Verification of Systems: Deadlock Analysis Based on Petri Nets

Štefan Hudák

Department of Computers and Informatics
FEI TU 04011 Košice,
Letná 9, Slovak Republic,
e-mail:stefan.hudak@tuke.sk

Abstract. *T*he present work is devoted to the study of deadlock problem in Place/Transition (P/T) nets, particularly to the exploration of how a deadlocks' presence can be revealed solely on the basis of the P/T net N_0 in question and a structure, here termed as *the fsa of the type* M_w, that represents the reachability set $\Re(N_0)$ of N_0. The structure can be obtained from N_0 following the original algorithm for solving the reachability problem RP for Petri Nets in general case by the author. It turns out, that the structure of M_w bears some important properties with respect to the deadlock analysis.
Deadlock analysis is an important part of system verification, so the results achieved can be of some value to that. It is demonstrated that results presented are quite significant, and cover some gap in both, theory and practice of the deadlock analysis of state-based systems, particularly those whose specification can be expressed via Petri Nets.

Keywords. Place/Transition Nets, deadlock analysis, reachability, finite state representation of the state reachability set, finite state automaton of the type M_w .

Key Terms. Mathematical Model, Specification Process, Verification Process

1 Introduction

In the development of (state-based) system, the design of a system in question, is the core of the process. The latter is actually a realization of the what requirement specification is about. There are two intrinsic activities of any development:*validation* and *verification*. The validation is the process of assurance that the design will produce the right system (according to requirements), while the verification assures that the design is carried on properly (according to a particular design principle)[21]. Verifying the system designed on the presence (absence) of deadlock situations (deadlock analysis-DA) might be a part of verification process. In the paper we pay attention to the deadlock analysis.

The approach to deadlock analysis applied in the paper is based on the reachability analysis [6] made on the representation (model) of the system designed in Petri Nets [8]. In [6] an original method (algorithm) to analyze and solve the reachability problem (RP) for Petri Nets in general case was introduced. The method is based on the structure, we have called it ***the finite state automaton of the type*** M_w, that is created by the algoritm, and the analysis of the structure based on results of automata theory and the convex analysis of the state space represented by M_w. Some authors in the field of Petri Nets used to use for representing state space of reachable states of Petri nets the structure termed as *coverability graph.* The two notions coincide to some extent, but differ in significant number of cases.
The approach is founded on the information that was neglected and suppressed by the RP algorithm while creating M_w, and thus hidden in it. The modification of M_w construction, that is introduced in this paper, discloses the information previously hidden to serve the purpose mentioned.
The paper consists of four parts. In the first part basic notions and results concerned Petri Nets, reachability and deadlock analysis are given. The second part deals with the algebraic properties of M_w. It turns out that M_w is finite state automaton, with some interpretations of its states via k-dimensional nonegative integer ω vectors. Each such ω vector represents a state subspace of PN in question, and can be thought of as a poset. That view on ω states allows us to establish relation among deadlocks and minimal or least elements of such the posets. In the third part we deal with the issue of disclosing the information previously hidden, and define more precisely the new notion and denotation of ωcoordinates. The fourth part consists of an application of the theory of ωcoordinates developed. The application is made to two PNs, which was introduced by T.Murata [1] as manifestation of the fact, that using coverability graphs as the representation of state space of PN is weak and insufficient for disclosing deadlocks in PN. The same conclusion was jumped to in [2].

2 Some basic preliminaries to DA

In the paper we denote by $\mathbb{N}$ the set of natural numbers $\{0, 1, 2, \ldots\}$, by $\mathbf{Z}$ the set of all integers, $\mathbf{Z}^k$ ($\mathbb{N}^k$) the set of k-dimensional (nonnegative)integer vectors.
A notion of (k-dimensional) *vector addition system* (VAS) W_k is a couple
$W_k = (q_0, W)$
where $q_0 \in \mathbb{N}^k$ is the initial state of W_k, W is a finite set of (k- dimensional integer) vectors. We call a reachable state vector of W_k each $q \in \mathbb{N}^k$ such that

1. $q = q_0 + w_{i_1} + \ldots + w_{i_n}$ for some integer $n \geq 0, w_{i_j} \in W, j = 1, \ldots, n$ and
2. for $\forall j (1 \leq j < n) : q_j = q_0 + w_{i_1} + \ldots + w_{i_j} \in \mathbb{N}^k$

Here by + we mean the operation of vector addition. We call the set of all such vectors *the reachability set* of VAS W_k, and denote it as $R(W_k)$. Given any VAS

$W_k = (q_0, W)$ then or any $q \in \mathbb{N}^k$ a problem whether $q \in R(W_k)$ is called *the reachability problem* of VAS (with respect to q). We will occasionally use the abbreviation $RP(q, W_k)$ for it.

With any VAS $W_k = (q_0, W)$ we can associate a tree structure, which we call *the vector state tree*, VST_w, and we mean by that a double labelled oriented rooted tree $VST_w = (T_w, Lab(V), Lab(E), q_0)$, $T_w = (V, E, r_0)$ is an oriented rooted tree , V - a set of vertices, $E \subseteq V \times V$ - a set of edges, $r_0 \in V$ - the root of T_w , $Lab(V) \subseteq \mathbb{N}^k$ -a set of vertex labels, $Lab(E) \subseteq W$ - a set of edge labels that are defined as follows: there are two labelling mappings $lab_1 : V \to Lab(V), lab_2 : E \to Lab(E)$ such that $lab_1(r_1) = q_0$ and any vertex of T_w $v \in V$ with $lab_1(v) = q$ has a son $u \in V$ with $lab_1(u) = q'$ and $lab(v, u) = a$ iff $q' = q+a$.

As a very consequence of the above definition we have that $Lab(V) = R(W_k)$ where $W_k = (q_0, W)$ is the VAS and we can alternatively write $VST_w = (T_w, R(W_k), Lab(E), q_0)$ □

2.1 Place/Transition Nets (P/T Nets).

Place/Transition (P/T) Nets stand here for a class of Petri Nets in which multiple arcs are allowed and places have unlimited capacities. For more details on PN we refer the reader to the literature , e.g. [8]. For any P/T net $N_0 = (P, T, pre, post, m_0)$, where P is a finite set of places, T is a finite set of transitions, $pre : P \times T \longrightarrow \mathbb{N}$ - preset function, and $post : P \times T \longrightarrow \mathbb{N}$ - postset function, that all define a structure on the set $P \cup T$. It is very common to represent the P/T Net /[1] by the oriented bipartite graph (Fig. 1).
Here we have:

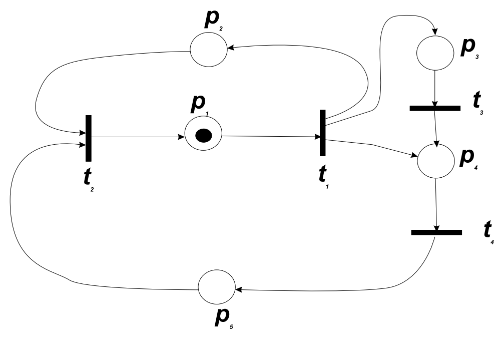

Fig. 1. Graph representation of Petri Net

[1] We will use Petri Net (PN) occasionally instead of P/T Net, so we consider them as synonyms

$$P = \{p_1, p_2, p_3, p_4, p_5\}$$
$$T = \{t_1, t_2, t_3, t_4\}$$

pre and *post* functions are given in Table 1 and Table 2 respectively.

Table 1:

P	T	pre(p,t)
p_1	t_1	1
p_2	t_2	1
p_5	t_2	1
p_3	t_3	1
p_4	t_4	1
otherwise		0

Table 2:

P	T	post(p,t)
p_1	t_2	1
p_2	t_1	1
p_3	t_1	1
p_4	t_1	1
p_4	t_3	1
p_5	t_4	1
otherwise		0

In Fig. 2 there is a correspondence shown between the graph representation of PN N and *pre* and *post* functions.

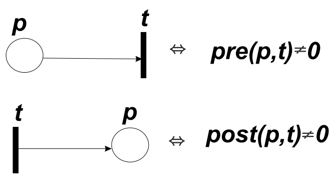

Fig. 2. The correspondence between the graph representation of PN and the *pre* and *post* functions

The following useful notations can be defined:

$^\bullet t = \{p \mid pre(p,t) \neq 0\}$ the set of preconditions of t
$t^\bullet = \{p \mid post(p,t) \neq 0\}$ the set of postconditions of t
$p^\bullet = \{t \mid pre(p,t) \neq 0\}$
$^\bullet p = \{t \mid post(p,t) \neq 0\}$

By the marking of PN $N = (P, T, pre, post)$ we mean a totally defined function

$$m : P \longrightarrow \mathbb{N} \tag{1}$$
(2)

We use m to describe the situation or configuration in PN N. Namely we say the condition represented by the place p in PN N holds iff $m(p) \neq 0$. Without loss of generality we assume that P and T have k and m elements respectively.

i.e. $P = \{p_1, p_2, , ..., p_k\}$, $T = \{t_1, t_2, , ..., t_m\}$ and we fix some ordering of both, places and transitions from now on. Using the ordering of places we can consider m to be the k-dimensional nonnegative integer vector, i.e. $\vec{m} \in \mathbb{N}^k$. More formally

$$\vec{m} = (m(p_1), m(p_2), ..., m(p_k))$$

and $m(p_i)$ is the value of m in p_i , $i = 1, 2, ..., k$, according to (1). In our example (Fig. 1) $m(p_i) = 1$ iff $i = 1$, or alternatively $\vec{m} = (1, 0, 0, 0, 0)$. For the simplicity we will use the denotation m for either interpretations of the marking m when it doesn't cause any troubles. We say t is enabled in m, and denote it m $\stackrel{t}{\vdash}$, iff for every $p \in^\bullet t$, $m(p) \geq pre(p, t)$. In Fig. 1 t_1 is enabled in $m = (1, 0, 0, 0, 0)$ because ${}^\bullet t_1 = \{p_1\}$ and $m(p_1) = 1$, and $pre(p_1, t_1) = 1$. In general, given PN N, a marking m of N, several transitions from T can be enabled in m. Once the transition t is enabled it can fire. The effect of the firing t in m is the creation of a new marking m' that depends on m and t. We use a denotation

$$m \stackrel{t}{\vdash} m'$$

and m' is defined in the following way:

$$m'(p) = \begin{cases} m(p) - pre(p,t) & p \in {}^\bullet t \setminus t^\bullet \\ m(p) + post(p,t) & p \in t^\bullet \setminus^\bullet t \\ m(p) - pre(p,t) + post(p,t) & p \in {}^\bullet t \cap t^\bullet \\ m(p) & otherwise \end{cases}$$

In PN N of Fig. 1 we can write $m = (1, 0, 0, 0, 0) \stackrel{t_1}{\vdash} m' = (0, 1, 1, 1, 0)$. Notice transitions t_3,t_4 will be enabled in m' either.

We say the sequence of transitions $\sigma = t_1 t_2 ... t_r$ is *admissible firing sequence* in PN N, provided a sequence of markings $m_0, m_1, ..., m_r$ exists and such that $m_{i-1} \stackrel{t_i}{\vdash} m_i$, $i = 1, 2, ..., r$. In that case we write $m_0 \stackrel{\sigma}{\vdash} m_r$, or simply $m_0 \stackrel{*}{\vdash} m_r$, when σ is immaterial. The marking m is to be called *the reachable marking* in N from m_0 (via σ). We fix the marking m_0 to be *the initial marking* of PN $N = (P, T, pre, post)$ and we denote it $N_0 = (N, m_0)$ or $N_0 = (P, T, pre, post, m_0)$. Given PN $N_0 = (P, T, pre, post, m_0)$ we define the set of reachable markings

$$\mathcal{R}(N_0) = \{m \mid m_0 \stackrel{\sigma}{\vdash} m, \},$$

We can define the language of PN N_0

$$L(N_0) = \{\sigma \in T^* \mid m_0 \stackrel{\sigma}{\vdash} m, \sigma \in T^*\}$$

and we call it *PN language.*

2.2 VAS and Petri Nets.

Let $N_0 = (P, T, pre, post, m_0)$ be a Petri Net with the initial marking m_0. Recall m_0 can be represented as a k-dimensional nonnegative integer vector, i.e. $\boldsymbol{m}_0 \in \mathbf{Z}^k$ and $\boldsymbol{m_0} = (m_0(p_1), \ldots, m_0(p_k))$. Let us fix an ordering of places in P and transitions in T, i.e. $P = \{p_1, \ldots, p_k\}$ and $T = \{t_1, \ldots, t_m\}$.

In PN literature (e.g. [6],[8]) we have the following characterization of the marking obtained (reached) in N_0 from initial marking m_0 under firing transition sequence $\sigma \in T^*$

$$m_0 \left|\!\overset{\sigma}{-}\right. m \Leftrightarrow \boldsymbol{m} = \boldsymbol{m}_0 + (c \cdot \Psi^T(\sigma))^T$$

and $\Psi(\sigma)$ is the Parikh mapping over the (ordered) alphabet T, and $\Psi^T(\sigma)$ stands for the transposition of the row vector $\Psi(\sigma)$.

Any transition $t \in T$ can be represented as a k-dimensional integer vector

$$\boldsymbol{t} = \boldsymbol{post}(t) - \boldsymbol{pre}(t)$$

and

$$\begin{aligned} \boldsymbol{post}(t) &= ((post(p_1, t), \ldots, post(p_k, t) \\ \boldsymbol{pre}(t) &= ((pre(p_1, t), \ldots, pre(p_k, t)) \end{aligned}$$

It can be easily seen that

$$m_0 \left|\!\overset{t}{-}\right. m \Leftrightarrow \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{t}$$

and we can construct for PN $N_0 = (P, T, pre, post, m_0)$ the vector addition system $W_k = (q_0, W)$ an such that $q_0 = \boldsymbol{m}_0, W = \{\boldsymbol{t} | t \in T\}$, and $k = cardP$. The following result holds

Theorem 1. *[6]*

For any PN $N_0 = (P, T, pre, post, m_0)$ there is an vector addition system $W_k = (q_0, W)$ and such that $\mathcal{R}(W_k) = \mathcal{R}(N_0)$, and $k = cardP$.
Proof: That follows from the above construction. □

2.3 Reachability Problem

Reachability problem for Petri nets attracted a lot of attention of experts in computer science community. It lasted pretty long time (almost 20 years) a solution to RP had been obtained [9],[10],[11],[12],[4]. A full account of the solution of RP by the author, including the complexity issue of RP- the upper bound of the worst-case time complexity of the solution, can be found in [6].
We are going now to describe shortly main steps of the author's RP solution.

1. Any P/T net $N_0 = (P, T, pre, post, m_0)$ can be assigned a vector adition system (VAS) $W_k = (q_0, W)$ via a representation of transitions of P/T net N_0 as vectors, where $W = \left\{ \vec{t_i} \mid t_i \in T \right\}$, $(q_0 = m_0)$ and $\vec{t_i} = (post(p_1, t_i) - pre(p_1, t_i), ..., post(p_k, t_i) - pre(p_k, t_i))$, provided $P = \{p_1, ..., p_k\}$. By that virtue the computations of P/T net N_0 are in 1-1 correspondence with computations of the VAS W_k and $\mathcal{R}(W_k) = \mathcal{R}(N_0)$ (see Theorem 1). The computations of the VAS W_k can be represented via rooted labelled tree, termed as *vector state tree* - VST_w, whose vertices are labelled by reachable states and edges are labelled by vectorized transitions.
2. Given VST_w and its vertex with the label q, it can be characterized by two languages:$\mathbf{X}_q$, $\mathbf{Y}_q$, preffix and suffix language respectively$^{/}$ which denotes labelling of paths leading to or from the vertex with the label q. The paths on VST_w can be classified, based on the length of the paths: finite and infinite, on the one side, and also based on the finite or infinite set of vector-states: vertex labellings on the other side. Any path on VST_w, outgoing from the root vertex r_0, labelled by q_0, can be assigned a sequence of its vertex labels (reachable) vector-states

$$s = \{q_0, q_1, ..., q_i, ...\} \tag{3}$$

The states in (3) are reachable states, that are vectors, i.e. $q_i \in \mathbf{N}^k$ (k=$||P||$), so for any pair of states in (3)- (q_i, q_j), $i < j$, we can test their comparability, w.r.t. the relation $\leq$ defined on vectors. (of the same dimension). The sequence (3) can be accompanied by the sequence of suffix(prefix) languages associated with the states of the sequence (3) . By the nature and due to properties of VASs and their computations that is clear that

$$q_i \leq q_j \Rightarrow \mathbf{Y}_{q_i} \subseteq \mathbf{Y}_{q_j} \tag{4}$$

The necessary and sufficient conditions can be formulated for a path being infinite with finite or infinite set of reachable states. Based on that a theory of transformation of infinite paths (a graph morphism), that allows pruning infinite paths and replacing them by loop-like subgraphs and thus transforming the tree into a rooted graph (vector state graph-*vsg*). The transformation $(T_{<_A^m})$ has a significant property that suffix language of the root of the original $VST_w - \mathbf{Y}_{q_0}$ is included in the suffix language of the root of the resulting vsg $T_{<_A^m}(q_0)$, i.e. $\mathbf{Y}_{q_0} \subseteq \mathbf{Y}_{T_{<_A^m}(q_0)}$. In the case the strong inequality holds between the two states on the path ($q \leq q'$ and $q \neq q'$), that causes introducing so-called ω-cordinates, that means replacing the cordinates of the both states in which the strong inequality ($<$) holds, by the special value ω, and thus creating the ω-*lized state* $\omega_A q$ and $\omega_A q'$, that become identical, i.e. $\omega_A q = \omega_A q'$ (A is the set of coordinate indices on which the relation $<$ holds).

By that virtue, due to the properties of ω ($\omega + a = \omega - a = \omega$ for any natural number a), any such the transformation has two-side effect: pruning the infinite path by replacing it by a finite (loop-like) subgraph, and lowering number of coordinates w.r.t. which a comparison satisfiability of reachable (macro) states should be checked. That guaranties that in a finite number of transformation steps a finite (rooted) vsg structure $\mathcal{T}_f^{\omega}$ can be obtained. The significant property of the vsg $\mathcal{T}_f^{\omega}$ is that $\mathbf{Y}_{T^*(q_0)} \supseteq \mathbf{Y}_{q_0}$, provided that $T^*(q_0)$ is the macrostate on which the initial state q_0 is mapped after the sequence of transformations denoted as T^*.

3. Vsg $\mathcal{T}_f^{\omega}$ can be thought of as a special kind of *finite state automaton (fsa)* with some interpretation of its states, and with the input alphabet $W = \left\{\overrightarrow{t_i} \,|t_i \in T\right\}$. The definition of the automaton (we used to call it *finite state automaton (fsa) (of the type)* M_w) can be given as $M_w = (Q_f, W, \delta, \rho_0)$, provided the vsg $\mathcal{T}_f^{\omega} = (Q_f, \mathcal{T}_f, \rho_0$ and $\mathcal{T}_f$ is the graph representation of state trasition funcion δ. To characterize the behaviour of fsa M_w we introduce special regular expressions (*wre*-vector regular expressions(w stands here after the set W of vectors)). Any wre α is given two semantics: $[\alpha]$-vector semantics; $[\![\ \alpha\ \ ]\!]$ - (ordinary) language semantics. Let $\mathcal{L}^{\rho_0}$ be the wre that denotes the language of M_w (i.e.$L(M_w)$= $[\![\ \mathcal{L}^{\rho_0}\ \ ]\!]$), and q_0 to be the initial state of VAS $\mathcal{W}_k$. Then $[q_0\mathcal{L}^{\rho_0}]$ denotes all reachable states. To be more precise

$$
\begin{aligned}
[u] &= [u] \ \text{ if } u \in W \\
[au] &= a + [u] \ \text{ if } a \in W \text{ and } u \in W^* \\
\forall q \in \mathbf{N}^k, u \in W^* \ [qu] &= q + [u] \text{ and } \forall i (1 \le i \le |u_i|).q_i = [qu_i] \in \mathbf{N}^k, \\
[q_0\mathcal{L}^{\rho_0}] &= \left\{ q^{'} | q^{'} = [q_0 u]\,, u \in \ [\![\ \mathcal{L}^{\rho_0}\ ]\!] \right\} \qquad (5)
\end{aligned}
$$

4. Having constructed fsa M_w it is worth to say few words about its structure w.r.t. how it can be useful in RP solving:
   - The structure of the state diagram of M_w is, in almost all cases, consisting of $n \geq 1$ strongly connected components (scc), due to transformations applied to VST_w initially and to vsg afterwards. The class of scc-like M_ws, can be divided into two subclasses. The first subclass contains M_ws whose states are labelled by simple (k-dimensional) nonnegative integer vectors. Such M_w manifests that P/T net in question N_0 (and corresponding VAS W_k) has finite set of reachable states. The second subclass consists of M_ws whose states are labelled by ω (k-dimensional) nonnegative integer vectors (vectors having at least one ω coordinate). Such M_w manifests that P/T net in question N_0 (and corresponding VAS W_k) has infinite set of reachable states.
   - The way how to solve the reachability problem w.r.t. a state $q \in \mathbf{N}^k$ will differ depending on whether $\mathcal{R}(N_0)$ is finite or infinite. In the finite case $RP(q, N_0)$ can be solved trivially by inspecting the state diagram of M_w and checking whether there is a state with the label q or not. In the second case we have to do the following steps:
     1) to find a state ρ of M_w such that $q \leq \rho$; if such a state does not exists, then $RP(q, N_0)$ has negative solution ($q \notin \mathcal{R}(N_0)$).

2) assume we found such the state ρ; now we have to construct a path leading from the root state ρ_0 that is the image of the initial state q_0 under chain of transformations ($\rho = T^*_{<^m_A}(q_0)$)(for more details see [6]). By that way a wre u over the alphabet $W_L \cup W$ (W_L is the alphabet of (ρ_0)-simple loops of scc, i.e. $W_L \subseteq W^*$) can be constructed, yielding the equation

$$[q_0 uv] = q \tag{6}$$

Wre u (under assumption of one scc in M_w, rooted in ρ_0) has the structure $u = \ell_1\ell_2...\ell_p$, where $\ell_i \in W_L$, $i = 1, 2, ..., p$ and v is a path leading from ρ_0 to ρ such that $q \leq \rho$.

– The equation (6) yields integer linear programming problem (ILP)

$$\begin{aligned} A\mathbf{X} = B(q), \quad B(q) = q - q_0 - [v] \\ A = ([\ell_1]^T, [\ell_2]^T \cdots [\ell_{m_0}]^T) \end{aligned} \tag{7}$$

provided $W_L = \{\ell_1, \ell_2, ..., \ell_{m_0}\}$.

5. ILP constructed does not express exactly conditions to hold for the reachability of the state q. The reason is that at building ILP (7) based on (6) some information is lost. Particularly the information that is connected with an ordering of loops passed, that is prescribed by definition of $[q_0uv]$(all reachable states by wre uv). To check the so called 'proper choice condition' property special test should be performed, that is expressed in the predicate $con_{W_k}(A,\ X_0,\ B_0)$. So finally the RP algorithm is
RP algorithm:
Given: VAS $W_k = (q_0, W)$, $q \in \mathbb{N}^k$ - a state to be decided reachable or not;

Step 1 : Create fsa M_w;
Step 2 : Construct $MILP_w(A, X_0, B(q), r)$;
Step 3 : if $MILP_w(A, X_0, B(q), r) \ = \ true$ then go to Step 4 else go to Step 5 ;
Step 4 : $q \in R(W_k)$. Stop.
Step 5 : $q \notin R(W_k)$. Stop.

We use the abbreviation

$$\begin{aligned} MILP_{W_k}(A,\ X_0,\ B(q)) \ \equiv \ & ILP_{W_k}(A,\ X_0,\ B(q)) \wedge \\ & con_{W_k}(A,\ X_0,\ B_0) \end{aligned}$$

Finally

$$RP(q, W_k) \equiv \ MILP_{W_k}(A,\ X,\ B(q))$$

Since ILP is decidable, and also due to finiteness of X_0 establishing truth of $con_{W_k}(A,\ X_0,\ B_0)$ is also decidable, so is the reachability problem.

3 Algebraic properties of M_w automaton

Let us have one more look at fsa $M_w = (Q, W, \delta, \rho_0)$. For simplicity let us assume that M_w consists of single scc with its root state ρ_0. Example of such M_w is depicted in Fig. 3

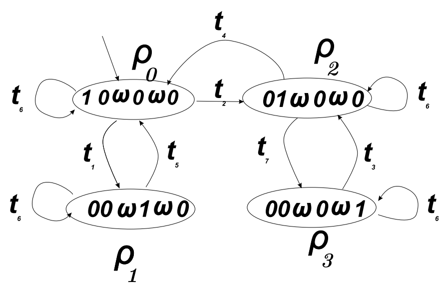

Fig. 3. State diagram of fsa M_w with a single strongly connected component

Notice that all states of the state diagram are labelled with ωvectors, e.g. $\rho_0 = (1, 0, \omega, 0, \omega, 0)$, $\rho_1 = (0, 0, \omega, 1, \omega, 0)$, $\rho_2 = (0, 1, \omega, 0, \omega, 0)$, $\rho_3 = (0, 0, \omega, 0, \omega, 1)$. Important feature of labels of the states of the M_w's state diagram is that they are mutually incomparable as vectors.
We can look at labels of the states of the M_w's state diagram as *macrostates*, that represent (cover) sets of reachable states. For that, we may call any macrostate ρ - a label of a state in M_w's state diagram, as the *reachable macrostate*. We will say that two macrostates ρ and $\rho^{'}$ are *comparable*, and we write $\rho \leq \rho^{'}$, provided that $\rho^{'}$ covers at least those reachable states, that are covered by ρ.
To express it more formally, we introduce for the macrostate ρ the set of covered reachable states- denoted by S_ρ i.e.

$$S_\rho = \{q | q \in R(N_0), q \leq \rho\}$$

$\mathbf{S}_\rho$ is simply *partially ordered set*(poset). The notion is well-known [19]. For any poset, particularly for S_ρ, there can be found the set of *minimal*, or *maximal* elements (states) respectively. The notions *the lower bound*, or *the upper bound* of the states of $\mathbf{S}_\rho$ are also well defined and used. The notion *the greatest lower bound* (glb), and *the least upper bound* (lub) are also used in that context. We only mention here, that while minimal (maximal) states belong to $\mathbf{S}_\rho$, that might not be true for glb or lub respectively. We will use the notation $\sqcup, \sqcap$ to denote the binary operation of calculating $lub(q, q^{'}) = q \sqcup q^{'}$, or $glb(q, q^{'}) = q \sqcap q^{'}$ respectively.

For the definition of poset, and further properties and other results reader can consult a specialized literature on the subject, e.g. see [18],[19].

For our purpose to use the information captured in fsa M_w for deadlock analysis we have to modify the algorithm of creating M_w. The information that is hidden in the state diagram of M_w is the value of particular coordinate of the state q at the moment when the coordinate is being ω-lized. .
To capture the information hidden (and lost) by the original algorithm to construct the fsa of the type M_w, we have to distinguish three types of indexing ω coordinates:

- ω_a^{Δ} - denotes an ω coordinate in a loop-root state ρ, with initial value of the coordinate a , with Δ as a loop added value to the coordinate after each repetition of the loop; such the ω is called the *independent root* ω ,
- $\omega_{\overline{b}}^{\,i,t,\Delta_j}$- denotes so-called *dependent* ω coordinate in a loop-root state ρ, that depends on i-th ω coordinate that should generate (via repetition of its loop) a minimal value v in i-th coordinate such that $v \geq pre(p_i, t)$ for the transition t to be firable in corresponding state while the initial value of the coordinate the dependant ω belongs to is b; Δ_j is the increase of the *dependent* ω (in the j-th coordinate) caused by the repetition of the loop started by the transition t.
- ω_c- denotes so-called *overflowed* ω coordinate with the minimal initial value of the coordinate at the time it was overflowed for the first time.

To get a flavour why we are introducing indexed ωs , we are now turning our attention to properties of the poset $\mathbf{S}_\rho = (S_\rho, \leq, \sqcup, \sqcap)$ with respect to a deadlock state π, that can be eventually covered by a macrostate ρ, i.e. $\pi \leq \rho$.

3.1 Properties of the poset $\mathbf{S}_\rho$ with respect to the deadlock analysis

In the previous section we have discovered, that any macrostate ρ can be taken as the poset $\mathbf{S}_\rho = (S_\rho, \leq, \sqcup, \sqcap)$. For the discovering a deadlock state of P/T net N_0 we would like to make a use of information captured in the M_w's state diagram, specifically in macrostates by which the states are labelled with.

Assume that we are given ω-state ρ, and to be more specific, let's say that

$$\rho = \omega_A q = (\rho_1, \rho_2, ..., \rho_k) \tag{8}$$

where $q = (q_1, q_2, ..., q_k)$ is a reachable state, $A \subseteq \{1, 2, ..., k\} = K$ is the set of indices in which ρ has ω - coordinates. To put it in other words that means that

$$\rho_j = \begin{cases} \varpi \text{ if } j \in A \\ q_j \text{ if } j \notin A \end{cases}$$

and $\varpi \in \left\{\omega_a^{\Delta_i}, \omega_{\overline{b}}^{\,i,t,\Delta_j}, \omega_c\right\}$.

Now we are introducing some notions.

First we fix the macrostate ρ and its representation (8). We define

$$Base\rho = \{\ q^B_{\rho,i} = \left(q_1^{'}, q_2^{'}, ..., q_k^{'}\right) | i \in A,\ q_\ell^{'} = \rho_\ell \ \text{ if }\ i \neq \ell,\ \ell \in K - \{i\}\,, \quad q_i^{'} = r \ \text{ if }\ \rho_i = \varpi\ \} \tag{9}$$

where $\varpi \in \left\{\omega_r^{\Delta_i}, \omega_r^{\underline{j,t,\Delta_i}}, \omega_r\right\}$.
That is clear that every $q^B_{\rho,i} \leq \rho$; in a case ρ has only one ω coordinate then $q^B_{\rho,i}$ is the $glb(\mathbf{S}_\rho)$. In the case that $||A|| > 1$ $q^B_{\rho,i}$ is the macrostate covering a set of minimal elements of the poset $\mathbf{S}_\rho$.

We will call the macrostates labelling states of M_w the *reachable macrostates*. Any macrostate $\pi \leq \rho$ we will call also the *reachable macrostate*. From that point of view we may consider elements of $Base\rho$ as the collection of reachable macrostates.

Still another notion should be introduced; we define

$$q^B_\rho = \left(q_1^{''}, q_2^{''}, ..., q_k^{''}\right) \tag{10}$$

where

$$q_j^{''} = r \Leftrightarrow \rho_j = \varpi$$
$$q_j^{''} = \rho_j \Leftrightarrow \rho_j \in \mathbf{N}$$

and $\varpi \in \left\{\omega_r^{\Delta_j}, \omega_r^{\underline{i,t,\Delta_j}}, \omega_r\right\}$ for some $r \in \mathbf{N}$.
It is clear that

$$q^B_\rho \leq \rho$$

The crucial problem is to decide whether q^B_ρ is reachable state or not. In the latter case it will be called *spurious state* [1]. We will postpone answering that question later on.

Any *deadlock state* of P/T net

$$N_0 = (P, T, pre, post, q_0)$$

is such a state q, that is

1. reachable state, i.e. $q \in \mathcal{R}(N_0)$, and
2. for any transition $t \in T$ and at least one $p \in^\bullet t$, $pre(p,t) > q(p)$

In other words there are not enough tokens at least in one of pre-places $^\bullet t$ of any $t \in T$.

Assume we have a deadlock state $d \in \rho$; that means that any reachable state $q \in \rho$ and such that $q \leq d$ will be a deadlock state either. From that we have immediately, that if q^B_ρ were reachable state, it would be a deadlock state of P/T net $N_0 = (P, T, pre, post, q_0)$ since it would have been either the least or minimal element of the poset $\mathbf{S}_\rho$. We can summarize the properties described.

Assertion 1 *Let* $\boldsymbol{S}_\rho = (S_\rho, \leq, \sqcup, \sqcap)$ *be the poset formed by the macrostate* ρ *of the fsa* M_w *representing the set of reachable states of a P/T net* $N_0 = (P, T, pre, post, q_0)$. *Then if there exists a deadlock state* d *in* ρ, *then at least one of minimal elements or the least element of the poset* $\boldsymbol{S}_\rho$*-* q_{min} *will be the deadlock state too and such that* $q_{min} \leq d$. □

So, it means that the least and minimal elements of the poset $\mathbf{S}_\rho$, if they exist, serve as a good indicator of presence and/or absence of deadlocks in the system represented by any P/T net.

In the algorithm of the construction of fsa M_w [6] we apply some transformations to the paths of the tree of computations of the VAS $W_k = (q_0, W)$, which results in introducing ω values into corresponding coordinates of a state vector. We have shown above there are three types of ω coordinates (ωcords in short):independent, dependent and overflowed ωcords.

The issue of creating independent ω coordinate is depicted in Fig. 4.

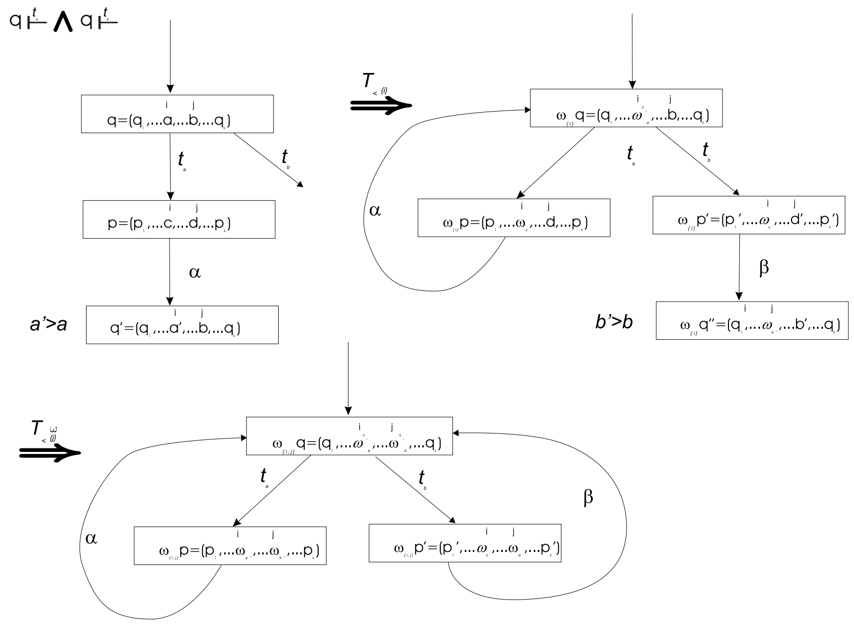

Fig. 4. Path transformation creating independent ωs

There is a state q, having values a and b in its i-th and j-th coordinate respectively. There are also two transitions (vectors) say t_a and t_b that are firable (applicable) in q. An aplication of t_a at q followed by other transitions leads to a state $q' = (q1, ...a^{'}, ..., b, ...q_k)$, where $a < a^{'}$ is a new value of the i-th coordinate, and according to the algorithm which construts automaton M_w the transformation $T_{<_i}$ applies, that creates a loop labelled with the string $\tau_a = t_a\alpha$, that replaces the path leading from q to $q^{'}$ which is labelled with $\tau_a = t_a\alpha$ as well. The effect of the transformation is that the state $q^{'}$ collapses to q creating a new state $\omega_{\{i\}}q = (q1, ..., \omega, ..., b, ...q_k)$. Because we are interested in the data keeping information on the value of the i-th coordinate which has been replaced by ω, we suggest to keep the data as a part of the new 'value' of the coordinate. Another information which we are interested in is the value $\Delta_i = a^{'} - a$ which expresses the increase of the i-th coordinate after repetition of the loop $\tau_a = t_a\alpha$. After all we prefer to denote the new ω value of the i-th coordinate by $\omega_a^{\Delta_i}$, and to call such the ωcoordinate to be *independent* ωcoordinate. Another *independent* ωcoordinate can be created due to firing the sequencce of transitions $\tau_b = t_b\beta$ starting at the state $\omega_{\{i\}}q$. Notice that in the intermediate state p in the *i-th* coordinate so called overflowed ω_c cord apears. The case of creating dependent ωcords can be visualized in similar way. Due to space lack we have to skip that and we refer the reader to [7].

4 Analysis of creating ωcoordinates

It was already mentioned the importance of the least and minimal states with respect to (wrt) deadlock analysis of P/T net (sect.3.1). Now we return back to the problem with an aim to analyze in the depth the issue of the role least or minimal states will play in the reachability and deadlok analysis. The main problem here is to decide in any particular case of least and/or minimal state q^B_ρ whether it is reachable or not.

Let us consider that $q^B_{\rho,i} \in Base\rho$; then $q^B_{\rho,i} \subseteq \Re(N_0)$ and thus $q^B_{\rho,i}$ is a macrostate covering a set of reachable states which all have the same i-th coordinate, say a_i. Moreover, the macrostate $q^B_{\rho,i}$ is the macrostate consisting of minimal states wrt macrostate ρ. The nature of the fsa of the type M_w [6] guarantees nonemptiness of $q^B_{\rho,i}$. The latter guarantees an existence of at least one reachable minimal element belonging to the macrostate $q^B_{\rho,i}$.

The state q^B_ρ can be considered to be the least element of the poset S_ρ, provided it is a reachable state, otherwise it can serve as a lower bound for the reachable states- elements of S_ρ. Very often such the state q^B_ρ will be just *spurious* reachable state, and there will be a need for q^B_ρ to be proved its reachability. To underpin that assertion some kind of analysis should be introduced first.

The case analysis of different types of $n > 1$ ωcoordinates have been accomplished [7]. In every case there that has been proven, that either $Base\rho \subseteq \Re(N_0)$, or $q^B_\rho \in \Re(N_0)$. The case of n=2 ωcoordinates is quite simple. The case of $n \geq 3$ is more complicated. There is few typical situations in the case of n=3 ωcoordinates that shows the table below.

In the table the entry (1, 1 $\rightarrow$, 1) stands for a dependence of ωcord *dep* on *ow*, and the entry (1, $\leftarrow$ 1, 1) stands for a dependence of ωcord *dep* on *ind*.
We illustrate that only for two cases:(3,0,0).

Table:Case analysis for 3 ωcords			
	Type of ωcords		
$N^{\underline{0}}$	ind	dep	ow
1	3	0	0
2	2	1	0
3	2	0	1
4	1	0	2
5	1	1→	1
6	1	←1	1
7	1	←2	0
8	0	2→	1
9	0	1→	2
10	0	0	3

Table 1. Case analysis for 3 ωcords

The results on creating ωcoordinates for the case $||A|| \leq 3$ can be generalized to any $n \in \mathbf{N}$. The conclusion we have come to is that ωcoordinates can assume one of the following forms.

a) **single indices**

$$ind\omega cord \quad Bdep\omega cord \quad ow\omega cord$$

$$\omega_a^{\Delta_i} \qquad \omega_{\underline{b}}^{\overline{B,t_b,\Delta_j}} \qquad \omega_c$$
$$B \subseteq K$$

b) **multiple indices**

$$ind\omega cord$$

$$\omega_a^{\Delta_i} \qquad \omega_{a,\ \ a'}^{\Delta_i,-} \qquad \omega_{a,\ \ a',a''}^{\Delta_i,-,-}$$

$$Bdep\omega cord$$

$$\omega_{\underline{b}}^{\overline{B,t_b,\Delta_j}} \ \omega_{\underline{b,}\ \ b'}^{\overline{B,t_b,\Delta_j},-} \ \omega_{\underline{b,}\ \ b',b''}^{\overline{B,t_b,\Delta_j},-,-}$$
$$B \subseteq K$$

$$ow\omega cord$$

$$\omega_c \qquad \omega_{c,c'} \qquad \omega_{c,c',c''}$$

We propose to use a generalized form to represent ωcoordinates, and we will call it as form (f).

$$\omega\left[\begin{pmatrix} h_1, \cdots h_k \\ d_1, \cdots d_k \end{pmatrix}\right] \tag{11}$$

where

$$\begin{pmatrix} h_1 \\ d_1 \end{pmatrix} \in \left\{ \begin{pmatrix} \Delta_i \\ a \end{pmatrix}, \begin{pmatrix} \frac{A, t, \Delta_i}{b} \end{pmatrix} \right\}$$
$$\begin{pmatrix} h_i \\ d_i \end{pmatrix}_{i>1} \in \left\{ \begin{pmatrix} - \\ a \end{pmatrix}, \begin{pmatrix} \lambda \\ c \end{pmatrix} \right\} \tag{12}$$

Here λ stands for *empty* symbol. Basically, in the case of overflowed ωcoordinates we will use just ω_c instead of ω_c^{λ} or ω_c^{-}. In the case of independent and dependent ωcoordinates we will use instead of empty symbol '-' to visualize the correspondence of high and low indices.

In our consideration we will use shorthand notation for indexed ωcoordinates as

$$\omega\left[I_k\right] \text{ where } I_k = \begin{pmatrix} h_1, \cdots, h_k \\ d_1, \cdots, d_k \end{pmatrix}$$

The following results have been proven as far as the generalization of the process of indexed ωcoordinates is concerned:

1. the form (f) of ωcoordinates has been chosen correctly, and it will be preserved by any application of $T_{<_A^{\omega}}$ transformation, and
2. the procedure to obtain the set of minimal elements of the poset represented by a macrostate ρ- $Base\rho$, is determined by a choice of proper combination of low indices.

5 A case study and further analysis of ωcoordinates

In his paper [1] T.Murata studied two PNs (Fig. 5) with respect to discovering liveness or deadlock, based on the coverability graphs of Petri Nets, the structure that is widely used to represent the state space of their reachable states. He showed that two PNs having the identical coverability graphs differ what concerns of liveness or deadlock properties. In this section we will use our method based on the finite automaton M_w and the properties of ωcoordinates to demonstrate the power of the approach to discover safely the deadlock of Petri nets in general and it will be demonstrated by the example Petri nets by T. Murata.

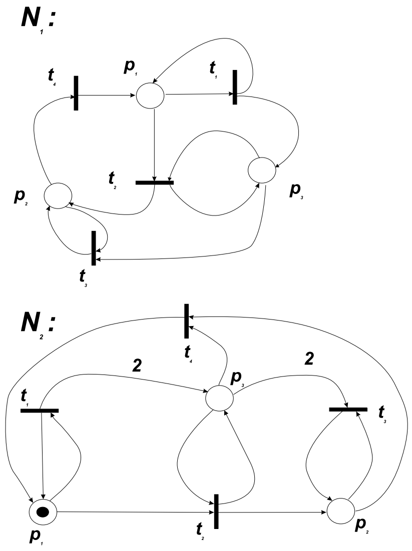

Fig. 5. Case study: Two Petri Nets with identical coverability graphs

Let us have a closer look at the two PNs. Comparing coverabiity graphs of the two PNs we can see they are indeed identical. Now we are going to apply the aproach based on the methodology developed, that is backed by our algorithm of constructing fsa of the type M_w (in some cases state diagrams of M_w and coverability graph coincide, but in some cases they look quite different). Construction of fsas of the type M_w fo the two nets can be seen in Fig. 6 and Fig. 7.

We can see that M_w automata are isomorphic, but they differ in ωcords as far as their indices are concerned.

Let us have a closer look at the M_w automata from that perspective. In M_w automaton of PN N_1 we have two macrostates: $\rho_1 = (1, 0, \omega_0^1)$ and $\rho_2 = (0, 1, \omega_{0,1})$. If we look at $\rho_1 = (1, 0, \omega_0^1)$ as at the poset, we can have the only minimal and thus the least (infimum) state

$$q_{\rho_1}^B = (1, 0, 0)$$

In the case of $\rho_2 = (0, 1, \omega_{0,1})$ we have the basis of this poset

$$Base(\rho_2) = \{(0, 1, 0), (0, 1, 1)\}$$

There are actually 2 minimal states.

Let us turn our atention to the net N_2. In M_w automaton of PN N_2 we have also two macrostates: $\rho_1 = (1, 0, \omega_{0,1}^{2,-})$ and $\rho_2 = (0, 1, \omega_{0,2})$. If we look at ρ_1 as at the poset, we can have here no infimum state, but we still have two minimal states that creates :

$$Base(\rho_1) = \{(1, 0, 0), (1, 0, 1)\}$$

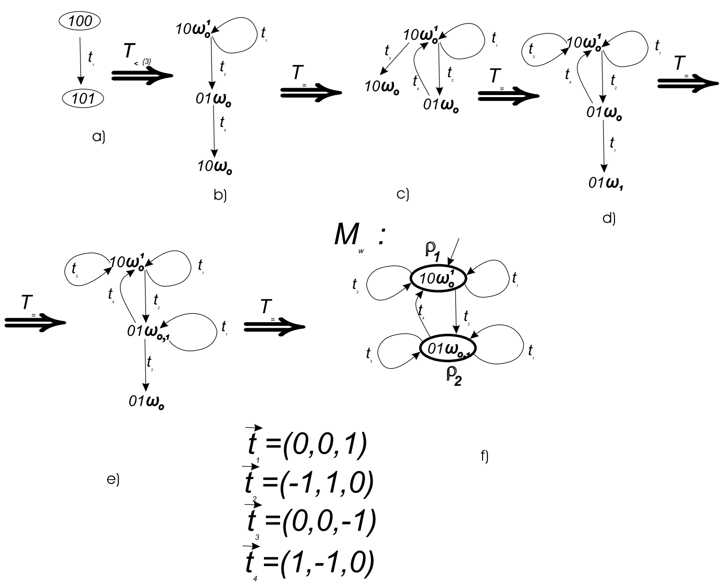

Fig. 6. M_w construction for the live Petri Net N_1 with inexed ωcords

If we look at ρ_2 as at the poset, we can have also here no infimum state, but we still have two minimal states that creates :

$$Base(\rho_2) = \{(0,1,0),(0,1,2)\}$$

5.1 Deadlock analysis

In [6] the deadlock problem is dealt with, based on the use of original RP algorithm. We wil use of the notion *'deadlock candidates'* states introduced there. The latter can be derived from the structure of PN in question.

Let us consider PN $N = (P,T,pre,post)$ and let $\overrightarrow{t} = -\,pre\overrightarrow{(p},t) + post\overrightarrow{(p},t)$. For any $t \in T$ and $p \in P$ we say

$$p\ covers\ t \Leftrightarrow_{df} pre(p,t) \neq 0 \tag{13}$$

In other words we are saying by (13) that

$$p\ covers\ t \Leftrightarrow_{df} t \in p^{\bullet} \tag{14}$$

The (13) and (14) simply mean that p is included in *t's* enabling. That is reasonable to define

$$C(p) = \{t \in T | p\ covers\ t\}$$

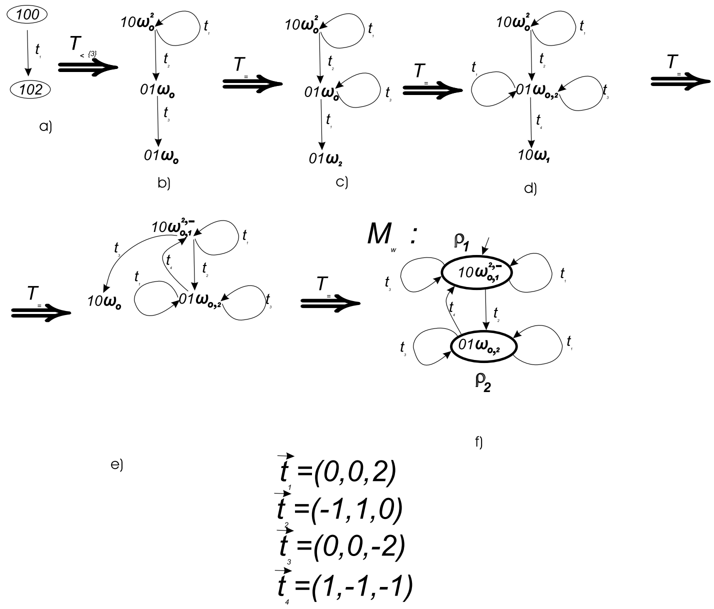

Fig. 7. M_w construction for the deadlock Petri Net N_2 with inexed ωcords

Obviously $C(p) = p^{\bullet}$. We are now looking for such a minimal subset C_i of P, that the union of the covers of places form the subset that will give the whole set T. We propose to call such the subset C_i *the minimal cover of* T. The notion minimal is connected with the number of places covering the set T. Notice, there can be more than one minimal cover of T.

We associate with each total cover of T -C_i, the set of deadlock markings, denoted as - $CanM_i$.

$$CanM_i = \{\ m \in \mathbb{N}^k | m \leq_{C_i} m_i, p \in C_i \Rightarrow m_i(p) \leq r-1, \\ r = min_{r_i}\ \{r_i | pre(p,t_i) = r_i, t_i \in p^{\bullet}\}\ \ \}$$

where $m \leq_{C_i} m_i \Leftrightarrow_{df} \forall p \in C_i : m(p) \leq m_i(p)$.
We can define the notion

$$CovT = \{C_i | C_i \subseteq P, C_i \text{ is total cover of T}\} \tag{15}$$

Based on the $CovT$ we may define now the overall set of dead markings

$$CanM = \Big\{ m \in \mathbb{N}^k | \exists C_i \in CovT : m \leq_{C_i} m_i, p \in C_i \Rightarrow m_i(p) \leq r-1, \\ r = min_{r_i}\ \{r_i | pre(p,t_i) = r_i, t_i \in p^{\bullet}\} \tag{16}$$

Notice that notions $CanM_i$ and $CanM$ are based only on the structure of PN in question and nothing is known about the reachability of the states contained there. That is why we have used to call them *'deadlock candidates'*, or *potential* deadlock states. Any of such the states becomes real deadlock provided it is reachable, the issue connected with dynamic aspect of the PN in question.
Now we are prepared to formalize the procedure, based on the method developed so far, as far as deadlock analysis is concerned. We do it in the form of a procedure.

DA(N_0):Algorithm for doing the deadlock analysis of Nets (P/T nets).
Input:
Petri Net (P/T net) $N_0 = (P, T.pre, post, m_0)$, of the type M_w of PN N_0
Output:
yes, if $D(N_0) \neq \Phi$
no, if $D(N_0) = \Phi$
Method:
Method is based on the results achieved in the analysis of nature of ωcoordinates, that occur in ω macrostates $\rho = \omega_A q$. Approach is based on interpretation of any such ω macrostate $\rho = \omega_A q$ as a representaion of the poset of reachable states in PN N_0, having minimal or the least elements (MoL states). The special procedure *CanM(*N_0 *)* is used for creation of the set of potential deadlocks of PN N_0.
Body of the algorithm:

begin
$D(N_0) \leftarrow \Phi$; */$D(N_0)$ - the set of MoL deadlock states/*
$D \leftarrow \Phi$; */D - variable - buffer for the actual set of MoL deadlock states/*
$C \leftarrow CanM(N_0)$; */D - variable - container of potential deadlock states /*
$\mathcal{S} \leftarrow Q^\omega$; */$Q^\omega$ - the set of states of fsa $M_w = (Q^\omega, W, \delta, \rho_0)$ W-the set of vectorized transitions,δ-transition function,ρ_0 containing m_0/*
while $\mathcal{S} \neq \Phi$ **do**;
begin
choose $\rho \in \mathcal{S}$;
$MoL \leftarrow Base\rho$;
$\mathcal{S} \leftarrow \mathcal{S} - \{\rho\}$;
if $C \cap MoL \neq \Phi$ **then** $D \leftarrow D \cup C \cap MoL$;
end
if $D = \Phi$ **then return *NO***
else $D(N_0) \leftarrow D$;
return *YES:*$D(N_0)$
end

The algorithm $DA(N_0)$ guarantees all MoL deadlocks will be found out and delivered as the set $D(N_0)$. Actually that can be considered as solving the problem of discovering presence or absence of deadlock states in the PN N_0.

5.2 Case study continued

We can now continue to analyze state diagrams of the two PNs. We wil use the results of previous section and particularly the result of the Lemma **??**.
So according to that we have to calculate now total covers for the two PNs.
In the tables below there are calculated both:the minimal total cover of T and pre-set for the net N_1.

Total Cover of T for PN N_1					
	t_1	t_2	t_3	t_4	Total Cover of T
p_1	$\vee$	$\vee$			$C_1 = \{p_1, p_2\}$
p_2			$\vee$	$\vee$	
p_3		$\vee$	$\vee$		

Function *pre* for PN N_1			
pre	p_1	p_2	p_3
t_1	1		
t_2	1		
t_3		1	
t_4		1	

$$
\begin{aligned}
Can\,M = Can\,M_1 &= \{0, 0, \omega\} \\
inf(1, 0, \omega_0^1) &= (1, 0, 0) \notin CanM \\
inf(0, 1, \omega_{0,1}) &\quad \text{nejestvuje} \\
Base((0, 1, \omega_{0,1})) &= \{(0, 1, 0), (0, 1, 1)\} \cap CanM = \{(0, 0, \omega)\} = \Phi
\end{aligned}
$$

So we jump to the conclusion that PN N_1 does not contain any deadlock!

Now we are going to turn our attention to the PN N_2. First we calculate minimal total cover of PN N_2 and pre-set for the PN N_2.

Total Cover of T for PN N_2					
	t_1	t_2	t_3	t_4	Total Cover of T
p_1	$\vee$	$\vee$			$C_1 = \{p_1, p_2\}$
p_2			$\vee$	$\vee$	$C_2 = \{p_1, p_3\}$
p_3		$\vee$	$\vee$	$\vee$	

Function *pre* for PN N_2				
pre	p_1	p_2	p_3	
t_1	1			
t_2	1		1	
t_3		1	2	
t_4		1	1	

$$
\begin{aligned}
CanM_1 = \{(0, 0, \omega)\}, CanM_2 &= \{(0, \omega, 0)\} \\
CanM &= \{(0, 0, \omega), (0, \omega, 0)\} \\
Base((1, 0, \omega_{0,1}^{2,-})) &= \{(1, 0, 0), (1, 0, 1)\} \\
Base((0, 1, \omega_{0,2})) &= \{(0, 1, 0), (0, 1, 2)\} \\
Base((1, 0, \omega_{0,1}^{2,-})) &\cap CanM = \Phi \\
Base(0, 1, \omega_{0,2}) \cap CanM = \{(0, 1, 0), (0, 1, 2)\} &\cap \{(0, 0, \omega), (0, \omega, 0)\} = \{(0, 1, 0)\}
\end{aligned}
$$

So we may jump to the conclusion that PN N_2 has indeed deadlock state $\{(0, 1, 0)\}$!

6 Conclusion

The issue of deadlock analysis is important for the development of discrete state-based systems. The method of discovering a presence, or an absence of deadlocks in the system coined and demonstrated in the paper is based on the study of the properties of the automaton M_w. We should mention that the results presented in the paper manifest the depth and the vitality of the new method to deal with the issue of reachability in Petri Nets, particularly the part which was connected with the study of the algebraic properties of interpretations of the automaton of the type M_w. In [6] there are some results presented on the nature of that interpretation. The automaton M_w bears some similarity with coverability graphs used in Petri Nets, but as it was proven, it is more powerful to deal with deadlock analysis. Beside of that, the structure of the automaton M_w plays the central role in reachability analysis of the systems (represented via PN) with infinite state space [6]. The most important property of M_w is its reusability for reachability analysis of the PN in question wrt to any other state, not only wrt to that it was constructed for initially. On the other side it turns out that one automaton of the type M_w, say $\mathcal{M}$ can serve in that role for whole class of PNs with the same number of places and some structure that induces corresponding set of transitions wich are consistent with the $\mathcal{M}$ structure. There is still another way how the M_w structure can be used. The fsa $\mathcal{M}$ can be thought of as a couple $\mathcal{M}$=(M,$\mathcal{I}$), where M and $\mathcal{I}$ stand for basic fsa without interpreted states and interpretation respectively. For any $k \in \mathbb{N}$- the number of places and given structure of basic fsa M we can construct corresponding interpretation $\mathcal{I}$ consistent with M. By that virtue, the same applies wrt doing deadlock analysis.

Due to space we have not dealt with the issue of modification of the algorithm of $\mathcal{M}$ construction, and also many details and proofs have been skipped. There can be found in [7]. At the workplace of the author there has been environment - termed as mFDTE [16], developed. The results will be implemented in the environment. The latter combines three formal descriptions of systems: Petri Nets, process algebra, and B AMN. The latter substantiate the acronym mFDTE-multi Formal Description Techniques Environment. More details can be found in [16].

Acknowledgments. This work was supported by the Slovak Research and Development Agency grant on Modelling, simulation and implementation of GPGPU-enabled architectures of high-throughput network security tools,under the contract No. APVV-0008-10. Further it was also partially supported by the grants No. 1/3140/06 of the VEGA-The Scientific Grant Agency of Slovakia, NATO CLG982698 grant, APVV grants No.APVV-0073-07, and SK-UA-0026-07.

References

1. Murata, T.: Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE. 77 (1989)

2. Xinning Y., JiantaoZ., and Xiaoyu S.: On Reachability Graphs of Petri Nets. Computers & Electrical Engineering. 29, 263–272 (2003)
3. Hudák, Š.: Extensions to Petri Nets. Habilitation Thesis (in Slovak), TU Košice, 107 p. (1980)
4. Hudák, Š.: The Recursive Decidability of the Reachability Problem for Vector Addition Systems. ASMM/84, Computing Laboratory, The University of Newcastle upon Tyne, England, 78 p. (1981)
5. Hudák, Š.: On the Reachability Problem for Vector Addition Systems. Proceedings of The Second European Workshop on Application and Theory of Petri Nets, Bad Honnef, Germany, 38 p. (1981)
6. Hudák, Š.: Reachability Analysis of Systems Based on Petri Nets.elfa s.r.o. Košice, ISBN 80-88964-07-5, 272 p. (1999)
7. Hudák, Š.: Deadlock Analysis of Place/Transition Nets.Manuscript, DCI FEI TU Kosice, 79 p. (2010)
8. Reisig, W.: Petri Nets: An Introduction. Springer Verlag, Heidelberg (1985)
9. Sacerdote, G.S., Tenney, L.: The Decidability of the Reachability Problem for Vector Addition Systems (preliminary version). Proceedings of the 9th Ann. ACM Symposium on Theory of Computing, New York (1977)
10. Muller, H.: On the Reachability Problem for Persistent Vector Replacement Systems. Computing Suppl. 3, 89–104. In: Knodel, H., Schneider, H.J. (eds.) Parallel Processes and Related Automata (1981)
11. Mayr, E.W.: An Algorithm for the General Reachability Problem in Petri Nets. SIAM J. of Computing. 13 (1984)
12. Kosaraju, S.R.: Decidability of Reachability in Vector Addition Systems. In Proc. 14th Annual ACM STOC, 267–281 (1982)
13. Valk, R.: Generalizations of Petri Nets. In: Gruska, J., Chytil, M. (eds.) MFCS 1981. LNCS vol.118, pp. 140–155 (1981)
14. Karp, R.M., Miller, R.E.: Parallel Program Schemata. Journal of Computer and System Sciences. 3, 147–195 (1969)
15. Keller, R.M.: Vector Replacement Systems: A formaism for Modeling Asynhronous Problms. Technical Report 117, Princeton University (1972)
16. Hudák, Š. et al.: A Support Tool for the Reachability and Other Petri Nets- Related Problems and Formal Design and Analysis of Discrete Systems. Problems in Programming. 20, 613-621, ISSN 1727-4907 (2008)
17. Hudák, Š.: De/compositional Reachability Analysis. Journal of Electrical Engineering, CS ISSN 0013-578X. 45, 424-431 (1994)
18. Stone, H.S.: Discrete Mathematical Structures and Their Applications. Science Research Associates Inc., Chicago, 402pp. (1973)
19. Best, E.: Semantics of Sequential and Parallel Programs. Prentice Hall Europe, ISBN 0-13-460643-4, 352pp. (1996)
20. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: Formal Languages and Their Relations to Automata. Addison Wesley, Reading, Mass (1969)
21. Lano, K.: The B Language and Method. A Guide to Practical Formal Development. Springer-Verlag London, ISBN 3-540-76033-4, 232pp (1996)