On Supervisory Policies That Enforce Liveness in a Class of Completely Controlled Petri Nets Obtained via Refinement

Ramavarpur S. Sreenivas

Abstract—The authors consider Petri nets (PN's) [3], where each transition can be prevented from firing by an external agent, the supervisor. References [5] and [6] contain necessary and sufficient conditions for the existence of a supervisory policy that enforces liveness in a PN that is not live. A PN is said to be live if it is possible to fire any transition from every reachable marking, although not necessarily immediately. The procedure in [5] and [6] involves the construction of the coverability graph (cf. [3, Sec. 5.1]), which can be computationally expensive. Using the refinement/abstraction procedure of Suzuki and Murata [8], where a single transition in a abstracted PN \( \mathcal{N} \) is replaced by a PN \( \mathcal{N} \) to yield a larger refined PN \( \mathcal{N} \), we show that when \( \mathcal{N} \) belongs to a class of marked-graph PN’s (cf. [3, Sec. 6.1]), there is a supervisory policy that enforces liveness in the refined PN \( \mathcal{N} \) if and only if there is a similar policy for the abstracted PN \( \mathcal{N} \). Since the coverability graph of the PN \( \mathcal{N} \) is smaller than that of the PN \( \mathcal{N} \), it is possible to achieve significant computational savings by using the process of abstraction on \( \mathcal{N} \). This is illustrated by example.

Index Terms—DEDS, liveness, Petri nets.

I. INTRODUCTION

Petri Nets (PN’s) [3] are popular tools for the modeling, analysis, control, and performance evaluation of large-scale discrete-event dynamic systems (DEDS). In this paper we concern ourselves with the property of liveness, a stronger version of the absence of deadlocks. A PN is said to be live (cf. [3, Sec. 4.3]) if it is possible to fire any transition from every reachable marking, although not necessarily immediately. In PN’s where each transition can be individually controlled by an external agent, the supervisors are called completely controlled PN’s (CCPN’s) [5]. References [5] and [6] present necessary and sufficient conditions for the existence of supervisory policies that enforce liveness in an arbitrary CCPN. The test procedure for these conditions involves the construction of the coverability graph (cf. [3, Sec. 5.1]) of a PN. The size of the coverability graph of a PN can be exponentially related to the number of places and transitions. Often this can be computationally burdensome. In this paper we use the refinement/abstraction procedure of Suzuki and Murata [8] to alleviate this computational burden. The refinement procedure of Suzuki and Murata involves replacing a single transition in a PN \( \mathcal{N} \) by a PN \( \mathcal{N} \), resulting in a PN \( \mathcal{N} \). The abstraction procedure is essentially the reversal of this process. In this paper we show that when \( \mathcal{N} \) is a live, marked-graph PN (MGPN) with some additional restrictions, there exists a supervisory policy that enforces liveness in the refined PN \( \mathcal{N} \) if and only if there exists a policy that enforces liveness in the abstracted PN \( \mathcal{N} \). Since the coverability graph of the abstracted PN \( \mathcal{N} \) is smaller that of the refined PN \( \mathcal{N} \), significant computational savings can be obtained. The extent of savings is illustrated via an example.

The paper is organized as follows: Section II introduces the notational preliminaries, Section III presents the main results, and finally in Section IV we present our conclusions along with recommendations for future research.

II. NOTATIONAL PRELIMINARIES AND REVIEW OF PRIOR WORK

A PN \( \mathcal{N} = (\Pi, T, \Phi, \mathbf{m}_0) \) is an ordered 4-tuple, where \( \Pi = \{p_1, p_2, \ldots, p_n\} \) is a set of \( n \) places, \( T = \{t_1, t_2, \ldots, t_m\} \) is a set of \( m \) transitions, \( \Phi \subseteq (\Pi \times T) \cup (T \times \Pi) \) is a set of arcs, \( \mathbf{m}_0 : \Pi \rightarrow \mathcal{N} \) is the initial-marking function (or the initial-marking), and \( \mathcal{N} \) is the set of nonnegative integers. The marking of a PN, \( \mathbf{m} : \Pi \rightarrow \mathcal{N} \), identifies the number of tokens in each place. For a given marking \( \mathbf{m} \) a transition \( t \in T \) is said to be enabled if \( \forall \mathbf{p} \in \Phi, \mathbf{m}(t) \geq 1 \), where \( \mathbf{p} := \{y | (y, x) \in \Phi \} \). For a given marking \( \mathbf{m} \) the set of enabled transitions is denoted by the symbol \( T_e(\mathcal{N}, \mathbf{m}) \). An enabled transition \( t \in T_e(\mathcal{N}, \mathbf{m}) \) can fire, which changes the marking \( \mathbf{m} \) to \( \mathbf{m}' \) according to the equation

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

where the symbol \( \text{card}(\bullet) \) is used to denote the cardinality of the set argument, and \( \mathbf{p} \). A string of transitions \( \sigma = t_1t_2 \cdots t_k \), where \( t_i \in T (i \in \{1, 2, \ldots, k\}) \) is said to be a valid firing string at the marking \( \mathbf{m} \), if 1) the transition \( t_1 \) is enabled at the marking \( \mathbf{m} \) and 2) for \( i \in \{1, 2, \ldots, k - 1\} \) the firing of the transition \( t_i \) produces a marking at which the transition \( t_{i+1} \) is enabled. Given an initial marking \( \mathbf{m}_0 \) the set of reachable markings for \( \mathbf{m}_0 \) denoted by \( \mathcal{R}(\mathcal{N}, \mathbf{m}_0) \), is the set of markings generated by all valid firing strings at the initial-marking \( \mathbf{m}_0 \) in the PN \( \mathcal{N} \). At a marking \( \mathbf{m}_1 \), if the firing of a valid firing string \( \sigma \) results in a marking \( \mathbf{m}_2 \), we represent it as \( \mathbf{m}_1 \rightarrow \mathbf{m}_2 \). A transition \( t \in T \) is live if \( \forall \mathbf{m}_1 \in \mathcal{R}(\mathcal{N}, \mathbf{m}_0), \exists \mathbf{m}_2 \in \mathcal{R}(\mathcal{N}, \mathbf{m}_1) \) such that \( t \in T_e(\mathcal{N}, \mathbf{m}_2) \). The PN \( \mathcal{N} \) is live if every transition \( t \in T \) is live. For any valid firing string \( \sigma \in T_e \), we use the symbol \#(\sigma, t) to denote the number of occurrences of the transition \( t \) in \( \sigma \), and the symbol \( |\sigma| \) to denote the length of the string \( \sigma \).

A PN \( \mathcal{N} = (\Pi, T, \Phi, \mathbf{m}_0) \) is said to be an MGPN, if \( \forall \mathbf{p} \in \Pi, \text{card}(\mathbf{p}) = \text{card}(\mathbf{p}) = 1 \). That is, in an MGPN every place has a unique input (output) transition. For a pair of transitions \( t_i, t_j \in T \), a path from \( t_i \) to \( t_j \) is a string of alternating transitions and places, \( p_1, p_2, p_3, \ldots, p_k, t_{k+1}, \ldots, p_{k+1}, t_{k+2}, \ldots, p_k, t_{k+3}, \ldots, p_k, t_{k+4}, \ldots, p_k, t_{k+5}, \ldots \) \( \subseteq \Phi \). The path \( P \) is said to be a simple path if every path from \( t_i \) to \( t_j \) contains \( P \) as a suffix. There is a closed-path that contains \( t_i \) if 1) \( P \) is not the null-string and 2) \( t_j = t_i \). The set of closed-paths in an MGPN is defined accordingly.

An MGPN is live if and only if every closed-path is marked at the initial-marking (cf. [3, Th. 6.5]). That is, the set of places in each closed path has a nonzero token-load at the initial-marking. In a live MGPN \( \mathcal{N} = (\Pi, T, \Phi, \mathbf{m}_0) \), the firing of a transition \( t_i \) is necessary for firing transition \( t_j \) if and only if there is a token-free path from \( t_i \) to \( t_j \) (cf. [4, Property 3]).

A CCPN [6] is expressed as an ordered 6-tuple: \( M = (\Pi, T, \Phi, \mathbf{m}_0, C, B) \), where \( \Pi = \{p_1, p_2, \ldots, p_n\} \) is a set of \( n \) state-places; \( T = \{t_1, t_2, \ldots, t_m\} \) is a set of \( m \) transitions, \( \Phi \subseteq (\Pi \times T) \cup (T \times \Pi) \) is a set of state-arcs; \( C = \{c_1, c_2, \ldots, c_m\} \) is the set of control-places; \( B = \{c, t_i \} \mid i = 1, 2, \ldots, m \) is the set of control-arcs; \( \mathbf{m}_0 : \Pi \rightarrow \mathcal{N} \) is the initial-marking function

\footnote{Note that \( \text{card}(C) = \text{card}(T) = m \).}
(or the initial-marking); and $N$ is the set of nonnegative integers. The CCPN $M = (\Pi, T, \Phi, m^0, C, B)$ contains the underlying PN $N = (\Pi, T, \Phi, m^0)$. As there is one control place assigned to each transition the underlying PN uniquely determines the CCPN. Therefore, in graphical representations of CCPN’s we do not explicitly represent the control-places.

A control $u : C \rightarrow \{0, 1\}$ assigns a token load of 0 or 1 to each control place. The control can also be interpreted as an $m$-dimensional binary vector $u \in \{0, 1\}^m$. It would help to view the control $u$ as follows: if the $i$th component of $u$, or $u(c_i)$, is 0 (1), then transition $t_i$, is control-disabled (control-enabled). For a given marking $m$ (control $u$), a transition $t_i \in T$ is said to be state-enabled (control-enabled) if $t_i \in T_r(N, m)$ (if $u(c_i) = 1$). A transition that is control-enabled and state-enabled can fire resulting in the marking given by (1). A supervisory policy $P : N^* \rightarrow \{0, 1\}^m$ is a partial map that assigns a control for each reachable marking and is possibly undefined for the unreachable markings.

For a given CCPN and supervisory policy $P$, a string of transitions $s = t_1 t_2 \cdots t_k$, where $t_i \in T$ ($i \in \{1, 2, \ldots, k\}$) is said to be a valid firing string under supervision at the marking $m^0$ that, if 1) the transition $t_{j_1}$ is state-enabled at the marking $m^0$, $P(m^0)_{j_1} = 1$ and 2) for $i \in \{1, 2, \ldots, k - 1\}$ the firing of the transition $t_{j_{i+1}}$ produces a marking $m'$ at which the transition $t_{j_{i+1}}$ is state-enabled and $P(m')_{j_{i+1}} = 1$. For a given supervisory policy $P$, the set of reachable markings under supervision for a CCPN $M$ with initial-marking $m^0$, denoted by $\mathcal{R}(M, m^0, P)$, is the set of markings generated by all valid firing strings under supervision at the marking $m^0$ in the CCPN $M$. For the CCPN, a transition $t_{j_i} \in T$ is live under $P$ if for all markings $m^0 = (\Pi, T, \Phi, m^0) \ni m^0, \exists m^1 \in \mathcal{R}(M, m^0, P)$, there is a transition $t'_{j_i}$ such that $t'_{j_i} \in T_r(N, m^1)$ and $P(m^1)_{j_i} = 1$. A supervisory policy $P$ enforces liveness in a CCPN $M$ if all transitions in $M$ are live under $P$.

References [5] and [6] contain a test for the existence of a supervisory policy that enforces liveness in a CCPN. The procedure involves testing the nonemptiness of a real-valued feasible region defined by linear inequalities. This procedure has a time complexity that is polynomially related to the number of variables, which is equal to the number of vertices in the coverability graph (cf. [3, Sec. 5.1]) of the underlying PN of the CCPN. However, the number of vertices in the coverability graph of a PN can be exponentially related to its size. In the next section we use the refinement/abstraction technique of Suzuki and Murata [8] to alleviate the computational burden of this procedure.

III. MAIN RESULTS

Let $N = (\Pi, T, \Phi, m^0)$ be a PN and $t_0 \in T$ be a transition in $N$. Also, let $\tilde{N} = (\Pi, \tilde{T}, \tilde{\Phi}, \tilde{m}^0)$ be a different (i.e., $\Pi \cap \Pi = T \cap \tilde{T} = \emptyset$) PN where $\{t_0, \tilde{t}_{out}\} \subseteq \tilde{T}$ are a pair of transitions in $\tilde{N}$. We now describe the refinement/abstraction technique of Suzuki and Murata [8]. The refined PN $\tilde{N} = (\Pi, \tilde{T}, \tilde{\Phi}, \tilde{m}^0)$ is obtained by replacing the transition $t_0$ in the PN $N$ by the PN $\tilde{N}$ as follows:

\[
\begin{align*}
\tilde{T} &= T \cup \tilde{T} \\
\tilde{\Phi} &= (\Phi \cup \tilde{\Phi}) - (\{t_0\} \times \Pi) - (\Pi \times \{t_0\}) \\
&\cup \{p, \tilde{t}_{out}\} \ni \tilde{\Phi} \cup \{\bar{v}, \tilde{t}_{out}\} \ni \tilde{\Phi}
\end{align*}
\]

\[\tilde{m}^0(\tilde{p}) = \begin{cases} 
\tilde{m}^0(p), & \text{if } \tilde{p} \in \Pi. \\
\tilde{m}^0(p), & \text{if } \bar{p} \in \Pi. 
\end{cases}\]

Conversely, the PN $N$ can be abstracted from $\tilde{N}$ by reversing the process of refinement. That is, in the refined PN $\tilde{N}$ the subnet defined by the PN $\tilde{N}$ is replaced by the transition $t_0$ which results in the abstracted PN $\tilde{N}$. Throughout this paper we will use the symbol $N(\tilde{N})$ to denote the abstracted (refined) PN. Fig. 1(a) contains the PN $\tilde{N}$ obtained by using the above construction on the PN’s $N$ and $\tilde{N}$ shown in Fig. 1(a) and (b) respectively. Suzuki and Murata derive sufficient (but not necessary) conditions under which the liveness of $N$ and $\tilde{N}$ imply the liveness of $\tilde{N}$ (cf. [8, Th. 11]).

In the remainder we concern ourselves with the existence of supervisory policies that enforce liveness in the CCPN $M$ that has $\tilde{N}$ as its underlying PN. The PN $\tilde{N}$ is assumed to be obtained by refining a transition $t_0$ in a PN $N$ by the subnet represented by a PN $\tilde{N}$. We show that when $\tilde{N}$ is a live MGPN, with an empty, simple path originating from $\tilde{t}_{out}$ to $t_{out}$, there is a policy that enforces liveness in the CCPN $M$ if and only if there is a policy that enforces liveness in $\tilde{M}$. This yields a “divide-and-conquer” procedure to testing the existence of supervisory policies that enforce liveness in large CCPN’s. The benefit to this approach is that testing the existence of a supervisory policy that enforces liveness in a CCPN $M$ can be significantly easier than the corresponding test for the CCPN $\tilde{M}$. Although the main result (cf. Theorem 3.1) is stated in terms of the PN $\tilde{N}$ being obtained from the PN’s $N$ and $\tilde{N}$ via the process of refinement, the applicability of this result to the efficient synthesis of supervisory policies for an arbitrary CCPN $M$ relies on the abstraction of the (possibly simpler) PN $N$ from (possibly complicated) PN $\tilde{N}$.

The results of [5] and [6] can be applied to the CCPN $M$, and the supervisory policy that enforces liveness in $M$ can be used to synthesize a policy that enforces liveness in $\tilde{M}$. We now state our main result, the proof of which follows from Lemmas 3.5 and 3.6. Lemma 3.5 presents a prescription for the synthesis of a supervisory policy that enforces liveness in the CCPN $M$ from a corresponding policy for the CCPN $\tilde{M}$. Lemma 3.6 establishes the fact that the existence of a supervisory policy that enforces liveness in the CCPN $\tilde{M}$ implies the existence of a similar policy for the CCPN $M$.

Theorem 3.1: Let $\tilde{N} = (\Pi, \tilde{T}, \tilde{\Phi}, \tilde{m}^0)$ be a live MGPN such that for a pair of distinct transitions $\{t_{in}, \tilde{t}_{out}\} \subseteq \tilde{T}$, there is a simple path from $t_{in}$ to $\tilde{t}_{out}$ that is empty at the initial-marking $\tilde{m}^0$. For an arbitrary PN $N = (\Pi, T, \Phi, m^0)$ with a distinct transition $t_0 \in T$, $(\Pi \cap \Pi = T \cap \Pi = \emptyset)$, let $\tilde{N} = (\Pi, \tilde{T}, \tilde{\Phi}, \tilde{m}^0)$ be the PN obtained by refining the transition $t_0$ by the PN $\tilde{N}$ as illustrated above. If $\tilde{M}(M)$ is a CCPN with the underlying PN $\tilde{N}(N)$, then there is a supervisory policy that enforces liveness in the CCPN $M$ if and only if there exists a supervisory policy that enforces liveness in the CCPN $\tilde{M}$.

We note that the PN $\tilde{N}$ in Fig. 1(b) is a live MGPN as all closed-paths are marked. Additionally, every path originating from $t_{in}$ to $\tilde{t}_{out}$ will have the path $P = t_{in}t_{out}t_{out}$ as a suffix. So, $P$ is a simple path from $t_{in}$ to $\tilde{t}_{out}$, which is empty at the initial-marking. The PN $N$ shown in Fig. 1(a) is not live, and neither is the PN $\tilde{N}$ shown in Fig. 1(c). Let $\tilde{M}(M)$ be the CCPN that has the PN shown in Fig. 1(c) [Fig. 1(a)] as its underlying PN. According to Theorem 3.1, there is a supervisory policy that enforces liveness, the CCPN $\tilde{M}$, if and only if there is a supervisory policy that enforces liveness in the CCPN $M$. As a part of the Proof of Lemma 3.5 we show that a supervisory policy that enforces liveness in $\tilde{M}$ can be readily converted into a policy that enforces liveness in $\tilde{M}$. We now derive a collection of results that are critical to the Proof of Lemmas 3.5 and 3.6, which together establish Theorem 3.1.

For the class of MGPN’s defined above, Lemma 3.1 establishes:

1) the number of occurrences of transition $\tilde{t}_{out}$ never exceeds that of transition $t_{in}$ in any valid firing string at the initial-marking and
2) at any point in the evolution of the tokens in $N$, the number of occurrences of transition $t_{out}$ can be made equal to that of $t_{in}$, without firing $t_{in}$. Due to the limitations of space we only present a sketch of
the proofs of the various lemmas in this paper. The detailed proofs of the various lemmas in this paper can be found in [7].

Lemma 3.1: Let \( \tilde{\ell}_{\text{in}}, \tilde{\ell}_{\text{out}} \) be two distinct transitions in a live, MGPN \( \tilde{N} = (\tilde{I}, \tilde{T}, \tilde{\Phi}, \tilde{m}^0) \), such that at the initial-marking \( \tilde{m}^0 \) there is an empty, simple path originating from \( \tilde{\ell}_{\text{in}} \) to \( \tilde{\ell}_{\text{out}} \), then for any valid firing string \( \tilde{\sigma} \in \tilde{T}^* \) at the initial-marking \( \tilde{m}^0 \):

1) \( \#(\tilde{\sigma}, \tilde{\ell}_{\text{in}}) \geq \#(\tilde{\sigma}, \tilde{\ell}_{\text{out}}) \);
2) if \( \#(\tilde{\sigma}, \tilde{\ell}_{\text{in}}) > \#(\tilde{\sigma}, \tilde{\ell}_{\text{out}}) \), then \( \exists \tilde{\sigma}_1 \in (\tilde{T} - \{\tilde{\ell}_{\text{in}}\})^* \) such that 1) \( \tilde{\sigma} \tilde{\sigma}_1 \) is a valid firing string at the initial-marking \( \tilde{m}_0 \) and 2) \( \#(\tilde{\sigma} \tilde{\sigma}_1, \tilde{\ell}_{\text{in}}) = \#(\tilde{\sigma}, \tilde{\ell}_{\text{in}}) \).

The first part of this lemma is established by first observing that the simple path from \( \tilde{\ell}_{\text{in}} \) to \( \tilde{\ell}_{\text{out}} \) is empty at the initial-marking. Additionally, the number of tokens in the simple path from \( \tilde{\ell}_{\text{in}} \) to \( \tilde{\ell}_{\text{out}} \) equals the value of \( \#(\tilde{\sigma}, \tilde{\ell}_{\text{in}}) - \#(\tilde{\sigma}, \tilde{\ell}_{\text{out}}) \). This fact is then used to establish the second part of the lemma, where we note from the definition of a simple path that any path from \( \tilde{\ell}_{\text{in}} \) to \( \tilde{\ell}_{\text{out}} \) contains the simple path from \( \tilde{\ell}_{\text{in}} \) to \( \tilde{\ell}_{\text{out}} \) as a suffix. Therefore, there can be no token-free directed paths from \( \tilde{\ell}_{\text{in}} \) to \( \tilde{\ell}_{\text{out}} \). Since \( \tilde{N} \) is live, we infer \( \tilde{\ell}_{\text{out}} \) can fire without the firing of \( \tilde{\ell}_{\text{in}} \) ([4, Property 3]). Repeating this argument \( \#(\tilde{\sigma}, \tilde{\ell}_{\text{in}}) - \#(\tilde{\sigma}, \tilde{\ell}_{\text{out}}) \) times, we establish the existence of \( \tilde{\sigma}_1 \in (\tilde{T} - \{\tilde{\ell}_{\text{in}}\})^* \), such that \( \tilde{\sigma} \tilde{\sigma}_1 \) is a valid firing string at the initial-marking \( \tilde{m}_0 \) in \( \tilde{N} \) and \( \#(\tilde{\sigma} \tilde{\sigma}_1, \tilde{\ell}_{\text{in}}) = \#(\tilde{\sigma}, \tilde{\ell}_{\text{in}}) = \#(\tilde{\sigma}, \tilde{\ell}_{\text{out}}) \).

Formal details of the proof of this lemma can be found in [7].

Following Suzuki and Murata [8] we define functions \( f : \tilde{T}^* \rightarrow \tilde{T}^* \) and \( \tilde{f} : \tilde{T}^* \rightarrow \tilde{T}^* \) as follows:

\[
\begin{align*}
    f(\lambda) & = \lambda \\
    f(\tilde{i}) & = \begin{cases} 
        \tilde{i} & \text{if } \tilde{i} \in \tilde{T} - \{\tilde{\ell}_{\text{in}}\}, \\
        \tilde{0} & \text{if } \tilde{i} = \tilde{\ell}_{\text{in}}, \\
        \tilde{i} & \text{if } \tilde{i} \in \tilde{T}.
    \end{cases}
\end{align*}
\]

and

\[
\begin{align*}
    \tilde{f}(\lambda) & = \lambda \\
    \tilde{f}(\tilde{i}) & = \begin{cases} 
        \tilde{i} & \text{if } \tilde{i} \in \tilde{T}, \\
        \lambda & \text{otherwise}.
    \end{cases}
\end{align*}
\]

where \( \lambda \) is the null-string. The function \( f(\bullet) \) converts a firing string in \( \tilde{N} \) to a firing string in \( N(\tilde{N}) \).

Let \( \mathcal{P} : \mathcal{N}^{\text{card}(T)} \rightarrow \{0, 1\}^{\text{card}(T)} \) be a supervisory policy for the CCPN \( M \). We define a supervisory policy \( \tilde{\mathcal{P}} : \mathcal{N}^{\text{card}(T)} \rightarrow \{0, 1\}^{\text{card}(\tilde{T})} \) for the CCPN \( \tilde{M} \) as follows:

\[
\tilde{\mathcal{P}}(\tilde{m}) = \begin{cases} 
    \mathcal{P}(\Delta(\tilde{m})) & \text{if } \tilde{i} \in \tilde{T}, \\
    \mathcal{P}(\Delta(\tilde{m}))_{\tilde{i}} & \text{if } \tilde{i} = \tilde{\ell}_{\text{in}}, \\
    1 & \text{otherwise}.
\end{cases}
\]
where $\Delta : \mathcal{N}^{\text{mrd}(\Pi)} \rightarrow \mathcal{N}^{\text{mrd}(\Pi)}$ is defined as follows: $\forall p \in \Pi$,

$$\Delta(m)(p) = \begin{cases} m(p) + \sum_{\sigma \in \Pi} \delta_{p \in \Pi}(f_{m \rightarrow m}(p)), & \text{if } p \in t_0^*, \\ m(p), & \text{otherwise}, \end{cases}$$

and $P_{(t_{in}, t_{out})}$ denotes the set of places in the simple path from $t_{in}$ to $t_{out}$. Lemma 3.2 establishes the relationship from the marking resulting from the firing of appropriate firing strings in $M$, $M_0$, and $N$.

**Lemma 3.2:** Let $\hat{\sigma}$ be a firing string that is valid under the supervision of $\hat{P}$ at the initial-marking $m_0^0$ in $M$. Also, let $f(\hat{\sigma})$ be a firing string that is valid under the supervision of $P$ at the initial-marking $m_1^0$ in $M$, and $\hat{f}(\hat{\sigma})$ be a valid firing string in the PN $N$ at the initial-marking $m_0^0$. If $m_1^0 \rightarrow \hat{\sigma} \rightarrow m_1$ in $M$, $m_0^0 \rightarrow f(\hat{\sigma}) \rightarrow m_1$ in $M$, and $m_0^0 \rightarrow \hat{f}(\hat{\sigma}) \rightarrow m_1$ in $N$, then $\forall p \in \Pi$

$$m(p) = \begin{cases} m(p) + \sum_{\sigma \in \Pi} \delta_{p \in \Pi}(f_{m \rightarrow m}(p)), & \text{if } p \in t_0^*, \\ m(p), & \text{otherwise}. \end{cases}$$

In Lemma 3.4 we show that any firing string $\sigma \in T^*$ that is valid under the supervision of $P$ at the initial-marking $m_0^0$ can be effectively simulated by a firing string $\hat{\sigma} \in \hat{T}^*$ that is valid under the supervision of $\hat{P}$ at the initial-marking $m_0^0$.

**Lemma 3.4:** For any firing string $\sigma \in T^*$ that is valid in $M$ under the supervision of $P$ at the initial-marking $m_0^0$. $\exists \hat{\sigma} \in \hat{T}^*$ that is valid in $M$ under the supervision of $P$ at the initial-marking $m_0^0$, such that $f(\hat{\sigma}) = \sigma$.

We now present a sketch of the proof of the above lemma. First we let $\sigma = \sigma_1 \sigma_2 \ldots \sigma_n \sigma_{n+1}$, where $\#(\sigma_1, t_0) = 0$, $\forall i \in \{1, 2, \ldots, n+1\}$. Using an induction argument we show that the $i$th occurrence of $t_0$ in $\sigma$ can be replaced by a string of transitions $\sigma_2 \ldots \sigma_{n+1}$ in $T^*$ such that $\#(\sigma_2 \ldots \sigma_{n+1}, t_{in}) = 1$, $\#(\sigma_{n+1}, t_{out}) = 0$, and the resulting string of transitions, $\sigma_1 \sigma_2 \ldots \sigma_{n+1}$, will be permitted under the supervision of $P$ at the initial-marking $m_0^0$ in $M$. The result follows from the fact that $f(\hat{\sigma}_1 \hat{\sigma}_2 \ldots \hat{\sigma}_{n+1}) = \sigma$. Due to space limitations the formal proof of this lemma is omitted, and it can be found in [7].

**Lemma 3.5:** Establishes the sufficiency of Theorem 3.1, and Lemma 3.6 establishes its necessity.

**Lemma 3.5:** If the supervisory policy $P$ enforces liveness in the CCPN $M$, then the policy $P$ enforces liveness in the CCPN $M$.

**Proof:** Let $\hat{\sigma} \in \hat{T}^*$ be any firing string such that $m_0^0 \rightarrow \hat{\sigma} \rightarrow m_1^0$ under the supervision of $P$ in $M$. We show that any $i \in \hat{T}$ can be fired after the valid firing string $\sigma$.

**Case 1:** $\{i \in T - \{t_0\}\}$. From Lemma 3.3 we know $m_0^0 \rightarrow \hat{f}(\hat{\sigma}) \rightarrow m_1^0$ in $M$ under the supervision of $P$. Since $P$ enforces liveness, $\exists \sigma_i \in T^*$ such that $f(\sigma_i, t_0)$ is valid under the supervision of $P$ at the initial-marking $m_0^0$ in $M$. By Lemma 3.4 we know $\exists \hat{\sigma}_i \in \hat{T}$ such that $f(\hat{\sigma}_i, t_0) = \sigma_i$ and $\dot{\sigma}_i, t_0$ is valid under the supervision of $P$ at the initial-marking $m_0^0$ in $M$. Hence every transition in $T - \{t_0\}$ is live under the supervision of $P$ in $M$.

**Case 2:** $\{i \in T \setminus \{t_{in}\}\}$. Using the same argument as in Case 1, we establish the existence of a firing string $f(\hat{\sigma}_{\{i\}}, t_{in})$ that is valid under the supervision of $P$ at the initial-marking $m_0^0$ in $M$. By Lemma 3.4, we infer the firing string $f(\hat{\sigma}_{\{i\}}, t_{in})$ can be simulated by a firing string $\hat{\sigma}_{\{i\}} \in \hat{T}^*$ in $M$, where $f(\hat{\sigma}_{\{i\}}) = \sigma_{\{i\}}$. From the definition of $f(\bullet)$ we infer $\exists \hat{\sigma}_{\{i\}} \in \hat{T}$ such that $f(\hat{\sigma}_{\{i\}}, t) = \sigma_{\{i\}}$ and $\sigma_{\{i\}} t_{in}$ is valid under the supervision of $P$ at the initial-marking $m_0^0$ in $M$. This implies $i$ is live under the supervision of $P$ in $M$.

**Case 3:** $\{i \in T \setminus \{t_{in}\}\}$. From Lemma 3.3 we know $m_0^0 \rightarrow \hat{f}(\hat{\sigma}) \rightarrow m_1^0$ in the PN $N$. Since $N$ is live, it is possible to fire $i$, although not necessarily immediately, starting at the marking $m_1^0$. We consider two subcases.

**Case 3a:** If there is no token-free path from $t_{in}$ to $i$ at $m_1^0$ in $N$, then since $N$ is a live MGPNN, $\exists \hat{\sigma}_i \in \hat{T}$ such that $\dot{\sigma}_i$ is a valid firing string in the PN $N$ that fires at $t_{in}$ and $\#(\hat{\sigma}_i, t_{in}) = 0$ (cf. [4, Property 3]). From the construction of $N$ and the definition of $P$, we infer the firing string $\sigma_i t_{in}$ is valid under the supervision of $P$ at the initial-marking $m_0^0$.

**Case 3b:** If there is a token-free path from $t_{in}$ to $i$ at $m_1^0$ in $N$, $i$ cannot fire until $i$ has fired once in $N$ (cf. [4, Property 3]). Since $i$ is live under the supervision of $P$ in $M$ (cf. Case 2), $\exists \hat{\sigma}_i \in \hat{T}$ such that $\#(\hat{\sigma}_i, t_{in}) = 1$ and $m_0^0 \rightarrow \dot{\sigma}_i \rightarrow m_1^0$ under the supervision of $P$ in $M$. From Lemma 3.3 we infer $m_0^0 \rightarrow f(\hat{\sigma}_i) \rightarrow m_1^0$ in the PN $N$. Since $f(\hat{\sigma}_i) \rightarrow m_1^0$, all paths from $t_{in}$ to $i$ in the PN $N$ will be nonempty at the marking $m_1^0$. Since $N$ is live, we infer $\exists \hat{\sigma}_i \in \hat{T}$ such that $\#(\hat{\sigma}_i, t_{in}) = 0$ and $\#(\hat{\sigma}_i, i) = 1$ and $\sigma_i$ is a valid firing string under the supervision of the marking $m_1^0$ in $N$ (cf. [4, Property 3]). From the construction of $N$ and the definition of $P$, we infer $\sigma_i$ is a valid firing string under the supervision of $P$ at the marking $m_1^0$ in $M$. Hence $i$ is live under the supervision of $P$ in $M$.
Theorem 3.2 [5, 6]: For a given CCPN $M = (\Pi, T, \Phi, m_0, C, B)$, with an underlying PN $N = (\Pi, T, \Phi, m_0)$, there exists a supervisory policy $P : N^* \rightarrow \{0, 1\}$ that enforces liveness, if and only if there is a valid firing string $s = \sigma_1 \sigma_2$, in $N$, starting from $m_0$, such that: 1) $m^0 \geq m_1$ and 2) all transitions in $T$ appear at least once in the string $\sigma_2$, where $m_0 \rightarrow \sigma_1 \rightarrow m_1 \rightarrow \sigma_2 \rightarrow m^0$ in the PN $N$.

The conditions of Theorem 3.2 can be tested by investigating the existence of specific paths (cf. [6] for details) in the coverability graph (cf. [3, Sec. 6.1]) of the PN $N$. The time-complexity of this procedure is polynomially related to the number of vertices in the coverability graph of the PN $N$, which in turn can be exponentially related to the size $\max\{\text{card}(\Pi), \text{card}(T)\}$ of the PN $N$.

Lemma 3.6: If there exists a supervisory policy that enforces liveness in $\tilde{M}$, then there exists a supervisory policy that enforces liveness in $M$.

Proof: If there is a supervisory policy that enforces liveness in $\tilde{M}$, from Theorem 3.2 we infer that in the PN $\tilde{N}$: 1) $\tilde{m}^0 \rightarrow \tilde{\sigma}_1 \rightarrow \tilde{m}_1 \rightarrow \tilde{\sigma}_2 \rightarrow \tilde{m}_2$; 2) $\tilde{m}^0 \geq \tilde{m}_1$; and 3) all transitions in $\tilde{T}$ appear at least once in $\tilde{\sigma}_2$. Let $P_{\text{trivial}} : N_{\text{card}(T)} \rightarrow \{0, 1\}$ be the trivial supervisory policy of permanently controlling all transitions in $T(\tilde{T})$. It is easy to see that if $P = P_{\text{trivial}}$, then $\tilde{P} = P_{\text{trivial}}$. Applying Lemma 3.3 to the case when $P = P_{\text{trivial}}$, we infer that in the PN $N$, $m^0 \rightarrow f(\tilde{\sigma}_1) \rightarrow m_1 \rightarrow f(\tilde{\sigma}_2) \rightarrow m^0$. Since every transition in $\tilde{T}$ appears at least once in $\tilde{\sigma}_2$, we conclude every transition in $T$ appears at least once in $f(\tilde{\sigma}_2)$ also. From the fact that $\forall j \in \tilde{\Pi}$, $\tilde{m}^0(\tilde{\sigma}_1)(j) \geq \tilde{m}_1(\tilde{\sigma}_1)(j)$, and Lemma 3.2 when $P = P_{\text{trivial}}$, we conclude $\forall j \in \Pi$, $m^0(p) \geq m^0(p)$. From Theorem 3.2 we conclude there is a supervisory policy that enforces liveness in $M$.

Theorem 3.1 follows directly from Lemmas 3.5 and 3.6. According to Theorem 3.1, the existence of a supervisor that enforces liveness in $\tilde{M}$ is equivalent to the existence of a supervisor that enforces liveness in $M$, provided the PN $\tilde{N}$ is a live MGPN, with a simple path from $i_{\text{in}}$ to $i_{\text{out}}$ that is empty at the initial-marking. Significant computational savings can be gained if the coverability graph of the underlying PN of the CCPN $M$ is smaller than that of the PCCPN $\tilde{M}$.

We illustrate the utility of the results of this paper by an example. Let $\tilde{M}$ be the CCPN with an underlying PN $\tilde{N}$ as shown in Fig. 1(c). The PN $\tilde{N}$ is obtained by refining the transition $t_0$ in the PN $N$ shown in Fig. 1(a) by the PN $N$ shown in Fig. 1(b). The PN $\tilde{N}$ is a live MGPN with a simple path $P = \tilde{i}_0 \tilde{t}_0 \tilde{i}_{\text{out}}$ from $\tilde{i}_0$ to $\tilde{i}_{\text{out}}$ that is empty at the initial-marking. It is worthwhile to note that the time-complexity of testing if a given PN $\tilde{N} = (\tilde{\Pi}, \tilde{T}, \tilde{\Phi}, \tilde{m}_0)$ is an MGPN is $O(\text{card}(\Pi) \times \text{card}(T))$. When presented with the refined PN $\tilde{N}$, a candidate for $N$ can be identified in at least $O(k^2)$ time, where $k = \max\{\text{card}(\Pi), \text{card}(T)\}$. Also, there are efficient procedures that test the liveness of an MGPN (cf. [2, Sec. 1.3 and Table I]). Essentially, the process of abstraction as suggested by the conditions of this paper is not computationally expensive. For any given refined PN $\tilde{N}$ there might be many possible candidates for the abstracted PN $N$ and the live MGPN $\tilde{N}$ that satisfies the requirements of this paper. However, finding a candidate abstracted PN $N$ with the smallest coverability graph can be computationally expensive.

The coverability graph of the PN $\tilde{N}$ shown in Fig. 1(c) has 80 vertices, while that of the PN $N$ has only four vertices. Clearly, testing the existence of a supervisory policy that enforces liveness in the CCPN $\tilde{M}$ is easier than the corresponding test for the CCPN $\tilde{M}$. There exists a supervisory policy that enforces liveness in the CCPN $\tilde{M}$. This can be inferred by Theorem 3.2 and the fact that in the PN $N$, $m^0 \rightarrow t_2 t_3 t_0 t_1 \rightarrow m^0$ and all transitions in $T$ appear once in the string $t_2 t_3 t_0 t_1$. The supervisory policy $P$ of preventing the firing of transition $t_1$ when there is a token in $p_2$ enforces liveness in $M$. From the Proof of Lemma 3.5 we infer the supervisory policy $\tilde{P}$ of preventing the firing of transition $t_1$ when there is a token in $p_2$ also enforces liveness in $\tilde{M}$.

IV. CONCLUSION

References [5] and [6] introduce a necessary and sufficient condition for the existence of a supervisory policy that enforces liveness in a CCPN. This procedure can be computationally expensive. Using the refinement/abstraction procedure of Suzuki and Murata [8], we presented a procedure of reducing the computational burden of this test under the circumstances that are enunciated in the paper. Using an example we illustrated the computational savings of this procedure. As a future research direction we suggest investigations into weakening the restrictions on the PN $N$ that yields a similar result. Toward this end, it might be worthwhile to investigate the application of the transformations that preserve liveness as those listed in [1] to the synthesis of supervisory policies in complex nonlive CCPN’s from similar policies for a simpler abstracted CCPN that is also nonlive.

REFERENCES


