Resource Allocation Systems: Some Complexity Results on the S^{4}PR Class

Juan-Pablo López-Grao[1] and José-Manuel Colom[2]

[1] Dpt. of Computer Science and Systems Engineering (DIIS)
[2] Aragonese Engineering Research Institute (I3A)
University of Zaragoza, Spain
{jpablo, jm}@unizar.es

Abstract. In recent times, Petri nets have consolidated as a powerful formalism for the analysis and treatment of deadlocks in Resource Allocation Systems (RAS). In particular, the methodological framework yielded by the S^{4}PR class has raised considerable interest on the grounds of a well-balanced compromise between modelling flexibility and the provision of sound and effective correction techniques. These are strengthened by the advantages of the abstraction process, which allows the effective application of these techniques to diverse application domains. Most of the works on this class focus on providing tools and algorithms for dealing with the so-called resource allocation problem. This paper takes a different approach to provide an insight into the inherent computational complexity of the problem, from the perspective of optimality in either prevention, avoidance or detection of deadlocks. In particular, we will prove that most of the problems involved fall within the category of NP or co-NP-complete problems.

1 Introduction

A Resource Allocation System (RAS) is, in rough words, a discrete event system in which a finite set of concurrent processes share a finite set of resources. This is strongly connected to the resource allocation problem, which consists in meeting the demand of resources by the set of processes, eventually accomplishing certain goals. From the qualitative standpoint, the objective is often dealing with the set of potential system deadlocks: the focus of this paper.

A RAS is in a deadlock state if a set of processes are indefinitely waiting for a set of resources that are already held by the same set of processes. Coffman defined in [1] four necessary conditions for the existence of a deadlock, but a general characterization remains elusive, leaving place for a wide family of works which study different subclasses of RAS, often providing solutions over abstract models that allow their application on different domains.

The strategies for handling deadlocks are categorized in three groups. Deadlock prevention techniques consist in constructing a system such that, by definition, no deadlock is reachable. Deadlock avoidance techniques ensure that a deadlock is not reachable by deciding on-line if a resource allocation request is granted or not, based on the current system state information (e.g., the banker's

E. Najm et al. (Eds.): FORTE 2006, LNCS 4229, pp. 323–338, 2006.

algorithm [2]). Finally, deadlock detection techniques act 'a posteriori', allowing the deadlock situation to occur and subsequently resolve it.

Among formal models, Petri nets [3] has proven to be a fruitful tool for the modelling, analysis and synthesis of RAS ([4,5,6,7]). In particular, the S^4PR class [8] (S^3PGR^2 in [7]) has attracted significant attention since it deals with a very general class of Sequential RAS (S-RAS, i.e., RAS in which the processes are sequential), while exist efficient characterizations for deadlock states, i.e. states from which a given transition cannot be fired anymore. Despite that most of those works stress the application on Flexible Manufacturing Systems, the fact that we employ a purely systemic approach enables applying this Petri net models, as well as their well-known analysis and synthesis techniques, to very different application domains, such as distributed systems or communication protocols.

The S^4PR class is capable to model systems in which the processes are allowed to decide between alternative execution paths all along their execution, provided there are no internal iterations. Besides, several resources of several types can be reserved at the same time, and they can be acquired and released at any execution state. Note that we assume that the resources are used in a conservative way by every process (i.e. the resources are *serially reusable*).

This work investigates the computational complexity on providing optimal solutions for the problems of deadlock prevention, avoidance and detection for S-RAS supported by the S^4PR class. Some previous works have successfully studied computational issues on S-RAS, although they differ from this both in the type of systems and the problems subject to analysis. In [2] the problem of deciding whether a resource allocation is safe is studied, and proved NP-complete for S-RAS with multi-resource requests and processes without routing decisions. In this model, resources that are freed in intermediary states are immediately required back. Additionally, some restrictions on this problem are presented, which are proved polynomial. In [9], it is proved that optimal deadlock avoidance is NP-complete for a subclass of S-RAS in which no alternative paths per process are allowed. Finally, in [10] the same problem is proven NP-complete for a class of S-RAS in which alternative paths are allowed, but only one resource type is used in each stage, which is again a subclass of our model.

In section 2, we provide a motivating example that hopefully will enlighten the scope of the S^4PR class. The class is also formally introduced, along with some basic results that are used in section 3. Section 3 is divided in four parts. First, we introduce the computational complexity of characterizing non-liveness for a marked S^4PR with an acceptable initial marking. Second, the results are extended to the case in which any arbitrary reachable marking is considered. This is strongly related to optimal deadlock prevention. Third, we state the computational complexity in determining the markings that are doomed to deadlock, which is the key to optimal deadlock avoidance and detection in this context. And four, the computational complexity in determining spurious markings is revealed, which severely affects the efficiency of structural techniques for this type of models. Finally, section 4 summarizes the conclusions of the paper.

2 The S^{4}PR Class

2.1 A Motivating Example

Suppose we are considering the installment of an on-line, on-demand video streaming service business on the Internet. In order to provide a reasonably good service, certain Quality of Service (QoS) requirements must be formally established and satisfied, for every requested transmission. These QoS specs obviously depend on a wide range of parameters such as the client type, her/his maximum supported bandwidth, the format and resolution of the requested video, etc.

To provide the service, we own a pool of video servers. These video servers are connected to a mesh network of router nodes. Some of these nodes act as gateways to the Internet. We will assume that multicast video streams will disseminate from the gateways onwards, so as to not increase our internal traffic. Figure 1 depicts the system structure (on the left, the video servers; on the right, the gateways; in the middle, the intermediate routers).

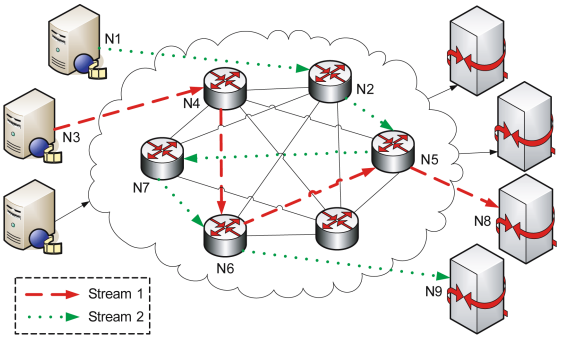

Fig. 1. Our video streaming system, simultaneously transmitting two video streams

A video stream is composed of a set of fixed-size packets that must be transmitted from the sender (video server) to the receiver (client). When a receiver requests a video stream to one of the servers, a virtual circuit is constructed. All the packets of the video stream will travel through the same virtual circuit. Besides, each node of the circuit assumes its own minimum resource requirements (CPU, storage, bandwidth) for processing and transmitting each packet of the stream. These requirements will be based on the QoS specs for the transmission.

Both (circuit and resource requirements) can be determined and established through a signaling protocol in a similar vein to RSVP [11,12]. In order to maximize our system productivity and reduce costs, however, we want to 'relax' the resource reservation strategy. Hence once a packet is effectively transmitted from a node to the next one, the required resources are freed, and must be reacquired for the next packet. Doing so, nodes can accept and manage a higher amount of concurrent streams minimizing resource idling. As a drawback, when the traffic is high and resources are overused, some jittering could appear since

some packets could be idle in intermediate nodes, waiting for the release of some required resources. In the worst case, a circular wait for resources could appear, and the system would reach a deadlock.

Such a kind of systems can be effectively modellized and studied via the S^4PR class. Figure 2 models the system of figure 1. The different constructive elements in the model will be presented in the next subsection. In the example, the system has reached a deadlock. The existing analysis and synthesis techniques will allow us to handle deadlocks. In particular, we will be able to apply prevention (e.g. disallow a pre-established circuit if there might be a potential deadlock situation), avoidance (e.g. retain temporarily packets if they lead to deadlock situations) or detection and correction techniques (e.g. abort a video stream to free resources and unlock the system). In the following, we will study the computational complexity of the optimal approach for these three strategies.

2.2 Formal Definition of the Class

From now on, we assume the reader has some basic knowledge on Petri nets. Some useful definitions are provided in the appendix A.

As it was already pointed out, the S^4PR is a P/T net class aimed to the modelling, analysis and synthesis of S-RAS. In an S^4PR, each process is a strongly connected state machine in which no internal cycles are allowed throughout its execution. Besides, each process has an initial local state in which no resource is used, represented by the *idle place*. Resources are modelled as tokens within the *resource places*, and their usage by every process is conservative, which imposes restrictions on the form of the set of p-semiflows. In formal terms:

Definition 1. *[13] Let $I_{\mathcal{N}}$ be a finite set of indices. An S^4PR is a connected generalized pure P/T 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) [idle places] *$P_0 = \bigcup_{i \in I_{\mathcal{N}}} \{p_{0_i}\}$.*
   (b) [process places] *$P_S = \bigcup_{i \in I_{\mathcal{N}}} P_{S_i}$, where $\forall\ i \in I_{\mathcal{N}}$, $P_{S_i} \neq \emptyset$, and $\forall\ i, j \in I_{\mathcal{N}}$, $i \neq j$, $P_{S_i} \cap P_{S_j} = \emptyset$.*
   (c) [resource places] *$P_R = \{r_1, r_2, r_3, ..., r_n\}, n > 0$.*
2. *$T = \bigcup_{i \in I_{\mathcal{N}}} T_i$, where $\forall i \in I_{\mathcal{N}}, T_i \neq \emptyset$, and $\forall i, j \in I_{\mathcal{N}}, i \neq j, T_i \cap T_j = \emptyset$.*
3. *For each $i \in I_{\mathcal{N}}$ the subnet generated by $\{p_{0_i}\} \cup P_{S_i}$, 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 $Y_r \in \mathbb{N}^{|P|}$ such that $\{r\} = \|Y_r\| \cap P_R, P_0 \cap \|Y_r\| = \emptyset, P_S \cap \|Y_r\| \neq \emptyset$, and $Y_r[r] = 1$.*
5. *$P_S = \bigcup_{r \in P_R} (\|Y_r\| \setminus \{r\})$.*

Meanwhile, we call *process net* [13] to the subnet generated by $\{p_{0_i}\} \cup P_{S_i} \cup P_{R_i}$ and T_i, where $i \in I_{\mathcal{N}}$ and $P_{R_i} = \{r \in P_R \mid (\|Y_r\| \cap P_{S_i} \neq \emptyset)\}$.

In the case of figures 1 and 2, each video stream is modelled as a concurrent sequential process. Resources associated to each node Ni are modellized using the places labelled R-Ni. Note that there could be several resource places per router

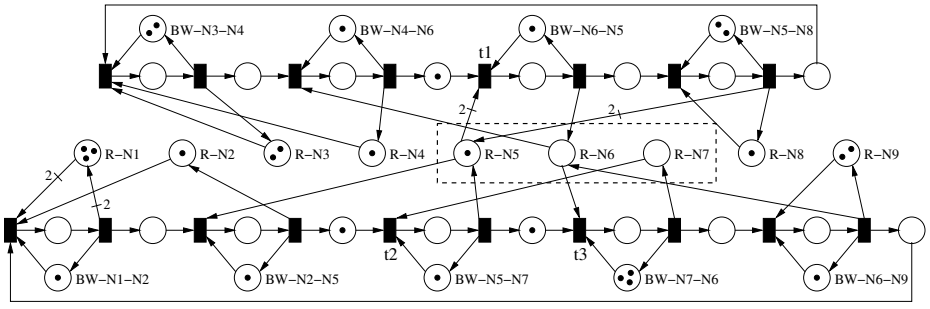

Fig. 2. A marked S^{4}PR which models the system in figure 1. The system is deadlocked.

(one per each resource type, be it physical, e.g. available storage space or CPU slots, or logical, e.g. maximum number of simultaneous packets). Equivalently, there is a resource place per each node interconnection, modelling the available bandwidth and labelled BW-Ni-Nj.

All these resources can be shared among both concurrent processes. In this case, the local resources of the nodes N5 and N6 (held by resource places R-N5 and R-N6) are shared among both video streams. The resources are requested, used and freed when a packet (a token in the process net) is visiting the corresponding node. Finally, the idle places limit the number of potentially concurrent packets per video stream (it is assumed that this number is finite). Speaking in general terms, it is worth noting here that idle places can also be seen as special resource places, and then interpreted as the maximum number of process instances in concurrent execution for each process type.

Definition 2. *[13] Let $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, C \rangle$ be an S^{4}PR. An initial marking m_0 is acceptable for $\mathcal{N}$ iff $||m_0|| = P_0 \cup P_R$ and $\forall p \in P_S, r \in P_R \; . \; m_0[r] \geq Y_r[p]$.*

Figure 4 depicts a marked S^{4}PR with an acceptable initial marking. The marking shown in figure 2 is *not* an acceptable initial marking but, however, it *is reachable from* an acceptable initial marking, as the reader can check. This acceptable initial marking would correspond to the system state in which no video stream has begun to transmit yet (and hence every resource is available).

2.3 Non-liveness Characterization in the S^{4}PR Class

During the paper, we will use the following definitions extensively. They will be used in several demonstrations and are basic for the non-liveness characterization that is stated in theorem 1. This well-known characterization will be the base for our first complexity result in section 3.

Definition 3. *[13] Let $\langle \mathcal{N}, m_0 \rangle$ be a marked S^{4}PR with an acceptable initial marking, $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, C \rangle$. Also, let $m \in RS(\mathcal{N}, m_0)$.*

*Then $t \in T$ is m-*process-enabled *iff $^\bullet t \cap P_S \neq \emptyset$ and $m[^\bullet t \cap P_S] > \mathbf{0}$. Otherwise, t is m-*process-disabled*. Besides, t is m-*resource-enabled *iff $\forall r \in {}^\bullet t \cap P_R$, $m[r] \geq Pre[r, t]$. Otherwise, t is m-*resource-disabled*.*

Theorem 1. *[13] Let $\langle \mathcal{N}, m_0 \rangle$, $\mathcal{N} = \langle P, T, C \rangle$ be an S^4PR with an acceptable initial marking. The system $\langle \mathcal{N}, m_0 \rangle$ is non-live iff exists a reachable marking $m \in RS(\mathcal{N}, m_0)$ such that the set $S_m \subseteq T$ of m-process-enabled transitions is non-empty and every transition in S_m is m-resource-disabled.*

The system in figure 2 is non-live; indeed, it is a total deadlock. The reader can easily check that the set of m-process-enabled transitions is $\{t1, t2, t3\}$ and each one of those is m-resource-disabled: the resource places R-N5, R-N6 and R-N7 disallow their firing.

3 Complexity Results

In this paper, we will assume the reader is instructed on the basics of complexity theory [14] and particularly NP-completeness. Onwards, several problems will be proved either NP or co-NP-complete. All the problem reductions will be based on the well-known (general) satisfiability problem of boolean formulas in conjunctive normal form, commonly named SATISFIABILITY (SAT), which is NP-complete. A brief reminder is included in appendix B.

3.1 Non-liveness

The problem of optimal deadlock prevention requires determining whether a given system is non-live, in order to apply correction techniques to make the system live, such as those presented in [13]. Here we will devoted to the study of the complexity of the problem of non-liveness for a given acceptable initial marking. In particular, we will demonstrate that this problem is NP-complete. A couple of basic demonstrations are previously required, and hence will be introduced in the following. The studied problem is formally defined in this way:

Problem 1. **S^4PR-Non-Liveness (S^4PR-NL)**
Given: A marked S^4PR $\langle \mathcal{N}, m_0 \rangle$, being m_0 an acceptable initial marking.
To decide: Is $\langle \mathcal{N}, m_0 \rangle$ non-live?

Proposition 1. *Let $\langle \mathcal{N}, m_0 \rangle$, $\mathcal{N} = \langle P, T, C \rangle$, be a marked S^4PR with an acceptable initial marking. Let m be a reachable marking $m \in RS(\mathcal{N}, m_0)$. Then exists a firing sequence σ, $m_0[\sigma\rangle m$, such that there is no t-semiflow X with $\boldsymbol{\sigma} - X \geq \mathbf{0}$.*

Proof. Without loss of generality, let X be a minimal t-semiflow such that $\boldsymbol{\sigma} - X \geq 0$. Then we will prove that there exists a firing sequence σ', $m_0[\sigma'\rangle m$, where $\boldsymbol{\sigma'} - X \ngeq 0$, and $\boldsymbol{\sigma'} = \boldsymbol{\sigma} - k \cdot X$, with $k \in \mathbb{N} \setminus \{0\}$.

m is potentially reachable from m_0 with σ' because of the net state equation: $m = m_0 + C \cdot \boldsymbol{\sigma} = m_0 + C \cdot (\boldsymbol{\sigma'} + k \cdot X) = m_0 + C \cdot \boldsymbol{\sigma'}$.

The sequence σ' is also firable because a t-semiflow X is a circuit of a state machine and the completion of X corresponds to the movement of a token in this state machine from the idle place throughout the circuit returning to the idle place. Taking into account that this token in the idle place does not use resources, while in the rest of the places of the circuit uses some resource, freezing this token

in the idle place leaves a greater number of resources to fire the rest of transitions of σ. Therefore σ' is also firable, reaching m. □

Lemma 1. *Let $\langle \mathcal{N}, m_0 \rangle$, $\mathcal{N} = \langle P, T, C \rangle$, be a marked S^{4}PR with an acceptable initial marking, and let m be a reachable marking from $\langle \mathcal{N}, m_0 \rangle$, $m \in RS(\mathcal{N}, m_0)$. Then exists a firing sequence σ from m_0 to m, $m_0[\sigma\rangle m$, such that $|\sigma| \leq K \cdot |T|$, where $K = \sum_{p \in P_0} m_0[p]$*

Proof. By proposition 1, a firing sequence σ_1 exists, $m_0[\sigma_1\rangle m$, such that there is no t-semiflow X with $\boldsymbol{\sigma_1} - X \geq \mathbf{0}$. Let us suppose that $|\sigma_1| > K \cdot |T|$. It is straightforward that there exists a transition $t \in T$ such that t is fired at least $K + 1$ times in σ_1. Since the process subnets are conservative, and the process places are empty in m_0, for every reachable marking $m' \in RS(\mathcal{N}, m_0)$, $\sum_{p \in P_0 \cup P_S} m'[p] = K$.

This means that if we labelled each token in the process places with a unique identifier $i \in [1, K]$, at least one of them should visit twice the process place p, where $\{p\} = {}^\bullet t \cap (P_0 \cup P_S)$, i.e., the active process (the token) should travel through a circuit of the state machine. Since every circuit in a S^{4}PR induces a minimal t-semiflow ([8]) then exists a t-semiflow X, $\boldsymbol{\sigma}_1 - X \geq \mathbf{0}$, contradicting the hypothesis. □

The size of the firing sequence σ in lemma 1 is polynomial in the size and population of the net. This will let us prove that S^{4}PR-NL is in NP.

Theorem 2. *S^{4}PR-NL is NP-easy.*

Proof. We will use the following problem for our demonstration:

Problem 2. **S^{4}PR-Bad-Marking (S^{4}PR-BM)**
Given: A marked S^{4}PR $\langle \mathcal{N}, m_0 \rangle$, being m_0 an acceptable initial marking,
and a firing sequence σ such that $(|\sigma| \leq K \cdot |T|), (m_0[\sigma\rangle m)$ and $(m \neq m_0)$,
where $K = \sum_{p \in P_0} m_0[p]$.
To decide: Does $\langle \mathcal{N}, m \rangle$ hold that every m-process-enabled transition is m-resource-disabled?

1. S^{4}PR-BM is in P. Given σ, m can be easily computed using the net state equation. For every transition, m-process-enabledness and m-resource-disabledness can be checked in deterministic linear time in the size of $\mathcal{N}$.
2. Let $(\mathcal{N}, m_0, \sigma)$ be a valid input for S^{4}PR-BM, being $(\mathcal{N}, m_0)$ an input for S^{4}PR-NL. Since the length of σ is polynomial in the size of the input, it is trivial to find two encodings $e_1(\mathcal{N}, m_0, \sigma)$ and $e_2(\mathcal{N}, m_0)$ such that $|e_1(\mathcal{N}, m_0, \sigma)| \leq c' \cdot |e_2(\mathcal{N}, m_0)|^c$, given c, c'.[1]
3. S^{4}PR-NL can be *verified* in deterministic polynomial time. By theorem 1, S^{4}PR-NL returns YES with input $(\mathcal{N}, m_0)$ iff exists a firing sequence σ, $m_0[\sigma\rangle m$ and $m \neq m_0$, such that every m-process-enabled transition is m-resource-disabled. In that case, by lemma 1, a firing sequence σ can be found such that with $|\sigma| \leq K \cdot |T|$. Thus S^{4}PR-NL($\mathcal{N}$, m_0) returns YES iff exists σ such that S^{4}PR-BM($\mathcal{N}$, m_0, σ) returns YES. □

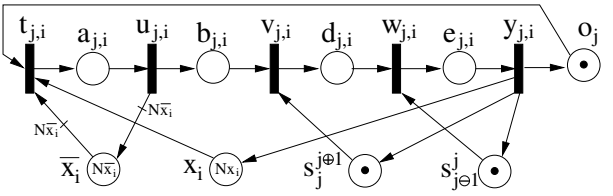

Fig. 3. SAT → S^{4}PR-NL. Net $\mathcal{N}_i^j$ for each literal x_i in $\mathcal{C}_j$.

Now we will devote to prove NP-hardness, reducing SAT to S^{4}PR-NL. Let $\mathcal{F} = \mathcal{C}_1 \cdot \mathcal{C}_2 \cdot ... \cdot \mathcal{C}_{N_c}$ be a formula in conjunctive normal form, and let $X = \{x_1, ...x_k\}$ be the set of its variables. For every $x_i \in X$ let N_{x_i} ($N_{\overline{x_i}}$) be the number of clauses of $\mathcal{F}$ in which the literal x_i ($\overline{x_i}$) appears.

Also please note that, for every $j \in [1, N_c]$, we define the index $j \oplus 1$ as either $j+1$ (iff $j < N_c$) or 1 (iff $j = N_c$). Similarly, we define the index $j \ominus 1$ as either $j-1$ (iff $j > 1$) or N_c (iff $j = 1$).

We will construct the net $\mathcal{N}_\mathcal{F}$ in the following compositional manner:

1. For every $x_i \in X$, $i \in [1, k]$, we add the place x_i (in case $N_{x_i} > 0$) and the place $\overline{x_i}$ (in case $N_{\overline{x_i}} > 0$).
2. For every clause $\mathcal{C}_j$, $j \in [1, N_c]$, we add two places to $\mathcal{N}_\mathcal{F}$, called o_j and $s_j^{j\oplus 1}$.
3. For every literal x_i in $\mathcal{C}_j$, $i \in [1, k]$, $j \in [1, N_c]$, we add four places ($a_{j,i}$, $b_{j,i}$, $d_{j,i}$, $e_{j,i}$) and five transitions ($t_{j,i}$, $u_{j,i}$, $v_{j,i}$, $w_{j,i}$, $y_{j,i}$), and we connect them to the rest of the net as depicted in figure 3.
4. For every literal $\overline{x_i}$ in $\mathcal{C}_j$, $i \in [1, k]$, $j \in [1, N_c]$, we add the same places and transitions as in the last point, but we do not *exactly* connect them as depicted in figure 3. Instead, we must follow the same pattern of the figure but interchanging x_i per $\overline{x_i}$, and N_{x_i} per $N_{\overline{x_i}}$.

In order to avoid unnecessary confusions, we want to remark the fact that the place $s_{j\ominus 1}^j$ in figure 3 is the same place as $s_{j'}^{j'\oplus 1}$, for $j' = j \ominus 1$ ($j = j' \oplus 1$).

The initial marking m_0 of every place will be as shown in figure 3. The reader can check that the resulting net system $\langle \mathcal{N}_\mathcal{F}, m_0 \rangle$ is a marked S^{4}PR with an acceptable initial marking, where $I_{\mathcal{N}_\mathcal{F}} = [1, N_c]$, every clause $\mathcal{C}_j$ results in a process net where o_j is the idle place, and the resource places are every x_i, $\overline{x_i}$, and $s_j^{j\oplus 1}$.

In figure 4 it is depicted the resulting net $\mathcal{N}_F$ for the formula $F = x1(x1 + \overline{x2})(x2 + \overline{x3})$. In this example, SAT(F) returns YES since the formula is satisfiable, e.g. assigning $x1$="true", $x2$="false" and $x3$="false".

Theorem 3. *SAT → S^{4}PR-NL*

Proof. We will prove that SAT(F) returns YES iff S^{4}PR-NL($\mathcal{N}_\mathcal{F}, m_0$) returns YES. By theorem 1, $\langle \mathcal{N}_\mathcal{F}, m_0 \rangle$ is non-live iff exists a reachable m, $m \neq m_0$, such that every m-process-enabled transition is m-resource-disabled. The four necessary conditions defined by Coffman [1] establish that in this state a circular

[1] By $|e|$ we denote the length of the encoding e.

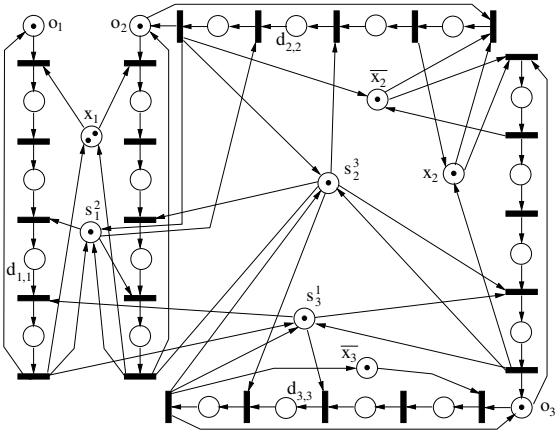

Fig. 4. SAT → S^{4}PR-NL. Example: $F = x1(x1 + \overline{x2})(x2 + \overline{x3})$

wait exists. This is only possible with a circular wait on the resource places $s_j^{j\oplus 1}$, since (by construction) the only transitions that can be m-process-enabled and m-resource-disabled are $v_{j,i}$ or $w_{j,i}$. Since it is also necessary that every locked process is in a "hold and wait" state on the blocking set of resources (as expressed by Coffman [1]), we can easily infer: $\langle \mathcal{N}_\mathcal{F}, m_0 \rangle$ is non-live iff exists $m \in RS(\mathcal{N}_\mathcal{F}, m_0)$ such that $\forall j \in [1, N_c]$. $\exists | i \in [1, k]$ such that $m[d_{j,i}] = 1$ (thus, $m[s_j^{j\oplus 1}] = m[a_{j,i}] = m[b_{j,i}] = m[e_{j,i}] = m[o_j] = 0$).

Now, $\forall j \in [1, N_c], i \in [1, k]$ such that $m[d_{j,i}] = 1$, there are two mutually exclusive alternatives: either (1) $Y_{x_i}[d_{j,i}] = 1$, $Y_{\overline{x_i}}[d_{j,i}] = 0$, or (2) $Y_{\overline{x_i}}[d_{j,i}] = 1$, $Y_{x_i}[d_{j,i}] = 0$. Note that Y_{x_i} and $Y_{\overline{x_i}}$ are the minimal p-semiflows induced by the resource places x_i, and $\overline{x_i}$, respectively.

By construction, (1) is applied to $\mathcal{N}_\mathcal{F}$ when literal x_i appears in the clause $\mathcal{C}_j$ of the formula $\mathcal{F}$. Equivalently, (2) is applied to $\mathcal{N}_\mathcal{F}$ when literal $\overline{x_i}$ appears in the clause $\mathcal{C}_j$ of the formula $\mathcal{F}$.

If (1) holds, then $\nexists j' \in [1, Nc]$, $j \neq j'$, such that $Y_{\overline{x_i}}[d_{j',i}] = 1$ and $m[d_{j',i}] = 1$. Otherwise, $t_{j,i}$ and $t_{j',i}$ should have been fired to reach m. But the firing of $t_{j,i}$ requires that no token from $\overline{x_i}$ is taken, and the firing of $t_{j',i}$ requires that no token from x_i is taken, so $t_{j,i}$ cannot be fired after $t_{j',i}$ and viceversa, leading to a contradiction. By an analogous reasoning, if (2) holds, then $\nexists j' \in [1, Nc]$, $j \neq j'$, such that $Y_{x_i}[d_{j',i}] = 1$ and $m[d_{j',i}] = 1$.

Let f be a truth assignment for the set of boolean variables X, $f : X \rightarrow \{$true, false, don't care$\}$. For every $x_i \in X$ we define $f(x_i)$ as:

- $f(x_i)$="true" iff $\exists j \in [1, N_c]$ such that $(m[d_{j,i}] = 1) \wedge (Y_{x_i}[d_{j,i}] = 1)$. This corresponds to case (1).
- $f(x_i)$="false" iff $\exists j \in [1, N_c]$ such that $(m[d_{j,i}] = 1) \wedge (Y_{\overline{x_i}}[d_{j,i}] = 1)$. This corresponds to case (2).
- $f(x_i)$="don't care", iff $\nexists j \in [1, N_c]$ such that $m[d_{j,i}] = 1$.

As we have seen, this assignments are mutually exclusive. Without loss of generality, we can finally redefine the non-liveness condition in the following way, which proves the hypothesis: $\langle \mathcal{N}_{\mathcal{F}}, m_0 \rangle$ is non-live iff exists a truth assignment f such that $\forall j \in [1, N_c]$. $\exists i \in [1, k]$ such that either $f(x_i) =$ "true" and x_i appears in $\mathcal{C}_j$, or $f(x_i) =$ "false" and $\overline{x_i}$ appears in $\mathcal{C}_j$. □

Note that, as expected, the net system in figure 4 is non-live: the total deadlock $\langle \mathcal{N}_{\mathcal{F}}, m \rangle$ is reachable from $\langle \mathcal{N}_{\mathcal{F}}, m_0 \rangle$, where $m[d_{1,1}] = m[d_{2,2}] = m[d_{3,3}] = m[x_1] = m[x_2] = 1$, being the rest of the places empty. Finally, we can conclude:

Theorem 4. *S^4PR-NL is NP-complete.*

Proof. S^4PR-NL is NP-hard since, by theorem 3, SAT is reducible to S^4PR-NL, and it is also NP-easy by theorem 2. □

3.2 Non-liveness Beyond the Initial Marking

The reader may have been left wondering why we chose to define the S^4PR problem beginning from an acceptable initial marking. Instead, we could have studied the more general problem of determining if, given $\langle \mathcal{N}, m \rangle$, $m \in RS(\mathcal{N}, m_0)$, the system is non-live. Indeed, the same complexity result applies: we can easily reduce this problem to S^4PR-NL. This is rather obvious from the fact that we can fire an arbitrary sequence from m trying to lead every active process to the idle places. If we are able to reach m_0, then the reduction applies. If we are not able to reach m_0, we will have found a marking such that every m-process-enabled transition is m-resource-disabled, and the system is thus non-live.

Note that this is not true in general for every solution of the net state equation, $m = m_0 + C \cdot X$, $X \gneqq 0$. The problem resides in the fact that S^4PR nets may have killing spurious solutions, i.e., solutions of the net state equation that are not reachable and which are non-live while the system $\langle \mathcal{N}, m_0 \rangle$ is live. Note that the problem of determining if a given marking is a spurious solution is studied in subsection 3.4, and it is proven to be co-NP-complete.

3.3 Deadlock Avoidance and Detection

In previous works ([2,9,10]), the complexity of the deadlock avoidance problem has been determined for different classes of RAS, in some sense more restrictive than the S^4PR category, as explained in section 1. These seminal results are based on the study of safeness (as defined in the deadlock prediction problem [2], "the existence of a feasible sequence in which to allocate the remaining resource requirements of the processes"). However, the process structure in these earlier models was finite and acyclic: once a process had satisfied all the resource requirements, it was terminated and hence removed from the system. On the other hand, a marked S^4PR does not have a target state; instead, the processes are structurally repetitive. Hence, it is desirable to ensure that the *feasible sequence* is arbitrarily long. This leads us to the following definition:

Definition 4. *Let $\langle \mathcal{N}, m_0 \rangle$, $\mathcal{N} = \langle P, T, C \rangle$ be an S^4PR with an acceptable initial marking, and let m be a reachable marking, $m \in RS(\mathcal{N}, m_0)$. Then $\langle \mathcal{N}, m \rangle$*

(or simply, m) is doomed to deadlock *iff* $\exists k \in \mathbb{N}$ *such that for every firable sequence* σ, $m[\sigma\rangle$, *exists* $t \in T$ *such that* t *is fired at most* k *times,* $\boldsymbol{\sigma}[t] \leq k$.

The negation of this property (i.e. m is not doomed to deadlock) is somehow an extension of that concept of safeness and leads us to the optimal deadlock avoidance strategy: in our relaxed terminology, a resource allocation will be "safe" iff m is not doomed to deadlock. Soon we will see that markings which are doomed to deadlock are well characterized in the S^{4}PR class.

In contrast, an optimal deadlock detection strategy should detect iff a marking m is doomed to deadlock, and apply recovery techniques in that case. It must be remarked that here we understand optimality in the strictest sense: the ability to detect the problem as soon as possible, i.e., as soon as a transition in the net is bound to die. Please note that other works define optimal detection as simply deciding iff there exists a transition which is effectively dead, i.e. no longer firable, in the current marking. The latter is less general and also computationally easier. The earlier will be proved co-NP-complete:

Problem 3. **S^{4}PR-Deadlock-Detection (S^{4}PR-DD)**
Given: A marked S^{4}PR $\langle \mathcal{N}, m_0 \rangle$, being m_0 an acceptable initial marking, and a reachable marking m, $m \in RS(\mathcal{N}, m_0)$.
To decide: Is $\langle \mathcal{N}, m \rangle$ doomed to deadlock?

Lemma 2. *Let* $\langle \mathcal{N}, m_0 \rangle$, $\mathcal{N} = \langle P, T, C \rangle$ *be an* S^4PR *with an acceptable initial marking, and let* m *be a reachable marking,* $m \in RS(\mathcal{N}, m_0)$. *Then* $\langle \mathcal{N}, m \rangle$ *(or simply,* m*) is doomed to deadlock iff* $m_0 \notin RS(\mathcal{N}, m)$.

Proof. The necessary part ("only if") is rather obvious: every minimal t-semiflow is firable in isolation from m_0. This means that we can build a repetitive sequence in which we successively fire every minimal t-semiflow, hence firing every transition an arbitrarily large number of times. Regarding the sufficient part ("if"), let us proceed by reduction to absurd. Suppose that $m_0 \notin \mathrm{RS}(N, m)$, and that exists an infinite finite sequence σ, $m[\sigma\rangle$ such that every transition is fired infinite times. In that case, every time a transition $t \in {}^\bullet P_0$ is fired in σ (so the marking of an idle place is increased), we can freeze the token in the correspondent idle place (i.e. leave the token there). Since the idle places are the unique places in which no resource is used, this augments the number of resources available in the system, so the rest of active processes (i.e. tokens in the process places) can be moved in the same way as in the original sequence σ. Proceeding this way, we could construct a sequence σ' that moves all the tokens to the idle places, reaching m_0, unless there exists a place $p \in P_S$ with frozen tokens in it ($m[\sigma'\rangle m'$, $m'[p] > 0$). But this is impossible, since that would imply that $p^\bullet$ is m'-resource-disabled. Since the number of available resources has not been decreased, that would imply that $p^\bullet$ was not infinitely firable in σ, reaching a contradiction. □

Thus the problem of deadlock avoidance can be reduced to the problem of determining the reachability of the initial marking: a problem that is NP-complete, as we will see.

Problem 4. **S^{4}PR-Reachable-Initial-Marking (S^{4}PR-RIM)**
Given: A marked S^{4}PR $\langle \mathcal{N}, m_0 \rangle$, being m_0 an acceptable initial marking, and a reachable marking m, $m \in RS(\mathcal{N}, m_0)$.
To decide: Is m_0 reachable from $\langle \mathcal{N}, m \rangle$?

Theorem 5. *S^{4}PR-RIM is NP-complete.*

Proof. In order to prove NP-easiness, let us introduce the following problem:

Problem 5. **S^{4}PR-Path-to-Initial-Marking (S^{4}PR-PIM)**
Given: A marked S^{4}PR $\langle \mathcal{N}, m_0 \rangle$, being m_0 an acceptable initial marking, a reachable marking $m \in RS(\mathcal{N}, m_0)$, and a firing sequence σ, $|\sigma| \leq K \cdot |T|$, where $K = \sum_{p \in P_0} m_0[p]$.
To decide: Is m_0 reached firing $m[\sigma\rangle$?

1. S^{4}PR-PIM is in P (this is rather trivial: checking the firability of every transition in the sequence can be done in deterministic linear time).
2. Let $(\mathcal{N}, m_0, m, \sigma)$ be a valid input for S^{4}PR-PIM, being $(\mathcal{N}, m_0, m)$ an input for S^{4}PR-NL. As the size of σ is polynomial in the number of transitions and population of the net, it is trivial to find two encodings $e_1(\mathcal{N}, m_0, m, \sigma)$ and $e_2(\mathcal{N}, m_0, m)$ such that $|e_1(\mathcal{N}, m_0, m, \sigma)| \leq c' \cdot |e_2(\mathcal{N}, m_0, m)|^c$, given c, c'.
3. S^{4}PR-RIM can be *verified* in deterministic polynomial time. By lemma 1, but reasoning over the reverse net, if m_0 is reachable there is a firing sequence σ, $m[\sigma\rangle m_0$, with ($|\sigma| \leq K \cdot |T|$). Hence, $S^4PR - NL$ returns YES with input $(\mathcal{N}, m_0)$ iff exists a firing sequence σ such that S^{4}PR-PIM returns YES.

Now that NP-easiness is proven, it is required to prove NP-hardness. But this part is rather straightforward, due to the fact that (as commented before) the problem of safeness in previous works ([2,9,10]) can be easily proven a subcase of S^{4}PR-RIM. Since the problem was already NP-hard for this models, we conclude that the problem is NP-hard through restriction [14]. □

Summing up, S^{4}PR-DD is co-NP-complete (i.e., optimal deadlock detection in the S^{4}PR is co-NP-complete)[2]. The problem of optimal deadlock avoidance remains NP-complete for the S^{4}PR class.

3.4 Spurious Markings

A spurious marking is a solution of the net state equation, $m = m_0 + C \cdot X$, $X \gneq \mathbf{0}$, that is not reachable from m_0. A killing spurious solution is a spurious marking such that $\langle \mathcal{N}, m \rangle$ is non-live. There exist Petri net subclasses, such as equal conflict (EQ) systems [15], for which killing spurious solutions are not possible. In those cases, the linear description provided by the net state equation can be used to determine the liveness of the system.

Unfortunately, the S^{4}PR class is not one of those classes, and this limits the potential of the net state equation for this purpose. Unless that, noticeably,

[2] However, we remind the reader that there exists a reachable marking m' such that it can be structurally characterized as a bad marking by theorem 1, but this does not affect the inherent computational complexity of the problem.

spurious solutions were efficiently detectable for a given S^{4}PR system. As we will see, however, this is a co-NP-complete problem:

Problem 6. **S^{4}PR-Spurious-Detection (S^{4}PR-SD)**
Given: A marked S^{4}PR $\langle \mathcal{N}, m_0 \rangle$, being m_0 an acceptable initial marking, and $m \in \mathbb{N}^{|P|}$, $m = m_0 + C \cdot X$, $X \geq \mathbf{0}$.
To decide: Is m an spurious marking?

Intuitively, m is an spurious marking iff m_0 is not reachable from m in its reverse (note that there may be isolated spurious solutions, i.e. not connected to the reachability space). Meanwhile, the reverse net of a S^{4}PR is another S^{4}PR. This is quite trivial, since the polarity inversion of the incidence matrix does not affect its (left or right) annullers, so the p and t-semiflows are preserved with respect to $\mathcal{N}$.

It is easy to see now that S^{4}PR-SD is co-NP-complete. This is bad news since, unless NP=P, this implies that we cannot *verify* that a marking is spurious in deterministic polynomial time using *solely* the structure of the net.

4 Conclusions

RAS is an abstraction of real systems allowing to concentrate on the study of problems such as deadlocks due to the sharing of resources used in mutual exclusion. Modelling RAS with Petri nets is particularly easy through the identification of processes with state machines and resources with monitor places representing the allocation of copies of resources. As a consequence, the S^{4}PR subclass has already been proven specially useful and suitable for the RAS abstraction of Flexible Manufacturing Systems (FMS) [13]. For this reason, we have devoted an insight on the complexity of some problems related to handling with deadlocks using this kind of models.

As expected, many of the important problems are proven computationally intractable, and for this reason, the heuristics presented in [8,13] have special interest. Regarding optimal deadlock prevention, we have established that the problem of determining if a marked S^{4}PR is non-live is NP-complete. Besides, we have provided evidence for NP-completeness of optimal deadlock avoidance for this class, generalizing earlier results for other types of RAS which were already proven NP-hard. This was accomplished thanks to proving the equivalence of this problem with that of deciding the reachability of the initial marking. The inverse problem (optimal deadlock detection, in the strictest sense) is co-NP-complete.

Moreover, because the mathematical methods presented in [13] are based on the net state equation, an insight on the complexity of the detection of spurious markings is also relevant. The intractability of the problem, along with the existence of killing spurious solutions, constrains the practicality of the net state equation for determining non-liveness.

Finally, a motivating example was also introduced, in order to depict the utility of our conceptual framework in the study and correction of deadlock problems in distributed systems and protocols, beyond the FMS context.

Obviously, the modelling power of the S^{4}PR class is limited if we consider, e.g., certain applications coming from the world of distributed computing. The

generalization of the S^4PR subclass for modelling RAS hence emerges as an appealing future research direction. [16] is a first effort in this vein. For the generalized net classes, the problems here studied will fall, at best, within the same complexity classes. Nevertheless, their study will give us insight on more complex behaviours that can observed in these systems [16].

References

1. Coffman, E.G., Elphick, M., Shoshani, A.: System deadlocks. ACM Computing Surveys **3**(2) (1971) 67–78
2. Gold, E.M.: Deadlock prediction: Easy and difficult cases. SIAM Journal on Computing **7**(3) (1978) 320–336
3. Murata, T.: Petri nets: Properties, analysis and applications. Proceedings of the IEEE **77**(4) (1989) 541–580
4. Lautenbach, K., Thiagarajan, P.S.: Analysis of a resource allocation problem using Petri nets. In Syre, J.C., ed.: Proc. of 1st European Conf. on Parallel and Distributed Processing, Toulouse, Cepadues Editions (1979) 260–266
5. Ezpeleta, J., Colom, J., Martínez, J.: A Petri net based deadlock prevention policy for flexible manufacturing systems. IEEE Trans. on Robotics and Automation **11**(2) (1995) 173–184
6. Xie, X., Jeng, M.D.: ERCN-merged nets and their analysis using siphons. IEEE Trans. on Robotics and Automation **29**(4) (1999) 692–703
7. Park, J., Reveliotis, S.A.: Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings. IEEE Trans. on Automatic Control **46**(10) (2001) 1572–1583
8. Tricas, F.: Deadlock analysis, prevention and avoidance in sequential resource allocation systems. PhD thesis, University of Zaragoza, Zaragoza (2003)
9. Lawley, M., Reveliotis, S.: Deadlock avoidance for sequential Resource Allocation Systems: Hard and easy cases. Int. Journal of Flexible Manufacturing Systems **13** (2001) 385–404
10. Sulistyono, W., Lawley, M.: Deadlock avoidance for manufacturing systems with partially ordered process plans. IEEE Trans. on Robotics and Automation **17**(6) (2001) 819–832
11. Brade, R., Zhang, L., Berson, S., Herzog, S., Jamin, S.: RFC 2205: Resource ReSerVation Protocol – Version 1 Functional Specification (1997)
12. Villapol, M., Billington, J.: Analysing properties of the resource reservation protocol. In Van der Aalst, W. and Best, E., eds.: Proc. of 24th Int. Conf. on Applications and Theory of Petri Nets. Vol. 2679 of LNCS. Springer–Verlag (2003) 377–396
13. Tricas, F., García-Vallés, F., Colom, J., Ezpeleta, J.: A Petri net structure-based deadlock prevention solution for sequential resource allocation systems. In: Proc. of IEEE Int. Conf. on Robotics and Automation, Barcelona, Spain (2005) 272–278
14. Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA (1979)
15. Teruel, E., Silva, M.: Liveness and home states in equal conflict systems. In Ajmone Marsan, M., ed.: Proc. of 14th Int. Conf. on Applications and Theory of Petri Nets. Vol. 691 of LNCS. Springer–Verlag (1993) 415–432
16. López-Grao, J.P., Colom, J.M.: Lender processes competing for shared resources: Beyond the S^4PR paradigm. In: Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics, Taipei, Taiwan (2006) To appear.

A Petri Nets: Basic Definitions

A *place/transition net* (P/T net) is a 3-tuple $\mathcal{N} = \langle P, T, W \rangle$, where W is a total function $W : (P \times T) \cup (T \times P) \to \mathbb{N}$, being P, T non empty, finite and disjoint sets. Elements belonging to the sets P and T are called respectively *places* and *transitions*, or generally nodes. P/T nets can be represented as a directed bipartite graph, where places (transitions) are graphically denoted by circles (rectangles): let $p \in P$, $t \in T$, $u = W(p,t)$, $v = W(t,p)$, there is a directed arc, labelled u (v), beginning in p (t) and ending in t (p) iff $u \neq 0$ ($v \neq 0$).

The *preset* (*poset*) or set of input (output) nodes of a node $x \in P \cup T$ is denoted by ${}^\bullet x$ ($x^\bullet$), where ${}^\bullet x = \{y \in P \cup T \mid W(y,x) \neq 0\}$ ($x^\bullet = \{y \in P \cup T \mid W(x,y) \neq 0\}$). The preset (poset) of a set of nodes $X \in bag(P) \cup bag(T)$ is denoted by ${}^\bullet X$ ($X^\bullet$), where ${}^\bullet X = \{y \mid y \in {}^\bullet x, x \in X\}$ ($X^\bullet = \{y \mid y \in x^\bullet, x \in X\}$

A *generalized P/T net* is a net with positive arc weights. If the arc weights are unitary (i.e., W can be defined as a total function $(P \times T) \cup (T \times P) \to \{0,1\}$) the net is called *ordinary*. A *state machine* is an ordinary net such that for every transition $t \in T$, $|{}^\bullet t| = |t^\bullet| = 1$.

Let $\mathcal{N} = \langle P, T, W \rangle$ be a P/T net. Its *reverse net* $\mathcal{N}^r = \langle P, T, W^r \rangle$ is the same net with its arcs inverted, i.e. $W^r(p,t) = W(t,p)$ and $W^r(t,p) = W(p,t)$.

A self-loop place $p \in P$ is a place such that $p \in p^{\bullet\bullet}$. A *pure P/T net* (also self-loop free P/T net) is a net with no self-loop places. In pure P/T nets, the net can be also defined by the 3-tuple $\mathcal{N} = \langle P, T, C \rangle$, where C is called the *incidence matrix*, $C[p,t] = W(p,t) - W(t,p)$.

A *marking* m of a P/T net $\mathcal{N}$ is a vector $\mathbb{N}^{|P|}$, assigning a finite number of marks $m[p]$ (called *tokens*) to every place $p \in P$. Tokens are represented by black dots within the places. The *support* of a marking, $\|m\|$, is the set of places which are marked in m, i.e. $\|m\| = \{p \in P \mid m[p] \neq 0\}$.

We define a *marked P/T net* (also P/T net system) as the duple $\langle \mathcal{N}, m_0 \rangle$, where $\mathcal{N}$ is a P/T net, and m_0 is a marking for $\mathcal{N}$, also called *initial marking*. $\mathcal{N}$ is said to be the structure of the system, while m_0 represents the system state.

Let $\langle \mathcal{N}, m_0 \rangle$ be a marked P/T net. A transition $t \in T$ is *enabled* (also *firable*) iff $\forall p \in {}^\bullet t \;.\; m_0[p] \geq W(p,t)$, which is denoted by $m_0[t\rangle$. The *firing* of an enabled transition $t \in T$ changes the system state to $\langle \mathcal{N}, m_1 \rangle$, where $\forall p \in P \;.\; m_1[p] = m_0[p] + C[p,t]$, and is denoted by $m_0[t\rangle m_1$. A *firing sequence* σ from $\langle \mathcal{N}, m_0 \rangle$ is a non-empty sequence of transitions $\sigma = t_1\, t_2 \ldots t_k$ such that $m_0[t_1\rangle m_1[t_2\rangle \ldots m_{k-1}[t_k\rangle$. The firing of σ is denoted by $m_0[\sigma\rangle t_k$. We call the *firing count* vector $\boldsymbol{\sigma}$ of σ to the Parikh mapping $\sigma \to \mathbb{N}^{|T|}$ (i.e. $\boldsymbol{\sigma}[t]$ is equal to the number of times t appears in σ). The support of $\boldsymbol{\sigma}$ is denoted by $\|\boldsymbol{\sigma}\|$.

A marking m is *reachable* from $\langle \mathcal{N}, m_0 \rangle$ iff there exists a *firing sequence* σ such that $m_0[\sigma\rangle m$. The *reachability set* $RS(\mathcal{N}, m_0)$ is the set of reachable markings, i.e. $RS(\mathcal{N}, m_0) = \{m \mid \exists\, \sigma \;.\; m_0[\sigma\rangle m\}$.

The *net state equation* of a marked P/T net $\langle \mathcal{N}, m_0 \rangle$ is an algebraic equation defined as $m = m_0 + C \cdot \boldsymbol{\sigma}$, where $\boldsymbol{\sigma} \gneq \mathbf{0}$. Every reachable marking holds the net state equation, but there may exist solutions to the equation which are not reachable markings. Thus we will call m a *potentially reachable marking*. The

potential reachability set $PRS(\mathcal{N}, m_0)$ is defined as $PRS(\mathcal{N}, m_0) = \{m \mid \exists\, \boldsymbol{\sigma} \in \mathbb{N}^{|T|}.\ m = m_0 + C \cdot \boldsymbol{\sigma}, \boldsymbol{\sigma} \gneq \mathbf{0}\}$.

A transition $t \in T$ is *live* iff for every reachable marking $m \in RS(\mathcal{N}, m_0)$, $\exists m' \in RS(\mathcal{N}, m)$ such that $m'[t\rangle$. The system $\langle \mathcal{N}, m_0 \rangle$ is *live* iff every transition is live. Otherwise, $\langle \mathcal{N}, m_0 \rangle$ is *non-live*. A transition $t \in T$ is *dead* iff there is no reachable marking $m \in RS(\mathcal{N}, m_0)$ such that $m[t\rangle$. The system $\langle \mathcal{N}, m_0 \rangle$ is a *total deadlock* iff every transition is dead, i.e. no transition is firable. A *home state* m_k is a marking such that it is reachable from every reachable marking, i.e. $\forall m \in RS(\mathcal{N}, m_0)\ .\ m_k \in RS(\mathcal{N}, m)$. The net system $\langle \mathcal{N}, m_0 \rangle$ is *reversible* iff m_0 is a home state.

A *p-semiflow* (*t-semiflow*) is a vector $Y \in \mathbb{N}^{|P|}$, $Y \neq \mathbf{0}$ ($X \in \mathbb{N}^{|T|}$, $X \neq \mathbf{0}$), which is a left (right) annuler of the incidence matrix, $Y \cdot C = \mathbf{0}$ ($C \cdot X = \mathbf{0}$). The support of a p-semiflow (t-semiflow) is denoted $\|Y\|$ ($\|X\|$), and its places (transitions) are said to be covered by Y (X). The P/T net $\mathcal{N}$ is *conservative* (*consistent*) iff every place (transition) is covered by a p-semiflow (t-semiflow). A *minimal p-semiflow* (*minimal t-semiflow*) is a p-semiflow (t-semiflow) such that the g.c.d of its non-null components is one and its support $\|Y\|$ ($\|X\|$) is not an strict superset of the support of another p-semiflow (t-semiflow).

A *path* π of a P/T net $\mathcal{N}$ is a sequence of nodes $\pi = x_1\ x_2\ ...\ x_n$ such that the odd components are places and the even components transitions, or viceversa, and for every pair (x_i, x_{i+1}), $W(x_i, x_{i+1}) \neq 0$. An *elementary path* is a path such that $\forall i, j \in [1, n]\ .\ x_i \neq x_j$, except for $x_1 = x_n$ (which is allowed). A *general circuit* is a path such that $x_1 = x_n$. An *elementary circuit* (or simply *circuit*) is both an elementary path and a general circuit.

B The Problem of Satisfiability(SAT)

Let $X = \{x_1, ..., x_n\}$ be a set of boolean variables. By the process of *truth assignment*, every variable in X is assigned one value: either true or false. Let $x_i \in X$, we call a *literal* to either x_i or its negation, $\overline{x_i}$. Intuitively, if the variable x_i is assigned the value true, the literals x_i and $\overline{x_i}$ are true and false, respectively (and viceversa if false is assigned). We define a *clause* $\mathcal{C}_j$ as a non-empty set of literals. The value of a clause is the disjunction of its literals, i.e., it is true iff at least one literal is true; and false otherwise. Finally, a *formula* $\mathcal{F}$ is a non-empty set of clauses, and its value is the conjunction of them, i.e., it is true iff all its clauses are true; false otherwise.

Without loss of generality, we will assume that, given a formula $\mathcal{F} = \mathcal{C}_1 \cdot ... \cdot \mathcal{C}_k$ and the set of its variables X, every variable $x_i \in X$ appears in at least one clause, and also that x_i appears at most once in each clause, be it negated or not.

Problem 7. **SATISFIABILITY (SAT)**
Given: A formula F and the set of its variables X.
To decide: Is there a truth assignment for X such that $\mathcal{F}$ is true?