Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Linear Programming Techniques for the Identification of Place/Transition Nets

Maria Paola Cabasino, Alessandro Giua, Carla Seatzu

Abstract— **In previous works we presented a procedure based on integer programming to identify a Petri net, given a finite prefix of its language. In this paper we show how to tackle the same problem using linear programming techniques, thus significantly reducing the complexity of finding a solution. The procedure we propose identifies a net whose number of places is equal to the cardinality of the set of disabling constraints. We provide a criterion to check if the computed solution has a minimal number of places, and, if such is not the case, we discuss two approaches to reduce this number.**

I. Introduction

Identification is a classical problem in system theory: given an observed behavior, it consists in determining a system whose behavior approximate the observed one.

In the context of Petri nets, it is common to consider as observed behavior the language of the net, i.e., the set of transition sequences that can be fired starting from the initial marking. Assume that a language $\mathcal{L} \subset T^*$ is given, where T is a given set of n transitions. Let this language be finite, prefix-closed and let k an integer greater than or equal to the length of the longest string it contains. The identification problem we consider consists in determining the structure of a net N, i.e., the matrices $Pre, Post \in \mathbb{N}^{m\times n}$, and its initial marking $M_0 \in \mathbb{N}^m$ such that the set of all firable transition sequences of length less than or equal to k is $L_k(N, M_0) = \mathcal{L}$.

Note that the set $\mathcal{L}$ explicitly lists *positive examples*, i.e., strings that are known to belong to the language, but also, implicitly, defines several *counterexamples*, namely all those strings of length less than or equal to k that do not belong to the language. Thus from the observed language one can construct a set of *enabling* constraints $\mathcal{E}$, i.e., a set of pairs (y, t), such that transition t should be enabled after sequence σ has fired, where y is the firing vector of σ, and a set of *disabling* constraints $\mathcal{D}$, i.e., a set of pairs (y, t) such that transition t should not be enabled after sequence σ has fired, where y is the firing vector of σ.

In previous work [3] we showed that this problem (and a related series of more general identification problems) can be solved using integer programming. The main drawback of this approach is its computational complexity, in the sense that the number of variables grows exponentially with the length of the longest string in $\mathcal{L}$, and problems of this kind may easily become intractable.

In this paper we show how to tackle the same problem using linear programming techniques, thus significantly reducing the complexity of solving an identification problem. The main idea is to look for particular solutions of the identification problem that are called $\mathcal{D}$-canonical, i.e., nets with a number of places equal to the cardinality of the set $\mathcal{D}$. We show that: (a) if a given identification problem has solution, then it also has a $\mathcal{D}$-canonical solution; (b) this particular solution can be computed solving a linear programming problem.

The procedure we propose identifies a net whose number of places is equal to the cardinality of the set of disabling constraints, which may be large, although an equivalent net with a much smaller number of places may exist. We provide a criterion to check if the computed solution has a minimal number of places, and, if such is not the case, we discuss two approaches to reduce this number.

In an on-line learning context, that is usually adapted in the traditional identification paradigms, positive and negative examples are presented to the learner on-the-fly. The learner holds a current hypothesis/model that supports all positive examples (and none of the negative examples) provided thus far. This problem, that from a computational point of view is easier than that considered in this paper, can be solved using our procedure considering only positive examples and negative examples and giving no constraints on strings that are neither positive nor negative examples.

Among previous approaches for Petri net identification (see [3] for a detailed discussion) we would like to recall the work of Hiraishi [9], Meda and Mellado [12], [13], Bourdeaud'huy and Yim [2], Dotoli *et al.* [6], [7], Li *et al.* in [10], Chung *et al.* [4]. We also mention the approach based on the *theory of regions* whose objective is that of deciding whether a given graph is isomorphic to the reachability graph of some free labeled net and then constructing it (see Badouel and Darondeau [1] for a survey).

II. Preliminaries

In this section we first recall the Petri net formalism used in the paper, referring to [14] for a comprehensive introduction to Petri nets.

Then we define a special class of linear constraint sets and prove an important property of such a class, that will be useful in the solution of our identification problem.

A. Background on Petri nets

A *Place/Transition net* (P/T net) is a structure $N = (P, T, Pre, Post)$, where P is a set of m places; T is a set

M.P. Cabasino, A. Giua and C. Seatzu are with the Dept. of Electrical and Electronic Engineering, University of Cagliari, Piazza D'Armi, 09123 Cagliari, Italy. Email: {cabasino,giua,seatzu}@diee.unica.it.

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

of n transitions; $Pre: P \times T \to \mathbb{N}$ and $Post: P \times T \to \mathbb{N}$ are the *pre*– and *post*– incidence functions that specify the arcs; $C = Post - Pre$ is the incidence matrix.

A *marking* is a vector $M : P \to \mathbb{N}$ that assigns to each place of a P/T net a nonnegative integer number of tokens, represented by black dots. We denote $M(p)$ the marking of place p. A P/T *system* or *net system* $\langle N, M_0 \rangle$ is a net N with an initial marking M_0.

A transition t is enabled at M iff $M \geq Pre(\cdot, t)$ and may fire yielding the marking $M' = M + C(\cdot, t) = M + C \cdot \vec{t}$, where $\vec{t} \in \mathbb{N}^n$ is a vector whose components are all equal to 0 except the component associated to transition t that is equal to 1. We write $M\ [\sigma\rangle$ to denote that the sequence of transitions σ is enabled at M, and we write $M\ [\sigma\rangle\ M'$ to denote that the firing of σ yields M'. Note that in this paper we always assume that two or more transitions cannot simultaneously fire (non-concurrency hypothesis).

A marking M is *reachable* in $\langle N, M_0 \rangle$ iff there exists a firing sequence σ such that $M_0\ [\sigma\rangle\ M$. In such a case the state equation $M = M_0 + C \cdot \vec{\sigma}$ holds, where $\vec{\sigma} \in \mathbb{N}^n$ is the *firing vector* of σ, i.e., the vector whose ith entry represents the number of times the transition t_i is contained in σ. The set of all markings reachable from M_0 defines the *reachability set* of $\langle N, M_0 \rangle$ and is denoted $R(N, M_0)$.

Given a Petri net system $\langle N, M_0 \rangle$ we define its *free-language*[1] as the set of its firing sequences

$$L(N, M_0) = \{\sigma \in T^* \ \mid \ M_0[\sigma\rangle\}.$$

We also define the set of firing sequences of length less than or equal to $k \in \mathbb{N}$ as:

$$L_k(N, M_0) = \{\sigma \in L(N, M_0) \ \mid \ |\sigma| \leq k\}.$$

Finally given a language $\mathcal{L} \subset T^*$ and a vector $y \in \mathbb{N}^n$ we denote

$$\mathcal{L}(y) = \{\sigma \in \mathcal{L} \mid \vec{\sigma} = y\}$$

the set of all sequences in $\mathcal{L}$ whose firing vector is y.

B. Special constraint sets

We define a special class of linear constraint sets (CS).

***Definition** 1:* Given $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, consider the linear constraint set:

$$\mathcal{C}(A, b) = \{x \in \mathbb{R}^n \mid Ax \geq b\}.$$

The set $\mathcal{C}(A, b)$ is called:

- *ideal:* if $x \in \mathcal{C}(A, b)$ implies $\alpha x \in \mathcal{C}(A, b)$ for all $\alpha \geq 1$;
- *rational:* if $A \in \mathbb{Q}^{m \times n}$ and $b \in \mathbb{Q}^m$, i.e., if the entries of matrix A and of vector b are rational. ■

The following result provides a simple characterization of ideal CS's.

***Proposition** 2:* A linear constraint set $\mathcal{C}(A, b)$ is ideal if $b \geq 0$.

Proof. Since $Ax \geq b \geq 0$ then for all $\alpha \geq 1$ it holds $A(\alpha x) \geq Ax \geq b$, hence it is ideal. □

***Proposition** 3:* If a CS is ideal and rational, then it has a feasible solution if and only if it has a feasible integer solution.

Proof. The *if* part is trivial.

To prove the *only if* part, we reason as follows. If there exists a solution there exists a basis solution x_B , i.e., such that

$$x_B = A_B^{-1} b,$$

where A_B is obtained by A selecting a set of basis columns.

If the CS is rational the entries of A_B and b are rational, hence the entries of A_B^{-1} and of x_B are rational as well.

If the CS is ideal, we just need to multiply the rational vector x_B by a suitable positive integer to obtain an integer solution. □

III. P/T NET IDENTIFICATION

The problem we consider in this paper can be formally stated as follows.

***Problem** 4:* Let $\mathcal{L} \subset T^*$ be a finite prefix-closed language[2], and

$$k \geq \max_{\sigma \in \mathcal{L}} |\sigma|$$

be an integer greater than or equal to the length of the longest string in $\mathcal{L}$. We want to identify the structure of a net $N = (P, T, Pre, Post)$ and an initial marking M_0 such that

$$L_k(N, M_0) = \mathcal{L}.$$

The unknowns we want to determine are the elements of the two matrices $Pre,\ Post \in \mathbb{N}^{m \times n}$ and the elements of the vector $M_0 \in \mathbb{N}^m$. ■

Associated to an identification problem are the two sets defined in the following.

***Definition** 5:* Let $\mathcal{L} \subset T^*$ be a finite prefix-closed language and let $k \in \mathbb{N}$ be defined as in Problem 4.

We define the *set of enabling conditions*

$$\begin{aligned} \mathcal{E} &= \{(y,t) \mid (\exists \sigma \in \mathcal{L}) : |\sigma| < k, \sigma \in \mathcal{L}(y), \sigma t \in \mathcal{L}\} \\ &\subset \mathbb{N}^n \times T \end{aligned} \tag{1}$$

and the *set of disabling conditions*

$$\begin{aligned} \mathcal{D} &= \{(y,t) \mid (\exists \sigma \in \mathcal{L}) : |\sigma| < k, \sigma \in \mathcal{L}(y), \sigma t \notin \mathcal{L}\} \\ &\subset \mathbb{N}^n \times T. \end{aligned} \tag{2}$$

■

Clearly, a solution to Problem 4 is a net $\langle N, M_0 \rangle$ such that:

- for all $(y, t) \in \mathcal{E}$ transition t is enabled after the firing of all $\sigma \in \mathcal{L}(y)$, i.e., $M_0[\sigma\rangle M_y[t\rangle$, where $M_y = M_0 + C \cdot y$ represents the marking reached after the firing of sequence σ.
- for all $(y, t) \in \mathcal{D}$ transition t is disabled after the firing of $\sigma \in \mathcal{L}(y)$, i.e., $M_0[\sigma\rangle M_y \neg[t\rangle$.

[1] As it will appear in the next subsection, *free* specifies that no labeling function is assigned to the considered Petri net system.

[2] A language $\mathcal{L}$ is said to be *prefix-closed* if for any string $\sigma \in \mathcal{L}$, all prefixes of σ are in $\mathcal{L}$.

We can characterize the number of places required to solve our identification problem.

***Definition** 6:* Let $\mathcal{L}$ be a finite prefix-closed language on alphabet T, whose words have length less than or equal to k. Given the set of disabling conditions (2) let $m_{\mathcal{D}} = |\mathcal{D}|$.

We say that a Petri net system $\langle N, M_0 \rangle$ with set of places P and $L_k(N, M_0) = \mathcal{L}$, is *$\mathcal{D}$-canonical* if

1) $|P| = m_{\mathcal{D}}$;
2) there exists a bijective mapping $h : \mathcal{D} \to P$ such that, for all $(y, t) \in \mathcal{D}$, place $p = h(y, t)$ satisfies

$$M_y(p) \triangleq M_0(p) + C(p, \cdot) \cdot y < Pre(p, t),$$

i.e., place p disables t after any $\sigma \in \mathcal{L}(y)$. ■

In simple words, a net system $\langle N, M_0 \rangle$ is *$\mathcal{D}$-canonical* if a different place is associated to each element in the set of disabling constraints $\mathcal{D}$.

***Proposition** 7:* Let $\mathcal{L}$ be a finite prefix-closed language on alphabet T, whose words have length less than or equal to k.

If there exists a net system $\langle \tilde{N}, \tilde{M}_0 \rangle$ such that $L_k(\tilde{N}, \tilde{M}_0) = \mathcal{L}$, then there exists a net system $\langle N, M_0 \rangle$ that is $\mathcal{D}$-canonical.

Proof. Let

$$P_{(y,t)} = \{p \in \tilde{P} \mid \tilde{M}_0(p) + \tilde{C}(p, \cdot) \cdot y < \widetilde{Pre}(p, t)\}$$

be the set of all places of $\tilde{N}$ that disable transition t after sequence $\sigma \ \in \mathcal{L}(y)$ has occurred (here $\tilde{C}$ is the incidence matrix of $\tilde{N}$). For each pair $(y, t) \in \mathcal{D}$ let $h(y, t)$ be one place arbitrary selected from $P_{(y,t)}$; let P be the set of selected places and $m = |P|$.

Two different cases may occur.

Case 1: $m = m_{\mathcal{D}}$, i.e., a different place has been selected from any set $P_{(y,t)}$. In such a case we define N as the net obtained from $\tilde{N}$ removing all places not in P (if any), and assume M_0 as the restriction of $\tilde{M}_0$ to places in P. We claim that $L_k(N, M_0) = \mathcal{L}$. In fact, since we have removed some places from $\langle \tilde{N}, \tilde{M}_0 \rangle$ then $L(\tilde{N}, \tilde{M}_0) \subseteq L(N, M_0)$. On the other hand, by construction we know that for all words wt of length less than or equal to k it holds

$$wt \notin L(\tilde{N}, \tilde{M}_0) \qquad \Longrightarrow \qquad wt \notin L(N, M_0),$$

hence $L_k(N, M_0) = L_k(\tilde{N}, \tilde{M}_0) = \mathcal{L}$.

By construction, a different place in P is associated to any couple $(y, t) \in \mathcal{D}$, thus proving that $\langle N, M_0 \rangle$ is $\mathcal{D}$-canonical.

Case 2: $m < m_{\mathcal{D}}$, i.e., some place $p \in P$ has been selected from $P_{(y,t)}, P_{(y',t')}, P_{(y'',t'')}, \ldots$, for two or more different pairs $(y, t), (y', t'), (y'', t''), \ldots$ in $\mathcal{D}$. In such a case we add to the net — without changing its language — additional places $p', p'' \ldots$ such that $Pre(p, \cdot) = Pre(p', \cdot) = Pre(p'', \cdot) = \cdots$, $Post(p, \cdot) = Post(p', \cdot) = Post(p'', \cdot) = \cdots$, and $M_0(p) = M_0(p') = M_0(p'') = \cdots$, thus obtaining a net with $m_{\mathcal{D}}$ places.

We redefine $h(y', t') = p'$, $h(y'', t'') = p''$, ... Function h is now bijective and the resulting net system is $\mathcal{D}$-canonical. □

***Theorem** 8:* Let us consider a finite prefix-closed language on alphabet T, whose words have length less than or equal to k and let $\mathcal{E}$ and $\mathcal{D}$ be the corresponding sets of enabling and disabling conditions.

Let

$$\mathcal{N}(\mathcal{E}, \mathcal{D}) \triangleq$$
$$\begin{cases} M_0 + Post \cdot y & \\ \qquad -Pre \cdot (y + \vec{t}\,) \geq 0 & \forall (y, t) \in \mathcal{E} \\ M_0(p_{(y,t)}) + Post(p_{(y,t)}, \cdot) \cdot y & \\ \qquad -Pre(p_{(y,t)}, \cdot) \cdot (y + \vec{t}\,) \leq -1 & \forall (y, t) \in \mathcal{D} \\ M_0 \in \mathbb{R}_{\geq 0}^{m_{\mathcal{D}}} & \\ Pre, Post \in \mathbb{R}_{\geq 0}^{m_{\mathcal{D}} \times n} & \end{cases} \tag{3}$$

Consider a net system $\langle N, M_0 \rangle$ with $N = (P, T, Pre, Post)$. The system $\langle N, M_0 \rangle$ is a $\mathcal{D}$-canonical solution of the identification problem 4 iff Pre, $Post$, M_0 are integer solutions of CS (3).

Proof: We first show that any integer solution $\langle N, M_0 \rangle$ of CS (3) is a solution of Problem 4.

- Any constraint $M_0 + Post \cdot y - Pre \cdot (y + \vec{t}\,) \geq 0$ can be rewritten as $M_y = M_0 + (Post - Pre) \cdot y \geq Pre(\cdot, t)$ or equivalently $M_y \geq Pre(\cdot, t)$ where $M_0[\sigma\rangle M_y$ for all $\sigma \in \mathcal{L}(y)$. This shows that transition t is enabled on $\langle N, M_0 \rangle$ from marking M_y and by induction on the length of σ (since language $\mathcal{L}$ is prefix-closed) we conclude that $\sigma t \in \mathcal{L}$.
- Assume that sequence $\sigma \in \mathcal{L}(y)$ is firable on the net and $M_0[\sigma\rangle M_y$. If for at least a place p in the net it holds $M_0(p) + Post(p, \cdot) \cdot y - Pre(p, \cdot) \cdot (y + \vec{t}\,) \leq -1$, then $M_y = M_0 + (Post - Pre) \cdot y \ngeq Pre(\cdot, t)$ or equivalently $M_y \ngeq Pre(\cdot, t)$. This shows that transition t is not enabled on $\langle N, M_0 \rangle$ from marking M_y and we conclude that $\sigma t \notin \mathcal{L}$.

Since net $\langle N, M_0 \rangle$ satisfies all enabling and disabling constraints, $L_k(N, M_0) = \mathcal{L}$.

We now show that any solution of CS (3) is $\mathcal{D}$-canonical. In fact, the mapping $h(y, t) = p_{(y,t)}$ for each couple $(y, t) \in \mathcal{D}$ is bijective.

We now show that any $\mathcal{D}$-canonical net system $\langle N, M_0 \rangle$ with $L_k(N, M_0) = \mathcal{L}$ is a solution of CS (3). In fact, let $h : \mathcal{D} \to P$ be the bijective function of the net system. If we define $p_{(y,t)} = h(y, t)$ for all $(y, t) \in \mathcal{D}$, then all equations in CS (3) are satisfied. □

***Proposition** 9:* The linear CS (3) is ideal and rational.

Proof: We first observe that the linear CS (3) can be rewritten as a set of linear inequalities of the form $Ax \geq b$ as follows. Let us denote as pre_i and $post_i$ the i-th row of matrices Pre and $Post$, respectively, for $i = 1, \ldots, m_{\mathcal{D}}$.

For any $(y, t) \in \mathcal{E}$ the first matrix inequality in (3) can be rewritten as the following set of $m_{\mathcal{D}}$ scalar inequalities:

$$\begin{bmatrix} 1 & y^T & -(y + \vec{t}\,)^T \end{bmatrix} \cdot \begin{bmatrix} M_{0,i} \\ post_i^T \\ pre_i^T \end{bmatrix} \geq 0$$

where $i = 1, \ldots, m_{\mathcal{D}}$. Analogously, for any $(y, t) \in \mathcal{D}$ the second scalar inequality in (3) can be rewritten as:

$$\begin{bmatrix} -1 & -y^T & (y + \vec{t}\,)^T \end{bmatrix} \cdot \begin{bmatrix} M_{0,i} \\ post_i^T \\ pre_i^T \end{bmatrix} \geq 1.$$

This defines matrix A and vector b. Since A and b have integer entries, CS (3) is rational. Since $b \geq 0$, by Proposition 2 CS (3) is ideal. □

The following theorem provides a practical and efficient procedure to solve our identification problem.

***Theorem** 10:* The identification problem 4 admits a solution if and only if the (linear) CS (3) is feasible.

Proof: (if) By Proposition 9 CS (3) is ideal and rational thus by Proposition 3 if it has solutions, then it also has feasible integer solutions. However, by Theorem 8 this also implies that such integer solutions are also solutions of the identification problem 4.

(only if) If there exists a solution of Problem 4, by Proposition 7 there also exists a net system that is $\mathcal{D}$-canonical. But all $\mathcal{D}$-canonical systems are solutions of CS (3) by Theorem 8, thus CS (3) is feasible. □

Note, finally, that once a solution of CS (3) is found, if this solution is rational we can always find an integer solution by simply multiplying Pre, $Post$ and M_0 by a suitable $\alpha \geq 1$.

IV. Place reduction

One drawback of the identification procedure outlined in the previous section consists in the requirement that the net contains a number of places equal to $m_{\mathcal{D}} = |\mathcal{D}|$ although an equivalent net with a smaller number of places may exist. Note that in the worst case the cardinality of the set $\mathcal{D}$ can be $|T|^k$ [3].

We propose two (types of) approaches to overcome this problem. In the first approach, that we call *place pre-reduction*, CS (3) is written in a modified form, using a reduced number of places. In the second approach, that we call *place post-reduction*, we first determine a solution of the standard CS (3) obtaining a net with $m_{\mathcal{D}}$ places and then we identify redundant places that can be removed without affecting the correctness of the result.

A. Place pre-reduction

We start with a general result that allows one to check if the net obtained by solving CS (3) has a minimal number of places. The test we propose requires solving a series of modified CS's and this is why we present this result in the subsection devoted to the pre-reduction.

***Definition** 11:* Consider a partition

$$\Pi(\mathcal{D}) = \{D_1, D_2, \ldots, D_q\}$$

of the set $\mathcal{D}$. The sets D_i are called *blocks* of partition $\Pi(\mathcal{D})$.

We define the following CS

$$\mathcal{N}(\mathcal{E}, \Pi(\mathcal{D})) \triangleq \begin{cases} M_0 + Post \cdot y & \\ \qquad -Pre \cdot (y + \vec{t}\,) \geq 0 & \forall (y,t) \in \mathcal{E} \\ M_0(p_i) + Post(p_i, \cdot) \cdot y & \\ \qquad -Pre(p_i, \cdot) \cdot (y + \vec{t}\,) < 0 & \forall (y,t) \in D_i, \\ & i = 1, \ldots, q \\ M_0 \in \mathbb{R}^q_{\geq 0} & \\ Pre, Post \in \mathbb{R}^{q \times n}_{\geq 0} & \end{cases} \tag{4}$$

where $\mathcal{E}$ and $n = |T|$ have the usual meaning as in Theorem 10. ■

The only difference between CS (4) and CS (3) consists in the fact that in the former only q places (as many as the blocks of partition $\Pi(\mathcal{D})$) are used: place p_i ($i = 1, \ldots, q$) will ensure that all disabling conditions in D_i are enforced. It is immediate to prove, with the same reasoning of Proposition 9, that CS (4) is rational and ideal, and that any of its integer solutions is a solution to the identification problem 4.

***Definition** 12:* Given the identification problem 4, a partition $\Pi(\mathcal{D}) = \{D_1, D_2, \ldots, D_q\}$ with q blocks is said to be

- *feasible* if CS $\mathcal{N}(\mathcal{E}, \Pi(\mathcal{D}))$ admits a solution;
- *minimal* if it is feasible and there exists no other partition $\Pi'(\mathcal{D})$ with $q' < q$ that is feasible.

■

Thus the number of blocks of a minimal partition represents the minimal number of places that a net solving the given identification problem may have.

The following corollary trivially follows from the previous definitions and from Theorem 10.

***Corollary** 13:* A net with q places solution of the identification problem 4 exists iff there exists a feasible partition $\Pi(\mathcal{D}) = \{D_1, D_2, \ldots, D_q\}$ with q blocks solution of CS (4).

We now state an intuitive result that allows one to determine if a partition is minimal.

***Proposition** 14:* A feasible partition with q blocks is minimal iff there exists no feasible partition with $q - 1$ blocks.

Proof: The *only if* part follows from the definition of minimal partition.

To prove the *if* part we need to show that if no feasible partition with $q - 1$ blocks exists, then no partition with a smaller number of blocks is feasible. This can be proved by contradiction, by means of the same argument used in the proof of Proposition 7, Case 2. In fact, assume there exists a feasible partition with $q' < q - 1$ blocks; then there exists a net solving the identification problem with q' places. However, we can add an arbitrary number of duplicate place to this net and this implies that there exists a net solving the identification problem with $q' + 1, q' + 2, \ldots, q - 1$ places. Thus, according to Proposition 13, there exists a feasible partition with $q - 1$ blocks which is a contradiction. □

According to the previous proposition to prove that a feasible partition $\Pi(\mathcal{D})$ with q blocks is minimal it is necessary to check the feasibility of all partitions $\mathcal{D}$ with $q - 1$ blocks. However, the number of partitions of a set of cardinality n into k blocks is given by the *Sterling number of the second kind*[3] $S(n,k)$ [5] which may be too large for an exhaustive analysis.

With the terminology introduced in this section, CS (3) can be seen as a special case of CS (4) when the considered

[3] An explicit formula for the Sterling number of the second kind is

$$S(n,k) = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^n.$$

partition $\Pi(\mathcal{D})$ contains all singleton sets, i.e., it is the unique partition with $m_{\mathcal{D}}$ blocks. Thus it is possible to check if this partition is minimal by checking that all partitions of $m_{\mathcal{D}}-1$ blocks (obtained by merging any two singleton sets) are not feasible. There exists

$$S(m_{\mathcal{D}}, m_{\mathcal{D}}-1) = \frac{m_{\mathcal{D}}(m_{\mathcal{D}}-1)}{2}$$

of these partitions.

Although the previous results provide a procedure for reducing the number of places of a Petri net, a brute force search to determine a minimal feasible partition is not viable given the large number of such partitions. We conclude this subsection with an informal discussion on how it may possible to exploit some additional information on the net to determine a feasible partition of cardinality smaller than $m_{\mathcal{D}}$.

As an example, assume it is known that a transition t has only one input place — such is the case if the net to identify or a subnet of it containing t is a state machine. In such a case, it is possible to consider a partition of $\mathcal{D}$ in which a single block

$$D = \{(y_1, t), \ldots, (y_r, t)\} \subset \mathcal{D}$$

contains all disabling conditions for transition t and a single place p will be used to disable t after a sequence $\sigma_i \in \cup_{i=1}^{r} \mathcal{L}(y_i)$ has been executed.

As a second example, assume it is known that transitions t and t' are in a free choice relation, i.e., there exists a place $p = {}^{\bullet}t = {}^{\bullet}t'$ that is the unique input place for both transitions. In such a case, it is possible to consider a partition of $\mathcal{D}$ in which a single block

$$D = \{(y_1, t), \ldots, (y_r, t), (y'_1, t'), \ldots, (y_{r'}, t')\} \subset \mathcal{D}$$

contains all disabling conditions for transitions t and t' which will be enforced by place p.

***Example** 15:* Let us consider a language

$$\mathcal{L} = \{\varepsilon, t_1, t_1t_2, t_1t_3, t_1t_2t_3, t_1t_3t_2, t_1t_3t_3\}$$

and let $k = 3$. We have additional information: the transition t_1 has only one input place and transitions t_2 and t_3 are in a free choice relation. The set of enabling and disabling constraints are respectively:

$$\mathcal{E} = \{(\varepsilon, t_1), (t_1, t_2), (t_1, t_3), (y_{12}, t_3), (y_{13}, t_2), (y_{13}, t_3)\}$$

and

$$\mathcal{D} = \{(\varepsilon, t_2), (\varepsilon, t_3), (t_1, t_1), (y_{12}, t_1), (y_{12}, t_2), (y_{13}, t_1)\},$$

where y_{12}, y_{13} are the firing vectors of t_1t_2 and t_1t_3 respectively. The additional information allows us to consider two different blocks of $\mathcal{D}$:

$$D_1 = \{(t_1, t_1), (y_{12}, t_1), (y_{13}, t_1)\} \subset \mathcal{D}$$

and

$$D_2 = \{(\varepsilon, t_2), (\varepsilon, t_3), (y_{12}, t_2)\} \subset \mathcal{D}.$$

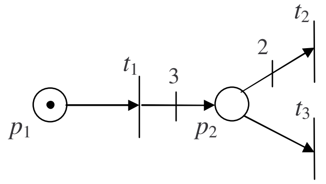


Fig. 1. The Petri net systems in Example 15.

Let us observe that $\mathcal{D} = D_1 \cup D_2$, then the Petri net solution has $|P| = 2$. A net system solution of $\mathcal{N}(\mathcal{E}, \mathcal{D})$ computed with a commercial LP solver (LINDO) is reported in Fig. 1.

Note that such a solution has been determined associating a linear objective function $f(M_0, Pre, Post)$ to $\mathcal{N}(\mathcal{E}, \mathcal{D})$, and solving the resulting linear programming problem. In particular, we assumed

$$f(M_0, Pre, Post) = 1^T \cdot M_0 + 1^T \cdot Pre \cdot 1 + 1^T \cdot Post \cdot 1.$$

Note finally that the solution we found out was integer. ■

B. Place post-reduction

Once a net has been identified solving CS (3) (or even CS (4)) it is always possible to check if some of the places are *redundant* and can be removed without affecting the correctness of the result. This check is based on the notion of *minimal hitting set* defined in the following.

***Proposition** 16:* Consider a net system $\langle N, M_0 \rangle$ solution of the identification problem 4 and define for all disabling conditions $(y, t) \in \mathcal{D}$ the set

$$P_{(y,t)} = \{p \in P \mid M_0(p) + C(p, \cdot) \cdot y < Pre(p, t)\} \quad (5)$$

consisting of all places of the net that disable transition t after a sequence $\sigma \in \mathcal{L}(y)$ has been executed.

Assume $\hat{P} \subset P$ is a hitting set for all $P_{(y,t)}$'s, i.e., $\hat{P} \cap P_{(y,t)} \neq \emptyset$ for all $(y, t) \in \mathcal{D}$. Then the net system $\langle \hat{N}, \hat{M}_0 \rangle$ obtained from $\langle N, M_0 \rangle$ removing all places in $P \setminus \hat{P}$ is a solution of the identification problem 4.

Proof: As already discussed in the proof of Proposition 7 the removal of a place does not affect any enabling condition. Furthermore, if $\hat{P}$ is a hitting set for all $P_{(y,t)}$'s then it is capable of enforcing all disabling conditions in $\mathcal{D}$. Hence $L_k(\hat{N}, \hat{M}_0) = L_k(N, M_0)$. □

The places in $P \setminus \hat{P}$ that can be removed from the net system $\langle N, M_0 \rangle$ without changing its language $L_k(N, M_0)$ are called *redundant places*.

Since the net $\tilde{N}$ has set of places $\hat{P}$, to obtain a net with a minimal set of places we need to determine the minimal hitting set. This problem is known to be NP-hard and there exists several ways to compute minimal hitting sets (see [11] for a review). Here we present a straightforward algorithm based on integer programming.

***Proposition** 17:* Consider a net system $\langle N, M_0 \rangle$ with $m = |P|$ places solution of the identification problem 4. For all disabling conditions $(y, t) \in \mathcal{D}$, let $z_{(y,t)} \in \{0, 1\}^m$

be the characteristic vector of set $P_{(y,t)}$ defined in (5), i.e., $z_{(y,t)}(p) = 1$ if $p \in P_{(y,t)}$, else $z_{(y,t)}(p) = 0$.

Consider the following integer programming problem (IPP):

$$\begin{array}{lll} \min & 1^T \cdot x & \\ s.t. & x^T \cdot z_{(y,t)} \geq 1 & \forall (y,t) \in \mathcal{D} \\ & x \in \{0,1\}^m & \end{array} \qquad (6)$$

and let x^* be an optimal solution.

Then a minimal hitting set for all $P_{(y,t)}$'s is the set $\hat{P} = \{p \in P \mid x^*(p) = 1\}$.

Proof: It is immediate to see that any feasible solution x of IPP (6) is the characteristic vector of a hitting set for all $P_{(y,t)}$'s because it contains at least one element from each of these sets ($x^T \cdot z_{(y,t)} \geq 1$). The optimal solution x^* has the minimal number of non-zero components, hence it corresponds to a minimal hitting set. □

***Example** 18:* Let

$$\mathcal{L} = \{\varepsilon, t_1, t_2\}$$

and $k = 2$. We observe that this is a particular case of a language containing no word of length k, i.e., all words of length k have to be disabled.

The set of enabling and disabling constraints are respectively:

$$\mathcal{E} = \{(\varepsilon, t_1), (\varepsilon, t_2)\}$$

and

$$\mathcal{D} = \{(t_1, t_1), (t_1, t_2), (t_2, t_1), (t_2, t_2)\}.$$

A net system solution of $\mathcal{N}(\mathcal{E}, \mathcal{D})$ is reported in Fig. 2.(a): here it is obviously $|P| = m_{\mathcal{D}} = 4$.

Note that such a solution has been determined associating the same linear objective function $f(M_0, Pre, Post)$ used in Example 15 to $\mathcal{N}(\mathcal{E}, \mathcal{D})$, and solving the resulting linear programming problem.

We now try to reduce the number of places using the place post-reduction approach. To this aim for any (y,t) we compute the set $P_{(y,t)}$ defined as in equation (5). Being $P_{(t_1,t_1)} = \{p_{11}, p_{12}, p_{21}\}$, $P_{(t_1,t_2)} = \{p_{12}, p_{21}\}$, $P_{(t_2,t_1)} = \{p_{12}, p_{21}\}$, $P_{(t_2,t_2)} = \{p_{12}, p_{21}, p_{22}\}$, it is immediate to see that a possible hitting set for all $P_{(y,t)}$'s is $\hat{P} = \{p_{21}\}$, that is also minimal. The resulting net system $\langle \hat{N}, \hat{M}_0 \rangle$ obtained from the previous one removing all places in $P \setminus \hat{P}$ is shown in Fig. 2.(b). All solutions, obtained with LINDO, are integer. ■

It is important to observe that the place post-reduction procedure does not necessarily ensure that the resulting net is the solution of a given identification procedure with the minimal number of places. In fact, it only determines, amongst the solutions of a given identification procedure *that can be obtained from a solution N by removing places*, the one with a minimal number of places. The following example will clarify this point.

***Example** 19:* Let

$$\mathcal{L} = \{\varepsilon, t_1, t_2, t_3\}$$

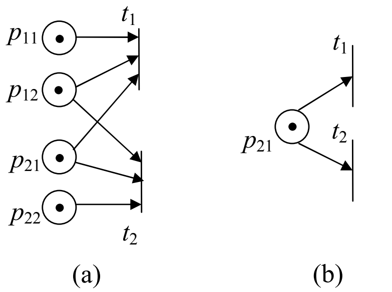
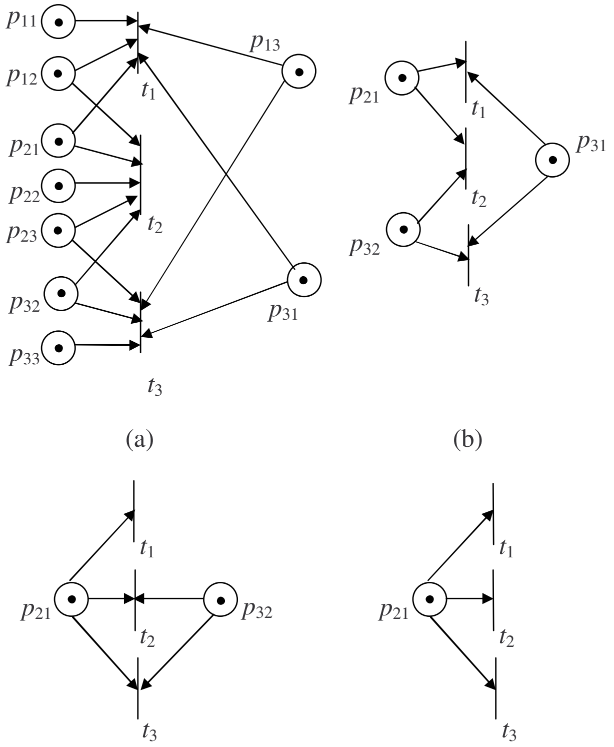


Fig. 2. The Petri net systems in Example 18.



(a) (b)

(c) (d)

Fig. 3. The Petri net systems in Example 19.

and $k = 2$, thus as in Example 18, all words of length k have to be disabled. The set of enabling and disabling constraints are respectively:

$$\mathcal{E} = \{(\varepsilon, t_1), (\varepsilon, t_2), (\varepsilon, t_3)\}$$

and

$$\begin{array}{rcl} \mathcal{D} & = & \{(t_1, t_1), (t_1, t_2), (t_1, t_3), (t_2, t_1), (t_2, t_2), \\ & & (t_2, t_3), (t_3, t_1), (t_3, t_2), (t_3, t_3)\}. \end{array}$$

A net system solution of CS (3) is shown in Fig. 3.(a), where obviously $|P| = m_{\mathcal{D}} = 9$. It has been determined solving a linear programming problem whose objective function is the same of that used in Example 15.

Now, being $P_{(t_1,t_1)} = \{p_{11}, p_{12}, p_{13}, p_{21}, p_{31}\}$, $P_{(t_1,t_2)} = \{p_{12}, p_{21}\}$, $P_{(t_1,t_3)} = \{p_{13}, p_{31}\}$, $P_{(t_2,t_1)} = \{p_{12}, p_{21}\}$, $P_{(t_2,t_2)} = \{p_{12}, p_{21}, p_{22}, p_{23}, p_{32}\}$, $P_{(t_2,t_3)} = \{p_{23}, p_{32}\}$,

$P_{(t_3,t_1)} = \{p_{13}, p_{31}\}$, $P_{(t_3,t_2)} = \{p_{23}, p_{32}\}$, $P_{(t_3,t_3)} = \{p_{13}, p_{23}, p_{31}, p_{32}, p_{33}\}$, it is easy to see that a possible hitting set for all $P_{(y,t)}$'s is $\hat{P} = \{p_{21}, p_{31}, p_{32}\}$. The net system $\langle \hat{N}, \hat{M}_0 \rangle$ obtained from $\langle N, M_0 \rangle$ removing all places in $P \setminus \hat{P}$ is shown in Fig. 3.(b).

Note that the same hitting set can also be obtained solving the IPP (6).

Such a solution is not minimal in terms of places, as it can be verified using the place pre-reduction procedure.

To this aim we first look at a solution with two places, namely using the notation of Definition 11, we check if there exists a solution with only two blocks $\mathcal{D}'_1$ and $\mathcal{D}'_2$, obtained by merging two of the three blocks

$$\mathcal{D}_1 = \{(t_1,t_1),(t_1,t_2),(t_2,t_1),(t_2,t_2)\},$$

$$\mathcal{D}_2 = \{(t_2,t_3),(t_3,t_2),(t_3,t_3)\}$$

and

$$\mathcal{D}_3 = \{(t_1,t_3),(t_3,t_1)\}$$

relative respectively to the places p_{21}, p_{32} and p_{31} of the net in Fig. 3.(b).

In particular, we find that CS (4) admits a solution when

$$\mathcal{D}'_1 = \{t_1,t_1),(t_1,t_2),(t_1,t_3),(t_2,t_1),(t_2,t_2),(t_3,t_1)\}$$

and

$$\mathcal{D}'_2 = \{(t_2,t_3),(t_3,t_2),(t_3,t_3)\},$$

where

$$\mathcal{D}'_1 = \mathcal{D}_1 \cup \mathcal{D}_3.$$

The resulting net $\langle \bar{N}, \bar{M}_0 \rangle$ is shown in Fig. 3.(c), where $\bar{P} = \{p_{21}, p_{32}\}$.

We can further on reduce the net finding a minimal hitting set. Being, $P_{(t_1,t_1)} = \{p_{21}\}$, $P_{(t_1,t_2)} = \{p_{21}\}$, $P_{(t_1,t_3)} = \{p_{21}\}$, $P_{(t_2,t_1)} = \{p_{21}\}$, $P_{(t_2,t_2)} = \{p_{21}, p_{32}\}$, $P_{(t_2,t_3)} = \{p_{21}, p_{32}\}$, $P_{(t_3,t_1)} = \{p_{21}\}$, $P_{(t_3,t_2)} = \{p_{21}, p_{32}\}$, $P_{(t_3,t_3)} = \{p_{21}, p_{32}\}$, the unique minimal hitting set is $\hat{P}' = \{p_{21}\}$. The net system $\langle \hat{N}', \hat{M}'_0 \rangle$ obtained from $\langle \bar{N}, \bar{M}_0 \rangle$ removing all places in $\bar{P} \setminus \hat{P}'$ is minimal and is shown in Fig. 3.(d).

Note that in all cases, the net solutions, obtained with LINDO, are integer. ■

As a final remark we observe that the notion of redundant place that we give here is different from the one of *implicit place* used by other authors [8]. In fact, an implicit place is a place that can be removed from a net system $\langle N, M_0 \rangle$ without changing its overall behavior $L(N, M_0)$. On the contrary, a redundant place according to our definition is a place that can be removed from the net without changing the finite prefix behavior $L_k(N, M_0)$. Thus our notion is weaker and the techniques used in [8] to determine implicit places cannot be used in our framework.

V. Conclusions

We have presented a procedure to identify a place/transition net from a finite prefix of its language. The procedure is based on linear programming, and this is a major advantage over previous approaches that were based on integer programming. The number of places of the resulting net may not be minimal, but we have discussed several techniques that may be used to reduce it.

As a possible line for future research, we mention the extension of this procedure to more general cases, such as those based on the knowledge of the reachability/coverability tree of the net to identify. Moreover, we plan to use the results here presented in selected application domains.

References

[1] E. Badouel and P. Darondeau. Theory of regions. *Lecture Notes in Computer Science: Lectures on Petri Nets I: Basic Models*, 1491:529–586, 1998.

[2] T. Bourdeaud'huy and P. Yim. Synthse de rseaux de Petri partir d'exigences. In *Actes de la 5me conf. francophone de Modlisation et Simulation*, pages 413–420, Nantes, France, September 2004.

[3] M.P. Cabasino, A. Giua, and C. Seatzu. Identification of Petri nets from samples of their languages. *DEDS*, 17(4):447–474, 2007.

[4] S.L. Chung, C.L. Li, J.C. Wu, and S.T. Wang. Online modeling refinement for discrete event systems. *Proc. of the IEEE Conference on Systems, Man and Cybernetics*, 3:2739–2744, 2003.

[5] W.F. (IV) Doran and D.B. Wales. The partition algebra revisited. *Journal of Algebra*, 231(1):265–330, September 2000.

[6] M. Dotoli, M.P. Fanti, and A.M. Mangini. On-line identification of discrete event systems by interpreted Petri nets. *Proc. of the IEEE Conference on Systems, Man and Cybernetics*, 4:3040–3045, 2006.

[7] M. Dotoli, M.P. Fanti, and A.M. Mangini. Real time identification of discrete event systems using Petri nets. *Automatica*, 4(5):1209–1219, 2008.

[8] F. Garcia-Valles and J.M Colom. Implicit places in net systems. In *Proc. of the 8th Int. Work. on Petri Nets and Performance Models*, pages 104 – 113, Zaragoza, Spain, 1999.

[9] K. Hiraishi. Construction of a class of safe Petri nets by presenting firing sequences. In Jensen, K., editor, *Lecture Notes in Computer Science; 13th International Conference on Application and Theory of Petri Nets 1992, Sheffield, UK*, volume 616, pages 244–262. Springer-Verlag, June 1992.

[10] L.X. Li, Y. Ru, and C.N. Hadjicostis. Least-cost firing sequence estimation in labeled Petri nets. In *Proc. 45th IEEE Conf. on Decision and Control*, San Diego, California USA, December 2006.

[11] L. Lin and Y. Jiang. The computation of hitting sets: Review and new algorithms. *Information Processing Letters*, 86:177–184, 2003.

[12] M.E. Meda-Campaa and E. Lpez-Mellado. Incremental synthesis of Petri net models for identification of discrete event systems. In *Proc. 41th IEEE Conf. on Decision and Control*, pages 805–810, Las Vegas, Nevada USA, December 2002.

[13] M.E. Meda-Campaa and E. Lpez-Mellado. Required event sequences for identification of discrete event systems. In *Proc. 42th IEEE Conf. on Decision and Control*, pages 3778–3783, Maui, Hawaii, USA, December 2003.

[14] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.