semiflows can be extended to static P-semiflows.

Consider a *cWN* $\mathcal{N}$ obtained from the composition of two *cWN*s L and R, and a pair of P-semiflows v of L and w of R. We will define the *extension* of v with respect to R—denoted $\widehat{v}^R$— as a vector of dimension $|P_L \cup P_R|$, given by:

$$\forall p \in P_L \cup P_R,\ \widehat{v}^R(p) = \begin{cases} v(p) & \text{if } p \in P_L \\ 0 & \text{otherwise} \end{cases}$$

The dual definition applies for the extension of w with respect to L.

5.4.2.1 Under independent parallel composition

From the description of the incidence matrix of a *cWN* $\mathcal{N}$ resulting from the parallel composition of two *cWN*s L and R, as presented in Section 5.3.1:

$$W_{\mathcal{N}} = \begin{pmatrix} W_L & 0 \\ 0 & W_R \end{pmatrix}$$

it can be deduced that the extension of the P-semiflows of L and R, with respect to the places of the other sub-component, are P-semiflows of the resulting *cWN* $\mathcal{N}$.

Proposition 5.6 *Given the generative family of P-semiflows of two cWNs, L and R, the generative family of P-semiflows of the cWN $\mathcal{N}$—resulting from the parallel composition of L and R—is formed by the appropriate extensions of the P-semiflows that belong to the generative families of L and R.*

Proof.- Applying contradiction, let us suppose that there is a minimal P-semiflow u of $\mathcal{N}$ that is not an extension of a minimal P-semiflow of L or R, i.e. it is not in the generative family of $\mathcal{N}$, as defined.

If u covers (assigns a non-zero weight to) only the places of L, then, given that u is minimal, u corresponds to an extension of a P-semiflow in the generative family of P-semiflows of L. If this holds then u belongs to the generative family of P-semiflows of $\mathcal{N}$. The dual holds if u only covers places of R.

If u covers places of both L and R, then it is possible to create two P-semiflows u_1, u_2 of $\mathcal{N}$, defined as:

$$\forall p \in P_L \cup P_R,$$

$$u_1(p) = \begin{cases} u(p) & \text{if } p \in P_L \\ 0 & \text{otherwise} \end{cases}$$

$$u_2(p) = \begin{cases} u(p) & \text{if } p \in P_R \\ 0 & \text{otherwise} \end{cases}$$

such that $\|u_1\| \subset \|u\|$ and $\|u_2\| \subset \|u\|$, which means that u is not minimal, contradicting the hypothesis. ■

5.4.2.2 Under choice composition

Let us recall the form of the incidence matrix of the *cWN* $\mathcal{N}$ resulting from the choice composition of two *cWN*s, L and R, as shown in Section 5.3.2:

$$\begin{array}{r c} & \begin{array}{cccc} T_L & T_R & t_l & t_r \end{array} \\ \begin{array}{r} P_L - Choice(L) \\ P_R - Choice(R) \\ Choice(L) \\ Choice(R) \\ p_c \end{array} & \begin{pmatrix} W'_L & 0 & 0 & 0 \\ 0 & W'_R & 0 & 0 \\ Ch_L & 0 & w_l & 0 \\ 0 & Ch_R & 0 & w_r \\ 0 & 0 & -X & -X \end{pmatrix} \end{array}$$

In this case the extension of a P-semiflow is defined to consider not only the places of the other sub-component, but also to include an entry for the choice place p_c. So, given a P-semiflow v of L we define the *choice extension* of v with respect to R—denoted $\widehat{v}_c^R$—as a vector of dimension $|P_L \cup P_R| + 1$, such that:

$$\forall p \in P_L \cup P_R \cup \{p_c\},\ \ \widehat{v}_c^R(p) = \begin{cases} v(p) & \text{if } p \in P_L \\ 0 & \text{otherwise} \end{cases}$$

The dual definition applies for the extension of a P-semiflow of R with respect L.

If v is such that $\forall p \in Choice(L),\ v(p) = 0$, then the choice extension $\widehat{v}_c^R$ of v will be a P-semiflow of $\mathcal{N}$.

$$\begin{aligned} \widehat{v}_c^R \cdot W_{\mathcal{N}} =& \langle \sum_{p \in P_L - Choice(L)} \widehat{v}_c^R(p) \cdot W'_L(p, \cdot), 0, 0, 0\rangle \\ =& \langle \sum_{p \in P_L - Choice(L)} v(p) \cdot W_L(p, \cdot), 0, 0, 0\rangle \end{aligned}$$

Given that $\forall p \in Choice(L),\ v(p) = 0$ and that v is a P-semiflow of L, it holds that $\widehat{v}_c^R \cdot W_{\mathcal{N}} = 0$. The same applies for a P-semiflow of R that assigns zero to all places in $Choice(R)$.

In the case that $\exists p \in Choice(L),\ v(p) \neq 0$, the choice extension of v will not constitute a P-semiflow of $\mathcal{N}$. The same holds for P-semiflows of R that assign a value greater than zero to at least one place in $Choice(R)$.

By analysing the structure of the incidence matrix $W_{\mathcal{N}}$, we can observe that: a P-semiflow u of $\mathcal{N}$ that covers the choice place p_c must also cover at least one place in $Choice(L)$ and one place in $Choice(R)$.

If $u(p_c) > 0$ then $u(p_c) \cdot -X = -u(p_c) < 0$. Consider the column of the incidence matrix corresponding to transition t_l. In order to be able to annul $W_{\mathcal{N}}(\cdot, t_l)$ it must hold that:

$$\sum_{p \in Choice(L)} u(p) \cdot W_{\mathcal{N}}(p, t_l) = \sum_{p \in Choice(L)} u(p) > 0$$

Let us introduce the following notation:

$$v_c = \sum_{p \in Choice(L)} v(p) \qquad w_c = \sum_{p \in Choice(R)} w(p)$$

and

$$mult_v = \frac{lcm(v_c, w_c)}{v_c} \qquad mult_w = \frac{lcm(v_c, w_c)}{w_c}$$

where lcm corresponds to the integer *least common multiple* function. This leads us to the following proposition.

Proposition 5.7
Given a pair of P-semiflows v of L and w of R such that $v_c \cdot w_c > 0$, we can obtain a P-semiflow u of $\mathcal{N}$ defined as:

$$\forall p \in P_{\mathcal{N}},$$
$$u(p) = \begin{cases} mult_v \cdot v(p) & \textit{if } p \in P_L \\ mult_w \cdot w(p) & \textit{if } p \in P_R \\ lcm(v_c, w_c) & \textit{otherwise } (p = p_c) \end{cases}$$

Proof.- Let us view u as a vector with the form: $u = \langle u_L, u_R, u_{Ch_L}, u_{Ch_R}, u_{p_c} \rangle$ where u_L is a sub-vector which covers the places in $P_L - Choice(L)$, u_R covers the places in $P_R - Choice(R)$, u_{Ch_L} covers the places in $Choice(L)$, u_{Ch_R} covers the place in $Choice(R)$ and u_{p_c} is the entry for the choice place. We then have that:

$$u \cdot W_{\mathcal{N}} = \langle u_L \cdot \bigotimes_{t \in T_L} W_{\mathcal{N}}(\cdot, t), u_R \cdot \bigotimes_{t \in T_R} W_{\mathcal{N}}(\cdot, t), u_{Ch_L} \cdot w_l - u_{p_c} \cdot X, u_{Ch_R} \cdot w_r - u_{p_c} \cdot X \rangle$$

where

$$u_L \cdot \bigotimes_{t\in T_L} W_{\mathcal{N}}(\cdot,t) = \bigotimes_{t\in T_L}\left(\sum_{p\in P_L - Choice(L)} u(p)\cdot W'_L(p,t) + \sum_{p\in Choice(L)} u(p)\cdot Ch_L(p,t)\right)$$

$$u_R \cdot \bigotimes_{t\in T_R} W_{\mathcal{N}}(\cdot,t) = \bigotimes_{t\in T_R}\left(\sum_{p\in P_R - Choice(R)} u(p)\cdot W'_R(p,t) + \sum_{p\in Choice(R)} u(p)\cdot Ch_R(p,t)\right)$$

$$u_{Ch_L}\cdot w_l - u_{p_c}\cdot X = \left(\sum_{p\in Choice(L)} u(p)\right) - u(p_c)$$

$$u_{Ch_R}\cdot w_r - u_{p_c}\cdot X = \left(\sum_{p\in Choice(R)} u(p)\right) - u(p_c)$$

Substituting the values of u in the equations above and following the hypothesis of the proposition, we then have that:

$$\begin{aligned} u_L \cdot \bigotimes_{t\in T_L} W_{\mathcal{N}}(\cdot,t) = \bigotimes_{t\in T_L}\Bigg(&\sum_{p\in P_L - Choice(L)} mult_v\cdot v(p)\cdot W'_L(p,t) \\ &+ \sum_{p\in Choice(L)} mult_v\cdot v(p)\cdot Ch_L(p,t)\Bigg)\end{aligned}$$

$$\begin{aligned} u_R \cdot \bigotimes_{t\in T_R} W_{\mathcal{N}}(\cdot,t) = \bigotimes_{t\in T_R}\Bigg(&\sum_{p\in P_R - Choice(R)} mult_w\cdot w(p)\cdot W'_R(p,t) \\ &+ \sum_{p\in Choice(R)} mult_w\cdot w(p)\cdot Ch_R(p,t)\Bigg)\end{aligned}$$

$$u_{Ch_L}\cdot w_l + u_{p_c}\cdot X = \left(\sum_{p\in Choice(L)} mult_v\cdot v(p)\right) - lcm(v_c,w_c)$$

$$u_{Ch_R}\cdot w_r + u_{p_c}\cdot X = \left(\sum_{p\in Choice(R)} mult_w\cdot w(p)\right) - lcm(v_c,w_c)$$

By the definitions of v_c, $mult_v$, w_c and $mult_w$, and given that v is a P-semiflow of L and w is a P-semiflow of R, it holds that:

$$u\cdot W_{\mathcal{N}} = 0$$

■

Furthermore, we can prove that if v and w are minimal, i.e. they belong to the generative families of their respective *cWN*s, then u—the P-semiflow of $\mathcal{N}$ generated as proposed in Proposition 5.7—is minimal in the set of P-semiflows of $\mathcal{N}$.

By contradiction, let us suppose that there is a minimal P-semiflow z of $\mathcal{N}$ such that $\|z\| \subseteq \|u\|$. As in the proof of Proposition 5.7, u and z can be represented in the following manner:

$$z = \langle z_L, z_R, z_{Ch_L}, z_{Ch_R}, z_{p_c} \rangle \qquad u = \langle u_L, u_R, u_{Ch_L}, u_{Ch_R}, u_{p_c} \rangle$$

Given that z is a P-semiflow of $\mathcal{N}$, it holds that:

$$\bigotimes_{t \in T_L} z_L \cdot W'_L + z_{Ch_L} \cdot Ch_L = 0$$

Therefore, $\langle z_L, z_{Ch_L} \rangle$ is a P-semiflow of L, and the same holds for $\langle u_L, u_{Ch_L} \rangle$.

From the definition of u in Proposition 5.7, we know that $\|v\| = \|\langle u_L, u_{Ch_L} \rangle\|$. The fact that $\|z\| \subseteq \|u\|$ implies that:

$$\|z_L\| \subseteq \|u_L\| \;\text{ and }\; \|z_R\| \subseteq \|u_R\|$$

Therefore, $\|\langle z_L, z_{Ch_L} \rangle\| \subseteq \|v\|$, which means that v is not minimal in the set of P-semiflows of L. This contradicts our hypothesis. The dual argument using w instead of v applies for the symbolic P-semiflows of R.

From the combination of these propositions, it can be deduced that every minimal P-semiflow of $\mathcal{N}$, can be obtained by combining (as explained in Proposition 5.7) minimal P-semiflows of the sub-components. In this way we can find the generative family of P-semiflows of $\mathcal{N}$.

5.4.2.3 Under place fusion operations

As mentioned in Section 5.3.3, binary composition operations that involve fusion of places between two components, in order to create a third *cWN*, can be seen as a parallel composition of the components into a *cWN* $\mathcal{S}$ and then an internal fusion of the corresponding places. We have already examined how to obtain the P-semiflows of a *cWN* $\mathcal{S}$ resulting from the parallel composition of two other *cWN*s, namely L and R, (see Section 5.4.2.1). The extension of the P-semiflows of the generative families of L and R form the generative family of P-semiflows of $\mathcal{S}$. We will therefore limit our study to fusion of places within a single component.

Let us denote by $\mathcal{N}$ the *cWN* resulting from applying a place fusion composition operation over the *cWN* $\mathcal{S}$. Given the generative family of symbolic

P-semiflows of $\mathcal{S}$, let us consider how to obtain the symbolic P-semiflows of $\mathcal{N}$. The following proposition implies that from the symbolic P-semiflows of $\mathcal{S}$ that assign the same weight to the the places being fused we can obtain symbolic P-semiflows of $\mathcal{N}$.

Proposition 5.8 *Given a symbolic P-semiflow v of a cWN $\mathcal{S}$, such that for two distinguished places $p_e, p_f \in P_{\mathcal{S}} : v(p_e) = v(p_f)$, we can obtain a symbolic P-semiflow $\tilde{v}$ of the cWN $\mathcal{N}$—resulting after the fusion of the places p_f and p_e—defined as:*

$$\forall p \in P_{\mathcal{S}},\ \tilde{v}(p) = \begin{cases} v(p) & \text{if } p \in P_{\mathcal{S}} - \{p_e, p_f\} \\ v(p_e) & \text{otherwise } (p = p_{fe}) \end{cases}$$

Proof.- In Section 5.3.3 it has been shown that the fusion of two places within a PN is reflected at the level of the incidence matrix as the sum of the rows corresponding to the places fused. This same pattern holds for *cWN*s, but now the entries of the places are matrices of dimension $|C(p)| \times (\sum_{t \in T} |C(t)|)$, where $T = T_{\mathcal{S}} = T_{\mathcal{N}}$. Given $W_{\mathcal{N}}$ the incidence matrix of $\mathcal{N}$, we need to prove that:

$$\forall t \in T,\ \ v(p_e) \cdot W_{\mathcal{S}}(p_e, t) + v(p_f) \cdot W_{\mathcal{S}}(p_f, t) = \tilde{v}(p_{fe}) \cdot W_{\mathcal{N}}(p_{fe}, t)$$

From the definition of $W_{\mathcal{N}}(p_{ef}, \cdot)$:

$$\begin{aligned} \forall t \in T,\ \ \tilde{v}(p_{fe}) \cdot W_{\mathcal{N}}(p_{fe}, t) =& \tilde{v}(p_{fe}) \cdot (W_{\mathcal{S}}(p_e, t) + W_{\mathcal{S}}(p_f, t)) \\ =& \tilde{v}(p_{fe}) \cdot W_{\mathcal{S}}(p_e, t) + \tilde{v}(p_{fe}) \cdot W_{\mathcal{S}}(p_f, t) \end{aligned}$$

From the definition of $\tilde{v}$ we know that $\tilde{v}(p_{fe}) = v(p_e)$ and from the initial condition $v(p_e) = v(p_f)$. By appropriately substituting for $v(p_e)$ or $v(p_f)$ the instances of $\tilde{v}(p_{fe})$ in the above equation we can obtain the result desired. ■

In the case of place fusion between two components L and R, it holds that $\forall v$ in the generative family of P-semiflows of L such that $v(p_e) = 0$, the extension of v with respect to the places of R, will define a symbolic P-semiflow of $\mathcal{N}$. The same holds for the P-semiflows of R that assign 0 to the entries of place p_f.

Furthermore, we can show that only symbolic P-semiflows that satisfy the conditions defined in Proposition 5.8 induce a symbolic P-semiflow of $\mathcal{N}$.

Proposition 5.9 *Given a cWN $\mathcal{S}$ and its set of symbolic P-semiflows $\mathcal{J}$. Only symbolic P-semiflows $v \in \mathcal{J}$ that satisfy $v(p_e) = v(p_f) > 0$ can induce a symbolic P-semiflow w of $\mathcal{N}$ with $w(p_{ef}) > 0$.*

Proof.- Consider v a symbolic P-semiflow of $\mathcal{S}$ such that $v(p_e) \neq v(p_f)$. Let us define the set $P' \subset P_{\mathcal{S}}$ as $P' = P_{\mathcal{S}} - \{p_e, p_f\}$. It then holds that:

$$\forall t \in T_{\mathcal{S}},\ \left(\sum_{p \in P'} v(p) \cdot W_{\mathcal{S}}(p,t)\right) + v(p_e) \cdot W_{\mathcal{S}}(p_e,t) + v(p_f) \cdot W_{\mathcal{S}}(p_f,t) = 0$$

Now we know that $T_{\mathcal{S}} = T_{\mathcal{N}}$ and that $\forall p \in P',\ W_{\mathcal{S}}(p,t) = W_{\mathcal{N}}(p,t)$, so

$$\forall t \in T_{\mathcal{N}},\ \left(\sum_{p \in P'} v(p) \cdot W_{\mathcal{N}}(p,t)\right) + v(p_e) \cdot W_{\mathcal{S}}(p_e,t) + v(p_f) \cdot W_{\mathcal{S}}(p_f,t) = 0$$

Knowing that $\forall t \in T_{\mathcal{N}}, W_{\mathcal{N}}(p_{ef},t) = W_{\mathcal{S}}(p_e,t) + W_{\mathcal{S}}(p_f,t)$, to define a symbolic P-semiflow of $\mathcal{N}$ from v we must find a vector w such that

$$\forall t \in T_{\mathcal{N}},\ v(p_e) \cdot W_{\mathcal{S}}(p_e,t) + v(p_f) \cdot W_{\mathcal{S}}(p_f,t) = w(p_{ef})(W_{\mathcal{S}}(p_e,t) + W_{\mathcal{S}}(p_f,t)) \tag{5.1}$$

Let us analyse Equation 5.1. The only case of place fusion where

$$(p_e^{\bullet} \cup {}^{\bullet}p_e) \cap (p_f^{\bullet} \cup {}^{\bullet}p_f) \neq \emptyset$$

is when applying a closing operation involving an input and output place of a transition. For all other cases of operations involving place fusion the definition of the operation forbids the fusion of places with common input or output transitions. In the case where places being fused do not have a common transition it is possible to divide the set of transitions into three disjoint sets: T_e of transitions connected to p_e, T_f of transitions connected to p_f and T_0 the remaining transitions. Given this we have that $\forall t \in T_e,\ W_{\mathcal{S}}(p_f,t) = 0$, equally $\forall t \in T_f,\ W_{\mathcal{S}}(p_e,t) = 0$. Combining these results with Equation 5.1 we have that:

$$\forall t \in T_e,\ v(p_e) \cdot W_{\mathcal{S}}(p_e,t) = w(p_{ef}) \cdot W_{\mathcal{S}}(p_e,t) \Rightarrow v(p_e) = w(p_{ef})$$

and

$$\forall t \in T_f,\ v(p_f) \cdot W_{\mathcal{S}}(p_f,t) = w(p_{ef}) \cdot W_{\mathcal{S}}(p_f,t) \Rightarrow v(p_f) = w(p_{ef})$$

which implies that $v(p_e)$ must equal $v(p_f)$.

For the case that p_e and p_f share a transition t_s, one must be an input and the other an output place of the transition. Given that the output place in this case must also be a final place then it cannot be input to any transition or output to any other transition, therefore:

$$\forall t \in T_{\mathcal{N}} - \{t_s\},\ W_{\mathcal{N}}(p_f,t) = 0$$

this will mean that for it to hold that

$$\forall t \in T_{\mathcal{N}} - \{t_s\}, \quad v(p_e) \cdot W_S(p_e, t) = w(p_{ef}) \cdot W_S(p_e, t)$$

it must be that

$$v(p_e) = w(p_{ef})$$

Substituting this result in Equation 5.1 for t_s, we then have that it must hold that:

$$v(p_e) \cdot W_S(p_e, t_s) + v(p_f) \cdot W_S(p_f, t_s) = v(p_e) \cdot (W_S(p_e, t_s) + W_S(p_f, t_s))$$

which implies that $v(p_e) = v(p_f)$.

■

Notice that the symbolic P-semiflow w obtained from the symbolic P-semiflow v may not be minimal.

Algorithm 5.2.- (Obtains the set Min of symbolic P-semiflows of $\mathcal{S}$, that assign equal weights to the entries of p_e and p_f)

/* Suppose $\mathcal{Q}$ the matrix representation of $\mathcal{J}$, the set of minimal symbolic P-semiflows of $\mathcal{S}$, where the rows of $\mathcal{Q}$ correspond to the minimal symbolic P-semiflows. */

{

- Let $AUX = \mathcal{Q}$ and $Comb = I_{\mathcal{J}}$

/**where $I_{\mathcal{J}}$ is an identity matrix of dimension $|\mathcal{J}|$.* */

- For $i = 1$ to $|S_c(p_e)|$ do
    - Add to the matrix $[Comb|AUX]$ all the rows which are linear combinations of pairs of rows of $[Comb|AUX]$ and which annul the i^{th} column of AUX.
    - Eliminate from $[Comb|AUX]$ the rows with a non-zero i^{th} column.

end do

- The rows of $Comb$ will determine the P-semiflows of $\mathcal{S}$ that assign equal values to p_e and p_f: $\mathcal{H} = Comb \cdot W_{\mathcal{J}}$

/**From the matrix $\mathcal{H}$ we can obtain the set Min, of minimal symbolic P-semiflows that assign the same values to the entries of p_e and p_f.* */

- Let $Min = \emptyset$
- For all rows $v \in \mathcal{H}$ do
    - Obtain the canonical vector of v, denoted e
    - If $\forall w \in Min$, $||e|| \not\supseteq ||w||$ then let $Min = Min \cup \{e\}$

end do

}

To obtain a generative set of symbolic P-semiflows of $\mathcal{N}$ we need a way to build a generative set Min of minimal symbolic P-semiflows of $\mathcal{S}$ that satisfy that

$\forall v \in Min,\;\; v(p_e) = v(p_f)$. The elements of Min will be minimal with respect to the set of symbolic P-semiflows that satisfy the condition, but not necessarily minimal with respect to the generative family of P-semiflows of $\mathcal{S}$. Algorithm 5.2 follows the approach of Algorithm 5.1 on page 98 with respect to the idea of linear combination of rows of a matrix, but in this case the rows will be the entries of p_e and p_f of the symbolic P-semiflows of $\mathcal{S}$. The columns of the matrix will correspond to the symbolic colours of the p_e (which are equal to those of p_f). The aim of the algorithm is then to find linear combinations of these rows that result in rows that assign equal weights to the symbolic colours of p_e and p_f. So, consider the matrix $\mathcal{Q}$ of dimension $|\mathcal{J}| \times |S_c(p_e)|$, where $\mathcal{J}$ is the set of minimal symbolic P-semiflows of $\mathcal{S}$ and $S_c(p_e)$ is the set of symbolic colours of p_e, hence of p_f. The entries of $\mathcal{Q}$ will then be defined as:

$$\forall j \in \mathcal{J},\; \forall \bar{c} \in S_c(p_e),\; \mathcal{Q}(j, \bar{c}) = j(p_e, \bar{c}) - j(p_f, \bar{c})$$

5.4.2.4 Under synchronisation

Similarly to the case of composition operations involving place fusion between two different *cWN*, synchronisation between transitions of two *cWN*s can be seen as a parallel composition followed by an internal synchronisation. Therefore, we will limit our analysis to the case of internal synchronisation.

The following proposition states that all P-semiflows of the original *cWN* are also P-semiflows of the resulting *cWN*, after applying the synchronisation operation.

Proposition 5.10 *Suppose $\mathcal{S}$ is a cWN with two distinguished transitions $t_l, t_r \in T_{\mathcal{S}}$ such that $({}^\bullet t_l \cup t_l^\bullet) \cap ({}^\bullet t_r \cup t_r^\bullet) = \emptyset$. Every P-semiflow of $\mathcal{S}$ will be a P-semiflows of the cWN $\mathcal{N}$, resulting from the synchronisation of the transitions t_l and t_r in $\mathcal{S}$.*

Proof.- Consider any P-semiflows v of $\mathcal{S}$. We want to prove that $v \cdot W_{\mathcal{N}} = 0$, i.e. that v is also a P-semiflow of $\mathcal{N}$. From analysing the incidence matrix of the *cWN* $\mathcal{N}$, we know that:

$$\forall t \in T_{\mathcal{N}} - \{t_{lr}\},\; W_{\mathcal{N}}(\cdot, t) = W_{\mathcal{S}}(\cdot, t)$$

where t_{lr} is the transition of $\mathcal{N}$ resulting from the synchronisation of the transitions t_l and t_r in $\mathcal{S}$. Given that v is a P-semiflow of $\mathcal{S}$, then:

$$\forall t \in T_{\mathcal{N}} - \{t_{lr}\},\; v \cdot W_{\mathcal{N}}(\cdot, t) = 0$$

We are then left with the case of the transition t_{lr}. It must hold that

$$\sum_{p\in P_{\mathcal{N}}} v(p)\cdot W_{\mathcal{N}}(p,t_{lr}) = 0$$

From the hypothesis we know that $({}^\bullet t_l \cup t_l^\bullet)\ \cap\ ({}^\bullet t_r \cup t_r^\bullet) = \emptyset$. Given that there is no place fusion involved in the operation, it holds that $P_{\mathcal{N}} = P_{\mathcal{S}}$. It is then possible to partition the set of places $P_{\mathcal{N}}$ into three parts, namely $P_l = ({}^\bullet t_l \cup t_l^\bullet)$, $P_r = ({}^\bullet t_r \cup t_r^\bullet)$ and $P_{rst} = P_{\mathcal{N}} - P_l - P_r$. We then have that:

$$\sum_{p\in P_{\mathcal{N}}} v(p)\cdot W_{\mathcal{N}}(p,t_{lr}) = \\ \sum_{p\in P_l} v(p)\cdot W_{\mathcal{N}}(p,t_{lr}) + \sum_{p\in P_r} v(p)\cdot W_{\mathcal{N}}(p,t_{lr}) + \sum_{p\in P_{rst}} v(p)\cdot W_{\mathcal{N}}(p,t_{lr})$$

Given that the input and output places of t_l and t_r are all in $P_l \cup P_r$, then

$$\sum_{p\in P_{rst}} v(p)\cdot W_{\mathcal{N}}(p,t_{lr}) = 0$$

leaving:

$$\sum_{p\in P_{\mathcal{N}}} v(p)\cdot W_{\mathcal{N}}(p,t_{lr}) = \sum_{p\in P_l} v(p)\cdot W_{\mathcal{N}}(p,t_{lr}) + \sum_{p\in P_r} v(p)\cdot W_{\mathcal{N}}(p,t_{lr})$$

from the definition of the incidence matrix of $\mathcal{N}$ we have that:

$$\forall p \in P_l,\ \forall\langle \bar{c}_l, \bar{c}_r\rangle \subseteq C(t_{lr}),\ W_{\mathcal{N}}(p,t_{lr})(\cdot,\langle \bar{c}_l,\bar{c}_r\rangle) = W_{\mathcal{S}}(p,t_l)(\cdot,\bar{c}_l)$$

and

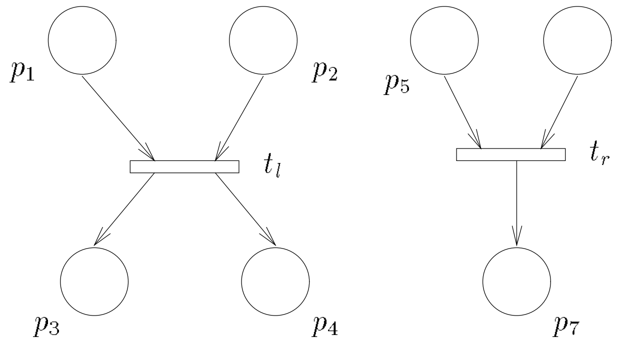
$$\forall p \in P_r,\ \forall\langle \bar{c}_l, \bar{c}_r\rangle \subseteq C(t_{lr}),\ W_{\mathcal{N}}(p,t_{lr})(\cdot,\langle \bar{c}_l,\bar{c}_r\rangle) = W_{\mathcal{S}}(p,t_r)(\cdot,\bar{c}_r)$$

Therefore,

$$\sum_{p\in P_{\mathcal{N}}} v(p)\cdot W_{\mathcal{N}}(p,t_{lr}) = \\ \bigotimes_{\langle \bar{c}_l,\bar{c}_r\rangle\subset C(t_{lr})} \left(\sum_{p\in P_l} v(p)\cdot W_{\mathcal{S}}(p,t_l)(\cdot,\bar{c}_l) + \sum_{p\in P_r} v(p)\cdot W_{\mathcal{S}}(p,t_r)(\cdot,\bar{c}_r)\right) = 0$$

This means that v is also a P-semiflow of $\mathcal{N}$. ■

Notice that we have not said that the set of P-semiflows of $\mathcal{N}$ is that of S. The most elementary example of how new P-semiflows can arise from a transition synchronisation is the case of a cWN S formed from the parallel synchronisation of two bWNs:



The minimal (symbolic) P-semiflows of S are:

$$\begin{array}{ccccccc}
p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 \\
\langle 1, & 0, & 1, & 0, & 0, & 0, & 0 \ \rangle, \\
\langle 1, & 0, & 0, & 1, & 0, & 0, & 0 \ \rangle, \\
\langle 0, & 1, & 1, & 0, & 0, & 0, & 0 \ \rangle, \\
\langle 0, & 1, & 0, & 1, & 0, & 0, & 0 \ \rangle, \\
\langle 0, & 0, & 0, & 0, & 1, & 0, & 1 \ \rangle, \\
\langle 0, & 0, & 0, & 0, & 0, & 1, & 1 \ \rangle
\end{array}$$

where the i^{th} digit corresponds to the count given to place p_i. When the transitions t_l and t_r are synchronised the existing P-semiflows are also P-semiflows of the resulting component $\mathcal{N}$, however there are new minimal P-semiflows created, namely:

$$\begin{array}{ccccccc}
p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 \\
\langle 1, & 0, & 0, & 0, & 0, & 0, & 1 \ \rangle, \\
\langle 0, & 1, & 0, & 0, & 0, & 0, & 1 \ \rangle, \\
\langle 0, & 0, & 1, & 0, & 1, & 0, & 0 \ \rangle, \\
\langle 0, & 0, & 1, & 0, & 0, & 1, & 0 \ \rangle, \\
\langle 0, & 0, & 0, & 1, & 1, & 0, & 0 \ \rangle, \\
\langle 0, & 0, & 0, & 1, & 0, & 1, & 0 \ \rangle
\end{array}$$

We do not present a method to detect when and which new P-semiflows are created as a consequence of a transition synchronisation. However, from the analysis of different examples we are able to present a set of common characteristics identifying cases where new P-semiflows are created. Consider t_l and t_r the synchronising transitions and t_{syn} the transition resulting from their synchronisation. If there exists a $t \in T_{\mathcal{N}} - \{t_{syn}\}$ such that

1. The set of input places of t is a subset or equal to the set of output places of t_{syn}, the set of output places of t is a subset of set or equal to the set of input places of t_{syn} and

$$\forall p \in P_{\mathcal{N}}, (W_{\mathcal{N}}(p,t) = -W_{\mathcal{N}}(p,t_{syn})) \vee (W_{\mathcal{N}}(p,t) = 0) \tag{5.2}$$

2. it was not the case that in S, t already satisfied condition 5.2 with t_1 or t_2, and

3. the only transition with which t has common places is with t_{syn}.

Figure 5.6 shows two *cWN*s that satisfy the conditions stated above. The synchronisation of the transitions t_l and t_r in **A** will generate the P-semiflows

$$\begin{array}{cccc} p_a & p_b & p_1 & p_2 \\ \langle 1, & 0, & 0, & 1 \rangle \\ \langle 0, & 1, & 1, & 0 \rangle \end{array} \qquad \text{and}$$

not existent in **A**.

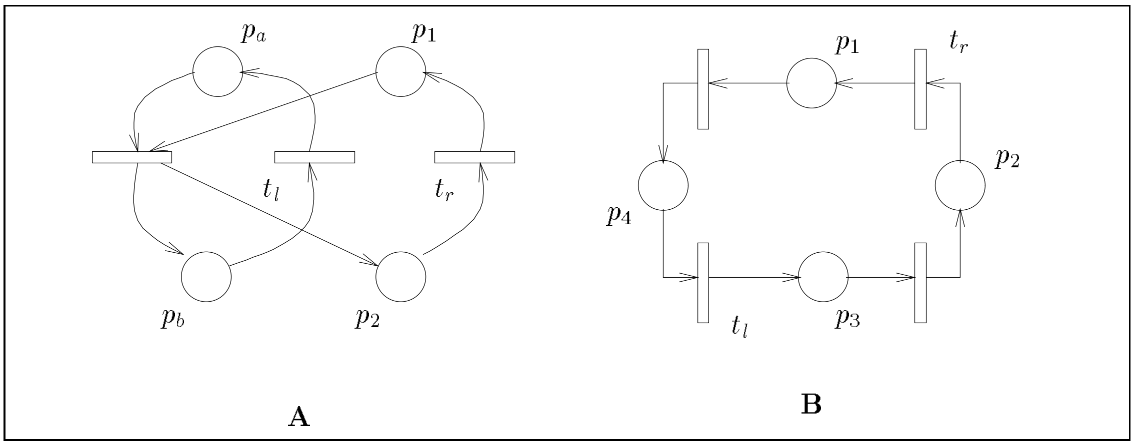


Figure 5.6: Example of a *cWN* S from which by applying transition synchronisation of t_1 and t_2, we obtain a *cWN* with other P-semiflows apart from those of S

In **B** the synchronisation of the transitions t_l and t_r generate de new P-semiflows:

$$\begin{array}{cccc} p_1 & p_2 & p_3 & p_4 \\ \langle 1, & 0, & 0, & 1 \rangle \\ \langle 0, & 1, & 1, & 0 \rangle \end{array} \qquad \text{and}$$

5.4.3 Example of the compositional construction of symbolic P-semiflows

Let us obtain the symbolic P-semiflows of the message communication system presented in Section 5.2.2.1, based on the generative families of minimal symbolic P-semiflows of each of the *bWN*s that compose it, as presented in Table 5.1. To facilitate the reading we will included once again in this section, in Table 5.4.3.

First we will compose the *bWN*s **B** and **D** into a component **BD** (see Figure 5.7) by the fusion of their respective places *T.Message buf.* The first symbolic

Minimal Symbolic P-semiflows of **A**	
Idle	T.Message buf
$\langle \lvert D_{2,1}\rvert \cdot S_{1,1}\rangle$	$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1}\rangle$
0	$\langle S_{1,1}, X_1^1, S_{2,1}\rangle$
0	$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2}\rangle$
0	$\langle S_{1,1}, X_1^1, S_{2,2}\rangle$

Minimal Symbolic P-semiflows of **D**	
T.Message buf	Idle
$\langle S_{1,1}, S_{1,1}, S_{2,2}\rangle$	$\langle \lvert D_{2,1}\rvert \cdot S_{1,1}\rangle$
$\langle S_{1,1}, X_1^1, S_{2,1}\rangle$	0
$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1}\rangle$	0

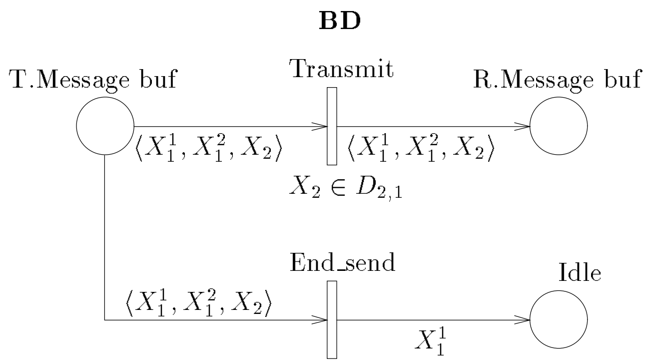
Minimal Symbolic P-semiflows of **B**	
R.Message buf	T.Message buf
$\langle S_{1,1}, X_1^1, S_{2,1}\rangle$	$\langle S_{1,1}, X_1^1, S_{2,1}\rangle$
$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1}\rangle$	$\langle S_{1,1},, S_{1,1} - X_1^1, S_{2,1}\rangle$
$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2}\rangle$	0
$\langle S_{1,1}, X_1^1, S_{2,2}\rangle$	0
0	$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2}\rangle$
0	$\langle S_{1,1}, X_1^1, S_{2,2}\rangle$

Minimal Symbolic P-semiflows of **C**	
T.Message buf	R.Message buf
$\langle S_{1,1}, X_1^1, \lvert D_{2,2}\rvert \cdot S_{2,1}\rangle$	$\langle S_{1,1}, X_1^1, \lvert D_{2,1}\rvert \cdot S_{2,2}\rangle$
$\langle S_{1,1}, S_{1,1} - X_1^1, \lvert D_{2,2}\rvert \cdot S_{2,1}\rangle$	$\langle S_{1,1}, S_{1,1} - X_1^1, \lvert D_{2,1}\rvert \cdot S_{2,2}\rangle$
$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2}\rangle$	0
$\langle S_{1,1}, X_1^1, S_{2,2}\rangle$	0
0	$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1}\rangle$
0	$\langle S_{1,1}, X_1^1, S_{2,1}\rangle$

Table 5.2: Symbolic P-semiflows of the *bWN*s **A**,**B**,**C** and **D**, modelling the basic functions of a message communication system.

P-semiflow is obtained from linear combination of the extensions of the first symbolic P-semiflow of **B** with the second symbolic P-semiflow of **D**, covering the objects with colours in the symbolic colour $\langle S_{1,1}, X_1^1, S_{2,1}\rangle$. The second symbolic P-semiflow is obtained from linear combination of the extensions of the second symbolic P-semiflow of **B** with the third symbolic P-semiflow of **D**, covering the objects with colours in the symbolic colour $\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1}\rangle$. The third symbolic P-semiflows is obtained from the linear combination of the extension of the symbolic P-semiflow resulting from the sum of the third and fourth symbolic P-semiflow of **B** with the extension of the third symbolic P-semiflow of **D**. Finally, the last two symbolic P-semiflows of **BD** are obtained from the extension of the last two symbolic P-semiflows of **B**, that assign zero to the entries of *T.Message buf*.

Following the compositional construction, we sequentially compose the place *R.Message buf* of **BD** with the place with the same name of *cWN* **C**, forming a new *cWN* **CBD**(see Figure 5.8). We will differentiate the places *T.Message buf*



Minimal Symbolic P-semiflows of **BD**		
T.Message buf	R.Message Buf	Idle
$\langle S_{1,1}, X_1^1, S_{2,1} \rangle$	$\langle S_{1,1}, X_1^1, S_{2,1} \rangle$	0
$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1} \rangle$	$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1} \rangle$	0
$\langle S_{1,1}, S_{1,1}, S_{2,2} \rangle$	0	$\langle \lvert D_{2,2} \rvert \cdot S_{1,1} \rangle$
0	$\langle S_{1,1}, X_1^1, S_{2,2} \rangle$	0
0	$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2} \rangle$	0

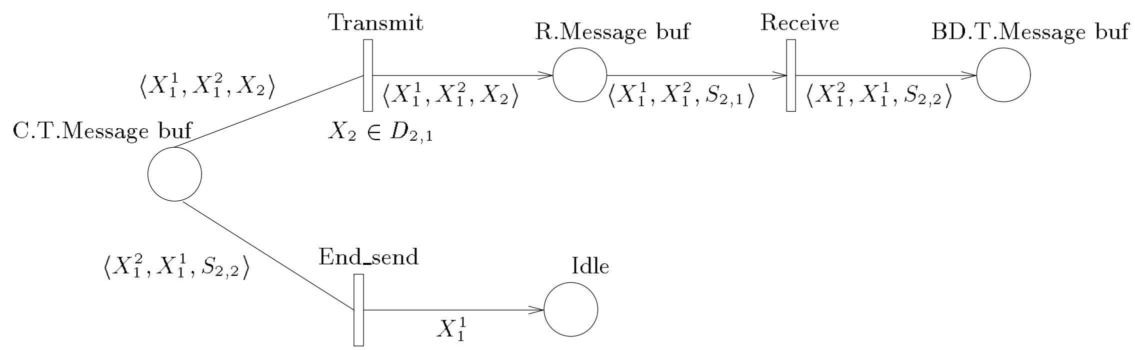
Figure 5.7: Minimal symbolic P-semiflows of the *cWN* **BD**.

using the dot notation introduced in Section 4.4. The first and second symbolic P-semiflows of **CBD** are obtained by the linear combination of the extensions of the first and second symbolic P-semiflows of **C** with the first and second symbolic P-semiflows of **BD**, respectively. The other P-semiflows are direct extensions of the remaining symbolic P-semiflows of **BD** and **C**.

We then go on to perform a closing operation over the component **CBD**, calling the resulting component **CBD′**(see Figure 5.9). The closing operation is defined over the places **C**.*T.Message buf* and **BD**.*T.Message buf*. The first symbolic P-semiflow is obtained from the linear combination of the first three and last two symbolic P-semiflows of component **CBD**. This was needed to obtain a symbolic P-semiflow of **CBD** that assigned the same value to the entries of **BD**.*T.Message buf* and **C**.*T.Message buf*.

Now we can sequentially compose component **A** with component **CBD′** (see Figure 5.10 on page 115), given that *T. Message* in **CBD′** is an entry place of the component. Again we differentiate the places with the same name by adding as a prefix the component they belong to. Notice that the entries of the *Idle* places are the same.

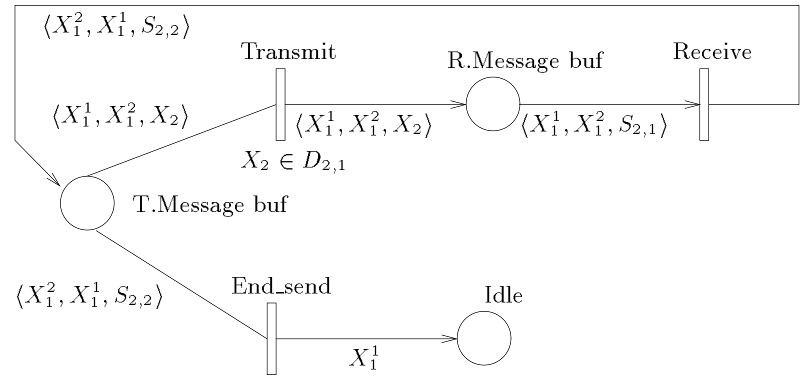
CBD



Minimal Symbolic P-semiflows of **CBD**			
BD.T.Message buf	R.Message Buf	Idle	C.T.Message buf
$\langle S_{1,1}, X_1^1, \lvert D_{2,2}\rvert S_{2,1}\rangle$	$\langle S_{1,1}, X_1^1, \lvert D_{2,2}\rvert S_{2,1}\rangle$	0	$\langle S_{1,1}, X_1^1, \lvert D_{2,1}\rvert S_{2,2}\rangle$
$\langle S_{1,1}, S_{1,1} - X_1^1, \lvert D_{2,2}\rvert S_{2,1}\rangle$	$\langle S_{1,1}, S_{1,1} - X_1^1, \lvert D_{2,2}\rvert S_{2,1}\rangle$	0	$\langle S_{1,1}, S_{1,1} - X_1^1, \lvert D_{2,1}\rvert S_{2,2}\rangle$
$\langle S_{1,1}, S_{1,1}, S_{2,2}\rangle$	0	$\langle \lvert D_{2,2}\rvert S_{1,1}\rangle$	0
0	$\langle S_{1,1}, X_1^1, S_{2,2}\rangle$	0	0
0	$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2}\rangle$	0	0
0	0	0	$\langle S_{1,1}, X_1^1, S_{2,1}\rangle$
0	0	0	$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,1}\rangle$

Figure 5.8: Minimal symbolic P-semiflows of the cWN **CBD**.

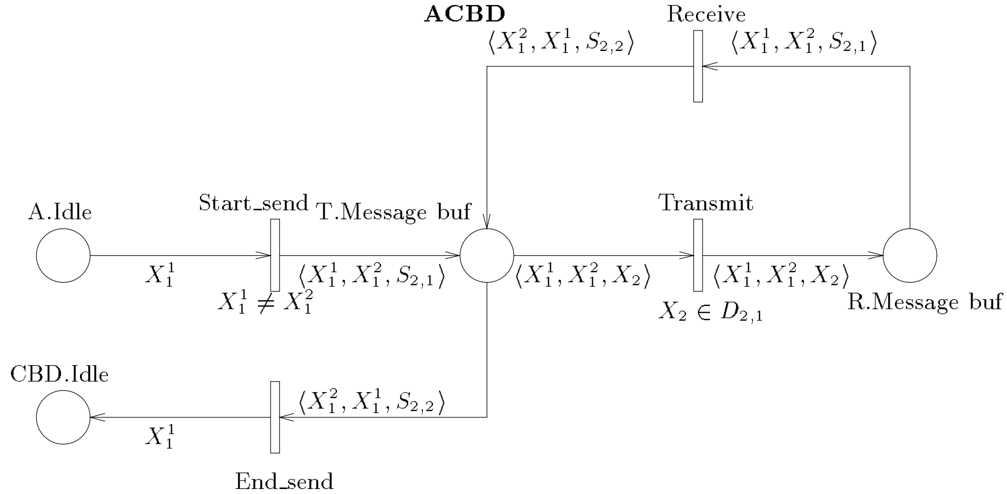
CBD′



Minimal Symbolic P-semiflows of **CBD′**		
T.Message buf	R.Message buf	Idle
$\langle S_{1,1}, S_{1,1}, \vert D_{2,2}\vert S_{2,1}\rangle + \langle S_{1,1}, S_{1,1}, \vert D_{2,1}\vert S_{2,2}\rangle$	$\langle S_{1,1}, S_{1,1}, \vert D_{2,2}\vert S_{2,1}\rangle$	$\langle \vert D_{2,1}\vert \vert D_{2,2}\vert S_{1,1}\rangle$
0	$\langle S_{1,1}, X_1^1, S_{2,2}\rangle$	0
0	$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2}\rangle$	0

Figure 5.9: Minimal symbolic P-semiflows of the *cWN* **CBD′**.

ACBD

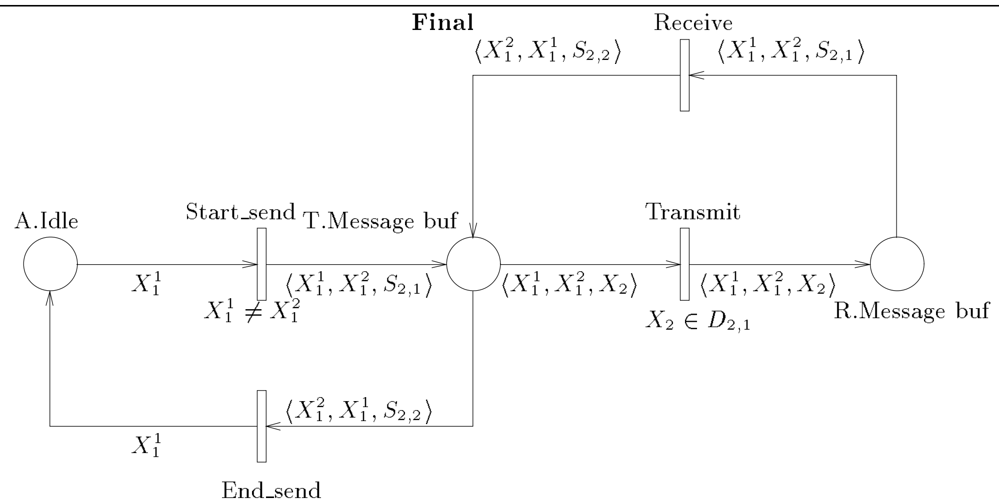


Minimal Symbolic P-semiflows of **ACBD**			
A.Idle	T.Message buf	R.Message buf	CBD′.Idle
$\langle \vert D_{2,1}\vert \vert D_{2,2}\vert S_{1,1}$	$\langle S_{1,1}, S_{1,1}, \vert D_{2,2}\vert S_{2,1}\rangle$ $+ \langle S_{1,1}, S_{1,1}, \vert D_{2,1}\vert S_{2,2}\rangle$	$\langle S_{1,1}, S_{1,1}, \vert D_{2,2}\vert S_{2,1}\rangle$	$\langle \vert D_{2,1}\vert \vert D_{2,2}\vert S_{1,1}\rangle$
0	0	$\langle S_{1,1}, X_1^1, S_{2,2}\rangle$	0
0	0	$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2}\rangle$	0

Figure 5.10: Minimal symbolic P-semiflows of the *cWN* **ACBD**.

Finally, we complete the composition of the model, by closing the component **ACBD**, obtained in the previous step. The places closed are the *Idle* places (see Figure 5.11 on page 116).

We have seen that it is possible to obtain the minimal symbolic P-semiflows of a *cWN*, based on the minimal symbolic P-semiflows of its components. The symbolic P-semiflows obtained cover all the places and all symbolic colours within the places. Therefore, the net is structurally bounded for all symbolic colours. Adding up the last two symbolic P-semiflows, we obtain a static P-semiflow. The first symbolic P-semiflow is also a static P-semiflow.



Minimal Symbolic P-semiflows of **Final**		
Idle	T.Message buf	R.Message Buf
$\langle \lvert D_{2,1}\rvert \lvert D_{2,2}\rvert S_{1,1}\rangle$	$\langle S_{1,1}, S_{1,1}, \lvert D_{2,2}\rvert S_{2,1}\rangle + \langle S_{1,1}, S_{1,1}, \lvert D_{2,1}\rvert S_{2,2}\rangle$	$\langle S_{1,1}, S_{1,1}, \lvert D_{2,2}\rvert S_{2,1}\rangle$
0	0	$\langle S_{1,1}, X_1^1, S_{2,2}\rangle$
0	0	$\langle S_{1,1}, S_{1,1} - X_1^1, S_{2,2}\rangle$

Figure 5.11: Minimal symbolic P-semiflows of the *cWN* **Final**.

5.5 Compositional construction of T-semiflows of *cWN*s

Unlike P-semiflows, it is not possible to obtain T-semiflows of a *bWN*. Let us remember that in *bWN*s the sets of entry places and final places cannot intersect; thus it is never possible to return to the initial marking after the firing of sequence of instances of the *bWN*'s transition. Therefore, a *bWN* has as symbolic T-semiflows the trivial T- semiflow $\vec{0}$ and any T-semiflow that only covers invalid instances of the transition.

In this section we will study how to combine T-semiflows of two *cWN*s to obtain T-semiflows of the *cWN* resulting from their composition. As we have pointed out in the previous section, it is not the objective of this work to propose a new method for the construction of T-semiflows of WNs. We will therefore start from the premise that there is a method by which we can find the generative family of symbolic T-semiflows of a *cWN*.

Following the same pattern as for the compositional construction of the incidence matrix and of the P-semiflows, let us analyse each type of compositional operation.

First, let us consider a *cWN* $\mathcal{N}$ obtained from the composition of two *cWN*s L and R, and a pair of T-semiflows x of L and y of R. We will define the *extension* of x with respect R—denoted $\widehat{x}^R$— as a vector of dimension $|T_L \cup T_R|$, given by:

$$\forall t \in T_L \cup T_R, \qquad \widehat{x}^R(t) = \begin{cases} x(t) & \text{if } t \in T_L \\ 0 & \text{otherwise} \end{cases}$$

The dual definition applies for the extension of y with respect L.

5.5.1 Under independent parallel composition

As we know, in independent parallel composition there is no place or transition fusion. Therefore, by simply extending the T-semiflows to cover the transitions of the *cWN*s being composed, it is possible to obtain the T-semiflows of the higher level component.

Proposition 5.11 *Given the generative family of T-semiflows of two cWNs, L and R, the generative family of T-semiflows of $\mathcal{N}$—the cWN resulting from the independent parallel composition of L and R—is formed from the extensions of the T-semiflows of the generative families of L and R.*

Proof.- By contradiction, let us suppose that there is a minimal T-semiflow z of $\mathcal{N}$ such that it is not an extension of a minimal T-semiflow of L or of R, i.e. it is not in the generative family of T-semiflows of $\mathcal{N}$ obtained as described in the proposition.

If z only covers the transitions of L, then given that it is minimal, it must correspond to the extension of a minimal symbolic T-semiflows of L, i.e. it belongs to the generative family of symbolic T-semiflows of $\mathcal{N}$. The same applies if it covers only places of R.

If it covers places of both L and R, then it is possible to create two symbolic T-semiflows z_1 and z_2 of $\mathcal{N}$ defined as follows:

$$\forall t \in T_L \cup T_R,$$
$$z_1(t) = \begin{cases} z(t) & \text{if } t \in T_L \\ 0 & \text{otherwise} \end{cases}$$
$$z_2(t) = \begin{cases} z(t) & \text{if } t \in T_R \\ 0 & \text{otherwise} \end{cases}$$

such that $\|z_1\| \subset \|z\|$ and $\|z_2\| \subset \|z\|$, which means that z is not minimal. ■

5.5.2 Under choice composition

As in the case of compositional construction of P-semiflows, it is necessary to define another extension for T-semiflows of the components participating in a choice composition. A choice composition adds two new transitions to the set of transitions, therefore the extension of T-semiflows of the sub-components must consider these new transitions in order to cover all transitions of the resulting *cWN* $\mathcal{N}$. Consider the choice composition of two *cWN*s, L and R, and the resulting *cWN* $\mathcal{N}$. Given a T-semiflow x of L we define the *choice extension* of x—denoted $\widehat{x_c^R}$—with respect to the transitions in R as a vector of dimension $|T_{\mathcal{N}}|$, whose entries are given by:

$$\forall t \in T_{\mathcal{N}},\ \widehat{x_c^R}(t) = \begin{cases} x(t) & \text{if } t \in T_L \\ 0 & \text{otherwise} \end{cases}$$

The dual applies for the T-semiflows of R.

Recall the structure of the *cWN* $\mathcal{N}$ resulting from the choice composition of two *cWN*s L and R. Figure 5.12 shows an example of a choice composition (the arc functions have been omitted, since they are not relevant for the study).

Notice that if either t_l or t_r fire, then it is not possible to return to the initial marking of p_c. Therefore, it is not possible to define T-semiflows of $\mathcal{N}$ that cover either of the transitions t_l or t_r. The T-semiflows of $\mathcal{N}$ will be formed by the choice extensions of the T-semiflows of L and R, which assign zero to the entries corresponding to the transitions t_l and t_r.

5.5.3 Under place fusion operations

5.5.3.1 Sequential composition

Consider the sequential composition of two *cWN*s L and R to form a *cWN* $\mathcal{N}$. For this it is necessary to select a subset FS' of the final places of L and a subset

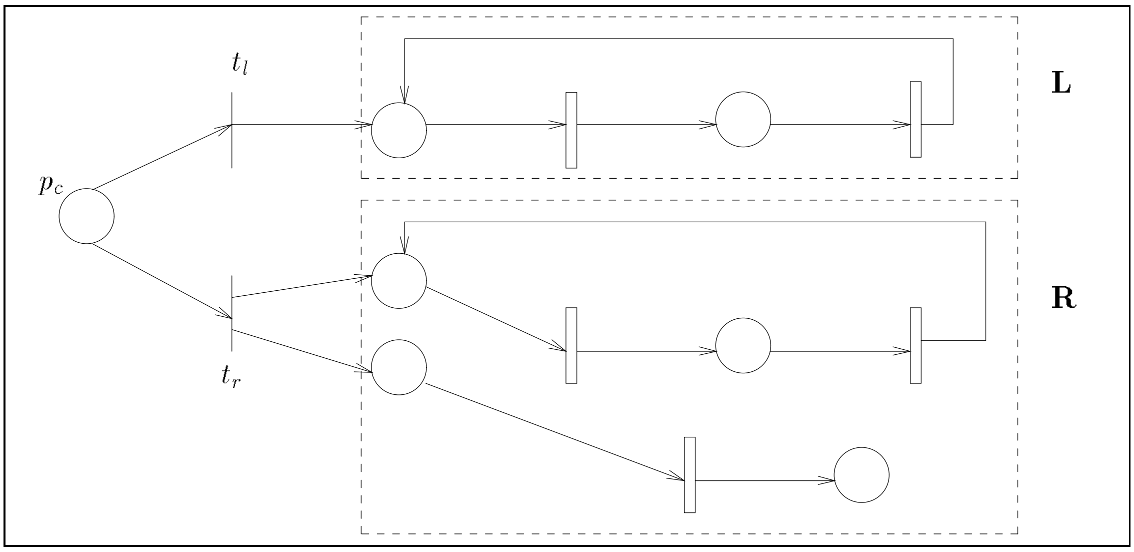


Figure 5.12: Structure of a cWN obtained by choice composition.

ES' of the entry places of R, and define a correspondence function between these two sets.

The T-semiflows of L cannot cover transitions that have output places belonging to FS_L, since firing these transitions will create a marking from which the initial marking is not accessible. However, the T-semiflows of R can contain transitions whose input places belong to the entry set (ES_R).

Firing a transition $t_r \in T_R$ can only change the marking of a place $p_f \in P_L$ if p_f has been fused with a place $p_e \in P_R$, and the firing of t_r changes the marking of p_e in R. Similarly, firing a transition $t_l \in T_L$ can only alter the marking of a place $p_e \in P_R$, if p_e has been fused with a place $p_f \in P_L$ whose marking is changed when transition t_l is fired. This means that the only places in P_L whose markings could be altered by a transition $t_r \in T_R$ are those in FS' and the only places in P_R whose markings could be altered by a transition $t_l \in T_L$ are those in ES'.

These observations lead us to conclude that the set of minimal T-semiflows of $\mathcal{N}$ is formed by the extension of the minimal T-semiflows of L and R.

5.5.3.2 Competing parallelism

An external competing parallelism can always be seen as a independent parallel composition followed by an internal competing parallelism. Therefore, we will limit our analysis to internal competing parallelism.

Internal competing parallelism is reflected at the level of the incidence matrix as the sum of the corresponding entries of the places being fused. Let us remember that it is not possible to fuse places that share input or output transitions. So consider a cWN $\mathcal{S}$ and the fusion of two of its places p_1 and p_2. Let us denote by T_1 the set of input and output transitions of p_1, by T_2 the set of input and output transitions of p_2, and by T_{rst} the rest of the transitions of the cWN. The incidence matrix can then be structured in the following manner:

$$\begin{array}{r c} & \begin{array}{ccc} T_1 & T_2 & T_{rst} \end{array} \\ \begin{array}{r} P-\{p_1,p_2\} \\ p_1 \\ p_2 \end{array} & \begin{pmatrix} W_1^P & W_2^P & W_{rst}^P \\ w_1 & 0 & 0 \\ 0 & w_2 & 0 \end{pmatrix} \end{array}$$

The incidence matrix of the resulting cWN $\mathcal{N}$ after the fusion of the place p_1 and p_2 into a place $p_{1,2}$, will then have the form:

$$\begin{array}{r c} & \begin{array}{ccc} T_1 & T_2 & T_{rst} \end{array} \\ \begin{array}{r} P-\{p_1,p_2\} \\ p_{1,2} \end{array} & \begin{pmatrix} W_1^P & W_2^P & W_{rst}^P \\ w_1 & w_2 & 0 \end{pmatrix} \end{array}$$

By studying the structure of the incidence matrix we can observe that a T-semiflow of $\mathcal{S}$ will also be a T-semiflow of $\mathcal{N}$. Given x, a T-semiflow of $\mathcal{S}$, we know that it holds that:

$$\forall t \in T_1,\ w_1(t) \cdot x(t) = 0 \text{ and } \forall t \in T_2,\ w_2(t) \cdot x(t) = 0$$

Therefore, $W_{\mathcal{N}} \cdot x = 0$. However, it is possible that new semiflows are created as in the case of P-semiflows under transition synchronisation. Observe that there can be a vector that anulls the row of $p_{1,2}$ and the rows corresponding to the places in $P - \{p_1, p_2\}$, but that does not anull the row corresponding to p_1 or to p_2. The following cWN is an example of such a case, where p_1 and p_2 are the places participating in the competing parallelism.

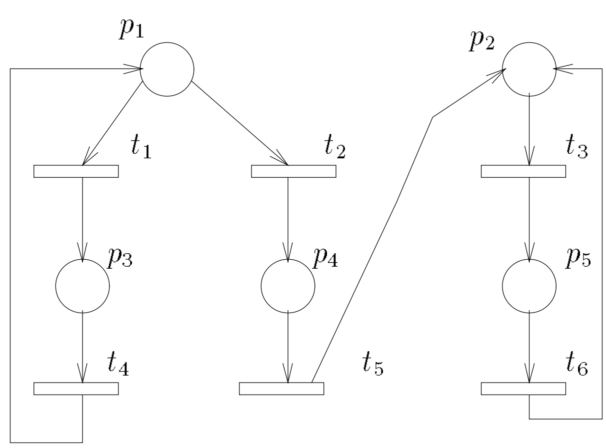


Figure 5.13: Example of a cWN for which if an internal competing parallelism operation is applied, new T-semiflows (different from the existing) are created.

The fusion of the places p_1 and p_2 generates a new T-semiflow that assigns a count of 1 to transitions t_2 and t_5 and 0 to all other transitions.

As for the case of P-semiflows under transition synchronisation we have identified the characteristics of a group of *cWN*s for which new competing parallelism composition results in the generation of new T-semiflows. It must be made clear that this is not necessarily the only group that generates this situation. Consider the places p_1 and p_2, and $p_{1,2}$ the place resulting from their fusion. If there exists a $p \in P_{\mathcal{N}} - \{p_{1,2}\}$ such that

1. The set of input transitions of p is a subset or equal to the set of output transitions of $p_{1,2}$ and the set of output transitions of p is a subset or equal to the set of input transitions of $p_{1,2}$,
$$\forall t \in T_{\mathcal{N}}, (W_{\mathcal{N}}(p,t) = -W_{\mathcal{N}}(p_{1,2},t)) \vee (W_{\mathcal{N}}(p,t) = 0) \tag{5.3}$$
2. It was not the case that in S p already satisfied condition 5.3 with p_1 or p_2, and
3. the only place with which p has common transitions is $p_{1,2}$.

The example given in Figure 5.13 satisfies these conditions, where $p = p_3$.

5.5.3.3 Closing operation

The closing operation fuses a final place with an entry place within a *cWN*. None of the other composition operations considers the fusion of pairs of final places, therefore a final place can only be an output place of a single transition. Two final places of the same transition cannot be fused with a single entry place or else we would be creating parallel arcs.

Consider the places p_e and p_f involved in the closing operation over a *cWN* $\mathcal{S}$, and the incidence matrix $W_{\mathcal{S}}$, represented in such a way that the last two place entries of the matrix correspond to the entries of the places p_e and p_f, respectively, and the last transition column corresponds to the transition t_f (the transition for which p_f is an output place).

$$W_{\mathcal{S}} = \begin{pmatrix} W' & W_{t_f} \\ w'_{p_e} & w_{t_f}(p_e) \\ 0 & w_{t_f}(p_f) \end{pmatrix}$$

We must remember that p_f cannot be input to any transition.

At the level of the incidence matrix, the fusion of p_f with p_e is reflected as the sum of their corresponding place entries, giving:

$$W_{\mathcal{N}} = \begin{pmatrix} W' & W_{t_f} \\ w'_{p_e} & w_{t_f}(p_e) + w_{t_f}(p_f) \end{pmatrix}$$

A T-semiflow x of $\mathcal{S}$ satisfies the condition:

$$\left(\sum_{t \in T_S - \{t_f\}} w'_{p_e}(t) \cdot x(t)\right) + w_{t_f}(p_e) \cdot x(t_f) = 0 \quad \text{and} \quad w_{t_f}(p_f) \cdot x(t_f) = 0$$

Therefore, it also satisfies the condition:

$$\left(\sum_{t \in T_S - \{t_f\}} w'_{p_e}(t) \cdot x(t)\right) + (w_{t_f}(p_e) + w_{t_f}(p_f)) \cdot x(t_f) = 0$$

We can then conclude that x is also a T-semiflow of $\mathcal{N}$. Unfortunately, under the closing operation it is not possible to conclude, as for the other operations involving place fusion, that the set of minimal T-semiflows of the resulting *cWN* is formed directly from the T-semiflows of the original *cWN* $\mathcal{S}$. The fusion of an entry place with a final place, may imply the existence of new T-semiflows in the resulting net. A necessary condition for the existence of a T-semiflow, different from the trivial solution, is that there exists a set T_{sem} of transitions such that $T_{sem}^{\bullet} = {}^{\bullet}T_{sem}$. Only if this condition holds, is it possible to return the initial marking of a place to its original state after the firing of the transition sequence defined for T_{sem}. This means that only when performing a closing operation over a *cWN* can we obtain T-semiflows of the resulting $\mathcal{N}$ that were not derived from the existing ones in the sub-components.

5.5.4 Under synchronisation

So far, the analysis of compositional construction of T-semiflows has not required us to consider the colours of the *cWN*s. However, for synchronisation operations this is not the case. The transitions that are synchronised can have different colour domains, each defined by the set of variables or parameters of the transitions. The transition resulting from the synchronisation of two transitions, will "inherit" the variables of those transitions. Therefore, its colour domain will be formed by the Cartesian product of the colour domains of the transitions that compose it.

As for the analysis of P-semiflows for composition operations involving place fusion, the synchronisation of two *cWN*s can be seen as the parallel composition of the *cWN*s followed by an internal synchronisation of a pair of transitions. In this way we can concentrate on studying what happens in the case of synchronisation of transitions within a *cWN*.

Given the generative family of symbolic T-semiflows of the *cWN* $\mathcal{S}$, let us consider how to obtain T-semiflows of the *cWN* $\mathcal{N}$ resulting from the fusion of the transitions t_l and t_r in $T_{\mathcal{S}}$. From Section 5.3.4 we know that the incidence matrix of $\mathcal{N}$ can be obtained in the following way.

$$\forall p \in P_{\mathcal{N}}, \forall t \in T_{\mathcal{N}}, \forall c \in C(p), \forall c' \in C(t),$$

$$W_{\mathcal{N}}(p,t)(c,c') = \begin{cases} W_{\mathcal{S}}(p,t_l)(c,c_l) & \text{if } t = t_{lr} \ \wedge \ p \in {}^\bullet t_l \cup t_l^\bullet \ \text{ where } c' = \langle c_l, c_r \rangle \\ W_{\mathcal{S}}(p,t_r)(c,c_r) & \text{if } t = t_{lr} \ \wedge \ p \in {}^\bullet t_r \cup t_r^\bullet \ \text{ where } c' = \langle c_l, c_r \rangle \\ 0 & \text{if } t = t_{lr} \wedge \ p \notin ({}^\bullet t_l \cup t_l^\bullet \cup {}^\bullet t_r \cup t_r^\bullet \ ) \\ W_{\mathcal{S}}(p,t)(c,c') & \text{otherwise} \end{cases}$$

The trivial case would be those minimal symbolic T-semiflows of $\mathcal{S}$ that do not cover either of the transitions being fused. These T-semiflows will constitute minimal symbolic T-semiflows of the resulting cWN $\mathcal{N}$.

A much more difficult case is that of minimal symbolic T-semiflows that cover both of the transitions being fused.

Consider a minimal symbolic T-semiflow x of $\mathcal{S}$ that satisfies that $x(t_l) > 0$ and $x(t_r) > 0$, and

$$\sum_{\bar{c} \subset C(t_l)} x(t_l, \bar{c}) = \sum_{\bar{c}' \subset C(t_r)} x(t_r, \bar{c}') \tag{5.4}$$

From the entries of x for the symbolic colours of t_r and t_l we can define the following multisets on $C(t_l)$ and $C(t_r)$, respectively:

$$cov(t_l) = \{x(t_l, \bar{c}_l) \cdot \bar{c}_l \mid \bar{c}_l \subset C(t_l)\}$$

and

$$cov(t_r) = \{x(t_r, \bar{c}_r) \cdot \bar{c}_r \mid \bar{c}_r \subset C(t_r)\}$$

Given the condition 5.4 we know that these multisets will have the same cardinality. From these multisets we can obtain a set of multisets COV defined as the set of all possible multisets formed by combinations of elements in $cov(t_l)$ with elements in $cov(t_r)$. To clarify the concept, let us view the sets $cov(t_l)$ and $cov(t_r)$ as bags with coloured balls. The balls in the bags can have different colours, and there can be more than one ball per colour. A multiset of COV will correspond to the result of an experiment that takes $|cov(t_l)|$ pairs of balls—formed by one of $cov(t_l)$ and one of $cov(t_r)$.

Consider $cov(t_{lr})$ a multiset in COV, where it holds that

$$\forall \bar{c}_l \subset C(t_l), \sum_{\bar{c}_r \subset C(t_r)} cov(t_{lr})(\langle \bar{c}_l, \bar{c}_r \rangle) = cov(t_l)(\bar{c}_l)$$

and

$$\forall \bar{c}_r \subset C(t_r), \sum_{\bar{c}_l \subset C(t_l)} cov(t_{lr})(\langle \bar{c}_l, \bar{c}_r \rangle) = cov(t_r)(\bar{c}_r)$$

and let us denote by $comb(\bar{c}) \in \mathbb{N}$ the multiplicity of the symbolic colour $\bar{c} \subset C(t_{lr})$ in the multiset $cov(t_{lr})$. We can apply the following proposition to obtain a symbolic T-semiflows z of $\mathcal{N}$.

Proposition 5.12 *Given a minimal symbolic T-semiflow x of $\mathcal{S}$, such that $x(t_l) > 0$ and $x(t_r) > 0$, and*

$$\sum_{\bar{c} \subset C(t_l)} x(t_l, \bar{c}) = \sum_{\bar{c}' \subset C(t_r)} x(t_r, \bar{c}')$$

we can then define a minimal symbolic T-semiflow z of $\mathcal{N}$ in the following way:

$$\forall t \in T_{\mathcal{N}}, \forall \bar{c} \subset C(t),$$
$$z(t, \bar{c}) = \begin{cases} comb(\bar{c}) & \textit{if } t = t_{lr} \textit{ and } \bar{c} \in cov(t_{lr}) \\ 0 & \textit{if } t = t_{lr} \textit{ and } \bar{c} \notin cov(t_{lr}) \\ x(t, \bar{c}) & \textit{otherwise} \end{cases}$$

Proof.- For a vector z—as defined above—to be a T-semiflow of $\mathcal{N}$ it must hold that:

$$W_{\mathcal{N}} \cdot z = \left(\sum_{t \in T - \{t_{lr}\}} W_{\mathcal{N}}(\cdot, t) \cdot z(t) \right) + W_{\mathcal{N}}(\cdot, t_{lr}) \cdot z(t_{lr}) = 0$$

By definition of z and from the definition of $W_{\mathcal{N}}$ we know that:

$$W_{\mathcal{N}} \cdot z = \left(\sum_{t \in T - \{t_{lr}\}} W_{\mathcal{S}}(\cdot, t) \cdot x(t) \right) + \left(\sum_{\bar{c} \in cov(t_{lr})} W_{\mathcal{N}}(\cdot, t_{lr})(\cdot, \bar{c}) \cdot comb(\bar{c}) \right)$$

Given the definition of the incidence matrix of $\mathcal{N}$, with respect to the incidence matrix of $\mathcal{S}$, we then have that for $p \in {}^\bullet t_l \cup t_l^\bullet$:

$$\sum_{\bar{c} \in cov(t_{lr})} W_{\mathcal{N}}(p, t_{lr})(\cdot, \bar{c}) \cdot z(t_{lr}, \bar{c}) = \sum_{\langle \bar{c}_l, \bar{c}_r \rangle \in cov(t_{lr})} W_{\mathcal{S}}(p, t_l)(\cdot, \bar{c}_l) \cdot comb(\langle \bar{c}_l, \bar{c}_r \rangle)$$

Recall that a symbolic colour $\bar{c} \in C(t_{lr})$ can be represented as a pair formed by a symbolic colour in $C(t_l)$ and a symbolic colour in $C(t_r)$. Now, for all multisets in COV it must hold that for every $\bar{c}_l \subset C(t_l)$ such that $x(t_l, \bar{c}_l) > 0$,

$$\sum_{\bar{c}_r \subset C(t_r)} comb(\langle \bar{c}_l, \bar{c}_r \rangle) = x(t_l, \bar{c}_l)$$

Therefore,

$$\sum_{\langle \bar{c}_l, \bar{c}_r\rangle \in cov(t_{lr})} W_{\mathcal{S}}(p,t_l)(\cdot,\bar{c}_l)\cdot comb(\langle \bar{c}_l, \bar{c}_r\rangle)$$

$$\begin{aligned} &= \sum_{\bar{c}_l \in cov(t_l)} \sum_{\bar{c}_r \in cov(t_r)} W_{\mathcal{S}}(p,t_l)(\cdot,\bar{c}_l)\cdot comb(\langle \bar{c}_l, \bar{c}_r\rangle) \\ &= \sum_{\bar{c}_l \in cov(t_l)} W_{\mathcal{S}}(p,t_l)(\cdot,\bar{c}_l)\cdot \sum_{\bar{c}_r \in cov(t_r)} comb(\langle \bar{c}_l, \bar{c}_r\rangle) \\ &= \sum_{\bar{c}_l \in cov(t_l)} W_{\mathcal{S}}(p,t_l)(\cdot,\bar{c}_l)\cdot x(t_l,\bar{c}_l) \end{aligned}$$

The same holds for every symbolic colour of t_r. Given that t_l and t_r have no input or output places in common and since x is a minimal symbolic T-semiflow of $\mathcal{S}$, then :

$$W_{\mathcal{N}} \cdot z = 0$$

■

This proposition applies in the case when the transitions that are to be synchronised are covered by the same minimal symbolic T-semiflow. This implies that the firing of one of them affects the enabling condition of the other in an indirect manner. It cannot be direct, since the transitions are not allowed to have common input or output places.

As we have mentioned, a T-semiflow will identify a set of transitions T_{sem} such that ${}^{\bullet}T_{sem} = T_{sem}^{\bullet}$. If this condition did not hold, then the firing of a transition in the sequence could alter the marking of a place that is not input or output of any other transition in the sequence. In consequence its marking could not be returned to its original value. Thus, in the case that one of the transitions being synchronised is covered by a minimal symbolic T-semiflow but the other transition is not, the transition resulting from the synchronisation is not covered by a minimal symbolic T-semiflow.

Let us now analyse the case when the transitions being fused are covered by different minimal symbolic T-semiflows. Similarly to the case of both transitions being covered by the same T-semiflow, we need to obtain symbolic T-semiflows where the sum of the weights assigned to the symbolic colours of a transition being synchronised equals the equivalent sum for the transition with which it synchronises.

Consider a pair of transitions t_l and t_r of a *cWN* $\mathcal{S}$, and two minimal symbolic T-semiflows of $\mathcal{S}$, namely x and y, such that $x(t_l) > 0$, $y(t_r) > 0$ and $\forall t \in T_{\mathcal{S}}, x(t) \cdot y(t) = 0$, i.e. the sets of transitions covered by x and y are disjoint. Let us obtain the following values:

$$cnt_x = \sum_{\bar{c} \subset C(t_l)} x(t_l, \bar{c}) \qquad cnt_y = \sum_{\bar{c}' \subset C(t_r)} y(t_r, \bar{c}')$$
$$div = lcm(cnt_x, cnt_y)$$

where *lcm* is the *least common multiple* function. Let us create two symbolic T-semiflows x' and y' of $\mathcal{S}$, defined as follows:

$$\forall t \in T_{\mathcal{S}},\ \forall \bar{c} \subset C(t),\ \ x'(t, \bar{c}) = \frac{div}{cnt_x} \cdot x(t, \bar{c})$$
$$\forall t \in T_{\mathcal{S}},\ \forall \bar{c} \subset C(t),\ \ y'(t, \bar{c}) = \frac{div}{cnt_y} \cdot y(t, \bar{c})$$

In this way we have created a pair of symbolic T-semiflows such that

$$\sum_{\bar{c}_l \subset C(t_l)} x'(t_l, \bar{c}_l) = \sum_{\bar{c}_r \subset C(t_r)} y'(t_r, \bar{c}_r)$$

i.e. the sum of the weights assigned to the symbolic colours of the transition t_l by x' equals the sum of the weights assigned to the symbolic colours of t_r by y'. Similarly to the case of a fusion of transitions covered by the same T-semiflow, let us consider the multisets

$$cov(t_l) = \{x'(t_l, \bar{c}_l) \cdot \bar{c}_l \mid \bar{c}_l \subset C(t_l)\}$$

and

$$cov(t_r) = \{y'(t_r, \bar{c}_r) \cdot \bar{c}_r \mid \bar{c}_r \subset C(t_r)\}$$

It is then possible to obtain T-semiflows of $\mathcal{N}$ applying the following proposition.

Proposition 5.13 *For every multiset*

$$cov(t_{lr}) = \{comb(\bar{c}_{lr}) \cdot \bar{c}_{lr} \mid \bar{c}_{lr} = \langle \bar{c}_l, \bar{c}_r \rangle \subset C(t_{lr})\}$$

in COV (the set of multisets obtained from all possible combinations of elements of $cov(t_l)$ and $cov(t_r)$), where $comb(\bar{c}_{lr}) \in \mathbb{N}$ is the cardinality of the symbolic colour $\bar{c}_{lr} \subset C(t_{lr})$ it is possible to define a symbolic T-semiflow z of $\mathcal{N}$ by:

$$\forall t \in T_{\mathcal{N}}, \forall \bar{c} \subset C(t),$$
$$z(t, \bar{c}) = \begin{cases} comb(\bar{c}) & \text{if } t = t_{lr} \text{ and } \bar{c} \in cov(t_{lr}) \\ 0 & \text{if } t = t_{lr} \text{ and } \bar{c} \notin cov(t_{lr}) \\ y'(t, \bar{c}) + x'(t, \bar{c}) & \text{otherwise} \end{cases}$$

where x' and y' are the T-semiflows, obtained from the T-semiflows x and y, that satisfy the condition that the sum of the weights that they assign to the symbolic colours of t_l and t_r, respectively, are equal.

Proof.- For the vector z to be a T-semiflow of $\mathcal{N}$ it must hold that:

$$W_{\mathcal{N}} \cdot z = \left(\sum_{t \in T - \{t_{lr}\}} W_{\mathcal{N}}(\cdot, t) \cdot z(t) \right) + W_{\mathcal{N}}(\cdot, t_{lr}) \cdot z(t_{lr}) = 0$$

According to the definition of z and of $W_{\mathcal{N}}$ we then have that:

$$W_{\mathcal{N}} \cdot z = \left(\sum_{t \in T - \{t_{lr}\}} W_{\mathcal{S}}(\cdot, t) \cdot ((x'(t) + y'(t)) \right) + \sum_{\bar{c} \in cov(t_{lr})} W_{\mathcal{N}}(\cdot, t_{lr})(\cdot, \bar{c}) \cdot comb(\bar{c})$$

Given the definition of the incidence matrix of $\mathcal{N}$ with respect to the incidence matrix of $\mathcal{S}$, we then have that for $p \in {}^{\bullet}t_l \cup t_l^{\bullet}$:

$$\sum_{\bar{c} \in cov(t_{lr})} W_{\mathcal{N}}(p, t_{lr})(\cdot, \bar{c}) \cdot comb(\bar{c}) = \sum_{\langle \bar{c}_l, \bar{c}_r \rangle \in cov(t_{lr})} W_{\mathcal{S}}(p, t_l)(\cdot, \bar{c}_l) \cdot comb(\langle \bar{c}_l, \bar{c}_r \rangle)$$

Now, for all multisets in COV it must hold that for every $\bar{c}_l \in C(t_l)$ such that $x'(t_l, \bar{c}_l) > 0$,

$$\sum_{\bar{c}_r \subset C(t_r)} comb(\langle \bar{c}_l, \bar{c}_r \rangle) = x'(t_l, \bar{c}_l)$$

Therefore,

$$\sum_{\langle \bar{c}_l, \bar{c}_r \rangle \in cov(t_{lr})} W_{\mathcal{S}}(p, t_l)(\cdot, \bar{c}_l) \cdot comb(\langle \bar{c}_l, \bar{c}_r \rangle) = \sum_{\bar{c}_l \in cov(t_l)} W_{\mathcal{S}}(p, t_l)(\cdot, \bar{c}_l) \cdot x'(t_l, \bar{c}_l)$$

The same analysis can be applied for t_r with y'. Given that t_l and t_r have no input or output places in common and that x is a minimal symbolic T-semiflow of $\mathcal{S}$, then :

$$W_{\mathcal{N}} \cdot z = 0$$

Let us remember that x and y cover different transitions, therefore so will x' and y'. ■

Furthermore, we can prove any minimal symbolic T-semiflow of $\mathcal{N}$ can be obtained from the symbolic T-semiflows of $\mathcal{S}$, using Propositions 5.12 and 5.13.

By contradiction, let us suppose that there is a minimal symbolic T-semiflow z of $\mathcal{N}$ that cannot be obtained from the symbolic P-semiflows of $\mathcal{S}$ with the propositions presented. For all symbolic colours $\bar{c}_l$ of t_l let us obtain the following value:

$$cnt(t_l, \bar{c}_l) = \sum_{\bar{c}_r \subset C(t_r)} z(t_{lr}, \langle \bar{c}_l, \bar{c}_r \rangle)$$

and similarly for all symbolic colours $\bar{c}_r \subset t_r$,

$$cnt(t_r, \bar{c}_r) = \sum_{\bar{c}_l \subset C(t_l)} z(t_{lr}, \langle \bar{c}_l, \bar{c}_r \rangle)$$

Consider a vector v of dimension $|T_{\mathcal{S}}|$ such that $\forall t \in T_{\mathcal{S}}$,

$$v(t) = \begin{cases} z(t) & \text{if } t \in T_{\mathcal{S}} - \{t_l, t_r\} \\ cnt(t_l, \bar{c}_l) & \text{if } t = t_l \\ cnt(t_r, \bar{c}_r) & \text{if } t = t_r \end{cases}$$

Then v is a symbolic T-semiflow of $\mathcal{S}$ that either satisfies Proposition 5.12 or it can be obtained by applying Proposition 5.13.

5.6 Conclusions

In this chapter we have studied the compositional construction of semiflows of *cWN*s. New types of semiflows for WNs have been defined, namely *symbolic* and *static* P-semiflows and *symbolic* T-semiflows, in order to exploit the characteristics of WNs. Symbolic semiflows are defined over *symbolic colours* of the colour domains of places and transitions. The definition of symbolic P-semiflows presented here differs from the existing definition of symbolic P-semiflows, mainly in that they assign equal value to all colours within a symbolic colour, not distinguishing between individual colours within a symbolic colour. For each compositional operation we have analysed and proposed a method to obtain the set of minimal symbolic and static P-semiflows and symbolic T-semiflows of the resulting *cWN*, based on the minimal P- and T- semiflows (of the same types) of its sub-components. These analyses are based on the relationship of the incidence matrix of the resulting component to the incidence matrix(ces) of the *cWN*(s) being composed.

The number of symbolic P-semiflows that have to be handled is, in principle, considerably less than if we calculated P-semiflows based on individual colours. The methods proposed allow the use of general arc functions. However, it is not the aim of this work to offer an alternative method to obtain P-semiflows of a CP-net or WNs in general. The propositions made are tightly coupled with the composition operations defined for *cWN*.

The compositional construction of symbolic (static) P-semiflows does not require us to start from *bWN*s. The methods proposed can be applied over higher level (non-*bWN*) components for which the set of minimal symbolic P-semiflows are known. However, to complement the method proposed we present an algorithm to obtain the symbolic P-semiflows of a *bWN*. This algorithm can be adapted to obtain the static P-semiflows instead.

A *bWN* has only one symbolic T-semiflow, namely the trivial symbolic T-semiflow $\vec{0}$. The methods proposed for the compositional construction of symbolic

T-semiflows require the set of minimal symbolic T-semiflows of the *cWN*s being composed. It is beyond the scope of this dissertation to offer a method to calculate such T-semiflows. Since for a T-semiflow to exist there must exist a set T_{sem} of transitions such that $T_{sem}^{\bullet} = {}^{\bullet}T_{sem}$, we can limit the search for new minimal T-semiflows to cases when a closing operation is performed.

To our knowledge there is no existing work that considers the compositional construction of P- and T-semiflows in PNs. The work that is most closely related is that of Christensen et al. on compositional construction of P- and T-invariants for coloured Petri nets under the framework of *Modular CP-nets* [CP92]. Our work differs substantially, not only because it works at the level of semiflows, which does not require the definition of an initial marking, but because the semiflows defined, and in consequence the invariants defined, reflect and exploit the symmetries in the behaviour of coloured objects in the system.

There is still much work that can be done in this area. We have limited our work to positive integer P- and T-semiflows. A natural extension of the work will be to study the compositional construction of other types of P- and T-flows (integer, rational, etc.). To improve the current work we could study ways of detecting sets of type T_{sem}, in order to be able to calculate the symbolic T-semiflows of the lower level *cWN*s. More work is also needed in the process of identifying the cases where new P- or T-semiflows are created and in determining which are the new P- or T-semiflows created.

We have not included the algorithms for the compositional construction of symbolic and static P-semiflows or for symbolic T-semiflows under each of the compositional operations. These can be obtained by applying the propositions given for each case. As an example in Appendix A we present an algorithm for the compositional construction of symbolic P-semiflows under the choice composition.

Chapter 6

State Space Analysis of Compositional WN systems

6.1 Introduction

State Space Analysis techniques allow the proof of many interesting properties of a system, such as, presence of deadlock, reachability and liveness, amongst others. They are based on the study of the graph that contains all possible evolutions of the PN system, known as the *Reachability Graph* (RG). In general, state space analysis techniques are very expensive due to the large space and time requirements of storing and constructing the RG.

Well-formed nets (WNs) provide a modelling framework in which intrinsic symmetries of the model are naturally detected. These symmetries can then be used to reduce the size of the underlying state space. Instead of constructing a RG based on ordinary markings, in [CDFH90], Chiola et al. propose the creation of the *Symbolic Reachability Graph* (SRG) of a WN. The algorithm for the generation of the SRG has the same structure as the one for generating the RG of GSPNs[AMBC+95], but it uses the concepts of *symbolic marking representations* and *symbolic firing rule*. The equivalence between the SRG and the RG from the point of view of the reachability of the markings is proven in [CDFH91]. This equivalence ensures that no information is lost by analysing the SRG instead of the RG.

Within the framework of compositional construction of *cWN* systems, it would be desirable to build the SRG of a *cWN* using the information about the SRGs of its sub-components. The objective would be to reduce the number of operations required to determine the evolution of the *cWN* by using, whenever possible, information already computed for the sub-components. In this chapter we study the difficulties encountered in trying to use the information of the SRGs of the sub-components to create the SRG of a *cWN*. Consequently, another type of reachability graph is introduced. This graph is the *Composed Reachability Graph* (CRG), based on the concept of a *composed marking*. A composed marking will represent the set of symbolic markings of the *cWN* formed by the combination of symbolic markings of its sub-components. We prove that it is possible to obtain all the information of the SRG of a *cWN* from the CRG, from the point of view of reachability. This ensures that no information is lost by analysing the CRG instead of the SRG.

Section 6.2 will introduce the concepts related to the SRG. Following this, in Section 6.3 we present the difficulties encountered while trying to propose a method for the compositional construction of the SRG. In Section 6.4 the CRG is introduced. We define a composed marking and define the concepts related to the construction of the CRG. We study how the set of enabled (composed) transition instances of a composed marking of a *cWN* can be determined according to the compositional operation employed to build the component. Following this, in Section 6.5 we study the properties of the CRG and its relation with the SRG of the *cWN*. To conclude, in Section 6.6, we summarise the work presented throughout the chapter.

6.2 Background

In this section we will review the concepts related to the SRG.

6.2.1 The Reachability Graph of a PN System

As presented in Chapter 2, the reachability graph (RG) of a PN system describes the evolution of the system from an initial marking M_0. The *Reachability Set* (RS) of the system is defined as the smallest set of markings such that:

- $M_0 \in RS$ and
- $(M_1 \in RS \ \wedge \ \exists t \in T : \ M_1[t\rangle M_2) \Rightarrow M_2 \in RS$

The RG is the labelled directed multi-graph, whose set of nodes is RS and whose set of arcs A is defined as follows:

- $A \subseteq RS \times RS \times T$
- $\langle M_i, M_j, t\rangle \in A \Leftrightarrow M_i[t\rangle M_j$

Properties such as absence of deadlock and liveness can be checked on the RG using classical graph analysis algorithms.

6.2.2 The Symbolic Reachability Graph (SRG) of a WN system

Well-formed nets (WNs) provide a modelling framework in which intrinsic symmetries are naturally detected. It is possible to identify two kinds of symmetries inside subsets of colour classes: rotation and general permutation [CDFH91]. These symmetries can then be used to reduce the size of the underlying state space. In order to exploit the idea of symmetry amongst the objects of the basic colour classes, instead of constructing a RG based on ordinary markings, it is possible to construct a *symbolic reachability graph* (SRG) [CDFH90]. The algorithm for the generation of the SRG has the same structure as the one for generating the RG of GSPNs[AMBC+95], but it uses the concepts of *symbolic marking representations* and *symbolic firing rule*.

A *symbolic marking* represents an equivalence class on the state space of the WN model. This equivalence is given in terms of the possible basic colour permutations that produce the same behaviour. The objects of a static sub-class are said to *behave homogeneously* in the net if they have the same distribution over the places of the net.

Using the same notation as for the definition of symbolic P-semiflows and T-semiflows, let us consider a group ξ of permutations on $\otimes_{i=1}^{n} C_i$ defined as in Chapter 5.

Definition 6.1 (Permutation of an ordinary marking, from [CDFH91]) *Let M be an ordinary marking of a WN $\mathcal{N}$, and $s \in \xi$ a permutation. Then $s \cdot M$, the permutation of M by s, is a marking defined by:*

$$\forall p \in P,\ \forall c \in C(p), s \cdot M(p,c) = M(p, s(c))$$

Definition 6.2 (Symbolic Marking, from [CDFH91]) *A* symbolic marking *$\mathcal{M}$ is an equivalence class of the relation Eq defined by:*

$$M\ Eq\ M' \Leftrightarrow \exists s \in \xi,\ M' = s \cdot M$$

To clarify the concept of symbolic marking, in Figure 6.1 we present an example of a message communication system taken from [CDFH91]. Notice that $\forall i \in \{2 \dots k\}$, $M(Idle, s_i) = 1$ and $M(Idle, s_1) = 0$. This reflects at least two distributions of objects of C_1 in the places of the net. One corresponds to object s_1, that has no instances in *Idle* and the other corresponds to all other objects of C_1 which each have one instance in *Idle*. In place *T.Message buf* it holds that

$$\forall m_i \in D_{2,1}\ \ M(\mathit{T.Message\ buf}, \langle s_1, s_2, m_i \rangle) = 1.$$

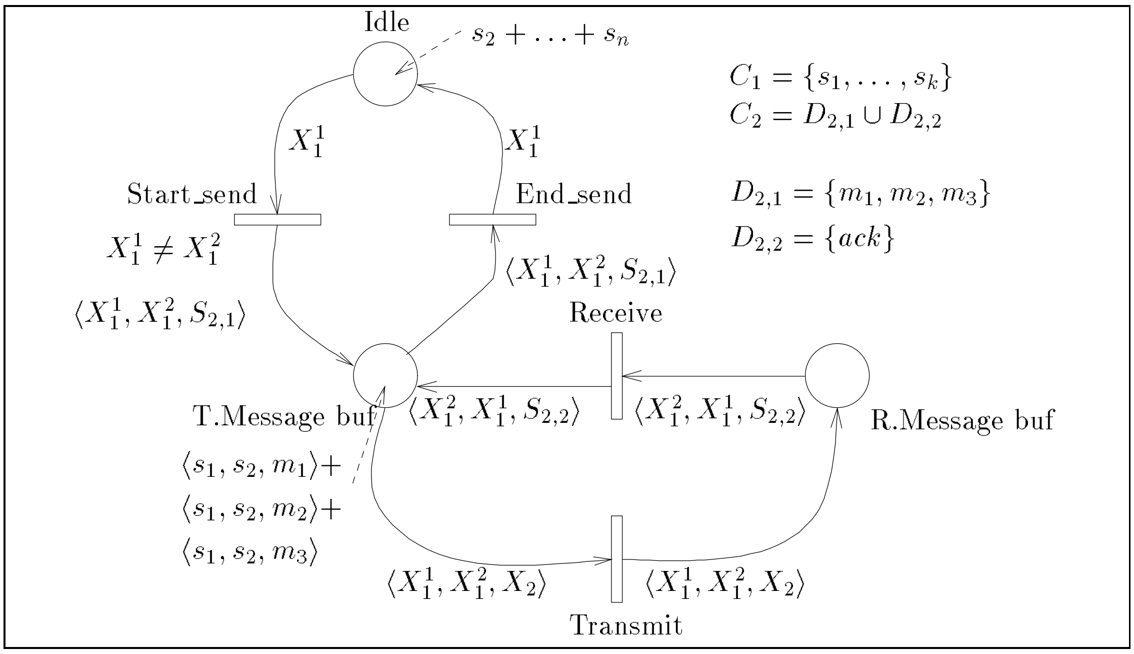


Figure 6.1: Example of a WN with an ordinary marking.

This means that s_2 behaves differently from the rest of the objects of its colour class, since it has an instance in *Idle* and it is present in *T.Message buf* as second member of all tuples. The third member of the tuples in *T.Message buf* correspond to the objects of the static subclass $D_{2,1}$. Each object appears in just one tuple.

So omitting the name of the objects, we could say that the objects of class C_1 are distributed in three different manners throughout the places of the net: the first describes an object that has no instance in *Idle* and is the first member of all tuples in *T.Message buf*; the second corresponds to an object that has one instance in *Idle* and appears as the second member in all colour tuples in T.Message buf; and the third corresponds to all other objects of the colour class C_1. Notice that which ever objects had been used to instantiate X_1^1 and X_1^2, for the firing of *Start_send* from the initial marking M_0 (where $M_0(Idle) = \{s_1, \ldots, s_n\}$ and $\forall p \in P - \{Idle\}$, $M_0(p) = 0$), the distribution of the coloured objects over the places of the WN would have been the same for all resulting markings. All permutations of the marking M have the same distribution of coloured objects over the places of the WN.

In a symbolic marking the permutations $s \in \xi$ are defined within static subclasses. Objects that can be permuted with each other in any firing instance, to produce markings that belong to the same equivalence class, form a *dynamic subclass*. A dynamic sub-class is denoted by Z_i^j, where C_i is the basic colour class to which the colours represented by the dynamic subclass belong; and j is the index that identifies the dynamic subclass, within the set of dynamic subclasses defined over the basic colour class C_i, for a given symbolic marking. Two different basic colours can be represented by the same dynamic subclass only if they have the same coloured objects distribution in all places, and if they belong to the same static subclass. In the case of ordered basic colour classes, only coloured objects with contiguous colours can be represented by the same dynamic subclass, and the ordering relation amongst basic objects is reflected by the ordering of indexes of the dynamic subclasses. A dynamic subclass is characterised by its cardinality ($card$; number of different objects represented by the dynamic subclass) and the static subclasses to which the objects belong (d). Figure 6.2 shows the symbolic marking to which the marking presented in Figure 6.1 belongs.

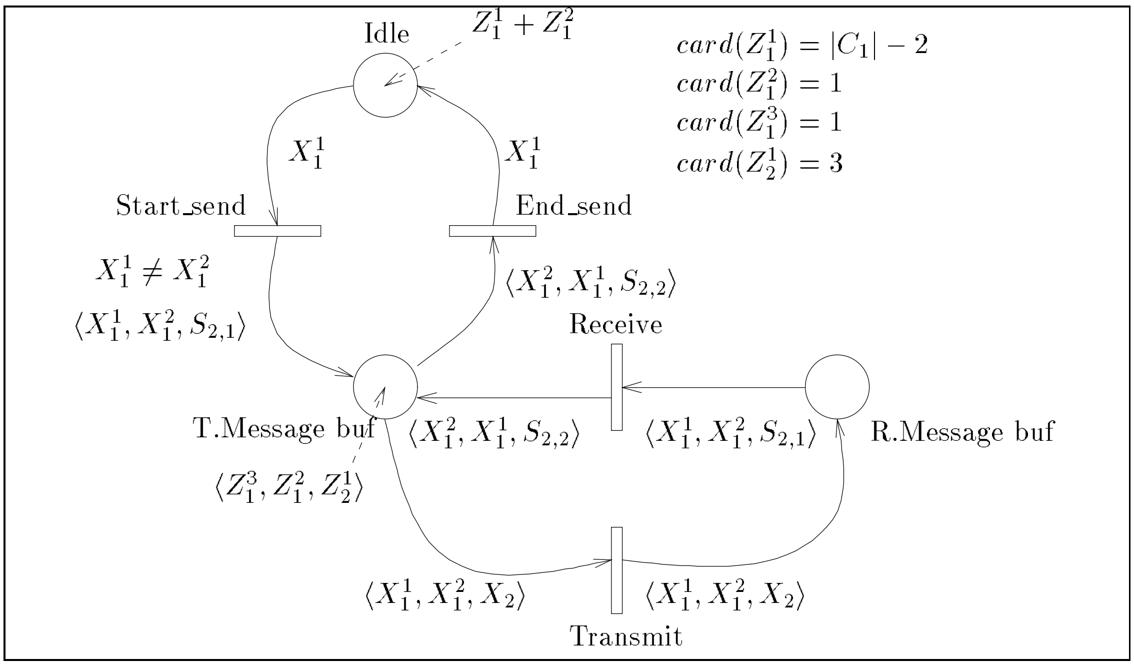


Figure 6.2: Example of a symbolic marking.

A representation $\mathcal{R}$ of a symbolic marking $\mathcal{M}$ is a compact description of $\mathcal{M}$. Formally, it is defined as a four-tuple $\mathcal{R} = \langle m, card, d, mark\rangle$, where

- $m : I \rightarrow \mathbb{N}^+$, such that $m(i)$ is the number of dynamic subclasses of C_i in $\mathcal{M}$.
- Given $\widehat{C}_i = \{Z_i^j \mid 0 < j \leq m(i)\}$ the set of dynamic subclasses of C_i in $\mathcal{M}$, $card : \cup_{i \in I} \widehat{C}_i \rightarrow \mathbb{N}$, such that $card(Z_i^j) > 0$ is the number of basic colours represented by the dynamic subclass and $\forall i$, $\sum_{j=1}^{m(i)} card(Z_i^j) = |C_i|$.

- $d : \cup_{i \in I} \widehat{C}_i \rightarrow \mathbb{N}$, is such that for $Z_i^j \in \widehat{C}_i$, $d(Z_i^j) = q$, with $q \in \{1, \ldots, n_i\}$, where n_i is the number of static subclasses of C_i, and $\forall i \in \{1, \ldots, n\}, \forall j, k \in \{0, \ldots, m(i)\}$, with $j < k$, $d(Z_i^j) \leq d(Z_i^k)$.

- $\forall p \in P, mark(p) : \widehat{C}(p) \rightarrow \mathbb{N}^+$, such that $mark(p)$ is the multiplicity of the valid tuples of dynamic subclasses in place p, given $\widehat{C}(p)$ the set of all possible tuples of dynamic subclasses according to the colour domain of place p.

In the following $\mathcal{M}$ will denote both a symbolic marking and its representation, unless otherwise stated.

As we have seen, a dynamic subclass can represent various colours belonging to a static subclass, but a colour can only be represented by one dynamic subclass. We will denote by $\widehat{D}_{i,q}$ the set of dynamic subclasses of $\widehat{C}_i$ (the set of dynamic classes of colour class C_i) that represent the colours of the static subclass $D_{i,q}$ of C_i, i.e.

$$\widehat{D}_{i,q} = \{Z_i^j \in \widehat{C}_i \mid d(Z_i^j) = q\}$$

All symbolic markings must satisfy the following condition:

$$\forall C_i,\ \forall D_{i,q},\ \sum_{Z_i^j \in \widehat{D}_{i,q}} card(Z_i^j) = |D_{i,q}| \tag{6.1}$$

i.e. the sum of the cardinalities of the dynamic subclasses related to a static subclass, must equal the cardinality of that static subclass.

Given the definition of a symbolic marking representation we observe that a symbolic marking can have many representations (see [CDFH93, CDFH91]). For example in the WN presented in Figure 6.2 it is possible to change the indexes of the dynamic subclasses, or define new dynamic subclasses (subclasses of the existing ones), to create alternative representations of the same symbolic marking. A first step towards an efficient algorithm for the enumeration of the SRG of a WN is the definition of a unique representation for each symbolic marking. To obtain a *canonical form* two properties are defined, namely *minimality* and *ordering* [CDFH90]. The idea behind a minimal symbolic marking is to have the smallest possible number of dynamic subclasses for each basic colour class. The minimal representation of a symbolic marking is unique within a permutation of indexes of dynamic subclasses. By properly ordering a minimal representation we can obtain a canonical representation. This ordering implies the adjustment of the indexes of the dynamic subclasses according to some unequivocal criterion. In [CDFH90, CDFH91] an algorithm is proposed for the computation of the canonical representation, based on the lexicographic ordering of the indexes and markings of the dynamic subclasses.

The minimality and ordering definitions require the introduction of the *marking projection function* $mark_{sp}$. This function is defined over the set of all possible tuples of dynamic subclasses from the Cartesian product of any subset of $\widehat{C}$ (without repetitions of the same basic colour class) [CDFH93]. For computing minimality the function is applied over tuples of arity one. Intuitively, the function works as follows. Given a symbolic marking $\mathcal{M}$ and a tuple of dynamic subclasses, it returns a vector of $|P|$ natural numbers, encoding the distribution of the tuple in the marking. The formal definition of $mark_{sp}$ can be found in [CDFH91].

To exploit the advantages of the symbolic marking representations it is necessary to construct the SRG starting from a symbolic initial marking representation and by generating reachable symbolic marking representations, without building the RG of ordinary markings and then grouping markings into equivalence classes. In order to accomplish this, a symbolic firing rule on the symbolic firing representations is defined [CDFH91].

In a *symbolic firing instance*, instead of coloured objects, dynamic subclasses are assigned to the transition parameters. This means that any object in the dynamic subclass can be assigned to the parameter. When several parameters of a transition t, defined over the same basic colour class C_i, are assigned to the same dynamic subclass Z_i^j, it is necessary to specify whether the parameters are instantiated to the same or different coloured objects within the dynamic subclass. Given the x^{th} parameter of type $C_i \in C(t)$ of a transition t, denoted by $param_i^x$, the parameter instance can be specified by the pair $\langle \lambda_i(x), \mu_i(x) \rangle = \langle j, k \rangle$, meaning that the parameter represents the k^{th} (arbitrarily chosen) element of Z_i^j. Notice that k must be less than or equal to the maximum of the number of parameters instantiated within Z_i^j and the cardinality of Z_i^j. A symbolic firing instance will then be represented by a tuple $[t, \lambda, \mu]$ where: t is the transition fired, $\lambda = \{\lambda_i : \{1, \cdots, e_i\} \rightarrow \mathbb{N}^+\}$ is the set of functions that assign a dynamic subclass to each of the parameters of t, and $\mu = \{\mu_i : \{1, \cdots, e_i\} \rightarrow \mathbb{N}^+\}$ is the set of functions that assign an object of a dynamic subclass to a parameter (e_i is the number of occurrences of the colour class C_i in the colour domain of t). A formal definition of a symbolic firing can be found in [CDFH90, CDFH91] or [CDFH93].

In order to define the *symbolic enabling* and *firing rules*, the dynamic subclasses are *split* into subclasses, one for each of the (arbitrarily) selected objects for the firing, and a dynamic subclass that represents all other (non-selected) objects of the original dynamic subclass. For ordered classes the instantiated dynamic subclasses are always split into dynamic subclasses of cardinality one. With such a definition of split marking, dynamic subclasses can be substituted for coloured objects in the transition firing. An extensive and formal description of the *splitting* process can be found in [CDFH93]. This description differs slightly from the one given in [CDFH90], where the dynamic subclasses are split into two dynamic subclasses.

The canonical representation of the symbolic marking $\mathcal{M}'$ obtained by firing the symbolic instance $[t, \lambda, \mu]$ enabled in $\mathcal{M}$ ($\mathcal{M}' = \mathcal{M}[t, \lambda, \mu\rangle$) is computed in four steps which use intermediate, not necessarily canonical, representations:

1. Splitting of $\mathcal{M}$ with respect to $[t, \lambda, \mu]$, given $[t, \lambda, \mu]$ enabled in $\mathcal{M}$.
2. Actual firing of $[t, \lambda, \mu]$ obtaining the symbolic marking $\mathcal{M}' = \mathcal{M}[t, \lambda, \mu\rangle$, not necessarily minimal.
3. Grouping of dynamic subclasses of $\mathcal{M}'$ to obtain a minimal marking representation $\mathcal{M}''$.
4. Ordering of the minimal representation of $\mathcal{M}''$ to obtain a canonical representation $\mathcal{M}'''$.

Figures 6.3 and 6.4 illustrate these steps for the symbolic firing of the transition instance $[Start\_send, \{\lambda_1 = Z_1^1, \lambda_2 = Z_1^1\}, \{\mu_1 = 1, \mu_2 = 1\}]$.

The algorithm for the construction of the RG of a PN system (coloured or uncoloured) is an iterative process, that takes an unvisited marking of RS, visits it to obtain its set of directly reachable markings and incorporates the elements of this set into the RS provided they are not already members. This process is the same for the construction of the *SRG*, but it is done over canonical symbolic markings. The symbolic reachability set (SRS) contains canonical symbolic markings reachable from the initial symbolic marking $\mathcal{M}_0$.

In order to ensure that no information about reachability is lost by analysing the SRG instead of the RG, the following properties relating the SRG with its corresponding RG are proven in [CDFH91].

Property 6.1 (Equivalence between symbolic and ordinary reachability) *Let $\mathcal{M}$ be a symbolic marking, and $[\mathcal{M}\rangle$ be the set of symbolic markings reachable from $\mathcal{M}$. Then*

$$\bigcup_{M \in \mathcal{M}} [M\rangle = [\mathcal{M}\rangle$$

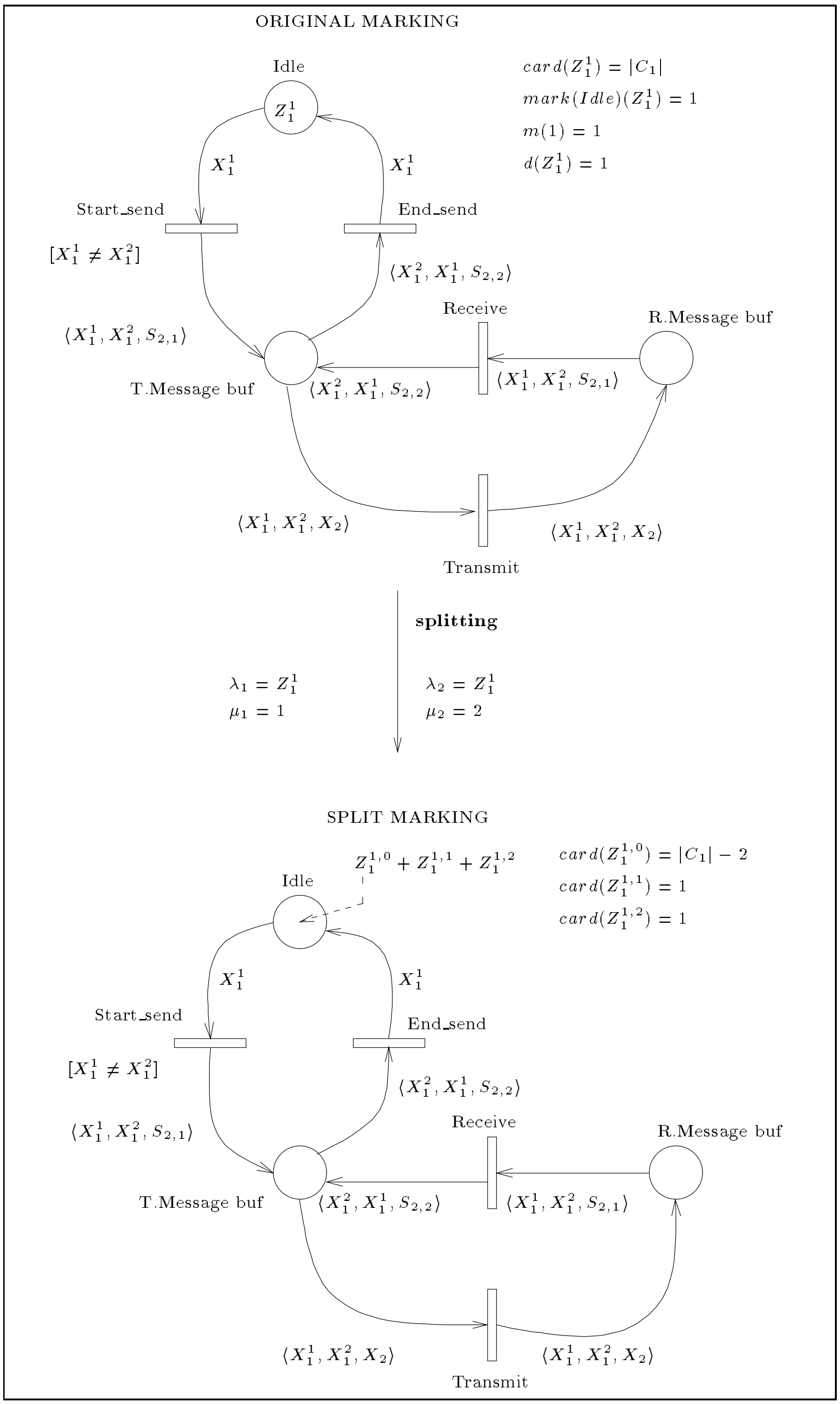


Figure 6.3: Example of the symbolic firing process (PART A).

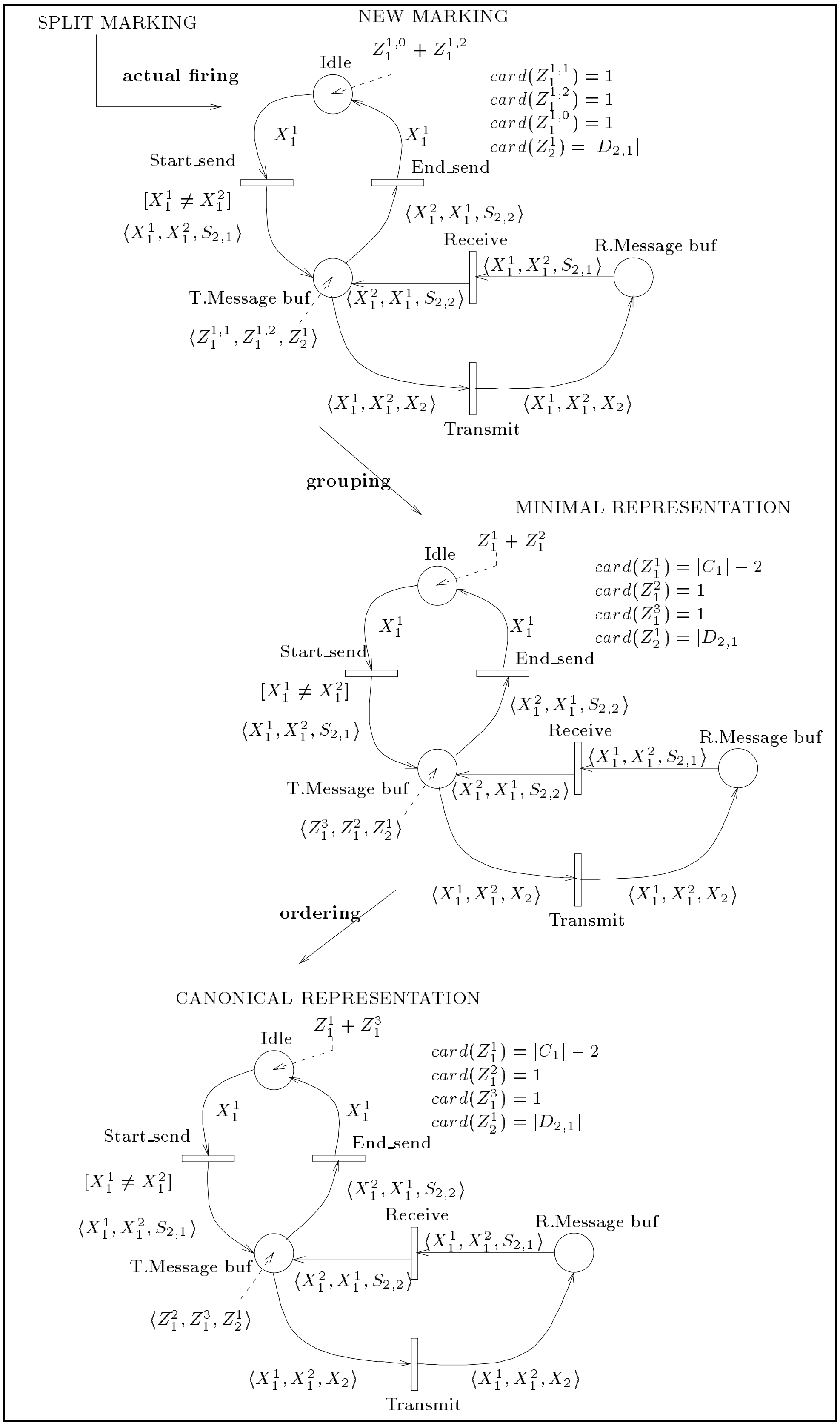


Figure 6.4: Example of the symbolic firing process (PART B).

Property 6.2 *Strong connectivity of RG $\Rightarrow$ strong connectivity of SRG, but not vice versa.*

Property 6.3

$$
\begin{aligned}
& \textit{Strong connectivity of the SRG} \\
& \wedge\ \forall i \in (0,h],\ \exists \mathcal{M}' \in SRG \textit{ such that } (\mathcal{M}'.m(i) = n_i) \\
& \wedge\ i \in (h,n],\ ((n_i > 1)\ \vee\ (\exists \mathcal{M}' \in SRG \textit{ such that } (\mathcal{M}'.m(i) = 1)) \\
& \Rightarrow \textit{Strong connectivity of RG}
\end{aligned}
$$

In the following section we will study how the SRG of a *cWN* can be constructed using as a starting point the SRGs of its sub-components.

6.3 Compositional construction of the SRG

As mentioned when introducing the symbolic firing process, the idea behind the construction of a SRG is to be able to use symbolic markings at all times, without having to return to the ordinary markings that they represent. It is necessary to identify enabled symbolic firing instances and minimise symbolic markings based on the distribution of their dynamic subclasses and not on the distribution of specific coloured objects.

The objective of the compositional construction of the SRG of a *cWN* $\mathcal{N}$ is to be able to re-use the information of the SRG(s) of the *cWN*(s) which composed form $\mathcal{N}$. In this way, we aim to reduce the number of operations required to determine the evolution of the *cWN* system.

6.3.1 Composition of symbolic markings

Let us denote by L and R the two *cWN*s to be composed to form a *cWN* $\mathcal{N}$. The distribution of coloured objects over the places defined by symbolic markings of L and R, are completely independent. We know that the symbolic markings of each component must satisfy the condition 6.1 (on page 135), i.e. the sum of the cardinalities of the dynamic subclasses associated with a static subclass, must equal the cardinality of the static subclass. When composing two symbolic markings $\mathcal{M}_L \in SRG_L$ and $\mathcal{M}_R \in SRG_R$ we must ensure that the resulting symbolic marking of $\mathcal{N}$ also satisfies condition 6.1. Therefore, the composition cannot consist of simply including all dynamic subclasses of $\mathcal{M}_L$ and of $\mathcal{M}_R$ as dynamic subclasses of a symbolic marking of $\mathcal{N}$. Otherwise, each colour of a static subclass will be represented by two dynamic subclasses (by one from $\mathcal{M}_L$ and by one from $\mathcal{M}_R$).

Let us study how to construct a symbolic marking of $\mathcal{N}$, given a symbolic marking of L and one of R. We will start by analysing the simplest case of composition, where the operation involves no place or transition fusion, i.e. independent parallel composition.

6.3.1.1 Obtaining a symbolic marking of a cWN

In Figure 6.5 we present a small example to illustrate the problem of obtaining a symbolic marking $\mathcal{M}_\mathcal{N}$ of $\mathcal{N}$ from the composition of the symbolic markings $\mathcal{M}_L$ of L and $\mathcal{M}_R$ of R, when the operation is an independent parallel composition. To keep the example simple we will only consider one basic colour class (non-ordered) with no static subclasses. The symbolic marking $\mathcal{M}_L$ has three dynamic subclasses, namely: $R_L.Z_1^1$, $R_L.Z_1^2$ and $R_L.Z_1^3$; and $\mathcal{M}_R$ has two dynamic subclasses, namely: $R_R.Z_1^1$ and $R_R.Z_1^2$. The symbolic marking representations of $\mathcal{M}_L$ and $\mathcal{M}_R$ satisfy condition 6.1; thus in both it holds that

$$\sum_{j=1}^{m(1)} Z_1^j = |C_1|$$

where $m(1)$ will be the number of dynamic subclasses of colour C_1. This condition will mean that the three dynamic subclasses of $\mathcal{M}_L$ represent all the colours of C_1, and so will the two dynamic subclasses of $\mathcal{M}_R$.

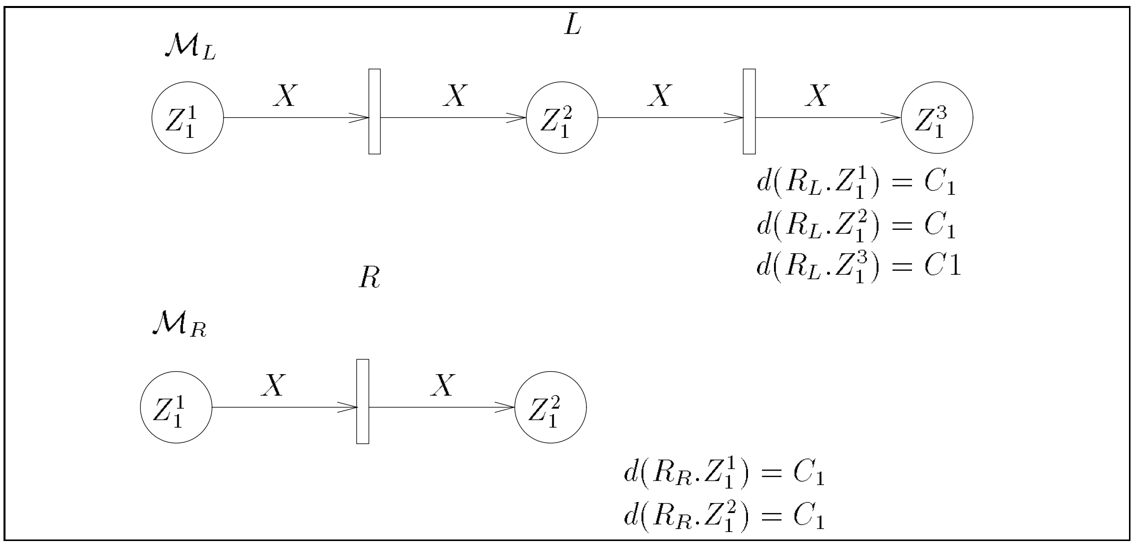


Figure 6.5: Example to illustrate composition of symbolic markings.

As mentioned before, if we were to simply combine the existing dynamic subclasses, we will end up with colours of C_1 being represented by more than one dynamic subclass in the combined symbolic marking $\mathcal{M}_{\mathcal{N}}$.

The set of basic colours represented by $R_L.Z_1^1$ in $\mathcal{M}_L$ might be just a part of the basic colours represented by $R_R.Z_1^2$ in $\mathcal{M}_R$, which will mean that the other colours represented by $R_R.Z_1^2$ are represented by other dynamic subclasses of $\mathcal{M}_L$.

With this example we can see that in order to obtain the set of all possible symbolic markings of $\mathcal{N}$ given $\mathcal{M}_L$ and $\mathcal{M}_R$, it is necessary to consider all the possible intersections between the sets of basic colours represented by the dynamic subclasses of $\mathcal{M}_L$ and those represented by the dynamic subclasses of $\mathcal{M}_R$. This must be done in such a way that the dynamic subclasses of the resulting symbolic marking satisfy condition 6.1 and that their markings in the places of $\mathcal{N}$ represent the distribution of coloured objects in both $\mathcal{M}_L$ and $\mathcal{M}_R$.

Table 6.1 represents the problem that we must solve in order to find all the possible intersections between the set of basic colours represented by the dynamic subclasses of $\mathcal{M}_L$ and $\mathcal{M}_R$ for the example in Figure 6.5. The set of dynamic subclasses—of a certain static subclass—determined by a symbolic marking, is a partition of the static subclass (or of the basic colour class in the case of no static subclasses). Thus, the problem we have is to find all possible intersections between two partitions of a static subclass or, in this case, of a basic colour class. The elements of a row must add up to the cardinality of the corresponding dynamic subclass of $\mathcal{M}_L$, and the elements of a column must add up to the cardinality of the corresponding dynamic subclass of $\mathcal{M}_R$.

$\|C_i\|$	$\|R_R.Z_1^1\|$	$\|R_R.Z_1^2\|$
$R_L.\|Z_1^1\|$	?	?
$R_L.\|Z_1^2\|$	?	?
$R_L.\|Z_1^3\|$	?	?

Table 6.1: Intersection between dynamic subclass of $\mathcal{M}_L$ and $\mathcal{M}_R$.

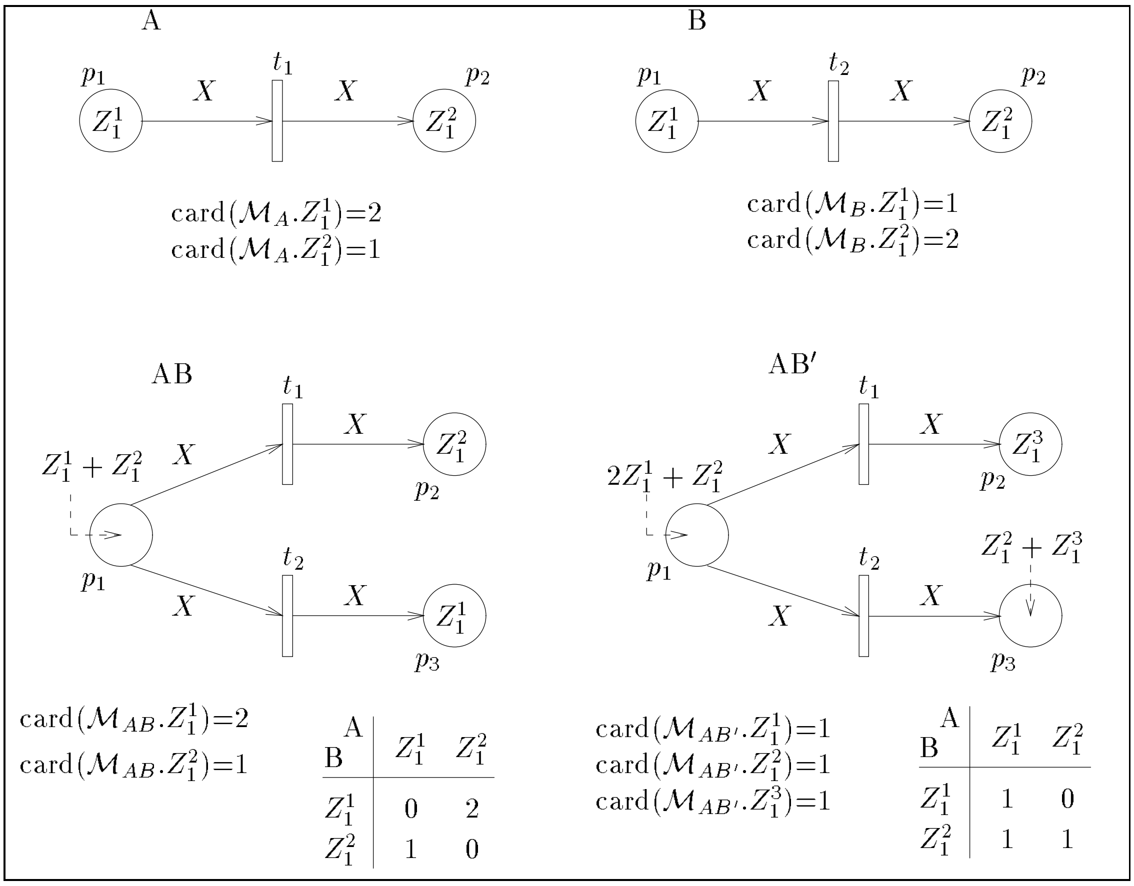
This problem has one or multiple solutions, which means that the composition of two symbolic markings, one of L and one of R, can generate multiple symbolic markings of $\mathcal{N}$. The resulting symbolic markings of $\mathcal{N}$ will have at least as many dynamic subclasses per static subclass as the maximum of the number of dynamic subclasses, for the static subclass, in $\mathcal{M}_L$ and the number of dynamic subclasses, for the static subclass, in $\mathcal{M}_R$. This is given by the fact that object with colours belonging to the static subclass already have different distributions over the places of L and R.

Let us denote by COMB a matrix solution to the problem described. An element COMB(i, j) will indicate the number of colours common to the dynamic subclasses $R_L.Z_1^i$ and $R_R.Z_1^j$. From each solution of this problem we can obtain a symbolic marking which will have as many dynamic subclasses as non-zero elements in the solution matrix. The non-zero elements will represent the cardinality of the dynamic subclass. Given a non-zero element $\mathrm{COMB}(i,j)$ the distribution of the corresponding dynamic subclass is given by the distribution of the dynamic subclasses $R_L.Z_1^i$ and $R_R.Z_1^j$, in the places of L and R, respectively. From now on we will denote by $\widehat{\mathcal{M}_\mathcal{N}}$ the set of symbolic markings obtained from the composition of symbolic markings $\mathcal{M}_L$ and $\mathcal{M}_R$.

In the example above there was only one basic colour class, with no static subclass for simplicity. In a more general scenario the problem of finding all the possible intersections must be done for each static subclass of each basic colour class.

In the case of ordered colour classes the are as many solutions as the number of rotations of the colour class. This is given by the fact that for ordered basic colour classes only contiguous objects can be represented by the same dynamic subclass, and the ordering relation among basic objects is reflected by the ordering of indexes of the dynamic subclasses. However, any colour of the class can be considered as the first.

From the analysis above we can observe that the composition of two *cWN*s, namely L and R, with initial markings $\mathcal{M}_{L,0}$ and $\mathcal{M}_{R,0}$, respectively can generate a *cWN*, $\mathcal{N}$, with one or several SRGs.



B \ A	Z_1^1	Z_1^2
Z_1^1	0	2
Z_1^2	1	0

B \ A	Z_1^1	Z_1^2
Z_1^1	1	0
Z_1^2	1	1

Figure 6.6: Example of a *cWN* system with several SRGs.

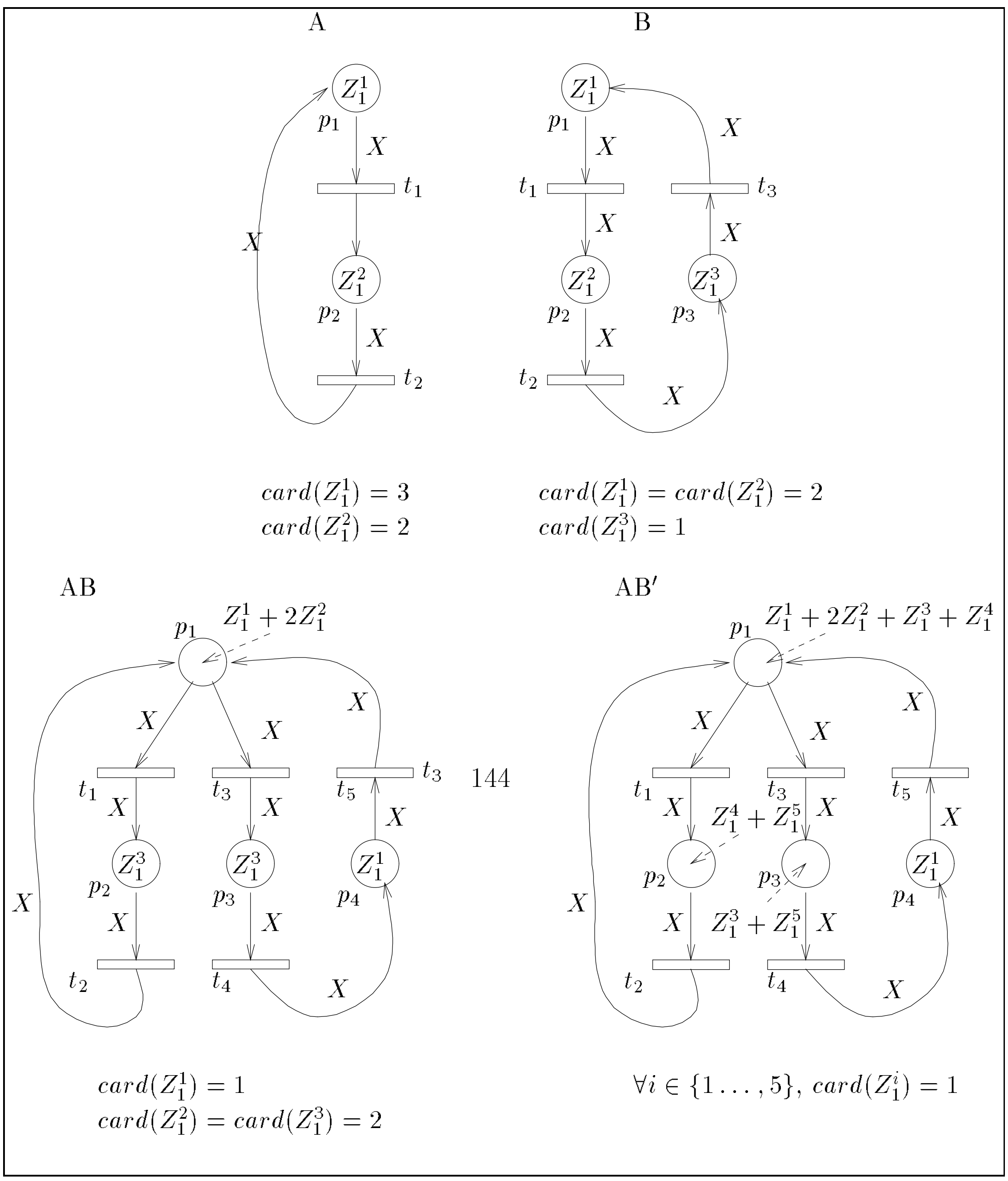


Figure 6.7: Example of a *cWN* system with one SRG.

In Figure 6.6 we present an example where the composition of the components generates more than one (two in this case) SRGs of $\mathcal{N}$. We will assume that all symbolic markings are defined over the same basic colour class, with no static subclasses. For reasons of simplicity the cardinality of the basic colour class, C_1, will be defined as being 3. Notice that although from both $\mathcal{M}_{AB,0}$ and $\mathcal{M}_{AB',0}$ it is possible to arrive at common markings, for example $[0, Z_1^1, Z_1^1]$ (where the first element corresponds to the place p_1, the second to p_2 and the third to p_3) with $card(Z_1^1) = 3$, we cannot arrive from one of them to the other. Therefore, $\mathcal{M}_{AB,0}$ is not in the SRG with initial marking $\mathcal{M}_{AB',0}$, and vice versa, meaning that we have two SRGs of $\mathcal{N}$. In Figure 6.7 we present an example where, although the composition of the initial symbolic markings of L and R can generate several symbolic markings of $\mathcal{N}$, all these markings belong to the same SRG of $\mathcal{N}$. In this example the cardinality of the basic colour class C_1 is considered to be 5. The markings $\mathcal{M}_{AB,0}$ and $\mathcal{M}_{AB',0}$, here presented, are two of the possible symbolic markings of $\mathcal{N}$ that can be generated from the composition of the initial symbolic markings of L and R. From both symbolic markings we can reach a symbolic marking $[Z_1^1, 0, 0, 0]$ with the cardinality of Z_1^1 corresponding to the cardinality of C_1. From this marking we can reach both $\mathcal{M}_{AB,0}$ and $\mathcal{M}_{AB',0}$. Therefore, they would be both in the same SRG. This happens with all symbolic markings generated from the combination of $\mathcal{M}_{L,0}$ and $\mathcal{M}_{R,0}$, therefore there is only one SRG of $\mathcal{N}$.

6.3.2 Reachable symbolic markings from the symbolic markings of a composed marking

The set of *directly reachable symbolic markings* (DRS) from $\mathcal{M}_L$ and the DRS from $\mathcal{M}_R$ are obtained by the symbolic firing of enabled symbolic instances in $\mathcal{M}_L$ and $\mathcal{M}_R$, respectively.

Let us go back to the example of the independent parallel composition of two *cWN*s, presented in Figure 6.5. Consider a symbolic firing instance $[t_r, \lambda_1 = Z_1^1, \mu_1 = 1]$ enabled in $\mathcal{M}_R$, such that upon firing it generates the symbolic marking $\mathcal{M}'_R$ (as shown in Figure 6.8). From the point of view of a symbolic marking $\mathcal{M} \in \widehat{\mathcal{M}}_{\mathcal{N}}$, the colour of the object selected by the instantiation $\langle \lambda_1(X), \mu_1(X) \rangle = \langle Z_1^1, 1 \rangle$ can be represented by any of the dynamic subclasses in $\mathcal{M}.\widehat{C_1}$ that represent colours originally represented by $\mathcal{M}_R.Z_1^1$. Colours of $\mathcal{M}_R.Z_1^1$ can be represented by, at most, 3 dynamic subclasses in any symbolic marking of $\widehat{\mathcal{M}}_{\mathcal{N}}$ (this is because $\mathcal{M}_L$ has 3 dynamic subclasses). Let us take a symbolic marking $\mathcal{M}_{\mathcal{N}} \in \widehat{\mathcal{M}}_{\mathcal{N}}$ such that there are three dynamic subclasses

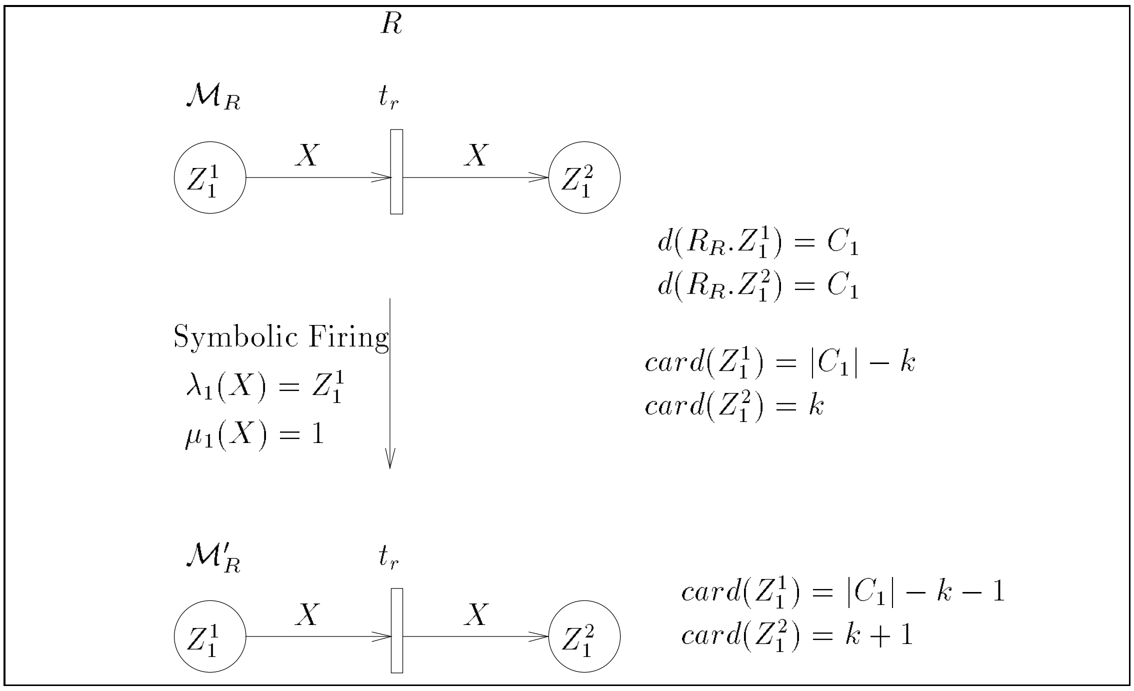


Figure 6.8: Example of symbolic firing.

$\mathcal{M}_{\mathcal{N}}.Z_1^i$, $\mathcal{M}_{\mathcal{N}}.Z_1^j$ and $\mathcal{M}_{\mathcal{N}}.Z_1^k$ that contain colours that were originally represented by $\mathcal{M}_R.Z_1^1$. The symbolic instance $\langle \lambda_1, \mu_1 \rangle$ will then be transformed into the symbolic instances: $\langle \lambda_{\mathcal{N},1}, \mu_{\mathcal{N},1} \rangle$, $\langle \lambda'_{\mathcal{N},1}, \mu'_{\mathcal{N},1} \rangle$ and $\langle \lambda''_{\mathcal{N},1}, \mu''_{\mathcal{N},1} \rangle$ where:

$$\lambda_{\mathcal{N},1}(X) = R_{\mathcal{N}}.Z_1^i \quad \wedge \quad \mu_{\mathcal{N},1}(X) = 1$$
$$\lambda'_{\mathcal{N},1}(X) = R_{\mathcal{N}}.Z_1^j \quad \wedge \quad \mu'_{\mathcal{N},1}(X) = 1 \text{ and}$$
$$\lambda''_{\mathcal{N},1}(X) = R_{\mathcal{N}}.Z_1^k \quad \wedge \quad \mu''_{\mathcal{N},1}(X) = 1$$

The case that we have shown is very simple, because the transition has only one parameter, but what happens when the transition has more than one parameter? In this case, it is necessary to consider the cardinality of the dynamic subclasses of the symbolic marking $\mathcal{M}_{\mathcal{N}}$ of $\widehat{\mathcal{M}_{\mathcal{N}}}$. It must hold that $\forall C_i \in C(t_r)$, $\forall X$ parameter of t_r, $\mu_{\mathcal{N},i}(X) \leq |\lambda_{\mathcal{N},i}(X)|$. Given that for $[t_r, \lambda, \mu]$ in $\mathcal{M}_R$ it holds that $\mu_i(X) \leq |\lambda_i(X)|$, then we must consider all possible assignments of $\mu_{\mathcal{N},i}(X)$, such that for X, Y, parameters of t_r of type C_i, the following conditions hold:

1. If both parameters were instantiated to the same dynamic subclass in $\mathcal{M}_L$, but not to the same colour in the dynamic subclass and if they are both assigned to the same dynamic subclass in $\widehat{\mathcal{M}_{\mathcal{N}}}$, then they must be assigned to different colours within the dynamic subclass of $\mathcal{M}_{\mathcal{N}}$.

$$\lambda_i(X) = \lambda_i(Y) \text{ and } \mu_i(X) \neq \mu_i(Y) \text{ and } \lambda_{\mathcal{N},i}(X) = \lambda_{\mathcal{N},i}(Y) \Longrightarrow \mu_{\mathcal{N},i}(X) \neq \mu_{\mathcal{N},i}(Y)$$

2. If both parameters were instantiated to the same dynamic subclass in $\mathcal{M}_L$ and to the same colour within dynamic subclass, then they must both be assigned to the same dynamic subclass in $\mathcal{M}_\mathcal{N}$ and to the same colour within the dynamic subclass.

$$\lambda_i(X) = \lambda_i(Y) \text{ and } \mu_i(X) = \mu_i(Y) \Longrightarrow$$
$$\lambda_{\mathcal{N},i}(X) = \lambda_{\mathcal{N},i}(Y) \text{ and } \mu_{\mathcal{N},i}(X) = \mu_{\mathcal{N},i}(Y)$$

Consider an enabled symbolic transition instance $[t_r, \lambda, \mu]$ of a transition $t_r \in T_R$ enabled in $\mathcal{M}_R$, such that $\mathcal{M}_R[t_r, \lambda, \mu\rangle \mathcal{M}'_R$. We can prove that the set of symbolic markings, $\widehat{\mathcal{M}'_\mathcal{N}}$ of $\mathcal{N}$— generated by the composition of $\mathcal{M}_L$ and $\mathcal{M}'_R$—contains the set of symbolic markings obtained from firing each of the possible interpretations of the symbolic instance $[t_r, \lambda, \mu]$, in each of the corresponding symbolic markings in $\widehat{\mathcal{M}_\mathcal{N}}$ (the set of markings resulting from the composition of $\mathcal{M}_L$ and $\mathcal{M}_R$). However, we would still need to be able to associate a symbolic marking of $\mathcal{M}_\mathcal{N} \in \widehat{\mathcal{M}_\mathcal{N}}$ with the symbolic marking $\mathcal{M}'_\mathcal{N} \in \widehat{\mathcal{M}'_\mathcal{N}}$ resulting from firing the interpretation of the symbolic transition $[t_r, \lambda, \mu]$ enabled in $\mathcal{M}_\mathcal{N}$. Unfortunately, this is not as straightforward as obtaining the set of symbolic markings. The only way of determining the symbolic marking generated by the firing of a symbolic transition instance is to actually perform the symbolic firing. Nevertheless, knowing that $\mathcal{M}_R[t_r, \lambda, \mu\rangle \mathcal{M}'_R$ and that $\mathcal{M}_L$ combined with $\mathcal{M}'_R$ forms $\widehat{\mathcal{M}'_\mathcal{N}}$, allows us to determine the set of enabled symbolic transition instances in each symbolic marking of $\widehat{\mathcal{M}'_\mathcal{N}}$, based on the symbolic transition instances enabled in $\mathcal{M}'_R$ and $\mathcal{M}_L$.

6.4 The Composed Reachability Graph (CRG)

In the previous section we showed the difficulties of constructing the SRG of a cWN using the information of the SRGs of its sub-components. We saw that we can obtain a group of symbolic markings of the $SRG(s)$ of $\mathcal{N}$ by the composition of symbolic markings in SRG_L with symbolic markings in SRG_R. However, there is no way of deducing, from the arcs of SRG_L and SRG_R, which symbolic transition instance must be fired in order to go from one symbolic marking in a SRG of $\mathcal{N}$ to another. The only way of knowing this is by actually performing the symbolic firing of each enabled symbolic transition instance.

An alternative solution would be to leave the group of symbolic markings expressed in terms of the symbolic markings of the sub-components. The evolution of the system can then be obtained and expressed in terms of the evolution of the symbolic markings of its sub-components. A sub-component may itself be a composition of two *cWN*s; therefore the symbolic markings of the higher level component will be expressed in terms of composition of compositions of symbolic markings. This introduces the concept of *composed markings*.

6.4.1 Composed markings

Definition 6.3 (Composed Marking) *Consider a cWN $\mathcal{N}$, obtained from the composition of two cWNs L and R, by applying a composition operation $\star$. A* composed marking $\widehat{\mathcal{M}_\mathcal{N}}$ *of $\mathcal{N}$ is recursively defined by:*

- *a composition of two symbolic markings ($\mathcal{M}_L \star \mathcal{M}_R$), where $\mathcal{M}_L$ is a symbolic marking of L and $\mathcal{M}_R$ is a symbolic marking of R; or*
- *a composition of a symbolic marking with a composed marking ($\mathcal{M}_L \star \widehat{\mathcal{M}_R}$ or $\widehat{\mathcal{M}_L} \star \mathcal{M}_R$), where $\widehat{\mathcal{M}_L}$ is a composed marking of L and $\widehat{\mathcal{M}_R}$ is a composed marking of R; or*
- *a composition of two composed markings ($\widehat{\mathcal{M}_L} \star \widehat{\mathcal{M}_R}$).*

Formally:

$$\widehat{\mathcal{M}_\mathcal{N}} ::= \mathcal{M}_L \star \mathcal{M}_R \ \mid \ \mathcal{M}_L \star \widehat{\mathcal{M}_R} \ \mid \ \widehat{\mathcal{M}_L} \star \mathcal{M}_R \ \mid \ \widehat{\mathcal{M}_L} \star \widehat{\mathcal{M}_R}$$

Notice that we are using the same notation $\widehat{\mathcal{M}}$ to refer to a composed marking, as we did for the set of markings representing a composition of two symbolic markings. This has been done because they represent the same set of symbolic markings of $\mathcal{N}$.

We want to be able to refer to the components of a composed marking, without having to specify each time if it is a symbolic marking or a composed marking itself. For this reason we introduce the concept of a *marking component* of a composed marking.

Definition 6.4 (Marking Component) *A* marking component *($\mathcal{E}$) of a composed marking ($\widehat{\mathcal{M}_\mathcal{N}}$) is either a symbolic marking or a composed marking (as stated in Definition 6.3). The composed marking component corresponding to the L component will be denoted by $\mathcal{E}_L$; similarly, the R marking component will be denoted $\mathcal{E}_R$. A composed marking is formed by the composition of two marking components.*

A composed marking can be seen as a tree where the leaves are symbolic markings—what we will call *lowest level marking components*. These non-divisible marking components, correspond to symbolic markings of non-divisible *cWN*s. These *cWN*s are not necessarily *bWN*s, but components for which we have already calculated the SRGs. From now on we will talk of composed markings when referring also to symbolic markings of the lowest level components, unless otherwise specified.

In the same manner as the transition from the RG to the SRG, our goal is to build and express the CRG in terms of composed markings, without having to generate all the symbolic markings represented by a composed marking and apply the symbolic firing rules over each symbolic marking. It is therefore necessary to determine how a composed marking can evolve into another composed marking. This requires the definition of a *composed transition instance* and a *composed firing rule*. Apart from this we also need to be able to determine the set of composed transition instances enabled in a composed marking. The idea is to deduce the evolution of one composed marking into the another by analysing the evolution of its sub-components.

For this we must consider that the marking of a sub-component may be altered when it is composed with another sub-component. Let us consider a composition of two *cWN*s, namely L and R, by a binary composition operation that involves the fusion of places to form a *cWN* $\mathcal{N}$. The initial composed marking $\widehat{\mathcal{M}_{\mathcal{N},0}}$, obtained from the composition of the initial composed markings $\widehat{\mathcal{M}_{L,0}}$ of L and $\widehat{\mathcal{M}_{R,0}}$ of R. $\widehat{\mathcal{M}_{\mathcal{N},0}}$ can evolve into another composed marking if there is an enabled composed transition instance in $\widehat{\mathcal{M}_{\mathcal{N},0}}$. Intuitively a composed transition instance is similar to a symbolic transition instance but the parameters of the transition can be instantiated to dynamic subclasses originally belonging to composed markings of different sub-components of $\mathcal{N}$. Under place fusion it is possible that in a place of $\mathcal{N}$ there are dynamic subclasses belonging to different marking components. Recall that the marking of the place resulting from the fusion of places is the sum of the markings of the places being fused.

Let us take the example of Figure 6.9 and suppose the composition of L and R by a competing parallelism operation where $\Lambda(L.p_1) = R.p_1$. The transition $R.t_2$ was not enabled in $\mathcal{M}_{R,0}$. However, if we consider that the dynamic subclass $\mathcal{M}_{L,0}.Z_1^1$ has at least one colour in common with the dynamic subclass $\mathcal{M}_{R,0}.Z_1^1$ then the instances where the parameter X of t_2 is assigned to the common element will be enabled in $\widehat{\mathcal{M}_{\mathcal{N},0}}$.

Dynamic subclasses of a composed marking.- Dynamic subclasses are defined at the level of symbolic markings. As we have seen in the example presen-

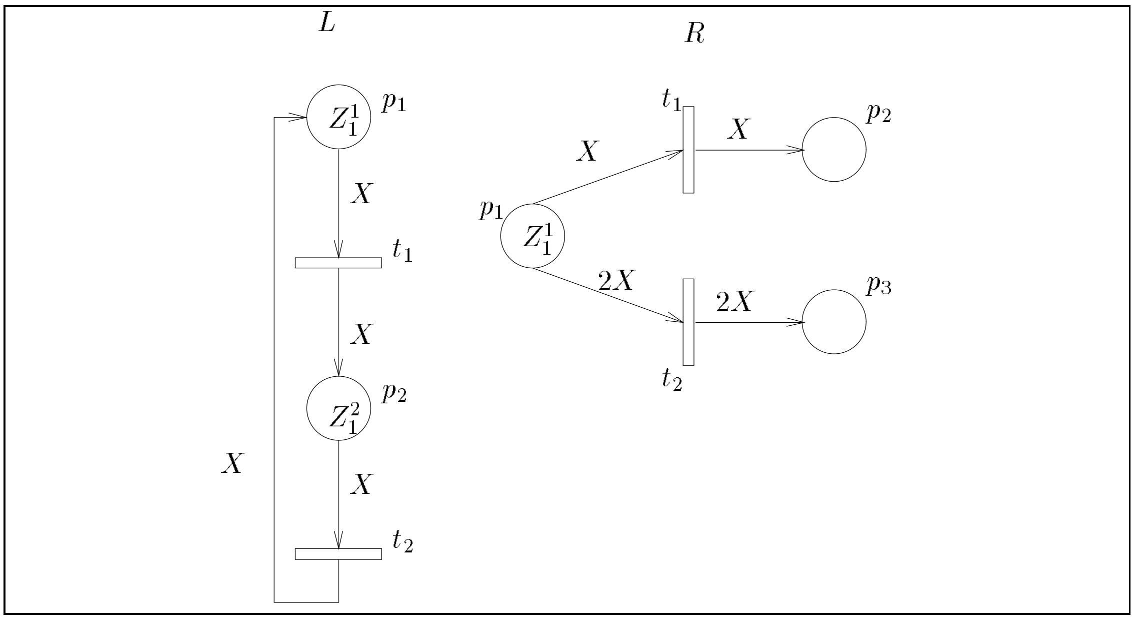


Figure 6.9: Study of the competing parallelism composition.

ted in Figure 6.9, at the level of composed markings we must be able to refer to dynamic subclasses defined at the level of symbolic markings and to dynamic subclasses of the composed markings, defined as a consequence of the intersection of dynamic subclasses of symbolic markings. To identify to which marking component a dynamic subclass belongs, we augment the definition of dynamic subclasses with this information. In the case of a dynamic subclass defined as the intersection of two others, we keep track of the dynamic subclasses that form it. The distribution of the dynamic subclasses is defined as the sum of the distributions of the original dynamic subclasses that form it. Formally a dynamic subclass of a composed marking, termed a *composed dynamic subclass* is defined as follows:

Definition 6.5 (Composed dynamic subclass) *A* composed dynamic subclass *of a composed marking, denoted $\widehat{Z_i^k}$, is a 4 tuple:*

$$\langle dyn\_ss, card, q, y \rangle$$

where,

- $dyn\_ss = \{(com_k, i, j)\}$ *is the ordered set of dynamic subclasses of lowest level components that are intersected to form the composed dynamic subclass. In the triple (com_k, i, j), j represents the index of the dynamic subclass within the set $\widehat{C}_i$, of dynamic subclasses of colour C_i, in the lowest level marking component com_k in which the dynamic subclass was originally formed. It must hold that $\forall (com_k, i, j), (com_g, i, h) \in dyn\_ss,\ k \neq g,$*

i.e. dynamic subclasses of the same symbolic marking cannot intersect. The order of tuples within the set is defined as follows:

$$\forall (com_k, i, j), (com_g, i, h) \in dyn\_ss, \qquad \textit{if } k < g \textit{ then } (com_k, i, j) < (com_g, i, h)$$

i.e. (com_k, i, j) *has a lower position than* (com_g, i, h).

- *card is the number of basic colours represented by the dynamic subclass.*
- *q is the static subclass of* C_i *in which the dynamic subclass is defined.*
- *y is the index of the dynamic subclass in the current composed marking.*

The intersection of dynamic subclasses occurs as a consequence of the process of searching for possible enabled transitions instances in the marking components of a composed marking. An intersection of dynamic subclasses defines a subset of all possible intersections. In the example presented in Figure 6.9 the composed dynamic subclass of cardinality one obtained from the intersection of the dynamic subclasses $L.Z_1^1$ and $R.Z_1^1$, refers to all possible solutions where these two dynamic subclasses have at least one basic colour in common. In the same way it leads to the definition of the complementary set, the set where $L.Z_1^1$ and $R.Z_1^1$ have no colours in common. So when applying the search process a composed marking may be transformed into several composed markings. In general, from a composed marking with an intersection condition between the dynamic subclasses, for example $L.Z_1^1$ and $R.Z_1^1$ must intersect in at least one colour, we can only go to composed markings with the same condition or with a tighter condition. From having one element in common, we could go to having 2; however, we cannot go to having no elements in common. A composed marking where the dynamic sub-classes $L.Z_1^1$ and $R.Z_1^1$ have no 2 elements in common will be a *restriction* of the composed marking where they have only one element in common.

Definition 6.6 (Restriction of a composed marking)
A composed marking $\widehat{\mathcal{M}}$ *is said to be a* restriction *of another* $\widehat{\mathcal{M}}'$ *if they are formed by the same marking components and the intersection defined over the dynamic subclasses of the lowest level marking components by* $\widehat{\mathcal{M}}$ *are tighter than those defined by* $\widehat{\mathcal{M}}'$.

As for a symbolic marking, a composed marking can also be identified by its representation.

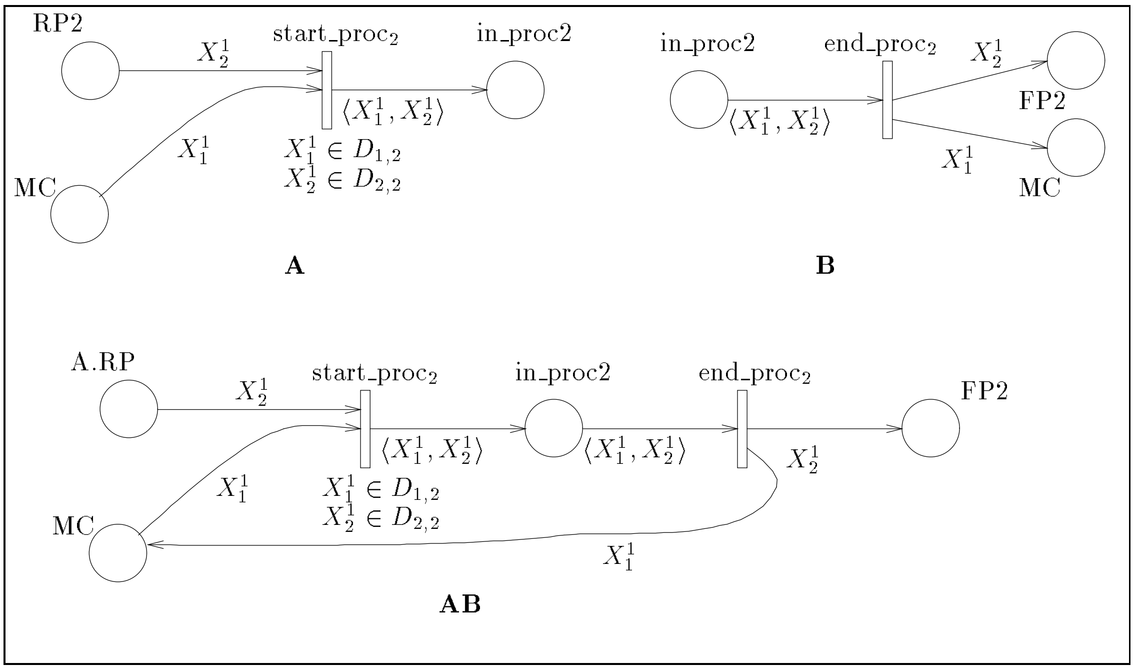


Figure 6.10: Processing of rough parts of type 2.

Definition 6.7 (Representation of a Composed Marking) *A representation $\widehat{\mathcal{R}_\mathcal{N}}$ of a composed marking $\widehat{\mathcal{M}_\mathcal{N}}$ is an 8-tuple,*

$$\widehat{\mathcal{R}_\mathcal{N}} = \langle \mathcal{E}_L, \mathcal{E}_R, FP_L, FP_R, FT_L, FT_R, \nu, \varrho \rangle$$

where:

- *$\mathcal{E}_L$ and $\mathcal{E}_R$ are the marking components which, composed, form $\widehat{\mathcal{M}_\mathcal{N}}$.*
- *FP_L and FP_R are the subsets of places, of L and R respectively, which are the domain and image of the place fusion function ν applied in the composition operation;*
- *FT_L and FT_R are the subset of transitions, of L and R respectively, which are the domain and image of the transition fusion function ϱ applied in the composition operation.*
- *ν is the function that relates the elements of FP_L to elements of FP_R, i.e. defines the fusion of places of L and R.*
- *ϱ is the function that relates the elements of FT_L to elements of FT_R, i.e. defines the fusion of transitions of L and R.*

To illustrate the representation of a composed marking, let us introduce the following example, which is part of the FMS model presented in Chapter 7.

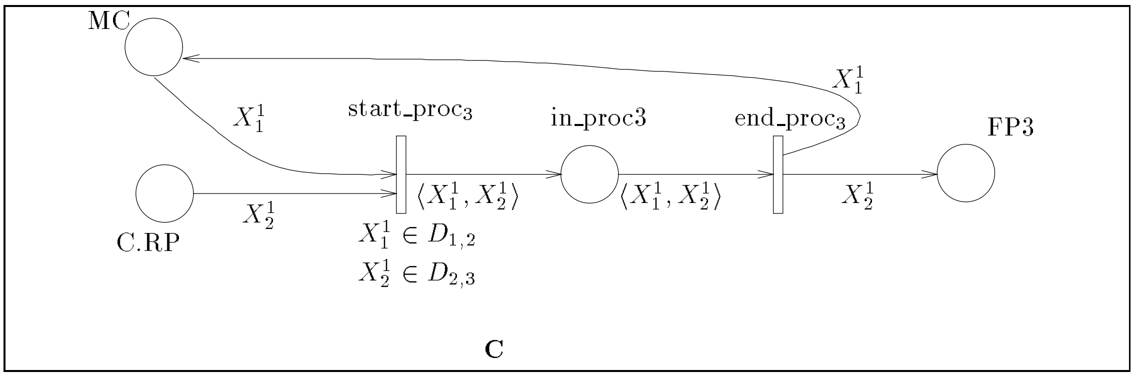


Figure 6.11: Processing of rough parts of type 3.

However, the subsystems considered here do not correspond directly to the subsystems defined in Chapter 7. For this example it is only necessary to know that there are three types of rough parts and three types of machines. The colour classes of the model will correspond to machines (colour class C_1) and rough parts (colour class C_2). The types of rough parts and of machines will be represented by static subclasses. Rough parts of types 2 and 3 are processed by machines of type 2. The component A, in Figure 6.10, represents the first stage of the processing of a part of type 2. Component B, in the same figure, represents the final stage of the processing of a part of type 2. These components are composed (sequentially followed by a closing operation) to form component AB (see Figure 6.10). Component AB will represent the processing of rough parts of type 2, to obtain finished parts of the same type. Component C (see Figure 6.11) represents the processing of rough parts of type 3 to produce finished parts of the same type. These two processes share the machines of type 2. Consider the $cWN\,\mathcal{N}$ formed by the competing parallelism composition of AB with C over the place MC, representing the free machines, as shown on Figure 6.12. Given $\widehat{\mathcal{M}_{AB}}$ and $\widehat{\mathcal{M}_C}$ the representations of composed markings of components AB and C, respectively, the representation $\widehat{\mathcal{R}_\mathcal{N}}$ of a composed marking $\widehat{\mathcal{M}_\mathcal{N}}$ obtained from the composition of $\widehat{\mathcal{M}_{AB}}$ and $\widehat{\mathcal{M}_C}$ is given by[1]:

$$\langle \widehat{\mathcal{M}_{AB}}, \widehat{\mathcal{M}_C}, \{AB.MC\}, \{C.MC\}, \emptyset, \emptyset, \{(AB.MC, C.MC)\}, \emptyset\rangle \tag{6.2}$$

Representation of the initial composed marking.- To obtain the initial composed marking of the CRG of a cWN system we use the initial symbolic markings of the components participating in the composition operation. This makes the initial composed marking a special case, since it can be determined

[1]The place fusion function is represented as a set of ordered pairs.

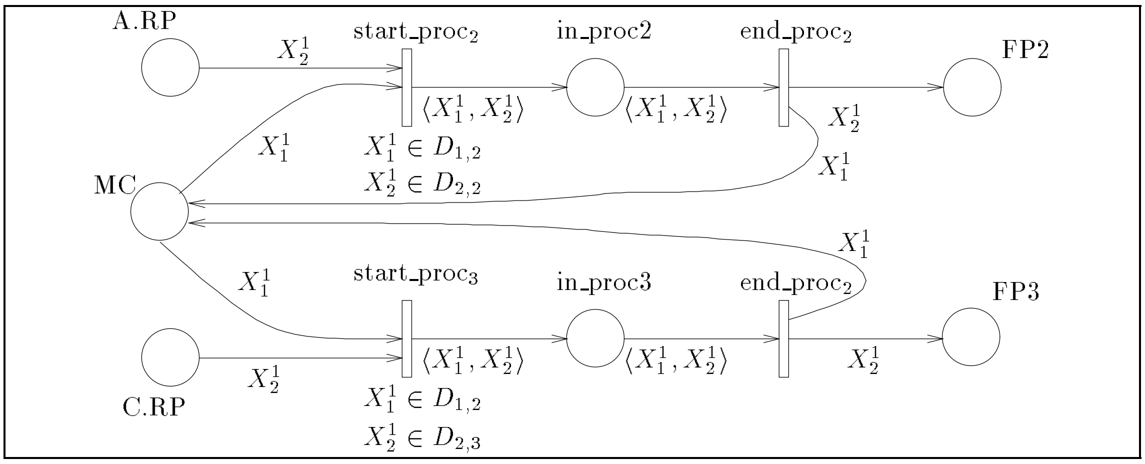


Figure 6.12: *cWN* modelling the processing of rough parts of type 2 and 3.

from symbolic markings and not as a consequence of a composed firing. We will denote by $\widehat{Z_i^j}$ both dynamic subclasses and composed dynamic subclasses. Their meaning will depend on the type of the marking components (symbolic markings or composed markings) that they belong to.

The representation $\widehat{\mathcal{R}_\mathcal{N}}$ of the initial composed symbolic marking $\widehat{\mathcal{M}_\mathcal{N}}$ will have the following characteristics:

- $\widehat{\mathcal{R}_\mathcal{N}}.m_i = \mathcal{E}_L.m_i + \mathcal{E}_R.m_i$;
  The set of dynamic subclasses of $\widehat{\mathcal{M}_\mathcal{N}}$ for a basic colour class C_i, is given by the union of the sets of dynamic subclasses of C_i defined in each marking component. The dynamic subclasses are numbered in such a way that the first $\mathcal{E}_L.m_i$ dynamic subclasses correspond to the dynamic subclasses of the left marking component, and the rest to the right marking component. The order of the dynamic subclasses within a marking component is preserved in the composed marking.
  $\widehat{C}_i = \mathcal{E}_L.\widehat{C}_i \cup \mathcal{E}_R.\widehat{C}_i$;

$$\forall C_i,\ j \in \{1, \ldots, \mathcal{E}_L.m_i\},\qquad \widehat{\mathcal{R}_\mathcal{N}}.\widehat{Z_i^j} = \mathcal{E}_L.\widehat{Z_i^j}$$

$$\forall C_i,\ j \in \{1, \ldots, \mathcal{E}_R.m_i\},\qquad \widehat{\mathcal{R}_\mathcal{N}}.\widehat{Z_i^k} = \mathcal{E}_R.\widehat{Z_i^j};\ \ (\text{with } k = j + \mathcal{E}_L.m_i)$$

- $\forall \widehat{Z_i^j} \in \widehat{\mathcal{R}_\mathcal{N}}.\widehat{C}_i$,

$$d(\widehat{\mathcal{R}_\mathcal{N}}.\widehat{Z_i^j}) = \begin{cases} d(\mathcal{E}_L.\widehat{Z_i^j}) & \text{if } j \le \mathcal{E}_L.m_i \\ d(\mathcal{E}_R.\widehat{Z_i^g}) & \text{otherwise (where } g = j - \mathcal{E}_L.m_i) \end{cases}$$

  where the function d is defined in the same way as for dynamic subclasses.

- $\forall \widehat{Z_i^j} \in \widehat{\mathcal{R}_{\mathcal{N}}}.\widehat{C}_i$,

$$card(\widehat{\mathcal{R}_{\mathcal{N}}}.\widehat{Z_i^j}) = \begin{cases} card(\mathcal{E}_L.\widehat{Z_i^j}) & \text{if } j \leq \mathcal{E}_L.m_i \\ card(\mathcal{E}_R.\widehat{Z_i^g}) & \text{otherwise (where } g = j - \mathcal{E}_L.m_i) \end{cases}$$

- $\forall p \in P_{\mathcal{N}}, \ \forall \hat{c} \in \widehat{\mathcal{R}_{\mathcal{N}}}.\widehat{C}(p)$,

$$mark(p)(\widehat{\mathcal{R}_{\mathcal{N}}}.\hat{c}) = \begin{cases} mark(p)(\mathcal{E}_L.\hat{c}') & \text{if } p \in P_L - FP_L \wedge \\ & \exists \hat{c}' \in \mathcal{E}_L.\widehat{C}(p), \ \hat{c}' = dec(\hat{c}) \\ \sum_{\{p':\nu(p')=p} mark(p')(\mathcal{E}_L.\hat{c}') & \text{if } p \in FP_R \wedge \\ & \exists \hat{c}' \in \mathcal{E}_L.\widehat{C}(p'), \ \hat{c}' = dec(\hat{c}) \\ mark(p)(\mathcal{E}_R.\hat{c}') & \text{if } p \in P_R \wedge \\ & \exists \hat{c}' \in \mathcal{E}_R.\widehat{C}(p), \ \hat{c}' = dec(\hat{c}) \end{cases}$$

where $\hat{c}$ represents a tuple of composed dynamic subclasses in the colour domain of p and dec is a function that translates the tuple $\hat{c} \in \widehat{\mathcal{R}_{\mathcal{N}}}.\widehat{C}(p)$, expressed in terms of the indexes of the composed dynamic subclasses of $\widehat{\mathcal{R}_{\mathcal{N}}}$, into a tuple expressed in terms of the indexes of the (composed) dynamic subclasses of the the sub-components.

6.4.2 Evolving from the initial composed marking

We have seen that the composed marking of one sub-component may be influenced by the composed marking of the other sub-component. This situation is formalised by the introduction of the concept of *composed sub-marking*. Intuitively, a composed sub-marking $\widehat{\mathcal{M}_{\mathcal{N}}^L}$ is a composed marking of a sub-component L of $\mathcal{N}$, such that its set of composed dynamic subclasses corresponds to the set of composed dynamic subclasses of $\widehat{\mathcal{M}_{\mathcal{N}}}$, and the distribution of the dynamic subclasses in the places of L is the same as for the corresponding places in $\mathcal{N}$. The objective behind this definition is to be able to obtain the set of enabled transition instances of the composed marking, from the set of transition instances of its marking components, reflecting the influence of one marking component on the marking of the other. Formally, a composed sub-marking is defined as:

Definition 6.8 (Composed sub-marking) *Consider a cWN $\mathcal{N}$ obtained from the composition of two cWNs L and R. The* composed sub-marking *of a composed marking $\widehat{\mathcal{M}_{\mathcal{N}}}$ of $\mathcal{N}$ with respect to the places of the sub-component L—denoted by $\widehat{\mathcal{M}_{\mathcal{N}}^L}$— is the composed marking of L defined by:*

$$\forall C_i, \ \widehat{\mathcal{M}_{\mathcal{N}}^L}.\widehat{C}_i = \widehat{\mathcal{M}_{\mathcal{N}}}.\widehat{C}_i, \quad and$$

$$\forall p \in P_L, \ \forall \hat{c} \in \widehat{\mathcal{M}_{\mathcal{N}}^L}.\widehat{C}(p),$$

$$mark(p)(\widehat{\mathcal{M}_{\mathcal{N}}^L}.\hat{c}) = \begin{cases} mark(p)(\widehat{\mathcal{M}_{\mathcal{N}}}.\hat{c}) & p \in P_L - FP_L \\ mark(\nu(p))(\widehat{\mathcal{M}_{\mathcal{N}}}.\hat{c}) & otherwise \end{cases}$$

For R the definition is the dual, i.e the composed marking of R defined by:

$$\forall C_i,\ \widehat{\mathcal{M}^R_{\mathcal{N}}}.\widehat{C}_i = \widehat{\mathcal{M}_{\mathcal{N}}}.\widehat{C}_i, \quad and$$
$$\forall p \in P_R,\ \forall \hat{c} \in \widehat{\mathcal{M}^R_{\mathcal{N}}}.\widehat{C}(p), mark(p)(\widehat{\mathcal{M}^R_{\mathcal{N}}}.\hat{c}) = mark(p)(\widehat{\mathcal{M}_{\mathcal{N}}}.\hat{c})$$

Recall that all places of R are places of $\mathcal{N}$.

Notice that the definition of a composed sub-marking may imply that within a lowest level marking component there are two or more dynamic subclasses of the composed marking representing the objects of the same colour.

A composed sub-marking will represent the marking of the fusion places. If we were to represent a composed marking in terms of its composed sub-markings the marking of the common places would be repeated in the sub-marking, thus they would be "counted" twice. For this reason we introduce the concept of *complement* of a composed sub-marking.

Definition 6.9 (Complement of a composed sub-marking)
Consider a composed marking $\widehat{\mathcal{M}_{\mathcal{N}}}$ of $\mathcal{N}$ and its composed sub-marking (let us take the left one) $\widehat{\mathcal{M}^L_{\mathcal{N}}}$. The complement *of the composed sub-marking $\widehat{\mathcal{M}^L_{\mathcal{N}}}$, denoted $\overline{\widehat{\mathcal{M}^R_{\mathcal{N}}}}$, is defined as the composed marking of R given by:*

$$\forall C_i,\ \overline{\widehat{\mathcal{M}^R_{\mathcal{N}}}}.\widehat{C}_i = \widehat{\mathcal{M}_{\mathcal{N}}}.\widehat{C}_i, \quad and$$
$$\forall p \in P_R,\ \ \forall \hat{c} \in \overline{\widehat{\mathcal{M}^R_{\mathcal{N}}}}.\widehat{C}(p),$$
$$mark(p)(\overline{\widehat{\mathcal{M}^R_{\mathcal{N}}}}.\hat{c}) = \begin{cases} mark(p)(\mathcal{E}_R.\hat{c}) & p \in P_R - FP_R \\ 0 & otherwise \end{cases}$$

The dual applies for the complement of the composed sub-marking $\widehat{\mathcal{M}^R_{\mathcal{N}}}$.

In general, a composed marking can then be represented as a composition of a composed sub-marking and its complement. In the case of the sequential composition the fusion function is not one-to-one. Therefore, if we represented the composed marking in this way, the marking of a fusion place is reproduced as many times as the number of output places fused to that particular fusion place. However, this will not constitute a problem, since for the sequential composition it is possible to always represent a composed marking in terms of its right hand-side sub-marking and its left hand-side complement. This is due to the fact that the final places cannot be input places to any transition in L, therefore their marking can only affect the enabled condition of transitions of R. This will be discussed with further detail in section 6.4.4.1.

6.4.2.1 Composition over a single component

Until now we have analysed the construction of the CRG when the operation involved is a binary composition operation, but what happens when the composition is done over a single component? For this case we introduce the concept of a *collapsed marking*.

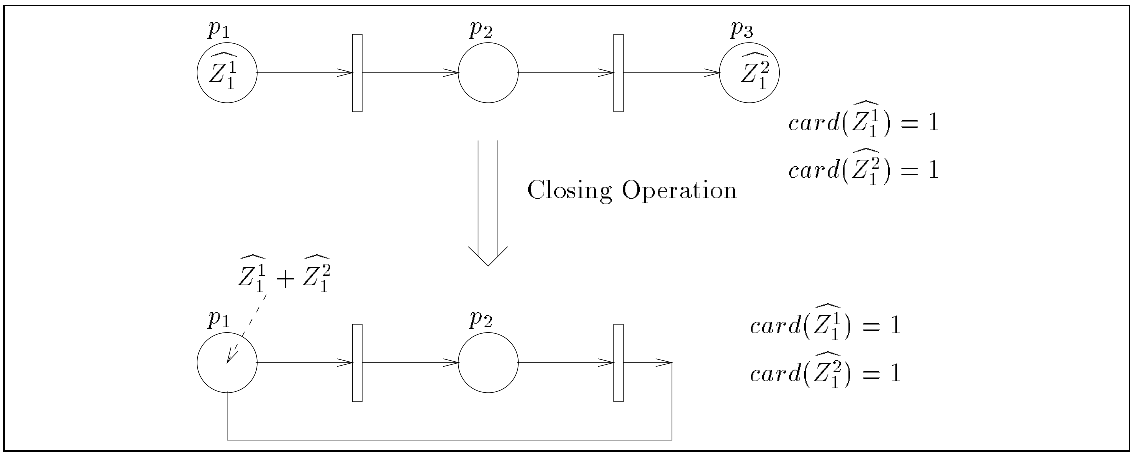


Figure 6.13: Example of a collapsed marking.

Recall that all composition operations involving place fusion within the same component define one-to-one functions from P_{dom} to P_{rng}, for which it holds either that $P_{dom} \subseteq ES$ or that $P_{dom} \subseteq FS$; and $P_{rng} \subset ES$, i.e. image places can only be entry places. The set of places resulting from the fusion of places in P_{dom} with places in P_{rng} will be denoted by P_{fuse}. We will denote by p_e a place in P_{dom}, by p_f a place in P_{rng}, and by p_{ef} the place in P_{fuse} resulting from the fusion of p_e with p_f.

Definition 6.10 (Collapsed marking) *Consider a composed marking $\widehat{\mathcal{M}_\mathcal{S}}$ of a cWN $\mathcal{S}$. The* collapsed marking *of $\widehat{\mathcal{M}_\mathcal{S}}$ is the composed marking $\widehat{\mathcal{M}_{\mathcal{S}'}}$ of $\mathcal{S}'$, obtained from $\widehat{\mathcal{M}_\mathcal{S}}$ as follows:*

$$\forall p \in P_{S'},\ \forall \hat{c} \in \widehat{C}_{\mathcal{S}'}(p),$$

$$mark(p)(\widehat{\mathcal{M}_{\mathcal{S}'}}.\hat{c}) = \begin{cases} mark(p_f)(\widehat{\mathcal{M}_\mathcal{S}}.\hat{c}) + mark(p_e)(\widehat{\mathcal{M}_\mathcal{S}}.\hat{c}) & \textit{if } p = p_{ef} \in P_{fuse} \\ mark(p)(\widehat{\mathcal{M}_\mathcal{S}}.\hat{c}) & \textit{otherwise} \end{cases}$$

The representation of a collapsed marking will be similar to that for the composed marking, except that the left component is considered to be empty. The set of places FP_L and FP_R will, respectively correspond to the sets P_{dom} and P_{rng} of $P_\mathcal{S}$.

Notice that the fact that $\widehat{\mathcal{M}_S}$ is minimal will not imply that $\widehat{\mathcal{M}_{S'}}$ is also minimal. It may be necessary to re-group the composed dynamic subclasses of $\widehat{\mathcal{M}_S}$ and to order the new composed dynamic subclasses to obtain a canonical representation. Figure 6.13 shows an example of how, when applying a closing operation over a *cWN*, it is possible to obtain a non-minimal composed marking although the original composed marking was minimal. Let us suppose that $\widehat{Z_1^1}$ and $\widehat{Z_1^2}$ are defined over the same lowest level component and over the same static subclass. Then they can be grouped into a single composed dynamic subclass since they are defined over the same static subclass and have the same marking distribution.

The concept of composed sub-marking is also not applicable in the context of unary composition operations. Nevertheless, we still want to be able to obtain the marking component that generates a marking $\mathcal{M}_{\mathcal{S}'}$ of $\mathcal{S}'$. The marking of a place p_{ef} resulting from a fusion, could be assigned to either the place p_e in the domain of the fusion function or to p_f in the range. Following this idea we introduce the concept of *extended marking.*

Definition 6.11 (Extended composed marking) *Consider a cWN $\mathcal{S}'$ obtained by an unary composition operation over a cWN $\mathcal{S}$. Given a composed marking $\widehat{\mathcal{M}_{\mathcal{S}'}}$ of $\mathcal{S}'$ the* extended composed marking *of $\widehat{\mathcal{M}_{\mathcal{S}'}}$ with respect to the places in its domain, is the composed marking $\widehat{\mathcal{M}_{\mathcal{S}}^{dom}}$ of $\mathcal{S}$, defined as:*

$$\forall p \in P_S, \forall \hat{c} \in \widehat{C}(p),$$

$$mark(p)(\widehat{\mathcal{M}_{\mathcal{S}}^{dom}}.\hat{c}) = \begin{cases} mark(\nu(p))(\widehat{\mathcal{M}_{\mathcal{S}'}}.\hat{c}) & \text{if } p \in P_{dom} \\ mark(p)(\widehat{\mathcal{M}_{\mathcal{S}'}}.\hat{c}) & \text{if } p \notin P_{rng} \cup P_{dom} \\ 0 & \text{otherwise} \end{cases}$$

Recall that in composition operations that involve place fusion over single components the fusion function is one-to-one. A place resulting from a fusion is identified by the name of the corresponding place in the range. The extended composed marking of marking $\widehat{\mathcal{M}_{\mathcal{S}'}}$ with respect to the places in the range, denoted $\widehat{\mathcal{M}_{\mathcal{S}}^{rng}}.\hat{c}$, will be given by:

$$\forall p \in P_S, \forall \hat{c} \in \widehat{C}(p),$$

$$mark(p)(\widehat{\mathcal{M}_{\mathcal{S}}^{rng}}.\hat{c}) = \begin{cases} mark(p)(\widehat{\mathcal{M}_{\mathcal{S}'}}.\hat{c}) & \text{if } p \notin P_{dom} \\ 0 & \text{otherwise} \end{cases}$$

6.4.3 Canonical Composed marking

In order to be able to build an effective algorithm for the construction of the CRG we need to be able to uniquely identify each composed marking. In the way that it has been defined, the representation of a composed marking may not be unique. We can obtain the same set of symbolic markings of the resulting $cWN\ \mathcal{N}$ by composing different pairs of symbolic markings of the components. This could lead to false information about the system evolution, for example not detecting that we can return to a composed marking, because of its different representations. The definition of a unequivocal criterion for the representation of a composed marking requires the concepts of a *minimal composed marking* and an *ordered composed marking*. These two conditions will define a *canonical composed marking*.

Minimality As for symbolic markings, the minimality requirement refers to the number of dynamic subclasses for each basic class [CDFH93]. We want to have the smallest possible number of composed dynamic subclasses since this allows us to economise on both the marking representation and the number of possible firing instances. In the case of composed markings, a marking component may have composed dynamic subclasses that do not belong to it. For example, this would happen in the case of a composed sub-marking when the composition operation involves place fusion and there exist places that are being fused that have non-zero markings. At the level of composed marking we have not determined all the possible intersections between the sets of coloured objects represented by the dynamic subclasses of the marking components. Therefore, the minimality criterion for symbolic markings is only applicable over dynamic subclasses that originally belong to the same symbolic marking, i.e. to the same lowest level component of the composed marking. At the level of composed markings, two composed dynamic subclasses can be grouped if they intersect the same set of dynamic subclasses of the lowest level components.

Definition 6.12 (Minimal composed marking) *A composed marking representation $\widehat{\mathcal{R}_{\mathcal{N}}}$ is* minimal *if and only if for all lowest level marking components its set of dynamic subclasses is minimal according to the minimality criterion of SRGs, and*

$$\begin{aligned}\forall i \leq h,\ \forall j,k, \quad j \neq k \Rightarrow & (dyn\_ss_k \neq dyn\_ss_j) \ \vee \ (mark_{sp}(\widehat{Z_i^j}) \neq mark_{sp}(\widehat{Z_i^k})) \\ & \vee \ (d(\widehat{Z_i^j}) \neq d(\widehat{Z_i^k}))\end{aligned}$$

and

$$\begin{aligned}\forall i > h,\ \forall j,k, \quad k = \oplus j \Rightarrow & (dyn\_ss_k \neq dyn\_ss_j) \ \vee \ (mark_{sp}(\widehat{Z_i^j}) \neq mark_{sp}(\widehat{Z_i^k})) \\ & \vee \ (d(\widehat{Z_i^j}) \neq d(\widehat{Z_i^k}))\end{aligned}$$

where dyn_ss_k and dyn_ss_j are the sets of dynamic subclasses of the lowest level marking components that are intersected by the composed dynamic subclasses Z_i^k and Z_i^j,respectively, $\oplus$ is the successor function over dynamic subclasses of an ordered class, h is the number of non-ordered basic colour classes and the function $mark_{sp}$ is the marking projection function defined in Section 6.2.2.

Ordering The ordering criterion that is defined works on three levels. The first one refers to the indexing of dynamic subclasses at the level of lowest level components. Within a lowest level component the ordering process of the SRG algorithm is applied. The indexes of the dynamic subclasses are updated within the intersection sets. The second one works on the indexing of dynamic subclasses at the level of composed markings. This refers to ordering of the triples representing the dynamic subclasses intersected in a composed dynamic subclasses and to the ordering of the indexes of the composed dynamic subclasses within the composed marking. The set of dynamic subclasses participating in an intersection is ordered according to the lowest level component they belong to, as described in the definition of composed dynamic subclasses (see Definition 6.5). The composed markings are then indexed according to their intersection sets (dyn_ss). Given two composed dynamic subclasses $Z_i^a = \langle dyn\_ss_a, card_a, q_a, a\rangle$ and $Z_i^b = \langle dyn\_ss_b, card_b, q_b, b\rangle$ of a composed marking, it holds that $a < b$ if and only if $q_a < q_b$; or $q_a = q_b$ and for their first elements in dyn_ss_a and dyn_ss_b, respectively (com_k, i, j) and (com_g, i, h), it holds that $k < g$ or $(g = k \ \wedge \ j < h)$. For example given the following composed dynamic subclasses:

$$Z_1^a = \langle\{(1,1,3),(2,1,2)\},1,2,a\rangle$$

and

$$Z_1^b = \langle\{(1,1,2)\},2,2,b\rangle$$

it must hold that $b < a$. Finally the third ordering level refers to the marking components of a composed marking. A composed marking can be represented in at least two ways: given a composed marking $\widehat{\mathcal{M}_\mathcal{N}}$ it can be represented as the composition of the composed markings $\widehat{\mathcal{M}^L_\mathcal{N}}$ and $\overline{\widehat{\mathcal{M}^R_\mathcal{N}}}$ or as the composition of $\widehat{\mathcal{M}^R_\mathcal{N}}$ and $\overline{\widehat{\mathcal{M}^L_\mathcal{N}}}$. The third ordering criterion enforces that a composed marking must always be represented as a composition of a right hand side composed submarking $\widehat{\mathcal{M}^R_\mathcal{N}}$ with its corresponding left hand side $\overline{\widehat{\mathcal{M}^L_\mathcal{N}}}$ complement. It has been decided this way because, in the presence of place fusion, the places of the right-hand side components are the ones that represent the fused places.

Definition 6.13 (Ordered composed marking)
A composed marking representation is ordered *if and only if*

- *it is represented as the composition of a right hand side composed sub-marking with its corresponding left hand side complement;*
- *for all composed dynamic subclasses in the composed marking the set of dynamic subclasses intersected in the composed dynamic subclass is ordered according to the criterion presented in Definition 6.5; and given two composed dynamic subclasses* $Z_i^a = \langle dyn\_ss_a, card_a, q_a, a\rangle$ *and* $Z_i^b = \langle dyn\_ss_b, card_b, q_b, b\rangle$ *of a composed marking, it holds that* $a < b$ *if and only if* $q_a < q_b$*; or* $q_a = q_b$ *and for their first elements in* dyn_ss_a *and* dyn_ss_b*, respectively* (com_k, i, j) *and* (com_g, i, h)*, it holds that* $k < g$ *or* $(g = k \ \wedge \ h < j)$*; and*
- *for all its lowest level components it holds that the indexes of its dynamic subclasses are ordered according to the ordering criterion for symbolic markings.*

We want dynamic subclasses of a sub-component to be continuously numbered, therefore the readjustment of dynamic subclasses at the level of symbolic markings will generate the readjustment of dynamic subclasses at the level of the composed marking.

6.4.4 Composed Firing

The idea of a composed firing is to obtain the set of reachable composed markings from a composed marking without having to generate the symbolic markings it represents. Additionally, we want to use the information given by the CRGs and/or SRGs of the sub-components of the *cWN*.

Symbolic transition instances are defined at the level of symbolic markings. At this level parameters of a transition are associated with dynamic subclasses of the symbolic marking. We have already introduced dynamic subclasses of composed markings. In the same way we need to refer to transition instances of a composed marking. In principle, we could consider that transition instances of symbolic markings are the same as transition instances of a composed marking. The difference would be that the function λ does not return dynamic subclasses of a symbolic marking but composed dynamic subclasses of a composed marking. However, the problem is not this straightforward, since we must consider that a transition of a higher level component might be the result of the synchronisation of two transitions of lower level components. In the construction of the SRG of a WN we had that a symbolic transition instance represented a set of basic colour transition instances. In a similar manner a *composed transition instance* will represent a set of symbolic transition instances.

Definition 6.14 (Composed transition instance)
A composed transition instance *is either :*

- *a symbolic transition instance of a component of the composed marking, with the function λ defined over composed dynamic subclasses, rather than dynamic subclasses of symbolic markings; or*
- *the synchronisation of two composed transition instances (one from each component);*

Formally, it is represented by a triple :

$$\langle t_{\mathcal{N}}, \lambda_{\mathcal{N}}, \mu_{\mathcal{N}} \rangle$$

where

- *$t_{\mathcal{N}}$ is either a transition of L, a transition of R, or a transition resulting from a synchronisation of a transition of L with a transition of R;*
- *$\lambda_{\mathcal{N}}$ is defined over the set of parameters of $t_{\mathcal{N}}$; for each parameter it determines the composed dynamic subclass from which the parameter will be instantiated; and*
- *$\mu_{\mathcal{N}}$ depends on the cardinalities of the composed dynamic subclasses of $\widehat{\mathcal{M}_{\mathcal{N}}}$ selected for the parameter of $t_{\mathcal{N}}$; for each variable X_i^j of $t_{\mathcal{N}}$ it instantiates it with a member of the corresponding composed dynamic subclass selected by $\lambda_{\mathcal{N}}$.*

6.4.4.1 Enabled composed transition instances

To determine which is the set of enabled composed transition instances of a composed marking we examine the set of enabled composed transition instances of its sub-markings. The decomposition of the composed marking into sub-markings is determined by the composition operation used to create the component with which the composed marking is associated.

A composed sub-marking, say the left one ($\widehat{\mathcal{M}^L_{\mathcal{N}}}$), of a composed marking $\widehat{\mathcal{M}_{\mathcal{N}}}$ may not be included in the CRG of the corresponding sub-component. In this case, it is necessary to obtain the set of enabled composed transition instances of the composed sub-marking. One way to do this would be to obtain all the possible symbolic markings of the composed sub-marking and for each one, obtain the set of enabled symbolic transition instances. However, this contradicts our goal of building the CRG working at the level of composed markings. The chosen alternative is to apply a top down approach and consider the sub-marking as a composed marking, and continue until we reach the lowest level components or a sub-marking which is in the CRG of its corresponding sub-component. Composed markings that do not belong to the CRG of its corresponding component are incorporated to it.

Under sequential composition Consider the composed marking $\widehat{\mathcal{M}_{\mathcal{N}}}$ of a cWN $\mathcal{N}$ resulting from a sequential composition of a cWN L with another R $(L; R)$. We know that no places in FS'_L can be input to any transition in L. This means that the marking of the places in FS'_L can only affect the enabled condition of transitions in R. Therefore, the set of enabled composed transition instances of the composed marking $\widehat{\mathcal{M}_{\mathcal{N}}}$ will be given by the union of the set of enabled transition instances in :

$$\overline{\widehat{\mathcal{M}^{L}_{\mathcal{N}}}} \text{ and } \widehat{\mathcal{M}^{R}_{\mathcal{N}}}$$

Under independent parallel composition Under independent parallel composition there is no place fusion involved, therefore the marking of one of the components will not affect the marking of the other when composed. This means that the set of enabled composed instances of a composed marking will be formed by the union of the sets of enabled composed instances of its composed sub-markings. The composed marking will be decomposed into:

$$\widehat{\mathcal{M}^{L}_{\mathcal{N}}} \text{ and } \widehat{\mathcal{M}^{R}_{\mathcal{N}}}$$

It is decomposed into its sub-markings and not marking components because we want to work at the level of composed dynamic sub-classes.

Under choice composition If the choice place has marking zero, then the decomposition is equal to that under independent parallel composition. In the case where the choice place is different than zero there will be composed transitions instances of the output transitions of the choice place that will be enabled. All enabled transition instances of the sub-components will remain enabled. The process of determining the enabled composed transitions instances will then consist on decomposing the composed marking into its respective sub-markings and verifying the enabled condition of the transitions output to the choice composition.

Under competing parallelism composition between two components Competing parallelism composes two *cWN*s, L and R, by fusing places that belong to the entry sets (ES) of the components. A one-to-one function $\Lambda : ES'_L \rightarrow ES_R$, where $ES'_L \subseteq ES_L$, defines the ordered pairs of places to be fused. This means that the marking of a sub-component can affect the marking of the other. The composed marking has then to be decomposed into its sub-markings:

$$\widehat{\mathcal{M}^L_{\mathcal{N}}} \text{ and } \widehat{\mathcal{M}^R_{\mathcal{N}}}$$

The set of enabled composed transition instances will be formed by the set of composed transitions instances in each of the sub-markings. The marking of the places participating in the operation is represented in each of the sub-markings.

Under competing parallelism composition over a single component The competing parallelism operation over a *cWN* S, fuses pairs of places of the entry set of S (ES_S) to produce a *cWN* S'. The following property simplifies the process of determining the set of enabled composed transitions instances in a composed marking of S'.

Proposition 6.1 *The symbolic transition instances enabled in a composed marking* $\widehat{\mathcal{M}_S}$ *of* S*, will also be enabled in its collapsed composed marking* $\widehat{\mathcal{M}_{S'}}$ *of* S'*.*

Proof.- Let us call $\widehat{R_S}$ the representation of the composed marking $\widehat{\mathcal{M}_S}$. For a composed transition instance $[t, \lambda, \mu]$ to be enabled in $\widehat{\mathcal{M}_S}$, it must hold that:

$$\forall p \in P_S,\ W^-(p,t)(\lambda,\mu) \leq R_S.mark(p) \text{and} \Phi(t)(\lambda,\mu)$$

This last condition refers to the fact that the standard predicate associated with the transition must evaluate to TRUE.

The marking of a fusion place is obtained by the sum of the markings of the places that are fused. Therefore, the marking of a dynamic subclass in a place cannot decrease, maintaining in this way the enabled state. Places not involved in the fusion operation maintain the marking of the original composed marking $\widehat{\mathcal{M}_S}$. We can then conclude that enabled composed transition instances in $\widehat{\mathcal{M}_S}$ will also be enabled composed instances in $\widehat{\mathcal{M}_{S'}}$. ■

However, this does not mean that there cannot be new composed transition instances enabled in $\widehat{\mathcal{M}_{S'}}$ that were not enabled in $\widehat{\mathcal{M}_S}$. Only if it holds that the marking of all places participating in the fusion is zero, can we guarantee that the set of enabled composed transition instances of $\widehat{\mathcal{M}_{S'}}$ be that of $\widehat{\mathcal{M}_S}$. If this is not the case, then the set of enabled symbolic transition instances of $\widehat{\mathcal{M}_S}$ may only be part of the set of enabled symbolic transition instances of $\widehat{\mathcal{M}_{S'}}$. It is then necessary to check if any other composed transition instance, apart from those already enabled in $\widehat{\mathcal{M}_S}$, is enabled in $\widehat{\mathcal{M}_{S'}}$.

Under closing composition The closing operation is defined over a single cWN S. It fuses a final place of S, denoted p_f, with an entry place of S, denoted p_e, producing a place p_{ef}. The resulting cWN will be denoted S'. The place p_{ef} will be an entry place of $ES_{S'}$, i.e. $ES_{S'} = ES_S - \{p_e\} \cup \{p_{ef}\}$.

In a composed marking $\widehat{\mathcal{M}_S}$ where the place p_f has a non-zero marking it is possible that, when fusing p_f with p_e, composed transition instances that were not enabled in $\widehat{\mathcal{M}_S}$ become enabled in $\widehat{\mathcal{M}_{S'}}$. Here we can apply the same approach as for the competing parallelism over a single component, and use the information of $E(\widehat{\mathcal{M}_S})$ (all composed transition instances enabled in $\widehat{\mathcal{M}_S}$ will be enabled in $\widehat{\mathcal{M}_{S'}}$).

Under synchronisation Consider a composed marking $\widehat{\mathcal{M}_\mathcal{N}}$ of $\mathcal{N}$ and its marking components $\widehat{\mathcal{E}_L}$ and $\widehat{\mathcal{E}_R}$. As for the case of independent parallel composition there is no place fusion involved. Therefore, in general, the set of enabled composed transition instances of $\widehat{\mathcal{M}_\mathcal{N}}$ will be obtained from the union of the sets of enabled transition instances of its marking components. The enabled condition of the transition t_{lr}, resulting from the fusion of $t_l \in T_L$ and $t_r \in T_R$, will be determined by the enabled condition of composed instances of t_l and t_r in $\widehat{\mathcal{E}_L}$ and $\widehat{\mathcal{E}_R}$, respectively, and by the predicates defined over the fused transition t_{lr}, that can relate variables of t_l with variables of t_r.

Given the marking components $\widehat{\mathcal{E}_L}$ and $\widehat{\mathcal{E}_R}$ such that $t_l \in \mathrm{E}(\widehat{\mathcal{E}_L})$ and $t_r \in \mathrm{E}(\widehat{\mathcal{E}_R})$, we need to obtain the combinations of transition instances of t_l and t_r enabled in $\widehat{\mathcal{E}_L}$ and $\widehat{\mathcal{E}_R}$, respectively, that satisfy the predicates of t_{lr}. An expensive solution would be to consider all possible combinations and verify if they satisfy the predicates. The alternative, more efficient, solution is to form the pairs of enabled transition instances of t_{lr} according to the transition's predicates, similarly to the process of searching for the enabled transition instances in the SRG algorithm.

Given the restriction that transitions with common input or output places cannot be synchronised, the analysis for the synchronisation of transitions within the same component does not differ from the case of synchronisation between different components. The set of places of a cWN $\mathcal{S}$ and of the cWN $\mathcal{S}'$, obtained after the synchronisation of two transitions in $\mathcal{S}$, is the same. Therefore, in principle a composed marking of $\widehat{\mathcal{M}_\mathcal{S}}$ of $\mathcal{S}$, could be a composed marking of $\mathcal{S}'$.

As for the case of two components, we must consider that the synchronisation of the transitions and the predicates of the synchronised transition t_{lr}.

6.4.4.2 Firing of an enabled composed transition instance

The set of Directly Reachable Markings (DRS) from a composed marking $\widehat{\mathcal{M}}$ is formed by the set of composed markings generated by the firing of the enabled composed transition instances in $\widehat{\mathcal{M}}$. In the construction of the SRG the DRS of a symbolic marking is obtained by firing its enabled symbolic transition instances. In composed markings this process is slightly more complicated.

As we have studied in the previous section, the set of enabled composed transition instances of a composed marking can be obtained by recursively analysing the sub-markings of the composed marking and using the information about the set of enabled composed transition instances in the sub-markings. In the case where—at a certain level of decomposition—the composed sub-marking belongs to the CRG of its respective sub-component, the composed sub-marking obtained from firing an enabled composed transition instance can be obtained from the CRG without the need to perform the firing of the composed transition instance. This step can be done for all composed transition instances enabled in the original composed marking. If we arrive at the lowest level and the composed sub-marking does not belong to the CRG of the sub-component, then it is necessary to perform the *composed firing* of the enabled composed transition instances.

Based on the definition of a symbolic firing, a *composed firing* consists of the following four steps:

Splitting : The process of splitting composed dynamic subclasses is a generalisation of the splitting process of dynamic subclasses of symbolic markings. The splitting of a composed dynamic subclass will imply the splitting of the dynamic subclasses that it intersects (the set of dynamic subclass in *dyn_ss*). The splitting of a composed dynamic subclass in a sub-marking must be reflected in the composed marking it originated from. This means that it is reflected in all the complementing sub-markings of the sub-marking being analysed.

Firing : The firing step basically corresponds to the firing step from a symbolic marking, the difference being that here it is done on split composed dynamic subclasses.

Minimisation : The minimality criterion is applied to eliminated redundant (composed) dynamic subclasses. It is first applied over the sets of dynamic subclasses of the lowest level components. This is reflected at the level of dynamic subclasses formed by intersection, and then we apply the minimality over composed dynamic subclasses.

Ordering : This step consists of the renaming of the dynamic subclasses of the composed markings obtained, following the definition of an ordered composed marking. The ordering of the composed dynamic subclasses is only performed at the level of the composed sub-marking.

6.4.5 Obtaining the set of directly reachable composed markings from a composed marking

In general, the composed sub-marking obtained from the firing of composed transition instance $[t, \lambda, \mu]$ in a composed sub-marking of the original composed marking $\widehat{\mathcal{M}}$, must be composed with its complement to obtain the composed marking $\widehat{\mathcal{M}'}$ reachable from $\widehat{\mathcal{M}}$ by the firing of $[t, \lambda, \mu]$ (assuming that t is not formed by the synchronisation of transitions). This will be termed the *reconstruction process*. The marking formed by the composition of the two sub-markings must then be reordered applying the reordering steps of higher level components as explained in Section 6.4.3.

The process of obtaining the DRS of a composed marking will then be formed by the following steps:

1. *Decompose*: using knowledge about the composition operation applied to construct the component, this step decomposes the composed marking $\widehat{\mathcal{M}_{\mathcal{N}}}$ into sub-markings;

2. *Search*: For each composed sub-marking find the set of enabled transition instances. These first two steps are recursive.

3. *Build $E(\widehat{\mathcal{M}_{\mathcal{N}}})$*: obtain the set of enabled composed transition instances of the composed marking.

4. *Fire*: For each sub-marking with enabled composed transition instances obtain its DRS.

5. *Reconstruct*: Re-group the sub-markings in the DRSs with their respective complements to create a composed marking of higher levels of composition.

6. *Order*: Apply ordering criterion to the composed markings obtained in the previous step. These last two stages use the inverse recursion of the first two.

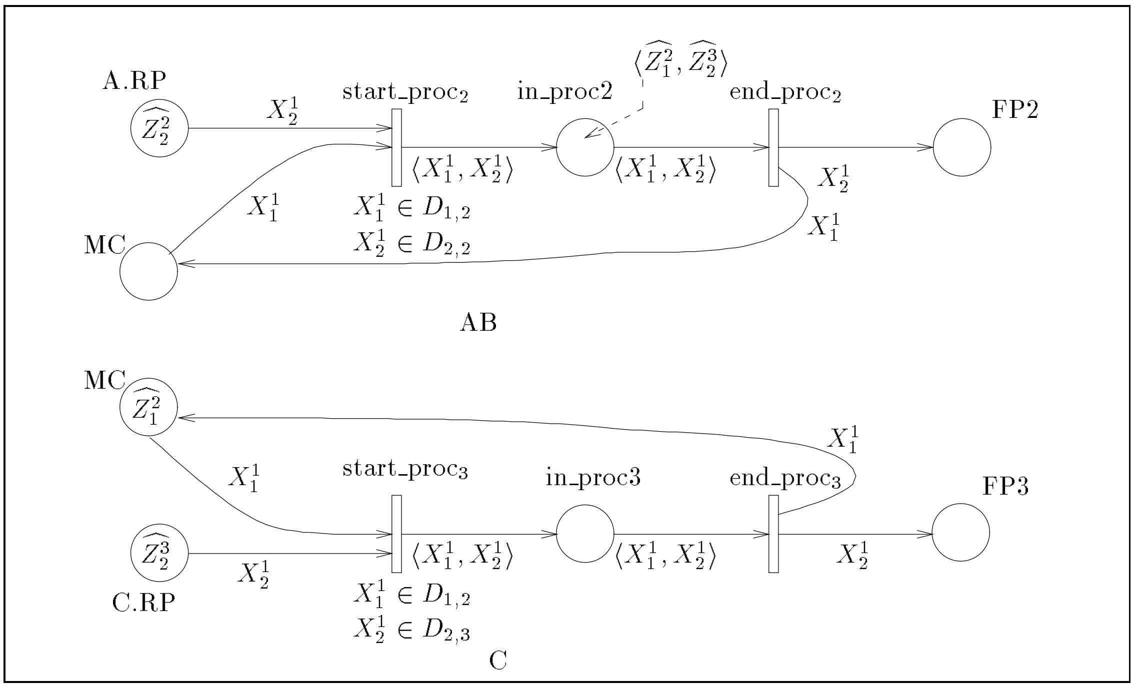


Figure 6.14: Markings components of $\widehat{\mathcal{M}_{\mathcal{N}}}$

Example of the construction of the DRS of a composed marking To aid the understanding of the concepts related to composed markings, we introduce the following condensed notation. A composed marking will be formed by a group of tuples, each one corresponding to a lowest level component of the marking. The number of elements in each tuple will be given by the number of places in the lowest level component. Each element of a tuple represents the marking of the place in terms of the dynamic subclasses of the composed markings of the lowest level component.

Given the composed markings of AB and C, as shown in Figure 6.14, in Figure 6.15 we present the steps required to obtain the set of directly reachable composed markings from the composed marking $\widehat{\mathcal{M}_{\mathcal{N}}}$ of the $cWN\,\mathcal{N}$, generated by applying the competing parallelism composition over the components AB and C, with $\Lambda(AB.MC) = C.MC$. The composed marking $\widehat{\mathcal{M}_{\mathcal{N}}}$ is represented as:[2]

$$\widehat{\mathcal{M}_{\mathcal{N}}} = (\underbrace{[\overset{MC}{0}, \overset{RP}{\widehat{Z_2^2}}, \overset{inproc}{\langle 0,0\rangle}]}_{A} \mid \underbrace{[\overset{inproc}{\langle \widehat{Z_1^2}, \widehat{Z_2^3}\rangle}, \overset{MC}{0}, \overset{FP2}{0}]}_{B}) \mid \underbrace{[\overset{MC}{\widehat{Z_1^3}}, \overset{RP}{\widehat{Z_2^4}}, \overset{inproc}{\langle 0,0\rangle}, \overset{FP3}{0}]}_{C}$$

Underlined tuples correspond to the component being analysed and the leaves of the trees are directly reachable composed markings.

[2]The indexes of the dynamic subclasses have been changed when representing the composed marking.

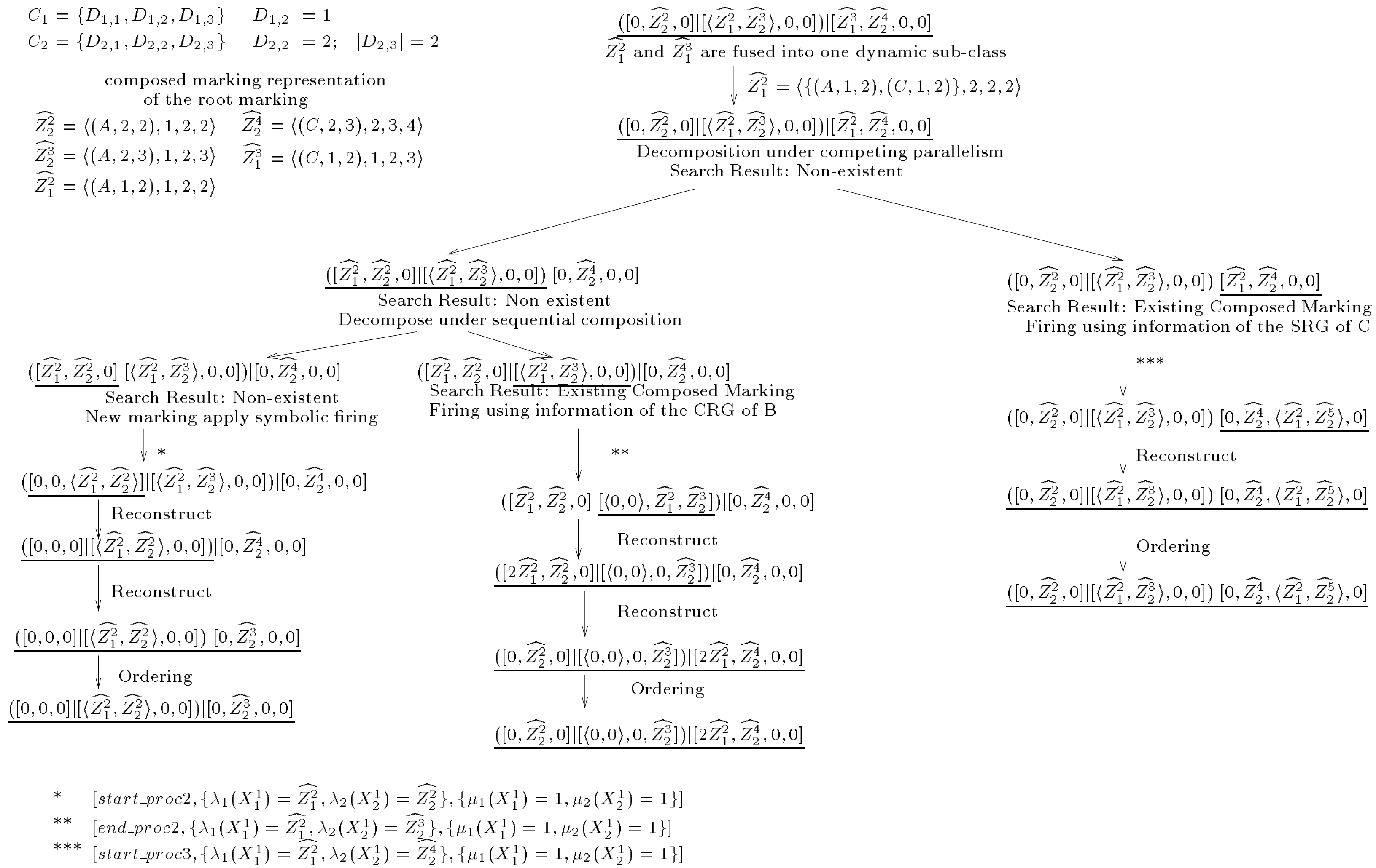


Figure 6.15: Example showing the construction of the CRG of $\mathcal{N}$.

Notice that the composed dynamic subclasses $\widehat{Z_1^1}$ and $\widehat{Z_1^4}$ exist although they are not present in the condensed notation. They represent the coloured objects with colours in the static subclasses $D_{1,1}$ and $D_{1,3}$ (machines of types 1 and 3), respectively. In the same manner the composed dynamic subclass $\widehat{Z_2^1}$, representing parts of type 1, exists but is not present in the condensed notation, since its marking is zero for all places of $\mathcal{N}$. The composed dynamic sub-classes $\widehat{Z_1^2}$ and $\widehat{Z_1^3}$ can be fused into one given that they both represent all the elements in the static sub-class $D_{1,2}$. In this example the composed marking $\widehat{\mathcal{M}_{\mathcal{N}}}$ is also a symbolic marking, since there is only one possible intersection between the composed dynamic subclasses of each of the components.

6.4.6 Description of algorithm for the generation of the CRG

Having analysed the process of building the DRS from a composed marking we now present an extended description of the algorithm to construct the CRG of a $cWN_{\mathcal{N}}$.

Algorithm 6.3.- **(Algorithm for the generation of the CRG of a** *cWN*)

Initialisation Generate the initial composed marking, including it in the set of unvisited composed markings.

Main iteration For each unvisited marking do

Decomposition and Search.-
Recursively decompose the composed marking and search for enabled composed transition instances. In the case of reaching a lowest level marking component, compute the set of enabled composed transition instances of the lowest level component using the SRG algorithm. In the search for composed transition instances we also consider the possibility of intersection of dynamic subclasses. If the composed sub-marking belongs to the CRG of the corresponding sub-component, obtain the information of the set of enabled composed transition instances from the CRG of the sub-component.

Building the set of enabled composed transition instances.-
In this stage, we obtain the set of enabled composed transition instances by considering the marking components at each level of decomposition until the top level is reached.
In the case of synchronisation composition, it is necessary to verify if the transitions being synchronised are both enabled and that the predicates of the resulting transition are satisfied in the composed transition instance.
Notice that if there is a composed marking where for all its lowest level marking components there is only one dynamic subclass per static subclass, then there is only one possible combination. Therefore, there is only one SRG. In order to avoid unnecessary redundant combinations in such composed markings the algorithm intersects, for each static subclass, all the lowest level dynamic subclasses of that class.

Obtaining the DRS.-
Having the set of enabled composed transition instances and the information

about the marking component where each one is enabled, obtain the DRS of composed markings from the composed marking being visited. The minimality and ordering processes must be applied to each composed marking obtained.

Including composed markings of the DRS in the CRG.- For each element in the DRS check if it is already in the CRG. If it is not, include it and add the corresponding arc, labelled with the composed transition instance. A composed marking is considered to be included in the CRG if it has the same representation as a composed marking in the CRG, or if its representation is a restriction of the representation of a composed marking in the CRG (see Definition 6.6).

6.5 The relation between the CRG and the SRG

In principle, the CRG of a *cWN* system represents the set of SRGs of the *cWN* systems associated with all the valid combinations of the initial symbolic markings of the lowest level components. What we want to know is if, by constructing the CRG of a *cWN* instead of all its possible SRGs according to the initial symbolic markings of its lowest level components, we can obtain the same qualitative behaviour information about the *cWN* systems.

The first property that we propose establishes the relation between the CRG and the symbolic markings of its SRGs from the point of view of reachability of markings.

Proposition 6.2 (Relation between composed and symbolic reachability) *Let $\widehat{\mathcal{M}}$ be a composed marking, $\mathcal{M}$ a symbolic marking in $\widehat{\mathcal{M}}$, and $[\widehat{\mathcal{M}}\rangle$ be the set of composed markings reachable from $\widehat{\mathcal{M}}$. Then*

$$\bigcup_{\mathcal{M}\in\widehat{\mathcal{M}}} [\mathcal{M}\rangle \subseteq [\widehat{\mathcal{M}}\rangle$$

Proof.- The proposition suggests that all symbolic markings reachable from the symbolic markings represented by the composed marking $\widehat{\mathcal{M}}$, are symbolic markings represented by the composed markings reachable from $\widehat{\mathcal{M}}$, i.e.

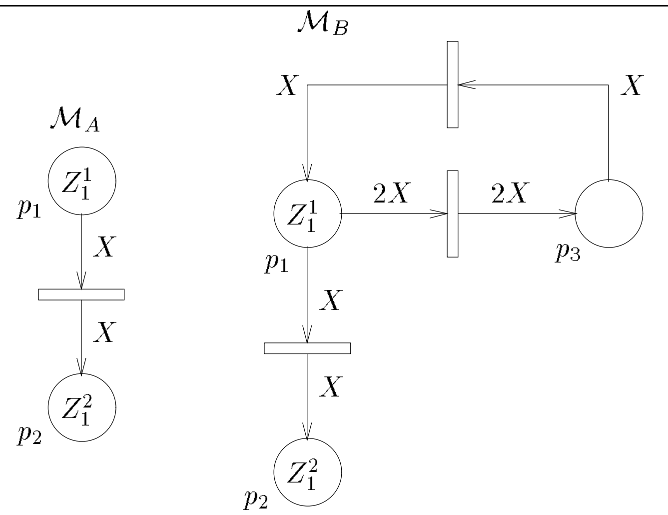
$$\forall \mathcal{M} \in \widehat{\mathcal{M}},\ \forall \mathcal{M}' \in [\mathcal{M}\rangle,\ \exists \widehat{\mathcal{M}}' \text{ such that } \mathcal{M}' \in \widehat{\mathcal{M}}'$$

We will prove this by contradiction. Let us suppose that there is a symbolic marking $\mathcal{M}'$ reachable from $\mathcal{M} \in \widehat{\mathcal{M}}$ such that $\forall \widehat{\mathcal{M}} \in [\widehat{\mathcal{M}}\rangle,\ \mathcal{M}' \notin \widehat{\mathcal{M}}$.

If $\mathcal{M}' \in [\mathcal{M}\rangle$ then there is an enabled symbolic transition instance $[t, \lambda, \mu]$ which when fired produces the symbolic marking $\mathcal{M}'$. The symbolic marking $\mathcal{M}$ defines an intersection between the dynamic sub-classes of the lowest level marking components of $\widehat{\mathcal{M}}$. This means that from the symbolic transition instance

$[t, \lambda, \mu]$ we can obtain a composed transition instance $[t, \lambda', \mu']$ generalising the intersection of dynamic sub-classes defined by $\mathcal{M}$. The composed firing process preserves the dynamic sub-class intersections defined and reflects the splitting of composed dynamic sub-classes at the level of the lowest level marking components. By applying the intersections defined at the level of the composed marking $\widehat{\mathcal{M}}'$ (obtained by $\widehat{\mathcal{M}}[t, \lambda', \mu'\rangle$), we obtain a symbolic marking $\mathcal{M}''$ that is reachable from $\mathcal{M}$ by the firing of the symbolic transition instance $[t, \lambda, \mu]$. Since $\mathcal{M}'$ does not belong to any $\widehat{\mathcal{M}'} \in [\widehat{\mathcal{M}}\rangle$, then $\mathcal{M}' \neq \mathcal{M}''$. This would mean that there are two symbolic markings reachable from the symbolic marking $\mathcal{M}$ by the firing of the symbolic transition instance $[t, \lambda, \mu]$. This is not possible, therefore, we arrive at a contradiction. ■

The opposite, $\cup_{\mathcal{M} \in \widehat{\mathcal{M}}} [\mathcal{M}\rangle \supset [\widehat{\mathcal{M}}\rangle$, is not true. This can be seen in the counter-example presented in Figure 6.16, where the components A and B are composed by competing parallelism over their places p_1.



$$|D_{1,q}| = 5$$
$$d(\mathcal{M}_A.Z_1^1) = d(\mathcal{M}_A.Z_1^2) = d(\mathcal{M}_B.Z_1^1) = d(\mathcal{M}_B.Z_1^2)$$
$$card(\mathcal{M}_A.Z_1^1) = 2 \qquad card(\mathcal{M}_B.Z_1^1) = 2$$
$$card(\mathcal{M}_A.Z_1^2) = 3 \qquad card(\mathcal{M}_B.Z_1^2) = 3$$

Figure 6.16: Counter-example to prove that $\cup_{\mathcal{M} \in \widehat{\mathcal{M}}} [\mathcal{M}\rangle \supset [\widehat{\mathcal{M}}\rangle$ is not true.

Notice that the composed marking

$$\widehat{\mathcal{M}_1} = [\overset{p_1}{0}, \overset{p_2}{\widehat{Z_1^1}}] || [\overset{p_1}{\widehat{Z_1^2} + \widehat{Z_1^3}}, \overset{p_2}{\widehat{Z_1^4}}, \overset{p_3}{0}]$$

with

$\widehat{Z_1^1} = \mathcal{M}_A.Z_1^2$, $\widehat{Z_1^2} = \mathcal{M}_A.Z_1^1$, $\widehat{Z_1^3} = \mathcal{M}_B.Z_1^1$ and $\widehat{Z_1^4} = \mathcal{M}_B.Z_1^2$

is reachable from the composed marking

$$\widehat{\mathcal{M}_2} = [0, \widehat{Z_1^1}] \mid [2\widehat{Z_1^2}, \widehat{Z_1^1}, 2\widehat{Z_1^3}]$$

originating from considering $\mathcal{M}_A.Z_1^1 = \mathcal{M}_B.Z_1^1$ and $\mathcal{M}_A.Z_1^2 = \mathcal{M}_B.Z_1^2$. This is because a marking reachable from $\widehat{\mathcal{M}_2}$ is a restriction of $\widehat{\mathcal{M}_1}$.

However, the symbolic marking

$$[0, Z_1^1 + Z_1^2] \mid [Z_1^1 + Z_1^3, Z_1^2 + Z_1^3]$$

in $\widehat{\mathcal{M}_1}$ with $card(Z_1^1) = card(Z_1^3) = 2$ and $card(Z_1^1) = 1$, is not reachable from any symbolic marking in $\widehat{\mathcal{M}_2}$.

The cardinality of a composed marking depends on the number of dynamic sub-classes of its lowest level marking components and on the restrictions imposed by the composed dynamic sub-classes of the composed marking. These restrictions are determined by the intersection of dynamic sub-classes of the lowest level marking components.

Property 6.4 (Cardinality of a composed marking)
Let us denote by $G_{i,q}$ the number of valid intersections between dynamic sub-classes of the lowest level marking components for a basic colour class C_i and its static sub-class $D_{i,q}$. Then the number of symbolic markings represented by $\widehat{\mathcal{M}}$, denoted $\mid \widehat{\mathcal{M}} \mid$, is given by:

$$\mid \widehat{\mathcal{M}} \mid = \left(\prod_{i=1}^{n} \prod_{q=1}^{n_i} G_{i,q} \right)$$

For ordered classes there are as many possible valid intersection as rotations of the ordered class. In the case of a static sub-class whose dynamic sub-classes of the lower level components all have cardinality equal to the cardinality of the static sub-class there is only one possible intersection.

These properties ensure that no information about the SRGs of the cWN is lost by analysing the CRG instead of each of its possible SRGs. The following propositions relate properties of the SRG with those of the CRG.

Proposition 6.3 *Strong connectivity of a SRG of $\mathcal{N}$ $\Rightarrow$ Strong connectivity of the CRG, but not vice versa.*

Proof.- From Proposition 6.2 we know that given a symbolic marking $\mathcal{M}$ in a composed marking $\widehat{\mathcal{M}}$, and a symbolic marking $\mathcal{M}' \in \widehat{\mathcal{M}'}$, such that $\mathcal{M}'$ is reachable from $\mathcal{M}$, then $\widehat{\mathcal{M}'}$ is reachable from $\widehat{\mathcal{M}}$. ■

Proposition 6.4 *Strong connectivity of the CRG* $\wedge$ $\exists\widehat{\mathcal{M}} \in$*CRG such that all its lowest level dynamic sub-classes correspond to static sub-classes* $\Rightarrow$ *there is only one SRG in the CRG and it is strongly connected.*

Proof.- If there is a composed marking for which all the dynamic sub-classes of the lowest level markings correspond to static sub-classes then, there is only one possible intersection between the dynamic sub-classes of the lowest level marking components. This intersection corresponds to the one in which the resulting dynamic sub-classes also correspond to static sub-classes. Given that the CRG is strongly connected, then from all other composed markings we can arrive at this distinguished composed marking. The firing of enabled symbolic transition instances in a symbolic marking produces a symbolic marking of the same SRG. Therefore, there is only one SRG in the CRG. Furthermore this SRG in strongly connected. ∎

In the same manner that a composed marking can represents several symbolic markings, a composed transition instance, and in consequence a composed firing may represent several symbolic instances. Using a similar notation to the one employed in [CDFH93] for the relation between the symbolic firing and the ordinary firing, we conjecture the following property.

Property 6.5 (Relation between symbolic and composed firing) *Let* $\widehat{\mathcal{M}}$ *and* $\widehat{\mathcal{M}}'$ *be two composed markings of the CRG, and let* $\mathcal{M} \in \widehat{\mathcal{M}}$ *be a symbolic marking in a SRG of the CRG. Let* $\mathcal{A}_{\mathcal{M},\widehat{\mathcal{M}}'}$ *be a set of arcs connecting* $\mathcal{M}$ *to any symbolic marking of a subset of symbolic markings of* $\widehat{\mathcal{M}}'$ *and* $\mathcal{A}_{\widehat{\mathcal{M}},\widehat{\mathcal{M}}'}$ *be the composed arc of the CRG connecting* $\widehat{\mathcal{M}}$ *to* $\widehat{\mathcal{M}}'$*.*

Then there exists a mapping ω *from* $\mathcal{A}_{\mathcal{M},\widehat{\mathcal{M}}'}$ *to* $\mathcal{A}_{\widehat{\mathcal{M}},\widehat{\mathcal{M}}'}$ *such that:*

- *if the label of an arc* $a \in \mathcal{A}_{\mathcal{M},\widehat{\mathcal{M}}'}$ *is* $[t, \lambda, \mu]$ *then the label of* $\omega(a)$ *is* $[t, \lambda', \mu']$ *with* $\forall C_i$*,* $\forall X \in param_i$ *of t,* $\lambda'_i(X) \supseteq \lambda(X)$ *and* $\mu'_i(X) = \mu'(X)$*.*

- *if the label of a composed arc* $a \in \mathcal{A}_{\widehat{\mathcal{M}},\widehat{\mathcal{M}}'}$ *is* $[t, \lambda', \mu']$ *then the cardinality of the reciprocal image of a mapping denoted* $\mid \omega^{-1}(a) \mid$ *is:*

$$\prod_{i=1}^{h} \prod_{q=1}^{n_i} G_{i,q} / \lambda_i$$

the product of the number of all possible intersection of the lowest level dynamic sub-classes for each dynamic sub-class of each basic colour, restricted by the intersections defined by the function λ *for the parameters of the transition.*

It is also possible to deduce more specific behavioural properties of the system using its CRG instead of the SRG, such as:

Proposition 6.5 (Deadlock)
Deadlock in a CRG implies deadlock in all SRGs of the CRG.

Proof.- A *cWN* system contains a deadlock if it can arrive at a state where no composed transition instances can be fired. If a composed marking arrives at a absorbing state it is because there is no possible intersection of lowest level dynamic sub-classes which enables a composed transition instance. This means that for all symbolic markings of the composed marking there are no enabled symbolic transition instance. ■

Property 6.6 (Deadlock free)
A deadlock free CRG implies that there is at least one deadlock free SRG in the SRGs of the CRG.

Proof.- This comes as a direct consequence from the previous property. ■

Definition 6.15 (Live transition in a CRG) *A transition t is said to be* live *in a CRG if and only if for every composed marking $\widehat{\mathcal{M}}$ reachable from the initial composed marking there exists a composed marking $\widehat{\mathcal{M}}'$ reachable from $\widehat{\mathcal{M}}$ such that there is a composed instance of t enabled in $\widehat{\mathcal{M}}'$.*

Proposition 6.6 (Liveness) *A live transition t in a CRG implies that t is live in at least one SRG of the CRG.*

Proof.- A composed marking $\widehat{\mathcal{M}}'$ is reachable from another $\widehat{\mathcal{M}}$ if and only if there exists a sequence of composed transition firings σ enabled in $\widehat{\mathcal{M}}$ such that $\widehat{\mathcal{M}}[\sigma\rangle\widehat{\mathcal{M}}'$. A sequence of composed firings can contain several (parallel) symbolic firing sequences, from different SRGs. All of these symbolic firing sequences start from symbolic markings in $\widehat{\mathcal{M}}$ but some of them can end in symbolic markings of intermediate composed markings produced in the sequence, but there must be at least one symbolic firing sequence from a symbolic marking in $\widehat{\mathcal{M}}$ to a symbolic marking in $\widehat{\mathcal{M}}'$.

Consider a composed marking $\widehat{\mathcal{M}}$ reachable from the initial composed marking, such that t is enabled in $\widehat{\mathcal{M}}$. For this condition to hold there must exist at least one symbolic marking $\mathcal{M} \in \widehat{\mathcal{M}}$ where t is enabled. From the discussion above, we know that there must be a symbolic firing sequence from the initial marking, of the SRG that contains $\mathcal{M}$, to $\mathcal{M}$. Consider the composed firing of a enabled composed transition instance of t in $\widehat{\mathcal{M}}$ to produce a composed marking $\widehat{\mathcal{M}}'$, such that the composed instance of t contains a symbolic instance of t enabled in $\mathcal{M}$. Given that t is live in the CRG there must exist a composed marking reachable from $\widehat{\mathcal{M}}'$ where t is enabled. The generalisation of this reasoning allows us to conclude that there is at least one SRG where t is live.

■

6.6 Conclusions

In this chapter a method has been proposed to generate the state space of a *cWN* by using, whenever possible, state space information of its sub-components. We have introduced the concept of a *Composed Reachability Graph* (CRG), based on the definition of *composed markings*, *composed dynamic sub-classes* and *composed firing*.

We have presented the description of the algorithm for the construction of the CRG. To search for the set of the enabled composed transition instances in a composed marking the method proposed performs a recursive search over the sub-components of the *cWN*. Once the set of enabled transition instance is known, the composed markings resulting from the firing of the composed transition instances are obtained using, whenever possible, the information about CRGs and/or SRGs of the sub-components; otherwise a composed firing is performed.

Given the definition of the CRG and the steps for its construction, several properties relating the CRG to the SRG of a *cWN* have been proposed and proved. These properties allow us to use the analysis of the CRG, instead of the SRG, to obtain behavioural information about the *cWN*. A *cWN* can have several SRGs, depending on the sets of dynamic sub-classes of the markings of its sub-components. When a CRG represents a single SRG, a composed marking can represent a single symbolic marking or a group of symbolic markings belonging to the SRG, i.e. the composed marking can be seen as the grouping of different states of the system. By using the CRG algorithm, we avoid having to compute each SRG of the *cWN* (if it is the case), but are still able to deduce their behaviour.

As written by Valmari in [Val94], a key question posed to every new effort-saving method is: how much saving does it yield in practice? The answer to this question depends very much on the nature of the relation between the components (loosely or tightly coupled) and on the composition operation applied. The ideal case is when applying the CRG algorithm to the independent parallel composition of two *cWN*s. On the contrary, when composing using the competing parallel composition to form a tightly coupled system, the need to apply the SRG algorithm at the level of the sub-components increases, considerably decreasing the improvements made by generating the CRG instead of the SRG. This is especially true if there is only one SRG in the CRG. However, in general, the composed marking will always correspond to a group of symbolic markings. The exception is the case of all dynamic sub-classes being static sub-classes.

The difficulties of this method are mainly related to the definition of WNs. However, if generalised, the compositional approach proposed here could be adapted for use with less complicated PN formalisms. In these cases, composed markings will correspond to ordinary markings, since there will only be one possible intersection.

Some of the related work in the area of compositional construction of the state space of a PN system, such as that reported in [Val90, NM94], concentrate on the construction of the state space by viewing the system in a modular way (a system consisting of a set of modules or components), computing the state space for each module and then appropriately combining these spaces to obtain the final state space of the system. Christensen and Petrucci [CP95] take a different approach. By obtaining what is termed a *modular state space*, they preserve all information about the original state space, but do not generate the full reachability graph and show how properties of the overall system can be checked directly on the modular state space. Christensen and Petrucci point out that one of the disadvantages of the method proposed by [NM94], where the state space of the modules is generated first, is that it is possible to have modules with infinite state space while the overall system has a finite state space.

The approach we propose can be seen as a combination of the previous ones. The PN system (in this case a *cWN* system) is viewed as composed of other subsystems (sub-components) for which we may already have information about the state space. However, the state space of the subsystems or modules are not directly combined to build the state space of the overall system, but to construct the *cWN*'s state space using, whenever possible, the information of the sub-systems' state spaces. It does this regardless of the type of composition between the components. The method proposed here, coincides with the method proposed in [CP95], in that it preserves all information about the system, without generating a full reachability graph and checks the properties of the system at a composed or modular level. The use of WN as the basis of our compositional method, allows us to continue to use the symmetries of the PN system in order to reduce the size of the state space, but still to be able to deduce the same qualitative properties of the underlying Petri net system.

The structure of the decomposition process suggests that the implementation of this process can be done applying parallel programming techniques. The search for enabled composed transition instances in the sub-markings of a composed marking can be done independently. Communication between the processes is necessary to construct the set of enabled composed transition instances of the composed marking or to verify that the set of composed transition instances of a composed sub-marking is still enabled in the composed marking. The construction of a modular state space in [CP95] can be performed in parallel using only transitions local to modules. Nevertheless, it is necessary to keep track of the firing of synchronising transitions to synchronise the modules. In the case of cWN this problem of synchronisation is more difficult because it is also necessary to take into account any new predicates defined over the transition resulting from the synchronisation. Caselli et al. [CCM95] study different methods for the state space exploration for GSPN model. An interesting work would be to study how these methods can be used for the construction of the CRG, exploiting the compositional structure of the system.

Chapter 7

Case Study

7.1 Introduction

The objective of this chapter is to consolidate the methods proposed in this dissertation. We will combine the compositional methods proposed for the construction of *cWN* systems, for the construction of P- and T- semiflows of a *cWN* model and for the compositional construction of the state space of *cWN* systems, over a common example.

Section 7.2 describes the system to be modelled. Starting with the modelling process, in Section 7.3 we identify the basic colours classes of the *cWN* model. From the systems description, in Section 7.4, the system is divided into subsystems. For each these subsystems we deduce its set of basic functions which will be associated with the *bWN*s of the *cWN* system modelling the subsystem. In Section 7.5 we obtain the symbolic P-semiflows of the *bWN* of each subsystem and in Section 7.6 we obtain the SRG of the *bWN*s. The SRG presented are only those that have at least one reachable symbolic marking different than the initial one. From the *bWN* of each subsystem in Section 7.7 it is shown how the *bWN*s are composed to obtain the *cWN* systems modelling each subsystem. Only the construction process of one of these subsystems is given in detail. This section also covers the construction of the generative family of symbolic P-semiflows of each of the subsystems. The symbolic T-semiflows of the subsystems are also presented. However, in must be noted that these have not been obtained in a compositional manner. The construction of the CRG is shown only for one of the subsystems of the system. Given the *cWN*s modelling the subsystems and their structural and state space information, in Section 7.8, these *cWN*s are composed to form the *cWN* system modelling the overall system. The symbolic P- and T-semiflows are presented, however, for space and tractability reasons the CRG of the overall system is not. Nevertheless, we discuss how the characteristics of the

CRGs of the cWNs modelling the subsystems can be re-used for the construction of the state space of the cWN modelling the FMS system. In Section 7.9 we present an alternative cWN system to model the FMS system, considering a different set of basic colour classes.

7.2 Description of the system: A Flexible Manufacturing System (FMS)

The model that we will study is a modification of the Flexible Manufacturing System (FMS) proposed in [CT91]. The FMS has three types of machine, namely, Mc_1, Mc_2 and Mc_3. Machines of type Mc_1 process rough parts of type $Part_1$, while machines of type Mc_2 process rough parts of type $Part_2$ and $Part_3$. Finished parts of the different types can be shipped. Finished parts of type $Part_1$ and $Part_2$ can also be assembled together (one part of each type), to form a new part of type $Part_{12}$ by using machines of type Mc_3. The probability for a part of type $Part_1$ or $Part_2$ to be shipped is the same as the probability for it to be assembled. Finished parts of type $Part_{12}$ are then shipped. To maintain a constant inventory when parts are shipped the same number of rough parts used to form the finished part re-enter the system.

7.3 Determining the basic colour classes

From the system's description it is possible to identify two basic types of objects, namely, machines and parts. Parts can be rough or finished. Finished parts can be identified by the rough parts used to build them. Our basic colour classes will then be machines and (rough) parts. Each of these basic colour classes can be divided into three static sub-classes corresponding to the different types of machines and rough parts, respectively:

$$\begin{aligned} \text{Machines} =& \{Mc_1, Mc_2, Mc_3\} \\ \text{Parts} =& \{Part_1, Part_2, Part_3\} \end{aligned}$$

In the cWN models the colour classes will be identified by C_1 for Machines and C_2 for Parts; and the static sub-classes as $D_{1,1}, D_{1,2}$ and $D_{1,3}$ for the sub-classes Mc_1, Mc_2 and Mc_3, and $D_{2,1}, D_{2,2}$ and $D_{2,3}$ for the sub-classes $Part_1, Part_2$ and $Part_3$.

7.4 Basic functions of the FMS

The FMS described can produce four types of final products: finished parts of types $Part_1$, $Part_2$ or $Part_3$, or parts of type $Part_{12}$ (by combining finished parts of types $Part_1$ and $Part_2$). We will consider the production process for each type of finished part as a subsystem of the system. For each subsystem we identify its basic functions. These basic functions will be modelled by *bWN*s.

7.4.1 Production of finished parts of type $Part_1$ (subsystem 1)

To produce a finished part of type $Part_1$ we need a rough part of the same type and a machine of type Mc_1. When produced, the finished part can be, with equal probability, either shipped or used to form a part of type $Part_{12}$. A similar process is applied over parts of type $Part_2$, using machines of type Mc_2. To model both subsystems with the same *cWN* it would be necessary to be able to select the type of machine to be used according to the part to be produced. This can be done by the use of predicates over the corresponding transitions, such as: the part selected is of type 1 and the machine is of type 1, or the part is of type 2 or 3 and the machine is of type 2. However, in order to maintain the model simple and understandable we have opted for representing them as different subsystems.

The predicates that can be defined over transitions do not include a conditional case. Therefore, we will represent these two subsystems by different *cWN* models.

Basic Functions

Start processing: This function corresponds to the first stage of processing a part. It takes a part of type $Part_1$ and a machine of type Mc_1, to process the rough part. Its output consists of partially processed parts of type $Part_1$ and for each one the identification of the machine processing it. The *bWN* model of this function is presented in Figure 7.1.(A_1). The transition representing this function is termed $start\_proc_1$.

Input: rough parts of type $Part_1$ and machines of type Mc_1

Output: partially processed parts of type $Part_1$, with the identification of the machine (of type Mc_1) processing them (one machine per part).

End processing: This function finalises the processing of a part of type $Part_1$. After obtaining a finished part the corresponding machine is freed and returned to the pool of free machines. The finished part is ready to be either shipped or used to form a part of type $Part_{12}$. The bWN model of this function is presented in Figure 7.1.(B_1). The transition representing this function is termed end_proc_1.

Input: partially processed parts of type $Part_1$, with the identification of the machine (of type Mc_1) processing them.

Output: free machines of type Mc_1 and finished parts of type $Part_1$

Ship finished part: This function ships parts of type $Part_1$. As said in the system description, to maintain a constant inventory, it is assumed that when a part is shipped the same number of rough parts used to produce it re-enter the system. The bWN model of this function is presented in Figure 7.1.(C_1). The transition representing this function is termed $ship_1$.

Input: finished parts of type $Part_1$ to ship

Output: rough parts of type $Part_1$

Send part for further processing: When a finished part of type $Part_1$ is selected to be used to produce a part of type $Part_{12}$, this function takes a finished part of type $Part_1$ and prepares it to hand it over to the subsystem that produces parts of type $Part_{12}$. The bWN model of this function is presented in Figure 7.1.(D_1). The transition representing this function is termed $to\_combine_1$.

Input: finished parts of type $Part_1$ to send for further processing

Output: parts of type $Part_1$ ready for further processing

Information	Place
Rough Parts	RP
Machines	MC
Parts of type $Part_1$ being processed by machines of type Mc_1	in_proc1
Finished parts type $Part_1$	FP1
Parts of type $Part_1$ to ship	SP1
Parts of type $Part_1$ for further processing	PFP1
Parts of type $Part_1$ to combine	CP1

Table 7.1: Correspondence between the names given to places in the bWNs of subsystem 1 and the inputs and outputs of the basic functions.

Table 7.1 summarises the correspondence between place and transition names and what they represent.

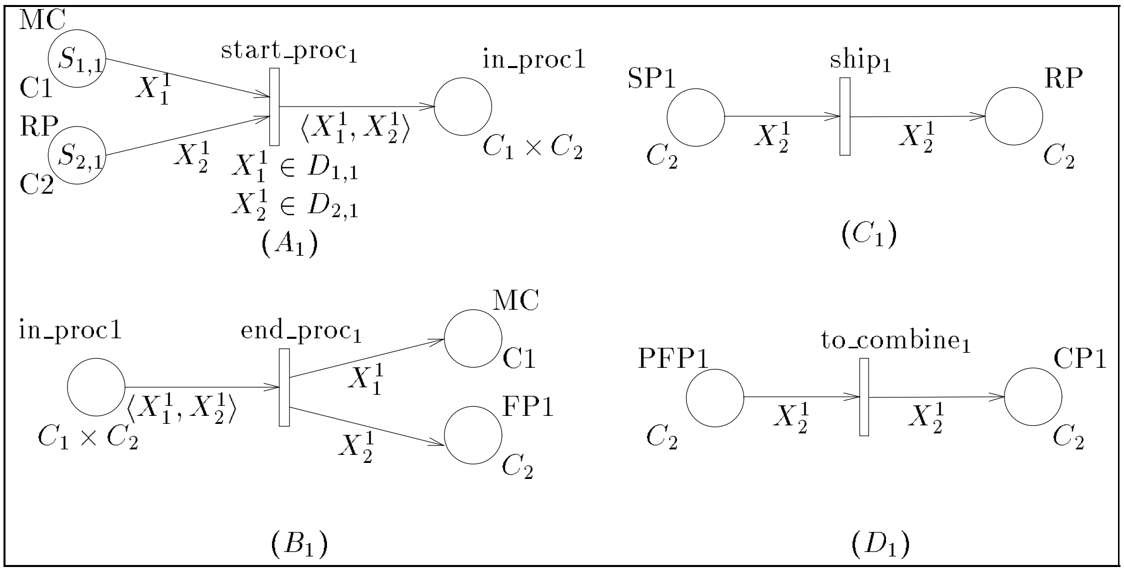


Figure 7.1: *bWN*s of the subsystem 1.

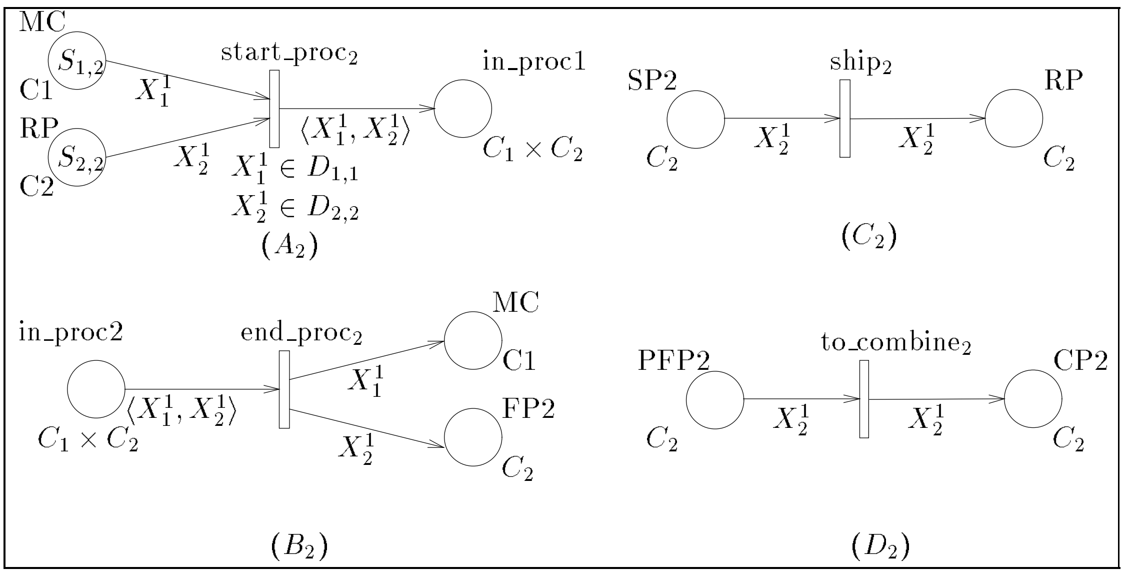


Figure 7.2: *bWN* of subsystem 2.

7.4.2 Production of finished parts of type $Part_2$ (subsystem 2)

The basic functions of the subsystem that process parts of type $Part_2$ (subsystem 2) are similar to those of subsystem 1, differing only in the fact that the type of machine used to process parts of type $Part_2$ is Mc_2. Figure 7.2 shows the *bWN*s corresponding to the basic functions of subsystem 2.

7.4.3 Production of finished parts of type $Part_3$ (subsystem 3)

Subsystem 3 differs from the first two in that the finished parts (of type $Part_3$) produced by the subsystem can only be shipped. Therefore, its set of basic functions will be similar to the functions: Start processing, End processing and Ship finished part, of the previous two subsystems. Finished parts of type $Part_3$ are produced by machines of type Mc_2. Figure 7.3 shows the *bWN*s corresponding to the basic functions of the subsystem 3.

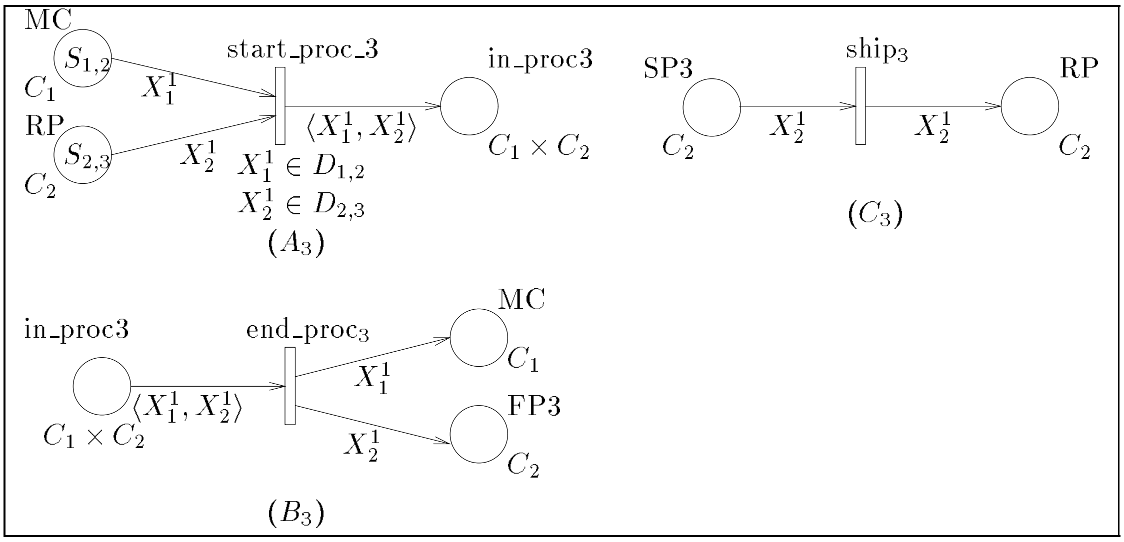


Figure 7.3: *bWN*s of subsystem 3.

7.4.4 Production of parts of type $Part_{12}$ (subsystem 4)

Although the basic functions of this subsystem are similar to the basic functions of the other three subsystems, this subsystem operates over finished parts and not over rough parts like the other subsystems. Parts of type $Parts_{12}$ are produced from finished parts of type $Part_1$ and $Part_2$ using machines of type Mc_3. Once the $Parts_{12}$ parts are ready they can be shipped.

Basic Functions

Start processing: This function takes a finished part of type $Part_1$, a finished part of type $Part_2$ and, using a machine of type Mc_3, begin the process of producing a part of type $Part_{12}$. The *bWN* model of this function is presented in Figure 7.4.(A_4). The transition representing this function is termed *start_proc*$_{12}$.

Information	Place
Rough Parts	RP
Machines	MC
Parts to combine	Pcom
Parts of type $Part_{12}$ being processed by machines of type Mc_3	in_proc12
Finished parts type $Part_{12}$	FP12
Parts of type $Part_{12}$ to ship	SP12

Table 7.2: Correspondence between names given to places in the *bWN*s of subsystem 4 and the inputs and outputs of the basic functions.

Input: finished parts of type $Part_1$, finished parts of type $Part_2$ and machines of type Mc_3.

Output: partially processed parts of type $Part_{12}$ and the identification of the machines processing them (one machine for each part).

End processing: This function finalises the processing of a part of type $Part_{12}$. After obtaining a finished part the machine used is freed and returned to the pool of free machines. The finished part is then ready for shipping. The *bWN* model of this function is presented in Figure 7.4.(B_4). The transition representing this function is termed end_proc_{12}.

Input: partially processed parts of type $Part_{12}$ and the identification of the machine processing them.

Output: free machines of type Mc_3 and a part of type $Part_{12}$ ready for shipping.

Ship finished part: This function ships finished parts of type $Part_{12}$. To maintain a constant inventory of rough parts, after the part of type $Part_{12}$ is shipped, a rough part of type $Part_1$ and another of type $Part_2$ re-enter the system. The *bWN* model of this function is presented in Figure 7.4.(C_4). The transition representing this function is termed $ship_{12}$.

Input: parts of type $Part_{12}$ ready for shipping.

Output: rough parts of type $Part_1$ and $Part_2$ (replacing the parts used in the production of parts of type $Part_{12}$).

Table 7.2 summarises the correspondence between place and transition names and what they represent.

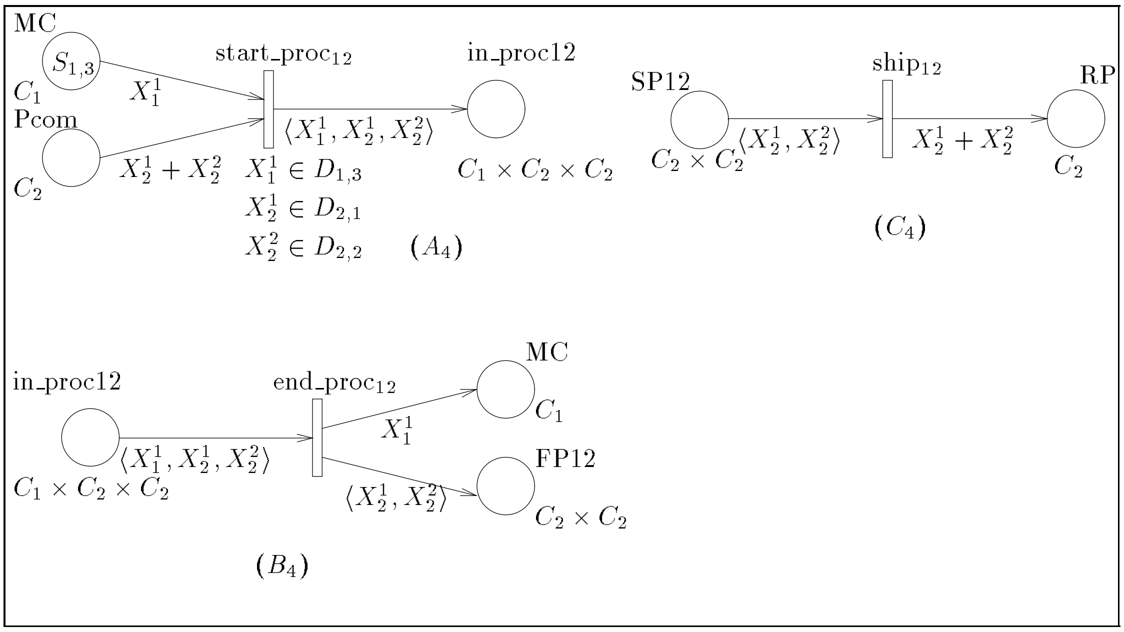


Figure 7.4: *bWN*s of subsystem 4.

7.5 Symbolic P-semiflows of the *bWN*s

Using the Algorithm 1 (presented on page 98), we calculate the minimal symbolic P-semiflows of the basic components. We will only present the minimal symbolic P-semiflows of the *bWN* of the subsystems 1 and 4, considering them to be a good representation of the rest of the subsystems of the model. The minimal P-semiflows shown are those covering more than one place in the *bWN*. Symbolic P-semiflows that cover only one place, assigning non-zero values to the colours of the static sub-classes that are not considered by the arc functions, are not shown. However, they are assumed to form part of the generative family of symbolic P-semiflows of the *bWN*s.

Subsystem 1

Minimal Symbolic P-semiflows of **A1**		
RP	MC	in_proc1
$\langle S_{2,1} \rangle$	0	$\langle S_{1,1}, S_{2,1} \rangle$
0	$\langle S_{1,1} \rangle$	$\langle S_{1,1}, S_{2,1} \rangle$

Minimal Symbolic P-semiflows of **B1**		
in_proc1	MC	FP1
$\langle S_{1,1}, S_{2,1} \rangle$	0	$\langle S_{2,1} \rangle$
$\langle S_{1,1}, S_{2,1} \rangle$	$\langle S_{1,1} \rangle$	0

Minimal Symbolic P-semiflows of **C1**	
SP1	RP
$\langle S_{2,1} \rangle$	$\langle S_{2,1} \rangle$

Minimal Symbolic P-semiflows of **D1**	
Pcom1	CP1
$\langle S_{2,1} \rangle$	$\langle S_{2,1} \rangle$

Subsystem 4

Minimal Symbolic P-semiflows of **A4**		
Pcom	MC	in_proc12
$\langle S_{2,1}\rangle$	0	$\langle S_{1,3}, S_{2,1}, S_{2,2}\rangle$
$\langle S_{2,2}\rangle$	0	$\langle S_{1,3}, S_{2,1}, S_{2,2}\rangle$
0	$\langle S_{1,3}\rangle$	$\langle S_{1,3}, S_{2,1}, S_{2,2}\rangle$

Minimal Symbolic P-semiflows of **B4**		
in_proc12	MC	FP12
$\langle S_{1,3}, S_{2,1}, S_{2,2}\rangle$	$\langle S_{1,3}\rangle$	0
$\langle S_{1,3}, S_{2,1}, S_{2,2}\rangle$	0	$\langle S_{2,1}, S_{2,2}\rangle$

Minimal Symbolic P-semiflows of **C4**	
SP12	RP
$\langle S_{2,1}, S_{2,2}\rangle$	$S_{2,1}$
$\langle S_{2,1}, S_{2,2}\rangle$	$S_{2,2}$

Recall that bWNs only have the trivial T-semiflow $(\vec{0})$.

7.6 SRGs of the bWNs

The symbolic reachability graph (SRG) with more than one symbolic marking can only be obtained for the A bWNs of the of the subsystems 1, 2 and 3. The symbolic markings are represented as triples, where the first element corresponds to the place MC, the second to the place RP and the third to the in_proc place. We will assume that the FMS system modelled has 1 machine of type Mc_1, 2 machines of type Mc_2, 2 machines of type Mc_3, 3 rough parts of type $Part_2$ and 2 of each other type of rough part.

SRG_{A_1}

$$[Z_1^1, Z_2^1, \langle 0, 0\rangle] \qquad card(Z_1^1) = 1, \quad card(Z_2^1) = 2$$

$$\downarrow$$

$$[0, Z_2^1, \langle Z_1^1, Z_2^2\rangle] \qquad card(Z_1^1) = 1, \quad card(Z_2^1) = 1, \quad card(Z_2^2) = 1$$

SRG_{A2}

$[Z_1^2, Z_2^2, \langle 0, 0\rangle]$

$\mathrm{card}(Z_1^2) = 2$
$\mathrm{card}(Z_2^2) = 3$

$$\downarrow$$

$[Z_1^2, Z_2^2, \langle Z_1^3, Z_2^3\rangle]$

$\mathrm{card}(Z_1^2) = 1$
$\mathrm{card}(Z_1^3) = 1$
$\mathrm{card}(Z_2^2) = 2$
$\mathrm{card}(Z_2^3) = 1$

$$\downarrow$$

$[0, Z_2^2, \langle Z_1^2, Z_2^3\rangle + \langle Z_1^3, Z_2^4\rangle]$

$\mathrm{card}(Z_1^2) = 1$
$\mathrm{card}(Z_1^3) = 1$ $\mathrm{card}(Z_2^2) = 1$
$\mathrm{card}(Z_2^3) = 1$ $\mathrm{card}(Z_2^4) = 1$

SRG_{A3}

$[Z_1^3, Z_2^3, \langle 0, 0\rangle]$

$\mathrm{card}(Z_1^3) = 2$
$\mathrm{card}(Z_2^3) = 2$

$$\downarrow$$

$[Z_1^3, Z_2^3, \langle Z_1^4, Z_2^4\rangle]$

$\mathrm{card}(Z_1^3) = 1$
$\mathrm{card}(Z_2^3) = 2$
$\mathrm{card}(Z_2^3) = 1$
$\mathrm{card}(Z_2^4) = 1$

$$\downarrow$$

$[0, 0, \langle Z_1^3, Z_2^3\rangle + \langle Z_1^4, Z_2^4\rangle]$

$\mathrm{card}(Z_1^3) = 1$
$\mathrm{card}(Z_1^4) = 1$
$\mathrm{card}(Z_2^3) = 1$
$\mathrm{card}(Z_2^4) = 1$

Notice that in the SRG of component A_2, although not present in the symbolic markings, the dynamic sub-classes Z_1^1 with $d(Z_1^1) = D_{1,1}$ and Z_2^1 with $d(Z_2^1) = D_{2,1}$ are considered to exist with cardinality 0; and similarly for Z_1^1, Z_1^2, Z_2^1 and Z_2^2 in SRG_{A_3}. The labels of the symbolic transition firings have been omitted, since for every symbolic marking there is only one enabled symbolic instance of the transition of the corresponding bWN.

7.7 Compositional construction of the subsystems

All of the subsystems described are more or less built in the same manner, mainly as sequential composition of its functions. Therefore, we will only give detailed information about the construction of one of the subsystems. We have chosen subsystem 1 because it is one of the two subsystems (the other one being subsystem 2) that uses more composition operations.

Construction steps

1. Apply the choice operation over the *bWN*s C_1 and D_1, with $Choice(C_1)$ = SP1 and $Choice(D_1)$ = PFP1. The resulting component will be termed choi_$Part_1$(see Figure 7.5), and will have interface:

   - ES = { pc }
   - FS = { CP1, RP }
   - ST = { ship$_1$, to_combine$_1$ }

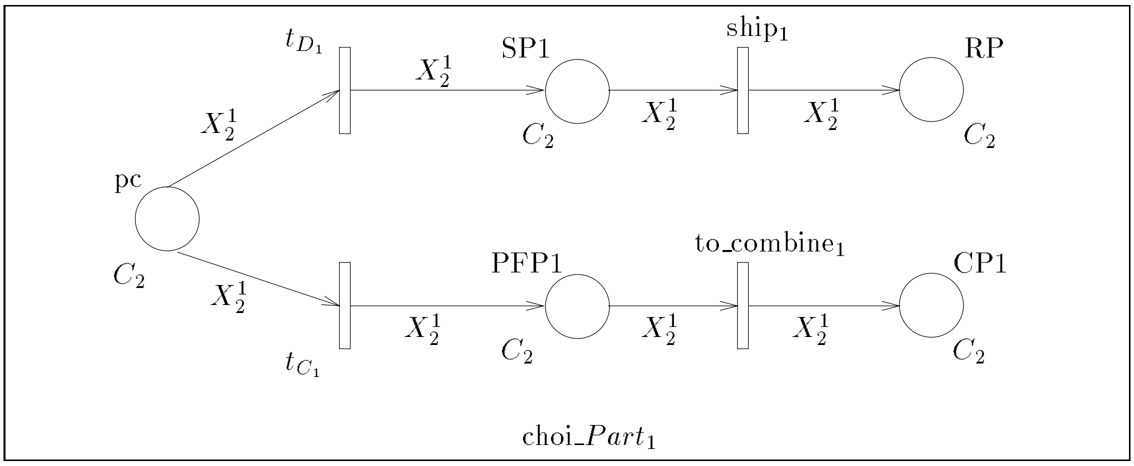


Figure 7.5: Component choi_$Part_1$, obtained from a choice composition of the *bWN*s C_1 and D_1 of subsystem 1.

2. Sequentially compose the *bWN*s A_1 and B_1 to form the *cWN* Produce_$Part_1$ (see Figure 7.6) with $FS'_{A_1} = \{A_1.\text{in\_proc}_1\}$, $ES'_{B_1} = \{B_1.\text{in\_proc}_1\}$ and $\Gamma(A_1.\text{in\_proc}_1) = B_1.\text{in\_proc}_1$. The interface of Produce_$Part_1$ will be:

   - ES = { RP, A_1.MC }
   - FS = { FP1, B_1.MC }

- $ST = \{$ start_proc$_1$, end_proc$_1$ $\}$

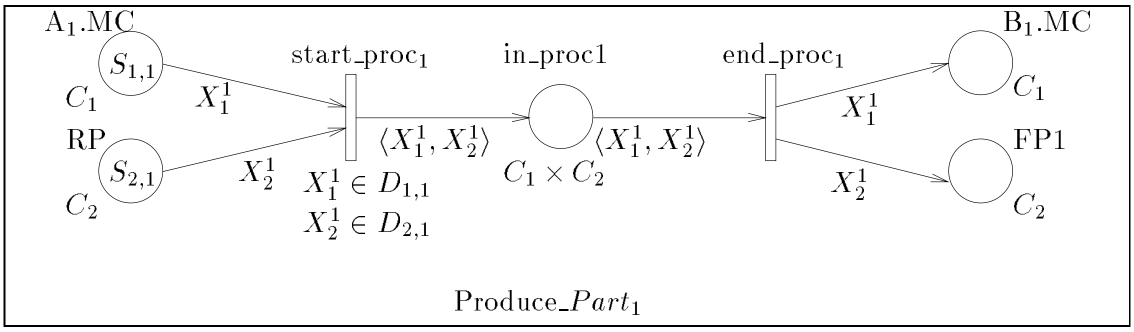


Figure 7.6: Component Produce_$Part_1$; obtained from the sequential composition of the components A_1 and B_1 of subsystem 1.

3. Apply the closing operation over component Produce_$Part_1$, with $\Theta(B1.MC) = A1.MC$. The resulting component will be termed Produce.b_$Part_1$(see Figure 7.7), and will have the following interface:

    - $ES = \{$ RP, MC $\}$
    - $FS = \{$ FP1 $\}$
    - $ST = \{$ start_proc$_1$, end_proc$_1$ $\}$

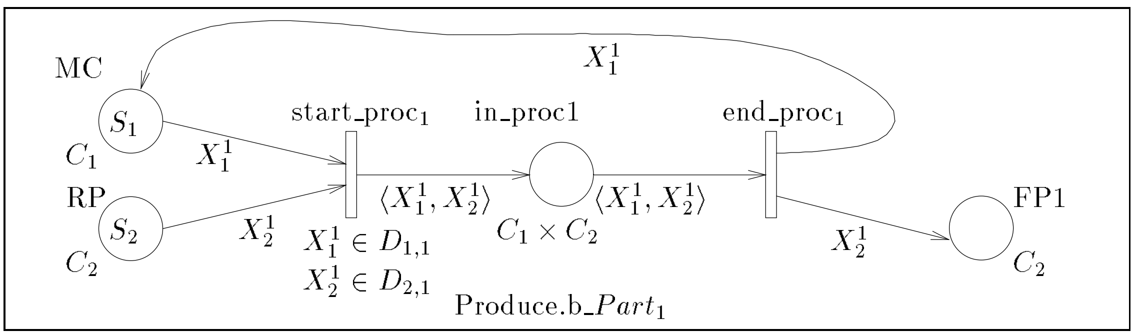


Figure 7.7: Component Produce.b_$Part_1$; obtained by closing the component Produce_$Part_1$.

4. Sequentially compose the *cWN*s Produce.b_$Part_1$ and choi_$Part_1$, with $\Gamma(FP1) = pc$. The *cWN* resulting from the operation will be termed process_$Part_1$(see Figure 7.8(a)), and will have interface:

    - $ES = \{$ Produce.b_$Part_1$.RP, MC $\}$
    - $FS = \{$ CP1, choi_$Part_1$.RP $\}$
    - $ST = \{$ start_proc$_1$, end_proc$_1$, ship$_1$, to_combine$_1$ $\}$

5. Finally we reflect the re-entering of rough parts when finished parts are shipped. By closing subsystem1 with

$$\Theta(\text{Produce.b\_}Part_1.\text{RP}) = \text{choi\_}Part_1.\text{RP}$$

we obtain the *cWN* subsystem1 (see Figure 7.8(b)) which models the subsystem that produces finished places of type $Part_1$. The interface of subsystem 1 will then be:

   - ES = { RP, MC }
   - FS = { CP1 }
   - ST = { start_proc$_1$, end_proc$_1$, ship$_1$, to_combine$_1$ }

The interfaces of the subsystems 2, 3 and 4 are defined as follows:

- Interface subsystem 2:
  - ES = { RP, MC }
  - FS = { CP2 }
  - ST = { start_proc$_2$, end_proc$_2$, ship$_2$, to_combine$_2$ }
- Interface subsystem 3:
  - ES = { RP, MC }
  - $FS = \emptyset$
  - ST = { start_proc$_3$, end_proc$_3$, ship$_3$ }
- Interface subsystem 4:
  - ES = { Pcom1, Pcom2}
  - FS = { RP }
  - ST = { start_proc$_{12}$, end_proc$_{12}$, ship$_{12}$ }

7.7.1 Minimal symbolic P-semiflows of the subsystems

The symbolic P-semiflows of the subsystems can be obtained from the symbolic P-semiflows of the *bWN*s using the methods defined in Chapter 5. Let us show how we obtain the symbolic P-semiflows of subsystem 1. Notice that the symbolic P-semiflows obtained are also static P-semiflows. This occurs because there is at most one occurrence of each basic colour class in the colour domains of the places.

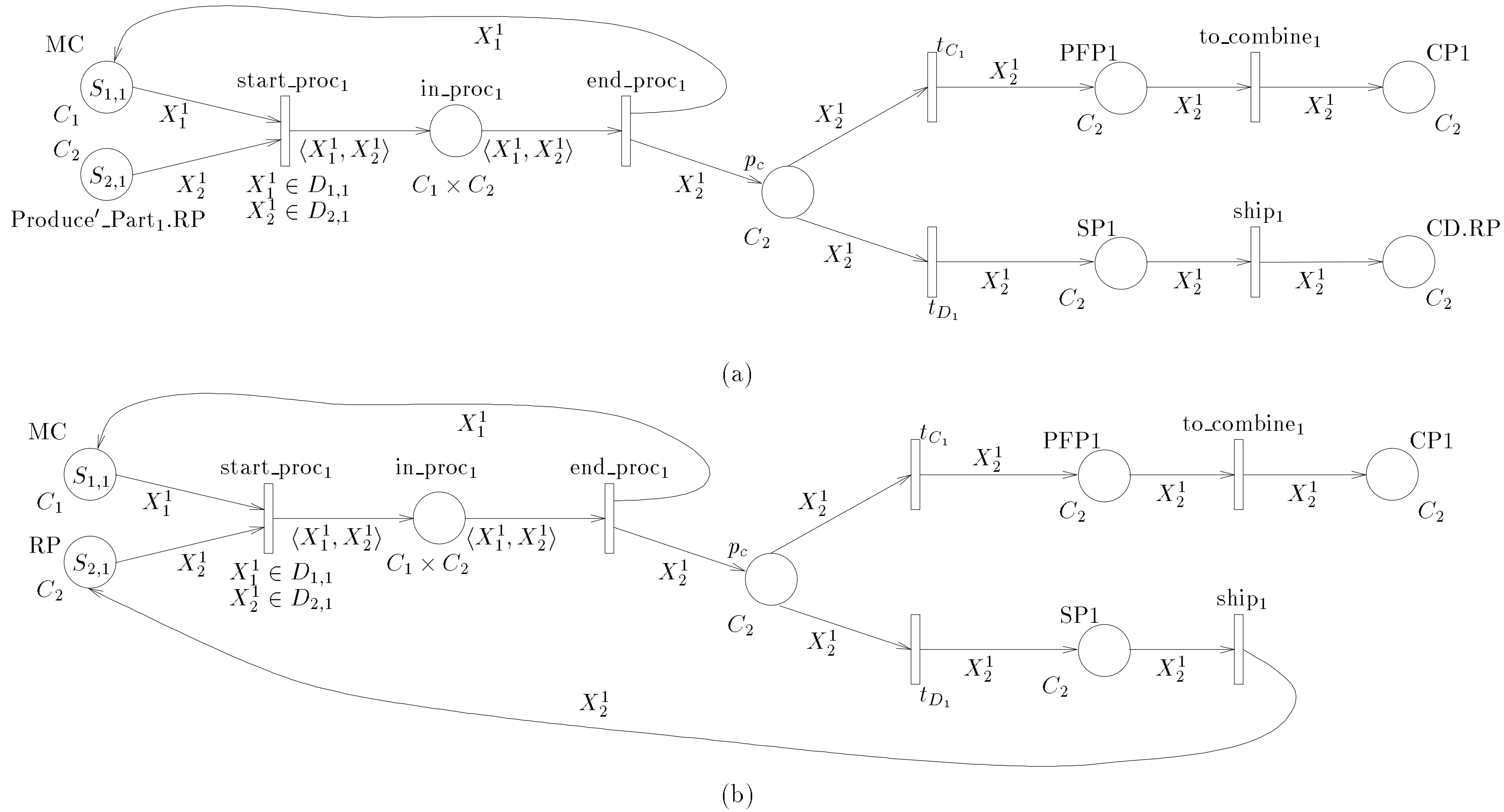


Figure 7.8: Component process_$Part_1$ (a) and cWN representing the subsystem 1 (b). The latter is produced by closing the cWN process_$Part_1$.

The same occurs for the colour domains of the transitions. As for the minimal P-semiflows of the *bWN*s we will omit the symbolic P-semiflows that cover only one place.

For the *cWN* choi_$Part_1$ (obtained from the choice composition of the *bWN*s C_1 and D_1) we obtain the following minimal symbolic P-semiflow:

pc	SP1	RP	Pcom1	CP1
$S_{2,1}$	$S_{2,1}$	$S_{2,1}$	$S_{2,1}$	$S_{2,1}$

The minimal symbolic P-semiflows of the *cWN* Produce_$Part_1$ (obtained from the sequential composition of the *bWN*s A_1 and B_1) will be:

RP	A_1.MC	in_proc1	B_1.MC	FP1
$S_{2,1}$	0	$\langle S_{1,1}, S_{2,1}\rangle$	0	$S_{2,1}$
$S_{2,1}$	0	$\langle S_{1,1}, S_{2,1}\rangle$	$S_{1,1}$	0
0	$S_{1,1}$	$\langle S_{1,1}, S_{2,1}\rangle$	$S_{1,1}$	0
0	$S_{1,1}$	$\langle S_{1,1}, S_{2,1}\rangle$	0	$S_{2,1}$

The first two are obtained by combining the first symbolic P-semiflow of A_1 with the minimal P-semiflows of B_1. The second two are obtained from the combination of the second symbolic P-semiflow of A_1 with the minimal symbolic P-semiflows of B_1. Notice that only the first and third (symbolic) P-semiflows assign the same value to A_1.MC and B_1.MC. If we add the second and fourth P-semiflows we obtain a P-semiflow that assigns the same values to A_1.MC and B_1.MC. However, when fusing these two places the resulting P-semiflow will contain the support of the collapsed P-semiflows of the other two. Therefore, the minimal symbolic P-semiflows of the *cWN* Produce.b_$Part_1$ (obtained from the closing of Produce_$Part_1$) will be:

RP	MC	in_proc1	FP1
$S_{2,1}$	0	$\langle S_{1,1}, S_{2,1}\rangle$	$S_{2,1}$
0	$S_{1,1}$	$\langle S_{1,1}, S_{2,1}\rangle$	0

The *cWN* process_$Part_1$ obtained from the sequential composition of the *cWN*s Produce.b_$Part_1$ and choi_$Part_1$ will have the following minimal symbolic P-semiflows:

Produce.b_$Part_1$.RP	MC	in_proc1	pc	SP1	Pcom1	CP1	choi_$Part_1$.RP
$S_{2,1}$	0	$\langle S_{1,2}, S_{2,1}\rangle$	$S_{2,1}$	$S_{2,1}$	$S_{2,1}$	$S_{2,1}$	$S_{2,1}$
0	$S_{1,1}$	$\langle S_{1,2}, S_{2,1}\rangle$	0	0	0	0	0

The first symbolic P-semiflow is obtained from the combination of the first symbolic P-semiflows of Produce.b_$Part_1$ and choi_$Part_1$, and the second from the extension of the second symbolic P-semiflow of Produce.b_$Part_1$.

Finally the minimal symbolic P-semiflows of the subsystem 1, obtained by applying the closing operation over the RP places of process_$Part_1$ are given by:

RP	MC	in_proc1	pc	SP1	Pcom1	CP1
$S_{2,1}$	0	$\langle S_{1,2}, S_{2,1}\rangle$	$S_{2,1}$	$S_{2,1}$	$S_{2,1}$	$S_{2,1}$
0	$S_{1,1}$	$\langle S_{1,2}, S_{2,1}\rangle$	0	0	0	0

The set of minimal symbolic P-semiflows, excluding those that assign non-zero values to only one place, of the other subsystems of the FMS are given by:

Minimal Symbolic P-semiflows of subsystem 2						
RP	MC	in_proc2	pc	SP2	Pcom2	CP2
$S_{2,2}$	0	$\langle S_{1,2}, S_{2,2}\rangle$	$S_{2,2}$	$S_{2,2}$	$S_{2,2}$	$S_{2,2}$
0	$S_{1,2}$	$\langle S_{1,2}, S_{2,2}\rangle$	0	0	0	0

Minimal Symbolic P-semiflows of subsystem 3			
RP	MC	in_proc3	SP3
$S_{2,3}$	0	$\langle S_{1,2}, S_{2,3}\rangle$	$S_{2,3}$
0	$S_{1,2}$	$\langle S_{1,2}, S_{2,3}\rangle$	0

Minimal Symbolic P-semiflows of subsystem 4			
Pcom	MC	in_proc12	SP12
$S_{2,1}$	0	$\langle S_{1,3}, S_{2,1}, S_{2,2}\rangle$	$\langle S_{2,1}, S_{2,2}\rangle$
0	$S_{1,3}$	$\langle S_{1,3}, S_{2,1}, S_{2,2}\rangle$	0

7.7.2 Minimal symbolic T-semiflows of the subsystems

Recall that non-trivial T-semiflows are non-existent in bWN. In the FMS proposed the T-semiflows only appear at the level of subsystems. The following are the minimal symbolic T-semiflows (excluding the trivial) of the subsystems described. These T-semiflows are not calculated by the compositional methods proposed in this dissertation. However, from them it is possible to obtain the set of minimal symbolic T-semiflows of the FMS system applying the compositional methods described in Chapter 5. Subsystem 4 has only the trivial ($\vec{0}$) symbolic T-semiflow.

Minimal Symbolic T-semiflows of subsystem 1					
start_proc$_1$	end_proc$_1$	ship$_1$	to_combine$_1$	t$_{C_1}$	t$_{D_1}$
$\langle S_{1,1}, S_{2,1}\rangle$	$\langle S_{1,1}, S_{2,1}\rangle$	$S_{2,1}$	0	$S_{2,1}$	0

Minimal Symbolic T-semiflows of subsystem 2					
start_proc$_2$	end_proc$_2$	ship$_2$	to_combine$_2$	t$_{C_2}$	t$_{D_2}$
$\langle S_{1,2}, S_{2,2}\rangle$	$\langle S_{1,2}, S_{2,2}\rangle$	$S_{2,2}$	0	$S_{2,2}$	0

Minimal Symbolic T-semiflows of subsystem 3		
start_proc$_3$	end_proc$_3$	ship$_3$
$\langle S_{1,2}, S_{2,3}\rangle$	$\langle S_{1,2}, S_{2,3}\rangle$	$S_{2,3}$

7.7.3 CRGs of the subsystems

The composed reachability graph (CRG) of the subsystems will each have one SRG. This is because in the initial marking all dynamic sub-classes correspond to static sub-classes. Nevertheless, by applying the CRG algorithm the number of transitions whose enabling condition does not need to be checked in a composed marking can be considerably reduced. For example, consider the SRG of the *bWN* A_1. When this component is sequentially composed with the *bWN* B_1 (to form component Produce.b_$Part_1$), we can limit the search for new enabled composed transition instances to the transitions in B_1. This can be done since we know that the final places of A_1 cannot be input to any transition in A_1, and that all complements of the sub-marking of the Produce.b_$Part_1$ with respect to the component B_1 will be sub-markings of the symbolic markings in the SRG of A_1. Therefore information about the enabled condition of *start_proc*$_1$ (the transition of A_1) and possible the marking of A_1 after the firing of start_proc$_1$ (if *start_proc*$_1$ is enabled), can be obtained using information about the SRG of A_1. For space reasons and to maintain the tractability of the example presented we only present the CRG (SRG) of subsystem 1 (see Figure 7.9). The states are represented as 7-tuples with the first element representing the marking of the place MC, the second RP, the third in_proc1, the fourth p_c, the fifth PFP1, the sixth SP$_1$ and the seventh CP$_1$.

7.8 Compositional construction of the *cWN* modelling the FMS system

To build the final system from the *cWN*s representing each of the subsystems we need to apply the following composition operations:

1. Competing parallelism of the subsystem 1 and 2, to build subsystem_A with

$$\Lambda(\text{subsystem 1.RP}) = \text{subsystem 2.RP and}$$
$$\Lambda(\text{subsystem 1.MC}) = \text{subsystem 2.MC}$$

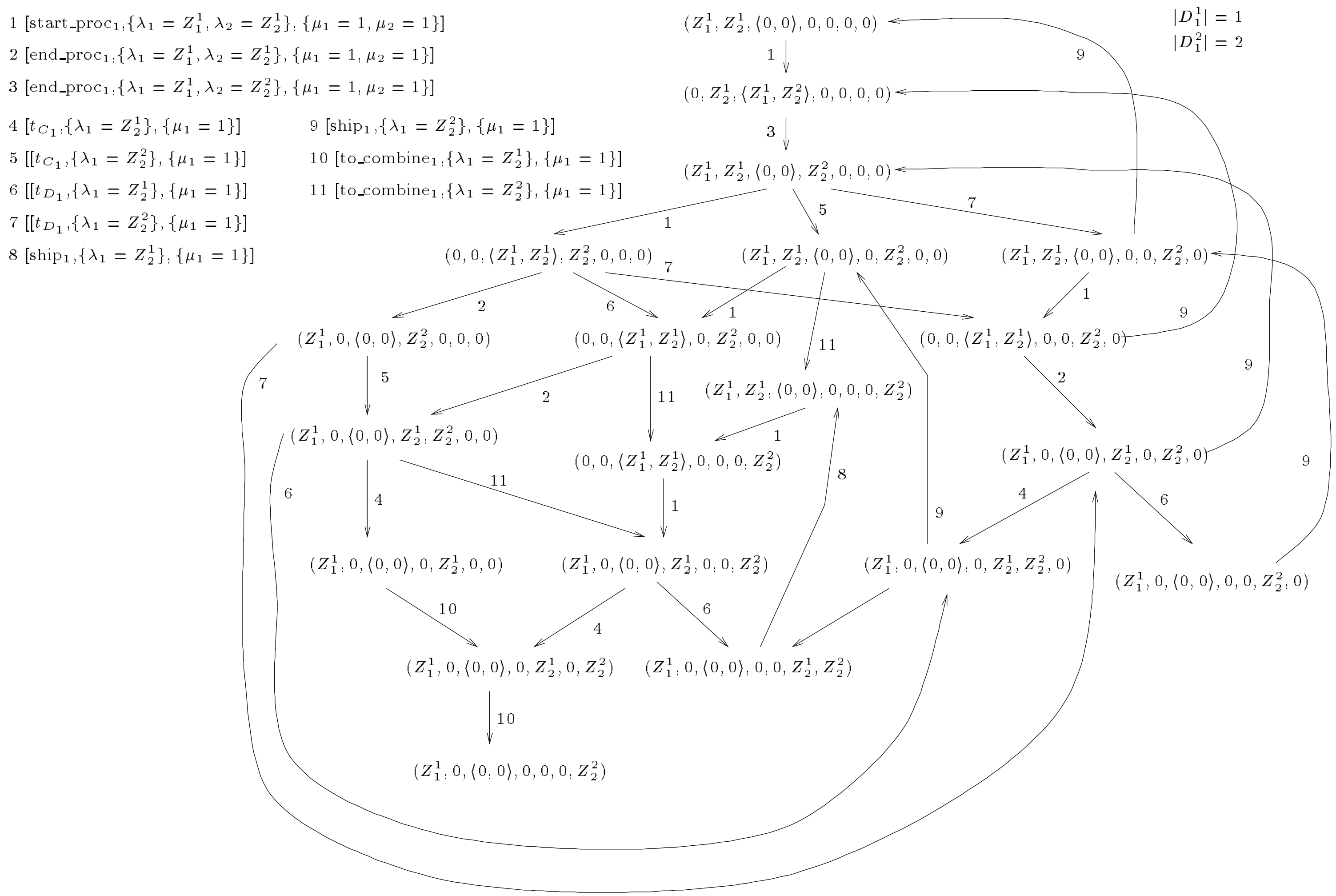


Figure 7.9: CRG (SRG) of subsystem 1.

2. Competing parallelism of the *cWN* subsystem_A with the *cWN* representing subsystem 3, to build subsystem_B. Here

$$\Lambda(\text{subsystem\_A.RP}) = \text{subsystem 3.RP and}$$
$$\Lambda(\text{subsystem\_A.MC}) = \text{subsystem 3.MC}$$

3. Sequential composition subsystem_B with subsystem 4, to build subsystem_C, with

$$\Gamma(\text{subsystem\_B.PFP}_1) = \text{subsystem 4.CP}_1 \text{ and}$$
$$\Gamma(\text{subsystem\_B.PFP}_2) = \text{subsystem 4.CP}_2$$

4. Internal competing parallelism in subsystem_C, creating subsystem_D, with

$$\Lambda(\text{subsystem\_B.MC}) = \text{subsystem 4.MC}$$

5. Closing of subsystem_D with

$$\Theta(\text{subsystem\_B.RP}) = \text{subsystem 4.RP}$$

The resulting *cWN* system corresponds to the model of the FMS system is presented in Figure 7.10.

7.8.1 Symbolic semiflows of the *cWN*

The minimal symbolic P- and T- semiflows of the *cWN* model of the FMS system can be obtained from the set of P- and T- semiflows, respectively, of the *cWN* models of the subsystems (see Figure 7.3). The symbolic semiflows that cover only one place or one transition, depending on the case, have been omitted from the set of minimal semiflows presented.

The first three symbolic T-semiflows correspond to the extensions of the symbolic T-semiflows of the subsystems 1, 2 and 3. The last symbolic T-semiflow is formed when closing the subsystem_C.

7.8.2 CRG of the *cWN*

Given the information about the SRGs of the subsystems let us analyse how this information can be used to construct the CRG of the overall system. There will only be one SRG since the dynamic sub-classes of the initial marking of the subsystems all correspond to static sub-classes. Given the dimensions of the

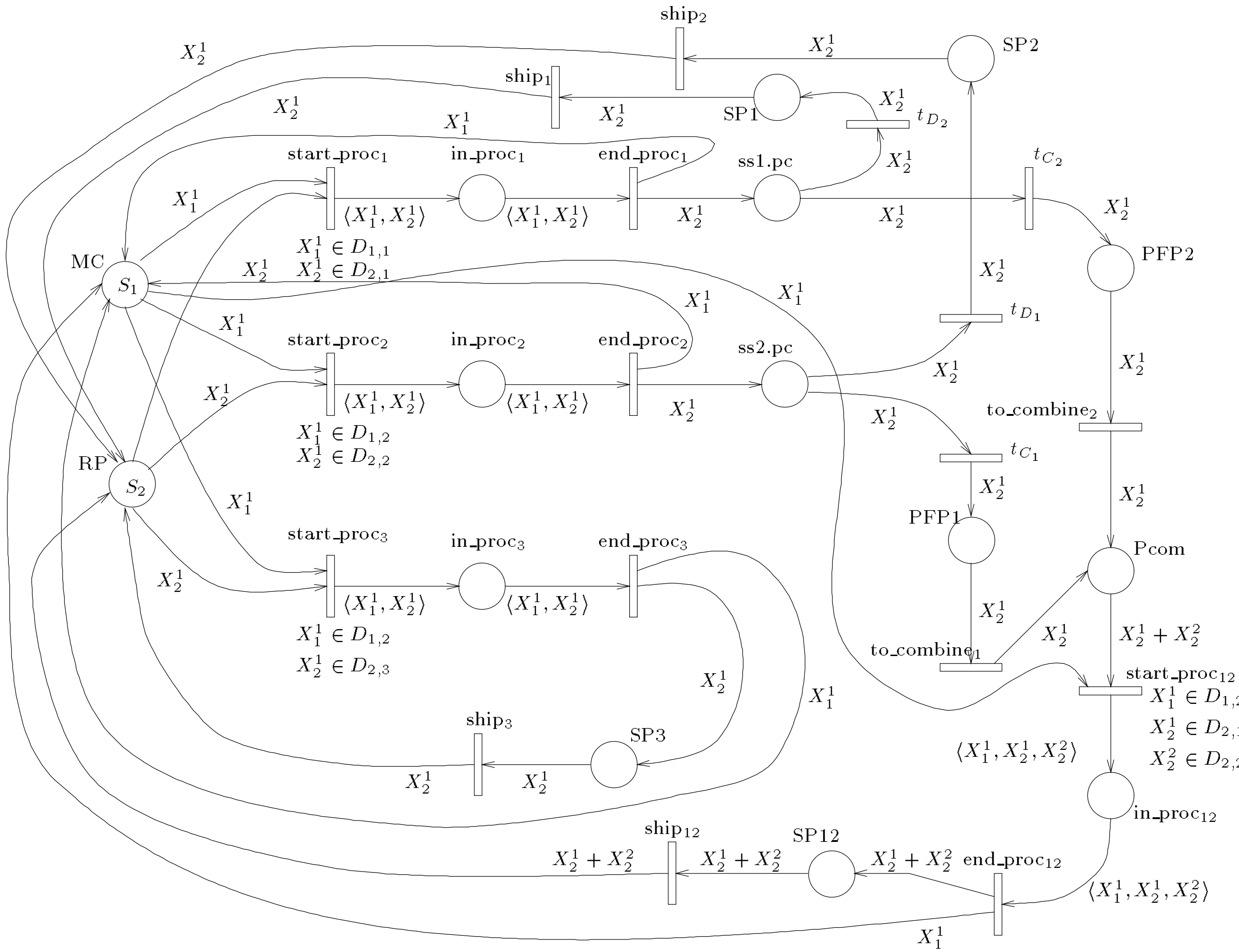


Figure 7.10: *cWN* system modelling the FMS system

Minimal Symbolic P-semiflows

MC	RP	in_proc1	in_proc2	in_proc3	in_proc12	SP1	SP2	SP3	SP12	ss1.pc	ss2.pc	Pcom	PFP1	PFP2
$S_{1,1}$	0	$\langle S_{1,1}, S_{2,1}\rangle$	0	0	0	0	0	0	0	0	0	0	0	0
$S_{1,2}$	0	0	$\langle S_{1,2}, S_{2,2}\rangle$	$\langle S_{1,2}, S_{2,3}\rangle$	0	0	0	0	0	0	0	0	0	0
$S_{1,3}$	0	0	0	0	$\langle S_{1,3}, S_{2,1}, S_{2,2}\rangle$	0	0	0	0	0	0	0	0	0
0	$S_{2,1}$	$\langle S_{1,1}, S_{2,1}\rangle$	0	0	$\langle S_{1,3}, S_{2,1}, S_{2,2}\rangle$	$S_{2,1}$	0	0	$S_{2,1}$	$S_{2,1}$	0	$S_{2,1}$	$S_{2,1}$	0
0	$S_{2,2}$	0	$\langle S_{1,2}, S_{2,2}\rangle$	0	$\langle S_{1,3}, S_{2,1}, S_{2,2}\rangle$	0	$S_{2,2}$	0	$S_{2,2}$	0	$S_{2,2}$	$S_{2,2}$	0	$S_{2,2}$
0	$S_{2,3}$	0	0	$\langle S_{1,2}, S_{2,3}\rangle$	0	0	0	$S_{2,3}$	0	0	0	0	0	0

Minimal Symbolic T-semiflows

start_proc$_1$	end_proc$_1$	ship$_1$	to_combine$_1$	t$_{C_1}$	t$_{D_1}$	start_proc$_2$	end_proc$_2$	ship$_2$	to_combine$_2$	t$_{C_2}$	t$_{D_2}$	start_proc$_3$	end_proc$_3$	ship$_3$	start_proc$_{12}$	end_proc$_{12}$	ship$_{12}$
$\langle S_{1,1}, S_{2,1}\rangle$	$\langle S_{1,1}, S_{2,1}\rangle$	$S_{2,1}$	$S_{2,1}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	$\langle S_{1,2}, S_{2,2}\rangle$	$\langle S_{1,2}, S_{2,2}\rangle$	$S_{2,2}$	$S_{2,2}$	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	$\langle S_{1,2}, S_{2,3}\rangle$	$\langle S_{1,2}, S_{2,3}\rangle$	$S_{2,3}$	0	0	0
$\langle S_{1,1}, S_{2,1}\rangle$	$\langle S_{1,1}, S_{2,1}\rangle$	0	$S_{2,1}$	0	$S_{2,1}$	$\langle S_{1,2}, S_{2,2}\rangle$	$\langle S_{1,2}, S_{2,2}\rangle$	0	$S_{2,2}$	0	$S_{2,2}$	0	0	0	$\langle S_{1,3}, S_{2,1}, S_{2,2}\rangle$	$\langle S_{1,3}, S_{2,1}, S_{2,2}\rangle$	$\langle S_{2,1}, S_{2,2}\rangle$

Table 7.3: P- and T- semiflows of the *cWN* model of the FMS system.

CRG we will not present it, however, we will discuss under which conditions it is possible to re-use the information about the SRGs of the subsystems.

From the view point of the subsystems, the fusion of the RP places and the MC places is reflected as a change of the initial symbolic marking. To be able to re-use the information about the SRG of a particular subsystem it is necessary that the symbolic marking restricted to the places of the subsystem corresponds to a symbolic marking in the SRG of the subsystem. In this sense, the information about SRG of the subsystem 1 can only be used in those markings where the places RP and MC have only elements of $Part_1$ and Mc_1, respectively, or have no coloured objects. Since the transition start_proc$_1$ takes as input rough parts of type $Part_1$ and machines of type Mc_1, we know that there will be no other types of parts or machines in the other places of the subsystem. For all symbolic markings of the cWN of the FMS system that do not satisfy this condition it is necessary to search for the enabled transition in the subsystem 1.

The subsystems 2 and 3 share the machines of type Mc_2 in the place MC. This means that in order to re-use the information about subsystem 2 (or 3) not only is it necessary that there are no rough parts of type $Part_1$ or $Part_3$ (or $Part_1$ or $Part_2$ for subsystem 3) in RP and no machines of types Mc_1 and Mc_3 in place MC, but it also must hold that the total number of machines of type Mc_2 in the places of the subsystem is 2 (the number of machines of this type in the cWN of subsystems 2 and 3).

7.9 An alternative cWN system

As discussed in the previous section, the construction of the CRG of the cWN system modelling the FMS system cannot use much of the information about the CRG of the subsystems. Let us analyse the cause of this and see how the situation can be improved.

The system description refers to three types of machines and three types of rough parts. When modelling systems with WNs the tendency is to group colours identifying the same type of objects, in this case machines and rough parts, and refinements of this basic colour classes are defined as static sub-classes. Colour domains are defined at the level of basic colour classes. Applying this approach to the modelling of the FMS presented leads to the definition of a unique place representing machines (MC) and of another representing rough parts (RP). However, the behaviour of the different types of machines are completely independent and different, similarly the behaviours of the rough parts.

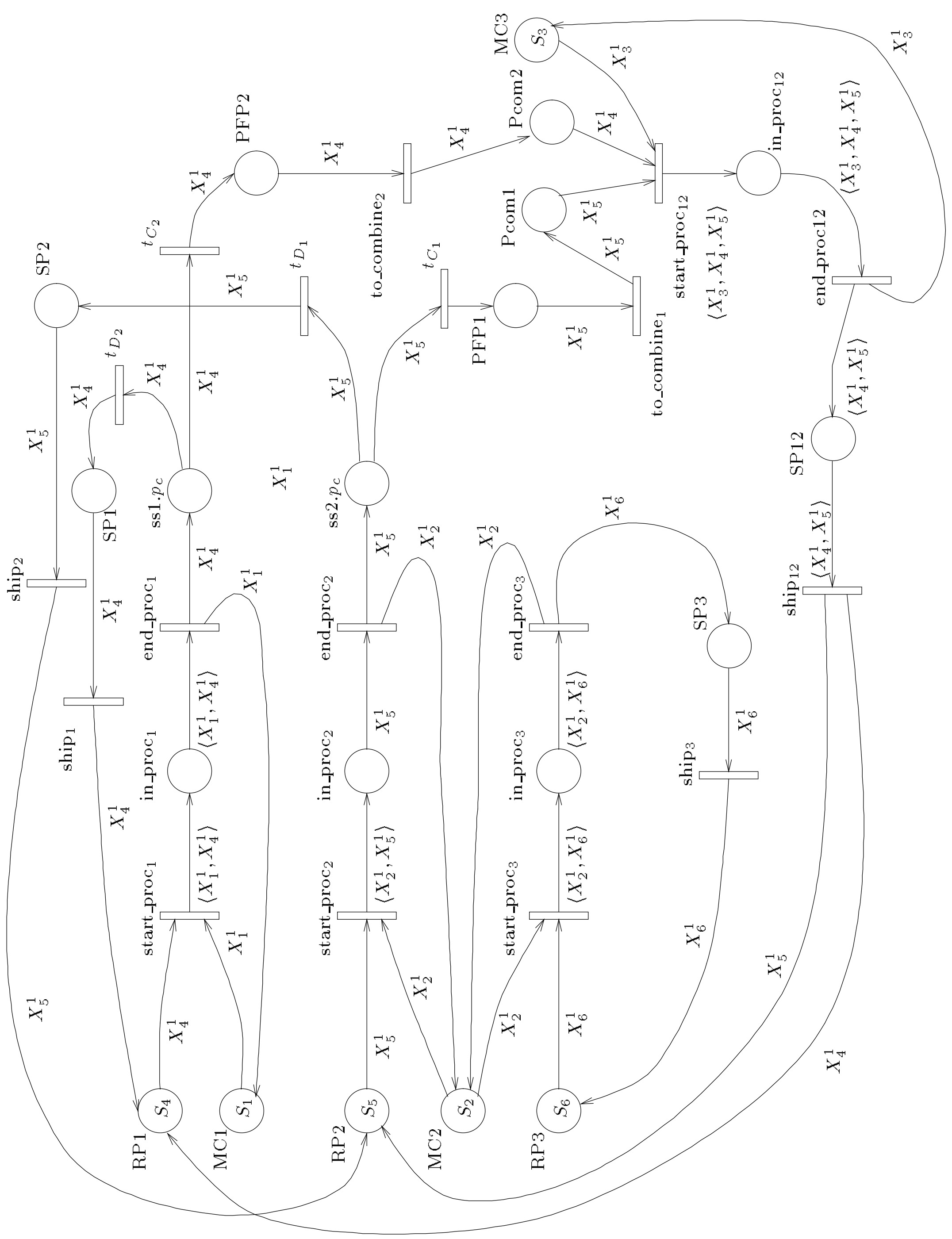


Figure 7.11: An alternative cWN system to model the FMS system.

Symmetric behaviour can be better exploited if instead of grouping the machines into a unique colour class we defined a basic colour class for every type of machine and for every type of rough part. In this way we can define places with colour domains corresponding to types of machines or (rough) parts. For the compositional construction of the *cWN* system presented here this has noticeable advantages. If the input places of the different types of machines and rough parts are separated, then the CRGs of the *cWN* representing the subsystem 1 can be completely reused in the CRG of the FMS. If we used a single place to represent free machines of the different types, the fusion of the place MC in subsystem 1 with the place MC in subsystem 2 would imply that, although the presence of machines of type Mc_2 does not affect the enabled condition of start_proc$_1$, from the point of view of subsystem 1 the marking of the place MC is different, which makes it necessary to check whether star_proc$_1$ is enabled or not. The same applies for the rough parts of type $Part_1$, only processed by machines of type Mc_1. In a similar manner the information about the subsystems 2 and 3 could be used in those cases where the number of machines of type Mc_2 in the places of the subsystem correspond to the number of machines of the same type in the symbolic markings of the subsystem.

With the first *cWN* system, where rough parts where all represented in one place and machines in another, the structure of the model would suggest that the transitions start_proc$_1$, start_proc$_2$ and start_proc$_3$, could, in principle, process any type of machine or rough part. This is restricted by the incorporation of predicates on these transitions. In a *cWN* system where the types of rough parts and machines are separated, the structure of the *cWN* system reflects the behaviour of the FMS system.

Figure 7.11 presents an alternative *cWN* system to model the FMS system. In this case, we will not go through the compositional construction process, confident that the reader is at this stage familiar with the process. The main difference is the separation of the places representing the different types of rough parts and machines. Although the net is larger, the CRG (SRG) has the same dimension as the first system and the symbolic P- and T- semiflows are equivalent to those of the first model, separating the places MC and RP for the P-semiflows. However, as discussed above it is possible to re-use much more information about the SRG of the subsystems and unnecessary verifications of the enabled condition of transition can be avoided.

In both of the *cWN* systems presented (the original and the alternative model of the FMS system) the CRG of the *cWN* system is also a SRG. This is because in the initial marking of the sub-components the composed dynamic sub-classes represent complete static sub-classes. This means that there is only one intersection possible amongst the dynamic sub-classes of the sub-components.

Let us consider the case where the subsystem 2 has as initial marking the following:

$$\langle \overset{MC}{Z_1^2}, \overset{RP}{Z_2^2}, \overset{in\_proc2}{0}, \overset{Pc}{0}, \overset{PFP2}{0}, \overset{SP2}{0}, \overset{CP2}{0} \rangle$$

such that d(Z_1^2)=$D_{1,2}$ (machines of type 2), d(Z_2^2)=$D_{2,2}$ (parts of type 2) and card($Z_1^2 < |D_{1,2}|$); and subsystem 3 has initial marking:

$$\langle \overset{MC}{Z_1^2}, \overset{RP}{Z_2^3}, \overset{in\_proc3}{0}, \overset{SP3}{0} \rangle$$

such that d(Z_1^2)=$D_{1,2}$ (machines of type 2), d(Z_2^3)=$D_{2,2}$ (parts of type 3) and card($Z_1^2 < |D_{1,2}|$).

When composing the initial symbolic markings of the subsystems to obtain the initial composed marking of the FMS system we have the case where there is more than one possible intersection between the dynamic sub-classes Z_1^2 of the subsystems 2 and 3. The initial composed marking will then represent several symbolic markings and the CRG will contain several SRGs.

Chapter 8

Extending the methodology to Stochastic Well-formed nets

8.1 Introduction

So far we have studied the compositional construction and analysis of WN systems with no notion of time. Without such notion it is only possible to analyse qualitative properties of the system modelled. In this chapter we incorporate the notion of time into *cWN*s and study its effect on the proposed methods for the construction and analysis of *cWN* systems. This can be seen as a step towards the definition of a methodology for the performance oriented compositional construction and analysis of Stochastic Well-formed net systems modelling parallel and distributed systems.

To begin the chapter, in Section 8.2 we offer the necessary background on Petri net formalisms with the notion of time. Starting from the basic definition of *Transition Timed Petri nets* we build up to the definition of *Stochastic Well-formed nets* [CDFH91, CDFH93], which will be the formalism used when incorporating the notion of time into *cWN*s. In Section 8.3 we review some of the existing work in the area of compositionality in SPN. Many of the works reviewed introduce compositionality mainly to facilitate the procedures for quantitative analysis of SPN models. Others, more related to our approach, introduce compositionality for the construction of SPN models using the characteristics of SPNs to represent different behaviours of the models. We then go on, in Section 8.4, to define a *basic SWN* (*bSWN*) and a *compositional SWN* (*cSWN*), based on the concepts of a *bWN* and of a *cWN*, respectively. We study how the composition operations can be redefined based on the definition of *cSWN*s.

Structural analysis is not influenced by the incorporation of time or priorities, therefore the methods proposed for compositional structural analysis of *cWN*s are applicable to *cSWN*s. This is not the case with the construction of the Composed Reachability Graph (CRG). In Section 8.5 we study the necessary changes in the compositional methods for the construction of the CRG to appropriately support the notions of time and priority. We conclude this chapter by highlighting the main modifications which have been necessary and by presenting the direction of future work in this area.

8.2 Time in Petri net formalisms

The objective of this section is to offer a general overview of the characteristics of (stochastic) timed Petri nets, as background material to the contents of this chapter.

8.2.1 Stochastic Petri nets

Transition Timed Petri Nets (TTPN) are an extension of Petri nets (PN). With this modification quantitative analysis of the system is also possible. In TTPN a firing delay is associated with each transition. Formally, a TTPN system is defined as:

$$TTPN = \langle P, T, In, O, M_0, \Delta \rangle$$

where $\langle P, T, In, O, M_0 \rangle$ defines a PN system and $\Delta = \{d_1, \ldots, d_{|T|}\}$ is a set of delays associated with each of the transitions of the PN system.

The interpretation of the delay is given according to the firing rule selected. Two different firing rules have being proposed for TTPNs. Under the first rule a transition completes the firing operation in three phases:

starting phase consumes no time; tokens are removed from the input places;

firing-in-progress phase with which the delay is associated; and

ending phase consumes no time; tokens are added to the output places.

The second firing rule assumes that a transition firing operation is atomic, i.e. the removal of tokens from input places and the placement of tokens in output places is a single indivisible operation. Here the delay associated with the transition represents the time interval for which the transition must remain enabled before it can fire. Transitions are said to be *conflicting* if they are simultaneously enabled. Under this second proposal, conflicts are resolved using a *race model* to determine which of the enabled transitions will fire. When a marking enables several transitions all activities associated with the transitions are assumed to execute in parallel, so that the next marking change is due to the transition that wins the race (finishes first). The implementation of the race policy has three possibilities. It can be considered that once a race has been won all other transitions remember the amount of time for which they have already been enabled, that they remember the amount of time since last enabled if still enabled after the marking change produced by the firing of the winning transition, or that they forget it and start all over again, when next enabled.

A *Stochastic Petri Net* (SPN) system is obtained from the TTPN system by associating an exponentially distributed random firing delay with each transition. Formally a SPN system is defined as:

$$SPN = \langle P, T, In, O, M_0, \mathcal{F} \rangle$$

where $\langle P, T, In, O, M_0 \rangle$ defines a PN system and $\mathcal{F} = \{fr_1, \ldots, fr_m\}$ is a set of exponentially distributed firing rates associated with each of the transitions of the PN system. Conflicts in SPNs are resolved using a race policy where the participating transitions forget the amount of time for which they have been enabled, once the race has been won.

In [Mol82] it is shown that due to the memoryless property of the exponential distribution of the firing delays, the RG of a SPN system is isomorphic to a continuous time Markov Chain (CTMC), where the transition rate from a state i (corresponding to a marking M_i of the SPN) to a state j (corresponding to a marking M_j) is equal to the sum of the rates of the transitions that connect the markings M_i and M_j in the RG. If the CTMC is ergodic[1], its solution provides the steady state distribution on the markings of the SPN system. From this distribution it is possible to obtain quantitative estimates of the behaviour of the SPN system [AMBC84]. One of the main limitations of SPNs is that the number of states in the associated CTMC grows very quickly with the dimensions of the net.

8.2.2 Generalised stochastic Petri nets

Generalised SPNs (GSPNs) [AMBC84] try to mitigate the problem of state space explosion of the CTMC by defining two types of transitions: *timed transitions* (with exponentially distributed firing rates) and *immediate transitions* (with firing time 0). Immediate transitions have firing priority over timed transitions, and different priority levels can be defined over immediate transitions. Formally a

[1]A finite CTMC is said to be *ergodic* if and only if its graph representation is strongly connected. For further details and a more formal definition the reader is referred to [AMBC$^+$95].

GSPN is a defined as:

$$GSPN = \langle P, T, In, O, M_0, \theta, \pi \rangle$$

where θ assigns firing rates to timed transitions and weights to immediate transitions. The function π assigns priorities to the transitions. Figure 8.1 shows an example of a GSPN. Timed transitions are represented by empty boxes and immediate transitions by non-arrowed lines.

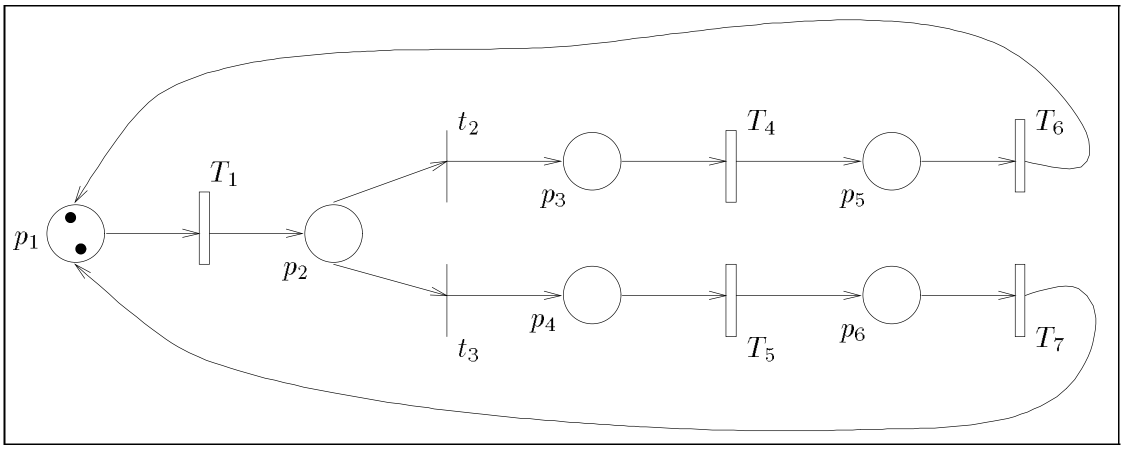


Figure 8.1: Example of a GSPN.

Conflicts amongst timed transitions are resolved as in SPNs. In the case of simultaneously enabled immediate transitions with equal priorities the weights associated with each of the enabled transitions are employed to calculate the firing probability of each of them. The probability that an immediate transition t_i, enabled in a marking M, is selected to fire will be given by:

$$P(t_i \mid M) = \frac{\theta(t_i)}{\sum\limits_{t \in E_{im}(M)} \theta(t)}$$

where $E_{im}(M)$ is the set of immediate transitions enabled in marking M and $\theta(t_i)$ is the weight associated with the immediate transition t_i.

A marking is said to be *tangible* if it enables no immediate transition, otherwise it is called *vanishing*. A marking which does not enable any transition is *absorbing*; hence it is tangible by definition.

8.2.3 Stochastic Well-formed nets (SWNs)

The step from WNs to Stochastic Well-Formed nets (*SWN*s) [CDFH91, CDFH93] is similar to that from untimed PNs with priorities to GSPNs (see [AMBC+95] for going from an untimed PN to a GSPN). Transitions can be *timed* (with an exponentially distributed delay function) or *immediate* (with firing time zero). In order to exploit the presence of symmetry, not only from a logical, but also from a stochastic point of view, mean values of transition firing delays can be dependent only on static subclasses; they cannot be a function of the individual objects in the class. In this way all objects of a given static subclass give rise to the same transition firing delay. This can be formalised by the introduction of the following notation [CDFH93]; let

$$\tilde{C}_i = \{D_{i,1}, \cdots, D_{i,q}\}$$

be the set of static subclasses of a basic colour class C_i. Given a transition t with colour domain $C(t)$, the colour domain of t defined in terms of the static subclasses of the basic colour classes in $C(t)$, is given by

$$\tilde{C}(t) = \left\{ \bigotimes_{i=1}^{n} \bigotimes_{j=1}^{e_i} D_{i,u(i,j)} \mid 0 < u(i,j) \leq n_i \right\}$$

where e_i denotes the number of occurrences of C_i in $C(t)$ and $u(i,j)$ determines which of the static subclasses of C_i is represented in the j^{th} occurrence of C_i in the colour domain of t.

In the same manner for all colour tuples $c \in C(t)$, $c = \bigotimes_{i=1}^{n} \bigotimes_{j=1}^{e_i} \; c_i^j$ it is possible to represent c in terms of the static subclasses:

$$\tilde{c} \in \tilde{C}(t), \; \tilde{c} = \bigotimes_{i=1}^{n} \bigotimes_{j=1}^{e_i} \; \tilde{c}_i^j$$

such that $\tilde{c}_i^j \in D_{i,q}$ if and only if $c_i^j \in D_{i.q}$. A *static partition of a marking* M denoted $\tilde{M}(p) \in Bag(\tilde{C}(p))$ is defined as:

$$\tilde{M}(p)(\tilde{c}) = \sum_{c' : \tilde{c}' = \tilde{c}} M(p)(c')$$

This marking partition represents, for the place p and for each Cartesian product of static subclasses, the set of tuples in the place that belong to the same Cartesian product of static subclasses. The markings within a symbolic marking $\mathcal{M}$ will satisfy the following property:

$$\forall M, M' \in \mathcal{M}, \;\; \forall p \in P, \;\; \tilde{M}(p) = \tilde{M}'(p)$$

Definition 8.1 (Stochastic Well-Formed net (from [CDFH93])
A Stochastic Well-Formed coloured net *is a triple SWN=* $\langle WN, \pi, \theta \rangle$ *such that*

- WN *is a well-formed Petri net; we distinguish two types of transitions, namely timed and immediate;*

- $\pi : T \rightarrow \mathbb{N}^{+}$ *is the priority function. Timed transitions will have priority zero, while immediate transitions will have priorities greater than zero;*

- θ *is a function defined on the set of transitions* T *such that*

$$\theta(t) : \tilde{C}(t) \times \bigotimes_{p \in P} Bag(\tilde{C}(p)) \rightarrow \mathbb{R}^{+}$$

For any timed transition t, the function $\theta(t)(\tilde{c}, \tilde{M})$ represents the average firing rate for any instance $[t, c]$ of transition t enabled in marking M. If t is an immediate transition, the same function is interpreted as the weight to be normalised within a conflict set in order to obtain the following probability that the transition instance $[t, c]$ will fire in M:

$$\frac{\theta(t)(\tilde{c}, \tilde{M})}{\sum_{M[t',c'\rangle} \theta(t')(\tilde{c}', \tilde{M})}$$

As for GSPNs, immediate transitions have firing priority when conflicting with timed transitions.

8.3 Compositionality in Stochastic Petri nets

The methods studied in Chapter 3 concentrated on the the composition of PN components for the analysis and deduction of structural or state space related properties of the system modelled. As previously stated, the incorporation of the notion of time permits the analysis of quantitative or performance properties of the system modelled. The use of SPNs in the study of the performance of concurrent systems is a well established field. In this section we review some of the existing work in the area of compositionality in SPN systems. As in the case of the compositional methods working on untimed PNs these methods are motivated by different objectives.

Quantitative analysis

There have been many studies devoted to the definition of compositional methods that facilitate and improve the procedures for quantitative analysis of SPN models. Donatelli's work on Superposed Stochastic Automata (SSA) [Don93] and Superposed GSPN (SGSPN) [Don94] emphasises composition from the point of view of the solution of the associated Markov process. SSA is a subclass of SPN, that can be considered as a set of stochastic state machines (SSMs). The SSMs of a SSA are considered to be joined by synchronising transitions, where transitions that synchronise are assumed to have the same rate. Donatelli shows how the solution of the Markov chain associated with a SSA (SGSPN) can be obtained without having to generate or store its complete infinitesimal generator matrix.

Buchholz papers [Buc92, Buc93, Buc94a] on Hierarchical High Level Petri nets emphasise hierarchical decomposition methods, based on asynchronous composition of coloured GSPNs and SWNs. Analysis is based on a divide and conquer approach combined with behaviour preserving reduction techniques on subsets of the reachability set. Subnets are formed by a detailed view of the local parts and an aggregate view of the rest of the net, corresponding to the environment of the subnet. Subnets communicate with their environments by exchanging tokens. Aggregation groups states of the associated Markov Chain, which is solved using sets of state classes.

The works by Haddad and Moreaux [HM95, HM96] combine the results of Donatelli and Buchholz to study efficient methods for the analysis of quantitative properties of a GSPN systems without having to build the whole Markov chain associated with it. These works combine aggregation and decomposition methods. In [HM95] they emphasise synchronisation of subnets and in [HM96] they study the asynchronous composition of high level Petri nets. Their objective is to develop an aggregation method based on the SWN formalism while keeping the advantage of the decomposition methods for asynchronous composition of systems.

In [Buc95] Buchholz combines equivalence and composition notions to propose a method for the qualitative and quantitative analysis of SPN systems. This approach also allows the compositional construction of the state space and of the generator matrix. Subnets can be composed by synchronisation of transitions. The equivalence relation defined in Buchholz approach allows the substitution of a subnet of a component by a equivalent representation without altering the quantitative behaviour of the complete net.

In [HHMR97] Hermanns et al. combine the composition and substitutive equivalences for the construction and performance analysis of GSPN systems. The hierarchical construction of GSPN systems by composition of smaller nets is used both to construct complex GSPN systems and during the generation of the state space. For the latter they propose a method for the stepwise compositional reduction of the state space, based on equivalence notions that preserve the properties (performance in particular) of the GSPN system. The composition operations defined are based on the parallel and hiding operators of LOTOS.

Construction of models

In [BDF95], the Process/Resource (P/R) boxes [BDC93] are combined with GSPN and SWN for performance modelling. Here the composition rules of P/R boxes are extended to a high level formalism. Following this work, Donatelli and Franceschinis, in [DF96], propose the PSR (Processes, Services, Resources) methodology, based on P/R Boxes. The resource level describes the operations offered by the hardware. At this level timed and immediate transitions are combined. Timed transitions represent activities and immediate transitions are used to represent the start and the end of activities. The other two levels (processes and services) have only immediate transitions, thus the time factor emerges from the hardware level. Using a labelled GSPN system it is possible to offer services to a higher level (by labels over immediate transitions) or define cooperation between transitions of the same level (by labels over timed transitions). The service level is used to implement the services requested at the processes level with those offered by the resource level.

Ferscha's work ([Fer92]) is developed as part of a Computer Aided Parallel Software Engineering (CAPSE) system. A Programming Resource Mapping net (PRM) model serves as an integrated performance model of parallel processing systems. Complex structures of parallel programs are represented by compositions of processes. Processes can be composed by sequential composition, parallel composition, communication, alternative or iteration. The performance of the system is not calculated by using methods based on the CTMC associated with GSPNs. Instead the author uses a combination of simulation and aggregation.

In [SM91], Sanders and Meyer define composition operations to build a model taking advantage of the structure of the system modelled and the performance variables that are to be obtained from the analysis of the model. This method is proposed over Stochastic Activity Networks (SAN), a flavour of Petri Nets with possibility of defining complex enabling conditions and (marking dependent) state changes for transitions. The determination of the performance variables desired is used to choose an appropriate notion of state, typically reducing the number of states that must be considered for an analytical solution without requiring the generation of a detailed state space.

8.4 From *cWN*s to *cSWN*s

In order to be able to incorporate timing specifications into *cWN*s we extend *cWN*s with the characteristics of SWNs. This will allow the performance analysis of the system modelled as well as offering methods for the compositional construction and structural and state space analysis of the system. This follows the idea of SPA as an extension of PA.

8.4.1 Stochastic Process Algebra

Stochastic Process Algebra (SPA) constitute an extension of Process Algebra. They are generally based on untimed PA and extend the basic actions with exponential delays. With the use of SPA it is possible to do performance analysis in process algebras. Examples of SPA are PEPA [Hil94a], TIPP [GHR93], MPA [Buc94b], EMPA [BBG95] and Markovian LOTOS [HR96]. By combining the notions of observational equivalence and compositional reasoning, offered by PA, with the modelling of the timing behaviour by means of random variables, it is possible to obtain compositional methods for the specification and performance evaluation of concurrent systems. The generation of the underlying Markov process is then used to derive performance measures of the modelled system. The SPA diverge in their definition of actions, in their syntax and in their operational semantics.

In many SPAs actions are divided into *active* and *passive*. Passive actions have no delay associated with them, they represent processes waiting to communicate. Active actions are divided into *timed* (with exponential distributed delays) and *immediate* (taking no time).

A conflict in a SPA occurs in a state where there is more than one executable active action. As in GSPNs, conflicts in SPAs are, in general (according to the types of actions and of delays supported by the individual SPAs), solved by applying a race policy (in the case of all conflicting actions being timed), priorities (in the case of timed and immediate actions) and or weighted probabilities or different levels of priorities in the case of all conflicting actions being immediate.

The definition of synchronisation, otherwise termed cooperation, of actions in the various SPAs varies mainly according to the way in which the delay associated with the resulting action is defined and the types of actions that are allowed to take part.

8.4.2 Compositional SWN ($cSWN$) systems

The extension of cWNs to support timing specification follows the idea that bWNs represent system functions and that the composition operations represent relations between these functions. For this reason, and to maintain the simplicity of the model, the type of transitions (timed or immediate) is determined by the definition of a *bSWN* and of the composition operations.

8.4.3 The basic component

A *basic SWN* (*bSWN*) is a *bWN* in which the transition of the *bWN* has an exponentially distributed delay associated with it. It has been defined in this manner to avoid the construction of SWN systems that take no time. Immediate transitions will only be introduced as part of a probabilistic choice.

Formally, a *bSWN* $\mathcal{N}$ is defined as:

$$\mathcal{N} = \{bWN, \pi_{\mathcal{N}}, \theta_{\mathcal{N}}\}$$

where

- *bWN* is a basic well-formed net with a transition t,
- $\pi_{\mathcal{N}}$ is the priority function, defined as $\pi(t) = 0$ (timed transitions have priority 0),
- $\theta_{\mathcal{N}}$ is the delay function that determines the average firing rate of t.

In the same manner that *cWN*s have being defined, we can define a *composable SWN* as:

Definition 8.2 *A* composable SWN *(cSWN) is either a bSWN or a composition of cSWNs.*

$$cSWN ::= bSWN \mid cSWN * cSWN \mid \circ cSWN$$

where $*$ represents any binary composition operation and $\circ$ any unary operation.

Having defined a *bSWN* and a *cSWN* let us now study the necessary changes in the composition operations defined for *cWN* in order to support the concepts of *bSWN* and *cSWN*.

8.4.4 The compositional operations

In general, the composition operations require no major change other than the definition of the priority and delay or weight functions (π and θ) in the resulting *cSWN*s.

For most cases in the composition of two *cSWN*s L and R to obtain a *cSWN* $\mathcal{N}$ the π and θ functions will be defined in the following way:

$\forall t \in T_{\mathcal{N}}$,

$$\pi_{\mathcal{N}}(t) = \begin{cases} \pi_L(t) & \text{if } t \in T_L \\ \pi_R(t) & \text{if } t \in T_R \end{cases}$$

and

$\forall t \in T_{\mathcal{N}}$,

$$\theta_{\mathcal{N}}(t) = \begin{cases} \theta_L(t) & \text{if } t \in T_L \\ \theta_R(t) & \text{if } t \in T_R \end{cases}$$

For place fusion operations over a single component *cSWN* S, to obtain a *cSWN* $\mathcal{N}$, it holds that $\pi_{\mathcal{N}} = \pi_S$ and $\theta_{\mathcal{N}} = \theta_S$.

However, the choice composition and the synchronisation operation require further changes. These are explained in detail below.

8.4.4.1 Choice (pre-selection) composition

Maintaining the idea of a choice composition as a logical decision, in the context of *cSWN*s the choice composition is defined as a probabilistic selection of the sub-component to which a given type of information should be transferred. Each sub-component is assigned a weight $w(Q)$, where Q is the name of the sub-component. This weight is associated with the probability that the sub-component Q will receive the information related to the choice.

The transitions associated with each of the participating sub-components will be immediate transitions. The weight assigned to the immediate transition t_Q (associated with the sub-component Q) will be $w(Q)$.

The initial marking of a component should be non-vanishing. This will mean that the choice place will have initial marking zero. The idea is that before making a decision, there must be some processing.

The functions π and θ of the *cSWN* $\mathcal{N}$ resulting from the choice composition of two *cSWN*s L and R, represented as $L + R/\{w(L), w(R)\}$, will be defined as

- $\forall t \in T_{\mathcal{N}}$,

$$\pi_{\mathcal{N}}(t) = \begin{cases} \pi_L(t) & \text{if } t \in T_L \\ \pi_R(t) & \text{if } t \in T_R \\ 1 & \text{if } t = t_L \ \vee \ t = t_R \end{cases}$$

  and

- $\forall t \in T_{\mathcal{N}}$,

$$\theta_{\mathcal{N}}(t) = \begin{cases} \theta_L(t) & \text{if } t \in T_L \\ \theta_R(t) & \text{if } t \in T_R \\ w(L) & \text{if } t = t_L \\ w(R) & \text{if } t = t_R \end{cases}$$

8.4.4.2 Synchronisation

One of the main advantages that PNs have for modelling concurrent systems is the number and variety of synchronisations that they can represent. In [Hil94b] Hillston reviews several types of synchronisation. In [RM96] we have modelled most of these synchronisations using SWN models. However, in this chapter we have limited our work to synchronisations that are represented by the fusion of pairs of timed transitions. This has been done in order to maintain the simplicity of the methods proposed and to avoid the introduction of inhibitor arcs and of immediate transition in operations other than the choice composition.

The types of synchronisation introduced as part of the compositional method proposed are *impolite communication* and *patient communication* [Hil94b].

In patient communication the interaction is assumed to represent a communication or shared task. The rate of each individual transition represents the capacity of the component to complete its part of the shared task. The interaction is completed by both components working together at the rate of the slower one. Therefore the rate assigned to the fused transition is equal to the minimum of the rate of t_1 and the rate of t_2, $r(t_{syn}) = min\{r(t_1), r(t_2)\}$.

Impolite communication consists of the synchronisation of two transitions each representing a communication event, such that both transitions "transfer information" at the same time. The first to finish its transfer will terminate the communication. This means that the duration of the communication will be distributed as the minimum of the individual distributions. Since the individual transitions are exponentially distributed the transition resulting from their synchronisation will exponentially distributed with rate equal to the sum of the individual rates $(r(t_{syn}) = r(t_1) + r(t_2))$.

These synchronising operations differ only in the rate assigned to the fused transition t_{syn}. Given that both min and $+$ are associative algebraic operations, and that their representations produce a single fused transition t_{syn}, we can apply either of these operations over more than two transitions, fusing pairs of transitions at a time.

The functions π and θ of the $cSWN$ $\mathcal{N}$ resulting from the synchronisation of two $cSWN$s L and R represented as $L \diamond R/\{t_1, t_2\}$, for the Patient Communication, or by $L \odot R/\{t_1, t_2\}$, for the Impolite Communication, will be defined as

- $\forall t \in T_{\mathcal{N}}$,

$$\pi_{\mathcal{N}}(t) = \begin{cases} \pi_L(t) & \text{if } t \in T_L \\ \pi_R(t) & \text{if } t \in T_R \end{cases}$$

  and

- $\forall t \in T_{\mathcal{N}}$,

$$\theta_{\mathcal{N}}(t) = \begin{cases} \theta_L(t) & \text{if } t \in T_L \\ \theta_R(t) & \text{if } t \in T_R \\ \theta_L(t_1) + \theta_R(t_2) & \text{if } t = t_{syn} \text{ and the operation is an} \\ & \text{Impolite communication} \\ min(\theta_L(t_1), \theta_R(t_2)) & \text{if } t = t_{syn} \text{ and the operation is a} \\ & \text{Patient communication} \end{cases}$$

where t_1 and t_2 are the transitions being synchronised.

In the case of the synchronisation being applied over a single *cSWN* S, π and θ functions of the resulting *cSWN* $\mathcal{N}$ are defined as:

- $\forall t \in T_{\mathcal{N}}$,

$$\pi_{\mathcal{N}}(t) = \begin{cases} 0 & \text{if } t = t_{syn} \\ \pi_S(t) & \text{otherwise} \end{cases}$$

  and

- $\forall t \in T_{\mathcal{N}}$,

$$\theta_{\mathcal{N}}(t) = \begin{cases} \theta_S(t) & \text{if } t \neq t_{syn} \\ \theta_S(t_1) + \theta_S(t_2) & \text{if } t = t_{syn} \text{ and the operation is an} \\ & \text{Impolite communication} \\ min(\theta_S(t_1), \theta_S(t_2)) & \text{if } t = t_{syn} \text{ and the operation is a} \\ & \text{Patient communication} \end{cases}$$

8.5 Compositional analysis of *cSWN*s

If we considered all transitions of a *cSWN* to have equal priorities, the application of the proposed compositional methods for the analysis of *cWN* systems to *cSWN* would be direct The compositional construction of P- and T- semiflows are independent of the notion of time, and so are the methods for the construction of the CRG (Composed Reachability Graph) when all transitions are considered to have the same firing priority. However, the incorporation of transition priorities is not that straightforward. The main problem lies in the verification of the enabled state of a transition. Not only must it have the necessary number and type of coloured objects in its input places, but also no transition with a higher priority can be enabled.

8.5.1 The Composed Reachability Graph of a *cSWN*

Consider two *cSWN*s L and R composed to form a *cSWN* $\mathcal{N}$. Given $\mathcal{M}_L$, a symbolic marking of L, and $\mathcal{M}_R$, a symbolic marking R, we want to obtain the set of *directly reachable symbolic markings* (DRS) of the composed marking $\widehat{\mathcal{M}_{\mathcal{N}}}$ formed from the composition of $\mathcal{M}_L$ and $\mathcal{M}_R$. The different levels of priority could imply that a symbolic transition instance enabled in $\mathcal{M}_L$ is no longer enabled in $\widehat{\mathcal{M}_{\mathcal{N}}}$. This would happen if there was a symbolic transition instance enabled in $\mathcal{M}_R$ associated with a transition of higher priority than that of the transition associated with the instance enabled in $\mathcal{M}_L$.

As in *cWN*s, to determine which is the set of enabled composed transition instances of a composed marking, we examine the set of enabled composed transition instances of its sub-markings. To find the enabled composed transition instances of the sub-markings we either use the information of the CRG of the sub-components, whenever possible, or recursively decompose the sub-marking viewing it as a composed marking. Once the enabled instances of the sub-markings have been obtained it is necessary to verify if they will still be enabled at the level of the composed marking.

The initial composed marking of all *cSWN*s is defined to be non-vanishing, since the initial marking of a choice place is assumed to be zero and choice places are the only input places to immediate transitions.

8.5.2 Enabled composed transition instances

Consider a composed marking $\widehat{\mathcal{M}_{\mathcal{N}}}$ of a *cSWN* $\mathcal{N}$, and its composed sub-markings $\widehat{\mathcal{M}^L_{\mathcal{N}}}$ and $\widehat{\mathcal{M}^R_{\mathcal{N}}}$. If both sub-markings are of the same type (i.e. vanishing or tangible) then :

$$\mathrm{E}(\widehat{\mathcal{M}_{\mathcal{N}}}) = \mathrm{E}(\widehat{\mathcal{M}^L_{\mathcal{N}}}) \ \cup \ \mathrm{E}(\widehat{\mathcal{M}^R_{\mathcal{N}}})$$

i.e. the set of enabled transitions of $\widehat{\mathcal{M}_{\mathcal{N}}}$ is formed by the union of the enabled transitions in its sub-markings; otherwise the set of enabled composed transition instances will correspond to the set of composed transition instances of the vanishing sub-marking.

In the case of competing parallelism composition there are further considerations that must be made. These are explained below.

8.5.2.1 Under competing parallelism

Let us consider the possibility of performing a fusion (under competing parallelism) of a choice place with an "ordinary" (non-choice) entry place. The arc function from a choice place into any of its associated immediate transitions is always defined as the identity function (according to the colour domain of the choice place), which has cardinality one. Therefore, if there is at least one object in a choice place then there is an immediate transition enabled. In a marking where the fused place has at least one object the immediate transitions associated with the choice place will always have firing priority over the timed transition(s) associated with the ordinary entry place, not allowing the timed transition(s) to ever fire. For this reason we impose the following restriction over the fusion of entry places.

Restriction 8.1.-
Choice places can only be fused with other choice places.

Given this restriction the determination of the enabled composed transition instances follows more or less the same process as in other compositional operations. If a marking component of a composed marking is a vanishing marking in its corresponding sub-component, then it is only necessary to determine the enabled state of immediate transitions in the composed marking. Immediate transitions previously enabled in the marking components will continue to be enabled in the composed marking. To verify the enabled condition of immediate transitions it is only necessary to check that their corresponding choice places have markings greater than zero.

In the case of competing parallelism over a single component Proposition 6.1 *"The symbolic transition instances enabled in a composed marking $\widehat{\mathcal{M}_S}$ of S, will also be enabled in its collapsed composed marking $\widehat{\mathcal{M}_{S'}}$ of S'"*
still holds.

Having transitions of different priorities, for a composed transition instance $[t, \lambda, \mu]$ to be enabled in $\widehat{\mathcal{M}_S}$ it must hold both that:

- $\forall p \in P_S,\ W^-(p,t)(\lambda,\mu) \leq R_S.mark(p)$ and $\Phi(t)(\lambda,\mu)$, i.e. the standard predicate associated with the transition must evaluate to TRUE; and
- $\forall t' \in T_S$ with $\pi(t') > \pi(t)$, $\exists p \in P_S$ such that $W^-(p,t)(\lambda,\mu) > R_S.mark(p)$ or $\neg\Phi(t)(\lambda,\mu)$, i.e. no symbolic transition instance associated with a transition of a higher priority is enabled.

In a *cSWN* immediate transitions can only appear as part of a choice composition. Therefore, given a enabled symbolic transition instance associated with an immediate transition, this symbolic instance cannot be disabled when fusing its input choice place with another choice place.

When the enabled composed transition instance is associated with a timed transition, it means that there is no higher priority transitions with an enabled composed firing instance, i.e. all choice places, if any, have marking zero. The marking of a fusion place is obtained by the sum of the markings of the places that are fused. Therefore the marking of a dynamic subclass in a place cannot decrease, maintaining in this way the enabled state. Places not involved in the fusion operation maintain the marking of the original composed marking $\widehat{\mathcal{M}_S}$. We can therefore conclude that enabled composed transition instances in $\widehat{\mathcal{M}_S}$ will also be enabled composed instances in $\widehat{\mathcal{M}_{S'}}$.

As we stated when analysing the *cWN*s, this does not mean that there cannot be new composed transition instances enabled in $\widehat{\mathcal{M}_{S'}}$ that were not enabled in $\widehat{\mathcal{M}_S}$.

To verify if new symbolic transition instances are enabled we can use information about the type of the composed marking of $\widehat{\mathcal{M}_S}$. If $\widehat{\mathcal{M}_S}$ is vanishing then using Proposition 6.1 we can deduce that $\widehat{\mathcal{M}_{S'}}$ will also be vanishing. This implies that we only need to check for new composed transition instances amongst immediate transitions. Similarly if $\widehat{\mathcal{M}_S}$ is a tangible marking, then so will be $\widehat{\mathcal{M}_{S'}}$. Therefore the search for new enabled composed transition instances will be limited to timed transitions. The same type of analysis can be applied for the closing operation.

8.5.3 Properties of the CRG

Following the deductions applied when using the SRG instead of the RG [CDFH93], in order to derive an improved technique for performance evaluation based on the CRG instead of the SRG, it is necessary to know how to prove/test the ergodicity of the Markov Chain(s) and how to compute its transition rates. Proposition 6.3 on page 173 defines the necessary condition for ergodicity:

Necessary condition for ergodicity:

> *Strong connectivity of a SRG of* $\mathcal{N} \Rightarrow$ *Strong connectivity of the CRG, but not vice versa,*

and Proposition 6.5, on the same page, defines a sufficient condition for ergodicity:

Sufficient condition for ergodicity:

> *Strong connectivity of the CRG* $\wedge\ \exists \widehat{\mathcal{M}} \in$ *CRG such that all its lowest level dynamic sub-classes correspond to static sub-classes* $\Rightarrow$ *there is only one SRG in the CRG and it is strongly connected.*

All other properties of the CRG are still valid when considering timed transitions and different levels of priorities.

8.6 Conclusions

In this chapter we have extended the definition of *cWN*s to incorporate the notion of time and have studied the effects of these changes in the compositional construction and analysis methods proposed. These are the first steps towards the definition of a methodology for the compositional construction and analysis of performance-oriented PN systems of parallel and distributed systems.

We have defined *compositional Stochastic WNs* (*cSWN*), based on the definition of *cWN*s and using Stochastic Well-formed nets (SWNs) as the basic formalism. A *cSWN* is defined as a *cWN* with two additional functions: a priority function π and a delay or weight function θ. We explain how these functions are defined in the *cSWN* resulting from a composition operation. To maintain the choice operation as completely probabilistic, we have defined the transitions added by the choice operation as immediate transitions. The synchronisation operation has also been redefined to take into account different types of synchronisation between timed transitions.

The methods proposed for the structural analysis of *cWN*s are completely independent of the notion of time and priority, therefore these methods require no modification for their application to *cSWN*s. Unfortunately, the same cannot be said for the methods for the compositional construction of the state space of *cWN*s. Here the time factor has no effect; however the different levels of transition priorities affects the process of determining of enabled transition instances. We have studied these effects and proposed solutions in order to apply the compositional methods to *cSWN*s.

Immediate transitions can be incorporated into the model only as part of a choice composition. This, and the fact that it is not possible to define predicates for immediate transitions or arc functions for immediate transitions different from the identity function, restrict the representation of certain types of behaviour. Pre-emption and more complex types of synchronisation are examples of such behaviours. However, we want to avoid the definition of functions that consume no time and to maintain the same approach of design and construction. Therefore, a way to increase and give more flexibility to the use of immediate transition in the models would be by incorporating other operations which model the desired behaviours.

Considering the different types of transitions (i.e. timed or immediate) present in *cSWN*s, there is still some work that can be done to improve the method for the compositional construction of the CRG. Simple improvements would be to start computing the set of enabled composed transition instances by analysing the sub-components with immediate transitions. This follows the approach of construction of the RG and the SRG, and avoids having to generate the enabled sets of transitions for sub-components with tangible sub-markings whenever there are sub-components with immediate transitions with an enabled composed transition instance.

In this chapter we have also reviewed some of the existing work in the area of compositionality in SPN. Many of the works reviewed concentrate on the use of compositionality to improve the procedures for the quantitative analysis of SPN. Future work on *cSWN*s will be aimed at trying to obtain a compositional method for the quantitative analysis of *cSWN*s. Alternatives could be to study the CRG and its numerical characteristics, or to use other method, such as, approximation or bounds of the performance of *cSWN*s or to identify sub-components that have specific numerical properties.

Chapter 9

Conclusions

9.1 Introduction

This chapter presents a summary of the main results of this dissertation. In Section 9.3 we evaluate the methodology proposed, analysing how it meets the criteria presented in Chapter 3 for a methodology for the modelling of parallel and distributed systems. Finally, in Section 9.4, the directions of future work and on future development of the methodology proposed are discussed.

9.2 Summary

A methodology for the compositional construction and analysis of WN systems has been presented, aimed at offering a method to support the specification and design of parallel and distributed systems. WNs allow a natural representation of complex distributed systems, maintaining the same expressive power as the unconstrained coloured PN formalism.

The compositional method presented is based on the definition of a set of composition operations. These operations have been defined mimicking the operators of Process Algebra (PA), taking into account, at the same time, the characteristic of WNs and the behaviours that they can model. The WN models that can be built using the composition operations proposed are termed Composable WNs (cWNs). They constitute a subclass of WNs where there are no inhibitor arcs and where immediate transitions can only be incorporated as part of a probabilistic choice. A cWN can be built starting from basic WNs (bWNs) or from the composition of other cWNs components. A cWN has an interface by which it can communicate with other cWNs. Composition is resolved by place fusion, transition fusion or using a sub-net. To guide the modeller in the compositional construction of cWN models, general guidelines, based on a precedence relation between the composition operations, have been discussed.

The main objective of compositionality is to be able to re-use the information about the sub-components of a component to obtain information about the component's behaviour. In this sense, in this dissertation we have studied the compositional construction of structural and state space information of *cWN* systems. The study of the compositional construction of the structural characteristics of a *cWN* is based on the analysis of the matrix representation of the net—the *incidence matrix*—and how it can be built from the incidence matrices of its sub-components. From this analysis we study and propose methods for the construction of semiflows of a *cWN* model based on the semiflows of its sub-components. Using the structured definition of colour classes and arc functions in WNs new—higher level—semiflows has been defined. *Symbolic* P-semiflows and T-semiflows are semiflows that assign weights to groups of colours—termed *symbolic colours*—within the colour domain of places and transitions, respectively. *Static* P-semiflows are defined as a generalisation of the concepts of symbolic P-semiflows. They are based on the concept of *static colour domain*, which defines a higher level of colour grouping.

For the compositional construction of the state space of a *cWN* we have introduced the concept of the *Composed Reachability Graph* (CRG), based on the concepts of *composed markings*, *composed transition instances* and *composed firing*. A composed marking will represent the group of symbolic markings of a *cWN* that can be produced by the combination of symbolic markings of its sub-components (one of each). Similarly a composed transition instance will represent a group of symbolic transition instances. The CRG of a *cWN* can, in principle, contain several SRGs of the *cWN*. The method proposed for the construction of the CRG uses, whenever possible, knowledge about the CRGs and/or SRGs of its the sub-components and about the composition operations employed for the construction of the *cWN*. We have studied the relationship between the SRG(s) of a *cWN* system and its CRG and have proved that the state space analysis over composed markings allows the verification of state space properties of the system, such as reachability, absence of deadlock and liveness.

To consolidate the understanding of the methods and concepts introduced, in Chapter 7 we have applied the methods proposed to the modelling of a Flexible Manufacturing System.

In Chapter 8 we study the extension of *cWN*s to support the notion of time. Composable Stochastic Well-formed nets (*cSWN*s) are defined and we study the changes that have to be made to the compositional methods proposed in order to use them for the compositional construction of $cSWN$ systems. The incorporation of the notion of time in the methods proposed offers the basis for the definition of a complete method for the compositional construction and analysis of performance-oriented PN systems of parallel and distributed systems.

9.3 Evaluation of the methodology proposed

In Chapter 3 we have seen that the methods proposed to incorporate compositional features into PNs vary according to the underlying Petri net class employed, the characteristics of the (basic) components and the set of composition operations defined. These features are strongly influenced by the motivation and objectives for incorporating compositionality into PNs. The methodology proposed in this dissertation is in this sense no exception. The underlying PN class, the set of composition operations defined and the definition of a basic component have been determined by the objective of proposing a method for the compositional construction and analysis of performance-oriented PN systems modelling parallel and distributed systems.

The behavioural properties of PN systems are, in general, very hard to analyse due to the size of the net and of its state space. The approach taken by many studies that define PN components and composition operations has been to work on restricted classes of nets, which, in general, makes the task easier [BDC92]. They are based on the definition of restrictions over either the composition operations and/or over the structural characteristics of the components [ES90, ES91b, ES91a, Sou93, Sou91b, SM90, Sou91a, BG96, RTS96]. Most of these studies concentrate on the preservation of properties rather than on the verification of them. Many of the difficulties encountered when obtaining the methods proposed in this dissertation were due to the characteristics of the WN formalism and the variety and flexibility of the composition operations defined. Our objective was not the preservation of properties, but to reuse the information about the characteristics of the sub-components to obtain the characteristics of the component. Rather than identifying how the system can be decomposed into subnets that are known to have certain properties and restricting the way in which these subnets can be combined, we depart from the system's description and from there deduce what will constitute a sub-component of the system. The construction of the cWN is based on the structure of the system modelled and not on the structural characteristics of the sub-components. In the same way as Battiston et al. [BBCDC95] we have aimed at overcoming the difficulty of structuring nets in a way that they reflect the structure of the system modelled.

In Chapter 3 we presented the set of criteria that we would take into consideration for the definition of a method for the compositional construction and analysis of PN systems modelling parallel and distributed systems. Let us discuss how the methodology proposed meets these criteria.

Model-system relation : The model of the system and the way it is built should reflect the system's structure. *bWN*s have been defined to model the basic functions of a system. The method proposed encourages the use of a top/down approach for the design of the system combined with a bottom/up approach for its construction. The system can be divided into subsystems which can be modelled separately and then composed to form the overall model. The composition operations have been defined to model both asynchronous and synchronous communication between the system's components. It is possible to model parallel execution, cooperation, competition for resources and message passing, which are common features of parallel and distributed systems. With the use of WN as the basic formalism it is possible to take advantage and reflect the symmetrical behaviour of the system's parts.

The definition of a basic component : By defining basic components it is possible to construct models in a regular and progressive manner, starting from components that have a common characteristic. We have defined *bWN*s that, as stated in the previous point, are intended to represent the basic functions of the system modelled. *cWN* systems can be built starting from *bWN*s and using the composition operations defined. As shown in Chapter 5 the definition of a *bWN* also allows us to propose a method for the compositional construction of symbolic and static P-semiflows of a *cWN* starting from the symbolic and static P-semiflows of the *bWN*s.

The PN formalism : WNs allow the representation of the symmetries in the behaviour of the systems components. The well-known problem of state space explosion is mitigated in WNs by the definition of the SRG, based on the concept of symbolic markings, which works on groupings of ordinary markings. The method here proposed uses the idea of the SRG and builds on it, defining a new type of marking grouping, termed *composed markings*.

The extension of *cWN*s to *cSWN*s, presented in Chapter 8, permits the representation of timing characteristics of the system modelled in the *cWN* system. This makes it possible to obtain performance information of the system modelled.

Re-usability : Re-using the models and information about the sub-components to obtain the model and information about a component saves time and effort. This is the main contribution of this dissertation. The methods proposed are aimed at using, whenever possible, the information about the characteristics of the sub-components. Not only is it possible to re-use the cWN model of a part of the system as a component of several other cWN models, but the information about its structural characteristics can, in general, be used to obtain the structural characteristics of the cWN models it forms part of. Similarly, the state space of the cWN—in more limited way—can be re-used when changing the environment of the cWN.

In this work we combine the idea of structural composition of PN models with the idea of using, as much as possible, the structural and state space information about the sub-component to obtain the structural and state space information about a component. There is still much work that can be done to improve and complement the methodology proposed. In the following section we discuss the directions for further developments of this methodology and of future work in the area.

9.4 Future Work

9.4.1 Implementation of the methodology

The most immediate extension of this work is the implementation of the methods proposed. The compositional nature of these methods suggests the possibility of implementing them using parallel programming techniques. Already, Christensen and Petrucci [CP95] have mentioned the adequacy of using parallel programming to implement their method for the modular construction of the state space of a modular coloured Petri net. Caselli et al. [CCM95] analyse different methods for the state space exploration of GSPN models. An interesting area of work would be to study how these methods can be used for the construction of the CRG, exploiting the compositional structure of the system.

In the methodology proposed here not only can we implement the state space construction in parallel, but the same can be done with the compositional construction of semiflows. The P-semiflows of the cWNs being composed can be computed in parallel and then appropriately combined. Furthermore, within a *bWN* the computation of the symbolic P-semiflows relating pairs of entry and final places can also be carried out in parallel (see Algorithm 2 on page 107).

9.4.2 Extension of the set of composition operations

We have defined various restrictions over the characteristics of the components and the way they can be composed. A logical extension of this work would be to gradually relax these restrictions. However, we must take into account that some of the restrictions have been imposed to avoid the modelling of undesirable behaviours such as: starvation of timed transitions or the modelling of systems that take no time in *cSWN*s. Rather than relaxing all the restrictions imposed, an alternative would be to incorporate new composition operations that extend the expressive power of the *cWN*, or *cSWN*s to be more general, for modelling parallel and distributed systems.

The definition of the basic component and of the composition operations in *cSWN* do not offer direct support for the modelling of pre-emption, fork and join synchronisation or other types of cooperation—such as timed polite communication or timed synchronisation [Hil94b]—common in parallel and distributed systems. To maintain the compositional approach, we can define new composition operations to model these behaviours. In this way it is possible to follow the same pattern for the study of methods for the compositional construction and analysis of the *cSWN*s. In [RM96] the set of synchronisations proposed was based on the different types of synchronisations studied by Hillston in [Hil94b]. In this dissertation we have limited our study to a sub-set of these types of synchronisations.

9.4.3 Quantitative analysis of *cWN*s

The information on the performance of a system modelled by a PN is determined by the quantitative analysis of the PN system. In [CDFH90] it is proved that the Markov chain obtained from the SRG corresponds to an exact lumping[1] [KS60] of the Markov chain that would be obtained from the RG. Investigating the possibility that the CRG is a lumped version of a SRG would be an interesting topic for future work. This would improve the quantitative analysis of large symmetric and loosely coupled parallel and distributed systems. The proposal of efficient numerical techniques for the solution of the *cSWN* models, can also

[1] A partitioning of the states of a Markov chain is said to be *exactly lumpable* if for any two states within one partition the aggregate transition rates into the states of any other partition are the same; where the aggregate transition rate from a state into a partition corresponds to the sum of the transition rates from the domain state to every state in the image partition.

be approached by studying the numerical properties of sub-components and the influence of the composition operations over these properties. Based on this it is possible to study if and when, under the compositional method proposed, it is possible to obtain *cSWN* systems that have, for example, product-form solution and/or tensor product solution. The study of other solution techniques such as the definition of methods for the computation of bounds or approximation techniques are would also be interesting. An interesting study would be the combination of the method here presented with the method presented by Haddad and Moreaux in [HM95, HM96] on the analysis of SWN systems by combining aggregation and decomposition.

9.4.4 Study of other types of P- and T- flows

The methods for the compositional construction of P- and T- semiflows has been limited to the definition and construction of positive integer flows (semiflows). An interesting topic for future work would be to study and propose methods for the compositional construction of other types of flows such as integer and rational flows.

9.4.5 Combination with other PN based methodologies for the modelling of parallel and distributed systems

In Chapter 3 we reviewed some of the existing methodologies for the modelling of parallel and distributed systems in such a way that the model of the hardware can be modelled independently of the application model, and vice versa [DF96, Fer92, BDF95]. This is a very important feature since it allows us to analyse the same application over different hardware configurations, or different software applications over the same hardware. In Ferscha's method [Fer92] the consumption of resources is represented in a very simple manner that does not allow the representation of resources with complex behaviours. In the PRS methodology [DF96] it is possible to model resources with complex behaviour. However, the sub-components within each level are modelled as flat nets and the analysis techniques employed to obtain the characteristics of the system correspond to the conventional methods for flat PN systems. It would be interesting to study how the PRS methodology can be combined with the *cSWN* methodology to be able to build the sub-components at the different levels in a compositional manner and analyse the structural and state space characteristics of the system in the same way. This would require the definition of new composition operations such as pre-emption, fork and join synchronisation and synchronisation of untimed events, amongst others, as mentioned in Section 9.4.2.

9.4.6 Combination with transformation techniques

The combination of transformation techniques with the compositional construction of cWN systems would allow both the refinement or abstraction of models, and the compositional combination of sub-models [Che91, ES91b, Ber87]. As we have seen, this requires the definition of equivalence relations (for transformations) and composition operations (for composition).

Appendix A

A.1 Compositional construction of symbolic P-semiflows under choice composition

Algorithm 1.4.- **(Obtaining the generative family $\mathcal{J}$ of symbolic P-semiflows of $\mathcal{N}$ built from a choice composition)**

/* Given the generative families of P-semiflows of the $cSWN$s L and R, G_L and G_R, respectively, it obtains the generative family of P-semiflows of $\mathcal{N}$, the $cSWN$ resulting from the choice composition of L and R */
{

- Let $\mathcal{J} = \emptyset$;
- Let $AUX_L = \emptyset$ and $AUX_R = \emptyset$

/ AUX_L and AUX_R will contain the P-semiflows whose sum of the entries for the places participating in the choice is greater than zero */*

For all $v \in G_L$ do
- Obtain $v_c = \sum\limits_{p \in Choice(L)} v(p)$
- If $v_c = 0$ then
  - Obtain the choice extention $\widehat{v}_c^R$ of v
  - Let $\mathcal{J} = \mathcal{J} \cup \{\widehat{v}_c^R\}$

else
  - Include v in the set of P-semiflows of L to be combined

$$AUX_L = AUX_L \cup \{(v, v_c)\}$$

end if

end do

For all $w \in G_R$ do
- Obtain $w_c = \sum\limits_{p \in Choice(R)} w(p)$
- If $w_c = 0$ then
  - Obtain the choice extention $\widehat{w}_c^L$ of w
  - Let $\mathcal{J} = \mathcal{J} \cup \{\widehat{w}_c^L\}$

else
  - Include w in the set of P-semiflows of R to be combined

$$AUX_R = AUX_R \cup \{(w, w_c)\}$$

end if

end do

For all $(v, v_c) \in AUX_L$ do

For all $(w, w_c) \in AUX_R$ do

- Obtain:

$$mult_v = lcm(v_c, w_c)/v_c$$
$$mult_w = lcm(v_c, w_c)/w_c$$

- Generate the P-semiflow u of $\mathcal{N}$:

$$\forall p \in P_{\mathcal{N}}$$
$$u(p) = \begin{cases} mult_v \cdot v(p) & \text{if } p \in P_L \\ mult_w \cdot w(p) & \text{if } p \in P_R \\ lcm(v_c, w_c) & \text{otherwise } (p = p_c) \end{cases}$$

- Let $\mathcal{J} = \mathcal{J} \cup \{u\}$

end do

end do

}

Bibliography

[AMBC82] M. Ajmone Marsan, G. Balbo, and G. Conte. Comparative performance analysis of single bus multiprocessor architectures. *IEEE Transactions on Computers*, C-31(12), December 1982.

[AMBC84] M. Ajmone Marsan, G. Balbo, and G. Conte. A class of generalized stochastic Petri nets for the performance analysis of multiprocessor systems. *ACM Transactions on Computer Systems*, 2(1), May 1984.

[AMBC+95] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modeling with Generalized Stochastic Petri Nets*. John Wiley & Sons Ltd., 1995.

[BB89] T. Bolognesi and E. Brinksma. Introduction to the ISO specification language LOTOS. In C.A. Vissers P.H.J. van Eijk and M. Diaz, editors, *The Formal Description Technique LOTOS*, pages 23–73, Amsterdam, 1989. North-Holland.

[BB93] M. Barbeau and G. Bochmann. A subset of LOTOS with the computational power of place/transition-nets. In *Proceedings of the 14th International Conference on Applications and Theory of Petri Nets*, volume LNCS 691, pages 49–68, Illinois, U.S.A., June 1993. Springer-Verlag.

[BBCDC95] E. Battiston, O. Botti, E. Crivelli, and F. De Cindio. An incremental specification of a hydroelectric power plant control system using a class of modular algebraic nets. In G. De Michelis and M. Diaz, editors, *Proceedings of the* 16^{th} *International conference Application and Theory of Petri Nets 1995*, volume LNCS 935, pages 84–102, Torino, Italy, June 1995. Springer.

[BBG95] M. Bernardo, N. Busi, and R. Gorrieri. A distributed semantics for EMPA based on stochastic contextual nets. *The Computer Journal*, 38(7):492–509, 1995.

[BDC92] L. Bernardinello and F. De Cindio. A survey of basic net models and modular net classes. *Advances in Petri Nets 1992*, LNCS 609:304–351, 1992.

[BDC93] O. Botti and F. De Cindio. Process and Resource boxes: an integrated Petri net performance model for applications and architectures. In *Proceedings of the International Conference on Systems, Man and Cybernetics*, Le Toquet, France, 1993. IEEE.

[BDF95] O. Botti, S. Donatelli, and G. Franceschinis. SWN models of parallel architectures : an application in the field of plant automation systems. In *Case Studies, within the* 16^{th} *International conference Application and Theory of Petri Nets 1995*, volume LNCS 935, pages 82–112, Torino, Italy, June 1995.

[BDH92] E. Best, R. Devillers, and J.G. Hall. The Box Calculus: a new causal algebra with multi-level communication. *Advances in Petri Nets 1992*, LNCS 609, 1992.

[Ber86] G. Bertherlot. Improving the linearly characterization of P/T nets. *Advances in Petri Nets 1985*, LNCS 222:19–40, 1986.

[Ber87] G. Berthelot. Transformations and decompositions of nets. *Petri nets: Central Models and Their Properties, Advances in Petri Nets 1986, Part I, Proceedings of an Advanced Course, Bad Honnef, Sep. 1986*, LNCS 254:359–376, 1987.

[BFF+95] E. Best, H. Fleishhack, W. Fraczak, R. Hopkins, H. Klaudel, and E. Pelz. A class of composable high level Petri nets with an application to the semantics of $B(PN)^2$. In G. De Michelis and M. Diaz, editors, *Proceedings of the* 16^{th} *International conference Application and Theory of Petri Nets 1995*, volume LNCS 935, pages 103–120, Torino, Italy, June 1995. Springer.

[BG96] M. Benalycherif and G. Girault. Behavioural and structural composition rules preserving liveness by synchronisation for coloured FIFO nets. In J. Billington and W. Reisig, editors, *Proceedings of* *the* 17^{th} *Application and Theory of Petri Nets 1996*, volume LNCS 1091, pages 73–92, Osaka, Japan, June 1996. Springer.

[BK85] J.A Bergstra and J.W. Klop. Algebra for Communicating Processes with Abstraction. *Journal of Theoretical Computer Science*, 37:77–121, 1985.

[BS92] M. Broy and T. Streicher. Modular functional modelling of Petri nets with individual tokens. *Advances in Petri Nets 1992*, LNCS 609:70–88, 1992.

[Buc92] P. Buchholz. A hierarchical view of GCSPNs and its impact on qualitative and quantitative analysis. *Journal of Parallel and Distributed Computing*, 15:207–224, July 1992.

[Buc93] P. Buchholz. Aggregation and reduction techniques for hierarchical GCSPNs. In *Proceedings of the 5th International Workshop on Petri Nets and Performance Models*, Toulouse, France, October 1993. IEEE Computer Society Press.

[Buc94a] P. Buchholz. Hierarchical high level Petri nets for complex system analysis. In R. Valette, editor, *Proceedings of the* 15^{th} *International Conference on Applications and Theory of Petri Nets*, volume LNCS 815, pages 119–138, Zaragoza, Spain, June 1994. Springer-Verlag.

[Buc94b] P. Buchholz. Markovian process algebra: Composition and equivalence. In U. Herzog and M. Rettelbach, editors, *Proceedings of the* 2^{nd} *Workshop on Process Algebras and Performance Modelling*, volume Bd. 27, Nr. 4, pages 11–30, Erlangen, Germany, July 1994.

[Buc95] P. Buchholz. A notion of equivalence for stochastic Petri nets. In G. De Michelis and M. Diaz, editors, *Proceedings of the* 16^{th} *International conference Application and Theory of Petri Nets 1995*, volume LNCS 935, pages 26–30, Torino, Italy, June 1995. Springer.

[CCM95] S. Caselli, G. Conte, and P. Marenzoni. Parallel state space exploration of GSPN models. In G. De Michelis and M. Diaz, editors, *Proceedings of the 16th International Conference on Applications and Theory of Petri Nets*, volume LNCS 935, pages 181–200, Torino, Italy, June 1995. Springer.

[CDF91] G. Chiola, S. Donatelli, and G. Franceschinis. GSPNs versus SPNs: What is the actual role of immediate transitions? In *Proceedings from the 4th International Workshop on Petri Nets and Performance Models*, pages 20–31, Melbourne, Australia, December 1991. IEEE Computer Society Press.

[CDFH90] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. On well-formed coloured nets and their symbolic reachability graph. In *11th International Conference on Application and Theory of Petri Nets*, Paris, France, June 1990. Springer-Verlag.

[CDFH91] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. Stochastic well-formed coloured nets and multiprocessor modelling applications. *High-Level Petri Nets, Theory and Application*, 1991.

[CDFH93] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. Stochastic well-formed coloured nets and symmetric modeling applications. *IEEE Transactions on Computers*, 42:1343–1360, November 1993.

[CDFH97] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. A symbolic reachability graph for coloured Petri net. *Theoretical Computer Science (Logic, Semantics and Theory of Programming)*, 176(1–2):39–65, apr 1997.

[CH93] S. Christensen and N. D. Hansen. Coloured Petri nets extended with place capacities, test arcs and inhibitor arcs. In M. Ajmone Marsan, editor, *Proceedings of the 14th International Conference on Application and Theory of Petri Nets*, pages 186–223, Chicago, Illinois, USA, June 1993. Springer-Verlag.

[Che91] G. Chehaibar. Use of reentrant nets in modular analysis of coloured nets. *Advances in Petri Nets 1991*, LNCS 524:58–77, 1991.

[CHP93] J.M. Couvreur, S. Haddad, and J.F. Peyre. Generative families of positive invariants in coloured nets sub-classes. *Advances in Petri Nets 1993*, LNCS 674, 1993.

[Cou90] J.M. Couvreur. The general computations of flows for coloured nets. In *Proceedings of the 11th International Workshop on Applications and Theory of Petri Nets*, Paris, France, June 1990.

[CP92] S. Christensen and L. Petrucci. Towards a modular analysis of coloured Petri nets. In K. Jensen, editor, *Proceedings of the 13th International Conference on Applications and Theory of Petri Nets*, volume LNCS 616, Sheffield, U.K., June 1992. Springer-Verlag.

[CP95] S. Christensen and L. Petrucci. Modular state space analysis of coloured Petri nets. In G. De Michelis and M. Diaz, editors, *Proceedings of the 16th International Conference on Applications and Theory of Petri Nets*, volume LNCS 935, pages 201–217, Torino, Italy, June 1995. Springer.

[CS90] J.M. Colom and M. Silva. Convex geometry and semiflows in P/T nets. A comparative study. *Advances in Petri Nets 1990*, LNCS 483, 1990.

[CT91] G. Ciardo and K. Trivedi. A decomposition approach for stochastic Petri net models. In *Proceedings from the 4th International Workshop on Petri Nets and Performance Models*, pages 74–83, Melbourne, Australia, December 1991. IEEE Computer Society Press.

[DCDMPS81] F. De Cindio, G. De Michelis, L. Pomello, and C. Simone. Superposed automata nets. In *Proceedings of the 2nd European Workshop on Applications and Theory of Petri Nets*, pages 269–279, Bad Honnef, Germany, September 1981.

[DF96] S. Donatelli and G. Franceschinis. The PRS methodology: integrating hardware and software models. In J. Billington and W. Reisig, editors, *Proceedings of the* 17^{th} *International Conference on Application and Theory of Petri nets 1996*, volume LNCS 1091, pages 133–152, Osaka, Japan, June 1996. Springer.

[Don93] S. Donatelli. Superposed stochastic automata: a class of stochastic Petri nets with parallel solution and distributed state space. *Performance Evaluation*, 18:21–36, 1993.

[Don94] S. Donatelli. Superposed generalised stochastic Petri nets: definition and efficient solution. In R. Valette, editor, *Proceedings of the* 15^{th} *International Conference on Applications and Theory of Petri Nets*, volume LNCS 815, pages 20–24, Zaragoza, Spain, June 1994. Springer-Verlag.

[ES90] J. Esparza and M. Silva. On the analysis and synthesis of free choice systems. *Advances in Petri Nets 1990*, LNCS 483:243–286, 1990.

[ES91a] J. Esparza and M. Silva. Compositional synthesis of live and bounded free choice nets. In J.C.M. Baeten and J.F. Groote, editors, *Proceedings of the 2nd International Conference on Concurrency Theory, CONCUR91*, volume LNCS 527, pages 118–139, Amsterdam, The Netherlands, August 1991. Springer-Verlag.

[ES91b] J. Esparza and M. Silva. Top-down synthesis of live and bounded free choice nets. *Advances in Petri Nets 1991*, LNCS 524:118–139, 1991.

[Fer92] A. Ferscha. A Petri net approach for performance oriented parallel program design. *Journal of Parallel and Distributed Computing*, 15:188–206, July 1992.

[Fin92] G. Findlow. Obtaining deadlock-preserving skeletons for coloured nets. In K. Jensen, editor, *Proceedings from the 13th International Conference on Application and Theory of Petri Nets*, volume LNCS 616, pages 173–192, Sheffield, UK, June 1992.

[GHR93] N. Götz, U. Herzog, and M. Rettelbach. TIPP - introduction and application to protocol performance analysis. Technical report, University of Erlangen, March 1993.

[GL83] H.J. Genrich and K. Lautenbach. S-invariance in predicate transition nets. *Informatik Fachberichte 66: Application and Theory of Petri Nets*, pages 98–111, 1983.

[Hac72] M. Hack. Analysis of production schemata by Petri nets. Technical report, MIT, Boston, 1972.

[HC88] S. Haddad and J.M. Couvreur. Towards a general and powerful computation of flows for parametrized coloured nets. In *Proceedings of the 9th European Workshop on Applications and Theory of Petri Nets*, volume vol. II, Venice, Italy, June 1988.

[HG87] S. Haddad and C. Girault. Algebraic structure of flows of a regular coloured net. *Advances in Petri Nets 1987*, LNCS 266:73–88, 1987.

[HHB92] R. Hopkins, J. Hall, and O. Botti. A basic-net algebra for program semantics and its application to OCCAM. *Advances in Petri Nets 1992*, LNCS 609, 1992.

[HHMR97] H. Hermanns, U. Herzog, V. Mertsiotakis, and M. Rettelbach. Exploiting stochastic process algebra achievements for generalised stochastic Petri nets. In *Proceedings of the 7^{th} International Workshop on Petri nets and Performace Models 1997*, Saint Malo, France, June 1997. IEEE.

[Hil94a] J. Hillston. A compositional approach to performance modelling. PhD Thesis CST-107-94, Department of Computer Science, University of Edinburgh, April 1994.

[Hil94b] J. Hillston. The nature of synchronisation. In U. Herzog and M. Rettelbach, editors, *Proceedings of the 2^{nd} Workshop on Process Algebras and Performance Modelling*, volume Bd. 27, Nr. 4, pages 51–70, Erlangen, Germany, July 1994.

[HM95] S. Haddad and P. Moreaux. Evaluation of high-level Petri nets by means of aggregation and decomposition. In *Proceedings of the 6^{th} International Workshop on Petri Nets and Performance models*, pages 11–20, Durham, North Carolina, U.S.A., October 1995. IEEE Computer Society Press.

[HM96] S. Haddad and P. Moreaux. Asynchronous composition of high level Petri nets: A queantitative approach. In J. Billington and W. Reisig, editors, *Proceedings of the 17^{th} Application and Theory of Petri Nets 1996*, volume LNCS 1091, pages 192–211, Osaka, Japan, June 1996. Springer.

[Hoa85] C.A.R. Hoare. *Communicating Sequential Processes.* Prentice-Hall, 1985.

[HR96] H. Hermanns and M. Rettelbach. Towards a superset of Basic LOTOS for performance prediction. In M. Ribaudo, editor, *Proceedings of the 4^{th} Workshop on Process Algebras and Performance Modelling*, pages 77–94, Torino, Italy, July 1996. C.L.U.T.

[Jen92] K. Jensen. *Coloured Petri Nets.* Springer-Verlag, 1992.

[KMF90] J. Kramer, J. Magee, and A. Finkelstein. A constructive approach to the design of distributed systems. In *Proceedings of the 10th International Conference on Distributed Computing Systems*, Paris, France, May 1990. IEEE Computer Society Press.

[Kot78] V. E. Kotov. An algebra for parallelism based on Petri nets. *Mathematical Foundations of Computer Science*, LNCS 64, 1978.

[KS60] J.G. Kemeny and J.L. Snell. *Finite Markov chains*. Van Nostrand, Princeton, NJ, U.S.A., 1960.

[M$\ddot{8}$5] K. Müller. Constructable Petri nets. In *Elektr. Informationsverarbeitung und Kybernetik EIK 21, 4/5*, pages 171–199, 1985.

[Mil89] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[Mol82] M. K. Molloy. Performance analysis using stochastic Petri nets. *IEEE Transactions on Computers*, C-31(9):913–917, September 1982.

[NM94] M. Nomoti and T. Murata. Hierarchical reachability graph of bounded Petri nets for concurrent-software analysis. *IEEE Transactions on Software Engineering*, 20(5):325–337, May 1994.

[Pet81] J.L. Peterson. *Petri Net Theory and The Modeling of Systems*. Prentice-Hall, Englewood Cliffs, NJ., 1981.

[Rei82] W. Reisig. Deterministic buffer synchronization of sequential processes. *Acta Informatica*, 18:117:134, 1982.

[Rei85] W. Reisig. *Petri Nets, An Introduction*. Springer-Verlag, 1985.

[Rei86] W. Reisig. Place/Transition systems. *Petri nets: Applications and Relationships to Other Models of Concurrency; Advances in Petri Nets 1986, Part II*, LNCS 255:117–141, 1986.

[Rib95a] M. Ribaudo. On the relationship between stochastic Petri nets and stochastic process algebras. Dottorato di ricerca in informatica, Dipartamento di Informatica, Universita di Torino, May 1995.

[Rib95b] M. Ribaudo. Stochastic Petri net semantics for stochastic process algebras. In *Proceedings of the* 6^{th} *International Workshop on Petri nets and Performance models*, pages 148–157, Durham, North Carolina, USA, October 1995. IEEE Computer Society Press.

[RM96] I. Rojas M. Compositional construction of SWN models. Internal Report . CSG Report Series. ECS-CSG-21-96, Dept. of Computer Science. University of Edinburgh, January 1996.

[Roj95] I. Rojas. Compositional construction of SWN models. *The Computer Journal*, 38(7):612–621, 1995.

[RTS95] L. Recalde, E. Teruel, and M. Silva. On well-formedness analysis: The case of deterministic systems of sequential processes. In J. Desel, editor, *Proc. of the Int. Workshop on Structures in Concurrency Theory (STRICT)*, Workshops in Computing, pages 279–293. Springer, 1995.

[RTS96] L. Recalde, E. Teruel, and M. Silva. $\{SC\}^*$ECS: A class of modular and hierarchical cooperating systems. In J. Billington and W. Reisig, editors, *Proceedings of the 17^{th} International Conference on Application and Theory of Petri nets 1996*, volume LNCS 1091, pages 440–459. Springer, June 1996.

[SB88] Y. Souissi and N. Beldiceanu. Deterministic systems of sequential processes: theory and tools. *Concurrency 88*, LNCS 335:380–400, 1988.

[SB93] C. Sibertin-Blanc. A client-server protocol for the composition of Petri nets. In *Proceedings of the 14th International Conference on Applications and Theory of Petri Nets*, pages 377–396, Illinois, U.S.A., June 1993. Springer-Verlag.

[Sil85] M. Silva. *Las redes de Petri en la automática y la informática.* Editorial AC, Madrid, España, 1985.

[SM90] Y. Souissi and G. Memmi. Composition of nets via a communication medium. *Advances in Petri Nets 1990*, LNCS 483:457–470, 1990.

[SM91] W. Sanders and J. Meyer. Reduced base model construction methods for stochastic activity networks. *IEEE Journal on Selected Areas in Communication*, 9:25–36, January 1991.

[Sou91a] Y. Souissi. A modular approach for the validation of communication protocols using FIFO nets. In *Proceeding of the X^{th} In-*

ternational Symposium on Protocol Specification, Verification and Testing. North Holland, June 1991.

[Sou91b] Y. Souissi. On liveness preservation by composition of nets via a set of places. *Advances in Petri Nets 1991*, LNCS 524:277–295, 1991.

[Sou93] Y. Souissi. Deterministic systems of sequential processes: a class of structured Petri nets. *Advances in Petri net 1993*, LNCS 674:406–426, 1993.

[Val90] A. Valmari. Compositional state space generation. In *11th International Conference on Application and Theory of Petri Nets*, Paris, France, June 1990. Springer-Verlag.

[Val93] A. Valmari. Compositional state space generation. *Advances in Petri Nets 1993*, LNCS 674:427–457, 1993.

[Val94] A. Valmari. Compositional analysis with place-bordered subnets. In R. Valette, editor, *Proceedings of the* 15^{th} *Application and Theory of Petri Nets 1994*, volume LNCS 815, pages 531–547, Zaragoza, Spain, June 1994. Springer-Verlag.

[Vau87] J. Vautherin. Calculation of semi-flows for Pr/T-systems. In *International Workshop on Petri Nets and Performance Models, Madison, Wisconsin, U.S.A.*, pages 174–183. IEEE Computer Society Press, 1987.

[VM84] J. Vautherin and G. Memmi. Computation of flows for unary predicate transition nets. *Advances in Petri Nets 1984*, LNCS 188, 1984.