

CONCURRENCY ANALYSIS FOR COLOURED PETRI NETS

Cristian Vidraşcu

Faculty of Computer Science, University “Al. I. Cuza” of Iaşi, Romania

vidrascu@info.uaic.ro

Abstract The goal of this paper is to study the relationships between the concurrency-degrees of a coloured Petri net and those of its subnets.

Keywords: parallel/distributed systems, Petri nets, concurrency.

2000 MSC: 68Q85.

1. INTRODUCTION

A Petri net is a mathematical model used for the specification and the analysis of parallel/distributed systems. An introduction about Petri nets can be found in [3]. The concurrency-degrees are a measurement of the concurrency in Petri nets, which was first introduced in [2].

It is useful to introduce a measure of concurrency for parallel/distributed systems, for answering a question like this: What is the meaning of the fact that in the system S_1 the concurrency is greater than in the system S_2 ?

The problem of concurrency is studied for Petri nets, but, since the Petri nets are used as suitable models for real-world parallel or distributed systems, the results are applicable also to these systems. The basic idea is that the number of transitions which can fire simultaneously in a Petri net which models a given real system, can be used as an intuitive measure of the concurrency of that system.

An advantage of studying the concurrency-degrees is that they can be computed for the model during the design of a system, and this will usually lead to an improved design.

It is well-known that the behaviour of some distributed systems cannot be adequately modelled by classical Petri nets. Many extensions which increase the computational power and/or the expressive power of Petri nets have been thus introduced. One direction has led to high-level Petri nets.

High-level Petri nets are a powerful language for system modelling and validation. They are now in widespread use for many different practical and theoretical purposes in various fields of software and hardware development. The step from low-level (i.e. classical) nets to high-level nets can be compared to the step from assembly languages to modern programming languages with an elaborated type concept. While in low-level nets there is only one kind of token and the state of a place is described

by an integer or a boolean value, in high-level nets each token can carry complex information.

There are two basic kinds of high-level Petri nets: *predicate/transition nets* and *coloured Petri nets*, which are very similar, but they are defined and presented in two rather different ways. Predicate/transition nets are defined using the notation and concepts of many-sorted algebras. Coloured Petri nets are defined using types, variables and expressions.

Coloured Petri nets were defined by K. Jensen (see [1]) shortly after the first kind of high-level nets (developed by H.J Genrich and K. Lautenbach), and they are more user-friendly as a system modelling language than are the predicate/transition nets.

In this paper we make an analysis of concurrency for high-level Petri nets, using coloured Petri nets as the presentation formalism. More precisely, we study the relationships between the concurrency-degrees of a coloured Petri net and those of its subnets.

2. PRELIMINARIES

We assume the basic terminology and notation about sets, relations and functions, vectors, multi-sets and formal languages to be known. Let us just briefly remind that a *multi-set* m , over a non-empty set S , is a function $m : S \rightarrow \mathbb{N}$, usually represented as a formal sum: $\sum_{s \in S} m(s) \cdot s$. In the previous sum, $m(s)$ is the multiplicity of the s element in that multi-set (see [1]). A multi-set will be sometimes identified with a $|S|$ -dimensional vector. The operations and relations on multi-sets are defined component-wise. S_{MS} denotes the set of all multi-sets over S . The empty multi-set $\sum_{s \in S} 0 \cdot s$ is denoted by \emptyset . The *size* of the multi-set m is defined as $|m| = \sum_{s \in S} m(s)$. The multi-set m is called *infinite* iff $|m| = \infty$.

Also, we will assume as known the basic terminology and notation about P/T-nets, the classical Petri nets. For details the reader is referred to [3]. The concurrency-degrees are a measurement of the concurrency in Petri nets, which was first introduced in [2] for P/T-nets. A more general definition of concurrency-degrees for them, which takes into consideration the self-concurrency (i.e. the case of the transitions concurrently enabled with themselves), and a finer notion, namely the concurrency-degrees w.r.t. a set of transitions, was presented in [5].

In the sequel we establish the basic terminology, notation, and results concerning coloured Petri nets in order to give the reader the necessary prerequisites for the understanding of this paper (for details the reader is referred to [1]).

2.1. COLOURED PETRI NETS

We present the formal definition of non-hierarchical *CP-nets* (abbreviation used for coloured Petri nets) given by Kurt Jensen in [1] (that book contains also the definition of hierarchical CP-nets, which are nets obtained by composing non-hierarchical CP-nets, possibly on more levels of composition).

Remark 2.1. A non-hierarchical CP-net is formally defined as a n -tuple, as we will see below. However, the only purpose of this is to give a mathematically sound and unambiguous definition of CP-nets and their semantics. Any particular net, created by a modeller (either manually, or using the software tool `Design/CPN` for modelling and analysis of CP-nets), will always be specified in terms of a graphical representation, called CPN diagram (see [1]). This approach is analogous to the definition of directed graphs and non-deterministic finite automata. Formally, they are defined as pairs $G = (V, A)$ and as 5-tuples $A = (S, \Sigma, \delta, s_0, F)$, respectively, but usually they are represented by drawings containing sets of nodes, arcs and inscriptions.

To give the abstract definition of CP-nets it is not necessary to fix the concrete syntax in which the modeller writes the net expressions, and thus it will be assumed that such a syntax exists together with a well-defined semantics – making it possible in an unambiguous way to talk about: i) The *elements of a data type* (or *sort*), T . The set of all elements of T (i.e., the domain of T) is denoted by the type name T itself. ii) The *type of a variable*, v – denoted by $Type(v)$. iii) The *type of an expression*, $expr$ – denoted by $Type(expr)$. iv) The *set of variables occurring in an expression*, $expr$ – denoted by $Var(expr)$. v) A *binding*, b , of a set of variables, V – associating with each variable $v \in V$ an element $b(v) \in Type(v)$. vi) The *value obtained by evaluating an expression*, $expr$, in a binding, b – denoted by $expr < b >$. $Var(expr)$ is required to be a subset of the variables of b , and the evaluation is performed by substituting for each variable $v \in Var(expr)$ the value $b(v) \in Type(v)$ determined by that binding.

Therefore, the syntax and semantics which are used for net expressions specify a many-sorted universal algebra (i.e., a many-sorted signature Σ and a many-sorted algebra over signature Σ), together with the set of terms (i.e. expressions) constructed over this algebra using a set of sorted variables, and with a first-order many-sorted logic with equality (over signature Σ) used for evaluating the expressions.

An expression *without* variables is called *closed expression*. It can be evaluated in all bindings, and all evaluations give the same value – which it will often be denoted by the expression itself (i.e. we simply write “ $expr$ ” instead of the more pedantic “ $expr < b >$ ”).

Now we are ready to remind the definition of non-hierarchical CP-nets (see [1]). We use \mathbb{B} to denote the boolean type (containing the elements {false, true}, and having the standard operations from propositional logic), and when V is a set of variables, we use $Type(V)$ to denote the set of types $\{Type(v) \mid v \in V\}$. Some motivations and explanations of the individual parts of the definition are given immediately below the definition, and it is recommended that these are read in parallel with the definition.

Definition 2.1. A non-hierarchical CP-net, is a 9-tuple $CPN = (S, T, A, N, \Sigma, C, G, E, I)$ satisfying the following requirements: (a) S is a finite set of places. (b) T is a finite set of transitions. (c) A is a finite set of arcs, such that: $S \cap T = S \cap A = T \cap A = \emptyset$. (d) $N : A \rightarrow S \times T \cup T \times S$ is a node function, describing the nodes of an arc. (e) Σ is a finite set of non-empty types, called colour sets. (f) $C : S \rightarrow \Sigma$ is a

colour function. (g) $G : T \rightarrow \{\text{expr} \mid \text{expr expression}\}$ is a guard function, satisfying the requirement: $\forall t \in T : \text{Type}(G(t)) = \mathbb{B} \wedge \text{Type}(\text{Var}(G(t))) \subseteq \Sigma$. (h) $E : A \rightarrow \{\text{expr} \mid \text{expr is expression}\}$ is an arc expression function, satisfying the requirement: $\text{Type}(E(a)) = C(s(a))_{MS} \wedge \text{Type}(\text{Var}(E(a))) \subseteq \Sigma$, for all $a \in A$, where $s(a)$ is the place of $N(a)$. (i) $I : S \rightarrow \{\text{expr} \mid \text{expr closed expression}\}$ is an initialization function, satisfying $\forall s \in S : \text{Type}(I(s)) = C(s)_{MS}$.

Notation 2.1. We use $X = S \cup T$ to denote the set of all nodes (i.e. places and transitions). Moreover, we define a number of functions describing the relationship between neighbouring elements of the net structure (the name of the function begins with a capital letter when its value is a set of elements; sometimes the same name is used for several functions/sets, but from the argument(s) it will always be clear which one is referred to): i) $s : A \rightarrow S$ maps each arc, a , to the place from the pair $N(a)$. ii) $t : A \rightarrow T$ maps each arc, a , to the transition from the pair $N(a)$. iii) $o : A \rightarrow X$ maps each arc, a , to the source of a , i.e. the first component of $N(a)$. iv) $d : A \rightarrow X$ maps each arc, a , to the destination of a , i.e. the second component of $N(a)$. v) $A : S \times T \cup T \times S \rightarrow \mathcal{P}(A)$ maps each ordered pairs of nodes, (x_1, x_2) , to the set of its connecting arcs, i.e. $A(x_1, x_2) = \{a \in A \mid N(a) = (x_1, x_2)\}$. vi) $A : X \rightarrow \mathcal{P}(A)$ maps each node, x , to the set of its surrounding arcs, i.e. $A(x) = \{a \in A \mid \exists x' \in X : N(a) = (x, x') \vee N(a) = (x', x)\}$. vii) $In : X \rightarrow \mathcal{P}(X)$ maps each node, x , to the set of its input nodes, i.e. $In(x) = \{x' \in X \mid \exists a \in A : N(a) = (x', x)\}$. viii) $Out : X \rightarrow \mathcal{P}(X)$ maps each node, x , to the set of its output nodes, i.e. $Out(x) = \{x' \in X \mid \exists a \in A : N(a) = (x, x')\}$. ix) $X : X \rightarrow \mathcal{P}(X)$ maps each node, x , to the set of its surrounding nodes, i.e. $X(x) = In(x) \cup Out(x)$. All the previous functions can be extended, in the usual way, to take sets as input (then they all return sets and thus all the function names are written with a capital letter).

Having defined the structure of CP-nets, we are now ready to consider their behaviour – but first we introduce the following notation:

Notation 2.2. (i) $\text{Var}(t)$ denotes the set of variables of transition t , i.e. $\text{Var}(t) = \{v \mid v \in \text{Var}(G(t)) \vee \exists a \in A(t) : v \in \text{Var}(E(a))\}$. (ii) $E(x_1, x_2)$ denotes the expression of (x_1, x_2) , i.e. $\forall (x_1, x_2) \in S \times T \cup T \times S : E(x_1, x_2) = \sum_{a \in A(x_1, x_2)} E(a)$ (the sum indicates addition of expressions; it is well-defined because all expressions have as type the same multi-set).

Next we remind the meaning of a *transition binding*. Intuitively, a binding of a transition t is a substitution that replaces each variable of t with a colour of the correct type and such that the guard evaluates to **true**.

Definition 2.2. A binding of a transition t is a function b defined on the set $\text{Var}(t)$, such that: (i) $\forall v \in \text{Var}(t) : b(v) \in \text{Type}(v)$. (ii) $G(t) < b > = \text{true}$. By $B(t)$ we denote the set of all bindings for t .

Next we present the notions of token elements, binding elements, markings and steps.

Definition 2.3. (i) A token element is an ordered pair (s, c) , where $s \in S$ and $c \in C(s)$. The set of all token elements is denoted by TE . A marking is a multi-set over TE . The set of all markings is denoted by \mathbb{M} , i.e. $\mathbb{M} = TE_{MS}$. The initial marking M_0 is the marking obtained by evaluating the initialization expressions: $\forall (s, c) \in TE : M_0(s, c) = (I(s))(c)$.

(ii) A binding element is an ordered pair (t, b) , where $t \in T$ and $b \in B(t)$. The set of all binding elements is denoted by BE . A step is a non-empty and finite multi-set over BE . The set of all steps is denoted by \mathbb{Y} .

Notation 2.3. $BE(t) = \{t\} \times B(t)$ denotes the set of all binding elements corresponding to transition $t \in T$, and $TE(s) = \{s\} \times C(s)$ the set of all token elements corresponding to place $s \in S$.

Remark 2.2. There is a unique correspondence between a marking $M \in TE_{MS}$ and a function \bar{M} defined on S such that $\bar{M}(s) \in C(s)_{MS}$, given by: $(\bar{M}(s))(c) = M(s, c)$, $\forall s \in S, \forall c \in C(s)$. This permits us often to represent markings as functions defined on S . Analogously, there is a unique correspondence between a step Y and a function \bar{Y} defined on T such that $\bar{Y}(t) \in B(t)_{MS}$ is finite for all $t \in T$ and non-empty for at least one $t \in T$. Hence, we often represent steps as functions defined on T .

Notation 2.4. Let CPN be a CP-net, and $Y \in \mathbb{Y}$ a step. The multi-sets (functions) $Y^-, Y^+ : TE \rightarrow \mathbb{N}$ and the function $\Delta Y : TE \rightarrow \mathbb{Z}$ are defined by: $Y^-(s, c) = \left(\sum_{(t,b) \in Y} E(s, t) \langle b \rangle \right)(c)$, $Y^+(s, c) = \left(\sum_{(t,b) \in Y} E(t, s) \langle b \rangle \right)(c)$ and $\Delta Y(s, c) = Y^+(s, c) - Y^-(s, c)$, $\forall (s, c) \in TE$.

Now we can present the formal definition of the behaviour of a CP-net:

Definition 2.4. The behaviour of a CP-net CPN is given by the firing rule, which consists in: (i) the enabling rule: a step Y is enabled in a marking M (or Y is fireable from M), abbreviated $M[Y]_{CPN}$, iff $\forall s \in S : Y^-(s) \leq M(s)$; (ii) the computing rule: if $M[Y]_{CPN}$, then Y may occur at the marking M yielding a new marking M' , abbreviated $M[Y]_{CPN}M'$, defined by: $\forall s \in S : M'(s) = (M(s) - Y^-(s)) + Y^+(s)$.

Let us notice that, in this way, for every step Y of the CP-net CPN we have defined a binary relation on TE_{MS} , denoted by $[Y]_{CPN}$, such that: $M[Y]_{CPN}M' \Leftrightarrow Y^- \leq M$ and $M' = M + \Delta Y$. The notation " $[.]_{CPN}$ " will be simplified to " $[.]$ " anytime the CP-net can be understood from the context.

Definition 2.5. Let the step Y be enabled in the marking M . When $(t, b) \in Y$, we say that t is enabled in M for the binding b . We also say that (t, b) is enabled in M , and so is t . When $(t_1, b_1), (t_2, b_2) \in Y$ and $(t_1, b_1) \neq (t_2, b_2)$ we say that (t_1, b_1) and (t_2, b_2) are concurrently enabled, and so are t_1 and t_2 . When $Y(t, b) \geq 2$ we say that (t, b) is concurrently enabled with itself. And when $|Y(t)| \geq 2$ we say that t is concurrently enabled with itself.

Definition 2.6. A finite occurrence sequence is a sequence of markings and steps $M_1[Y_1]M_2[Y_2]M_3 \dots M_n[Y_n]M_{n+1}$ such that $n \in \mathbb{N}$, and $M_i[Y_i]M_{i+1}$ for all $1 \leq i \leq n$. M_1 is called the start marking, M_{n+1} is called the end marking, and the non-negative integer n is called the length of the sequence.

Analogously, an infinite occurrence sequence is a sequence of markings and steps $M_1[Y_1]M_2[Y_2]M_3 \dots$ such that $M_i[Y_i]M_{i+1}$ for all $i \geq 1$. M_1 is called the start marking of the sequence, which is said to have infinite length.

Definition 2.7. A marking M' is reachable from a marking M in the CP-net CPN iff there exists a finite occurrence sequence having M as start marking and M' as end marking. The set of all reachable markings from M is denoted by $[M]_{CPN}$, or by $RS(CPN, M)$.

3. CONCURRENCY-DEGREES FOR CP-NETS

In this section we present a measure of concurrency for CP-nets, introduced in [4] (a tech. report in Romanian), which was obtained by extending to CP-nets the measures defined for classical Petri nets in [5].

Definition 3.1. Let CPN be a CP-net and M an arbitrary marking of CPN. A step Y is called a maximal step enabled at the marking M iff Y is enabled at M and there exists no step Y' enabled at M with $Y' > Y$.

Notation 3.1. i) $BE(M)$ denotes the set of all binding elements enabled in M , i.e. $BE(M) = \{(t, b) \in BE \mid M[(t, b)]\} = \{Y \in \mathbb{Y} \mid M[Y] \wedge |Y| = 1\}$.

ii) $\mathbb{Y}(M)$ denotes the set of all steps enabled at M in the net CPN, i.e. the set of all multi-sets of binding elements (concurrently) enabled at M : $\mathbb{Y}(M) = \{Y \in \mathbb{Y} \mid M[Y]_{CPN}\}$.

iii) $\mathbb{Y}_{max}(M)$ denotes the set of all maximal steps enabled at M in CPN, i.e. the set of all maximal multi-sets of binding elements (concurrently) enabled at M : $\mathbb{Y}_{max}(M) = \{Y \in \mathbb{Y}(M) \mid \forall Y' \in \mathbb{Y} : Y' > Y \Rightarrow Y' \notin \mathbb{Y}(M)\}$.

Remark 3.1. Let $T_0 = \{t \in T \mid In(t) = \emptyset\}$ be the set of the transitions with no input nodes in the CP-net CPN. If $T_0 = \emptyset$, then it is easy to remark that the sets $\mathbb{Y}(M)$ and $\mathbb{Y}_{max}(M)$ are finite, for any marking M .

Otherwise, if $T_0 \neq \emptyset$, then, for any marking M of CPN, the set $\mathbb{Y}(M)$ is infinite, and $\mathbb{Y}_{max}(M) = \emptyset$. Indeed, there exists a transition $t_0 \in T_0$, because $T_0 \neq \emptyset$, and therefore $(t_0, b) \in \mathbb{Y}(M)$, for any binding $b \in B(t)$. Thus, $\mathbb{Y}(M) \neq \emptyset$, and, if $Y \in \mathbb{Y}(M)$ is an arbitrary step enabled at M , then also $Y_k = Y + k \cdot (t_0, b) \in \mathbb{Y}(M)$, for all $k \in \mathbb{N}$.

Thus, $\mathbb{Y}(M)$ is infinite. Moreover, for any step $Y \in \mathbb{Y}(M)$, the set $\mathbb{Y}(M)$ contains an infinite strictly increasing sequence of steps converging to the limit $Y^* : BE \rightarrow \mathbb{N} \cup \{\infty\}$, with

$$Y^*(t, b) = \begin{cases} \infty & , \text{ if } In(t) = \emptyset \\ Y(t, b) & , \text{ otherwise} \end{cases} , \text{ for all } (t, b) \in BE.$$

Obviously, there exists no maximal step enabled at M , thus $\mathbb{Y}_{max}(M) = \emptyset$.

Intuitively, the notion of concurrency-degree at a marking M of a CP-net represents the maximum (i.e. the supremum) number of binding elements concurrently enabled at M .

Definition 3.2. Let CPN be a CP-net and M an arbitrary marking of CPN . The concurrency-degree at M of the net CPN is defined by:

$$d(CPN, M) = \sup\{|Y| \mid Y \in \mathbb{Y}(M)\} . \quad (1)$$

Remark 3.2. Directly from definitions and Remark 3.1 it follows that

$$\forall M \in \mathbb{M} : d(CPN, M) = \begin{cases} \max\{|Y| \mid Y \in \mathbb{Y}_{max}(M)\} & , \text{ if } T_0 = \emptyset \\ +\infty & , \text{ otherwise} \end{cases}$$

and it does not depend on the initial marking of the net.

Definition 3.3. i) The inferior concurrency-degree of a CP-net CPN is defined by:

$$d^-(CPN) = \min\{d(CPN, M) \mid M \in [M_0]_{CPN}\} . \quad (2)$$

ii) The superior concurrency-degree of the net CPN is defined by:

$$d^+(CPN) = \sup\{d(CPN, M) \mid M \in [M_0]_{CPN}\} . \quad (3)$$

iii) If $d^-(CPN) = d^+(CPN)$, then this number is called the concurrency-degree of the net CPN and it is denoted by $d(CPN)$.

Remark 3.3. Directly from definitions we have

i) $0 \leq d^-(CPN) \leq d^+(CPN) \leq \infty$.

ii) The inferior concurrency-degree of a CP-net CPN , $d^-(CPN)$, represents the minimum number of binding elements (transitions) maximal concurrently enabled at any reachable marking of CPN . In other words, at any reachable marking M of the net CPN there exist $d^-(CPN)$ binding elements (transitions) concurrently enabled at M .

iii) The superior concurrency-degree of a CP-net CPN , $d^+(CPN)$, represents the supremum number of binding elements (transitions) maximal concurrently enabled at any reachable marking of CPN . In other words, at any reachable marking M of the net CPN there exist at most $d^+(CPN)$ binding elements (transitions) concurrently enabled at M .

iv) The concurrency-degree of the net CPN means that at any reachable marking M there exist $d(CPN)$ binding elements (transitions) concurrently enabled at M , and there is no reachable marking M' of CPN with more than $d(CPN)$ binding elements (transitions) concurrently enabled at M' .

As we could see from Remark 3.1, sometimes it can be useful to ignore some transitions of a net and to study the behaviour of the net w.r.t. the remaining transitions. Or we could ignore only some binding elements and study the behaviour of

the net w.r.t. the remaining binding elements. Therefore, we introduced the notion of concurrency-degree of a CP-net w.r.t. a subset of binding elements, and w.r.t. a subset of transitions.

Definition 3.4. Let CPN be a CP-net and $BE' \subseteq BE$ a subset of binding elements. A step over BE' , Y , is a step satisfying $Y(t, b) = 0$, for all $(t, b) \in BE - BE'$ (practically, Y is a non-empty and finite multi-set over the subset BE'). Therefore, by $\mathbb{Y}|_{BE'} = \mathbb{Y} \cap BE'_{MS}$ we will denote the set of all steps over BE' of the net CPN . Moreover, we will denote by $\mathbb{Y}|_{BE'}(M)$ the set of all steps over BE' enabled at M , and by $(\mathbb{Y}|_{BE'})_{max}(M)$ the set of all maximal steps over BE' enabled at M .

Definition 3.5. Let CPN be a CP-net, $BE' \subseteq BE$ a subset of binding elements, and M an arbitrary marking of CPN . The concurrency-degree w.r.t. BE' at the marking M of the net CPN , denoted by $d(CPN, BE', M)$, is defined by replacing $\mathbb{Y}(M)$ with $\mathbb{Y}|_{BE'}(M)$ in (1).

Definition 3.6. Let CPN be a CP-net and $BE' \subseteq BE$ a subset of binding elements. The inferior and superior concurrency-degree w.r.t. BE' of the net CPN , denoted by $d^-(CPN, BE')$ and $d^+(CPN, BE')$, resp., are defined by replacing $d(CPN, M)$ with $d(CPN, BE', M)$ in (2) and (3), resp. Moreover, if $d^-(CPN, BE') = d^+(CPN, BE')$, then this number, denoted by $d(CPN, BE')$, is called the concurrency-degree of the net w.r.t. BE' .

Remarks 3.2 and 3.3 hold similarly for these notions of concurrency-degrees w.r.t. a set of binding elements.

Furthermore, we can speak about concurrency-degrees w.r.t. a subset of transitions, $T' \subseteq T$, of a CP-net CPN . We take the set of all binding elements corresponding to the transitions from the subset T' , namely $BE' = \cup \{BE(t) | t \in T'\} = \cup \{\{t\} \times B(t) | t \in T'\}$, and we define all the above notions w.r.t. T' by using this set. Thus, a step over T' will be a step over BE' , $\mathbb{Y}|_{T'}$ will denote the set of all steps over T' , $\mathbb{Y}|_{T'}(M)$ will denote the set of all steps over T' enabled at M , and $(\mathbb{Y}|_{T'})_{max}(M)$ will denote the set of all maximal steps over T' enabled at M .

Definition 3.7. Let CPN be a CP-net, $T' \subseteq T$ a subset of transitions, and M an arbitrary marking of CPN . The concurrency-degree of the net w.r.t. T' at M is defined as $d(CPN, T', M) = d(CPN, \cup \{BE(t) | t \in T'\}, M)$.

Definition 3.8. Let CPN be a CP-net and $T' \subseteq T$ a subset of transitions of this net. The inferior and superior concurrency-degree w.r.t. T' of the net CPN , are defined as $d^-(CPN, T') = d^-(CPN, \cup \{BE(t) | t \in T'\})$ and $d^+(CPN, T') = d^+(CPN, \cup \{BE(t) | t \in T'\})$, resp. Moreover, if $d^-(CPN, T') = d^+(CPN, T')$, then this number, denoted by $d(CPN, T')$, is called the concurrency-degree w.r.t. T' of the net CPN .

Remarks 3.2 and 3.3 hold similarly for these notions of concurrency-degrees w.r.t. a set of transitions.

4. CONCURRENCY ANALYSIS FOR THE SUBNETS OF A NET

We can take into consideration the problem of modularization for coloured Petri nets: a CP-net can be “decomposed” into several modules, i.e. subnets of it, which have in common some locations of the net; these locations play the role of “interface” (i.e., they are *shared*) between two or more modules. Using this setting, the study of the concurrency in the global net can be done by analyzing the concurrency of the subnets which form that net.

Thus, it is useful to study the relationships between the concurrency-degrees of a CP-net and those of the subnets which compose that net.

The following result shows the connection between the concurrency-degree w.r.t. the union of two disjoint sets of transitions and the concurrency-degrees w.r.t. each of those two sets:

Theorem 4.1. *Let CPN be a CP-net and $T_1, T_2 \subseteq T$ two disjoint sets of transitions. The following inequalities hold, for any marking M of the net:*

$$d(\text{CPN}, T_1 \cup T_2, M) \leq d(\text{CPN}, T_1, M) + d(\text{CPN}, T_2, M) \quad (4)$$

$$d^+(\text{CPN}, T_1 \cup T_2) \leq d^+(\text{CPN}, T_1) + d^+(\text{CPN}, T_2) \quad (5)$$

Proof. i) Let CPN be a CP-net, $T_1, T_2 \subseteq T$ two disjoint sets of transitions, and M an arbitrary marking of CPN. Let $Y \in \mathbb{Y}_{|T_1 \cup T_2}(\text{CPN}, M)$ be an arbitrary step over $T_1 \cup T_2$ enabled at M . Since $T_1 \cap T_2 = \emptyset$, we can write:

$$\begin{aligned} Y^-(s) &= \sum_{(t,b) \in Y} E(s,t) \langle b \rangle = \sum_{t \in T_1 \cup T_2, b \in BE(t)} E(s,t) \langle b \rangle = \\ &= \sum_{t \in T_1, b \in BE(t)} E(s,t) \langle b \rangle + \sum_{t \in T_2, b \in BE(t)} E(s,t) \langle b \rangle, \end{aligned}$$

and thus, since $M[Y]_{\text{CPN}}$, by the step enabling rule it follows that $M(s) \geq Y^-(s)$, so we conclude that, for all places s :

$$\sum_{t \in T_1, b \in BE(t)} E(s,t) \langle b \rangle \leq M(s) \quad \text{and} \quad \sum_{t \in T_2, b \in BE(t)} E(s,t) \langle b \rangle \leq M(s).$$

First inequality means that the step denoted by $Y|_{T_1} \stackrel{\text{not}}{=} \{(t,b) \in Y \mid t \in T_1, b \in BE(t)\}$ is a step over T_1 fireable at M in CPN, i.e. $M[Y|_{T_1}]_{\text{CPN}}$. By the definition of the concurrency-degree at a marking for CP-nets, this means that $|Y|_{T_1}| \leq d(\text{CPN}, T_1, M)$.

Similarly, from the second inequality from above we deduce that the step denoted by $Y|_{T_2} \stackrel{\text{not}}{=} \{(t,b) \in Y \mid t \in T_2, b \in BE(t)\}$ is a step over T_2 enabled at M in CPN, and therefore we have that $|Y|_{T_2}| \leq d(\text{CPN}, T_2, M)$.

Obviously, $Y = Y|_{T_1} + Y|_{T_2}$, so we can conclude that

$$|Y| = |Y|_{T_1}| + |Y|_{T_2}| \leq d(CPN, T_1, M) + d(CPN, T_2, M).$$

Thus, we showed that $|Y| \leq d(CPN, T_1, M) + d(CPN, T_2, M)$, for all Y which are steps over $T_1 \cup T_2$ enabled at M in CPN . By taking the supremum in this inequality, after Y as a step over $T_1 \cup T_2$ enabled at M in CPN , and using the definition of the concurrency-degree at a marking for CP-nets, we obtain the desired inequality: $d(CPN, T_1 \cup T_2, M) \leq d(CPN, T_1, M) + d(CPN, T_2, M)$.

ii) Let CPN be a CP-net, and $T_1, T_2 \subseteq T$ two disjoint sets of transitions. By the definition of the superior concurrency-degree w.r.t. a set of transitions, we have the inequalities: $d(CPN, T_1, M) \leq d^+(CPN, T_1)$ and $d(CPN, T_2, M) \leq d^+(CPN, T_2)$, for any reachable marking.

Since, by pct. i), the inequality (4) holds for any arbitrary marking M , and, thus, it holds particularly for the reachable ones, we conclude that $d(CPN, T_1 \cup T_2, M) \leq d^+(CPN, T_1) + d^+(CPN, T_2)$, for all $M \in [M_0]_{CPN}$.

By taking the supremum after $M \in [M_0]_{CPN}$ in the above inequality, and using the definition of the superior concurrency-degree for CP-nets, we obtain the desired inequality: $d^+(CPN, T_1 \cup T_2) \leq d^+(CPN, T_1) + d^+(CPN, T_2)$. ■

Remark 4.1. Unfortunately, regarding the inferior concurrency-degree, neither an inequality like (5), nor one with an inverse sign, holds true (a counterexample was given in [4]).

Despite the fact that for the inferior concurrency-degree there exists no upper bound like the ones which exist for the concurrency-degree at a marking and for the superior concurrency-degree, we can still specify a lower bound for the inferior concurrency-degree of Petri nets, namely:

Theorem 4.2. Let CPN be a CP-net and $T_1, T_2 \subseteq T$ two disjoint sets of transitions. The following inequality holds:

$$d^-(CPN, T_1 \cup T_2) \geq \max\{d^-(CPN, T_1), d^-(CPN, T_2)\} \quad (6)$$

Proof. Let M be an arbitrary marking of the net CPN .

Obviously, any step over T_1 enabled at the marking M in CPN is also a step over $T_1 \cup T_2$ enabled at M in CPN (being a multiset over $T_1 \cup T_2$ having zero multiplicities for the elements from T_2).

Thus, by the definition of the concurrency-degree at a marking, we can deduce that $d(CPN, T_1 \cup T_2, M) \geq d(CPN, T_1, M)$, for any arbitrary marking M , and thus, particularly, also for any $M \in [M_0]_{CPN}$.

But, from the definition of the inferior concurrency-degree for CP-nets, it follows that $d(CPN, T_1, M) \geq d^-(CPN, T_1), \forall M \in [M_0]_{CPN}$. Thus, we conclude that $d(CPN, T_1 \cup T_2, M) \geq d^-(CPN, T_1), \forall M \in [M_0]_{CPN}$.

Similarly it can be shown that $d(CPN, T_1 \cup T_2, M) \geq d^-(CPN, T_2), \forall M \in [M_0]_{CPN}$. Using these two inequalities, we deduce that

$$d(CPN, T_1 \cup T_2, M) \geq \max\{d^-(CPN, T_1), d^-(CPN, T_2)\},$$

for all $M \in [M_0]_{CPN}$.

By taking the minimum after $M \in [M_0]_{CPN}$ in the previous inequality, and using the definition of the inferior concurrency-degree, we get the desired inequality: $d^-(CPN, T_1 \cup T_2) \geq \max\{d^-(CPN, T_1), d^-(CPN, T_2)\}$. ■

Remark 4.2. *The inequalities (4) and (5) from Theorem 4.1, as well as the inequality (6) from Theorem 4.2, can also be reformulated in terms of two disjoint sets of binding elements BE_1 and BE_2 (instead of the two disjoint sets of transitions T_1 and T_2), and the proofs are quite similar.*

Remark 4.3. *Moreover, the inequalities (4) and (5) from Theorem 4.1, as well as the inequality (6) from Theorem 4.2, hold also for the generalized case of any finite union of pairwise-disjoint sets of transitions. (This can be easily proved by applying the inequalities mentioned above, reiteratively for unions of two disjoint sets of transitions.)*

A particular case of these generalized inequalities is represented by the case when the sets of transitions are all singletons (i.e., each set has only one element). In this case we obtain the following result, which expresses the relationship between the concurrency-degree w.r.t. a set of transitions and the concurrency-degrees w.r.t. each individual transition from that set:

Corollary 4.1. *Let CPN be a CP-net and $T' \subseteq T$ a subset of transitions. Then the following inequalities hold, for any marking M :*

$$d(CPN, T', M) \leq \sum_{t \in T'} d(CPN, \{t\}, M) , \quad (7)$$

$$d^+(CPN, T') \leq \sum_{t \in T'} d^+(CPN, \{t\}) , \quad (8)$$

$$d^-(CPN, T') \geq \max_{t \in T'} d^-(CPN, \{t\}) . \quad (9)$$

Proof. These inequalities follow as simple consequences from Theorem 4.1 and 4.2, by applying reiteratively (by $|T'| - 1$ times) the corresponding inequalities from the two mentioned theorems. ■

Thus, an interesting question that arises is: when any of the inequalities (4), (5) and (6), or the analogous of inequality (5) for the inferior concurrency-degree of a CP-net becomes equality ?

Let us notice the following fact: if CPN is a CP-net, $T' \subseteq T$ a subset of transitions, and M_1, M_2 are two arbitrary markings of CPN such that $M_1(s) = M_2(s)$, for any location $s \in In(T')$, then any step over T' enabled at M_1 in CPN is also enabled at M_2 in CPN and viceversa. Thus, the set of steps over T' enabled at M_1 in the net CPN is equal with the set of steps over T' enabled at M_2 in CPN , and, therefore, $d(CPN, T', M_1) = d(CPN, T', M_2)$.

In other words, for CP-nets the concurrency-degree at a marking w.r.t. a set of transitions depends *only* on the components of that marking which correspond to the input places of the transitions from that set.

This remark gives us a structural property of a CP-net which is a sufficient condition for some of the above mentioned equalities to hold:

Theorem 4.3. *Let CPN be a CP-net, and $T_1, T_2 \subseteq T$ two disjoint subsets of transitions. If its structure has the property that $In(T_1) \cap In(T_2) = \emptyset$, then the following equalities hold, for all markings M :*

$$d(CPN, T_1 \cup T_2, M) = d(CPN, T_1, M) + d(CPN, T_2, M) \quad (10)$$

$$d^-(CPN, T_1 \cup T_2) \geq d^-(CPN, T_1) + d^-(CPN, T_2) \quad (11)$$

Proof. By inequality (4) from Theorem 4.1, for proving (10) it is sufficient to show that we have the inequality (4) with an invers sign in the hypothesis $In(T_1) \cap In(T_2) = \emptyset$ satisfied by the net CPN .

Let Y_1 be an arbitrary step over T_1 enabled at M , and Y_2 an arbitrary step over T_2 enabled at M . Then, by the enabling rule of a step, we have

$$Y_1^-(s) \stackrel{\text{not}}{=} \sum_{t \in T_1, b \in BE(t)} E(s, t) \langle b \rangle \leq M(s), \forall s \in S,$$

$$\text{and } Y_2^-(s) \stackrel{\text{not}}{=} \sum_{t \in T_2, b \in BE(t)} E(s, t) \langle b \rangle \leq M(s), \forall s \in S.$$

Let $Y = Y_1 + Y_2$. Thus, Y is a step over $(T_1 \cup T_2)$. Then

$$Y^-(s) \stackrel{\text{not}}{=} \sum_{t \in T_1 \cup T_2, b \in BE(t)} E(s, t) \langle b \rangle = Y_1^-(s) + Y_2^-(s), \forall s \in S.$$

Considering the fact that $t^-(s) = 0$ iff $s \notin In(t)$, for any $t \in T$ and $s \in S$, and since, by hypothesis, $In(T_1) \cap In(T_2) = \emptyset$, it follows that for any location $s \in S$, one and only one of the following three cases is possible:

- i) $s \in In(T_1)$. Then $s \notin In(T_2)$ and, thus, $Y_2^-(s) = 0$. So, we obtain that $Y^-(s) = Y_1^-(s) \leq M(s)$.
- ii) $s \in In(T_2)$. Then $s \notin In(T_1)$ and, thus, $Y_1^-(s) = 0$. So, we obtain that $Y^-(s) = Y_2^-(s) \leq M(s)$.
- iii) $s \in S - (In(T_1) \cup In(T_2))$. Then $s \notin In(T_1)$ and $s \notin In(T_2)$, and, therefore, $Y_1^-(s) = Y_2^-(s) = 0$. So, we obtain again that $Y^-(s) = 0 \leq M(s)$.

In conclusion, we proved that $Y^-(s) \leq M(s), \forall s \in S$. This inequality means that Y is a step over $(T_1 \cup T_2)$ enabled at the marking M in CPN .

By using the definition of the concurrency-degree at a marking w.r.t. a set of transitions, from this it follows that $|Y| \leq d(CPN, T_1 \cup T_2, M)$. Thus, since $Y = Y_1 + Y_2$, we have that $|Y_1| + |Y_2| = |Y| \leq d(CPN, T_1 \cup T_2, M)$.

Therefore, we proved that $d(CPN, T_1 \cup T_2, M) \geq |Y_1| + |Y_2|$, for any arbitrary step over T_1 enabled at M in CPN , Y_1 , and any arbitrary step over T_2 enabled at M in CPN , Y_2 .

By successively taking the supremum in the above inequality, after Y_1 as a step over T_1 enabled at M in CPN , and then after Y_2 as a step over T_2 enabled at M in CPN , and using the definition of the concurrency-degree at a marking for CP-nets, we obtain the desired inequality: $d(CPN, T_1 \cup T_2, M) \geq d(CPN, T_1, M) + d(CPN, T_2, M)$.

To prove the second part of this theorem, let $M \in [M_0]_{CPN}$ be an arbitrary reachable marking of the net CPN . Then, by the definition of the inferior concurrency-degree, we have that $d(CPN, T_1, M) \geq d^-(CPN, T_1)$ and $d(CPN, T_2, M) \geq d^-(CPN, T_2)$. Thus, by applying the first part of this theorem, we obtain that $d(CPN, T_1 \cup T_2, M) = d(CPN, T_1, M) + d(CPN, T_2, M) \geq d^-(CPN, T_1) + d^-(CPN, T_2)$, for all $M \in [M_0]_{CPN}$.

By taking the minimum after $M \in [M_0]_{CPN}$ in the previous inequality, and using the definition of the inferior concurrency-degree, we get the desired inequality: $d^-(CPN, T_1 \cup T_2) \geq d^-(CPN, T_1) + d^-(CPN, T_2)$. ■

Remark 4.4. We can formulate similar results to Corollary 4.1 and Theorem 4.3 (i.e. by reformulating them in terms of two disjoint sets of binding elements BE_1 and BE_2 , instead of the two disjoint sets of transitions T_1 and T_2) to show the connection between the concurrency-degree w.r.t. a subset of binding elements of a CP-net and the concurrency-degrees w.r.t. each individual binding element from that subset.

Remark 4.5. Obviously, equality (10) and inequality (11) from Theorem 4.3 hold also for the generalized case of any finite union of pairwise-disjoint sets of transitions. (This rem can be easily proved by applying (10), and respectively (11), reiteratively for unions of two disjoint sets of transitions.)

A particular case of these generalized equality and inequality for Petri nets is represented by the case when the sets of transitions are all singletons (i.e., each set has only one element). In this case we obtain the following result, which expresses a sufficient condition for having the above mentioned relationships between the concurrency-degree w.r.t. a set of transitions and the concurrency-degrees w.r.t. each individual transition from that set:

Corollary 4.2. Let CPN be a CP-net and $T' \subseteq T$ a subset of transitions. If CPN satisfies the property $In(t_1) \cap In(t_2) = \emptyset$, for any $t_1, t_2 \in T'$, then the following equality

holds for any marking M of the net CPN:

$$d(\text{CPN}, T', M) = \sum_{t \in T'} d(\text{CPN}, \{t\}, M) , \quad (12)$$

and, regarding the inferior concurrency-degree, the following inequality holds:

$$d^-(\text{CPN}, T') \geq \sum_{t \in T'} d^-(\text{CPN}, \{t\}) . \quad (13)$$

Proof. These relations follow as simple consequences from Theorem 4.3, by applying iteratively (by $|T'| - 1$ times) the corresponding relations from that theorem. ■

Remark 4.6. Unfortunately, the condition $\text{In}(T_1) \cap \text{In}(T_2) = \emptyset$ is not sufficient neither for having (5) with equality for the superior concurrency-degree, nor for having the similar equality for the inferior concurrency-degree.

However, inequality (11), which holds in the hypothesis $\text{In}(T_1) \cap \text{In}(T_2) = \emptyset$, represents an improvement of the lower bound given by inequality (6), for the inferior concurrency-degree of a CP-net.

5. CONCLUSION

In this paper we presented the notion of concurrency-degrees for high-level Petri nets, using coloured Petri nets as the presentation formalism, and we made an analysis of concurrency for them, i.e. we studied the relationships between the concurrency-degrees of a coloured Petri net and those of its subnets.

Since the CP-nets are used as suitable models for real parallel or distributed systems, the concurrency-degrees defined for CP-nets are an intuitive measure of the concurrency of the modelled systems, and, therefore, they have a practical importance. For instance, they are useful for the evaluation of the models in the process of designing such a system: after making a model of that system as a CP-net, the study of the concurrency-degree of the model will give to the designers information about the concurrency of that system, allowing them to notice the inefficient components of the system, i.e. the components with bottlenecks w.r.t. parallelism/concurrency (because of the non-optimum use of the system's resources, or because of other causes), and to make improvements of the model by remodelling those components in order to eliminate the causes of the bottlenecks. In this way, the evaluation of the model can produce useful feedback to the previous stages of the designing process, even until to the first stage of the specifications of the system, for making some improvements in that stage with the purpose of increasing the overall performance of the designed system.

Some problems remain to be studied, for example:

- finding the conditions for which the inequality (5) holds true with equality for the superior and resp. the inferior concurrency-degree of a CP-net;
- making some case studies on models of real-world systems.

References

- [1] K. Jensen, *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*, Volume 1, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1992.
- [2] T. Jucan, C. Vidraşcu, *Concurrency-degrees for Petri nets*, Studia Universitatis Babeş-Bolyai, Computer Science Section, **XLIV**(2)(1999), 3–15.
- [3] W. Reisig, *Petri Nets. An Introduction*, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, Berlin, 1985.
- [4] C. Vidraşcu, *Structural properties of Petri nets*, Technical report TR 04–04, Faculty of Computer Science, Univ. “Al.I.Cuza” of Iaşi, Dec. 2004. (in Romanian)
- [5] C. Vidraşcu, T. Jucan, *Concurrency-degrees for P/T-nets*, Sci. Annals of Univ. “Al.I.Cuza” of Iaşi, Computer Science Section, **XIII**(2003), 91–103.

