Robert DYLEWSKI[1]

University of Zielona Góra (1)

Assigning the cover of Petri nets with subnets of the automatic type

Abstract. *In the article the problem of the decomposition of a concurrent digital automaton for its state machine components is considered. A new method of cover is proposed with safe, live and reversible Petri nets. In order to determine basic and correct P-invariants which generate strongly connected subnets of the state machine type the suitable linear programming problem based on the matrix of marking corresponding to the global states of the net is used. The operating of the method for the example Petri nets is illustrated and benefits in relation to other methods are presented.*

Streszczenie. *W artykule rozważa się problem dekompozycji współbieżnego automatu cyfrowego na jego sekwencyjne składowe automatowe. Zaproponowano nową metodę pokrycia bezpiecznych, żywych i powracanych sieci Petriego. Do wyznaczania p-inwariantów podstawowych i poprawnych, generujących mocno spójne podsieci typu maszyna stanów używa się odpowiedniego zadania programowania liniowego, bazującego na macierzy znakowań odpowiadającej stanom globalnym sieci. Zilustrowano działanie metody dla przykładowych sieci Petriego i przedstawiono zalety w stosunku do innych metod. (**Wyznaczanie pokrycia sieci Petriego podsieciami typu automatowego**).*

Keywords: cover of Petri nets, *P*-invariants, state machine components, linear programming.
Słowa kluczowe: pokrycie sieci Petriego, *P*-inwarianty, sekwencyjne składowe automatowe, programowanie liniowe.

Introduction

For the description of the problem of the decomposition of a concurrent digital automaton for state machine components it is possible to apply Petri nets. Assigning strongly connected subnets of the automatic type comes to finding the invariants of the net places (*P*-invariants) which have suitable properties.

Petri net is a three-tuple $N = (P, T, F)$, where $P = \{p_1, p_2,..., p_m\}$ is a set of places, $T = \{t_1, t_2,..., t_n\}$ is a set of transitions, whereas $P \cap T = \varnothing$ and $P \cup T \neq \varnothing$, and $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation between places and transitions. Petri net can be presented in the form of a bepartible directed graph where nodes belong to P and T sets, whereas arcs belong to F set (see e.g. fig. 1). *Marked Petri net* is a four-tuple of $PN = (P, T, F, M_0)$, where $N = (P, T, F)$ is Petri net, whereas $M_0 : P \rightarrow \{0, 1, 2, 3, ...\}$ is the initial marking [1].

Transition t is called *enabled* in M marking if its each input place p contains at least one token, i.e. for each $p \in {}^*t = \{p \in P : (p,t) \in F\}$ is $M(p) \geq 1$. Transition which is enabled can be *fired*, which is based on taking one token from each input place $p \in {}^*t$ and giving one token for each output place $p \in t^* = \{p \in P : (t, p) \in F\}$. Marking M_2 is *reachable* from M_1 marking if there is a sequence of transitions $t_{i1}, t_{i2}, ..., t_{ik}$, after firing of whose there appears M_2 marking from M_1 marking. For initial M_0 marking and reachable markings from M_0 it is possible to create so called the *marking graph*, also called the reachability graph, where nodes correspond to reachable markings, and arcs correspond to fired transitions conducting one marking reachable in others (see e.g. fig. 2) [2].

The most important dynamic properties (dependent on the initial marking) of Petri net are safety, liveliness and reversibility. Marked Petri net $PN = (P, T, F, M_0)$ is called *safe* if for each marking M reachable from M_0 and for each $p \in P$ is $M(p) \leq 1$. Transition t is called *live* in PN net if it can be fired at least once in the sequence of transitions for each M marking reachable from M_0. *PN* net is called *live* if each transition $t \in T$ is live in this net. Marking Petri net PN is called *reversible* if for each M marking reachable from initial marking M_0, marking M_0 is reachable from M (see e.g. [1]).

For example, the interpreted steering Petri net (see details in [3]) should fulfil all three dynamic properties. The places of the net are treated as the local states of the modelled transition system, and reachable markings of the nets represent its global states. Transitions describe events taking place in a discrete system.

Marking equation of Petri nets

For the description of the structure of Petri nets as well as the investigation of structural and dynamic properties the *incidence matrix* $C_{m\times n}$ is applied, where rows are connected with places, and columns with transitions (see e.g. [1, 2, 3]). Element c_{ij} of the matrix C is connected with place p_i and transition t_j and is defined in the following way:

(1) $$c_{ij} = c_{ij}^{+} - c_{ij}^{-},$$

where: $$c_{ij}^{+} = \begin{cases} 1, \text{when } (t_j, p_i) \in F \\ 0, \text{when } (t_j, p_i) \notin F, \end{cases}$$

and $$c_{ij}^{-} = \begin{cases} 1, \text{when } (p_i, t_j) \in F \\ 0, \text{when } (p_i, t_j) \notin F. \end{cases}$$

A pair (p, t) is called a *self-loop* if p is both in input and output of transition t. For the net without a self-loop, matrix C defines unambiguously the structure of the net:

$c_{ij} = +1$ means that place p_i is the output for transition t_j, that is $(t_j, p_i) \in F$,

$c_{ij} = -1$ means that place p_i is the input for transition t_j, that is $(p_i, t_j) \in F$.

Using the incidence matrix C it is possible to give a necessary condition of reachability of marking M_2 from marking M_1. To make it easier, marking M is noted in the form of a column vector $M = [M(p_1), M(p_2),..., M(p_m)]^{\mathrm{T}}$.

If marking M_2 is reachable from M_1, so called *marking equation*

(2) $$M_2 = M_1 + C \cdot x,$$

has a nonnegative integer solution $\bar{x} = [x_1, x_2, ..., x_n]^T$, where value x_j defines how many times transition t_j is fired in the sequence of successive transitions conducting marking M_1 to M_2 [2]. *A characteristic vector* $\bar{x}$ defines which transitions and how many times have to be fired but unfortunately it does not give the order. The equation (2) brings a necessary condition to the reachability of marking M_2 from marking M_1. Therefore if the system of equations (2) is in contradiction for M_2, then M_2 is not reachable from M_1. The reverse implication is not true in a general case. It can happen that the equation (2) has for a particular M' a nonnegative integer solution x', but M' is not reachable from M_0 (that is none sequence of transitions conducting marking M_0 in M' corresponds to vector x'). Such a marking M' is called *spurious*. [2, 4].

_P_-invariants and their properties

In the further part of the article Petri nets, without a self-loop, which are live, safe and reversible are considered.

The equation of markings can also be used to examine the properties independent from the initial marking, but only from the structure of Petri nets.

Vector $\bar{y} = [y_1, y_2, ..., y_m]^T$ with nonnegative and integer coordinates is called *P-invariant* if it is a solution of the system of equations [1]:

(3) $$y^T \cdot C = \mathbf{0}.$$

Non trivial solutions $\bar{y} \neq \mathbf{0}$ become interesting.

The system of linear equations (3) can be solved by bringing the completed matrix of the system $\left[C^T, \mathbf{0}\right]$ to the base form and then assigning a general solution and base solutions (see e.g. [5]).

Assigned *P*-invariants can be applied, among the others, for the decomposition of a concurrent digital automaton as the sets of places determining its potential sequential automatic components [6].

If for some $\bar{y}$ the equation (3) is satisfied, then for vector $\bar{x}$ as well as markings M_0 and M satisfying the equation (2) becomes

(4) $$\bar{y}^T \cdot M = \bar{y}^T \cdot M_0 + \bar{y}^T \cdot C \cdot \bar{x} = \bar{y}^T \cdot M_0.$$

The equation is satisfied (4) regardless of value $\bar{x}$, thus for each marking M reachable from M_0.

For instance, for the net from example 1. one of *P*-invariants is described with $\bar{y} = [1, 0, 1, 0, 1, 1]^T$. It means that for each place M reachable from M_0 is $M(p_1) + M(p_3) + M(p_5) + M(p_6) = 1$.

If y' and y'' are *P*-invariants, then for integers $\alpha, \beta > 0$ vector $y = \alpha \cdot y' + \beta \cdot y'' \geq \mathbf{0}$ satisfies the equation (3), and thus is also *P*-invariant [3]. *P*-invariant $\bar{y} \neq \mathbf{0}$ is called *basic* if it is not a sum of other *P*-invariants different from $\mathbf{0}$.

P-invariant is called *correct* (meets Petri-Holt demand about the unity of place and time [7]) if it has exactly one representative in each reachable marking (a global state) [8]. For instance, *P*-invariant $\bar{y} = [1, 0, 1, 0, 1, 1]^T$ is basic and correct for the marking net from example 1 (see Examples).

On the basis of the equation (3) it is possible to determine *P*-invariants using ILP technique (integer linear programming), e.g. taking a linear criterion and the system of linear limitations in the form of (3) as well as $y \geq \mathbf{0}$, $y \in Z^m$ and $y_1 + y_2 + ... + y_m \geq 1$ (see e.g. [9, 10]). In the work of [11] other way of assigning *P*-invariants was proposed using decomposition of Petri nets on so called functional subnets.

The direct application of the incidence matrix C to assign *P*-invariants has however the following defects:

(a) it is not possible to avoid condition $y \in Z^m$ as the equation (3) can have non-integer solution , e.g. if $\bar{y} \neq \mathbf{0}$ is *P*-invariant, then $\bar{y}/k$ for $k \in N$ satisfies the equation (3), but it does not have to belong to Z^m;

(b) as a solution it is possible to obtain *P*-invariant, which is not basic (is a sum of basic *P*-invariants);

(c) as a solution it is possible to obtain *P*-invariant, which is not correct.

Assigning _P_-invariants

For the decomposition of nets for state machine components only these *P*-invariants are used which are basic and correct.

For marking M_k reachable from M_0 if *P*-invariant $\bar{y}$ is basic and correct, then appears an equation [8]:

(5) $$M_k^T \bar{y} = 1.$$

Let $MM_{m \times r}$ mean the *matrix of markings*, whose columns are vectors corresponding to reachable markings, where *r* is a number of reachable markings, altogether with M_0. Using (5) there is a condition:

If vector $\bar{y}$ is basic and correct *P*-invariant, then

(6) $$MM^T \bar{y} = \mathbf{1}.$$

The solutions of the above system of equations become basic and correct *P*-invariants if $y \in Z^m$ or vectors, which correspond to incorrect *P*-invariants or the sum of basic *P*-invariants (non base solution) if $y \notin Z^m$ (see example 3).

Using (6) it is possible to determine *P*-invariants, which have desired properties, are basic and correct, without the necessity of assuming that variables are integers. For instance, in order to assign *P*-invariant including the most (max) or the fewest (min) local states it is necessary to assign the solution of the problem in the form of:

(7) $$\begin{aligned} &\text{max (min)} && \mathbf{1}^T y \\ &\text{subject to} && MM^T y = \mathbf{1}, \\ & && y \geq \mathbf{0}. \end{aligned}$$

Applying directly only the incidence matrix and condition (3) it is impossible to assign $\max \mathbf{1}^T y$, because if y satisfies the system of equation $y^T \cdot C = \mathbf{0}$, then $y' = s \cdot y$, where $s \in \{2, 3, ...\}$ also satisfies this system.

Assigning the cover of Petri nets

Petri net, where $|{*t}| = |t{*}| = 1$ for each $t \in T$ is called *state machine* and is marked with SM. The subnet of Petri nets generated by basic and correct *P*-invariant is a component of SM type (SMC), i.e. a firmly coherent net SM (the subnet of the automatic type) [8, 10].

Let $II_{m\times q}$ mean the matrix, whose columns are vectors corresponding to the assigned (not necessarily all) basic and correct *P*-invariants, where q is the number of *P*-invariants.

The condition, for which a given set of basic and correct *P*-invariants could include the *cover* of the net with the subnets of the automatic type, can be presented in the form of:

there is $z \in \{0,1\}^q$ such that

(8) $$II \cdot z \geq \mathbf{1}.$$

If the condition (8) is satisfied for a given $\bar{z}$, then $\bar{z}(l)=1$ means that *P*-invariant I_l is taken for the cover of the net, $\bar{z}(l)=0$ means that *P*-invariant I_l is not taken for the cover of the net. If however the system of inequalities (8) has not a solution, it means that the set of *P*-invariants with the matrix II does not include the cover of the net.

Using (8) it is possible to assign the cover of the nets, without the necessity of assigning all basic and correct *P*-invariants. If the matrix II represents all *P*-invariants, which are basic and correct, then in order to cover the net it is enough to use as many *P*-invariants from the matrix II as there are local states in the global state with the biggest number of elements.

Algorithm 1. presents the proposal of assigning the cover of Petri net with subnets of the automatic type. In order to assign proper SMC the subproblems (7) are used.

The following markings have been accepted:

i – number of place; j – number of *P*-invariant; q – number of assigned *P*-invariants; MM – matrix of markings; II – matrix of assigned *P*-invariants ($II(:,j)$ – j-th column of matrix II); gM – maximum number of local states in global states; zI – vector, whose i-th coordinate informs how many times place p_i has been covered; z – vector, whose j-th coordinate informs if j-th *P*-invariants from matrix II is taken to cover the nets ($z(j)=1$), or not ($z(j)=0$).

Algorithm 1.

```
K.O.
zI = [0,...,0]^T ; II = [ ]; q = 0 ; gM = max(sum(MM));
K.1.
FOR i = 1 TO m DO
    IF zI(i) = 0 THEN
        Assign a base solution y for problem {(7), y(i) = 1};
        IF y ∈ Z^m THEN
            II = [II, y]; q = q + 1 ; zI = zI + y ;
        ELSE
            BREAK;
        END;
    END;
END;
z = [1,...,1]^T ;
IF q > gM THEN
    GO TO K.2.
ELSE
    STOP;
END;
K.2.
j = 1; sz = q;
WHILE (sz > gM) & (j < q) DO
    zIp = zI − II(:, j) ;
    IF zIp ≥ 1 THEN
        zI = zIp; z(j) = 0; sz = sz − 1 ;
    END;
    j = j + 1;
END;
```

If the problem in K.1. does not have a solution for a given i, then the net does not have basic and correct *P*-invariants covering i-th place. If in the solution y of the problem in K.1. there are non-integer coordinates, then *P*-invariant corresponds to the obtained vector is not correct; that is does not have SMC covering place i (see example 3).

After completing the step K.1. in the columns of matrix II there are the assigned basic and correct *P*-invariants. If the number of column sz (*P*-invariants) is bigger than gM, then not all assigned *P*-invariants are necessary to cover the net. In such a situation step K.2. is conducted, where redundant *P*-invariants are assigned. Eventually, in vector z there is the information, which of the assigned *P*-invariants are taken to cover the net ($z(j)=1$), and which ones not ($z(j)=0$), while $zI = II \cdot z \geq \mathbf{1}$.

Examples

In this part three essentially different examples of Petri nets have been presented, for which the algorithm described in the previous point has been tested. In all examples the nets are live, safe and reversible.

Example 1.

The example has been applied in [3] (p. 35). In fig. 1. Petri net has been presented, and in fig. 2. its reachability graph.

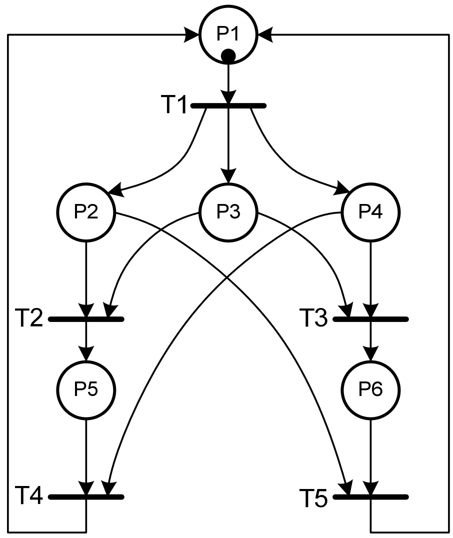

Fig. 1. Petri net for the example 1.

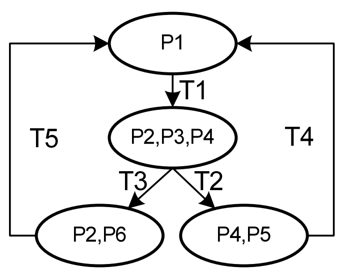

Fig. 2. Reachability graph for the net from example 1.

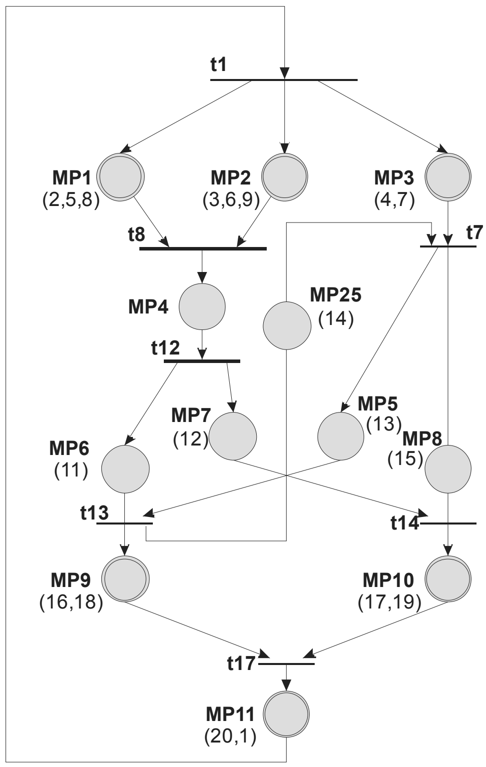

Fig. 3. Petri macronet for the example 2.

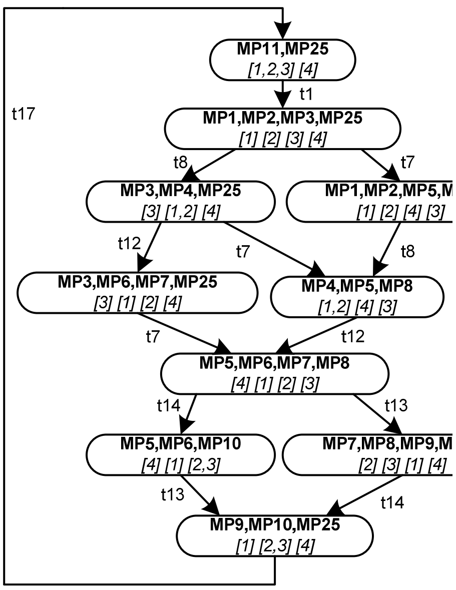

Fig. 4. Reachability graph for the macronet from example 2.

The reachable marking matrix corresponding to the reachability graph from fig. 2 has the form of:

$$MM^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Using algorithm 1 (variant with maximization in (7)) the matrix of *P*-invariants has been obtained in K.1.:

$$II^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

All three *P*-invariants are necessary to cover the nets ($gM = 3$). Therefore $z = [1, 1, 1]^{\mathrm{T}}$ and $zI = II \cdot z = [3, 1, 1, 1, 2, 2]^{\mathrm{T}}$.

Example 2.

The example comes from [12], it is also examined in the work of [13], the macronet is presented in fig. 3, and reachability graph in fig. 4. In the initial marking places MP11 and MP25 have been marked.

The matrix of reachable markings corresponding to the reachability graph from fig.4 has the form of:

$$MM^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

In the following matrix all basic and correct *P*-invariants for the examined net and the initial marking have been presented:

$$P^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Let I_j mean j-th *P*-invariant from matrix *P* (j-th row in matrix P^{T}), $j = 1, 2, \ldots, 7$. Using Algorithm 1 (variant with maximization in (7)) *P*-invariants $II = [I_7, I_6, I_3, I_5, I_4]$ have been obtained in K.1. As $gM = 4$ and $q = 5$ step K.2 is feasible, in which a redundant *P*-invariant I_5 is assigned. Thus $z = [1, 1, 1, 0, 1]^{\mathrm{T}}$ and $zI = [1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 3, 1]^{\mathrm{T}}$. In case of minimization in (7), $II = [I_1, I_2, I_3, I_4]$ is obtained. As this time $gM = q$, then $z = [1, 1, 1, 1]^{\mathrm{T}}$ and $zI = [1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 3, 1]^{\mathrm{T}}$.

It is necessary to observe that whether basic *P*-invariant is correct depends on the initial marking. For the studied net there are also two basic *P*-invariants $I_8 = [1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1]^{\mathrm{T}}$ and $I_9 = [0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1]^{\mathrm{T}}$, which however are not correct with the initial marking. It appears for them $M_0^{\mathrm{T}} I_8 = M_0^{\mathrm{T}} I_9 = 2$. For the initial marking, in which places MP5 and MP11 are marked, *P*-invariant I_5 is not correct whereas all the rest are correct, including I_8 and I_9.

Example 3.

The example was presented in [2] (p. 36), was also used in the work of [14], Petri net is presented in fig. 5.

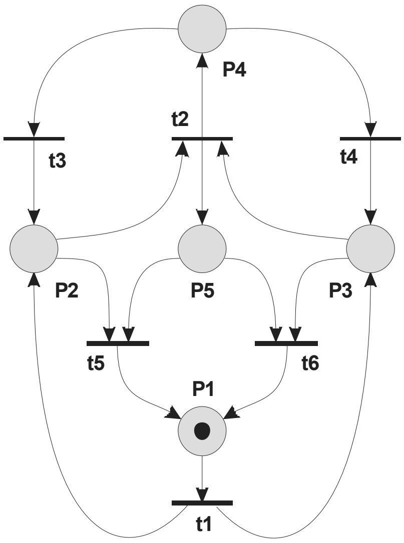

Fig.5. Petri net for the example 3.

For the net from fig. 5. the matrix of reachable markings has the form of:

$$MM^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

The only solution of the sub problem $\{(7), y(1) = 1\}$ for this net is vector $y = [1, 0.5, 0.5, 0.5, 0.5]^{\mathrm{T}}$. The remaining sub problems $\{(7), y(i) = 1\}$, for $i = 2, 3, 4, 5$ do not have solutions. Therefore the examined net has only one basic *P*-invariant $I = [2 \quad 1 \quad 1 \quad 1 \quad 1]^{\mathrm{T}}$. Moreover $MM^{\mathrm{T}} I = [2, 2, 2, 2, 2]^{\mathrm{T}}$, that is the only *P*-invariant has two representatives in each reachable marking, thus it is not correct. The net cannot be decomposed into the state machine components.

Conclusions

In the paper the method of assigning *P*-invariants of Petri nets and decomposition of the nets into components of the automatic type has been presented, based on the matrix of reachable markings. In this method only basic invariants of places are assigned, which also are correct with the applied initial marking. A solution of the suitable linear programming problem is used. Three examples of safe, live and reversible nets have been presented. For the nets fulfilling those three conditions, decomposition into SMC can be not possible (example 3), can be explicitly defined (see example 1), several variants of decomposition can be obtained as well (example 2).

Other methods of assigning the components of the automatic type are also being developed, which are based on reachable markings. The work of [15] presents the application of assigning the transversals of the subsets of concurrent places and the Gentzen sequent calculus, whereas in the work of [16] there is the application of colouring concurrency hypergraph.

The assigned components of the automatic type are applied to, among the others, coding the places of Petri nets or parallel decomposition of the nets [17, 6, 18]. The decomposition of Petri net for its state machine components simplifies the process of design and synthesis of a digital system.

REFERENCES

[1] Murata T., Petri Nets: Properties, Analysis and Applications, *Proceedings of the IEEE*, 77 (1989), 541-580

[2] Silva M., Introducing Petri nets, in *Practice of Petri Nets in Manufacturing*. London, U.K.: Chapman & Hall, (1993), 1–62

[3] David R., Alla H., *Petri Nets and Grafcet. Tools for modeling discrete event systems*. London: Prentice Hall, (1992)

[4] Dylewski R., Adamski M., Jabłoński J., Zastosowanie programowania liniowego do badania sieci Petriego, *Przegląd Elektrotechniczny*, 87 (2011), nr 11, 159-163

[5] Szpyrka M., *Sieci Petriego w modelowaniu i analizie systemów współbieżnych*. Warszawa: Wydawnictwa Naukowo-Techniczne, (2008)

[6] Bilinski K., Adamski M., Saul, J.M., Dagless E.L., Petri-net-based algorithms for parallel-controller synthesis, *Computers and Digital Techniques, IEE Proceedings*, 141 (1994), No. 6, 405-412

[7] Petri C.A., *Kommunikation mit Automaten*, Institut für Instrumentelle Mathematik, Schriften des IIM, n. 3, (1962)

[8] Dylewski R., Adamski M., Wyznaczanie P-inwariantów i pokrycia sieci Petriego z wykorzystaniem programowania liniowego, *Przegląd Elektrotechniczny*, 88 (2012), nr 3a, 165-169

[9] Lautenbach K., Linear algebraic techniques for place/transition nets, *LNCS*, 254 (1987), 142-167

[10] Pastor E., Cortadella J., Efficient Encoding Schemes for Symbolic Analysis of Petri Nets, *Proc. IEEE*, (1998), 790-795

[11] Zaitsev D.A., Compositional Analysis of Petri Nets, *Cybernetics and Systems Analysis*, 42 (2006), No. 1, 126-136

[12] Valette R., Etude comparative de deux outils de representation: Grafcet et reseau de Petri, *Le Nouvel Automatisme*, (1978), 337-382

[13] Adamski M., Kołopieńczyk M., Mielcarek K., Doskonała sieć Petriego w projektowaniu współbieżnych układów sterujących, *Pomiary Automatyka Kontrola* 57 (2011), nr 6, 656-660

[14] Carmona J., Cortadella J., Encoding Large Asynchronous Controllers With ILP Techniques, *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 27 (2008), 20-33

[15] Tkacz J., Adamski M., Wyznaczanie SM – pokrycia bezpiecznej sieci Petriego metodą komputerowego wnioskowania, *Pomiary Automatyka Kontrola* 57 (2011), nr 11, 1397-1400

[16] Wiśniewska M., Wiśniewski R., Zastosowanie kolorowania hipergrafów w procesie dekompozycji równoległej automatów współbieżnych, *Metody Informatyki Stosowanej* 23 (2010), nr 2, 151-157

[17] Adamski M., Węgrzyn M., Petri nets mapping into reconfigurable logic controllers, *Electronics and Telecommunications Quarterly*, 55 (2009), No. 2, 157-182

[18] Węgrzyn M., Wolański, P., Adamski M., Monteiro J.L., Coloured Petri net model of application specific logic controller programs, *ISIE '97., Proceedings of the IEEE International Symposium on Industrial Electronics*, 1 (1997), 7-11

***Authors**: dr Robert Dylewski, University of Zielona Góra, Faculty of Mathematics, Computer Science and Economterics, ul. Licealna 9, 65-417 Zielona Góra, E-mail: R.Dylewski@wmie.uz.zgora.pl.*