

# Using linear programming and the Petri net structure for deadlock prevention in sequential resource allocation systems

F. Tricas, F. García–Vallés, J.M. Colom, J. Ezpeleta

Departamento de Informática e Ingeniería de Sistemas

Universidad de Zaragoza

50018 Zaragoza.

{ftricas,gvalles,jm.ezpeleta}@posta.unizar.es

## Abstract

A new method for the deadlock prevention problem in concurrent systems where a set of processes share a set of common resources in a conservative way is proposed. The proposed approach can be applied to flexible manufacturing systems, modeled with Petri nets. The class of systems we will deal with eliminates some constraints imposed in previous research work: on–line routing decisions are allowed and no constraint is imposed about the number and type of system resources used at each processing step. In this paper, we present a set of important results related to the deadlock prevention problem in  $S^4PR$  nets. First, a liveness characterization is introduced, establishing how deadlocks can be studied in terms of circular waits. Second, we show how a circular wait situation corresponds to a particular marking related to a siphon of the Petri net model. Finally, this last characterization is used to obtain an iterative method that successively forbids deadlock related states, synthesizing the control necessary to ensure a final live behavior. The method can be implemented by means of the solutions of a set of integer linear programming problems.

## 1 Introduction

From an abstract point of view, the goal of the paper can be stated as follows: *we are looking for a control to be added to an automated system in such a way that the controlled system will be able to finish each production order.*

The point of view that we have adopted is the one that looks at the system as a Resource Allocation System (RAS). From this point of view, we

can see the system as composed of two main elements. *Processes*: each part that enters the system is a process. A process will be modeled as a token that moves through the Petri net. There is a set of types of processes: one type per each type of part to be produced. Each process is composed of a set of states, related to the different operations (either transformations or handling) to be carried out over the part that it models. Each state has associated a (multi-)set of resources needed for the corresponding processing step (including the buffering capacity to hold the part itself.) *Resources*: each physical element composing the cell (a machine, a buffer, a robot, a conveyor, a tool, etc.) is a resource. Each resource has a given capacity (the number of parts that, at a given time, the resource is able “to store/to be used by”.)

In this context, it is well–known that if a deadlock occurs at a given state, a circular–wait exists: a set of processes such that each process, in order to change its state (to advance to the next processing step), needs some resources which are, at that time, being used by some other process in the set. In order to deal with this problem, usually two approaches have been adopted. Both constrain the evolutions of the non–controlled system in such a way that no circular–wait state can be reached. But they establish the control in a slightly different way (even if, in some cases, it is difficult to find the border line between them): *Deadlock prevention*: off–line, obtaining a kind of “compiled” system. Once this has been done, the generated real–time controller incorporates the prevention policy and no additional control related to deadlock problems has to be considered (deadlock situations have been excluded a priori.) Different methods using this approach can

be found in [2, 6, 22, 11, 8, 21]. *Deadlock avoidance*: the additional control is established on-line, based on the notion of "safe state", and usually implemented as a one-step look-ahead method: assuming that the current state is safe (i.e., all the active processes can finish) and before allowing a process to evolve to a new state, it is checked that this last resulting state is also safe; if not, the transition is forbidden. This is a kind of "interpreted" implementation.

Since solving the problem for any given system is quite complicated, different partial models (corresponding to more restrictive application cases) have been solved. As stated above, these systems involve both, processes and resources. These partial models have been obtained imposing restrictions either on the class of processes considered or on the way that the resources can be used by a process at a given state. The main constraint related to the processes refers to the availability of different routings in the system; another important question is whether a part can choose different paths once it is in the system or not. The first feature is offered in some models, but most of them do not allow on-line decisions, and the path is fixed once the part selects one of the available routes [7, 9, 27]. Only a few studies [2, 6, 22, 12, 19, 3, 16] allow on-line decisions related to the part routing. The main constraint related to resources refers to the number and type of resources that were allowed to be used by a process at a given state. In most previous papers only one resource of a unique type (just the buffering capacity of the resource that holds the part) was allowed at each state of each process (the "Single-Unit RAS", as named in [15].) This constraint was relaxed in [22] and solved for the more general case in [20, 18, 2, 19, 16]. It is worth noting that [11, 25] provide solutions for unrestricted classes of Petri nets, with the only limitation that they transform the models to equivalent ordinary Petri nets.

Here we concentrate on those sequential RAS with routing flexibility and a multi-set of resources can be used at each processing step. Moreover, we also allow resources to be acquired/released in free (conservative) way.

Let us summarize in a very brief way the approach that we follow here. As in previous research work [6, 22] we are using Petri nets as formal models, and our approach is based on the structure of

the model: we try to get as much information as possible from the structure of the Petri net modeling the resource allocation system, avoiding in this way the state space explosion problem. In [2] a necessary condition for non-liveness based on siphons of the Petri net model of  $S^4PR$  nets is presented, and used to apply a deadlock prevention control policy. In [18], a more refined condition also based in siphons is given, allowing a more permissive solution for the same problem. Finally, [16] presents a liveness characterization based on the same structural components.

In this paper, we present a set of important results related to the deadlock control problem in  $S^4PR$  nets. First, the liveness characterization is presented, establishing how deadlocks can be studied in terms of circular waits. Second, we show how a circular wait situation corresponds to a particular marking related to a siphon of the Petri net model. The liveness characterization we provide is very similar to the one presented in [16]. Finally, this characterization is used to obtain an iterative method that successively forbids deadlock related states, synthesizing the control necessary to ensure a final live behavior. Other iterative methods are presented in [23, 2, 10, 11, 25] The method is based on the solution of integer linear programming problems, and it is implemented by means of the addition of some new places which behave as "virtual" resources, imposing some generalized mutual exclusions among some process states. Notice that [3, 16] also use mixed integer linear programming to test for the existence of deadlock problems.

The paper is organized as follows: Section 2 introduces the class of nets considered; Section 3 presents a liveness analysis for this class; Section 4 shows the proposed iterative control policy, which in Section 5 is applied to an example. Finally, in Section 6 some conclusions are presented.

## 2 A Class of Nets for Production Systems

The  $S^4PR$  class will be presented in a compact way. Check [24, 22] for a constructive, process-oriented approach.

**Definition 1** Let  $I_{\mathcal{N}}$  be a finite, non-empty, set of indices. A  $S^4PR$  is a connected generalized self-loop free Petri net  $\mathcal{N} = \langle P, T, C \rangle$  where:

1.  $P = P_0 \cup P_S \cup P_R$  is a partition such that:
  - (a)  $P_S = \bigcup_{i \in I_N} P_{S_i}$ , where for each  $i \in I_N$ ,  $P_{S_i} \neq \emptyset$ , and for each  $i, j \in I_N$ ,  $i \neq j$ ,  $P_{S_i} \cap P_{S_j} = \emptyset$ .
  - (b)  $P_0 = \bigcup_{i \in I_N} \{p_{0_i}\}$ .
  - (c)  $P_R = \{r_1, r_2, \dots, r_n\}$ ,  $n > 0$ .
2.  $T = \bigcup_{i \in I_N} T_i$ , where for each  $i \in I_N$ ,  $T_i \neq \emptyset$ , and for each  $i, j \in I_N$ ,  $i \neq j$ ,  $T_i \cap T_j = \emptyset$ .
3. For each  $i \in I_N$ , the subnet  $\mathcal{N}_{|(P_{0_i} \cup P_{S_i}, T_i)}$  is a strongly connected state machine such that every cycle contains  $p_{0_i}$ .
4. For each  $r \in P_R$  there exists a unique minimal  $P$ -Semiflow  $\mathbf{Y}_r \in \mathbf{N}^{|P|}$  such that  $\{r\} = \|\mathbf{Y}_r\| \cap P_R$ ,  $P_0 \cap \|\mathbf{Y}_r\| = \emptyset$ ,  $P_S \cap \|\mathbf{Y}_r\| \neq \emptyset$ , and  $\mathbf{Y}_r[r] = 1$ .
5.  $P_S = \bigcup_{r \in P_R} (\|\mathbf{Y}_r\| \setminus \{r\})$ .

In order to complete the modeling of the system dynamics, an initial marking must be provided. Tokens in a reachable marking can have different meanings: A token in a place  $p \in P_S$  will model an active process (a part being processed) whose state is modeled by means of place  $p$  (the part is at the state represented by this node.) Tokens in a place  $r \in P_R$  will model the available buffering capacity of resource  $r$  (*buffering capacity* will be used to represent either capacity or availability.) Markings need to represent states that have a physical meaning. In this sense, only *acceptable initial markings*, as defined in the following, will be considered. If the system is well defined, and its initial marking is "correct", all the markings that are reachable from it will represent possible states of the system, and will have a physical meaning.

**Definition 2** Let  $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, \mathbf{C} \rangle$  be a  $S^4PR$ . Then,  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$  with  $\mathbf{m}_0$  defined as follows is a  $S^4PR$  with an acceptable initial marking.

1.  $\forall p_0 \in P_0 . \mathbf{m}_0[p_0] > 0$ ;
2.  $\forall p \in P_S . \mathbf{m}_0[p] = 0$ ;
3.  $\forall p \in P_S . \forall r \in P_R . \mathbf{m}_0[r] \geq \mathbf{Y}_r[p]$ .

Let us remark the following facts: The initial marking of  $p_0$  (condition (1)) represents the maximal number of parts of the type modeled with this net that are allowed to be concurrently processed in the system. This initial marking can be chosen in such a manner that  $p_0$  becomes implicit [4], making this kind of systems suitable for the modeling of open systems (the maximal number of parts of the type modeled by the process Petri net concurrently processed is limited by the system itself via the capacities of the system resources.) No process is active at the initial state (condition (2).) The buffering capacity of each resource is such that each processing step can be executed when the isolated execution of one process is considered (condition (3).) These properties guarantee that when an acceptable initial marking is considered, a part can be processed in isolation, i.e. the system is well-defined.

For a given resource,  $r$ , and based on the minimal  $P$ -Semiflow  $\mathbf{Y}_r$ , the *holders of resource  $r$*  is defined as the set of process places using this resource.

**Definition 3** Let  $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, \mathbf{C} \rangle$  be a  $S^4PR$ . Let  $r \in P_R$ . The set of holders of  $r$  is the support of the minimal  $P$ -Semiflow  $\mathbf{Y}_r$  without the place  $r$ :  $\mathcal{H}_r = \|\mathbf{Y}_r\| \setminus \{r\}$ .

This definition can be extended in the natural way to sets of resources

Why the name "holder"? Let us consider the net in Figure 2 and the resource place  $R2$ . For it,  $\mathcal{H}_{R2} = \{P1\_2, P1\_3, P2\_1\}$ ; considering  $\mathbf{Y}_{R2} = R2 + 2 \cdot P1\_2 + 5 \cdot P1\_3 + P2\_1$ , each time a token enters place  $P1\_2$ , two tokens "disappear" from  $R2$  (maintaining the invariant relation), i.e., an active process in  $P1\_2$  is "holding" two capacity units of the physical resource represented by place  $R2$ .

In a  $S^4PR$  Petri net each transition has a unique input process state place (whose weight is equal to one) and zero or more input resource places. Extending the definitions presented in [27] for SURAS, and given a marking,  $\mathbf{m} \in \text{RS}(\mathcal{N}, \mathbf{m}_0)$ <sup>1</sup>,

a transition  $t$  is said to be *m-process-enabled* (or, *process-enabled at m*) if, and only if:  $\bullet t \in$

<sup>1</sup> $\text{RS}(\mathcal{N}, \mathbf{m}_0)$  denotes the reachability set of the net, and  $\text{PRS}(\mathcal{N}, \mathbf{m}_0)$  denotes the potential reachability set, obtained as the set of solutions of the state net equation. Check [13] for a good overview on Petri nets.

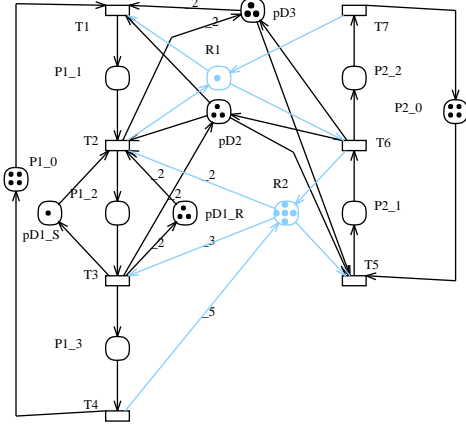


Figure 1: Control places proposed for the bad siphon  $D1$  of the net in Figure 2.

$P_S \neq \emptyset$ , and  $\mathbf{m}[\bullet t \cap P_S] \neq 0$ . That is, the transition is enabled by the corresponding process place (an active process is ready to fire, or a process is ready to be activated.) A transition is *m-resource-enabled* (or, *resource-enabled* at  $\mathbf{m}$ ) if, and only if:  $\forall r \in (\bullet t \cap P_R) \cdot \mathbf{m}[r] \geq \text{Pre}[r, t]$ . That is, no resource place is preventing the firing of  $t$ . Notice that a transition is enabled at marking  $\mathbf{m}$  if it is *m-resource-enabled* and *m-process-enabled*.

### 3 Liveness Analysis of $S^4PR$ Models

We are going to present a set of liveness characterizations for  $S^4PR$  nets. The first one (Theorem 4) does not use siphons, but concentrates on states where circular wait situations appear. The second one (Theorem 5), obtained from the first one, characterizes deadlock problems in terms of siphons and some related markings. Finally, the last one (Theorem 7) is also based on siphons, but establishes in a more clear way how deadlocked processes can be located around siphon components. All the proposed characterizations are equivalent to the one proposed in [14]. The main advantage of the one proposed in Theorem 7 is that it induces an efficient way of preventing deadlocks in  $S^4PR$  nets as it will be shown.

**Theorem 4 ([24])** Let  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$  be a marked  $S^4PR$ . The system is non-live if and only if there

exists a marking  $\mathbf{m} \in \text{RS}(\mathcal{N}, \mathbf{m}_0)$  such that the set of *m-process-enabled* transitions is non-empty and each one of these transitions is *m-resource-disabled*.

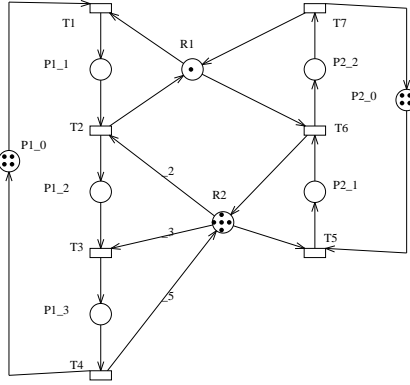
In the example of Figure 2, at marking  $\mathbf{m}_1 = 2 \cdot P1_2 + R1 + R2 + 2 \cdot P1_0 + 4 \cdot P2_0$ ,  $T3$  is the only *m<sub>1</sub>-process-enabled* transition, which is disabled by  $R2$ . Therefore, the system is non-live.

A marking  $\mathbf{m} \in \text{RS}(\mathcal{N}, \mathbf{m}_0)$  verifying the conditions of Theorem 4 will be called a *deadlocked marking*. The term *bad marking* will also be used. Theorem 4 relates non-liveness to the existence of a marking where active processes are blocked. Their output transitions need resources that are not available. These needed resources cannot be generated (released by the corresponding processes) by the system (the transitions are dead) because there exist a set of circular waits between the blocked processes. This concept of circular waits can be captured by the existence of a siphon (in Petri Net terms) whose resource places are the places preventing the firing of the process-enabled transitions. The following theorem shows that, when a bad marking as in Theorem 4 exists, a related siphon can be constructed; the reverse is also true. This establishes the bridge between behavior and model structure.

**Theorem 5 ([24])** Let  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$  be a marked  $S^4PR$ . The net is non-live if, and only if, there exists a marking  $\mathbf{m} \in \text{RS}(\mathcal{N}, \mathbf{m}_0)$ , and a siphon  $D$  such that  $\mathbf{m}[P_S] > 0$  and the firing of each *m-process-enabled* transition is prevented by a set of resource places belonging to  $D$ . Moreover, the siphon  $D$  is such that:  $D_R = D \cap P_R = \{r \in P_R \mid \exists t \in r^\bullet \text{ such that } \mathbf{m}[r] < \text{Pre}[r, t] \text{ and } \mathbf{m}[\bullet t \cap P_S] > 0\} \neq \emptyset$ ;  $D_S = D \cap P_S = \{p \in \mathcal{H}_{D_R} \mid \mathbf{m}[p] = 0\} \neq \emptyset$ ;

This theorem says that each one of these siphons is composed of resources with an insufficient marking for one of their input transitions that is process enabled, together with places that are holders of these resources and are empty at this marking.

In the example of Figure 2, at marking  $\mathbf{m}_1 = 2 \cdot P1_2 + R1 + R2 + 2 \cdot P1_0 + 4 \cdot P2_0$ , transition  $T3$  is dead and the siphon  $D1 = \{P1_3, P2_1, R2\}$  fulfills conditions stated in previous theorem:  $R2$  is preventing the firing of  $T3$ , which

Figure 2: A  $S^4PR$  with deadlock problems.

is process-enabled, and all the places in  $D1_S$  ( $D1_S = \{P1_3, P2_1\}$ ) are empty.

A siphon  $D$  and a marking  $\mathbf{m} \in \text{RS}(\mathcal{N}, \mathbf{m}_0)$  verifying the properties of Theorem 5 will be said to be a *bad siphon* and a *D-deadlocked marking*, respectively. For a given bad siphon  $D$ , in the following the next notation will be used:  $\forall p \in P_S \cdot \mathbf{Y}_{D_R}[p] = \sum_{r \in D_R} \mathbf{Y}_r[p]$ . Notice that  $\mathbf{Y}_{D_R}$  is the total amount of resource units belonging to  $D$  (in fact, to  $D_R$ ) used by each active process in  $p$ .

**Definition 6** Let  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$  be a marked  $S^4PR$ . Let  $D$  be a siphon of  $\mathcal{N}$ . Then,  $\mathcal{Th}_D = \mathcal{H}_{D_R} \setminus D_S$  is set of thieves of  $D^2$ .

The utility of this set will be understood later; for now, it should be clear that it represents places of the net that use resources of the siphon and do not belong to it.

The following liveness characterization establishes that when a  $S^4PR$  is not live, there exists a deadlocked marking such that all the active processes are “stealing” tokens from the set of resources of an associated siphon. This alternative characterization is useful to generate a deadlock prevention solution, allowing us to concentrate on siphons and their thieves, “forgetting” those active processes that are not related to the siphons, and

<sup>2</sup>We will use sometimes in the following  $\mathcal{Th}_{D_R}$  to show the relation among these two sets.

giving better computational results when controlling the system.

**Theorem 7 ([24])** Let  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$  be a marked  $S^4PR$ . The net is non-live if, and only if, there exists a siphon  $D$ , and a marking  $\mathbf{m}_D \in \text{RS}(\mathcal{N}, \mathbf{m}_0)$ , such that:

1.  $\mathbf{m}_D[P_S] > 0$ .
2.  $\mathbf{m}_D[P_S \setminus \mathcal{Th}_D] = 0$ .
3.  $\forall p \in \mathcal{Th}_{D_R}$  such that  $\mathbf{m}_D[p] > 0$ , the firing of each  $t \in p^\bullet$  is prevented by a set of resource places belonging to  $D$ .

This liveness characterization directly relates bad markings with system states in which all the active processes stay in thief places of a bad siphon. This will be specially useful when trying to control the system in order to ensure a live behavior since it shows that the potential problems are located around siphons.

#### 4 An iterative control policy

Let us present the proposed control policy, implemented in several steps. For this, the characterizations of Theorem 5 and Theorem 7 will be used, together with the net state equation.

Let us give some intuition about this using the reachability graph of the  $S^4PR$  of Figure 3, which is depicted in Figure 4. Reachable states can be classified into three categories: The first one (type 1) contains those markings from which  $\mathbf{m}_0$  is reachable. These markings are not involved in deadlock problems (the shadowed states in Figure 4.) The second class (type 2) is composed of those markings that are not  $D$ -deadlocked for any siphon, and such that  $\mathbf{m}_0$  is not reachable from them. Finally, the third class (type 3) is composed of those markings that are  $D$ -deadlocked for some siphon  $D$  (depicted as black boxes in the Figure.)

Since we are able to relate markings of type 3 with bad siphons, the control policy will be based on the addition of some restrictions related to bad siphons, trying to forbid as few states as possible, in order to prevent just the detected bad markings (markings of type 3.) Once a given marking has

been forbidden (by means of the addition of an adequate control place, which will impose firing restrictions), the resulting system still belongs to the  $S^4PR$  class. Therefore, the method can continue looking for a new bad marking, forbidding it, and so on, in an iterative way.

#### 4.1 Computation of deadlocked markings

The following proposition relates liveness with the existence of a solution for the proposed system of inequalities. The systems form a linear representation of a bad marking given a known bad siphon as introduced in the statement of Theorem 5.

**Proposition 8 ([24])** *Let  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$  be a marked  $S^4PR$ . The net is non-live if and only if there exist a siphon  $D$  and a marking  $\mathbf{m} \in \text{RS}(\mathcal{N}, \mathbf{m}_0)$  such that the following set of inequalities has, at least, one solution:*

$$\left\{ \begin{array}{ll} \mathbf{m}[P_S] > 0 \\ \forall t \in T \setminus P_0^\bullet, & \text{being } \{p\} = \bullet t \cap P_S, \\ & \mathbf{m}[p] \geq e_t \\ & e_t \geq \mathbf{m}[p]/\mathbf{sb}[p] \\ \forall r \in D_R, \\ \forall t \in r^\bullet \setminus P_0^\bullet, & \mathbf{m}[r]/\mathbf{Pre}[r, t] \geq e_{rt} \\ & e_{rt} \geq \frac{\mathbf{m}[r] - \mathbf{Pre}[r, t] + 1}{\mathbf{m}_0[r] - \mathbf{Pre}[r, t] + 1} \\ \forall r \in P_R \setminus D_R, & \forall t \in r^\bullet \setminus P_0^\bullet, e_{rt} = 1 \\ \forall t \in T \setminus P_0^\bullet, & \sum_{r \in \bullet t \cap P_R} e_{rt} < \\ & |\bullet t \cap P_R| + 1 - e_t \\ \forall t \in T \setminus P_0^\bullet, & e_t \in \{0, 1\} \\ \forall r \in P_R, & \forall t \in r^\bullet \setminus P_0^\bullet, e_{rt} \in \{0, 1\} \end{array} \right. \quad (1)$$

Let us make some comments about the variables used in these inequalities.  $\mathbf{sb}[p]$  denotes the structural bound of  $p$  [5]. The first inequality is the same as in Theorem 5 (there are active processes.) For each  $t \in T \setminus P_0^\bullet$ ,  $e_t$  indicates whether  $t$  is  $\mathbf{m}$ -process-enabled or not. If  $t$  is process-enabled,  $\mathbf{m}[p] \geq \mathbf{sb}[p]$ , so  $\mathbf{m}[p]/\mathbf{sb}[p] > 0$ , and, as  $e_t \in \{0, 1\}$ , it must be 1. Variable  $e_{rt}$  indicates whether  $t$  is enabled by  $r$  at  $\mathbf{m}$ . If  $t$  is enabled by  $r$  at  $\mathbf{m}$  ( $\mathbf{m}[r] \geq \mathbf{Pre}[r, t]$ ),  $\mathbf{m}[r]/\mathbf{Pre}[r, t] \geq 1$  and  $1 \geq (\mathbf{m}[r] - \mathbf{Pre}[r, t] + 1)/(\mathbf{m}_0[r] - \mathbf{Pre}[r, t] + 1) > 0$ ; therefore,  $e_{rt}$  must be 1. If  $t$  is not enabled by  $r$  ( $\mathbf{m}[r] < \mathbf{Pre}[r, t]$ ),  $\mathbf{m}[r]/\mathbf{Pre}[r, t] < 1$  and  $(\mathbf{m}[r] - \mathbf{Pre}[r, t] + 1)/(\mathbf{m}_0[r] - \mathbf{Pre}[r, t] + 1) \leq 0$ ; then,  $e_{rt}$  must be 0.

The existing bad siphons and their related bad markings need to be computed in order to control the system. Our next goal is to reformulate the above system of inequalities in order to be able to obtain a bad siphon, together with its related bad markings. The characterization presented in Theorem 5 allows a simple reformulation of these equations. To do that, an algebraic siphon characterization is necessary. In [1, 17] a characterization of this kind is given for traps. It is straightforward to adapt it to the case of siphons.

The result establishes that each solution of the following set of inequalities:  $v_p \leq \sum_{q \in \bullet t} v_q$ ,  $v_p \in \{0, 1\}$ ,  $\forall p \in P$ ,  $\forall t \in \bullet p$ , is a siphon (its components are those places whose associated variable  $v_p$  is 1.) As it will become clear later, this result is not adequate in this original form, and it has to be transformed into an equivalent form using negated logic (this approach is similar to the one proposed in [17] and also in [26].) A siphon is the set of places whose associated variables in the following set of inequalities is 0:  $v_p \geq \sum_{q \in \bullet t} v_q - |\bullet t| + 1$ ,  $v_p \in \{0, 1\}$ ,  $\forall p \in P$ ,  $\forall t \in \bullet p$ . In order to compute a bad siphon, conditions of Proposition 8 can be completed by the addition of the following equations: A set of constraints representing the siphon,  $v_p \geq \sum_{q \in \bullet t} v_q - |\bullet t| + 1$ ,  $v_p \in \{0, 1\}$ ,  $\forall p \in P$ ,  $\forall t \in \bullet p$ . A restriction that avoids the whole net as solution:  $\sum_{p \in P \setminus P_0} v_p < |P \setminus P_0|$ . A set of restrictions relating resource places that are avoiding the firing of a process-enabled transition and the siphon. For this,  $e_t$ ,  $e_{rt}$ , as in previous proposition are used together with the new introduced variables.

Let us show how this extension can be used to compute bad siphons and related bad markings.

The characterization introduced in this proposition will be not directly applicable to control the system, since a reachable marking is needed and we do not want to use reachable markings (our goal is to avoid the enumeration of the set of reachable markings.) Therefore, we will need to propose an alternative approach using the set of potentially reachable markings (markings obtained as solutions of the state equation.)

**Proposition 9 ([24])** *Let  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$  be a marked  $S^4PR$ . The net is non-live if and only if there ex-*

ist a siphon  $D$  and a marking  $\mathbf{m} \in \text{RS}(\mathcal{N}, \mathbf{m}_0)$  such that the system of inequalities (2) has a solution with  $D = \{p \in P_S \cup P_R \mid v_p = 0\}$ :

$$\left\{ \begin{array}{l} \forall p \in P \setminus P_0, \forall t \in \bullet p, v_p \geq \sum_{q \in \bullet t} v_q - |\bullet t| + 1 \\ \sum_{p \in P \setminus P_0} v_p < |P \setminus P_0| \\ \mathbf{m}[P_S] > 0 \\ \forall t \in T \setminus P_0^\bullet, \text{ being } \{p\} = \bullet t \cap P_S, \\ \mathbf{m}[p] \geq e_t \\ e_t \geq \mathbf{m}[p]/\text{sb}[p] \\ \forall r \in P_R, \\ \forall t \in r^\bullet \setminus P_0^\bullet, \mathbf{m}[r]/\text{Pre}[r, t] + v_r \geq e_{rt} \\ e_{rt} \geq \frac{\mathbf{m}[r] - \text{Pre}[r, t] + 1}{\mathbf{m}_0[r] - \text{Pre}[r, t] + 1} \\ e_{rt} \geq v_r \\ \forall t \in T \setminus P_0^\bullet, \sum_{r \in \bullet t \cap P_R} e_{rt} < |\bullet t \cap P_R| + 1 - e_t \\ \forall p \in P \setminus P_0, v_p \in \{0, 1\} \\ \forall t \in T \setminus P_0^\bullet, e_t \in \{0, 1\} \\ \forall r \in P_R, \forall t \in r^\bullet \setminus P_0^\bullet, e_{rt} \in \{0, 1\} \end{array} \right. \quad (2)$$

**Proposition 10 ([24])** Let  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$  be a marked  $S^4PR$ . If net is non-live, there exists a marking  $\mathbf{m} \in \text{PRS}(\mathcal{N}, \mathbf{m}_0)$ , with  $\mathbf{m}[P_S] > 0$ , and a siphon  $D$  such that the following system of inequalities has, at least, one solution with  $D = \{p \in P_S \cup P_R \mid v_p = 0\}$ :

$$\left\{ \begin{array}{l} \mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \bar{\sigma} \\ \mathbf{m} \geq 0, \bar{\sigma} \in \mathbb{Z}_+^{|T|} \end{array} \right. \quad (3)$$

System (2)

This proposition does not provide a complete characterization (as it was the case in Proposition 9.) It only provides a necessary condition for deadlock. The reason is the (possible) existence of spurious solutions: markings that are solution of the net state equation but are not reachable. This is not a problem when the objective is to obtain a live system: the only consequence can be that control places also forbid some markings which are not reachable. In this way, a system with more control than needed can be obtained which will be, in any case, live. A siphon and the corresponding marking fulfilling conditions in Proposition 10 will be called a *potential bad siphon* and a *potential  $D$ -deadlocked marking*, respectively. However, and for the sake of simplicity, they will be called bad siphon and  $D$ -deadlocked marking.

The approach we are going to propose does not obtain all the solutions of the system of Proposition 10. The considered method will obtain a bad siphon, for later controlling it by means of the addition of the adequate place, and iteratively continue computing and controlling new bad siphons. The reason for this is clear: the added control will modify the system behavior and some bad markings associated to another siphons can be forbidden.

We are going to transform the system of equations into another one that will obtain just one siphon as solution. This raises the question of how to decide which siphon to control. The proposed approach selects the siphon with a minimal number of places in the hope that controlling first smaller siphons may help to avoid the control of the bigger ones. The following corollary introduces the problem.

**Corollary 11** Let  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$  be a marked  $S^4PR$ . If the net is non-live, then there exist a siphon  $D$  and a marking  $\mathbf{m} \in \text{PRS}(\mathcal{N}, \mathbf{m}_0)$  such that the following set of inequalities has, at at least, one solution with  $D = \{p \in P_S \cup P_R \mid v_p = 0\}$ :

$$\left\{ \begin{array}{l} \text{maximize} \quad \sum_{p \in P \setminus P_0} v_p \\ \text{s.t.} \quad \text{System (3)} \end{array} \right. \quad (4)$$

The solution of this problem is a bad siphon,  $D$ , and a  $D$ -deadlocked marking,  $\mathbf{m}$ . No special consideration has been done about the  $D$ -deadlocked marking associated to the siphon, while some restrictions about minimality have been done for  $D$ .

Nevertheless, we do not want to avoid only just this  $D$ -deadlocked marking but also all the deadlocked markings related to the siphon. In consequence, a new problem needs to be solved: once the siphon is known, which are the deadlocked markings for it? The approach considered here is to compute some selected ‘representative’ markings that can be used to avoid all the related  $D$ -deadlocked markings. This will be accomplished here in either one of two alternative ways: looking at the maximal number of resources available at  $D$ -deadlocked markings; looking at the minimal number of active processes at  $D$ -deadlocked markings. For this, it will be useful to return to Proposition 8. The equations presented there were constructed supposing that the siphon was known. Let us use them in order to construct the associated  $D$ -restrictions.

The restriction  $\mathbf{m}[P_S \setminus Th_D] = 0$  from Theorem 7 can be added since the siphon is now known.

**Definition 12** Let  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$  be a marked  $S^4PR$ . Let  $D$  be a bad siphon. The set of  $D$ -restrictions is:

$$\begin{cases} \mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \bar{\sigma} \\ \mathbf{m} \geq 0, \bar{\sigma} \in \mathbf{Z}_+^{T_1} \\ \mathbf{m}[P_S \setminus Th_D] = 0 \\ \text{System (1)} \end{cases} \quad (5)$$

**Definition 13** Let  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$  be a marked  $S^4PR$ . Let  $D$  be a bad siphon,  $m_D^{max}$  and  $m_D^{min}$  are defined as follows:

$$ILPP_D^{max} : \quad m_D^{max} = \text{maximize } \sum_{r \in D_R} \mathbf{m}[r] \\ \text{s.t. restrictions (5)}$$

$$ILPP_D^{min} : \quad m_D^{min} = \text{minimize } \sum_{p \in Th_D} \mathbf{m}[p] \\ \text{s.t. restrictions (5)}$$

These two problems are, in some way, equivalent: either both have solution or none of them has solution. They look for deadlocked markings, concentrating on different points of view. That is, while  $m_D^{max}$  looks at the number of tokens in  $D_R$  at deadlocked markings,  $m_D^{min}$  looks at the number of active processes in places belonging to  $Th_D$  that are "stealing" tokens from  $D$  at deadlocked markings.

When referring to a particular  $ILPP$  problem of the ones presented in Definition 13,  $ILPP_D^{max}$  or  $ILPP_D^{min}$  will be used. When referring to any of them  $ILPP_D$  will be used.

Once a bad siphon  $D$  has been computed, it can be controlled using  $ILPP_D^{max}$  or  $ILPP_D^{min}$  in order to prevent  $D$ -deadlocked markings in two different ways: adding one control place ensuring that processes in  $Th_D$  are not using more resources than  $\mathbf{m}_0[D_R] - m_D^{max}$ . If this is the adopted approach (called the  $D$ -resource approach), the system will be said to be  $D$ -resource-controlled; alternatively, adding a control place ensuring that there will be no more than  $m_D^{min} - 1$  active process in places belonging to  $Th_D$ . If this is the adopted approach (called the  $D$ -process approach), the system will be said to be  $D$ -process-controlled. If the adopted method is not specified, the resulting system will be said to be  $D$ -controlled.

**Definition 14** Let  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ , be a non-live  $S^4PR$ . Let  $D$  be a bad siphon, and  $m_D^{max}$  and  $m_D^{min}$  as in Definition 13. Then, the associated  $D$ -resource place,  $p_D$ , is defined by means of the addition of the following incidence matrix row and initial marking:  $\mathbf{C}^{p_D}[p_D, T] = -\sum_{p \in Th_D} \mathbf{Y}_{D_R}[p] \cdot \mathbf{C}[p, T]$ , and  $\mathbf{m}_0^{p_D}[p_D] = \mathbf{m}_0[D] - (m_D^{max} + 1)$ . The associated  $D$ -process place,  $p_D$ , is defined by means of the addition of the following incidence matrix row and initial marking:  $\mathbf{C}^{p_D}[p_D, T] = -\sum_{p \in Th_D} \mathbf{C}[p, T]$ , and  $\mathbf{m}_0^{p_D}[p_D] = m_D^{min} - 1$ .

To exemplify the previous definition, let us come back to the  $S^4PR$  in Figure 2.  $D1 = \{P1_3, P2_1, R2\}$  was a bad siphon.

According to Definition 14, two different control places can be added:  $D1$ -resource place:  $\mathbf{C}^{p_{D1-R}}[p_{D1-R}, T] = [0, -2, 2, 0, 0, 0, 0]$ ,  $\mathbf{m}_0^{p_{D1-R}}[p_{D1-R}] = 3$ , and  $D1$ -process place  $\mathbf{C}^{p_{D1-S}}[p_{D1-S}, T] = [0, -1, 1, 0, 0, 0, 0]$ ,  $\mathbf{m}_0^{p_{D1-S}}[p_{D1-S}] = 1$ .

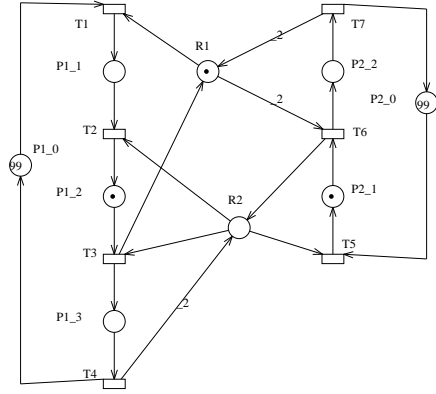
Figure 1 shows these places.

Now, two important properties need to be proved for the added places. First, we are going to show that the initial markings for  $D$ -control places are non-negative (this is needed to ensure that the  $D$ -controlled net is a well-defined Petri net.) As a second step, it will be shown that the added place can be seen as a new (virtual) resource (this is needed in order to iterate the process.) For this second property, two things are needed:  $p_D$  must verify structure conditions to be a resource, and the (extended) marking must be acceptable in the resulting  $S^4PR$  (See Definitions 1 and 2.)

**Lemma 15 ([24])** Let  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ , be a non-live  $S^4PR$ . Let  $D$  be a bad siphon, and  $m_D^{max}$  and  $m_D^{min}$  as in Definition 13. Let  $\langle \mathcal{N}^{p_D}, \mathbf{m}_0^{p_D} \rangle$ ,  $\mathcal{N}^{p_D} = \langle P_0 \cup P_S \cup P_R \cup \{p_D\}, T, \mathbf{C}^{p_D} \rangle$  be the net obtained by the addition of the  $D$ -resource place or the  $D$ -process place. Then,  $\mathbf{m}_0^{p_D}[p_D] > 0$ .

Lemma 15 proves that the initial markings computed for a  $D$ -resource or a  $D$ -process control place are non-negative. The initial marking computed for a  $D$ -resource place can be non-acceptable. The question is to know which conditions ensure that such markings are acceptable.



Figure 3: A  $S^4PR$  net.

In Lemma 16 we are going to prove the following facts: The initial marking computed for a  $D$ -process place is always acceptable. Any  $D$ -process place will be also called  $D$ -process control place since it is always possible to use it to control the system. The initial marking computed for a  $D$ -process place will be acceptable if and only if  $\mathbf{m}_0^{pD}[p_D] \geq \max_{p \in Th_D} \{\mathbf{Y}_{DR}[p]\}$ . Those  $D$ -resource places for which the initial marking computed in Definition 13 is acceptable will be called  $D$ -resource control places.

**Lemma 16 ([24])** *Let  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ , be a non-live  $S^4PR$ . Let  $D$  be a bad siphon, and  $m_D^{max}$  and  $m_D^{min}$  as in Definition 13. The net  $\langle \mathcal{N}^{pD}, \mathbf{m}_0^{pD} \rangle$ ,  $\mathcal{N}^{pD} = \langle P_0 \cup P_S \cup P_R \cup \{p_D\}, T, \mathbf{C}^{pD} \rangle$ , obtained by the addition of a  $D$ -process control place or a  $D$ -resource control place is a marked  $S^4PR$ .*

As a consequence, the added place can be considered as a new (virtual) resource whose holders are  $\mathcal{H}_{pD} = Th_D$ . Let us prove that the addition of any of the considered control places strictly reduces the size of the potentially reachable set.

**Lemma 17 ([24])** *Let  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ , be a non-live  $S^4PR$ . Let  $D$  be a bad siphon, and  $m_D^{max}$  and  $m_D^{min}$  as in Definition 13. Let  $\langle \mathcal{N}^{pD}, \mathbf{m}_0^{pD} \rangle$ ,  $\mathcal{N}^{pD} = \langle P_0 \cup P_S \cup P_R \cup \{p_D\}, T, \mathbf{C}^{pD} \rangle$ , the*

$S^4PR$  obtained by the addition of the  $D$ -process place or the  $D$ -resource place. Then,  $|\text{PRS}(\mathcal{N}^{pD}, \mathbf{m}_0^{pD})| < |\text{PRS}(\mathcal{N}, \mathbf{m}_0)|$

In our experience, the use of the  $D$ -resource approach gives more permissive controlled systems. In consequence, the approach we are going to propose will try to first apply the  $D$ -resource-control; if the resulting initial marking is not acceptable, then apply the  $D$ -process-control.

Notice that no potential reachable marking in  $\text{PRS}(\mathcal{N}^{pD}, \mathbf{m}_0^{pD})$  can be  $D$ -deadlocked, which implies the same property for any reachable marking, since  $\text{RS}(\mathcal{N}^{pD}, \mathbf{m}_0^{pD}) \subseteq \text{PRS}(\mathcal{N}^{pD}, \mathbf{m}_0^{pD})$ .

## 5 Preventing deadlock problems in $S^4PR$

We have concentrated on the prevention of the bad markings related to a given bad siphon. An iterative algorithm is going to be proposed to control the whole system. It is structured in the following steps:

1. Compute a bad siphon using system (4).
2. Compute  $m_D^{max}$  (Definition 13).
  - If the corresponding control place has an acceptable marking, go to the following step (Definition. 14).
  - If not, compute  $m_D^{min}$  (Definition 13).
3. Add the control place. (Definition 14).
4. Go to the first step, taking as input the partially controlled system, until no bad siphons exist.

**Theorem 18 ([24])** *Let  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$  be a marked  $S^4PR$ . The proposed algorithm applied to  $\langle \mathcal{N}, \mathbf{m}_0 \rangle$  terminates. The resulting controlled system,  $\langle \mathcal{N}^C, \mathbf{m}_0^C \rangle$ , is live.*

Let us use the  $S^4PR$  in Figure 3 to see how the algorithm works. In order to see the effect of the control policy, its reachability graph is depicted in Figure 4. The figure shows the deadlocked states (#16, #17, #18), the ones that lead in an inevitable way to them (#6, #11, #12, #13), and the ones a maximally permissive control policy should left (the rest.) Notice that the original system has 20 reachable markings from which the policy should

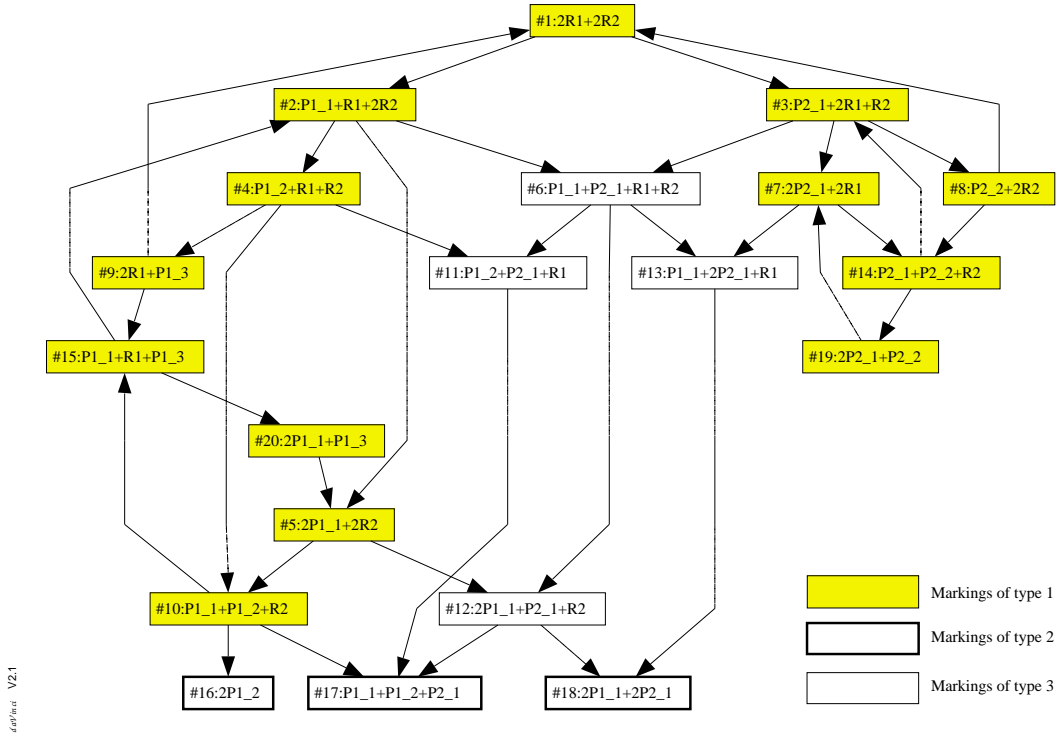


Figure 4: Reachability graph of the first net of Figure 3

left 13. The markings forbidden by each restriction are shown in the figure by means of lines labeled with the name of the control place: the control place forbids the markings under the corresponding line.

**Iteration 1:** The first bad siphon computed is  $D = \{P1\_3, P2\_1, R2\}$ . Solving the associated ILPP problem,  $m_D^{max} = 0$ , which generates the control place  $RCP1D$  whose associated invariant is:  $P1\_2 + RCP1D = \mathbf{m}_0[D] - (m_D^{max} + 1) = 1$ . It has an acceptable initial marking and it is added to the system.

**Iteration 2:** System 4 obtains the siphon  $D = \{P1\_3, P2\_3, R1, R2\}$ . Solving the associated ILPP problems,  $m_D^{max} = 0$ , which generates the control place  $RCP2D$  whose associated invariant is:  $P1\_1 + 2 \cdot P1\_2 + P2\_1 + RCP2D = \mathbf{m}_0[D] - (m_D^{max} + 1) = 2$ . It has an acceptable initial marking and it is added to the system.

**Iteration 3:** A new siphon obtained using System 4 is  $D = \{P1\_2, P2\_1, RCP2D\}$ . Solving

the associated ILPP problems,  $m_D^{max} = 0$ , which generates the control place  $RCP3D$  whose associated invariant is:  $P1\_1 + RCP3D = \mathbf{m}_0[D] - (m_D^{max} + 1) = 1$ . It has an acceptable initial marking and it is added to the system.

**Iteration 4:** The next siphon obtained solving System 4 is  $D = \{P1\_2, P2\_2, R1, RCP2D\}$ . Solving the associated ILPP problems,  $m_D^{max} = 1$ , which generates the control place  $RCP4D$  whose associated invariant is:  $2 \cdot P1\_1 + P2\_1 + RCP4D = \mathbf{m}_0[D] - (m_D^{max} + 1) = 2$ . It has an acceptable initial marking and it is added to the system. No new bad siphon exists and the algorithm terminates.

This system has 10 reachable states. Notice that the method also forbids 3 good states, showing that in the general case, it is not maximally permissive. To control the original  $S^4PR$ , four siphons have been computed and four new places have been added. In Figure 5 we can see the resulting  $S^4PR$ .

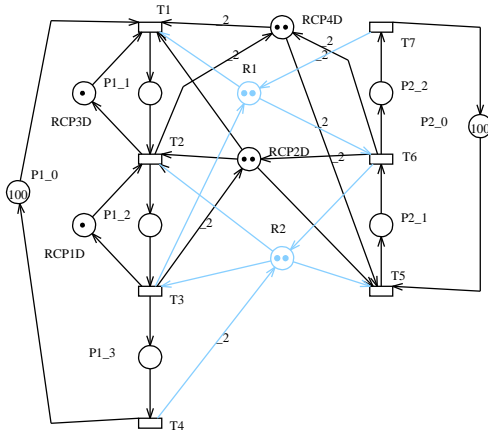


Figure 5: The  $S^4PR$  net obtained by controlling the system in Figure 3

Figure 6 shows the effect of the four control places added.

## 6 Conclusions

This paper has concentrated on the class of  $S^4PR$  nets, which model sequential  $RAS$  where on-line routing decisions are allowed and no constraint is imposed about the number and type of system resources used at each processing step. Three equivalent liveness characterizations have been presented.

These liveness characterizations for  $S^4PR$  nets have been used to control the initial system, obtaining a final live controlled system. The control is based on the use of siphons of the Petri net model from a structural point of view. This has the advantage of avoiding the computation of the system reachability set, avoiding the state explosion problem; it has been implemented in an iterative way, applying the control in an incremental way: at each iteration, one (potentially) bad siphon is computed and controlled. Therefore, the method does not need to compute at each iteration the set of minimal siphons (as is the case in [2], for instance), which makes it applicable for more complex systems.

## 7 Acknowledgement

This paper was originally presented at ICRA 2005 conference [21], and it has been adapted to appear here.

The authors are with the Group of Discrete Event Systems Engineering (GISED), at Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza. María de Luna 3, 50015 Zaragoza. Spain. Phone: 34 976 761949 Fax: 34 976 761914 E-mail: {ftricas,gvalles,jm,ezpeleta}@posta.unizar.es This work has been partially supported by the Spanish research projects CICYT TIC2001-1819.

## References

- [1] H. Alaiwan and J.M. Toudic. Recherche des semi-flots, des verrous et des trappes dans les réseaux de Petri. *Technique et Science Informatiques*, pages 103–112, 1985. In French.
- [2] K. Barkaoui, A. Chaoui, and B. Zouari. Supervisory control of discrete event systems based on structure of Petri nets. In *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC'97)*, pages 3750–3755, Orlando, Florida, USA, October 1997. IEEE.
- [3] Feng Chu and Xiaolan Xie. Deadlock analysis of Petri nets using siphons and mathematical programming. *IEEE Trans. Rob. Automat.*, 13(6):793–804, dec 1997.
- [4] J.M. Colom and M. Silva. Improving the linearly based characterization of P/T nets. In *Proc. of the 10th Int. Conference on Application and Theory of Petri Nets*, pages 52–73, Bonn, Germany, June 1989.
- [5] J.M. Colom and M. Silva. Improving the linearly based characterization of P/T nets. In G. Rozenberg, editor, *Advances in Petri Nets 1990*, volume 483 of *Lecture Notes in Computer Science*, pages 113–145. Springer Verlag, Berlin, 1991.
- [6] J. Ezpeleta, J.M. Colom, and J. Martínez. A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Trans. Rob. Automat.*, 11(2):173–184, April 1995.

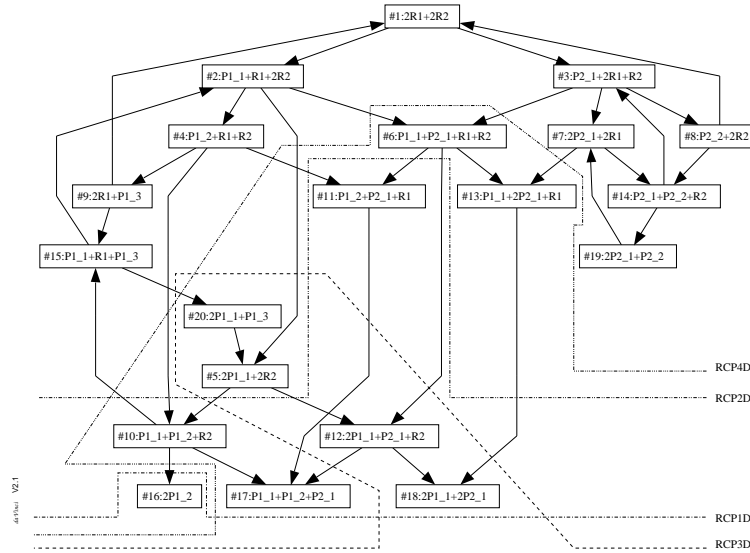


Figure 6: Reachability set of the net in Figure 3 . The states under the lines are prevented by the addition of the respective control places

- [7] M.P. Fanti, B. Maione, S. Mascolo, and B. Turchiano. Event-based feedback control for deadlock avoidance in flexible production systems. *IEEE Trans. Rob. Automat.*, 13(3):347–363, June 1997.
- [8] Alessandro Giua and Carla Seatzu. Liveness enforcing supervisors for railway networks using  $ES^2PR$  Petri nets. In *Proceedings of the 6th International Workshop on Discrete Event Systems*, pages 55–66, Zaragoza, Spain, October 2002. IEEE.
- [9] F. Hsieh and S. Chang. Dispatching-driven deadlock avoidance controller synthesis for flexible manufacturing systems. *IEEE Trans. Rob. Automat.*, 10(2):196–209, April 1994.
- [10] Y. Huang, M. D. Jeng, Z. Xie, and S. Chung. Deadlock prevention policy based on Petri nets and siphons. *International Journal of Production Research*, 39(2):283–305, 2001.
- [11] Marian V. Iordache, John O. Moody, and Panos Antsaklis. Synthesis of deadlock prevention supervisors using Petri nets. *IEEE Trans. Rob. Automat.*, 18(1):59–68, February 2002.
- [12] Mark A. Lawley. Deadlock avoidance for production systems with flexible routing. *IEEE Trans. Rob. Automat.*, 15(3):1–13, jun 1999.
- [13] T. Murata. Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.
- [14] Jonghun Park and Spyros Reveliotis. Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings. *IEEE Trans. Automat. Contr.*, 46(10):1572–1583, October 2001.
- [15] S.A. Reveliotis. Accommodatingfms operational contingencies through routing flexibility. *IEEE Transactions on Robotics and Automation*, 15(1):3–19, February 1999.

- [16] Spyros A. Reveliotis. On the siphon-based characterization of liveness in sequential resource allocation systems. In *Proc. of the 24th Int. Conf. on Applications and Theory of Petri Nets (ICATPN 2003), Eindhoven, The Netherlands, June 23-27, 2003*, pages 241–255. Springer-Verlag, June 2003. Internal-Note: Submitted by: hr.
- [17] M. Silva. *Las Redes de Petri en la Automática y la Informática*. Editorial AC, Madrid, 1985.
- [18] F. Tricas, J.M. Colom, and J. Ezpeleta. A solution to the problem of deadlocks in concurrent systems using Petri nets and integer linear programming. In G. Horton, D. Moller, and U. Rude, editors, *Proc. of the 11th European Simulation Symposium*, pages 542–546, Erlangen, Germany, October 1999. The society for Computer Simulation Int.
- [19] F. Tricas, J.M. Colom, and J. Ezpeleta. Some improvements to the Banker's algorithm based on the process structure. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 2853–2858, San Francisco, USA, April 2000. IEEE.
- [20] F. Tricas and J. Ezpeleta. A Petri net solution to the problem of deadlocks in systems of processes with resources. In *Proc. of the 7th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, pages 1047–1056, Barcelona, October 1999.
- [21] F. Tricas, F.García-Vallés, J.M. Colom, and J. Ezpeleta. A petri net structure-based deadlock prevention solution for sequential resource allocation systems. In *Proceedings of 2005 International Conference on Robotics and Automation, ICRA 2005*, pages 272–278, Barcelona, Spain, April 2005.
- [22] F. Tricas, F. García-Vallés, J.M. Colom, and J. Ezpeleta. A structural approach to the problem of deadlock prevention in processes with resources. In *Proc. of the Int. Workshop on Discrete Event Systems(WODES'98)*, pages 273–278. IEE Control, IEE, August 1998.
- [23] F. Tricas and J. Martínez. An extension of the liveness theory for concurrent sequential processes competing for shared resources. In *Proc. of the 1995 IEEE Int. Conf. on Systems, Man and Cybernetics.*, pages 4119–4124, Vancouver, Canada, October 1995.
- [24] Fernando Tricas. *Deadlock Analysis, Prevention and Avoidance in Sequential Resource Allocation Systems*. Ph.D. Thesis. Departamento de Informática e Ingeniería de Sistemas. Universidad de Zaragoza, May 2003.
- [25] Iordache M. V. and P. J. Antsaklis. Synthesis of supervisors enforcing general linear vector constraints in Petri nets. *IEEE Trans. Automat. Contr.*, 48(11):2036–2039, November 2003.
- [26] Xiaolan Xie and MuDer Jeng. ERCN–merged nets and their analysis using siphons. *IEEE Trans Rob Automat.*, 15(4):692–703, August 1999.
- [27] K. Xing, B. Hu, and H. Chen. Deadlock avoidance policy for Petri–net modeling of flexible manufacturing systems with shared resources. *IEEE Trans. Automat. Contr.*, 41(2):289–295, February 1996.