

On Behavioural Synchronization in Net Systems

Hoang Chi Thanh
Department of Computer Science
Hanoi State University, Vietnam 10.000.
90 - Nguyen Trai, Dong Da, Ha noi, Vietnam

1. Introduction.

Large concurrent systems are difficult to design and analyse, because they can exhibit very complicated behaviours. So modular approaches have been used in investigating either the structure of models for concurrent [1, 5, 10] or their aspects [7], especially their semantics [6, 9].

To our knowledge V.E. Kotov was the first, who made the set of all Place/Transition nets become an algebra [5] by defining five operation on it. When using the decomposition method to find the behavioural function of finite (0,1) - marked nets [6]. Mazurkiewicz pointed out the behavioural synchronization in the term of traces for pairs of those nets, whose sets of places are disjoint.

Concurrent systems considered here are Net systems [8, 12]. Our purpose is to construct large systems out of smaller ones. The main requirement is to determine an explicit structure for their dynamic aspects. Based on the definition of synchronization of languages [2], we define a composition operation which makes not only the set of all net systems a commutative monoid but also the families of firing sequence languages generated by the net systems, closed under the respective synchronizations.

Hence, after having composed, the net system's behaviours, as well as independency relation and conflict relation can be built up from that of its immediate sub-components, without computing from beginning.

A special attention is paid to the family of contact - free net systems because this family is a kernel of all net systems in the sense that the family of firing sequence languages and trace languages generated by arbitrary net systems is not larger than that by contact - free net systems.

The paper is organized as follows. First, some basic notions and facts concerning net systems are given. Section 3 presents two basic behavioural representations for net systems: firing sequence

languages and trace languages and their relationship. The former is interpreted as an interleaving semantics and the latter as a non-interleaving one. The main theorems of this paper are contained in Section 4. They assert the synchronization for behaviours of contact - free net systems. Section 5 points out that these results still hold for the family of all net systems by extending the composition operation and using an equivalent transformation in [8]. Some concluding remarks are presented in the last section.

2. Net Systems.

Distributed systems usually have static and dynamic aspects. Net systems are one of sound models for representing these systems. In this section we introduce basic notions and notations used throughout the paper and formulate some useful facts.

A triple $N = (B, E, F)$ is called a net iff:

- B and E are disjoint sets,
- $F \subseteq (B \times E) \cup (E \times B)$ is a binary relation called the flow relation of N , such that:

$$\text{domain}(F) \cup \text{range}(F) = B \cup E.$$

The elements of B are called conditions and the elements of E are called events. The flow relation models a fixed "neighbourhood" relation between the conditions and events of a system. In the graphic representation, the conditions will be drawn as circles, the events as boxes and the elements of the flow relation as directed arcs.

Let $N = (B, E, F)$ be a net. Then, $X_N = B \cup E$ is the set of elements of N . For every $x \in X_N$:

$\cdot x = \{y \mid (y, x) \in F\}$ is called the pre-set of x ,

$x \cdot = \{y \mid (x, y) \in F\}$ is called the post-set of x .

A pair $(p, e) \in B \times E$ is called a *self-loop* iff $(p, e) \in F$ & $(e, p) \in F$. N is called *pure* iff F does not contain any self-loop.

A net N is called *simple* iff its two distinct elements do not have the same pre-and post-set, i.e.

$$\cdot x = y \wedge x \cdot = y \text{ implies } x = y \text{ for every } x, y \in X_N.$$

Let $N = (B, E, F)$ be a simple net. A subset $c \subseteq B$ is called a *case*. Let $e \in E$ and $c \subseteq B$. Then, e is said to be enabled at c (c -enabled, for short) iff $e \subseteq c \& e \cdot c = \emptyset$. We denote: $c[e > \cdot$. Let $e \in E$, $c \subseteq B$ and e be c -enabled. Then $d = (c - \cdot e) \cup e \cdot$ is called the reachable case from the occurrence of e in the case c , and we write: $c[e > d$. If $c[e_1 > c_1[e_2 > \dots c_n[e_{n+1} > d$, we shall write $c[[e_1 e_2 \dots e_{n+1} > d$. So we adopt the following definition of the reachability of N :

The *reachability relation* of N is the relation $R_N = (R1 \cup R1^{-1})^*$, where $R1 \subseteq \mathcal{P}(B) \times \mathcal{P}(B)$ is called the forward reachability in one step, given by:

$$(c, d) \in R1 \iff \exists e \in E, c[e > d.$$

Note that R_N is an equivalence relation. A *net system* we mean any quadruple: $\mathcal{N} = (B, E, F, \mathcal{C})$, where:

- (1) $N = (B, E, F)$ is a simple net, called *the underlying net* of \mathcal{N} ,
- (2) $\mathcal{C} \subseteq \mathcal{P}(B)$ is an union of some equivalence classes of R_N , such that:
 - $\forall e \in E, \exists c \in \mathcal{C}$, then e is c -enabled.

C is called *the state space* of \mathcal{N} . The state space reflects a transition system associated with the net system. An equivalence class of R_N is called an orbit. So the state space consists of one or more orbits.

Lemma 2.1 For every net system $\mathcal{N} = (B, E, F, C)$:

- 1) The underlying net N is a pure net.
- 2) The state space C is a covering of B .

Proof: 1) By the definition of net systems we have $(\forall e \in E, \exists c \in C), \cdot e \subseteq c \& c \cap e' = \emptyset$. This means $\forall e \in E, \cdot e \cap e' = \emptyset$. So N is pure.

2) We have to show that: $(\forall p \in B)(\exists c \in C), p \in c$.

Let $p \in B$. Since $domain(F) \cup range(F) = B \cup E, \exists e \in E$ such that $(p, e) \in F$ or $(e, p) \in F$. So $p \in \cdot e$ or $p \in e'$. By the definition of net systems, $\exists c \in C, e$ is c -enabled. Thus, $p \in \cdot e \subseteq c$ or $p \in e' \subseteq d = (c - e) \cup e' \in C$. •

Note that we admit the empty net system $\mathcal{N}_\emptyset = (\emptyset, \emptyset, \emptyset, \emptyset)$.

Example 2.1 Let $N = (B, E, F)$ be the simple net shown in Figure 1,

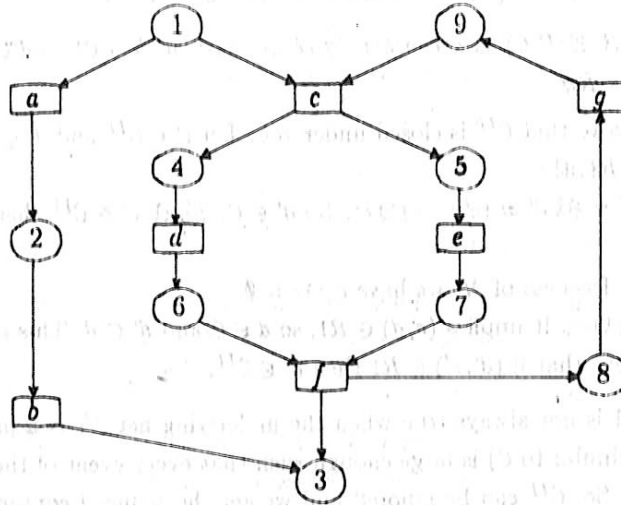


Fig. 1

and $C = \{\{1, 9\}, \{2, 9\}, \{3, 9\}, \{1, 8\}, \{2, 8\}, \{3, 8\}, \{4, 5\}, \{4, 7\}, \{5, 6\}, \{6, 7\}\}$.

The quadruple $\mathcal{N} = (B, E, F, C)$ is a net system.

Let $\mathcal{N} = (B, E, F, C)$ be a net system.

The net system \mathcal{N} is said to be *contact - free* iff for each $e \in E$ and for each $c \in \mathcal{C}$:

- $e \subseteq c \Rightarrow e \cap c = \emptyset$ and
- $e' \subseteq c \Rightarrow e' \cap c = \emptyset$.

Thus, in a contact - free net system the occurrence of an event in a case ensures that the event is enabled at the case. For example, the net system given in Example 2.1 is contact - free. We have pointed out in [10] that the above definition is equivalent to the definition of the safeness presented in [3]. In the major part of this paper we will pay our attention to the family of contact - free net systems. We show that from a contact - free net system we can get a reduced net system, which has no "redundant" cases. To do so, we introduce some new notions.

Let $\mathcal{N}_1 = (B, E, F, \mathcal{C}_1)$ and $\mathcal{N}_2 = (B, E, F, \mathcal{C}_2)$ be net systems having a common underlying net and holding the following condition: $(\forall c_1 \in \mathcal{C}_1)(\exists c_2 \in \mathcal{C}_2), c_1 \subseteq c_2$.

Then we shall write: $\mathcal{N}_1 \preceq \mathcal{N}_2$. In this case the contact - freeness of the "smaller" net system follows naturally from that of the "greater" one:

Theorem 2.1 *Let $\mathcal{N}_1, \mathcal{N}_2$ be net systems and $\mathcal{N}_1 \preceq \mathcal{N}_2$. Then:*

$$\mathcal{N}_2 \text{ is contact - free} \Rightarrow \mathcal{N}_1 \text{ is contact - free.}$$

Consider a net system $\mathcal{N} = (B, E, F, \mathcal{C})$. If its state space \mathcal{C} contains cases, which are proper subsets of other cases, then the state space can be divided into two disjoint parts as follows:

$$\mathcal{C}^{II} = \{c' \mid c' \in \mathcal{C} \& \exists c \in \mathcal{C}, c' \subset c\}, \mathcal{C}^I = \mathcal{C} - \mathcal{C}^{II}.$$

Theorem 2.2 *If $\mathcal{N} = (B, E, F, \mathcal{C})$ is a contact - free net system then \mathcal{C}^I and \mathcal{C}^{II} are closed under the reachability relation $R_{\mathcal{N}}$.*

Proof: It suffices to prove that \mathcal{C}^{II} is closed under $R_{\mathcal{N}}$. Let $c' \in \mathcal{C}^{II}$ and $d' \subseteq B$.

Assume that $(c', d') \in R_1$, thus

$\exists e \in E, e \subseteq c' \& e' \cap c' = \emptyset \& d' = (c' - e) \cup e'$. So $d' \in \mathcal{C}$. Since $c' \in \mathcal{C}^{II}$ then $\exists c \in \mathcal{C}, c' \subset c$ and $e \subseteq c' \Rightarrow e \subseteq c$.

Applying the contact - freeness of \mathcal{N} , we have $e' \cap c = \emptyset$.

Let us put $d = (c - e) \cup e'$. It implies $(c, d) \in R_1$, so $d \in \mathcal{C}$ and $d' \subset d$. This means $d' \in \mathcal{C}^{II}$. In a similar way we can prove that if $(d', c') \in R_1$ then $d' \in \mathcal{C}^{II}$. •

Note that Theorem 2.1 is not always true when the underlying net \mathcal{N}_1 is a proper subnet of \mathcal{N}_2 . It is obvious that \mathcal{C}^I (similar to \mathcal{C}) is large enough such that every event of the system is enabled, while \mathcal{C}^{II} may be not. So, \mathcal{C}^{II} can be ignored and we get the reduced contact - free net system: $\mathcal{N}^M = (B, E, F, \mathcal{C}^I)$ from \mathcal{N} .

This fact will be used in Section 4 for composing contact - free net systems.

3. Behaviours of Net Systems

The most primitive behavioural representation of net systems is set of firing sequences. One can show that it is a kernel for constituting other behaviours, e.g. traces or step sequences...

Let $\mathcal{N} = (B, E, F, \mathcal{C})$ be a net system and $\mathcal{N} = (B, E, F)$ its underlying net.

A relation $I \subset E \times E$ is said to be *the independency relation* of \mathcal{N} iff:

$$(e, f) \in I \iff (\cdot e \cup e \cdot) \cap (\cdot f \cup f \cdot) = \emptyset.$$

Note that I is a symmetric and irreflexive relation (a sir-relation). The independency relation describes that in a distributed system two actions are independent iff they do not share any resources.

$D = E \times E - I$ is called *the dependency relation* of \mathcal{N} .

Let $\alpha = e_1 e_2 \dots e_k \in E^*$, ($k \geq 0$). α is called a *firing sequence* of \mathcal{N} iff there exist $c_1, c_2, \dots, c_{k+1} \in \mathcal{C}$ such that $c_1[e_1 > c_2[e_2 > c_3 \dots c_k[e_k > c_{k+1}$.

The *language* of \mathcal{N} , denoted by $FS(\mathcal{N})$ is the set of all firing sequences of \mathcal{N} . Note that, we always may assume that $c[\epsilon > c$, where c is a case of \mathcal{N} , ϵ is the empty sequence of E^* . So $\epsilon \in FS(\mathcal{N})$, for every \mathcal{N} .

It is a well-known fact that the language generated by a net system is regular and closed under the In - operation, i.e. $FS(\mathcal{N}) = In(FS(\mathcal{N}))$, where for every alphabet A , and for every language $L \subseteq A^*$:

$$In(L) = \{x | x \in A^* \& \exists u, v \in A^*, uv \in L\}.$$

By the definition of the language generated by a net system and the construction of the reduced net system presented above, we have:

Corollary 3.1 *For every contact - free net system: $FS(\mathcal{N}) = FS(\mathcal{N}^M)$.*

Thus, every contact - free net system can be replaced by a behaviourally equivalent contact - free net system without "redundant cases". In the rest of this paper, unless otherwise stated, a contact-free net system means a reduced contact - free net system.

Let $\mathcal{N} = (B, E, F, \mathcal{C})$ be a net system. Let $e_1, e_2 \in E, e_1 \neq e_2$ and $c \in \mathcal{C}$. We say that e_1 and e_2 can *occur concurrently* at c , denoted $c[\{e_1, e_2\} >$ iff $c[e_1 >, c[e_2 >$ and $(e_1, e_2) \in I$. And e_1, e_2 are said to be *in conflict* at c iff $c[e_1 >$ and $c[e_2 >$ but $(e_1, e_2) \notin I$ (see [12]).

We will point out that the concurrency and the conflict of two events can be "seen" from the language generated by a net system. Now we consider the structure of the language.

Theorem 3.1 *If $\alpha \in FS(\mathcal{N})$ and $\alpha = uefv$, where $u, v \in E^*, (e, f) \in I$ then $\beta = ufev \in FS(\mathcal{N})$.*

Proof: By the definition of the language $FS(\mathcal{N})$, there exist $c_1, c_2, c_3, c_4, c_5 \in \mathcal{C}$, such that $c_1[[u > c_2[e > c_3[f > c_4[[v > c_5$. It is enough to show that $c_2[[fe > c_4$. Since $c_2[e > c_3$ then $\cdot e \subseteq c_2, e \cdot \cap c_3 = \emptyset$ and $c_3 = (c_2 - \cdot e) \cup e \cdot$. Similarly, since $c_3[f > c_4$ we have $\cdot f \subseteq c_3, f \cdot \cap c_4 = \emptyset$ and $c_4 = (c_3 - \cdot f) \cap f \cdot$. Because $\cdot f \subseteq c_3 = (c_2 - \cdot e) \cup e \cdot$ and $(e, f) \in I$ so we get $\cdot f \subseteq c_2$. On other hand $f \cdot \cap c_3 = \emptyset$, i.e. $((c_2 - \cdot e) \cup e \cdot) \cap f \cdot = \emptyset$. This implies $f \cdot \cap c_2 = \emptyset$. Put $c'_3 = (c_2 - \cdot f) \cup f \cdot$, we have $c_2[f > c'_3$. In a similar way we can prove that $c'_3[f > c_4$. That means $c_2[[fe > c_4$.

From the proof of this theorem we can see that if $\alpha = uefv \in FS(\mathcal{N})$ and $(e, f) \in I$ then there exists a case $c \in \mathcal{C}$, such that $c[\{e, f\} >$. Otherwise, if $c[\{e, f\} >$ then there exists at least one firing sequence $\alpha = uefv$, for some $u, v \in E^*$, such that $\alpha \in FS(\mathcal{N})$. (Of course, $\beta = ufev \in FS(\mathcal{N})$).

So, two events are concurrent iff there exists a firing sequence, in which one of them "stands" immediately behind the other and they are independent. The case they can occur concurrently at is just a case, at which either of them is enabled. In other words, e and f can occur concurrently at some case in the net system \mathcal{N} iff $ef \in FS(\mathcal{N})$ and $(e, f) \in I$.

When e, f are in conflict then first, $(e, f) \notin I$ and either of them may occur but not both. So e, f are in conflict iff $(e, f) \notin I$ and there are two firing sequences $\alpha_1 = ueg, \alpha_2 = ufh \in FS(\mathcal{N})$, where $u, g, h \in E \cup \{\epsilon\}, e \neq h$ and $f \neq g$.

So we introduce the following relations:

A relation $co \subseteq E \times E$ is called *the concurrency relation* of \mathcal{N} iff:

$$(e, f) \in co \iff (\exists c \in \mathcal{C}), c[\{e, f\}] > \text{ in } \mathcal{N}.$$

A relation $cl \subseteq E \times E$ is called *the conflict relation* of \mathcal{N} iff:

$$(e, f) \in cl \iff (\exists c \in \mathcal{C}), (e, f) \text{ are in conflict at } c \text{ in } \mathcal{N}.$$

Clearly, for a net system, $co \subseteq I$ and $cl \subseteq D$.

Now we define the trace language of a net system.

A concurrent alphabet (A, D) consists of a finite set A of symbols and a reflexive and symmetric relation D , the dependency relation. Its complement $A \times A - D$, denoted by I , is called the independency relation, which is symmetric and irreflexive. Let $\sim_D \subseteq A^* \times A^*$ be the following relation:

$$x \sim_D y \iff (\exists e, f \in A)(\exists u, v \in A^*), (e, f) \in I \& x = uefv \& y = ufev.$$

Define $\approx = (\sim_D)^*$, i.e. \approx is the symmetric and transitive closure of \sim_D . Note that \approx is an equivalence relation.

Let $[\alpha]_D$ denote the equivalence class of \approx containing α . It is called *Mazurkiewicz trace* over D . The quotient algebra $(A^*, o, [\epsilon]_D) / \approx$, where o is the concatenation, is called a trace algebra. Denote $T_D = \{[\alpha]_D | \alpha \in A^*\}$, $P_D = \mathcal{P}(T_D)$.

Let $\mathcal{N} = (B, E, F, \mathcal{C})$ be a net system. Then $N = (B, E, F)$ is its underlying net. Let I be its independency relation and $D = E \times E - I$ its dependency relation.

As presented in [6], we recall the reachability relation of N in the term of traces as follows:

The reachability of N is the least function $\mathcal{R}_N : \mathcal{P}(B) \times \mathcal{P}(B) \rightarrow P_D$ (with respect to the inclusion ordering of its values), such that:

- (1) $[\epsilon]_D \in \mathcal{R}_N(c, d) \iff c = d$;
- (2) $[e]_D \in \mathcal{R}_N(e, d) \iff c[e > d, \text{ for } e \in E$;
- (3) $t_1 o t_2 \in \mathcal{R}_N(c, d) \iff \exists s \in \mathcal{P}(B), t_1 \in \mathcal{R}_N(c, s) \& t_2 \in \mathcal{R}_N(s, d), \text{ for } t_1, t_2 \in T_D$.

The set $\tau(\mathcal{N}) = \bigcup_{c, d \in \mathcal{C}} \mathcal{R}_N(c, d)$ is called *the trace language* generated by \mathcal{N} . By Theorem 3.1, we have:

$$\alpha \in FS(\mathcal{N}) \iff [\alpha]_D \in \tau(\mathcal{N}), \text{ for every } \alpha \in E^*.$$

Corollary 3.2 For every net system \mathcal{N} : $\tau(\mathcal{N}) = FS(\mathcal{N}) / \approx$.

A trace generated by a net system is indeed a collection of a number of firing sequences generated by the net system. So the firing sequence behaviour and the trace language behaviour of a net system have some common properties. We will show this in the next sections.

4. A Monoid of Contact - free Net Systems

Our main aim is to construct large concurrent systems out of smaller ones (especially, of atomic components). The construction is based on the synchronization of languages. So we recall some necessary notions.

Let A be an alphabet and ϵ denote the empty sequence. Given two alphabets A, B such that $B \subseteq A$. Let $h_B : A^* \rightarrow B^*$ be an erasing homomorphism given by:

$$\forall a \in A, h_B(a) = a \text{ if } a \in B \text{ and } \epsilon \text{ otherwise;}$$

and $\forall x \in A^*, h_B(xa) = h_B(x)h_B(a)$. Instead of $h_B(x)$ we shall write $x|_B$ (x projected on B). Similarly, for every language $L \subseteq A^*$ and the alphabet $B : L|_B = \{x|_B \mid x \in L\}$.

For a language $L \subseteq A^*$, let \bar{L} denote the least alphabet constituting L :

$$\bar{L} = \{a \in A \mid \exists u, v \in A^*, uav \in L\}.$$

For two languages L_1, L_2 , the language $L_1 \# L_2$ is called the *synchronization* of L_1 with L_2 , is defined as follows:

$$L_1 \# L_2 = \{x \mid x \in (\bar{L}_1 \cup \bar{L}_2)^* \& x|_{\bar{L}_1} \in L_1 \& x|_{\bar{L}_2} \in L_2\}.$$

The synchronization ensures that the occurrence orders and the number of occurrences of any symbol in its every sequence are the same as in the respective sequences from which this sequence has been constituted. So the operation $\#$ will play an important role in composing net systems.

Given two net systems $\mathcal{N}_1 = (B_1, E_1, F_1, C_1)$ and $\mathcal{N}_2 = (B_2, E_2, F_2, C_2)$. Without loss of generality, we can assume that the simpleness remains valid in both these net systems, i.e.:

$$(\forall x, y \in X_{\mathcal{N}_1} \cup X_{\mathcal{N}_2}), x = y \wedge x' = y' \Rightarrow x = y.$$

Let $FS(\mathcal{N}_1)$ and $FS(\mathcal{N}_2)$ denote the firing sequence languages of $\mathcal{N}_1, \mathcal{N}_2$ respectively, and $FS = FS(\mathcal{N}_1) \# FS(\mathcal{N}_2)$ - the synchronization of $FS(\mathcal{N}_1)$ with $FS(\mathcal{N}_2)$.

It is natural to ask *whether one can build up a net system \mathcal{N} from \mathcal{N}_1 and \mathcal{N}_2 such that its firing sequence language is FS* . The answer will be in the affirmative.

A similar result in the terms of trace languages was achieved by Mazurkiewicz in [6] for pairs of B - disjoint nets ($B_1 \cap B_2 = \emptyset$).

In this and next section, our approach is devoted to the general case. First, we answer the question for the family of contact - free net systems, and then for the general case by using an equivalent transformation on net systems.

Given two contact - free net systems $\mathcal{N}_i = (B_i, E_i, F_i, C_i), i = 1, 2$. Let us define:

$\mathcal{N} = (B, E, F, C)$, where:

$$B = B_1 \cup B_2,$$

$$E = E_1 \cup E_2,$$

$$F = F_1 \cup F_2,$$

$$\mathcal{C} = \{c_1 \cap (B_1 - B_2) \cup c_1 \cap c_2 \cup c_2 \cap (B_2 - B_1) \mid c_1 \in \mathcal{C}_1 \& c_2 \in \mathcal{C}_2\}.$$

Denote: $\mathcal{N} = \mathcal{N}_1 \oplus \mathcal{N}_2$. We show that \mathcal{N} is just a net system that satisfies the above requirement. Furthermore, the contact - freeness is preserved by this synthesis.

Theorem 4.1 *If \mathcal{N}_1 and \mathcal{N}_2 are contact - free net systems then \mathcal{N} is also a contact - free net system and $FS(\mathcal{N}) = FS(\mathcal{N}_1) \# FS(\mathcal{N}_2)$.*

Proof: 1) First of all, we prove that the quadruple \mathcal{N} is a net system. It is easy to see that $N := N_1 \cup N_2 = (B_1 \cup B_2, E_1 \cup E_2, F_1 \cup F_2)$ is a simple net.

a) We have to show that \mathcal{C} is closed under the reachability relation R_N of N .

Let $c \in \mathcal{C}$ and $d \in B$.

Assume that $(c, d) \in R1_N$, that means: $\exists e \in E, \cdot e \subseteq c \& \cdot e \cap c = \emptyset \& d = (c - \cdot e) \cup e$.

We introduce some following auxiliary notations. For each $e \in E$

$$\cdot e_0 = \cdot e \cap B_1 \cap B_2, \cdot e_1 = \cdot e \cap (B_1 - B_2), \cdot e_2 = \cdot e \cap (B_2 - B_1),$$

$$e_0 = e \cap B_1 \cap B_2, e_1 = e \cap (B_1 - B_2), e_2 = e \cap (B_2 - B_1).$$

Of course, $\cdot e = \cdot e_0 \cup \cdot e_1 \cup \cdot e_2$ and $e = e_0 \cup e_1 \cup e_2$. Since $c \in \mathcal{C}$ then there exist $c_1 \in \mathcal{C}_1 \& c_2 \in \mathcal{C}_2$, such that

$$c = c_1 \cap (B_1 - B_2) \cup c_1 \cap c_2 \cup c_2 \cap (B_2 - B_1).$$

Let us put: $c'_1 = c_1 \cap (B_1 - B_2) \cup c_1 \cap c_2$ and $c'_2 = c_2 \cap (B_2 - B_1) \cup c_1 \cap c_2$. Thus, $c = c'_1 \cup c'_2$ and $\cdot e_0 \cup \cdot e_i \subseteq c'_i$, $i = 1, 2$. (Here $\cdot e_0 \cup \cdot e_i$ and $e_0 \cup e_i$ are just $\cdot e$ and e respectively in the net N_i , for $i = 1, 2$). Due to the contact - freeness of \mathcal{N}_1 and \mathcal{N}_2 , we have:

$$\cdot e_0 \cup \cdot e_i \subseteq c'_i \subseteq c_i \in \mathcal{C}_i \Rightarrow (e_i \cup e_0) \cap c_i = \emptyset, \quad i = 1, 2.$$

So: $d_i = (c_i - (\cdot e_0 \cup \cdot e_i)) \cup (e_0 \cup e_i) \in \mathcal{C}_i$, $i = 1, 2$.

$$\begin{aligned} d_1 \cap (B_1 - B_2) &= ((c_1 - (\cdot e_0 \cup \cdot e_1)) \cup (e_0 \cup e_1)) \cap (B_1 - B_2) \\ &= ((c_1 - (\cdot e_0 \cup \cdot e_1)) \cap (B_1 - B_2)) \cup ((e_0 \cup e_1) \cap (B_1 - B_2)) \\ &= (c_1 \cap (B_1 - B_2) - \cdot e_1) \cup e_1. \end{aligned}$$

Analogously, we obtain:

$$d_2 \cap (B_2 - B_1) = (c_2 \cap (B_2 - B_1) - \cdot e_2) \cup e_2. \text{ Since}$$

$$d_i \cap (B_1 \cap B_2) = ((c_i - (\cdot e_0 \cup \cdot e_i)) \cup (e_0 \cup e_i)) \cap (B_1 \cap B_2)$$

$$= (c_i \cap B_1 \cap B_2 - \cdot e_0) \cup e_0, \text{ for } i = 1, 2, \text{ thus}$$

$$d_1 \cap d_2 = (d_1 \cap B_1 \cap B_2) \cap (d_2 \cap B_1 \cap B_2)$$

$$= (c_1 \cap c_2 - \cdot e_0) \cup e_0.$$

Applying the above equalities, we get:

$$d = (c - \cdot e) \cup e$$

$$= (c_1 \cap (B_1 - B_2) \cup c_1 \cap c_2 \cup c_2 \cap (B_2 - B_1) - (\cdot e_0 \cup \cdot e_1 \cup \cdot e_2)) \cup (e_0 \cup e_1 \cup e_2)$$

$$= ((c_1 \cap (B_1 - B_2) - \cdot e_1) \cup \cdot e_1) \cup ((c_1 \cap c_2 - \cdot e_0) \cup e_0) \cup ((c_2 \cap (B_2 - B_1) - \cdot e_2) \cup e_2)$$

$$= d_1 \cap (B_1 - B_2) \cup d_1 \cap d_2 \cup d_2 \cap (B_2 - B_1).$$

From the construction of \mathcal{C} , we have $d \in \mathcal{C}$.

In a similar way we can prove that if $(d, c) \in R1_N$ then $d \in C$.

b) We still have to show that: $\forall e \in E, \exists c \in C$ such that e is c -enabled.

Let $e \in E = E_1 \cup E_2$.

Assume that $e \in E_1 - E_2$, then $\exists c_1 \in C_1, c_1[e >]$. Hence $e \subseteq c_1 \cap (B_1 - B_2) \& e' \cap c_1 = \emptyset$. For some $c_2 \in C_2$, put $c = c_1 \cap (B_1 - B_2) \cup c_1 \cap c_2 \cup c_2 \cap (B_2 - B_1)$. Of course, $e \subseteq c \& e' \cap c = \emptyset$, i.e. $c[e >]$ in \mathcal{N} .

In the case when $e \in E_2 - E_1$, we can proceed similarly.

Now assume that $e \in E_1 \cap E_2$, then $\exists c_i \in C_i, c_i[e >]$ for $i = 1, 2$.

So $e_0 \cup e_i \subseteq c'_i$ and $(e_0 \cup e_i) \cap c_i = \emptyset$. Put $c = c_1 \cap (B_1 - B_2) \cup c_1 \cap c_2 \cup c_2 \cap (B_2 - B_1)$.

Hence $c \in C$ and we have: $e = e_1 \cup e_0 \cup e_2 \subseteq c'_1 \cup c'_2 = c$ and $e' \cap c = e' \cap (c'_1 \cup c'_2) = \emptyset$, i.e. $c[e >]$.

So the quadruple \mathcal{N} is a net system.

2) Now we prove that \mathcal{N} is contact-free.

Let $e \in E$ and $c \in C$ such that $e \subseteq c$.

Thus, there exist $c_1 \in C_1, c_2 \in C_2$ and $c = c_1 \cap (B_1 - B_2) \cup c_1 \cap c_2 \cup c_2 \cap (B_2 - B_1)$. By the contact-freeness of $\mathcal{N}_1, \mathcal{N}_2$ we have, for $i = 1, 2$, $(e_0 \cup e_i) \cap c_i = \emptyset$.

But $c'_i \subseteq c_i$, so $(e_0 \cup e_i) \cap c'_i = \emptyset$.

Hence $e' \cap c = e' \cap (c'_1 \cup c'_2) = (e_0 \cup e_1) \cap c'_1 \cup (e_0 \cup e_2) \cap c'_2 = \emptyset$.

In the case when $e \subseteq c$, the proof can be proceeded similarly.

3) To finish the proof, we have to show that, for each $\alpha \in E^*$:

$$\alpha \in FS(\mathcal{N}) \iff \alpha|_{E_i} \in FS(\mathcal{N}_i), i = 1, 2.$$

Alternatively, we may prove the following equivalent proposition:

(*) $c[[\alpha > d$ in \mathcal{N} , where

$$c = c_1 \cap (B_1 - B_2) \cup c_1 \cap c_2 \cup c_2 \cap (B_2 - B_1),$$

$$d = d_1 \cap (B_1 - B_2) \cup d_1 \cap d_2 \cup d_2 \cap (B_2 - B_1),$$

$$c_1, d_1 \in C_1 \& c_2, d_2 \in C_2$$

$$\iff c_i[[\alpha|_{E_i} > d_i \text{ in } \mathcal{N}_i, i = 1, 2. \quad (4.1)$$

We prove (4.1) by induction with respect to the length of firing sequences generated by the composed net system \mathcal{N} :

a) If $\alpha = \epsilon$ then this case is trivial.

b) If $\alpha = e$ and $e \in E$ then:

$$c[e > d \text{ in } \mathcal{N} \iff e \subseteq c \& e' \subseteq d \& c - e = d - e'.$$

Assume that $e \in E_1 - E_2$, then $e_1 = e, e_1 = e', e_0 = e_0 = e_2 = e_2 = \emptyset \& e \subseteq c_1, e' \subseteq d_1, c_1 - e = d_1 - e' \& c_2 = d_2$. So

$$(*) \iff c_1[e > d_1 \text{ in } \mathcal{N}_1 \& c_2[e > d_2 \text{ in } \mathcal{N}_2] \iff e \in FS(\mathcal{N}_1) \& e \in FS(\mathcal{N}_2).$$

In the case $e \in E_2 - E_1$, analogously we have $(*) \iff e \in FS(\mathcal{N}_1) \& e \in FS(\mathcal{N}_2)$.

Now let $e \in E_1 \cap E_2$, thus, for $i = 1, 2$:

$$e \cap B_i = e_i \cup e_o, e \cap B_i = e_o \cup e_i \text{ and } (e_i \cup e_o) \subseteq c_i \& (e_o \cup e_i) \subseteq d_i \& c_i - (e_i \cup e_o) = d_i - (e_o \cup e_i).$$

Hence $(*) \iff c_i[c > d_i \text{ in } \mathcal{N}_i, i = 1, 2.$

c) Assume that (4.1) holds for all $\alpha \in FS(\mathcal{N})$ of length less than or equal to n . We consider $e \in E^*$ with length $(\alpha e) = n + 1$. Thus,

$$c[[\alpha e > d \text{ in } \mathcal{N} \iff (\exists s \in \mathcal{C}), s = s_1 \cap (B_1 - B_2) \cup s_1 \cap s_2 \cup s_2 \cap (B_2 - B_1), s_1 \in \mathcal{C}_1, s_2 \in \mathcal{C}_2, c[[\alpha > s[e > d \text{ in } \mathcal{N} \iff c_i[[\alpha e_{|E_i} > s_i \text{ (by the inductive assumption) } \& s_i[e_{|E_i} > d_i \text{ (from the case b) in } \mathcal{N}_i(i = 1, 2) \iff c_i[[\alpha e_{|E_i} > d_i, i = 1, 2.$$

Thus, (4.1) holds in general, which completes the proof.

Note that after having composed, the state space \mathcal{C} may contain "redundant" cases. Using the reduction technique presented in Section 2 (Theorems 2.1 and 2.2) we can get a reduced net system composed from two given contact - free net systems. So we define:

$$\mathcal{N}_1 \oplus \mathcal{N}_2 = \mathcal{N}^M - \text{the composed net system of } \mathcal{N}_1 \text{ and } \mathcal{N}_2.$$

If confusion can be excluded, we will simply write \mathcal{N} instead of \mathcal{N}^M .

Let CFNS, CFL denote the family of all contact - free net systems and the family of all firing sequence languages generated by contact - free net systems, respectively. As an immediate consequence of Theorem 4.1, we have:

Corollary 4.1 $(CFNS, \oplus, \mathcal{N}_\emptyset)$ is a commutative monoid and CFL is closed under the synchronization operation $\#$.

In the practical point of view, the operation \oplus can be useful way for building large systems from smaller ones, especially, for constituting their state spaces.

Let $N = N_1 \cup N_2$ be the underlying net of $\mathcal{N} = \mathcal{N}_1 \oplus \mathcal{N}_2$ and I, D denote its independency and dependency relations, respectively. It is clear that $D = D_1 \cup D_2$. So $I = \overline{D} = \overline{D_1 \cup D_2} = \overline{I_1} \cup \overline{I_2}$. Using the sir-relations composition operation proposed in [10] we have:

$$I = I_1 \otimes I_2 = I_1 \cup I_2 - (E_1 \cap E_2) \times (E_1 \cap E_2) \cup I_1 \cap I_2 \cup (E_1 - E_2) \times (E_2 - E_1).$$

Let co, co_1, co_2 denote the concurrency relations, cl, cl_1, cl_2 the conflict relations of $\mathcal{N}, \mathcal{N}_1, \mathcal{N}_2$, respectively.

$$\text{Corollary 4.2 1) } co = co_1 \otimes co_2, \quad 2) \quad cl = cl_1 \cup cl_2.$$

So the composition operation preserves common pairs of concurrent events and develops concurrency.

Now we consider the synchronization of trace languages generated by net systems.

Given two concurrent alphabets (A, D) and (B, D') , where $B \subseteq A, D' \subseteq D$. The projection $h_B : A^* \rightarrow B^*$ can be extended to a mapping $h : T_D \rightarrow T_{D'}$ by setting: $h([\alpha]) = [h_B(\alpha)]$.

Let (A_1, D_1) and (A_2, D_2) be two concurrent alphabets. We define their union as:

$$(A, D) := (A_1, D_1) \cup (A_2, D_2) = (A_1 \cup A_2, D_1 \cup D_2).$$

Let $h_i : T_D \rightarrow T_{D_i} (i = 1, 2)$ be the corresponding projections. Given two trace languages $\mathcal{L}_1, \mathcal{L}_2$ over D_1 and D_2 , respectively. We define their *synchronization* $\mathcal{L}_1 \parallel \mathcal{L}_2$ as a trace language over D by:

$$\mathcal{L}_1 \parallel \mathcal{L}_2 = \{t \in T_D \mid h_1(t) \in \mathcal{L}_1 \& h_2(t) \in \mathcal{L}_2\}.$$

Return to contact - free net systems and the composition problem, we have:

Theorem 4.2 *If $\mathcal{N}_1, \mathcal{N}_2$ are contact - free net systems and \mathcal{N} is their composition then:*

$$\tau(\mathcal{N}) = \tau(\mathcal{N}_1) \parallel \tau(\mathcal{N}_2).$$

Proof: Follows from Corollary 3.2, Theorem 4.1 and the definition of the synchronization of trace languages.

Example 4.1 Consider the following net systems.

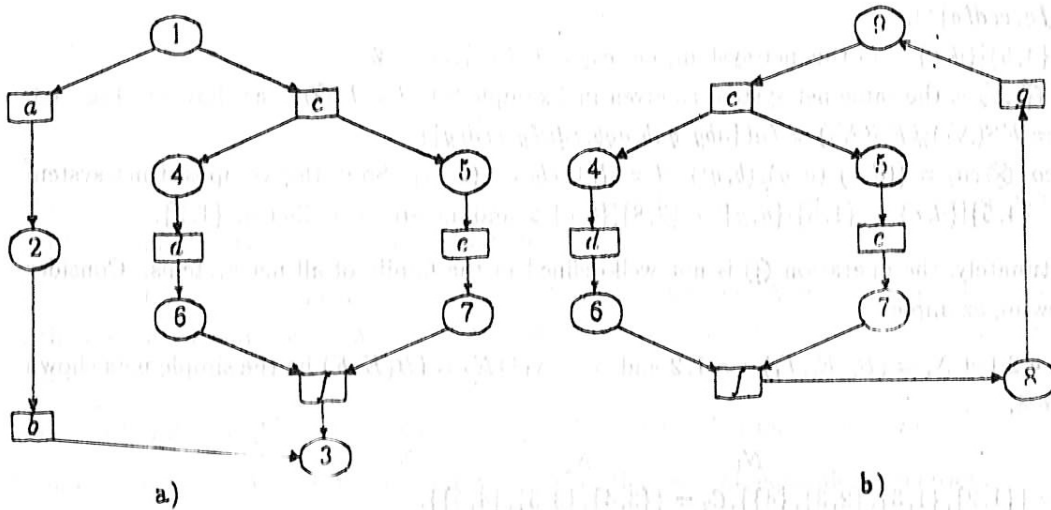


Fig. 2

Let $\mathcal{N}_1 = (B_1, E_1; F_1, C_1)$, where $N_1 = (B_1, E_1, F_1)$ is shown in Fig 2a), $C_1 = \{\{1\}, \{2\}, \{3\}, \{4, 5\}, \{4, 7\}, \{5, 6\}, \{6, 7\}\}$. This net system is contact-free. The independency relation I_1 is shown in Fig. 3a).

$FS(\mathcal{N}_1) = In(\{ab, cdef, cedf\})$. In this net system we have $\{4, 5\} \{d, e\} >$ and a, c are in conflict at $\{1\}$, $co_1 = \{(d, e)\}$ and $cl_1 = \{(a, c)\}$

Let $\mathcal{N}_2 = (B_2, E_2, F_2, C_2)$, where $N_2 = (B_2, E_2, F_2)$ is given in Fig. 2b). $C_2 = \{\{8\}, \{9\}, \{4, 5\}$,

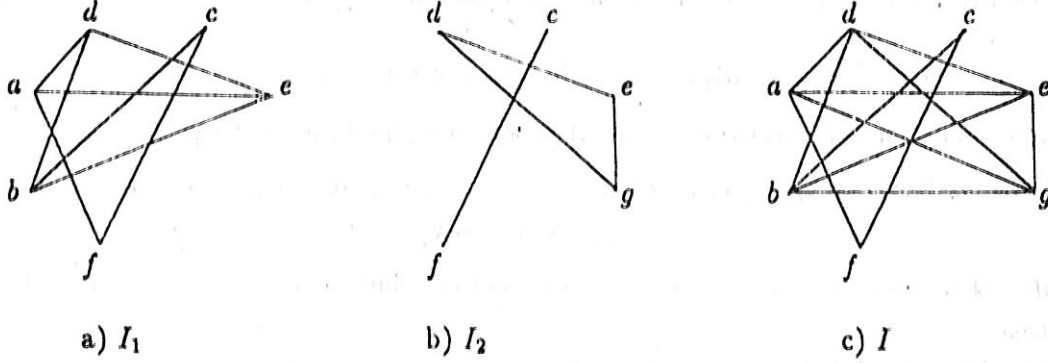


Fig. 3

$\{4, 7\}, \{5, 6\}, \{6, 7\}$. This net system is also contact - free and I_2 is shown in Fig. 3b). $FS(\mathcal{N}_2) = In(\{cdefg, cedfg\}^*)$.

Only $\{4, 5\}[\{d, e\}] >$ in this net system, i.e. $co_2 = \{(d, e)\}, cl_2 = \emptyset$.

$\mathcal{N} = \mathcal{N}_1 \oplus \mathcal{N}_2$ is the same net system as given in Example 2.1. $I = I_1 \otimes I_2$ as shown in Fig. 3c). $FS(\mathcal{N}) = FS(\mathcal{N}_1) \# FS(\mathcal{N}_2) = In(\{abg, gab, agb, edc, f, cedfg\})$.

$co = co_1 \otimes co_2 = \{(d, e), (a, g), (b, g)\}, cl = cl_1 \cup cl_2 = \{(a, c)\}$. So in this composed net system we have: $\{4, 5\}[\{d, e\}] >$, $\{1, 8\}[\{a, g\}] >$, $\{2, 8\}[\{b, g\}] >$ and a, c are in conflict at $\{1, 9\}$.

Unfortunately, the operation \oplus is not well-defined in the family of all net systems. Consider the following example.

Example 4.2 Let $\mathcal{N}_i = (B_i, E_i, F_i), i = 1, 2$ and $\mathcal{N} = \mathcal{N}_1 \cup \mathcal{N}_2 = (B, E, F)$ be the simple nets shown in Figure 4,

$\mathcal{N}_1 \qquad \mathcal{N}_2 \qquad \mathcal{N}$

and $\mathcal{C}_1 = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{4\}\}, \mathcal{C}_2 = \{\{3, 4\}, \{3, 5\}, \{4, 5\}\}$.

The net systems $\mathcal{N}_i = (B_i, E_i, F_i, \mathcal{C}_i), i = 1, 2$, are not contact - free.

By the definition of the composition operation proposed above, we have here:

$\mathcal{C} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{4\}, \{1, 2, 5\}, \{5\}, \{1, 3, 5\}, \{2, 3, 5\}, \{1, 5\}, \{2, 5\}, \{4, 5\}\}$.

\mathcal{C} is not closed under the reachability relation $R_{\mathcal{N}}$ because $\{1, 5\} \in \mathcal{C}, (\{1, 5\}, \{3\}) \in R1_{\mathcal{N}}$ but $\{3\} \notin \mathcal{C}$. So the quadruple $\mathcal{N} = (B, E, F, \mathcal{C})$ is not a net system.

Now we attempt to choose $\mathcal{C}^I = \max(\mathcal{C}) = \{\{1, 2, 5\}, \{1, 3, 5\}, \{2, 3, 5\}, \{4, 5\}\}$.

In this case $(\{1, 2, 5\}, \{2, 3\}) \in R1_{\mathcal{N}}$ but $\{2, 3\} \notin \mathcal{C}^I$. Even (B, E, F, \mathcal{C}^I) is not a net system too.

Nevertheless, the behavioural synchronizations of all net systems will be shown when using an equivalent transformation.

5. Synchronizations of Net Systems

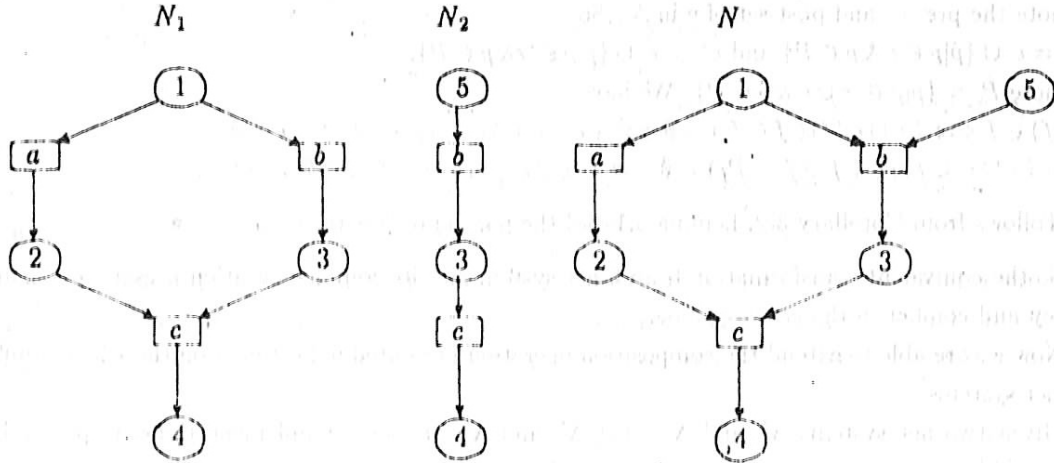


Fig. 4

In order to answer the question issued in Section 4 in the general case, we will use an equivalent transformation described in [8] for net systems.

Let $\mathcal{N} = (B, E, F, C)$ and $\mathcal{N}' = (B', E', F', C')$ be net systems. \mathcal{N} and \mathcal{N}' are called *equivalent* iff there exist two bijections: $\lambda : E \rightarrow E'$ and $\gamma : C \rightarrow C'$ such that, for all cases $c_1, c_2 \in C$ and each $e \in E : c_1[e > c_2 \iff \gamma(c_1)[\lambda(e) > \gamma(c_2)$.

Lemma 5.1 *If \mathcal{N} and \mathcal{N}' are equivalent then $FS(\mathcal{N}) = FS(\mathcal{N}')$ (upto isomorphism).*

Denote: $\mathcal{N} \sim \mathcal{N}'$ iff \mathcal{N} and \mathcal{N}' are equivalent. Note that \sim is an equivalence relation.

Let $\mathcal{N} = (B, E, F, C)$ be a net system and let $p, q \in B$.

- i) q is called *the complement* of p iff $p = q$ and $p = \bar{q}$.
- ii) \mathcal{N} is called *complete* iff every condition $p \in B$ has a complement $q \in B$.

Every net system can be transformed into an equivalent complete net system as follows:

Given a net system \mathcal{N} . Let $P \subseteq B$ be the set of those conditions which have no complement. For each $p \in P$, we add a new condition \hat{p} , and put:

$$F_P = \{(e, \hat{p}) | (p, e) \in F \& p \in P\} \cup \{(\hat{p}, e) | (e, p) \in F \& p \in P\}.$$

For each $c \in C$, let $\gamma(c) = c \cup \{\hat{p} | p \in P \& p \notin c\}$.

Denote $\hat{P} = \{\hat{p} | p \in P\}$ and $\gamma(C) = \{\gamma(c) | c \in C\}$.

Then the net system $\hat{\mathcal{N}} = (B \cup \hat{P}, E, F \cup F_P, \gamma(C))$ is the unique complementation of \mathcal{N} . It is obvious that $\mathcal{N} \sim \hat{\mathcal{N}}$.

Let I and \hat{I} be the independency relations of \mathcal{N} and $\hat{\mathcal{N}}$, respectively. We have:

Theorem 5.1 1) $I = \hat{I}$, 2) $\tau(\mathcal{N}) = \tau(\hat{\mathcal{N}})$.

Proof: 1) Let \bar{e} and e^- denote the pre-set and post-set of e in $\hat{\mathcal{N}}$, while $\cdot e$ and $e\cdot$, as usual, denote the pre-set and post-set of e in \mathcal{N} . So:

$$\bar{e} = e\cdot \cup \{\hat{p} | p \in e\cdot \& p \in P\} \text{ and } e^- = e\cdot \cup \{\hat{p} | p \in e\cdot \& p \in P\}.$$

Denote $\hat{P}_e = \{\hat{p} | p \in e\cdot \cup e\cdot \& p \in P\}$. We have:

$$\begin{aligned} (e, f) \in I &\iff (\cdot e \cup e\cdot) \cap (\cdot f \cap f\cdot) = \emptyset \iff (\cdot e \cup e\cdot) \cap (\cdot f \cup f\cdot) \cup \hat{P}_e \cap \hat{P}_f = \emptyset \\ &\iff (\cdot e \cup e\cdot \cup \hat{P}_e) \cap (\cdot f \cup f\cdot \cup \hat{P}_f) = \emptyset \iff (\bar{e} \cup e^-) \cap (\bar{f} \cup f^-) = \emptyset \iff (e, f) \in \hat{I}. \end{aligned}$$

2) Follows from Corollary 3.2, Lemma 5.1 and the part 1) of this theorem. •

So the equivalent transformation from a net system into its complementation preserves concurrency and conflict in these net systems.

Now we are able to extend the composition operation presented in Section 4 on the whole family of net systems.

Given two net systems \mathcal{N}_1 and \mathcal{N}_2 . Let $\hat{\mathcal{N}}_1$ and $\hat{\mathcal{N}}_2$ be their complementations, respectively. Hence, $\hat{\mathcal{N}}_1$ and $\hat{\mathcal{N}}_2$ are contact-free (see [8]). So we define:

$$\mathcal{N}_1 \oplus \mathcal{N}_2 := \hat{\mathcal{N}}_1 \oplus \hat{\mathcal{N}}_2$$

The family of net systems with the above operation and the identity \mathcal{N}_\emptyset becomes also a commutative monoid. Furthermore,

$$\begin{aligned} FS(\mathcal{N}_1 \oplus \mathcal{N}_2) &= FS(\mathcal{N}_1) \# FS(\mathcal{N}_2) \text{ and } \tau(\mathcal{N}_1 \oplus \mathcal{N}_2) = \tau(\mathcal{N}_1) \parallel \tau(\mathcal{N}_2), \\ co &= co_1 \otimes co_2 \text{ and } cl = cl_1 \cup cl_2. \end{aligned}$$

Summing up, we have:

Theorem 5.2 *The family of firing sequence languages (trace languages) generated by net systems and that by contact-free net systems are the same and they are closed under the respective synchronizations.*

Theorems 4.1, 4.2 and 5.2 give us an useful way to compute the behaviours of a composed net system from its components' behaviours, when they are already known (or easy to compute), without computing from beginning.

Let V, W be two monoids and let $\rho : V \rightarrow W$.

ρ is said to be *congruent* iff:

$$\forall u, u', v, v' \in V, \rho(u) = \rho(u') \wedge \rho(v) = \rho(v') \Rightarrow \rho(uv) = \rho(u'v').$$

Corollary 5.1 *FS and τ are congruent.*

6. Conclusion

We have presented our studies on a monoid of net system, whose operation is compatible with the synchronization of two basic semantics of nets: firing sequences and trace languages. The results are a step towards answering the question how some concurrent systems can co-operate

and what properties the composed systems have. Though the presented approach is devoted to net systems and their two basic behaviours, it can be applied as well to other models and other semantics. In many cases, the co-operation requires that an execution semantics of a composed concurrent system must be complete in the following sense. Every execution of subsystems is taken part to build up the execution semantics of the composed system, i.e.

$$L_1 \# L_2 \bar{L}_i = L_i \text{ for } i = 1, 2.$$

It causes to introduce the notion of a complete synchronization. An investigation of this property is under study.

Nevertheless, we believe that the behavioural synchronization will still be a basic characterization of the composition for many models of concurrent systems.

The author would like to thank R. Janicki and P.S. Thiagarajan for their valuable suggestions. This paper was written during my stay at the Tata Institute of Fundamental Research, Bombay, India. I'm grateful to all the members of the Computer Science Group of this Institute for helping me there.

References

1. L.A. Cherkasova, *On Models and Algebras for Concurrent Processes*, Lecture Notes in Computer Science 324, 1988, pp.27-43.
2. G.Gyory, E.Knuth & L.Ronyai, Grammatical Projections 1. *Elementary constructions*, Working paper II/3, Computer and Automation Institute, Hungarian Academy of Science, 1979.
3. R.Janicki, *Nets, Sequential Components and Concurrency Relations*, Theoretical Computer Science 29, 1984, pp.87-121.
4. R.Janicki, *Trace Semantics for Communicating Sequential Processes*, Tech. Report R-85-12, Institute for Electr. Systems, Univ. of Aalborg, Aalborg, Denmark, 1985.
5. V.E. Kotov, *An algebra for parallelism based on Petri nets*, Lecture Notes in Computer Science 64, 1978, pp.39-55.
6. A.Mazurkiewicz, *Semantics of concurrent systems: A modular fixed-point trace approach*, Lecture Notes in Computer Science 188, 1985, pp.353-375.
7. Y. Narahari and N.Viswanadham. *On the invariants of coloured Petri nets*, Lecture Notes in Computer Science 222, 1986, pp.330-345.
8. W.Reisig, *Petri Nets: An Introduction*, Springer-Verlag, 1985.
9. R.K.Shyamasundar, K.T.Narayana & T.Pitassi. *Semantics for Nondeterministic Asynchronous Broadcast Networks*, Lecture Notes in Computer Science 267, 1987, pp.72-83.
10. Hoang Chi Thanh, *Compositions of Marked Petri Nets*, Ph.D.Thesis, Warsaw Technical University, Poland, 1985, (in Polish).
11. Hoang Chi Thanh, *Behavioural Synchronization of Net Systems*, Proceedings of the National Seminar on Theoretical Computer Science, July 4-6, 1991, Madras, India, pp.136-145.
12. P.S. Thiagarajan, *Some behavioural aspects of Net theory*, Theoretical Computer Science 71, 1990, pp.133-153.

13. J. Winkowski, *Behaviours of Concurrent Systems*, Theoretical Computer Science 12, 1980, pp.36-60.

Abstract.

On Behavioural Synchronization in Net Systems

In this paper we define a new operation on Net systems. The operation is compositional with respect to two distinct semantics - interleaving based on firing sequences and non-interleaving based on traces.