On Partially Controlled Petri Nets that can be made Live by Supervision

Ramavarapu S. Sreenivas
Coordinated Science Laboratory &
Department of General Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801
rsree@uiuc.edu

Abstract—The existence of a supervisory policy that enforces liveness in partially controlled Petri nets (PNs) is undecidable in general (cf. theorem 5.3 and corollary 5.2, [9]). However, there can be families of partially controlled PNs for which the existence of a supervisory policy that enforces liveness can be tested and synthesized. In this paper we present choice-controlled PNs as an example of such a family.

Also, there can be specific examples of partially controlled PNs that can be made live by supervision. Typically, this is shown by producing a supervisory policy that enforces liveness in the example at hand. There can be no generalization of this approach in light of theorem 5.3 and corollary 5.2 in reference [9]. However, for partially controlled Free-Choice PNs, there is a necessary and sufficient characterization of supervisory policies that enforce liveness. In specific, the necessary and sufficient condition for the existence of a supervisory policy that enforces liveness in a fully controlled Free-Choice Petri Net (FCPN) in any reachable state of a FCPN. This condition can be used to verify that a given supervisory policy does enforce liveness in a specific partially controlled FCPN.

I. INTRODUCTION

A large class of systems can be modeled as systems with independent, interacting, concurrent components. Typically, each independent process is split into several operations; the execution of each operation is conditioned on the satisfaction of a set of logical preconditions. Upon the execution of any such operation, a new set of logical conditions is created that inhibit the execution of some operations and enables the execution of others in the system. The supervisory control of such systems requires an external agent to regulate, or limit, the operations of each component so as to guarantee a common objective. In this paper we concern ourselves with a stronger version of deadlock avoidance called liveness. From any reachable state of a live system, it should be possible for any of the components to execute any of its operations, although not necessarily immediately.

Petri nets (PNs) [6], [7] are an ideal choice of the modeling of such systems as they allow easy representation of the logical preconditions as the marking, and the operations can be represented as transitions. A PN is said to be live, if from any reachable marking it is possible to fire any transition, although not necessarily immediately. A Free-Choice Petri net (FCPN) is a restricted class of Petri nets (PNs) where every arc from a place to a transition is either the unique output arc from that place, or, it is the unique input arc to the transition. FCPNs can model the flow of objects in production systems [1].

In this paper we consider partially controlled PNs where the external agent, the supervisor, can prevent only some (i.e. not all) transitions from firing. We concern ourselves with the synthesis of supervisory policies that use the marking of the PN to decide transitions that should be prevented from firing. In general, it is not possible to decide if a given partially controlled PN can be made live via supervision (cf. theorem 5.3 and corollary 5.2, [9]). In this paper we are interested in identifying families of partially controlled PNs that can be made live via supervision. We identify the family of choice-controlled PNs (and a refinement to this family), and we show that it is possible to decide if any (refined) choice-controlled PN can be made live via supervision. In case the choice-controlled PN can be made live, we present the minimally restrictive policy that enforces liveness.

In some cases we might be interested in showing that a specific instance of a partially controlled PN can be made live by supervision. Usually, this is done by showing that a particular supervisory policy enforces liveness in a specific instance. We consider this problem for partially controlled FCPNs, and we present a necessary and sufficient condition that characterizes supervisory policies that enforce liveness. This condition is useful in establishing the fact that a certain supervisory policy enforces liveness in a partially controlled FCPN.

Section II contains the notations and definitions used in this paper. The main results are presented in section IV after a review of the relevant results in the literature in section III. The paper concludes with section V, where the main results of the paper are presented in precis.

II. NOTATIONS AND DEFINITIONS

A Petri net (PN) \( \mathcal{N} = (I,T,\Phi,\mathbf{m}^0) \) is an ordered 4-tuple, where \( I = \{p_1, \ldots, p_n\} \) is a set of \( n \) places, \( T = \{t_1, \ldots, t_m\} \) is a collection of \( m \) transitions, \( \Phi \subseteq (I \times T) \cup (T \times I) \) is a set of arcs, \( \mathbf{m}^0 : I \rightarrow \mathcal{N} \) is the initial marking function (or the initial marking), and \( \mathcal{N} \) is the set of non-negative integers. The state of a PN is given by the marking \( \mathbf{m} : I \rightarrow \mathcal{N} \) which indicates the distribution of tokens in each place. For a given marking \( \mathbf{m} \), a transition \( t \in T \) is said to be enabled if \( \forall p \in \mathbf{t}, \mathbf{m}(p) \geq 1 \), where
For a given marking \( m \) the set of enabled transitions is denoted by the symbol \( T_e(N, m) \). An enabled transition \( t \in T_e(N, m) \) can fire, which changes the marking \( m \) to \( m' \) according to the equation

\[
m'(p) = m(p) - \text{card}(p^* \cap \{t\}) + \text{card}(p \cap \{t\}),
\]

where \( * \) := \( \{y \mid (y, x) \in \Phi\} \). For a given marking \( m \) the set of enabled transitions is denoted by the symbol \( T_e(N, m) \). An enabled transition \( t \in T_e(N, m) \) can fire, which changes the marking \( m \) to \( m' \) according to the equation

\[
m'(p) = m(p) - \text{card}(p^* \cap \{t\}) + \text{card}(p \cap \{t\}),
\]

where \( * \) := \( \{y \mid (y, x) \in \Phi\} \) and the symbol \( \text{card}(\cdot) \) is used to denote the cardinality of the set argument.

A collection of places \( P \subseteq \Pi \) is said to be a siphon (trap) if \( *P \subseteq P^* \) \( (P^* \subseteq P) \). A trap (siphon) \( P \), is said to be minimal if \( \exists \bar{P} \subset P \) such that \( P^* \subseteq \bar{P} \) \( (\bar{P} \subseteq P^*) \).

A string of transitions \( \omega = t_1t_2 \cdots t_k \), where \( t_i \in T \) \( (i \in \{1, 2, \ldots, k\}) \) is said to be a valid firing sequence starting from the marking \( m \), if,

1) the transition \( t_1 \) is enabled under the marking \( m \), and
2) for \( i \in \{1, 2, \ldots, k-1\} \) the firing of the transition \( t_i \) produces a marking \( m' \) under which the transition \( t_{i+1} \) is enabled.

If \( m^1 \) results from the firing of \( \omega = t^* \) starting from the initial marking \( m^0 \), we represent it symbolically as \( m^0 \rightarrow \omega \rightarrow m^1 \). Given an initial marking \( m^0 \) the set of reachable markings for \( m^0 \) denoted by \( R(N, m^0) \), is defined as the set of markings generated by all valid firing sequences starting with marking \( m^0 \) in the PN \( N \).

A transition \( t \in T \) is a PN \( N = (\Pi, T, \Phi, m^0) \) is live, if

\[
\forall m^1 \in R(N, m^0), \exists m^1 \in R(N, m^1), \text{ such that } t \in T_e(N, m^1).
\]

A PN \( N = (\Pi, T, \Phi, m^0) \) is a Free-Choice PN (FCPN) if

\[
\forall p \in \Pi, \text{card}(p^*) > 1 \Rightarrow \text{card}(p^*) = \{p\}.
\]

In other words, a PN is Free-Choice if and only if an arc from a place to a transition is either the unique output arc from that place, or, is the unique input arc to the transition. Commoner’s Liveness Theorem (cf. [1], chapter 4, [7]) states an FCPN \( N \) is live if and only if every minimal siphon in \( N \) contains a minimal trap that has a non-empty token load.

We assume a subset of transitions, called controllable transitions, \( T_c \subseteq T \) can be prevented from firing by an external agent called the supervisor. The set of uncontrollable transitions, denoted by \( T_u \subseteq T \), is given by \( T_u = T - T_c \). If \( T_c = T \), then we say we have a fully-controlled PN, otherwise we have a partially controlled PN. The fully-controlled equivalent of a partially controlled PN \( N = (\Pi, T, \Phi, m^0) \), is a PN where \( T_c = T \). That is, a full-controlled equivalent of a partially controlled PN essentially assumes all transitions are controllable.

A supervisory policy \( P : N^o \rightarrow \{0, 1\}^m \), is a total map that returns an \( m \)-dimensional binary vector for each reachable marking. The supervisory policy \( P \) permits the firing of transition \( t_i \) at marking \( m \), only if \( P(m_i) = 1 \). If at a marking \( m \) all input places to a transition \( t_i \) contain a token, we say the transition \( t_i \) is state-enabled at \( m \). If \( P(m_i) = 1 \), we say the transition \( t_i \) is control-enabled at \( m \). A transition has to be state-enabled and control-enabled before it can fire. The fact that uncontrollable transitions cannot be prevented from firing by the supervisory policy is captured by the requirement that \( \forall m \in N^o, P(m_i) = 1 \), if \( t_i \in T_u \). A string of transitions \( \sigma = t_{j_1}t_{j_2} \cdots t_{j_k} \), where \( t_{j_i} \in T(i \in \{1, 2, \ldots, k\}) \) is said to be a valid firing string starting from the marking \( m \), if,

1) the transition \( t_{j_1} \) is enabled at the marking \( m \), \( P(m_{j_1}) = 1 \), and
2) for \( i \in \{1, 2, \ldots, k-1\} \) the firing of the transition \( t_{j_i} \) produces a marking \( m' \) at which the transition \( t_{j_{i+1}} \) is enabled and \( P(m_{j_{i+1}}) = 1 \).

The set of reachable markings under the supervision of \( P \) in \( N \) from the initial marking \( m^0 \) is denoted by \( R(N, m^0, P) \). A transition \( t_{j_i} \) is live under the supervision of \( P \) if

\[
\forall m \in R(N, m^0, P), \exists m \in R(N, m, P) \text{ such that } t_{j_i} \in T_e(m) \text{ and } P(m_{j_i}) = 1.
\]

A supervisory policy \( P \) enforces liveness if all transitions in \( N \) are live under \( P \).

III. REVIEW OF PREVIOUS RESULTS

Reference [9] contains a necessary and sufficient condition for the existence of a supervisory policy that enforces liveness in an arbitrary PN. This condition is undecidable in general, but if all the transitions in a PN are controllable, or, if the PN is bounded the above mentioned test is decidable. Furthermore, it is possible to synthesize a least-restrictive policy that enforces liveness for these special cases (cf. figure 10, [9]). The computational cost of testing the existence, and synthesizing the least restrictive policy that enforces liveness can be prohibitive in general. Reference [8] contains an interpretation of Commoner’s Liveness Theorem in the context of supervisory policies that enforce liveness in completely controlled FCPNs.

Theorem 4.1 in reference [11] states that if there is a supervisory policy that enforces liveness in a partially controlled FCPN \( N = (\Pi, T, \Phi, m^0) \), then there is a supervisory policy
that enforces liveness in the FCPN \( \bar{N} = (\Pi, T, \Phi, \bar{m}^0) \), where \( \bar{m}^0 \geq m^0 \). This means the set of initial markings for which there is a policy that enforces liveness is right-closed.

Every right-closed set can be represented by a finite-set of minimal elements, and reference [10] presents a minimally restrictive supervisory policy that enforces liveness in a partially controlled FCPN, provided the minimal elements of the above mentioned right-closed set are readily available, and the initial marking is greater than or equal to some minimal element. The issue of deciding if the right-closed set is non-empty for an arbitrary, partially controlled FCPN is still open to our knowledge. That is, given an arbitrary partially controllable FCPN, we cannot tell if there is a supervisory policy that enforces liveness.

Reference [3] concerns the issue of enforcing liveness in a subset of transitions using supervisory control that uses the linear-algebraic approach in reference [4]. The procedure presented in this reference is not guaranteed to halt in all instances, but when it does halt, it presents a supervisory policy that enforces liveness in the original PN. In a sense, one could say that the procedure for liveness enforcement in this reference implicitly identifies a class of PNs for which there is a supervisory policy that enforces liveness. Reference [5] identifies the class of \( S^2PG\Phi \)-nets as a family of PNs that can be made live via supervision under appropriate conditions. Reference [2] presents a monitor-based supervisory control procedure that enforces liveness in bounded PNs under appropriate conditions. In the spirit of these references, the present paper also identifies a family of PNs that can be made live via supervision.

IV. MAIN RESULTS

Observation 4.1 notes that there is a minimally restrictive supervisory policy that enforces liveness in a partially controlled PN if and only if there is a supervisory policy that enforces liveness in the PN.

Observation 4.1: (Theorem 6.1, [9]) There is a minimally restrictive supervisory policy that enforces liveness in a partially controlled PN \( N \) if and only if there is a policy that enforces liveness in \( N \).

Observation 4.2 notes that a minimally restrictive supervisory policy that enforces liveness in a completely controlled PN \( N = (\Pi, T, \Phi, m^0) \) does not disable a select set of transitions.

Observation 4.2: A minimally restrictive supervisory policy \( \mathcal{P} : \mathcal{N}^n \rightarrow \{0, 1\}^m \) that enforces liveness in a fully controlled PN \( N = (\Pi, T, \Phi, m^0) \) will never disable any transitions \( t \in T \), such that \( \{t\} = \emptyset \).

Proof: From Theorem 5.1 of reference [9], we know that if the minimally restrictive supervisory policy prevents the firing of a transition \( t \in T \) at marking \( m^1 \), and if \( m^1 \rightarrow t \rightarrow m^3 \), then at the marking \( m^2 \) there cannot be \( \omega_1, \omega_2 \in T^* \) such that (i) \( m^2 \rightarrow \omega_1 \rightarrow m^3 \rightarrow \omega_2 \rightarrow m^1 \), (ii) \( m^4 \geq m^3 \), and (iii) all transitions in \( T \) appear at least once in \( \omega_2 \).

Consider any firing string \( \omega_4 \in T^* \) that is valid under the supervision of \( \mathcal{P} \) at \( m^1 \). We infer that \( \exists \omega_5 \in \text{pr}(\omega_4), \exists t \in T \), such that \( m^1 \rightarrow \omega_5 \rightarrow m^5 \), \( m^5 \rightarrow \omega_5 \rightarrow m^1 \), \( \omega_5 \notin \text{pr}(\omega_4) \) but \( \bar{t} \notin T_c(N, m^6) \), where \( \text{pr}(\bullet) \) denotes the prefix-set of the string argument. This would then imply that \( \{t\} \neq \emptyset \) (if \( \bar{t} \neq t \), or, \( \{t, \bar{t}\} \subseteq \{t\}^* \). This is because if no such \( \bar{t} \neq t \) exists, then all firing strings that are valid at \( m^1 \) are also valid at \( m^2 \), and the minimally restrictive supervisory policy should have permitted the firing of \( t \) at \( m^1 \). Hence the observation.

While the existence of a supervisory policy that enforces liveness in an arbitrary, partially controlled PN is undecidable [9], there are families of PNs for which it is possible to decide if there is a supervisory policy that enforces liveness. We now present one such family of PNs, called choice-controlled PNs. A PN \( N = (\Pi, T, \Phi, m^0) \) is said to be choice-controlled if \( \forall p \in \Pi, \text{card}(p^*) > 1 \Rightarrow p^* \subseteq T_c \).

We now present one of the main results of the paper.

Theorem 4.3: The existence of a supervisory policy that enforces liveness in a choice-controlled PN is decidable.

Proof: There is a supervisory policy that enforces liveness in a choice-controlled PN if and only if there is a similar policy for the fully-controlled version of the same PN.

The “only if” part of the claim follows directly from the fact that any supervisory policy that enforces liveness in the choice-controlled PN also enforces liveness in the fully-controlled version of the same PN.

For the “if” part, we note that the minimally restrictive supervisory policy \( \mathcal{P} \) that enforces liveness in the fully-controlled version, also enforces liveness in the original, choice-controlled PN. This is established by noting that a firing string is valid under the supervision of \( \mathcal{P} \) in the fully-controlled version, if and only if it is valid under the supervision of \( \mathcal{P} \) in the partially controlled PN. This is established by an induction argument on the length of the firing string that uses the fact that any control-disabled transition in the fully-controlled version of the PN is a controllable transition in the original, choice-controlled PN.

Since the existence of a supervisory policy that enforces liveness in fully controlled PNs is decidable [9], the result follows.

Figure 10 of reference [9] presents a procedure for the synthesis of a minimally restrictive supervisory policy that enforces liveness in a fully controlled PN. If there is a supervisory policy that enforces liveness in a choice-controlled PN, the minimally restrictive supervisory policy for its fully-controlled equivalent is also a minimally restrictive policy for the choice-controlled PN.

The requirement of choice-controlled PNs: \( \forall p \in \Pi \), \( \text{card}(p^*) > 1 \Rightarrow p^* \subseteq T_c \), can be relaxed. For instance, if \( p \in \Pi \) in a PN \( N = (\Pi, T, \Phi, m^0) \), such that \( \text{card}(p^*) > 1 \)
and
\[ \forall m^1 \in \mathbb{R}(N,m^0), (m^1(p) \geq 1) \Rightarrow (\text{card}(T \setminus (m^1,N)\cap P^*)) \leq 1 \] (1)
then we can relax the requirement that \( P \subseteq T_c \). The proof of observation 4.2, with appropriate changes, serves as a proof that a minimally restrictive supervisory policy that enforces liveness in \( N \) will not disable any output transition of \( P \).

It should be noted that the requirement in equation 1 can be tested in finite-time using the coverability graph (cf. page 930, [9]) of \( N \). Using a proof similar to that of theorem 4.3, it can be shown that the existence of a supervisory policy for this relaxed family of choice-controlled PNs is also decidable.

The above result can be used to show that the fully uncontrollable PN shown in figure 1 is live. In graphical representations of controllable PNs a controllable (uncontrollable) transition is represented by a filled (unfilled) box. The PN shown in figure 1 contains no uncontrollable transitions. We first note that \( \text{card}(p_1^*) = \text{card}(p_2^*) = 2 \), and \( \text{card}(p_2^*) = 1 \). Also, from the structure of the PN we know that the condition in equation 1 is holds for \( p_1 \) and \( p_2 \). So, from applying theorem 4.3 to the relaxed family defined above, we know that there is a supervisory policy that enforces liveness in this fully uncontrolled PN if and only if there is a supervisory policy that enforces liveness in the fully-controlled equivalent. Since

\[
\begin{pmatrix}
1 \\
0 \\
1 \\
0 \\
0 \\
1 \\
0 \\
0 \\
1 \\
0
\end{pmatrix}
\rightarrow t_2t_4t_8t_5t_3t_6t_2t_4t_8t_5t_3t_6
\]

from theorem 5.1 of reference [9] we infer there is a supervisory policy that enforces liveness in the fully-controlled version of this PN shown in figure 1. From the adaptation of theorem 4.3 to the relaxed family of choice-controlled PNs, we infer that there is a supervisory policy that enforces liveness in the PN shown in figure 1. But since all transitions in this PN are uncontrollable, and we know that there is a supervisory policy that enforces liveness, we conclude that the PN shown in this figure is live.

We now turn our attention to a necessary and sufficient condition for the existence of a supervisory policy that enforces liveness in partially controlled FCPNs. If \( \mathcal{M}(m,t) \) denotes the collection of markings reachable from \( m \) under the supervision of \( P \), where \( t \) is state-enabled, reference [8] contains the following characterization of supervisory policies that enforce liveness in a completely controlled FCPN.

**Theorem 4.4**: [8] A supervisory policy \( P : N^0 \rightarrow \{0,1\}^m \), enforces liveness in an FCPN \( N = (\Pi,T,\Phi,m^0) \) if and only if the following conditions are satisfied:

1. \( \forall m^1 \in \mathbb{R}(N,m^0,P), \forall t \in T, \mathcal{M}(m,t) \neq \emptyset \Rightarrow \exists m \in \mathcal{M}(m,t) \cup P^*) \)
2. \( \forall m \in \mathbb{R}(N,m^0,P), \forall P \subseteq \Pi, \sum_{p \in P} m(p) \neq 0 \)

The proof of the above result in reference [8] holds, mutatis mutandis, for partially controlled FCPNs. We refrain from presenting the proof again in the interest of space. Condition 1 in the above theorem requires that the supervisory policy \( P \) enforces liveness in an arbitrary, partially controlled FCPN shown in figure 2 has two minimal siphons \( \{p_1,p_2,p_3,p_4\} \), there are no traps in this FCPN. Since \( p_1 \) is present in both siphons, it follows that the supervisory policy does not permit the emptying of siphons. Also, if the supervisory policy prevents the emptying of siphons, condition 2 of theorem 4.4 can be used to show that specific examples of fully controlled FCPNs that illustrate the need for the simultaneous satisfaction of the two conditions. These examples are also relevant to the case when the FCPN is partially controllable.

**Theorem 4.4** can be used to show that specific examples of partially controlled FCPNs can be made live via supervision. For instance, consider the partially controlled FCPN shown in figure 2. The (controllable) uncontrollable transitions are represented by (filled) unfilled boxes (i.e. \( T_c = \{t_1\}, T_u = \{t_2,t_3,t_4,t_5,t_6,t_7\} \)). Theorem 4.4 can be used to show that the supervisory policy that permits the firing of \( t_1 \) when the two are more tokens in \( p_1 \) ensures liveness. The FCPN shown in figure 2 has two minimal siphons \( P_1 = \{p_1,p_2,p_3\} \). There are no traps in this FCPN. Since \( p_1 \) is present in both siphons, it follows that the supervisory policy does not permit the emptying of siphons. Also, if the supervisory policy prevents the emptying of \( t_1 \) there must be only one token in \( p_1 \). The firing of the string \( t_2t_4t_8t_5 \) at this marking will result in a new marking where \( t_1 \) is both state- and control-enabled. From theorem 4.4 we can conclude that the policy enforce liveness in the partially controlled FCPN shown in figure 2 (which is not choice-controllable).

**V. Conclusions**

It is not possible to test the existence of a supervisory policy that enforces liveness in an arbitrary, partially controlled PN (cf. theorem 5.3 and corollary 5.2, [9]). So, in this paper we are after families of partially controlled Petri nets (PNs) for which we can decide if there is a supervisory policy that enforces liveness. We identified the family of choice-controlled PNs as one such family of partially controlled PNs. We then relaxed the requirements of choice-controlled PNs to obtain a relaxed family of choice-controlled PNs for
which it is possible to decide if a member of this class can be made live by supervision.

Finally, we turn to the problem of testing if a specific instance of a partially controlled PN can be made live by supervision. Restricting attention to the class of partially controlled Free-Choice PNs (FCPNs), we derive a necessary and sufficient condition for supervisory policies that enforce liveness. This condition finds use in situations where we are required to show that a specific supervisory policy enforces liveness in a partially controlled FCPN.

![Fig. 1. A partially-controlled PN that is a member of the relaxed family of choice-controlled PNs.](image)

![Fig. 2. A partially-controlled FCPN.](image)

VI. REFERENCES