On Linear Algebraic Techniques for Liveness Analysis of P/T Systems

Laura Recalde, Enrique Teruel, and Manuel Silva *

Dep. Informática e Ingeniería de Sistemas, Centro Politécnico Superior de Ingenieros, Universidad de Zaragoza, Maria de Luna 3, E-50015 Zaragoza, Spain

Abstract. Liveness is a basic property that in many discrete event dynamic systems is considered essential for their correct behavior. It expresses that no action (transition in P/T models) will ever become unattainable. A polynomial time necessary condition for the existence of a live and bounded marking of a P/T net is given. This condition is shown to be also sufficient for some subclasses. The applicability of these results is extended by the use of transformation techniques that allow to exploit them in the analysis of more general nets. Some results for the structural analysis of actual liveness are also overviewed, in particular, sufficient conditions for deadlock-freeness and absence of dead transitions.

1 Introduction

Liveness, the property that no action will ever become unattainable, is necessary for the correct behavior of many reactive systems. Moreover, properties related to *proper termination* of non reactive systems can be rephrased in terms of liveness of a modified system. Thus, liveness is frequently one of the basic requirements in a system, before undertaking the analysis of more complex properties. Together with liveness, *boundedness*, which ensures that the system will not overflow, is also usually required. In this paper we concentrate on the analysis of liveness in bounded systems by means of linear algebraic techniques.

Linear algebraic techniques (and in general structural techniques) intend to extract the maximal amount of information from the structure of the net, avoiding the enumeration of the state space. The results thus obtained are in general computationally much more efficient, although, on the other hand, they often only provide semidecision results (i.e., only necessary or sufficient conditions are obtained).

The paper reviews the application of linear algebraic techniques for liveness analysis, integrating results already published (references are provided), with some new contributions.

A possible strategy to analyze liveness of a given net system through linear algebraic techniques is the following:

* This work was supported in part by Contract CHRX-CT94-0452 (MATCH) within the HCM Programme of the EU.

1. Check if the net structure allows an initial marking that makes the system live. This property is called *structural (str.) liveness*. In case the net is not str. live, we conclude non liveness *for any* initial marking and stop.
2. In other case, check liveness of the net system for *the given* initial marking.

Observe for instance the net systems in Figure 1. The net on the left is not str. live. For any marking, if t_1 is fired "too many times" wrt. t_2 (or vice versa) it will deadlock beacuse t_3 will become unfireable. That is, a problem exists in the structure of the net. On the other hand, the net on the right can be proven to be str. live, although it is not live with the given marking (it would be live if a token were added to p_1 or if the existing token were put in p_3).

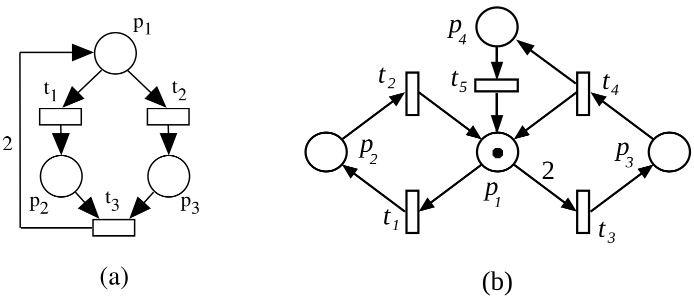


Fig. 1. A non str. live net system (a) and a str. live, but non live net system (b).

For the analysis of str. liveness we propose the use of efficient (i.e., polynomial time complexity) results based on strong connectedness of the net and some properties of its token-flow matrix (e.g., its rank). These results allow to disprove str. liveness in general nets and to prove it in some particular classes. For those situations in which we cannot decide, net *transformation* techniques can be used. If the transformations preserve liveness and/or non liveness, from the analysis of the transformed net we can obtain information about liveness of the original one.

When the net is str. live (or we have not been able to disprove it), we can also try linear algebraic techniques to analyze liveness, avoiding the enumeration of the state space. For general nets, some necessary conditions exist, which are also sufficient in some cases. Among these conditions are absence of dead transitions (every transition is enabled at least once) and deadlock-freeness (at every reachable marking at least one transition is enabled).

In Section 2 some basic definitions of P/T systems and concepts of structure theory are presented, together with the notation to be used. Subsection 3.1 is devoted to study liveness from the point of view of conflicts. It is proved that str. liveness or deadlock-freeness are preserved by the addition of certain arbiters that (partially) regulate some conflicts. This is applied in Subsection 3.2 to deduce the general necessary condition for liveness and boundedness known as *the* *rank theorem.* This condition is also sufficient in the case of *equal conflict* (EQ) systems or *deterministically synchronized sequential processes* (DSSP). These subclasses are studied in Subsection 3.3, which includes, besides the aforementioned particular version of the rank theorem, other results that simplify the analysis of liveness.

Transformation techniques can be used in the analysis of liveness when the previous results cannot decide or are not directly applicable. Besides the removal of certain arbiters, studied in Subsection 3.1, in Section 4 some other transformations are considered, namely the removal of implicit places or bypass transitions, equalization and release.

Dead transitions are studied in Section 5, and a general sufficient condition is obtained for their non existence. Section 6 analyzes the problem of deadlock-freeness, giving a sufficient condition for bounded systems, based on the absence of solutions of a system of inequalities in the integer domain. In Section 7 the application of the previous results is illustrated through an example.

2 Preliminaries and Notation

The reader is assumed to be familiar with basic Petri net concepts (see [14,26] for an introduction). Some preliminary material that is needed and the notation to be used are recalled here. (For the sake of readability, whenever a net or system is defined it "inherits" the definition of all the characteristic sets, functions, parameters,... with names conveniently marked.)

2.1 Nets and net systems

We denote a *P/T net* as $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post} \rangle$, where P and T are the sets of *places* and *transitions*, and $\mathbf{Pre}$ and $\mathbf{Post}$ are the $|P| \times |T|$ sized, natural valued, *incidence matrices.* For instance, $\mathbf{Post}[p,t] = w$ means that there is an *arc* from t to p with *weight* (or multiplicity) w. When all weights are one the net is *ordinary.* For pre- and postsets we use the conventional dot notation, e.g., ${}^{\bullet}t = \{p \in P \mid \mathbf{Pre}[p,t] \neq 0\}$, which can naturally be extended to sets of nodes. If $\mathcal{N}'$ is the *subnet* of $\mathcal{N}$ defined by $P' \subseteq P$ and $T' \subseteq T$, then $\mathbf{Pre}' = \mathbf{Pre}[P', T']$ and $\mathbf{Post}' = \mathbf{Post}[P', T']$. A P- (T-)subnet is defined by a subset of places (transitions) and all their adjacent transitions (places).

A *marking* is a $|P|$ sized, natural valued, vector. A *P/T system* is a pair $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$, where $\mathbf{m_0}$ is the initial marking. A transition t is *enabled* at $\mathbf{m}$ iff $\mathbf{m} \geq \mathbf{Pre}[P,t]$; its firing yields a new marking $\mathbf{m}' = \mathbf{m} + \mathbf{C}[P,t]$, where $\mathbf{C} = \mathbf{Post} - \mathbf{Pre}$ is the *token-flow* matrix of the net. This fact is denoted by $\mathbf{m} \xrightarrow{t} \mathbf{m}'$. An *occurrence sequence* from $\mathbf{m}$ is a sequence of transitions $\sigma = t_1 \cdots t_k \cdots$ such that $\mathbf{m} \xrightarrow{t_1} \mathbf{m}_1 \cdots \mathbf{m}_{k-1} \xrightarrow{t_k} \cdots$. The set of all the occurrence sequences, or *language*, from $\mathbf{m}$ is denoted by $\mathrm{L}(\mathcal{N}, \mathbf{m})$, and the set of all the reachable markings, or *reachability set*, from $\mathbf{m}$, is denoted by $\mathrm{RS}(\mathcal{N}, \mathbf{m})$. The reachability relation is conventionally represented by a *reachability graph* $\mathrm{RG}(\mathcal{N}, \mathbf{m})$ where the nodes are the reachable markings and there is an arc labeled t from node $\mathbf{m}'$ to $\mathbf{m}''$ iff $\mathbf{m}' \xrightarrow{t} \mathbf{m}''$.

A transition is *live* if it can ultimately occur from every reachable marking. A P/T system is *live* when every transition is live and it is *deadlock-free* when any reachable marking enables some transition. A net system is *reversible* when the initial marking (hence every marking) is reachable from any reachable marking. A net system is *bounded* when every place is bounded, i.e., its token content is less than some bound at every reachable marking. Boundedness precludes overflows, liveness ensures that no single action in the system can become unattainable, and reversibility informs on the possibility to return to every state.

2.2 Some concepts from structure theory

Structural techniques intend to obtain as much information for the analysis as possible from the net structure, avoiding the enumeration of the state space. Among these techniques, some are formulated on linear algebraic terms, based on the following observation: Given σ such that $\mathbf{m} \xrightarrow{\sigma} \mathbf{m}'$, and denoting by $\boldsymbol{\sigma}$ the *firing count vector* of σ, then $\mathbf{m}' = \mathbf{m} + \mathbf{C} \cdot \boldsymbol{\sigma}$. This is known as the *state equation* of $\mathcal{S}$. The set of all the markings that fulfill the state equation for a given $\mathbf{m} \in \mathbb{N}^{|P|}$, with $\boldsymbol{\sigma} \in \mathbb{N}^{|T|}$, is called the *linearized reachability set* (wrt. the state equation), LRS($\mathcal{S}$). Its elements (the linearly reachable markings) can be represented as nodes of the *linearized reachability graph* LRG($\mathcal{S}$), constructed like RG($\mathcal{S}$). Clearly, RS($\mathcal{S}$) $\subseteq$ LRS($\mathcal{S}$); the linearly reachable markings that are non reachable will be called *spurious* [6,27]. For instance, for the system in Figure 2, the shaded markings in the LRG are the spurious ones.

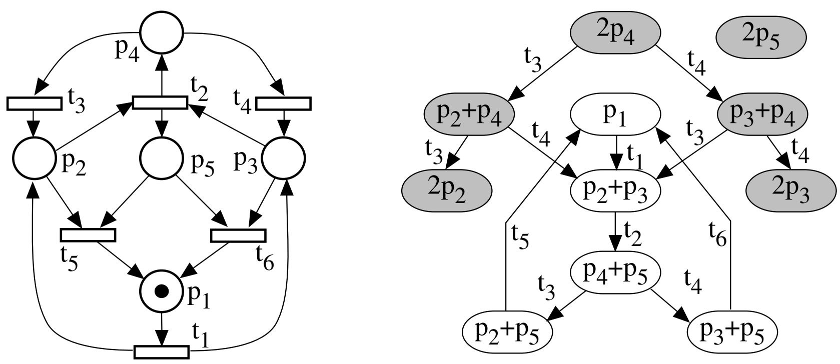


Fig. 2. A live system and its LRG.

The relaxation of the reachability relation using the net state equation provides a convenient way to analyze important properties using linear algebra and convex geometry. This method is specially well suited to deal with *safety* properties, such as marking bounds or mutual exclusions [26,27]. For example, a *str.*

bound for the marking of a place p can be obtained as:

$$\mathbf{sb}[p] = \max\{\mathbf{m}[p] \mid \mathbf{m} - \mathbf{C} \cdot \boldsymbol{\sigma} = \mathbf{m_0} \wedge \mathbf{m}, \boldsymbol{\sigma} \geq \mathbf{0}\} \tag{1}$$

In principle, Equation (1) is an integer programming problem. If integrality constraints (on $\mathbf{m}$ and $\boldsymbol{\sigma}$) are disregarded then (1) is a linear programming problem, that can be solved in polynomial time. The marking that yields the str. bound could be spurious, thus the bound may never be reached. (Nevertheless, for some net subclasses it is always reachable.)

A net $\mathcal{N}$ is *str. bounded* when $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is bounded for *every* $\mathbf{m_0}$. By application of *duality theory* of linear programming to the formulation of the str. bound of a place in (1), a characterization of str. boundedness can be derived (see for instance [27]): $\mathcal{N}$ is str. bounded iff $\mathbf{y} > \mathbf{0}$ exists such that $\mathbf{y} \cdot \mathbf{C} \leq \mathbf{0}$. When $\mathbf{y} \cdot \mathbf{C} = \mathbf{0}$ the net is said to be *conservative*. The dual property of str. boundedness is *str. repetitiveness*: $\mathcal{N}$ is str. repetitive iff $\mathbf{x} > \mathbf{0}$ exists such that $\mathbf{C} \cdot \mathbf{x} \geq \mathbf{0}$. When $\mathbf{C} \cdot \mathbf{x} = \mathbf{0}$ the net is said to be *consistent*. These latter properties inform on the existence of potential infinite behaviors: with a large enough marking, a sequence with firing count vector $\mathbf{x}$ would be fireable, and its occurrence would not decrease the marking, so it could be fired once and again.

Structural boundedness is defined abstracting from the initial marking in the definition of boundedness (i.e., boundedness *for every* initial marking). Other structural properties are defined likewise. For instance, *str. liveness* of a net informs about *existence* of a marking for which the net system is live. When a net is non str. live, *for every* initial marking the net system is non live, so the problem is rooted on the net, not on the marking.

When a net is str. live and str. bounded there exists some marking $\mathbf{m_0}$ such that $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is live, and boundedness is ensured whichever $\mathbf{m_0}$ is taken. In such a case liveness is a matter of the the initial marking, and we need not worry about boundedness. Notice that, in general, str. boundedness is *not* necessary for liveness and boundedness (although it happens to be in some selected subclasses). For instance, the system in Figure 3 is bounded, although not str. bounded (if $\mathbf{m_0} = [0, 4, 0]$ the system is not bounded). However, since str. boundedness is not over restrictive, and it is robust (it will not fail if the marking changes) and much simpler to check, it is often required instead of boundedness.

Annullers of $\mathbf{C}$, such as those appearing in the definition of conservativeness and consistency, play an important role in structure theory. *Flows* (*semiflows*) are integer (natural) annullers of $\mathbf{C}$. Right and left annullers are called T- and P-(semi)flows, respectively. We call a semiflow $\mathbf{v}$ *minimal* when its support, $\|\mathbf{v}\| = \{i \mid \mathbf{v}[i] \neq 0\}$, is not a proper superset of the support of any other, and the greatest common divisor of its elements is one. Flows are important because they induce certain invariant relations which are useful for reasoning on the behavior (e.g., if $\mathbf{y} \geq \mathbf{0}$ and $\mathbf{y} \cdot \mathbf{C} = \mathbf{0}$ then every $\mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ satisfies $\mathbf{y} \cdot \mathbf{m} = \mathbf{y} \cdot \mathbf{m_0}$.)

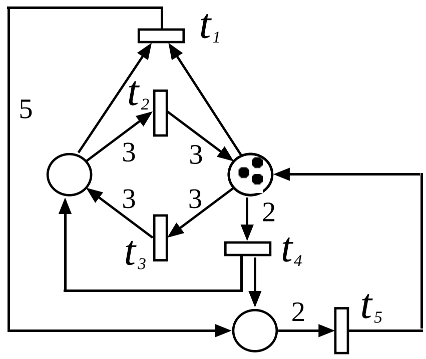


Fig. 3. A live and bounded system which is not str. bounded.

3 The Rank Theorems

Liveness is a property that is difficult to deal with structurally. Nevertheless, a careful investigation of the net structure leads to structural results that are helpful in the analysis of this property. These results prevent us from undertaking a costly liveness analysis when the net is not correct, and may help in the identification of problems in the model.

In this section we will start studying conflicts, mainly from a structural point of view. We will see how liveness can be preserved if a particular kind of conflicts are regulated by means of simple nets. This information about conflicts will be the basis to obtain a polynomial time general necessary condition for liveness and boundedness. Next, we will concentrate on some subclasses, for which better results can be obtained. In particular, we will prove that this necessary condition is a full characterization for them.

3.1 On Conflicts and Arbiters

We consider a *conflict* as the situation where not all that is enabled can occur at once. More formally, $t, t' \in T$ are in (effective) conflict relation at marking $\mathbf{m}$ iff there exist $k, k' \in \mathbb{N}$ such that $\mathbf{m} \geq k \cdot \mathbf{Pre}[P, t]$ and $\mathbf{m} \geq k' \cdot \mathbf{Pre}[P, t']$, but $\mathbf{m} \not\geq k \cdot \mathbf{Pre}[P, t] + k' \cdot \mathbf{Pre}[P, t']$. For this it is necessary that ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$, and in that case we say that t and t' are in str. conflict relation. The str. conflict relation (or *choice*) is an structural prerequisite for the behavioral property of conflict.

The str. conflict relation is not transitive, and we define the *coupled conflict* relation as its transitive closure. Each equivalence class is called a coupled conflict set denoted, for a given t, CCS(t). The set of all the equivalence classes is denoted by SCCS. When $\mathbf{Pre}[P, t] = \mathbf{Pre}[P, t'] \neq \mathbf{0}$, t and t' are in *equal conflict* (EQ) relation, meaning that they are both enabled whenever one is. This is an equivalence relation on the set of transitions and each equivalence class is an equal conflict set denoted, for a given t, EQS(t). An equal conflict set is called trivial if it is formed by just one transition. SEQS is the set of all the equal conflict sets of a given net.

In principle, P/T systems are non deterministic, i.e., there is no rule about which of the enabled transitions in a conflict should fire. However, there are occasions when this indeterminism has to be reduced somehow. For instance, to avoid reaching dangerous states, or to schedule the system under a timed interpretation. A way to reduce the non determinism is to (partially) regulate the conflict by means of a net system, called *arbiter*, some of whose transitions are merged with transitions of the system that has to be regulated (possibly the arbiter does not completely determine the solution of the conflict). That is, the regulated system is defined by the parallel composition of the system and its arbiter(s) [11]. A particular case of arbiters are *local arbiters*, that are applied to EQ sets:

Definition 1. Let e be an EQ set of a net $\mathcal{N}$. A local arbiter for e is a net $\mathcal{A}^e = \langle P^e, T^e, \mathbf{Pre}^e, \mathbf{Post}^e \rangle$ such that:

- $T^e \cap T = e$
- $T^{e\bullet} \cup {}^\bullet T^e = P^e$
- $P^e \cap P = \emptyset$

In general, neither liveness nor non liveness preservation is guaranteed by the addition of local arbiters to a system, as can be seen in Figure 4. The system on the left is live without the arbiter, but t and t' are never enabled when it is present. The opposite holds for the system on the right, which is live with the arbiter, but it is not so when it is removed. We can ensure however deadlock-freeness preservation:

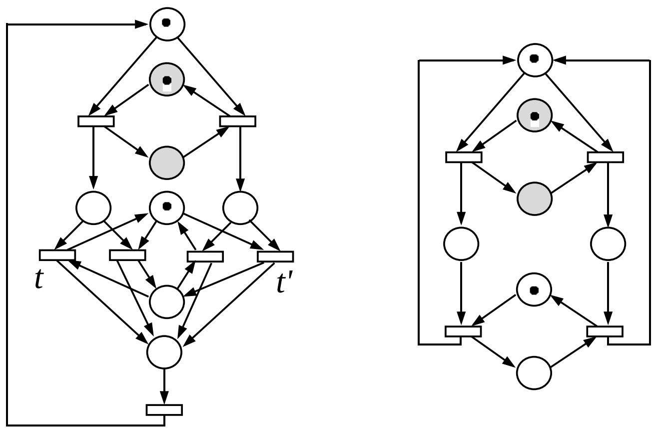


Fig. 4. Neither liveness nor non liveness are preserved in general when arbiters (the P-subnets generated by the shaded places) are added.

Proposition 2. *Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a P/T system, e an EQ set and $\langle \mathcal{A}^e, \mathbf{m_0}^e \rangle$ a local arbiter for e.*

If $\langle \mathcal{N}, \mathbf{m_0} \rangle$ and $\langle \mathcal{A}^e, \mathbf{m_0}^e \rangle$ are deadlock-free, then the system obtained fusing their common transitions, $\langle \mathcal{N}', \mathbf{m_0}' \rangle$, is deadlock-free. Moreover, if $\langle \mathcal{N}, \mathbf{m_0} \rangle$ and $\langle \mathcal{A}^e, \mathbf{m_0}^e \rangle$ do not have spurious deadlocks, neither does $\langle \mathcal{N}', \mathbf{m_0}' \rangle$.

Proof. Clearly, for any sequence that can be fired in the complete system, its projections on $\mathcal{N}$ and $\mathcal{A}^e$ can be fired in the associated systems. Since both of them are deadlock-free, at least a transition is enabled in each system. If one of these transitions is not in e, it can be fired. Otherwise, given that e is an EQ set, any transition enabled in $\mathcal{A}^e$ is enabled in the complete system too.

The proof for spurious deadlocks is analogous, reasoning on linearly reachable markings. □

Better results are obtained if we restrict the kind of arbiters we allow and relax the property by not fixing the marking of the arbiter. Since the purpose of arbiters is to reduce the indeterminism, it is quite common not to allow choices inside the arbiter, i.e., arbiters are frequently *choice free* systems ($\forall p \in P \ |p^\bullet| \leq 1$) [30]. Figure 5 (a) shows a simple system, taken from [16], in which parts are sent from *STORE* 1 to *STORE* 2 and *STORE* 3. The destination is decided by solving the equal conflict $e = \{t_2, t_4\}$. The P-subnet generated by the shaded places (p_4, p_5, p_6, and p_7) is a local choice free arbiter (in fact, an strongly connected marked graph) that imposes some restrictions on the way parts are distributed between the destination stores.

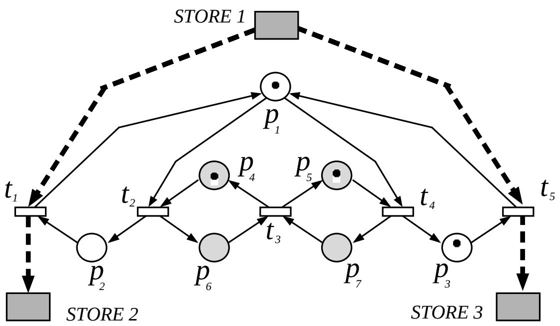


Fig. 5. An EQ system with a choice free local arbiter.

A particular case of choice free local arbiters are *(ordinary) circuit arbiters* (or *regulation circuits*), see Figure 6:

Definition 3. Let $\mathcal{N}$ be a P/T net, and let $e \in$ SEQS such that $|e| > 1$. A net $\mathcal{C}^e = \langle P^e, e, \mathbf{Pre}^e, \mathbf{Post}^e \rangle$ is an (ordinary) circuit arbiter for the equal conflict set e iff $\mathcal{C}^e$ is an ordinary net such that $P^e \cap P = \emptyset$ and its underlying graph is an elementary circuit.

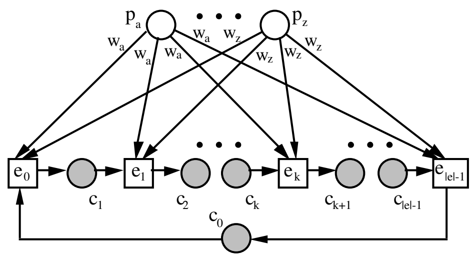


Fig. 6. A circuit arbiter (shaded places) merged on an equal conflict set.

Structural liveness is preserved by the addition of circuit arbiters to non-trivial EQ sets (see for instance [31]). This is one of the key results to obtain the liveness and boundedness necessary condition of Section 3.2. The same proof can be slightly modified to allow using the more general choice free local arbiters instead:

Proposition 4. *Let $\mathcal{N}$ be a P/T net, e an EQ set, $\mathcal{A}^e$ a strongly connected and consistent choice free local arbiter for e, and $\mathcal{N}'$ the net with the arbiter. If $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ is live and bounded, then there exists $\mathbf{m_0}'$ with $\mathbf{m_0}'[P] = \mathbf{m_0}$ such that $\mathcal{S}' = \langle \mathcal{N}', \mathbf{m_0}' \rangle$ is live and bounded (thus $\mathcal{N}'$ is str. live).*

Proof. A strongly connected and consistent choice free net is conservative (i.e., $\mathbf{y} > \mathbf{0}$ exists such that $\mathbf{y} \cdot \mathbf{C} = \mathbf{0}$) [30], and so it is str. bounded. Thus, boundedness of $\mathcal{S}$ implies boundedness of $\mathcal{S}'$ for every $\mathbf{m_0}'$ with $\mathbf{m_0}'[P] = \mathbf{m_0}$. Since $\mathcal{S}$ is live and bounded, the number

$$r_e = \max_{t,\mathbf{m}} \{ \min_{\sigma} \{ \#(e, \sigma) \mid \sigma t \in \mathrm{L}(\mathcal{N}, \mathbf{m}) \} \mid t \in T \wedge \mathbf{m} \in \mathrm{RS}(\mathcal{S}) \}$$

is well-defined for every transition t, where $\#(e, \sigma)$ represents the total number of firings of transitions of e in σ. This is a bound for the number of firings of transitions in e that are *required* to enable an arbitrary t from an arbitrary reachable marking. Now, put enough tokens in every $p \in P^e$ as to (1) enable its output transition at least that number of times, and (2) be able to fire the minimal T-semiflow of the arbiter, what in choice free systems implies its liveness and reversibility [30].

Now we prove that $\mathcal{S}'$ is live with this marking. Let $\mathbf{m}' \in \mathrm{RS}(\mathcal{S}')$ and $t \in T$. We shall prove that t can ultimately be enabled from $\mathbf{m}'$. We claim that there exists a marking $\mathbf{m}'' \in \mathrm{RS}(\mathcal{N}', \mathbf{m}')$ such that $\mathbf{m}''[P^e] = \mathbf{m_0}'[P^e]$. In that case, since (1) $\mathcal{S}$ is live, (2) $\mathbf{m}''[P] \in \mathrm{RS}(\mathcal{S})$, and (3) $\mathbf{m_0}'[P^e]$ has been defined in a way that it does not interfere when firing a sequence to enable an arbitrary t from an arbitrary reachable marking, then we can fire in $\langle \mathcal{N}', \mathbf{m}'' \rangle$ the same sequence that we could fire in $\langle \mathcal{N}, \mathbf{m}''[P] \rangle$ in order to enable t.

To prove the claim, let $\sigma_e = e_{i_1} e_{i_2} \cdots e_{i_k} \in \mathrm{L}(\mathcal{N}^e, \mathbf{m}'[P^e])$ be such that $\mathbf{m}'[P^e] \xrightarrow{\sigma_e} \mathbf{m_0}'[P^e]$, i.e., a sequence in the circuit arbiter returning to the initial marking. It is easy to see that a sequence such that its projection on e is σ_e can be fired in $\langle \mathcal{N}', \mathbf{m}' \rangle$. The idea is firing transitions not in e, which does not affect the marking of places in P^e, until e are P-enabled (their input places in P have enough tokens, no matter how many tokens there are in other places), which will eventually happen thanks to liveness of $\langle \mathcal{N}, \mathbf{m}'[P] \rangle$. Then, firing e_{i_1} which is also P^e-enabled according to our definition of σ_e, then firing more transitions not in e until e are P-enabled again and firing e_{i_2} which is also P^e-enabled, etc. □

The addition of a circuit arbiter increases the rank of the token-flow matrix in the number transitions (or places) in the circuit minus one. The proof shows that one place in the regulation circuit can be obtained by adding the rest of the places in the circuit, and that if it is removed, no other place in the circuit is a linear combination of the rest of the places in the system.

Lemma 5 ([31]). *Let $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ be a P/T system, and e a non-trivial EQ set, i.e., $|e| > 1$. Let $\mathcal{C}^e$ be a circuit arbiter for e, and let $\mathcal{N}'$ be the net $\mathcal{N}$ merged with the circuit arbiter $\mathcal{C}^e$ by sharing the transitions in e. If $\mathcal{S}$ is live and bounded, then* $\text{rank}(\mathbf{C}') = \text{rank}(\mathbf{C}) + |e| - 1$

This result can be extended to choice free local arbiters. The idea is to replace the choice free local arbiter by a weighted circuit with an equivalent token-flow matrix (a weighted circuit with the same T-semiflow will have an equivalent incidence matrix). The rank of the token-flow matrix of the complete system will not change. Neither liveness, nor boundedness will change by the addition of self-loops around the transitions of the circuit that do not belong to the EQ set, what reduces the problem to adding a weighted circuit arbiter to a live system. Observe that the proof of Lemma 5 is also valid, with slight modifications, for weighted circuit arbiters.

Proposition 6. *Let $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ be a P/T system, and e a non-trivial EQ set, i.e., $|e| > 1$. Let $\mathcal{A}^e$ be a strongly connected and consistent choice free local arbiter for e, and let $\mathcal{N}'$ be the net $\mathcal{N}$ merged with the choice free local arbiter by sharing some transitions in e. If $\mathcal{S}$ is live and bounded, then* $\text{rank}(\mathbf{C}') = \text{rank}(\mathbf{C}) + \text{rank}(\mathbf{C}^e) = \text{rank}(\mathbf{C}) + |T^e| - 1$

3.2 The Rank Theorem: a Liveness and Boundedness Necessary Condition

A well-known polynomial time necessary condition for liveness of a bounded net system, based solely on purely structural properties, is strong connectedness [23] and consistency (i.e., $\mathbf{x} > \mathbf{0}$ exists such that $\mathbf{C} \cdot \mathbf{x} = \mathbf{0}$) of the net [13,24]. Although, if str. boundedness is required, conservativeness is also necessary [13,24].

These conditions are very useful to discard models that are not correct before undertaking a more costly analysis. Unfortunately they are only necessary: there are strongly connected and consistent nets that cannot be lively and boundedly marked. We present here an improved — and still polynomial time — necessary condition that incorporates an upper bound for the rank of the token-flow matrix.

This condition was first deduced for free choice systems, and was derived from the problem of computability of visit ratios in stochastic free choice nets [5]. In general, the bound on the rank of the token-flow matrix is obtained through the regulation of each EQ set by means of a circuit arbiter, what increases the rank of the token-flow matrix in a fixed known amount (the number of transitions in the conflict minus one), Lemma 5. By Proposition 4, the regulation of non-trivial EQ sets with circuit arbiters preserves liveness and boundedness, thus this procedure can be applied to every non-trivial EQS. Moreover, since the regulated system is live and bounded, at least a repetitive sequence exists. This is the basis for the bound on the rank of the token-flow matrix in live and bounded systems.

Theorem 7 (The rank theorem). *If $\mathcal{S}$ is a live and bounded P/T system, then $\mathcal{N}$ is strongly connected, consistent, and* $\text{rank}(\mathbf{C}) < |\text{SEQS}|$.

Proof. Only the rank condition needs to be proven. Let $\mathcal{N}'$ be the net $\mathcal{N}$ together with circuit arbiters merged to *every* non-trivial equal conflict set. Applying Lemma 5 repeatedly after each circuit arbiter is merged, what can be done thanks to Proposition 4, and since $|e| = 1$ in trivial EQ sets, it follows that:

$$|T| - 1 \geq \text{rank}(\mathbf{C}') = \text{rank}(\mathbf{C}) + \sum_{e \in \text{SEQS}} (|e| - 1)$$

Rearranging the above inequality we obtain a bound for the rank:

$$\text{rank}(\mathbf{C}) \leq |T| - \sum_{e \in \text{SEQS}} (|e| - 1) - 1$$

Since $\sum_{e \in \text{SEQS}} |e| = |T|$, this bound is $|\text{SEQS}| - 1$, so the result follows. □

According to Theorem 7 the difference $|\text{SEQS}| - \text{rank}(\mathbf{C})$ should be positive. This difference somehow quantifies the interplay between choices (decisions) and direct or indirect synchronizations reflected in the dimension of the linear space of T-(semi)flows. That is, let X be the space of T-flows of a net $\mathcal{N}$ with $|T|$ transitions. If the net is conservative any T-flow can be transformed into a T-semiflow by adding a linear combination of other T-semiflows. Thus, $|\text{SEQS}| - \text{rank}(\mathbf{C}) = |\text{SEQS}| - (|T| - \dim(\text{X})) = \dim(\text{X}) - (|\text{T}| - |\text{SEQS}|)$. If the number of linearly independent T-semiflows ($\dim(\text{X})$) is less than the number of "free decisions" ($|T| - |\text{SEQS}|$) the system cannot be live and bounded.

Intuitively, from the proof of Theorem 7, it becomes apparent that the rank condition fails when some individual "choices between alternatives" are not independent from the rest of the system (synchronizations, other choices, etc.). Apart from "flow problems" (i.e., absence of consistency), absence of such independence indicates that a wrong decision taken in an individual choice may affect the rest of the system (to the point of "killing" it).

If the rank theorem detects a problem in the net, a procedure can be given to locate it: start adding regulation circuits to the non-trivial EQ sets, one at a time, and check if the rank increases in the right amount. When the addition of

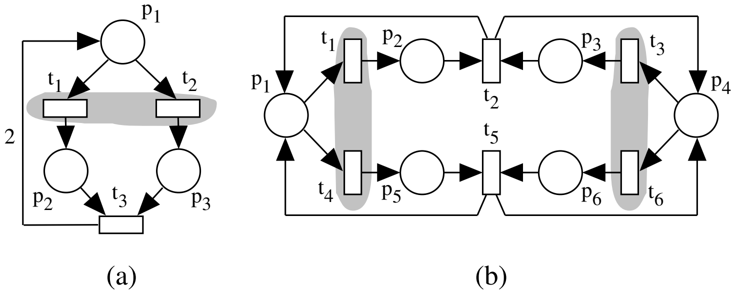


Fig. 7. Two conservative and consistent nets where the rank theorem detects non str. liveness. The (non-trivial) equal conflicts are shaded.

a regulation circuit does not increase the rank as expected, a conflict has been spotted that is (apparently) free, but whose resolution should be conditioned by the resolution of (some) of the conflicts regulated before. This relationship between the conflicts is reflected in the existence of a T-semiflow containing at least two transitions of this last conflict.

Let us illustrate this with the examples of Figure 7. In the net of Figure 7 (a) $\text{rank}(\mathbf{C}) = 2 = |\text{SEQS}|$ and the only minimal T-semiflow is $\mathbf{1}$. Notice that the fact that t_1 and t_2 are together in every T-semiflow — what is due to the synchronization or join transition t_3 — means that in every infinite sequence they should be fired in a fixed proportion (one to one in this case). Nevertheless, since the choice between t_1 and t_2 is free, the net does not prevent the violation of this proportion. *This mismatch between conflicts and synchronizations is what the rank theorem detects.* If we merge a circuit arbiter on t_1 and t_2, say c_0 from t_2 to t_1 and c_1 from t_1 to t_2, the rank is not increased: one place is clearly a linear combination of the other, say $\mathbf{C}[c_0, T] = -\mathbf{C}[c_1, T]$; but also c_1 is a linear combination of other places, namely $\mathbf{C}[c_1, T] = \mathbf{C}[p_1, T] + 2\mathbf{C}[p_2, T]$, what reveals the problem.

In the net of Figure 7 (b) $\text{rank}(\mathbf{C}) = 4 = |\text{SEQS}|$ and two minimal T-semiflows exist: $t_1 + t_2 + t_3$ and $t_4 + t_5 + t_6$. Now the synchronizations (t_2 and t_5) do not impose a given resolution of each conflict to allow infinite activity (the outcomes of each conflict are in different minimal T-semiflows), but they impose that each conflict is solved according to the other, what is again not guaranteed by the net structure where the choices are free. The rank theorem detects also this mismatch. If we merge a circuit arbiter, say on t_1 and t_4, it increases the rank. Now the net with the arbiter has a unique minimal T-semiflow, $\mathbf{1}$, and the second circuit arbiter does not increase the rank (after merging an arbiter on one conflict, a proportion between the outcomes of the other conflict has been fixed).

Unfortunately the condition for the existence of a live and bounded marking given by Theorem 7 is only necessary: there are strongly connected and consistent nets that fulfill the rank condition and cannot be lively and boundedly marked.

For instance, both systems in Figure 8 fulfill it, however the system in Figure 8 (a) is str. live (it is live with this marking) while for the net in Figure 8 (b) there is no marking making it live.

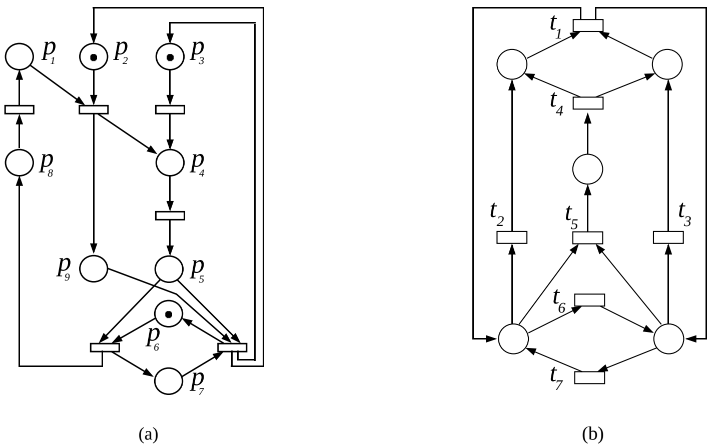


Fig. 8. Two nets for which the necessary condition in Theorem 7 does not allow to decide.

3.3 Rank Theorem and Subclasses

There are some net subclasses for which the rank theorem is also sufficient. Loosely speaking, these subclasses have in common that their syntactical constraints leave only conflicts that are essentially "choices between alternatives" (equal conflicts), what limits the possibility of representing competition. In other words, choices and synchronizations appear, but do not directly interfere.

Definition 8.

1. A P/T net is an *equal conflict* (EQ) net iff for all $t, t' \in T$ such that ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$, $\mathbf{Pre}[P, t] = \mathbf{Pre}[P, t']$ (i.e., SCCS = SEQS). An (extended) free choice net is an ordinary EQ net.
2. A P/T system $\langle P, T, \mathbf{Pre}, \mathbf{Post}, \mathbf{m_0} \rangle$ is a system of *deterministically synchronized sequential processes* (DSSP system, or simply a DSSP) when P is the disjoint union of $P_1, \ldots, P_n$, and B, T is the disjoint union of $T_1, \ldots, T_n$, and the following holds:
   (a) For every $i \in \{1, \ldots, n\}$ let $\mathcal{N}_i = \langle P_i, T_i, \mathbf{Pre}[P_i, T_i], \mathbf{Post}[P_i, T_i] \rangle$. Then, $\langle \mathcal{N}_i, \mathbf{m_0}[P_i] \rangle$ is a live and safe (i.e., strongly connected and 1-bounded) state machine.
   (b) For every $i, j \in \{1, \ldots, n\}$ if $i \neq j$ then $\mathbf{Pre}[P_i, T_j] = \mathbf{Post}[P_i, T_j] = \mathbf{0}$.
   (c) Places in b are called buffers and for every $b \in B$,

- $dest(b) \in \{1, \ldots, n\}$ exists such that $b^\bullet \subseteq T_{dest(b)}$.
- The equal conflict sets of the sequential processes are preserved by the buffers, i.e., if $t, t' \in p^\bullet$, where $p \in P_{dest(b)}$, then $\mathbf{Pre}[b, t] = \mathbf{Pre}[b, t']$.

A *DSSP net* is the net of a DSSP (system). A *DSSP marking* is a marking for a DSSP net that respects the monomarkedness of the state machines.

In EQ nets all conflicts are equal, generalizing (extended) free choice nets by allowing bulk services and arrivals. In DSSP several live and safe state machines cooperate by message passing through destination private buffer places. (Thus DSSP are modularly defined, the communication among modules being asynchronous.)

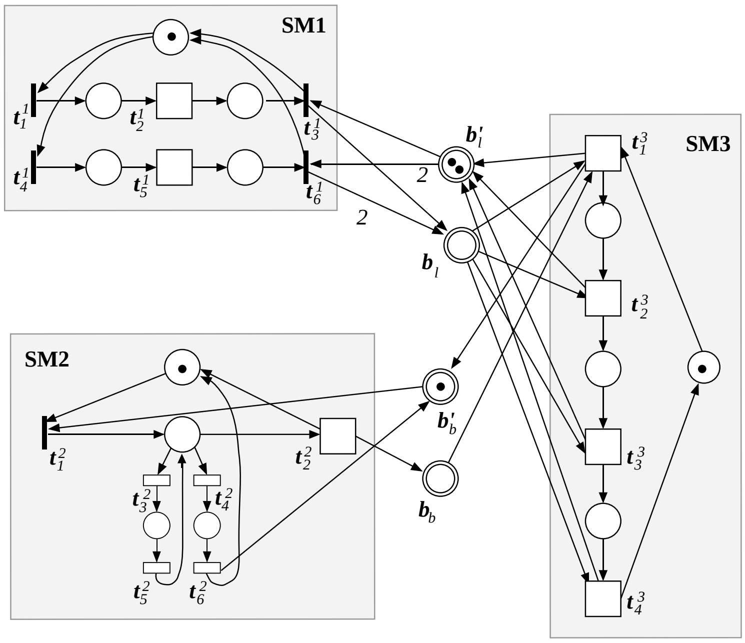


Fig. 9. A DSSP model of a manufacturing line: three state machines and four buffers.

The system in Figure 9 is a DSSP. It models a manufacturing line that makes tables and is composed of three workstations (SM 1, SM 2, and SM 3), that make legs, boards, or assemble a table. Each workstation is modeled by a live and safe state machine, identified by the superscript $i \in \{1, 2, 3\}$ at every node. SM 1 makes legs and can work in two modes, producing one or two legs at a time. SM 2 makes boards and can fail, either scrapping or not the board. SM 1 and SM 2 deposit their products in two respective intermediate buffers (b_l and b_b) from which they are collected by SM 3 to assemble the table. Buffers have a (physically) limited capacity. When a buffer is full its producer is *blocked* and when it is empty its consumer is *starved*. Places b_l' and b_b' represent the empty slots in their respective buffers. In the DSSP representation these places are also called buffers to distinguish them from the places in the SM. For easy identification buffers are signaled by a double circle (although they are normal places).

A very restricted version of DSSP in which buffers were "source private" and multiple arc weights were not allowed, was first proposed and studied in [21], It was generalized in [28] by allowing weights on arcs adjacent to buffers. In Definition 8 the "source private" restriction on the buffers is also removed. With this definition and under *interleaving* semantics, DSSP are a *strict generalization* of equal conflict systems. In other words, provided that only *sequential observations* are relevant (which is the case if we are interested in liveness or boundedness), equal conflict systems can be *simulated* by DSSP. The construction is simple (see Figure 10): add self-loop places marked with one token around each equal

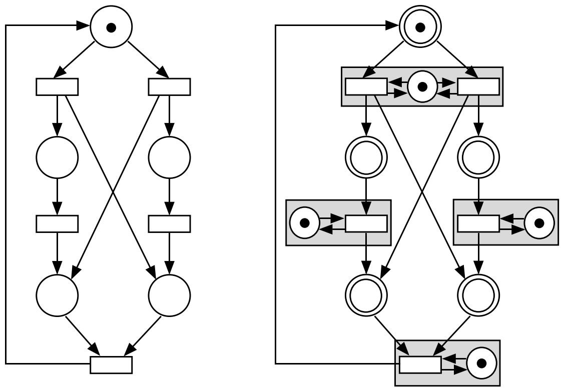


Fig. 10. Simulation of equal conflict systems by DSSP.

conflict set of a given equal conflict system. These self-loop places (with their adjacent transitions) are the sequential agents, and the original places of the equal conflict system play the role of buffers. Anyhow, we keep the distinction between EQ and DSSP all along the paper, since the membership problem for EQ is straightforward, while DSSP models appear after a restricted modular building process. That is, rather than recognizing a given net as being DSSP we know by construction, through a synthesis procedure that distinguishes functional entities and their communication, that a model is a DSSP.

An important result in DSSP and EQ systems, that greatly simplifies the analysis of liveness, is the equivalence of liveness and deadlock-freeness if the system is bounded and strongly connected. Moreover, the LRG is directed in live EQ systems and in live and consistent DSSP. This means that live EQ systems and live and bounded DSSP cannot have spurious deadlocks and, in particular, that it is possible to verify liveness in these subclasses by checking absence of solution of a linear system in the integer domain, as we will see in Section 6. Another useful result is that liveness is monotonic wrt. the marking in EQ systems, and wrt. the marking of the buffers in DSSP. These important properties are summarized in the following result:

Theorem 9 ([31,17]).

1. *A bounded, strongly connected DSSP (thus, also an EQ system) is live iff it is deadlock-free.*
2. *The* LRG *of a live EQ system or a live and consistent DSSP is directed, i.e., any pair of linearly reachable markings have a common successor. Thus, spurious deadlocks do not exist.*
3. *Liveness is monotonic wrt. the marking in EQ systems, and wrt. the marking of the buffers in DSSP.*

In general, a live system can be bounded without being str. bounded (see Figure 3). However, this cannot happen in live EQ systems or DSSP. This can be immediately deduced from the following lemma and the general liveness and boundedness necessary condition of Theorem 7. Lemma 10 for EQ is proved in [31], while its generalization to DSSP is contained in [20].

Lemma 10 ([20,31]). *Let $\mathcal{N}$ be a strongly connected DSSP (EQ) net and let* $\text{rank}(\mathbf{C}) < |\text{SEQS}|$*. Then it is consistent iff it is conservative.*

Corollary 11. *A live DSSP (EQ system) is bounded iff it is str. bounded.*

For DSSP, and hence also for EQ nets, the necessary condition of Theorem 7 for the existence of a live and bounded marking is also sufficient, what makes it a polynomial time structural characterization. This sufficient condition is based on the following result, which shows that, assuming the system has "good structural properties", a marking of the buffers exists that makes the system live. That is, problems due to wrong connections among the SM are reflected in the structural properties of the net. (A proof of the sufficient condition specific for EQ nets can be found in [31].)

Lemma 12 ([18,19]). *Let $\mathcal{N}$ be a strongly connected and conservative DSSP net with* $\text{rank}(\mathbf{C}) < |\text{SEQS}|$*. For every $\mathcal{N}_i$ let $\mathbf{m_0}_i$ be such that $\langle \mathcal{N}_i, \mathbf{m_0}_i \rangle$ is live (i.e., $\mathbf{m_0}_i \cdot \mathbf{1} \geq 1$). Then $\mathbf{m_0}$ exists such that $\mathbf{m_0}[P_i] = \mathbf{m_0}_i$ for every $1 \leq i \leq n$ and the system $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is live.*

Using some of the previous results, in [18] it is proven that a DSSP net is str. live and str. bounded iff it is consistent, conservative and $\text{rank}(C) = |\text{SEQS}| - 1$. A slightly stronger result is proven now:

Theorem 13. *Let $\mathcal{N}$ be a DSSP (EQ) net. A marking $\mathbf{m_0}$ such that $\mathcal{S}$ is live and bounded exists iff $\mathcal{N}$ is strongly connected, consistent, and* $\text{rank}(\mathbf{C}) = |\text{SEQS}| - 1$.

Proof. Applying the general necessary condition of Theorem 7, if $\langle N, \mathbf{m_0} \rangle$ is live and bounded, then it is consistent and $\text{rank}(\mathbf{C}) \leq |\text{SEQS}| - 1$. Moreover, by Lemma 10, it is conservative. The equality of $\text{rank}(\mathbf{C})$ and $|\text{SEQS}| - 1$ can be obtained observing that after adding the regulation circuits, as in the proof of Theorem 7, this net cannot have more than one minimal T-semiflow. (Being conservative, any T-flow is a linear combination of minimal T-semiflows [27].) Let $\mathbf{x}$ be a T-semiflow of the regulated net, and let $t \in ||\mathbf{x}||$. For any transition in $||\mathbf{x}||$, every output place must have at least one output transition in $||\mathbf{x}||$. Thus, all the transitions in $\text{EQS}(t)$ are also in $||\mathbf{x}||$ (because the places in the circuit arbiter have only one output transition), and all their output places in the SM have an output transition in $||\mathbf{x}||$. Applying repeatedly this argument, by strong connectedness of the SM all its transitions are in $||\mathbf{x}||$. Moreover, each output buffer of this SM has at least one output transition in $||\mathbf{x}||$. Since $\mathcal{N}$ is strongly connected and the buffers are "output private", repeating this argument with every SM, $||\mathbf{x}|| = T$. There cannot be two minimal T-semiflows with the same support, hence $\text{rank}(\mathbf{C}) = |\text{SEQS}| - 1$.

Using that any strongly connected and consistent DSSP with $\text{rank}(\mathbf{C}) < |\text{SEQS}|$ is conservative, the sufficient condition is immediately deduced from Lemma 12. □

The system in Figure 9 is strongly connected, consistent, and $\text{rank}(\mathbf{C}) = 15 = |\text{SEQS}| - 1$, therefore it can be lively and boundedly marked (for instance it is live and bounded with the marking shown). According to Theorem 9.3, what is needed is to put a large enough initial marking in the buffers.

The idea of taking some modules and communicating them by means of buffers in a restricted way is also the basis of $\{\text{SC}\}^*\text{ECS}$[19,20]. This class is defined by extending recursively the definition of DSSP: an $\{\text{SC}\}^*\text{ECS}$ is composed of a set of $\{\text{SC}\}^*\text{ECS}$ modules that communicate through a set of destination private buffer places, with live and safe SM as the simplest modules. Results analogous to Lemma 12, and Lemma 10 can also be proven for this class as they are for DSSP, with the only difference that the proofs here are done by induction on the number of "levels" used to build the net. Hence, the same characterization for the existence of a live and bounded marking holds for this class, too [19,20].

4 Transformation Techniques for the Analysis of Liveness

A straightforward application of the rank theorems provides no information about those systems that fulfill this condition, unless they belong to one of the subclasses for which it is sufficient for str. liveness (EQ systems, DSSP and $\{\text{SC}\}^*\text{ECS}$ for the moment). For instance, we cannot say whether the systems in Figure 8 are live or not.

Transformation and decomposition techniques have been traditionally used in analysis (see [2,3,25]). The aim has been usually to transform the system into another one as small as possible, that will be analyzed by enumeration. The previous results suggest that a different approach can be used, i.e., a complete reduction is not necessary if a decision can be taken in an intermediate step by means of structural methods. That is, in some cases, after applying some transformations, we can decide that the original system was not live by observing that the transformed system does not fulfill the rank theorem, or decide that it is live if we obtain a system that belongs to a particular subclass to with special results can be applied.

Most of the classical transformation techniques do not improve the applicability of the rank theorems. They neither affect to the difference between the number of EQ sets and the rank of the token-flow matrix (i.e., $|\text{SEQS}|-\text{rank}(\mathbf{C})$), nor do they transform the net into another one belonging to one of the subclasses for which particular results are known. However, they can allow in some cases to apply other more "effective" transformations. Not intending to be exhaustive, in this section we present some of these transformations, studying whether they preserve liveness, non liveness or both, and their effect on $|\text{SEQS}| - \text{rank}(\mathbf{C})$.

Besides transformations, also decompositions can be used in the analysis of a system. In particular, by applying results concerning arbiters, like Proposition 2, Proposition 4 and Proposition 6. Consider for instance the system in Figure 5. It can be seen as an EQ subsystem (in fact an SM with two tokens) and a choice free local arbiter. Applying Proposition 2 and Proposition 4, liveness of these subsystems allow to deduce that the complete system is deadlock-free and str. live.

4.1 Implicit Places and Bypass Transitions

In this section we will see how the removal of implicit places and its dual counterpart, bypass transitions, can help in the analysis of liveness.

Implicit Places A place in a net system is a constraint to the firing of its output transitions. If the removal of a place does not change the behavior of the original net system, it represents a redundancy in the system wrt. transition enabling and it can be removed. A place whose removal preserves the fireable sequences (i.e., the interleaving semantics) of the system is called a *(sequential) implicit place* [6].

Definition 14. Let $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ be a P/T system and $\mathcal{S}' = \langle \mathcal{N}', \mathbf{m_0}' \rangle$ the P/T system resulting from removing place p from $\mathcal{S}$. Place p is a *(sequential) implicit place* iff $\text{L}(\mathcal{N}', \mathbf{m_0}') = \text{L}(\mathcal{N}, \mathbf{m_0})$. If, additionally, $\text{RS}(\mathcal{N}', \mathbf{m_0}') = \text{RS}(\mathcal{N}, \mathbf{m_0})$ (the projection on the places of $\mathcal{N}'$), then $\mathbf{m}[p]$ is a marking redundancy and p is said to be a *marking-redundant implicit place*.

It follows from the definition that the elimination of an implicit place preserves deadlock-freeness, liveness, or marking mutual exclusion; if it is marking-redundant, then it additionally preserves boundedness or reversibility.

By means of linear relaxations, a sufficient structural condition for a place to be implicit can be stated. This condition is based on the solution of a linear programming problem, so it can be verified in polynomial time:

Proposition 15 ([6]). *Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a P/T system. A place $p \in P$ with initial marking $\mathbf{m_0}[p] \geq z$, where z is the optimal value of (2), is implicit:*

$$\begin{aligned} z = \ & min. \ \mathbf{y} \cdot \mathbf{m_0} + \mu \\ & s.t. \ \ \mathbf{y} \cdot \mathbf{C} \leq \mathbf{C}[p, T] \\ & \qquad \mathbf{y} \cdot \mathbf{Pre}[P, t] + \mu \geq \mathbf{Pre}[p, t] \quad \forall t \in p^\bullet \\ & \qquad \mathbf{y} \geq \mathbf{0}, \mathbf{y}[p] = 0 \end{aligned} \tag{2}$$

Problem (2) is feasible iff $\mathbf{y} \geq \mathbf{0}$ exists verifying $\mathbf{y}[p] = 0$ and $\mathbf{y} \cdot \mathbf{C} \leq \mathbf{C}[p, T]$, because the other inequalities can be made true by adequately choosing μ. Choosing $\mathbf{m_0}[p]$ big enough, place p becomes implicit, so we will say that a place p such that $\mathbf{y} \geq \mathbf{0}$, $\mathbf{y}[p] = 0$, and $\mathbf{y} \cdot \mathbf{C} \leq \mathbf{C}[p, T]$ is *str. implicit*. In order to preserve str. boundedness, we concentrate on *marking* str. implicit places:

Definition 16. Let $\mathcal{N}$ be a P/T net. A place $p \in P$ is a *marking str. implicit place* (MSIP) iff there exists $\mathbf{y} \geq \mathbf{0}$, $\mathbf{y}[p] = 0$ such that $\mathbf{y} \cdot \mathbf{C} = \mathbf{C}[p, T]$.

Notice that an MSIP is not necessarily implicit. For instance, p_6 is an MSIP in the system in Figure 11 (the sum of p_2 and p_{345}), but clearly it is not implicit. However, any MSIP is implicit if marked with enough tokens. For instance, two tokens in p_6 make it implicit. By the way, increasing the initial marking of p_6 kills the system, i.e., liveness is not monotonic wrt. $\mathbf{m_0}$.

Observe that for any (linearly) reachable state the marking of an MSIP can be obtained by adding to its initial marking a linear combination of the marking of the rest of the net. From this it is not difficult to see that str. boundedness is not affected by the addition or removal of MSIP. Structural liveness is also preserved if an MSIP is added, since for any marking of the rest of the system, a marking of this place exists that makes it implicit (Proposition 15).

Proposition 17. *Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a P/T system, and $p \in P$ an MSIP.*

1. *$\mathcal{N} \setminus \{p\}$ is str. bounded iff $\mathcal{N}$ is str. bounded.*
2. *If $\mathcal{N} \setminus \{p\}$ is str. live, then $\mathcal{N}$ is str. live.*
3. *If $\mathbf{m_0}[p] \geq z$, where z is the optimal value of (2), then $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is live iff $\langle \mathcal{N} \setminus \{p\}, \mathbf{m_0}[P \setminus \{p\}] \rangle$ is live.*

On the other hand, the removal of an MSIP does not modify the rank of the token-flow matrix, although it may affect to the EQ sets.

Take for instance the system in Figure 11. It is obtained from the system in Figure 8 (a), by fusion of series of places, thus any result about liveness of this system will hold for the system in Figure 8 (a), too. It is not an EQ, neither a DSSP. However removing p_{345}, which is an MSIP ($\mathbf{C}[p_{345}, T] = \mathbf{C}[p_6, T] + \mathbf{C}[p_9, T]$), yields an str. live and str. bounded EQ system (actually, a marked graph). Thus, the system in Figure 11 is str. live and str. bounded and so is the system in Figure 8 (a). That is, the use of implicit places and classical reduction techniques allows to deduce str. liveness of a system for which the rank theorems alone could not decide. Moreover, since the marking of p_{345} is 1, the same as the

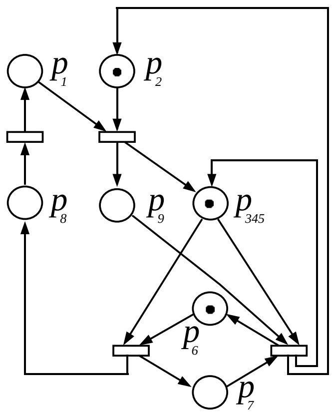


Fig. 11. This system can be obtained applying fusion of series of places to Figure 8 (a). It is an str. live and str. bounded EQ system plus an MSIP, p_{345}. Therefore, by Prop. 17 the system in Figure 8 (a) is str. live and str. bounded. Even more, since with one token p_{345} is implicit, the original system is live.

solution of (2), p_{345} is implicit. Therefore, liveness of the system without p_{345} (see Section 6) implies liveness of the system in Figure 11, and hence liveness of the system in Figure 8 (a).

On the contrary, notice that removing an MSIP (but not implicit) may well destroy str. liveness: The net in Figure 11 is str. live (it is live with the given marking) and $\mathcal{N} \setminus \{p_6\}$ is not str. live. In those classes for which liveness is monotonic wrt. the marking, such as EQ or DSSP (Theorem 9.3), since there always exists a live marking that makes the MSIP truly implicit, removing an MSIP preserves str. liveness.

Bypass Transitions *Duality reasonings* in net theory are essentially based on interchanging places and transitions, and possibly the orientation of arcs. This way, any structural object has a dual counterpart. For instance, P-semiflows are "dual objects" of T-semiflows, and the "dual property" of str. boundedness is str. repetitiveness. It must be noticed that, when dealing with not so algebraic objects or properties, duality reasonings should be applied with caution.

As it is done for instance in [8] (in the context of deriving dual top-down synthesis results for live and bounded free choice), marking str. bypass transitions (MSBT) can be defined applying duality to MSIP:

Definition 18. A transition $t \in T$ is a *marking str. bypass transition* (MSBT) iff there exists $\mathbf{x} \geq \mathbf{0}$, $\mathbf{x}[t] = 0$ such that $\mathbf{C} \cdot \mathbf{x} = \mathbf{C}[P, t]$.

Taking into account the state equation, the effect of an MSBT on the marking is the same as that of the (weighted sum of the) occurrences of the transitions indicated by the corresponding $\mathbf{x}$, occurrences that may not correspond to any fireable sequence. In other words, *the removal of an MSBT preserves the linearly reachable markings*, thus properties such as str. boundedness, conservativeness, etc. Nevertheless, some reachable markings may become spurious when an MSBT is removed. For instance, removing the black transition in Figure 12 (b), (which is an MSBT) makes $2p_2$ spurious. Moreover, as happened with MSIP, removal of MSBT do not modify the rank of the token-flow matrix, although it may affect to the EQ sets.

The same as their dual objects, marking-redundant implicit places, *bypass transitions* are those that preserve the reachable markings and their connectivity (note that the preservation of the connectivity in the case of implicit places follows from the preservation of the reachable markings, so it does not appear explicitly in their definition):

Definition 19. Let $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ be a P/T system and $\mathcal{S}' = \langle \mathcal{N}', \mathbf{m_0}' \rangle$ the P/T system resulting from removing transition t from $\mathcal{S}$. Transition t is a *bypass transition* iff $\mathrm{RS}(\mathcal{N}, \mathbf{m_0}) = \mathrm{RS}(\mathcal{N}', \mathbf{m_0}')$ and for every $\mathbf{m}_1, \mathbf{m}_2 \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$ such that there is a path from $\mathbf{m}_1$ to $\mathbf{m}_2$ in $\mathrm{RG}(\mathcal{N}, \mathbf{m_0})$ there is also a path in $\mathrm{RG}(\mathcal{N}', \mathbf{m_0}')$.

It follows from the definition that the elimination of a bypass transition preserves deadlock-freeness, liveness, marking mutual exclusion, boundedness, reversibility, etc.

Clearly, a transition that is a bypass must be an MSBT, although the converse is not true. For instance, transition t_5 in Figure 12 (a) is an MSBT ($t_5 = t_4 + 2\,t_1$), but not a bypass: Although the reachability set is the same, from a marking that puts the two tokens in the input place of t_5 we cannot reach any other marking if t_5 is removed (i.e., t_5 was not only a "short cut" of firing t_4 and twice t_1).

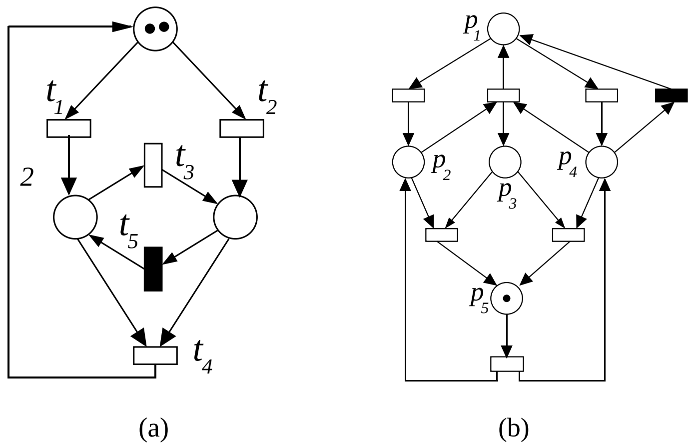


Fig. 12. The removing of MSBT (the black transitions) preserves neither str. liveness (a) nor non str. liveness (b).

The fact that MSBT are not necessarily (behavioral) bypasses makes difficult, in general, to (structurally) prove properties by addition/removal of MSBT. For instance, the live system in Figure 12 (a) becomes non str. live if we remove the black MSBT transition. In the other way, the net in Figure 12 (b) is (str.) live without the black MSBT transition, while not being so if we add it (the, now reachable, marking $2p_2$ is a deadlock).

In the following proposition we give a *simple sufficient condition for an MSBT to be a (behavioral) bypass*, since with the imposed conditions whenever t is fireable $\mathbf{x}$ is fireable in one step. Taking into account the properties of MSBT and bypass transitions, the rest of the statement follows:

Proposition 20. *Let* $\langle \mathcal{N}, \mathbf{m_0} \rangle$ *be a P/T system and* $t \in T$ *an MSBT, with* $\mathbf{x} \geq \mathbf{0}$, $\mathbf{x}[t] = 0$, *and* $\mathbf{C} \cdot \mathbf{x} = \mathbf{C}[P, t]$. *If* ${}^\bullet\|\mathbf{x}\| \cap \|\mathbf{x}\|^\bullet = \emptyset$, *then* t *is a bypass. Therefore:*

1. $\mathcal{N} \setminus \{t\}$ *is str. bounded iff* $\mathcal{N}$ *is str. bounded.*
2. *If* $\langle \mathcal{N}, \mathbf{m_0} \rangle$ *is live, then* $\langle \mathcal{N} \setminus \{t\}, \mathbf{m_0} \rangle$ *is live.*
3. *If* $\langle \mathcal{N} \setminus \{t\}, \mathbf{m_0} \rangle$ *is live, then* $\langle \mathcal{N}, \mathbf{m_0} \rangle$ *is deadlock-free and all the transitions in* $T \setminus \{t\}$ *are live.*

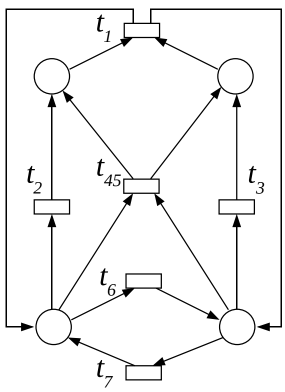


Fig. 13. An EQ system with a bypass transition, t_{45}.

Take the net $\mathcal{N}$ in Figure 13. It is obtained from the net in Figure 8 (b), by fusion of t_4, p_3 and t_5. Transition t_{45} is a bypass ($\mathbf{C}[P, t_{45}] = \mathbf{C}[P, t_2] + \mathbf{C}[P, t_3]$, and ${}^\bullet\{t_2, t_3\} \cap \{t_2, t_3\}^\bullet = \emptyset$) and $\mathcal{N}' = \mathcal{N} \setminus \{t_{45}\}$ is a consistent and conservative EQ net with $|\mathrm{SEQS}'| = \mathrm{rank}(\mathbf{C}')$. Therefore $\mathcal{N}'$ it is not str. live, and from Proposition 20, neither is $\mathcal{N}$, nor the net in Figure 8 (b).

In order to get useful conclusions via MSBT, instead of restricting them, we can restrict the nets to analyze. Taking into account that the LRS does not change with the addition of an MSBT, and no arc is deleted from LRG, the following holds:

Proposition 21. *Let $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ be a P/T system and $B \subset T$ a set of MSBT. If $\mathcal{S}' = \langle \mathcal{N} \setminus \{B\}, \mathbf{m_0} \rangle$ is a system without linearly reachable deadlocks (spurious or not), then $\mathcal{S}$ does not have linearly reachable deadlocks either.*

This result is useful if, after the removal of MSBT, we get into a class like EQ or DSSP. For instance, if some system is a live and bounded EQ or DSSP plus some MSBT, then it is deadlock-free (live, except perhaps the MSBT).

4.2 Equalization and Release

In this section we investigate two transformations that, under some conditions, preserve non (str.) liveness. This allows to obtain sufficient conditions out from the ones we know for EQ systems and DSSP, i.e., these will be the target classes we intend to transform the system into.

The transformations, *equalization* and *release*, while having a similar purpose, approach the problem differently: in equalization the enabling conditions of the transitions in a conflict are *hardened* to make it equal, while in release they are *weakened*. In both cases, the basic idea is, taking a place as reference, to decouple synchronization and conflict (of a coupled conflict set not being an equal conflict set): equalization asks for (the maximum) synchronization constraints before solving the conflict, while release solves the conflict a priori, disregarding the synchronization constraints. Both transformations may kill a live system (even structurally), but in case they do not, that is, if we can prove that the transformed net system is (str.) live, then we can deduce (str.) liveness of the original one.

Equalization Equalization is inspired in the *efc-representation* of [7], where it was used to derive liveness and boundedness of *regular marked Petri nets*, a class of ordinary nets containing live and bounded free choice. We define equalization as a local transformation that makes equal a str. conflict originated by a common input place by increasing the preconditions of the transitions to make them equal, while preserving the token-flow matrix. This definition is intended to be suitable not only to transform a net into EQ, but also for DSSP (the place is in that case selected to be the one that defines a choice from the SM point of view).

Being more precise: Let $\mathcal{N}$ be a P/T net and $p_0 \in P$ such that $p_0{}^\bullet$ is not an EQ set. The equalization of $\mathcal{N}$ wrt. p_0 defines a new net, $\mathcal{N}'$, with the same set of places and transitions and where the arc weights are such that:

- $\mathbf{Pre}'[p,t] = w_p$ and $\mathbf{Post}'[p,t] = \mathbf{Post}[p,t] - \mathbf{Pre}[p,t] + w_p$ if $t \in p^\bullet \cap p_0{}^\bullet$, where $w_p = \max_{t \in p_0{}^\bullet}\{\mathbf{Pre}[p,t]\}$
- $\mathbf{Pre}'[p,t] = \mathbf{Pre}[p,t]$ and $\mathbf{Post}'[p,t] = \mathbf{Post}[p,t]$ otherwise.

Consider the net system in Figure 14 (a). It models two agents, a reader and a writer that operate on a common database. The reader and the writer cannot simultaneously access it, and a monitor is in charge of accepting just a request at a time. The system is not an EQ, neither a DSSP, but it can be easily

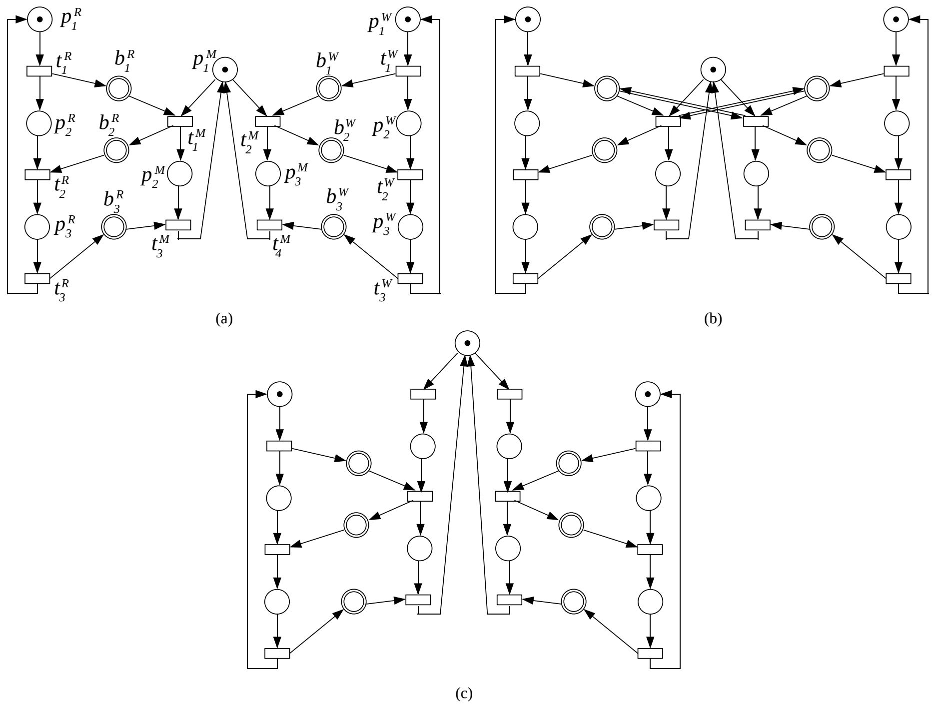


Fig. 14. A writer–reader access system to a common database.

transformed into a DSSP (and also an EQ in this case) by adding arcs from b_1^R to t_2^M and from b_1^W to t_1^M (Figure 14 (b)). In terms of our interpretation of the model, the monitor waits till both, the reader and the writer, ask permission to access the database before granting it to any.

Notice that equalization does not change the token-flow matrix, and thus structural properties of the net based on this matrix (consistency, conservativeness,...) are preserved. Some states can be made unreachable ($p_3^R p_2^M p_1^W$ for instance in the example in Figure 14), but provided that this does not kill the system and that any of its linearly reachable states has an effectively reachable successor, the original system will be live too.

Proposition 22. *Let $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ be a P/T system and $\mathcal{S}' = \langle \mathcal{N}', \mathbf{m_0} \rangle$ the system obtained equalizing $\mathcal{N}$ with respect to one or several places.*

1. *$\mathcal{N}'$ is str. bounded iff $\mathcal{N}$ is str. bounded.*
2. *If $\mathcal{S}'$ is a live system such that any linearly reachable marking has an effectively reachable successor, that is, for any marking $\mathbf{m} \in \mathrm{LRS}$, $\mathrm{RS}(\mathcal{N}', \mathbf{m}) \cap \mathrm{RS}(\mathcal{N}', \mathbf{m_0}) \neq \emptyset$, then $\mathcal{S}$ is also live.*

Proof. The first item comes directly from preservation of the token-flow matrix.

For the second, assume $\mathcal{S}$ non live. Then $t \in T$ and $\mathbf{m}_t \in \mathrm{RS}(\mathcal{S})$ exist such that t cannot be fired from any $\mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m}_t)$. Clearly, $\mathbf{m}_t \in \mathrm{LRS}(\mathcal{S}')$, thus applying the hypothesis, a marking $\mathbf{m}_1$ exists such that $\mathbf{m}_1 \in \mathrm{RS}(\mathcal{N}', \mathbf{m}_t) \cap \mathrm{RS}(\mathcal{N}', \mathbf{m_0})$. Since $\langle \mathcal{N}', \mathbf{m_0} \rangle$ is live, $\mathbf{m}_2 \in \mathrm{RS}(\mathcal{N}', \mathbf{m}_1)$ exists such that t is enabled. Contradiction, since $\mathbf{m}_2 \in \mathrm{RS}(\mathcal{N}, \mathbf{m}_t)$. □

The example in Figure 15 shows the interest of the linear reachability restriction for $\mathcal{S}'$. In the example, $\mathcal{S}$ (Figure 15 (a)) deadlocks (see its RG in Figure 15 (b)), although if we equalize wrt. p_1 (see $\mathcal{S}'$ in Figure 15 (c)) $\mathcal{S}'$ is live: its RG is the one in Figure 15 (b) without $p_2\, p_5$, which is now spurious (so Proposition 22 cannot be applied). Fortunately, this linear reachability restric-

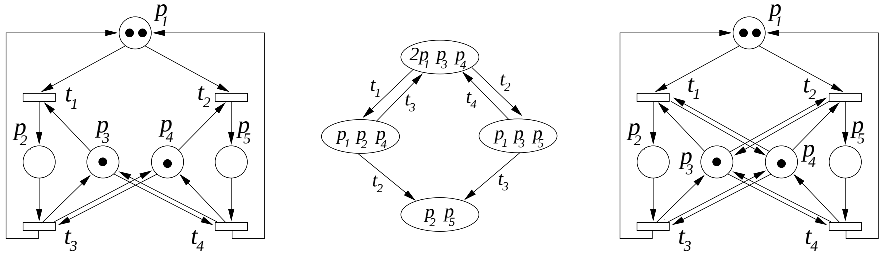


Fig. 15. A non live system whose equalization wrt. p_1 is live.

tion holds, for instance, if the LRG is directed, as it happens in live EQ systems or live and consistent DSSP (Theorem 9.2). In particular, if $\mathcal{N}'$ is an str. live and str. bounded EQ or DSSP, then $\mathcal{N}$ is str. live and str. bounded too.

Using EQ as target class, i.e., converting each coupled conflict set into an equal conflict set via equalization, allows to deduce the next general sufficient condition out from the str. liveness and str. boundedness characterization of EQ nets. It generalizes a result for ordinary nets in which free choice was used as the target class [7]. Moreover, if every minimal P-semiflow of the equalized system is live, the EQ system is live [31], and hence the original system is live.

Proposition 23 (A General Sufficient Condition [18]). *Let $\mathcal{S}$ be a P/T system. If $\mathcal{N}$ is strongly connected, consistent and* $\mathrm{rank}(\mathbf{C}) = |\mathrm{SCCS}| - 1$, *then it is str. live and str. bounded.*

Release Release is inspired in [10], where the idea was also to transform a given net to obtain a free choice net. We define release as a local transformation that makes equal a str. conflict originated by a common input place by weakening the conditions that enable the firing of transitions in a conflict, so that the conflict resolution is completely free and the synchronizations are done afterwards. Again, this transformation can lead, for instance, to an EQ or DSSP, although in a different way than equalization. As an example, in the system of Figure 14, the monitor decides in advance to whom the access to the database will be granted, no matter who is asking for it (Figure 14 (c)).

Being more precise: Let $\mathcal{N}$ be a P/T system and $p_0 \in P$ such that $p_0{}^\bullet$ is not an EQ set. We define the transformed net in two steps (Figure 16):

1. For each non-trivial EQ set in $p_0{}^\bullet$ with more than one input place, e, delete the arcs connecting the transitions and their input places, add one transition, t_e, one place, p_e, and arcs so that
   - $\mathbf{Pre}'[p, t_e] = \mathbf{Pre}[p, t]$ with $t \in e$,
   - $\mathbf{Pre}'[p_e, t] = 1$ if $t \in e$, 0 otherwise,
   - $\mathbf{Post}'[p_e, t_e] = 1$.

   In words, transform such EQ set into a synchronization plus a free choice afterwards.
2. Now every EQ set in $p_0{}^\bullet$ is trivial or has p_0 as unique input place. Let $out = \{t_1, \ldots, t_k\}$ be the transitions in $p_0{}^\bullet$ with more than one input place or with $\mathbf{Pre}[p_0, t] \neq 1$. Remove the arcs that connect p_0 to the transitions in out and add transitions $\{t'_1, \ldots, t'_k\}$, places $\{p_1, \ldots, p_k\}$, and arcs so that
   - $\mathbf{Pre}''[p_0, t'_k] = 1$,
   - $\mathbf{Pre}''[p_k, t_k] = \mathbf{Pre}'[p_0, t_k]$
   - $\mathbf{Post}''[p_k, t'_k] = 1$.

   This way, all the outputs of p_0 are equally possible and the synchronizations are done afterwards.

The marking of the new system is defined as the marking of the initial system and zero tokens in the new places.

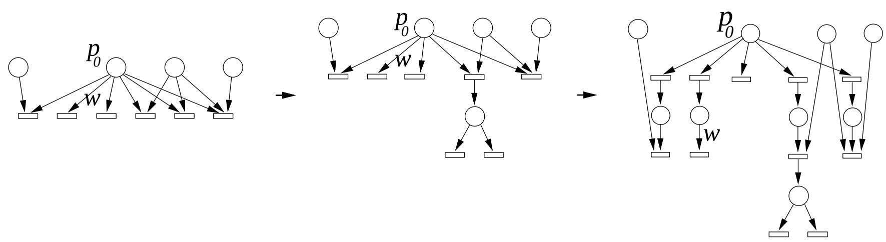


Fig. 16. Release of the transitions in the conflict wrt. p_0.

The first step in the release transformation is done in order to avoid killing the system in an absurd way. Imagine that we do not apply the first step and directly use the second step to "release" the system in Figure 17 (b) wrt. p and p', obtaining the system in Figure 17 (c). Firing t and t', the system deadlocks. The system would not deadlock if the EQ set had been grouped before, as in Figure 17 (a).

It is clear that even then, a live system can be killed if the conflicts are released, but when the released system is live, the original one is live too:

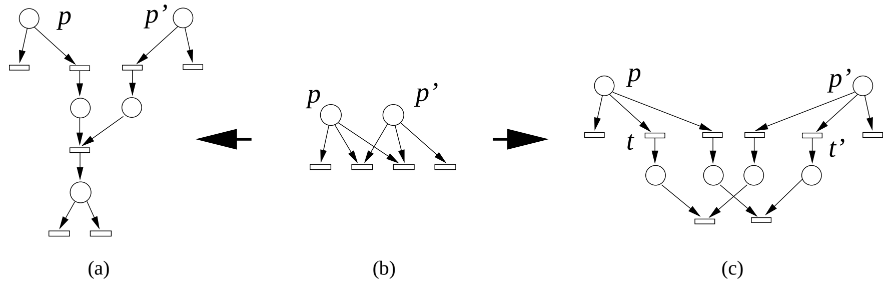


Fig. 17. Not grouping the EQ sets can lead to absurd killings of the system.

Proposition 24. *Let $\mathcal{S}$ be a P/T system and $\mathcal{S}'$ a system obtained releasing $\mathcal{S}$ with respect to one or several places.*

1. *$\mathcal{N}'$ is str. bounded iff $\mathcal{N}$ is str. bounded.*
2. *If $\mathcal{S}'$ is live, then $\mathcal{S}$ is live.*

Proof. It is not difficult to see that str. boundedness is not modified by release. For liveness, observe that any sequence in $\mathrm{L}(\mathcal{S})$ can be extended to a sequence in $\mathrm{L}(\mathcal{S}')$. Since $\mathcal{S}'$ is live, for any transition of $\mathcal{N}$ a sequence in $\mathrm{L}(\mathcal{S}')$ exists that enables it. This sequence can be projected so that it is in $\mathrm{L}(\mathcal{S})$. □

Release vs. Equalization Both release and equalization decrease the SEQS − $rank(\mathbf{C})$ difference (remember that if it is made zero or negative, then the target net violates Theorem 7, so it is non str. live and we cannot decide). By equalization the rank is maintained and |SEQS| is decreased, while by release the rank is augmented more than |SEQS| because each new transition increases the rank but some belong to the same EQ set. It can be shown that release and equalization *wrt. a place* decrease this difference in the same amount. However, they are not equivalent when a coupled conflict set is transformed into an equal conflict set, since release has to be done wrt. *all* the input places, while that is not always necessary for equalization. Take for instance the philosophers example in Figure 18 (a). To transform $\{t_1, t_3, t_5\}$ into an equal conflict set, release has to be done with respect to p_1, p_3, and p_5, making zero the difference between the rank of the token-flow matrix and |SEQS|, thus not allowing to decide, while equalization with respect to any two of them is enough, keeping the difference positive, thus allowing to prove str. liveness of the original net.

From a liveness preservation point of view, release and equalization are not comparable, i.e., in some cases the released system is live while the equalized system is non live, and vice-versa (Figure 18).

On the other hand, equalization does not have sense unless we have information about the LRG of the target system (Proposition 22). For instance, if we used {SC}*ECS as target class, equalization would be of no help, since in this class spurious deadlocks may exist [19].

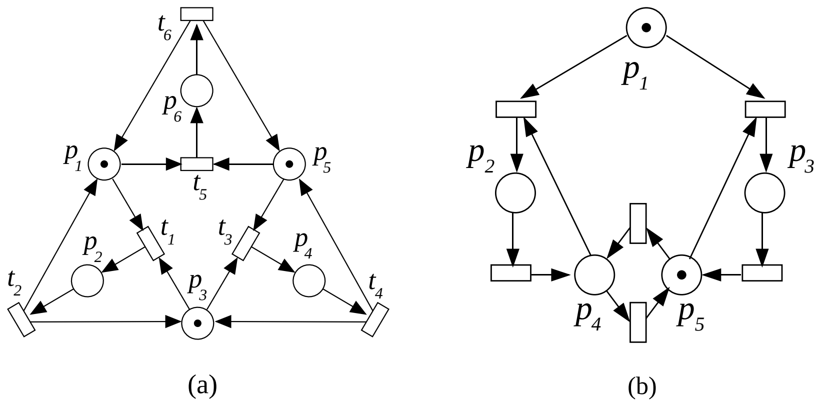


Fig. 18. Release of the system in (a) with respect to p_1, p_3 and p_5 makes it non (str.) live, while equalization with respect to the same places makes it live. However, the release of (b) with respect to p_1 is live and the equalization with respect to that place is dead.

4.3 Application of Transformations

Four main kinds of transformations have been considered in this section: removal of implicit places, removal of bypass transitions, equalization and release. Additionally, the removal of arbiters can be considered as a transformation–by–decomposition rule (see Proposition 4). In that sense, observe that a choice free local arbiter should not be removed if this reduces the rank of the token-flow matrix less than the number of transitions in the arbiter minus one (Proposition 6). These transformations (from $\mathcal{S}$ to $\mathcal{S}'$ or from $\mathcal{N}$ to $\mathcal{N}'$) can be classified into three main groups, according to what is preserved (structurally or behaviorally):

- Group I: Either $\mathcal{S}'$ live $\Leftrightarrow \mathcal{S}$ live, or $\mathcal{N}'$ str. live $\Leftrightarrow \mathcal{N}$ str. live
- Group II: Either $\mathcal{S}'$ live $\Rightarrow \mathcal{S}$ live, or $\mathcal{N}'$ str. live $\Rightarrow \mathcal{N}$ str. live
- Group III: Either $\mathcal{S}'$ non live $\Rightarrow \mathcal{S}$ non live, or $\mathcal{N}'$ non str. live $\Rightarrow \mathcal{N}$ non str. live

Table 1 summarizes most of the results about transformations. When the property relates liveness in both systems it is denoted with an “L”, and when the relationship is set at a structural level it is denoted with “SL”. Besides removal of implicit places, other classical reduction rules like pre- and post-fusion [2] or macroplaces [25] belong to group I, since they fully preserve liveness.

When using transformations to analyze a system, those in group II and those in group III cannot be mixed, because the ones in group II aim at proving (str.) liveness, while the one in group III try to prove non (str.) liveness. Transformations of group I can be used with any of them, since they keep all the information wrt. the properties under study.

A possible application strategy is: first, apply all the transformations you can of group I. Then, concentrate either in group II or group III. Assume for

		Preserves L&¬L	Preserves ¬L	Preserves L
I	Remove implicits	L,SL (Def. 14)		
II	Remove MSIP		SL (Prop. 17)	
	Remove local arbiters		SL (Prop. 4)	
	Equalization		L,SL (Prop. 22)	
	Release		L,SL (Prop. 24)	
III	Remove bypasses			L,SL (Prop. 21)

Table 1. Let $\mathcal{S}$ be the original system and $\mathcal{S}'$ the transformed one. The preservation of behavioral properties (liveness or non liveness) is denoted by "L", while the preservation of structural ones (str. liveness or non str. liveness) is denoted by "SL".

instance that group II is chosen, i.e., it is liveness of the system we intend to prove. Apply a transformation of group II and go on applying transformations of group I and group II till str. liveness or non str. liveness is proved or till you reach a dead end and no transformation can be applied. If the application of transformations leads to a non str. live system or to a dead end, then go back to the point where the last transformation of goup II was applied, i.e., undo all the transformations of group I and the last transformation of group II. If another transformation of group II can be applied, then proceed. Otherwise, repeat the backtracking.

It should be noticed that two different transformation sequences can lead to net systems with different properties, unless all the transformtions belong to group I.

Let us illustrate the suggested procedure with an example. Consider the system in Figure 19 (a). This net is consistent and conservative (thus str. bounded),

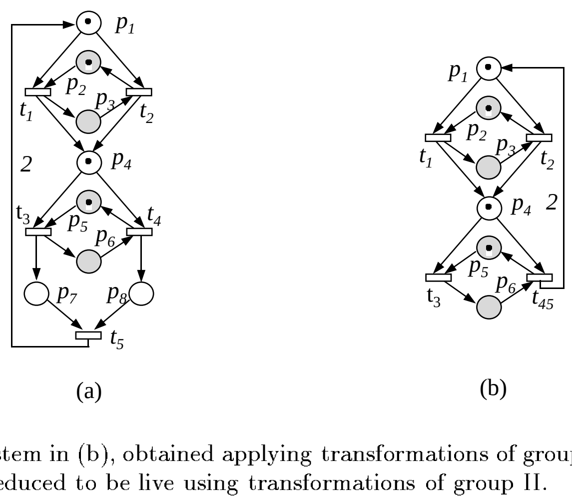


Fig. 19. The system in (b), obtained applying transformations of group I to the system in (a), can be deduced to be live using transformations of group II.

rank(**C**) = 4 and |SEQS| = 5, hence nothing can be said about liveness applying the general necessary condition (Theorem 7). Apply as many transformations as you can of group I: remove p_7 (implicit) and fuse transitions t_4 and t_5 into t_{45} (post-fusion [2]). The system obtained, Figure 19 (b), is irreducible wrt. group I. Assume we intend to prove str. liveness. Then, in the following we will use transformations in groups I and II. A possible transformation that can be done is to remove the circuit arbiter defined by p_5 and p_6. This makes the net non strongly connected, thus it is not str. live and str. bounded, and no conclusion can be obtained. Hence, this transformation must be undone. Instead of removing this circuit arbiter, we can also remove the circuit arbiter defined by p_2 and p_3. Now, places p_1 and p_4 can be fused into a macroplace [25], p_{14}, which is implicit. What remains after removing p_{14} is simply the circuit formed by p_5, t_3, p_6 and t_{45}, which is live. Thus, the original system is str. live.

5 Liveness and Dead Transitions

So far, we have concentrated on the use of the rank theorems for the analysis of str. liveness, not taking into account the actual marking of the system. This will allow to discard some non live systems. However, if the system is proven to be str. live (or we cannot decide whether it is str. live or not) liveness of the specific system should be studied. Using the general structural approach, based on the state equation relaxation, the system one might consider for the analysis of liveness is:

$$(\forall t, \mathbf{m})(\exists \boldsymbol{\sigma}')$$
$$(\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq \mathbf{0} \wedge \mathbf{m}' = \mathbf{m} + \mathbf{C} \cdot \boldsymbol{\sigma}' \geq \mathbf{0} \wedge \mathbf{m}' \geq \mathbf{Pre}[P, t] \wedge \boldsymbol{\sigma}, \boldsymbol{\sigma}' \geq \mathbf{0})$$

However, validity of the above predicate is neither sufficient for liveness, because the $\boldsymbol{\sigma}'$ may not be fireable, nor necessary, because an $\mathbf{m}$ invalidating it may be spurious. As a result we cannot even semidecide on this property, since we can only obtain necessary conditions for existentially quantified predicates, and sufficient conditions for universally quantified ones, due to the possible presence of spurious solutions.

Nevertheless, linear techniques can be used to give necessary conditions for liveness. *Dead transitions* are transitions that are not fireable from any reachable marking. The existence of *dead transitions* means that some actions can never be executed, what clearly implies that the system is not live. In other words, absence of dead transitions is necessary for liveness. *Deadlock-freeness* is another condition that is also clearly necessary for liveness. Notice that these two conditions are not related to each other, i.e., a system with dead transitions can be deadlock-free and a system without dead transitions can deadlock. For instance, the system in Figure 20 (a) is deadlock-free, while transitions t_3, t_4 and t_5 are dead; in the system in Figure 20 (b) no transition is dead and it deadlocks. In this section we will concentrate on dead transitions, and deadlock-freeness will be studied in the following one.

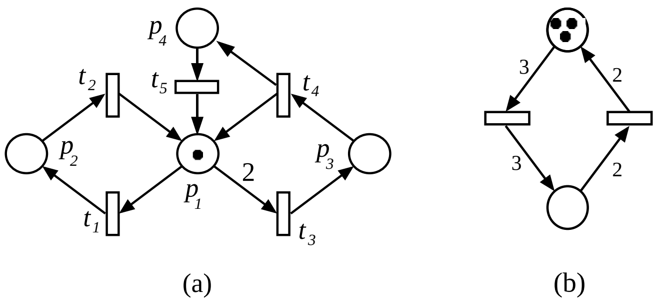


Fig. 20. Absence of dead transitions is neither necessary (a), nor sufficient (b), for deadlock-freeness.

An algebraic general sufficient condition for the existence of dead transitions can be obtained. Its negation will thus be a necessary condition for liveness.

Definition 25. Let $\mathcal{S}$ be a P/T system. A transition is dead iff it is not fireable from any reachable marking.

Proposition 26 (Sufficient condition for t dead).
Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a P/T system. If there is no (integer) solution to

$$\begin{aligned} \mathbf{m} - \mathbf{C} \cdot \boldsymbol{\sigma} &= \mathbf{m_0} \\ \mathbf{m} &\geq \mathbf{Pre}[P, t] \\ \mathbf{m} &\geq \mathbf{0} \\ \boldsymbol{\sigma} &\geq \mathbf{0} \end{aligned} \tag{3}$$

then t is dead.

The converse of Proposition 26 is not true. That is, even if (3) has a solution the transition could be dead, because such a solution might be non reachable, i.e., it might be an spurious solution. For instance, transition t_5 in Figure 20 (a) is dead, and a solution of (3) exists: $\mathbf{m} = [0, 0, 0, 1]$ and $\boldsymbol{\sigma} = [0, 0, 1, 1, 0]$ i.e., there exists an spurious marking that enables t_5.

Relaxing (3) into the reals, and using *duality theory* [15], an alternative formulation of the condition for a transition to be dead can be obtained. In this case, the dual problem of (3) leads to:

Corollary 27. *Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a P/T system. If there is a solution to*

$$\begin{aligned} \mathbf{y} \cdot \mathbf{C} &\leq \mathbf{0} \\ \mathbf{y} \cdot \mathbf{Pre}[P, t] - \mathbf{y} \cdot \mathbf{m_0} &> 0 \\ \mathbf{y} &\geq \mathbf{0} \ , \end{aligned} \tag{4}$$

then t is dead.

Existence of a solution to (4) can be interpreted as follows. If it has a solution, then a set of str. bounded places exists, defined by $||\mathbf{y}||$. This vector induces the following marking invariant relation: $\mathbf{y} \cdot \mathbf{m} \leq \mathbf{y} \cdot \mathbf{m_0}$. To fire transition t, a content of tokens with a "weight" given by $\mathbf{y} \cdot \mathbf{Pre}[P, t]$ is required on places in $||\mathbf{y}||$. But so much "token weight" can never be collected in these places, because $\mathbf{y} \cdot \mathbf{Pre}[P, t] > \mathbf{y} \cdot \mathbf{m_0}$, and every reachable marking must fulfill $\mathbf{y} \cdot \mathbf{m} \leq \mathbf{y} \cdot \mathbf{m_0}$.

If a P/T system is live, then there is no dead transition, i.e., all transitions are fireable at least once. Thus, we can state the following polynomial time necessary condition for liveness, that requires checking for absence of solution $|T|$ linear systems of equations in the worst case.

Corollary 28. *Let $\mathcal{S}$ be a P/T system. If $\mathcal{S}$ is live, then there is no $t \in T$ for which a solution to (4) exists.*

Clearly, absence of dead transitions is in general not sufficient for liveness. As usual, better results are obtained if subclasses of systems are considered. For instance, in free choice systems liveness is equivalent to str. liveness and every P-semiflow being marked [8], or alternatively, to str. liveness and every transition being non dead.

Proposition 29. *Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a free choice system. It is live and bounded iff*

1. *$\mathcal{N}$ is strongly connected, consistent and* $\text{rank}(\mathbf{C}) = |\text{SEQS}| - 1$*, and*
2. *no transition is dead*

Proof. If the system is live and bounded, it is clear that it does not contain any dead transition. Applying Theorem 13 the net is strongly connected, consistent and $\text{rank}(\mathbf{C}) = |\text{SEQS}| - 1$.

Assume the system strongly connected, consistent and $\text{rank}(\mathbf{C}) = |\text{SEQS}| - 1$, and no transition is dead. Applying Theorem 13 and Corollary 11, it is str. live and str. bounded. By the results in [31], liveness of a str. live and str. bounded EQ system (in particular free choice) is equivalent to liveness of all the P-subsystems that its minimal P-semiflows define. In the case of free choice nets these P-subsystems are strongly connected SM [9]. Consider a minimal P-semiflow, $\mathbf{y}$ and let t be a transition belonging to the SM it defines. Since t is not dead, a sequence σ exists that enables it, i.e., $\mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \geq \mathbf{Pre}[P, t]$. Thus, $\mathbf{y} \cdot \mathbf{m_0} \geq \mathbf{y} \cdot \mathbf{Pre}[P, t]$. Moreover, $\mathbf{y} \cdot \mathbf{Pre}[P, t] \geq 1$ because t is in the SM the P-semiflow defines. Therefore $\mathbf{y} \cdot \mathbf{m_0} \geq 1$, that is, the SM is marked and so it is live. Repeating with every minimal P-semiflow, liveness of the free choice system is deduced. □

Algebraically, liveness of a str. live and bounded free choice net (i.e., strongly connected consistent and $\text{rank}(\mathbf{C}) = |\text{SEQS}| - 1$) is equivalent to non existence of solution to (5), what can be checked in polynomial time.

$$\begin{aligned} \mathbf{y} \cdot \mathbf{C} &= \mathbf{0} \\ \mathbf{y} \cdot \mathbf{m_0} &= 0 \\ \mathbf{y} &\geq \mathbf{0} \end{aligned} \tag{5}$$

This property does not hold in general. In fact, it even fails in the simple case of weighted circuits (see Figure 20(b)). For this subclass (and EQ systems or DSSP) more information is obtained from the analysis of deadlock-freeness, which is equivalent to liveness if the system is bounded and strongly connected (Theorem 9.1).

6 Deadlock-Freeness

Deadlock-freeness is another necessary condition for liveness that can be (partially) analyzed using the state equation. In net terms, a deadlock corresponds to a marking from which no transition is fireable. Thus, absence of deadlock is also necessary for liveness, although in general not sufficient.

Clearly, every reachable deadlock is a solution to the state equation where every transition is disabled, what leads to the following basic general sufficient condition for deadlock-freeness:

Theorem 30. *Let $\mathcal{S}$ be a P/T system. If there is no (integer) solution to*

$$\begin{aligned} \mathbf{m} - \mathbf{C} \cdot \boldsymbol{\sigma} &= \mathbf{m_0} \\ \mathbf{m}, \boldsymbol{\sigma} &\geq \mathbf{0} \\ \textstyle\bigvee_{p \in {}^\bullet t} \mathbf{m}[p] &< \mathbf{Pre}[p,t] \quad \forall t \in T \end{aligned} \tag{6}$$

then $\mathcal{S}$ is deadlock-free.

Unfortunately, it is a non linear system and its resolution poses practical problems. A possible way of solving it is to use an alternative statement as the absence of solutions to a set of systems of linear equations in the integer domain. This set of systems is defined by taking each time one of the conditions linked disjunctively, i.e., considering for each transition one of its input places as the "culprit" for its disabledness. The problem is that we have to check $\prod_{t \in T} |{}^\bullet t|$ linear systems.

This can be improved by using specific transformation techniques [29]. Firstly, if $\mathbf{Pre}[P,t] \leq \mathbf{Pre}[P,t']$ for some t and t', the disabledness condition for t' can be removed. We can also remove from (6) the disabledness conditions of transitions that are known to be dead (Section 5). These transformations do not affect the set of solutions to (6), even over the reals.

Although disregarding some transitions applying the above arguments may be helpful, typically the more drastic reduction in the number of systems to check is produced by the results that we present in the sequel. They provide rules to rewrite the disabledness condition of a transition in a less complex way while preserving the set of *integer* solutions to (6).

Proposition 31 ([29]). *Let t be a transition such that for every $p \in \pi \subseteq {}^\bullet t$ the following holds: $\mathbf{sb}[p] \leq \mathbf{Pre}[p,t]$. Replacing in (6) for the disabledness condition corresponding to transition t the following (less complex) condition:*

$$\left(\sum_{p \in \pi} \mathbf{m}[p] < \sum_{p \in \pi} \mathbf{Pre}[p,t] \right) \vee \left(\bigvee_{p \in {}^\bullet t \setminus \pi} \mathbf{m}[p] < \mathbf{Pre}[p,t] \right)$$

the set of integer *solutions is preserved.*

By the application of this result to a transition t, the number of linear systems to be solved is divided by $\frac{|^\bullet t|}{|^\bullet t|-|\pi|+1}$, what is deduced from the ratio between the number of input places to the transition and the actual number of them that need being considered separately. In the particular case where $\pi = {}^\bullet t$, the disabledness condition is reduced to a linear inequality.

This is for instance the case of the system obtained from the one in Figure 11 by removing the implicit place p_{345}. It can be easily proven to be live by checking the absence of solutions of the following linear system:

$$\begin{aligned} \mathbf{m} - \mathbf{C} \cdot \boldsymbol{\sigma} &= \mathbf{m_0} \\ \mathbf{m}[p_8] = \mathbf{m}[p_6] &= 0 \\ \mathbf{m}[p_1] + \mathbf{m}[p_2] &\leq 1 \\ \mathbf{m}[p_7] + \mathbf{m}[p_9] &\leq 1 \\ \mathbf{m}, \boldsymbol{\sigma} &\geq \mathbf{0} \end{aligned} \tag{7}$$

(In this particular case, deadlock-freeness can also be proven by observing that this is a marked graph and no unmarked circuit exists.)

Also when *all but one* of the input places of a transition are such that their str. bound equals to the weight of the arc, the disabledness condition of the transition can be reduced to a linear inequality, applying the following result, which generalizes Proposition 31:

Proposition 32 ([29]). *Let t be a transition such that ${}^\bullet t = \pi \cup \{p'\}$, where $\mathbf{sb}[p'] > 0$ and $\mathbf{sb}[p] \leq \mathbf{Pre}[p,t]$ for every $p \in \pi$. Replacing in (6) for the disabledness condition corresponding to transition t the following (less complex) condition:*

$$\mathbf{sb}[p'] \cdot \sum_{p\in\pi} \mathbf{m}[p] + \mathbf{m}[p'] \leq \mathbf{sb}[p'] \cdot \sum_{p\in\pi} \mathbf{Pre}[p,t] + \mathbf{Pre}[p',t] - 1 \tag{8}$$

the set of integer *solutions is preserved.*

By the application of this result to a transition t, the number of linear systems to be solved is obviously divided by $|^\bullet t|$.

We can still further reduce the number of systems to solve by pre-applying a transformation to the system that preserves deadlock-freeness (actually, it preserves the projected language). The transformation, illustrated in Figure 21, can be applied as needed to every place p with homogeneous weighting ($\mathbf{Pre}[p, p^\bullet] = w\mathbf{1}$). After the transformation, we have one more transition ($t^{(p)}$ in the figure), the disabledness condition of which can be written as a linear inequality because the str. bound of $p^{(c)}$ is one. On the other hand, the str. bound of $p^{(b)}$ is also one, thus we have in each transition in $p^\bullet$ one input place less with str. bound greater than the weight (perhaps only one, or even none, remains, and then the disabledness condition for such transition can be written as a linear inequality too).

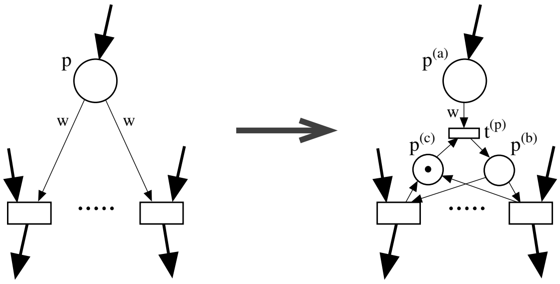


Fig. 21. A transformation preserving the projected language (in particular, preserving deadlock-freeness).

After the presented results, clearly the state equation based verification of deadlock-freeness reduces to checking non-existence of (integer) solution to a *single* linear system of inequalities in the case of str. bounded P/T systems with homogeneous weighting — in particular equal conflict systems — because the transformation in Figure 21 can be applied as necessary to enable Propositions 31 or 32. Moreover, since every P/T system can be simulated by another with homogeneous weighting preserving the projected language (see Figure 22 for an

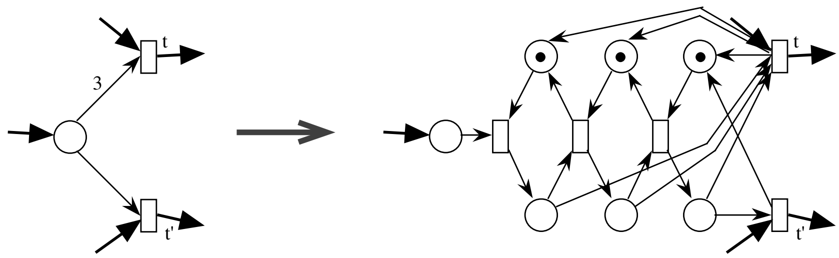


Fig. 22. Simulating weights with ordinary nets preserving the projected language.

illustrative example of the kind of transformation used), it follows that *every* str. bounded P/T system (or merely known to be k-bounded, because these can always be made str. bounded using the complementary place construction) can be transformed to require a single linear system of inequalities:

Theorem 33. *Let $\mathcal{S}$ be a str. bounded P/T system. Then (6) in the sufficient condition for deadlock-freeness given by Theorem 30 can be rewritten as a single system of linear inequalities.*

Although the transformation shown in Figure 21 can always be applied, more compact transformations exist for particular classes. For instance, for DSSP, where the only places with marking bound possibly greater than the corresponding arc weight are the buffers, it is possible to transform the net so that every transition has at most one input buffer, as it clearly follows from the rules shown in Figure 23. The first rule is obvious. The second rule can be applied in the case

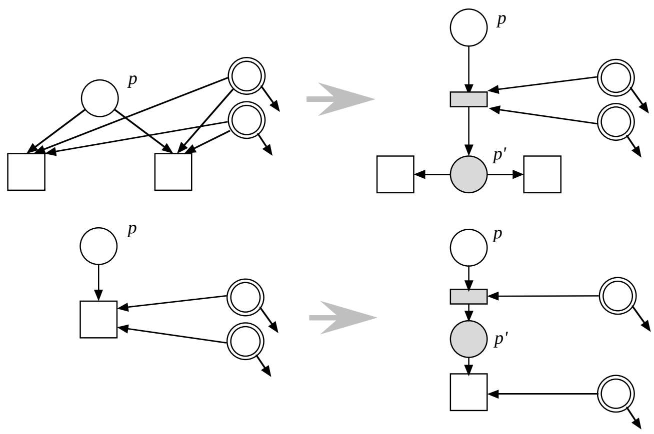


Fig. 23. Transformations to avoid having more than one input buffer in the same transition.

of DSSP because all the outputs of a buffer belong to the same sequential agent ("destination private" constraint). Therefore, when a transition requiring tokens from two buffers is enabled, these tokens can be collected in sequence because no other transition can remove them in the meanwhile (at any time, only one transition is enabled in a live and safe state machine).

Putting together the above results, a possible schema for the analysis of deadlock-freeness could be: First, compute the str. bound for the marking of every place. Remove every transition such that the str. bound for one of its input places is less than the weight of the arc to the transition (clearly these transitions will be dead). Remove also any transition t for which t' exists with $\mathbf{Pre}[P,t] > Pre[P,t']$. Then, select the transitions to which Proposition 32 cannot be applied (i,e., $\mathbf{sb}[p] > \mathbf{Pre}[p,t]$ for more than one input place). For each input place of these transitions with $\mathbf{sb}[p] > \mathbf{Pre}[p,t]$, make its output arcs homogeneously weighted (for instance, as in Figure 21) and apply the transformation in Figure 22. The equivalent system thus obtained verifies that for each transition t at most one input place p exists with $\mathbf{sb}[p] > \mathbf{Pre}[p,t]$. Using Proposition 32 a single system of inequalities is obtained. If this system does not have an integer solution, the system is deadlock-free.

Moreover, in the case of EQ systems or DSSP, absence of solutions to the system is not only sufficient, but also necessary for deadlock-freeness. The reason is that, a bounded and strongly connected deadlock-free EQ system or DSSP is live (Theorem 9.1), and therefore any marking that is solution of the state equation has a reachable successor (Theorem 9.2). Thus, there cannot be spurious deadlocks. Hence, analyzing liveness in these subclasses is specially simple: First, check if a marking that makes the system live and bounded exists by checking strong connectedness, consistency and rank($\mathbf{C}$) = |SEQS| − 1 (Theorem 13). Assuming this holds, checking deadlock-freeness (which, as said before, is equivalent to checking the absence of an integer solution to a system of inequalities) allows to conclude on liveness.

Transformation techniques may also allow to apply this method for the analysis of systems for which in principle could not be used. Assume we have been able to prove that a net is str. live by transforming it into an EQ or DSSP net. If the transformations preserve not only str. liveness but effective liveness, the equivalence of liveness and deadlock-freeness in these subclasses allows to reduce the problem to solving absence of deadlock in the transformed system. For example, liveness of the system in Figure 8 (a) can be deduced from the absence of solution to the system of equations in (7).

7 Example

In this section we will apply the previous results to analyze liveness in an example.

Consider the system represented in Figure 24, composed of two producers,

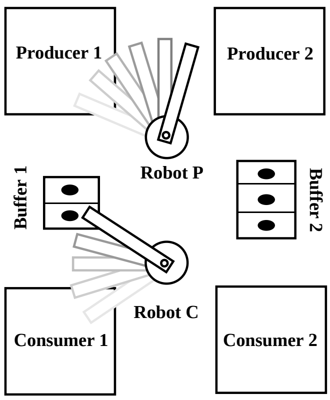


Fig. 24. A manufacturing system composed of two producers, two consumers, and two robots with two intermediate buffers.

two consumers, two robots and two buffers. Each producer manufactures one kind of parts that are consumed by one of the consumers. One robot takes the parts the producers have manufactured and transports them to their buffer, while the other one removes the parts from the buffers according to the consumers' requests. The buffers capacities are two and three respectively. A P/T model of this system is represented in Figure 25 (a).

This net is consistent, conservative, and rank($\mathbf{C}$) = 10. But |SEQS| = 12 and |SCCS| = 10, and thus the general necessary or sufficient conditions do not

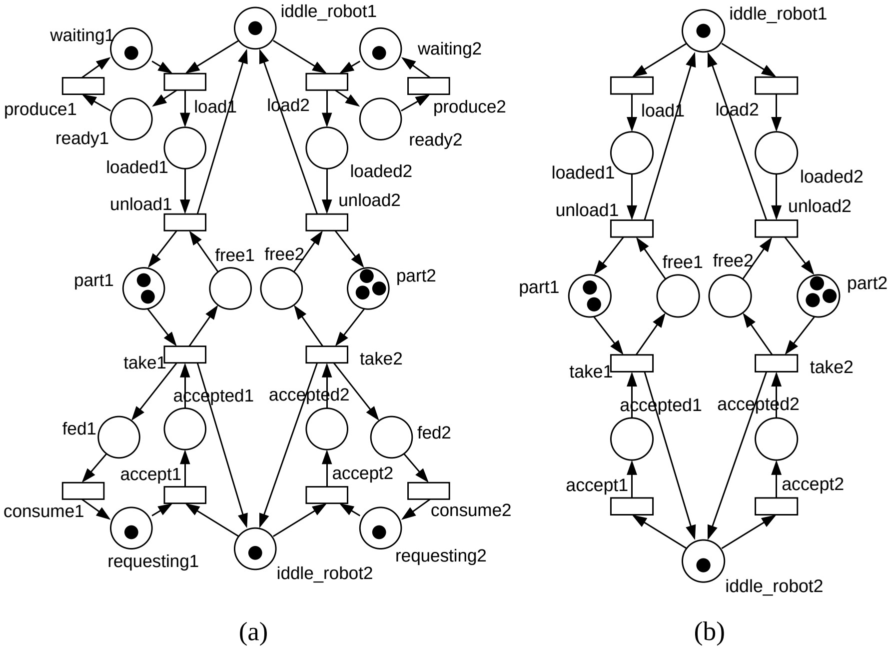
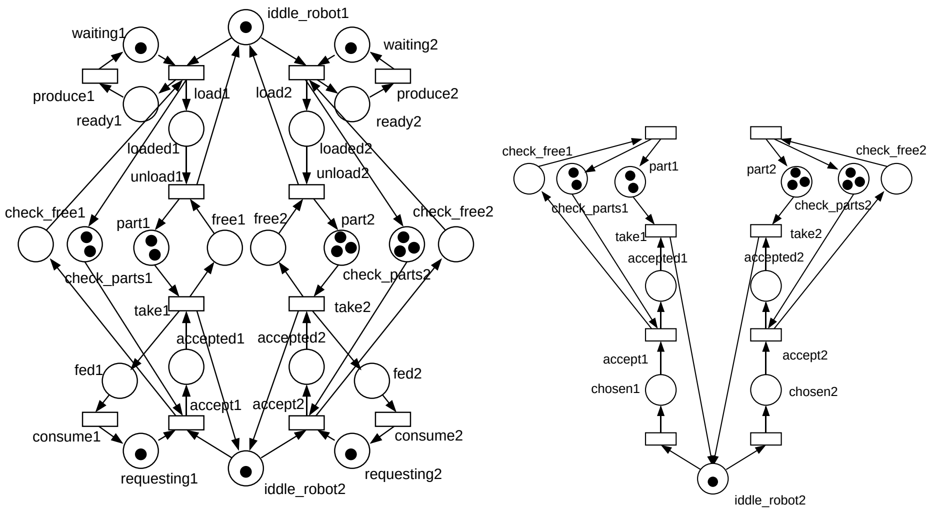


Fig. 25. A P/T representation of the manufacturing system in Figure 24.

allow to decide about liveness in any sense. However, some transformations can be applied that will allow to decide. Places *waiting1* and *ready1* and the transition *produce1* can be grouped into a single place that self-loops transition *load1* without affecting liveness (fusion of places [2,26]). Moreover, this new place is implicit, thus by Proposition 17, it can be removed without modifying liveness. Analogous transformations can be done to remove the circuits representing the other producer and both consumers. The system obtained is represented in Figure 25 (b). Now $\text{rank}(\mathbf{C}) = 6 = |\text{SEQS}|$, therefore cannot be lively marked and the system in Figure 25 (a) is not live.

Observe that problems in this system arise because robots do not check the existence of parts or space in the buffers before accepting a request to remove or put a part, what can lead to a deadlock. This can be solved by adding a pair of status variables that record the state of the buffers and which the robots will check before accepting a request by a producer or a consumer, Figure 26 (a). Again, liveness cannot be analyzed using the general conditions (the system is consistent, conservative and $\text{rank}(\mathbf{C}) = 10$, while $|\text{SEQS}| = 12$ and $|\text{SCCS}| = 10$), but this can be improved applying some transformations.

The circuits representing the consumers and the producers can be removed as before. The places that represent the number of empty positions in the buffers, *free1* and *free2*, are implicit. (They can be obtained by adding *loaded1*, *check_free1*, *iddle_robot2*, and *accepted2*, and *loaded2*, *check_free2*, *iddle_robot1*,



(a) (b)

Fig. 26. The addition of status variables makes the system live.

and *accepted1*, respectively, and solving the systems given in (2) it can be seen the marking is enough to make them implicit.) Therefore they can be removed and the system will be live or non live just the same. Now, *load_part1*, *loaded1* and *unload1* can be grouped into one transition (fusion of transitions [2,26]) and analogously *load_part2*, *loaded2* and *unload2*. This transforms *iddle_robot1* into an implicit place that can be removed. Applying release wrt. *iddle_robot2* the system is transformed into a free choice system. This system, represented in Figure 26 (b), is consistent, conservative, and $\text{rank}(\mathbf{C}) = 6 = |\text{SEQS}| - 1$. Thus, this system is str. live, and so is the original one. (This could also have been decided before doing the release, applying the sufficient condition of Proposition 23.) Moreover, applying the particular results of free choice nets, we can prove that this system is live just by checking that there is no dead transition, i.e., no solution of the system in (5) exists. Hence, by Proposition 29 we can deduce that the system in Figure 26 (a) is live.

8 Conclusions

In this paper several linear algebraic techniques have been proposed for the analysis of liveness. Other structural approaches using concepts based on graph theory, (namely particular classes of siphons) can be found in [1,12].

The analysis of liveness is divided in two main phases: First, analyzing str. liveness and second, if the net is str. live, or this could not be disproved, analyzing liveness. In the first part, it is the structure of the net we mainly concentrate on, and not the actual marking. We have given a general necessary condition for the existence of a live and bounded marking for a net, the *rank theorem* (Theorem 7). This condition allows to detect in some cases mismatches between choices and synchronizations that preclude liveness.

For some subclasses, in particular EQ systems or DSSP, the kind of problems this condition pinpoints to, are in fact the only possible ones, what makes it a complete characterization for the existence of a marking that makes the system live and bounded (Theorem 13). Moreover, other specific results, such as the equivalence of liveness and deadlock-freeness, or the absence of spurious deadlocks, makes the analysis of liveness for these subclasses specially simple (Theorem 9).

Unfortunately, if the system does not belong to one of these subclasses, it may be the case that the rank theorem does not conclude. We have studied some transformations that may help in the analysis of these systems. Table 1 summarizes the results obtained, which do not intend to be exhaustive.

If the system is finally proven to be str. live, or at least we have not been able to disprove it, liveness should be analyzed. Two basic necessary conditions for liveness have been studied using linear algebraic techniques: absence of dead transitions (Section 5) and deadlock-freeness (Section 6). For both of them sufficient conditions based on the analysis of the state equation have been given. In particular, it has been proven that a simple sufficient condition for deadlock-freeness, which is also necessary for EQ systems or DSSP, can be stated in terms of absence of solutions of a *single* system of linear inequalities in the integer domain (Theorem 33).

Although this method will not always allow to prove that a system is live, it will help to detect non liveness in most practical systems, avoiding the use of more costly techniques. It can also help to solve the problems these faulty systems may present by signaling the cause of their non liveness: existence of a structural pathology or of a marking leading to the existence of dead transitions or a deadlock (Section 7). For interesting net subclasses (EQ, DSSP), str. liveness is always decidable in polynomial time, and liveness for a particular $\mathbf{m_0}$ is also decidable using linear algebraic techniques. For free choice nets, liveness is equivalent to absence of dead transitions, what can be decided in polynomial time.

Acknowledgement

We thank Peter Kemper (University of Dortmund), Hassane Alla (University of Grenoble), José Manuel Colom, Joaquín Ezpeleta, and one anonymous referee for their helpful comments and suggestions.

References

1. K. Barkaoui and J.F. Pradat-Peyre. On liveness and controlled siphons in Petri nets. In Billington and Reisig [4], pages 57–72.

2. G. Berthelot. Checking properties of nets using transformations. In G. Rozenberg, editor, *Advances in Petri Nets 1985*, volume 222 of *Lecture Notes in Computer Science*, pages 19–40. Springer, 1986.
3. G. Berthelot. Transformations and decompositions of nets. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Central Models and their Properties. Advances in Petri Nets 1986, Part I*, volume 254 of *Lecture Notes in Computer Science*, pages 359–376. Springer, 1987.
4. J. Billington and W. Reisig, editors. *Application and Theory of Petri Nets 1996*, volume 1091 of *Lecture Notes in Computer Science*. Springer, 1996.
5. J. Campos, G. Chiola, and M. Silva. Properties and performance bounds for closed free choice synchronized monoclass queueing networks. *IEEE Trans. on Automatic Control*, 36(12):1368–1382, 1991.
6. J. M. Colom and M. Silva. Improving the linearly based characterization of P/T nets. In Rozenberg [22], pages 113–145.
7. J. Desel. Regular marked Petri nets. In J. Leeuwen, editor, *WG' 93:* 19^{th} *Int. Workshop on Graph-Theoretic Concepts in Computer Science*, volume 790 of *Lecture Notes in Computer Science*, pages 264–275. Springer, 1993.
8. J. Esparza and M. Silva. On the analysis and synthesis of free choice systems. In Rozenberg [22], pages 243–286.
9. M. H. T. Hack. Analysis of production schemata by Petri nets. Master's thesis, M.I.T., Cambridge, MA, USA, 1972. (Corrections in *Computation Structures Note* 17, 1974).
10. M. H. T. Hack. Petri net languages. Technical Report Technical Report 159, Laboratory for Computer Science, M.I.T., Cambridge, MA, USA, 1976.
11. W. E. Kluge and K. Lautenbach. The orderly resolution of memory access conflicts among competing channel processes. *IEEE Trans. on Computers*, 31(3):194–207, 1982.
12. K. Lautenbach and Hanno Ridder. Liveness in bounded Petri nets which are covered by t-invariants. In R. Valette, editor, *Application and Theory of Petri Nets 1994*, volume 815 of *Lecture Notes in Computer Science*, pages 358–375. Springer, 1994.
13. G. Memmi and G. Roucairol. Linear algebra in net theory. In W. Brauer, editor, *Net Theory and Applications*, volume 84 of *Lecture Notes in Computer Science*, pages 213–223. Springer, 1979.
14. T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.
15. K. G. Murty. *Linear Programming*. Wiley and Sons, 1983.
16. C. V. Ramamoorthy and G. S. Ho. Performance evaluation of asynchronous concurrent systems using Petri nets. *IEEE Trans. on Software Engineering*, 6(5):440–449, 1980.
17. L. Recalde, E. Teruel, and M. Silva. Modeling and analysis of sequential processes that cooperate through buffers. *IEEE Trans. on Robotics and Automation*. To appear.
18. L. Recalde, E. Teruel, and M. Silva. On well-formedness analysis: The case of deterministic systems of sequential processes. In J. Desel, editor, *Proc. of the Int. Workshop on Structures in Concurrency Theory (STRICT)*, Workshops in Computing, pages 279–293. Springer, 1995.
19. L. Recalde, E. Teruel, and M. Silva. {SC}*ECS: A class of modular and hierarchical cooperating systems. In Billington and Reisig [4], pages 440–459.

20. L. Recalde, E. Teruel, and M. Silva. A structural view of hierarchical deterministically synchronized sequential processes. Research report, Dep. Informática e Ingeniería de Sistemas, Universidad de Zaragoza, María de Luna, 3, 50015 Zaragoza, Spain, 1997.
21. W. Reisig. Deterministic buffer synchronization of sequential processes. *Acta Informatica*, 18:117–134, 1982.
22. G. Rozenberg, editor. *Advances in Petri Nets 1990*, volume 483 of *Lecture Notes in Computer Science*. Springer, 1991.
23. M. W. Shields. *An Introduction to Automata Theory*. Blackwell Scientific Publications, 1987.
24. J. Sifakis. Structural properties of Petri nets. In J. Winkowski, editor, *Mathematical Foundations of Computer Science 1978*, pages 474–483. Springer, 1978.
25. M. Silva. *Las Redes de Petri: en la Automática y la Informática*. AC, 1985.
26. M. Silva. Introducing Petri nets. In *Practice of Petri Nets in Manufacturing*, pages 1–62. Chapman & Hall, 1993.
27. M. Silva, E. Teruel, and J. M. Colom. Linear algebraic techniques for the analysis of net systems. In G. Rozenberg and W. Reisig, editors, *Advances in Petri Nets*, Lecture Notes in Computer Science. Springer. To appear.
28. Y. Souissi and N. Beldiceanu. Deterministic systems of sequential processes: Theory and tools. In *Concurrency 88*, volume 335 of *Lecture Notes in Computer Science*, pages 380–400. Springer, 1988.
29. E. Teruel, J. M. Colom, and M. Silva. Linear analysis of deadlock-freeness of Petri net models. In *Procs. of the* 2^{nd} *European Control Conference*, volume 2, pages 513–518. North-Holland, 1993.
30. E. Teruel, J. M. Colom, and M. Silva. Choice-free Petri nets: A model for deterministic concurrent systems with bulk services and arrivals. *IEEE Trans. on Systems, Man, and Cybernetics*, 27(1), 1997.
31. E. Teruel and M. Silva. Structure theory of equal conflict systems. *Theoretical Computer Science*, 153(1-2):271–300, 1996.