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

Ramavarapu 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 N is replaced by a PN $\tilde{N}$ to yield a larger refined PN $\hat{N}$, we show that when $\tilde{N}$ 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 $\hat{N}$ if and only if there is a similar policy for the abstracted PN N. Since the coverability graph of the PN N is smaller than that of the PN $\hat{N}$, it is possible to achieve significant computational savings by using the process of abstraction on $\hat{N}$. 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 PNs* (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 N by a PN $\tilde{N}$, resulting in a PN $\hat{N}$. The abstraction procedure is essentially the reversal of this process. In this paper we show that when $\tilde{N}$ is a live, *marked-graph* PN (MGPN) with some additional restrictions, there exists a supervisory policy that enforces liveness in the refined PN $\hat{N}$ if and only if there exists a policy that enforces liveness in the abstracted PN N. Since the coverability graph of the abstracted PN N is smaller than that of the refined PN $\hat{N}$, 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.

Manuscript received January 22, 1997. This work was supported in part by the National Science Foundation under Grant ECS-9409691.

The author is with the Coordinated Science Laboratory and Department of General Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: sree@deds.csl.uiuc.edu).

Publisher Item Identifier S 0018-9286(99)00567-X.

II. Notational Preliminaries and Review of Prior Work

A PN $N = (\Pi, T, \Phi, \mathbf{m}^0)$ is an ordered 4-tuple, where $\Pi = \{p_1, p_2, \cdots, p_n\}$ is a set of n *places*, $T = \{t_1, t_2, \cdots, t_m\}$ is a set of m *transitions*, $\Phi \subseteq (\Pi \times T) \cup (T \times \Pi)$ is a set of *arcs*, $\mathbf{m}^0 : \Pi \to \mathcal{N}$ is the *initial-marking function* (or the *initial-marking*), and $\mathcal{N}$ is the set of nonnegative integers. The *marking* of a PN, $\mathbf{m} : \Pi \to \mathcal{N}$, 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 $\forall p \in {}^\bullet t,\ \mathbf{m}(p) \geq 1$, where ${}^\bullet x := \{y \mid (y, x) \in \Phi\}$. For a given marking $\mathbf{m}$ the set of enabled transitions is denoted by the symbol $T_e(N, \mathbf{m})$. An enabled transition $t \in T_e(N, \mathbf{m})$ can *fire*, which changes the marking $\mathbf{m}^1$ to $\mathbf{m}^2$ according to the equation

$$\mathbf{m}^2(p) = \mathbf{m}^1(p) - \mathrm{card}(\mathrm{p}^\bullet \cap \{\mathrm{t}\}) + \mathrm{card}({}^\bullet\mathrm{p} \cap \{\mathrm{t}\}) \qquad (1)$$

where the symbol $\mathrm{card}(\bullet)$ is used to denote the cardinality of the set argument, and $x^\bullet := \{y \mid (x, y) \in \Phi\}$.

A string of transitions $\sigma = t_{j_1} t_{j_2} \cdots t_{j_k}$, where $t_{j_i} \in T$ $(i \in \{1, 2, \cdots, k\})$ is said to be a *valid firing string* at the marking $\mathbf{m}$, if: 1) the transition t_{j_1} is enabled at the marking $\mathbf{m}$ and 2) for $i \in \{1, 2, \cdots, k-1\}$ the firing of the transition t_{j_i} produces a marking at which the transition $t_{j_{i+1}}$ is enabled. Given an initial marking $\mathbf{m}^0$ the set of *reachable markings* for $\mathbf{m}^0$ denoted by $\Re(N, \mathbf{m}^0)$, is the set of markings generated by all valid firing strings at the initial-marking $\mathbf{m}^0$ in the PN N. At a marking $\mathbf{m}^1$, if the firing of a valid firing string σ results in a marking $\mathbf{m}^2$, we represent it as $\mathbf{m}^1 \to \sigma \to \mathbf{m}^2$. A transition $t \in T$ is *live* if $\forall \mathbf{m}^1 \in \Re(N, \mathbf{m}^0)$, $\exists$ a $\mathbf{m}^2 \in \Re(N, \mathbf{m}^1)$ such that $t \in T_e(N, \mathbf{m}^2)$. The PN N is *live* if every transition $t \in T$ is live. For any valid firing string $\sigma \in T^*$, we use the symbol $\#(\sigma, t)$ to denote the number of occurrences of the transition $t \in T$ in σ, and the symbol $|\sigma|$ to denote the length of the string σ.

A PN $N = (\Pi, T, \Phi, \mathbf{m}^0)$ is said to be an MGPN, if $\forall p \in \Pi,\ \mathrm{card}({}^\bullet\mathrm{p}) = \mathrm{card}(\mathrm{p}^\bullet) = 1$. 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 P from t_i to t_j is a string of alternating transitions and places, $t_i p_{k_1} t_{k_1} p_{k_2} t_{k_2} \cdots p_{k_l} t_j$, such that $\{(t_i, p_{k_1}), (p_{k_1}, t_{k_1}), (t_{k_1}, p_{k_2}), \cdots, (p_{k_l}, t_j)\} \subseteq \Phi$. 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 $N = (\Pi, T, \Phi, \mathbf{m}^0)$, 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 $\Pi = \{p_1, p_2, \cdots, p_n\}$ is a set of n *state-places*; $T = \{t_1, t_2, \cdots, t_m\}$ is a set of m *transitions*, $\Phi \subseteq (\Pi \times T) \cup (T \times \Pi)$ is a set of *state-arcs*; $C = \{c_1, c_2, \cdots, c_m\}$[1] is the set of *control-places*; $B = \{(c_i, t_i) \mid i = 1, 2, \cdots, m\}$ is the set of *control-arcs*; $\mathbf{m}^0 : \Pi \to \mathcal{N}$ is the *initial-marking function*

[1] Note that $\mathrm{card}(C) = \mathrm{card}(T) = m$.

(or the *initial-marking*); and $\mathcal{N}$ is the set of nonnegative integers. The CCPN $M = (\Pi, T, \Phi, \mathbf{m}^0, C, B)$ contains the underlying PN $N = (\Pi, T, \Phi, \mathbf{m}^0)$. 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* $\mathbf{u} : C \to \{0,1\}$ assigns a token load of 0 or 1 to each control place. The control can also be interpreted as an m-dimensional binary vector $\mathbf{u} \in \{0,1\}^m$. It would help to view the control $\mathbf{u}$ as follows: if the ith component of $\mathbf{u}$, or $\mathbf{u}(c_i)$ is 0 (1), then transition t_i is control-disabled (control-enabled). For a given marking $\mathbf{m}$ (control $\mathbf{u}$), a transition $t_i \in T$ is said to be state-enabled (control-enabled) if $t_i \in T_e(N, \mathbf{m})$ (if $\mathbf{u}(c_i) = 1$). A transition that is control-enabled and state-enabled can fire resulting in the marking given by (1). A *supervisory policy* $\mathcal{P} : \mathcal{N}^n \to \{0,1\}^m$ 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 $\mathcal{P}$, a string of transitions $\sigma = t_{j_1} t_{j_2} \cdots t_{j_k}$, where $t_{j_i} \in T$ $(i \in \{1, 2, \cdots, k\})$ is said to be a *valid firing string under supervision* at the marking $\mathbf{m}^1$, if: 1) the transition t_{j_1} is state-enabled at the marking $\mathbf{m}^1, \mathcal{P}(\mathbf{m}^1)_{j_1} = 1$ and 2) for $i \in \{1, 2, \cdots, k-1\}$ the firing of the transition t_{j_i} produces a marking $\mathbf{m}^i$ at which the transition $t_{j_{i+1}}$ is state-enabled and $\mathcal{P}(\mathbf{m}^i)_{j_{i+1}} = 1$. For a given supervisory policy $\mathcal{P}$, the set of *reachable markings under supervision* for a CCPN M with initial-marking $\mathbf{m}^0$, denoted by $\Re(M, \mathbf{m}^0, \mathcal{P})$, is the set of markings generated by all valid firing strings under supervision at the marking $\mathbf{m}^0$ in the CCPN M. For the CCPN M, a transition $t_{j_i} \in T$ is *live* under $\mathcal{P}$ if $\forall \mathbf{m}^1 \in \Re(M, \mathbf{m}^0, \mathcal{P})$, $\exists$ a $\mathbf{m}^2 \in \Re(M, \mathbf{m}^1, \mathcal{P})$ such that $t_{j_i} \in T_e(N, \mathbf{m}^2)$ and $\mathcal{P}(\mathbf{m}^2)_{j_i} = 1$. A supervisory policy $\mathcal{P}$ enforces liveness in a CCPN M if all transitions in M are live under $\mathcal{P}$.

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

$$\hat{\Pi} = \Pi \cup \tilde{\Pi}$$
$$\hat{T} = (T \cup \tilde{T}) - \{t_0\}$$
$$\hat{\Phi} = ((\Phi \cup \tilde{\Phi}) - (\{t_0\} \times \Pi) - (\Pi \times \{t_0\})) \cup \{(p, \tilde{t}_{\text{in}}) \mid (p, t_0) \in \Phi\} \cup \{(\tilde{t}_{\text{out}}, p) \mid (t_0, p) \in \Phi\}$$
$$\hat{\mathbf{m}}^0(\hat{p}) = \begin{cases} \mathbf{m}^0(p), & \text{if } \hat{p} \in \Pi, \\ \tilde{\mathbf{m}}^0(p), & \text{if } \hat{p} \in \tilde{\Pi}. \end{cases}$$

Conversely, the PN N can be *abstracted* from $\hat{N}$ by reversing the process of refinement. That is, in the *refined* PN $\hat{N}$ the subnet defined by the PN $\tilde{N}$ is replaced by the transition t_0 which results in the *abstracted* PN N. Throughout this paper we will use the symbol $N(\hat{N})$ 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 $\hat{N}$ (cf. [8, Th. 11]).

In the remainder we concern ourselves with the existence of supervisory policies that enforce liveness in the CCPN $\hat{M}$ that has $\hat{N}$ as its underlying PN. The PN $\hat{N}$ 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 $\tilde{t}_{\text{in}}$ to $\tilde{t}_{\text{out}}$, there exists a policy that enforces liveness in the CCPN $\hat{M}$ if and only if there is a policy that enforces liveness in 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 $\hat{M}$. Although the main result (cf. Theorem 3.1) is stated in terms of the PN $\hat{N}$ 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 $\hat{M}$ relies on the abstraction of the (possibly simpler) PN N from (possibly complicated) PN $\hat{N}$.

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 $\hat{M}$. 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 $\hat{M}$ from a corresponding policy for the CCPN M. Lemma 3.6 establishes the fact that the existence of a supervisory policy that enforces liveness in the CCPN $\hat{M}$ implies the existence of a similar policy for the CCPN M.

Theorem 3.1: Let $\tilde{N} = (\tilde{\Pi}, \tilde{T}, \tilde{\Phi}, \tilde{\mathbf{m}}^0)$ be a live MGPN such that for a pair of distinct transitions $\{\tilde{t}_{\text{in}}, \tilde{t}_{\text{out}}\} \subseteq \tilde{T}$, there is a simple path from $\tilde{t}_{\text{in}}$ to $\tilde{t}_{\text{out}}$ that is empty at the initial-marking $\tilde{\mathbf{m}}^0$. For an arbitrary PN $N = (\Pi, T, \Phi, \mathbf{m}^0)$ with a distinct transition $t_0 \in T$, $(\Pi \cap \tilde{\Pi} = T \cap \tilde{T} = \emptyset)$, let $\hat{N} = (\hat{\Pi}, \hat{T}, \hat{\Phi}, \hat{\mathbf{m}}^0)$ be the PN obtained by refining the transition t_0 by the PN $\tilde{N}$ as illustrated above. If $\hat{M}(M)$ is a CCPN with the underlying PN $\hat{N}(N)$, then there is a supervisory policy that enforces liveness in the CCPN $\hat{M}$ if and only if there exists a supervisory policy that enforces liveness in the CCPN 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 $\tilde{t}_{\text{in}}$ to $\tilde{t}_{\text{out}}$ will have the path $P = \tilde{t}_{\text{in}} p_6 \tilde{t}_{\text{out}}$ as a suffix. So, P is a simple path from $\tilde{t}_{\text{in}}$ to $\tilde{t}_{\text{out}}$, which is empty at the initial-marking. The PN N shown in Fig. 1(a) is not live, and neither is the PN $\hat{N}$ shown in Fig. 1(c). Let $\hat{M}(M)$ 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 $\hat{M}$, 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 M can be readily converted into a policy that enforces liveness in $\hat{M}$. 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 $\tilde{t}_{\text{out}}$ never exceeds that of transition $\tilde{t}_{\text{in}}$ in any valid firing string at the initial-marking and 2) at any point in the evolution of the tokens in $\tilde{N}$, the number of occurrences of transition $\tilde{t}_{\text{out}}$ can be made equal to that of $\tilde{t}_{\text{in}}$ without firing $\tilde{t}_{\text{in}}$. Due to the limitations of space we only present a sketch of

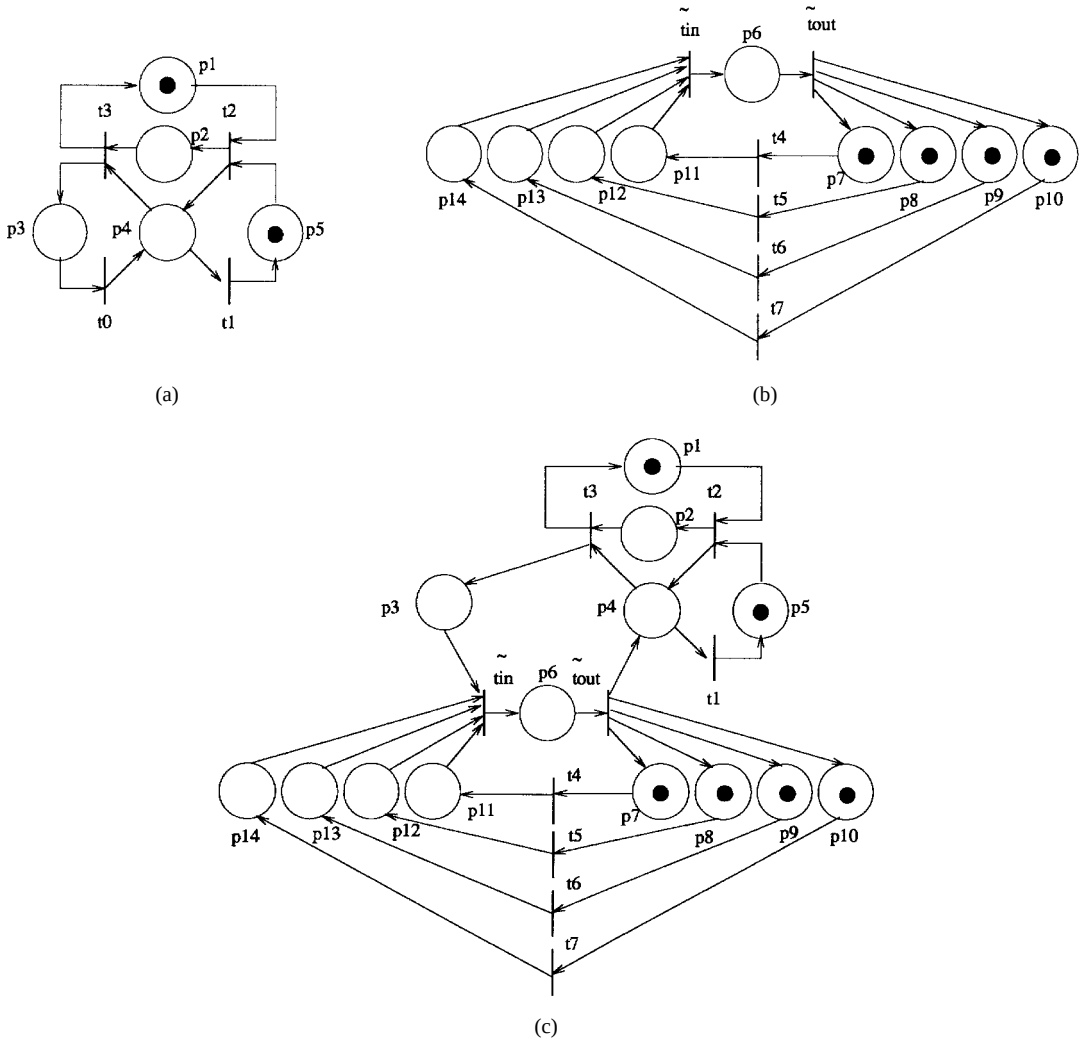

Fig. 1. An illustration of the abstraction/refinement procedure of Suzuki and Murata [8].

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

1) $\#(\tilde{\sigma}, \tilde{t}_{\rm in}) \geq \#(\tilde{\sigma}, \tilde{t}_{\rm out})$;
2) if $\#(\tilde{\sigma}, \tilde{t}_{\rm in}) > \#(\tilde{\sigma}, \tilde{t}_{\rm out})$, then $\exists \tilde{\sigma}_1 \in (\tilde{T} - \{\tilde{t}_{\rm in}\})^*$ such that 1) $\tilde{\sigma}\tilde{\sigma}_1$ is a valid firing string at the initial-marking $\tilde{\mathbf{m}}^0$ and 2) $\#(\tilde{\sigma}\tilde{\sigma}_1, \tilde{t}_{\rm in})$ $(=\#(\tilde{\sigma}, \tilde{t}_{\rm in}))$ $= \#(\tilde{\sigma}\tilde{\sigma}_1, \tilde{t}_{\rm out})$.

The first part of this lemma is established by first observing that the simple path from $\tilde{t}_{\rm in}$ to $\tilde{t}_{\rm out}$ is empty at the initial-marking. Additionally, the number of tokens in the simple path from $\tilde{t}_{\rm in}$ to $\tilde{t}_{\rm out}$ equals the value of $\#(\tilde{\sigma}, \tilde{t}_{\rm in}) - \#(\tilde{\sigma}, \tilde{t}_{\rm out})$. 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 $\tilde{t}_{\rm in}$ to $\tilde{t}_{\rm out}$ contains the simple path from $\tilde{t}_{\rm in}$ to $\tilde{t}_{\rm out}$ as a suffix. Therefore, there can be no token-free directed paths from $\tilde{t}_{\rm in}$ to $\tilde{t}_{\rm out}$. Since $\tilde{N}$ is live, we infer $\tilde{t}_{\rm out}$ can fire without the firing of $\tilde{t}_{\rm in}$ ([4, Property 3]). Repeating this argument $\#(\tilde{\sigma}, \tilde{t}_{\rm in}) - \#(\tilde{\sigma}, \tilde{t}_{\rm out})$ times, we establish the existence of $\tilde{\sigma}_1 \in (\tilde{T} - \{\tilde{t}_{\rm in}\})^*$, such that $\tilde{\sigma}\tilde{\sigma}_1$ is a valid firing string at the initial-marking $\tilde{\mathbf{m}}^0$ in $\tilde{N}$ and $\#(\tilde{\sigma}\tilde{\sigma}_1, \tilde{t}_{\rm in}) = \#(\tilde{\sigma}, \tilde{t}_{\rm in}) = \#(\tilde{\sigma}\tilde{\sigma}_1, \tilde{t}_{\rm out})$. Formal details of the proof of this lemma can be found in [7].

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

$$f(\lambda) = \lambda$$
$$f(\hat{t}) = \begin{cases} \lambda, & \text{if } \hat{t} \in \tilde{T} - \{\tilde{t}_{\rm in}\}, \\ t_0, & \text{if } \hat{t} = \tilde{t}_{\rm in}, \\ \hat{t}, & \text{if } \hat{t} \in T. \end{cases}$$
$$f(\hat{\sigma}\hat{t}) = f(\hat{\sigma})f(\hat{t}), \hat{\sigma} \in \hat{T}^*, \quad \text{and} \quad \hat{t} \in \hat{T},$$

and

$$\tilde{f}(\lambda) = \lambda$$
$$\tilde{f}(\hat{t}) = \begin{cases} \hat{t}, & \text{if } \hat{t} \in \tilde{T}, \\ \lambda, & \text{otherwise.} \end{cases}$$
$$\tilde{f}(\hat{\sigma}\hat{t}) = f(\hat{\sigma})f(\hat{t}), \hat{\sigma} \in \hat{T}^*, \quad \text{and} \quad \hat{t} \in \hat{T},$$

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

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

$$\hat{\mathcal{P}}(\hat{\mathbf{m}})_{\hat{t}} = \begin{cases} \mathcal{P}(\Delta(\hat{\mathbf{m}}))_{\hat{t}}, & \text{if } \hat{t} \in T, \\ \mathcal{P}(\Delta(\hat{\mathbf{m}}))_{t_0}, & \text{if } \hat{t} = \tilde{t}_{\rm in}, \\ 1, & \text{otherwise,} \end{cases}$$

where $\Delta : \mathcal{N}^{\mathrm{card}(\hat{\Pi})} \to \mathcal{N}^{\mathrm{card}(\Pi)}$ is defined as follows: $\forall p \in \Pi$,

$$\Delta(\hat{\mathbf{m}})(p) = \begin{cases} \hat{\mathbf{m}}(p) + \sum_{\hat{p} \in P(\tilde{t}_{\mathrm{in}}, \tilde{t}_{\mathrm{out}})} \hat{\mathbf{m}}(\hat{p}), & \text{if } p \in t_0{}^{\bullet}, \\ \hat{\mathbf{m}}(p), & \text{otherwise,} \end{cases}$$

and $P(\tilde{t}_{\mathrm{in}}, \tilde{t}_{\mathrm{out}})$ denotes the set of places in the simple path from $\tilde{t}_{\mathrm{in}}$ to $\tilde{t}_{\mathrm{out}}$. Lemma 3.2 establishes the relationship between the marking resulting from the firing of appropriate firing strings in $\hat{M}, M$, and $\tilde{N}$.

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

$$\mathbf{m}(p) = \begin{cases} \hat{\mathbf{m}}(p) + \sum_{\tilde{p} \in P(\tilde{t}_{\mathrm{in}}, \tilde{t}_{\mathrm{out}})} \hat{\mathbf{m}}(\tilde{p}), & \text{if } p \in t_0{}^{\bullet}, \\ \hat{\mathbf{m}}(p), & \text{otherwise,} \end{cases}$$

and $\forall \tilde{p} \in \tilde{\Pi}, \tilde{\mathbf{m}}(\tilde{p}) = \hat{\mathbf{m}}(\tilde{p})$, where $P(\tilde{t}_{\mathrm{in}}, \tilde{t}_{\mathrm{out}})$ denotes the set of places in the simple path from $\tilde{t}_{\mathrm{in}}$ to $\tilde{t}_{\mathrm{out}}$.

The details of the proof are skipped for brevity. The above result can be established by an induction argument over $|\hat{\sigma}|$, the length of $\hat{\sigma}$. The base case is established by letting $\hat{\sigma} = \lambda$. For the induction step we let $\hat{\sigma} = \hat{\sigma}_1 \hat{t}$, where $|\hat{\sigma}_1| = n$, for some $n \in \mathcal{N}$. The induction step for any $\tilde{p} \in \tilde{\Pi}$ follows directly from the definition of $\tilde{f}(\bullet)$ and the fact that the subnet $\tilde{N}$ is preserved intact in the construction of $\hat{N}$. The induction step for any $p \in \Pi$ is easily established for $\hat{t} \in \hat{T} - \{\tilde{t}_{\mathrm{in}}, \tilde{t}_{\mathrm{out}}\}$. Noting that $f(\tilde{t}_{\mathrm{in}}) = t_0$, we infer the firing of $\tilde{t}_{\mathrm{in}}$ in $\hat{N}$ corresponds to the firing of t_0 in N. Consequently, in N the token-load of the output places of t_0 would increase by unity. On the other hand, in $\hat{N}$ the firing of $\tilde{t}_{\mathrm{in}}$ will increase the token load of the output places of $\tilde{t}_{\mathrm{in}}$ by unity. Only one of these places belongs to $P(\tilde{t}_{\mathrm{in}}, \tilde{t}_{\mathrm{out}})$ as $P(\tilde{t}_{\mathrm{in}}, \tilde{t}_{\mathrm{out}})$ is a simple path from $\tilde{t}_{\mathrm{in}}$ to $\tilde{t}_{\mathrm{out}}$. Since the token load of this simple path is added to the output places of t_0, the induction step is established for the case when $\hat{t} = \tilde{t}_{\mathrm{in}}$. Finally, we note $f(\tilde{t}_{\mathrm{out}}) = \lambda$, so the token load of the places in N remain unchanged for this case, while in $\hat{N}$ the token load of the output places of $\tilde{t}_{\mathrm{out}}$ would increase by unity, while the sum of the token-loads of the places in $P(\tilde{t}_{\mathrm{in}}, \tilde{t}_{\mathrm{out}})$ will decrease by unity as $P(\tilde{t}_{\mathrm{in}}, \tilde{t}_{\mathrm{out}})$ is a simple path from $\tilde{t}_{\mathrm{in}}$ to $\tilde{t}_{\mathrm{out}}$. This establishes the induction step for the case when $\hat{t} = \tilde{t}_{\mathrm{out}}$, and the result is proven.

In Lemma 3.3 we show that any firing string that is valid under the supervision of $\hat{\mathcal{P}}$ in $\hat{M}$ corresponds to: 1) a firing string that is valid under the supervision of $\mathcal{P}$ in M and 2) a valid firing string in the PN $\tilde{N}$.

Lemma 3.3: For any firing string $\hat{\sigma} \in \hat{T}^*$ that is valid under the supervision of $\hat{\mathcal{P}}$ at the initial-marking $\hat{\mathbf{m}}^0$ in $\hat{M}$, the following observations hold.

1) $f(\hat{\sigma})$ is valid under the supervision of $\mathcal{P}$ at the initial-marking $\mathbf{m}^0$ in M.
2) $\tilde{f}(\hat{\sigma})$ is a valid firing string in the PN $\tilde{N}$ at the initial-marking $\tilde{\mathbf{m}}^0$.

This result can be established by induction on $|\hat{\sigma}|$, the length of $\hat{\sigma}$. The base case is easily established by letting $\hat{\sigma}$ equal the null-string. As the induction hypothesis the statement of the lemma is assumed to be true of any $\hat{\sigma}$ such that $|\hat{\sigma}| \leq n$ for some $n \in \mathcal{N}$. For the induction step we let $\hat{\sigma} = \hat{\sigma}_1 \hat{t}$, where $|\hat{\sigma}_1| \leq n$. The induction step is easily established for the cases when $\hat{t} \in \tilde{T} - \{\tilde{t}_{\mathrm{in}}\}$, or, $\hat{t} \notin \tilde{T}$. For the cases when $\hat{t} \in (T - \{t_0\}) \cup \{\tilde{t}_{\mathrm{in}}\}$, and $\hat{t} \in \tilde{T}$, using Lemma 3.2 it can be shown that $f(\hat{\sigma}_1 \hat{t})$ is valid under the supervision of $\mathcal{P}$ at the initial-marking $\mathbf{m}^0$ in M, and $\tilde{f}(\hat{\sigma}_1 \hat{t})$ is a valid firing string in the PN $\tilde{N}$ at the initial-marking $\tilde{\mathbf{m}}^0$. The details of this induction proof are skipped in the interest of space and can be found in [7].

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

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

We now present a sketch of the proof of the above lemma. First we let $\sigma = \sigma_1 t_0 \sigma_2 t_0 \cdots \sigma_n t_0 \sigma_{n+1}$, where $\#(\sigma_i, t_0) = 0$, $\forall i \in \{1, 2, \cdots, n+1\}$. Using an induction argument we show that the ith occurrence of t_0 in σ can be replaced by a string of transitions $\tilde{\sigma}_{2i-1} \tilde{\sigma}_{2i} \in \tilde{T}^*$ such that $\#(\tilde{\sigma}_{2i-1}, \tilde{t}_{\mathrm{in}}) = \#(\tilde{\sigma}_{2i}, \tilde{t}_{\mathrm{out}}) = 1$, $\#(\tilde{\sigma}_{2i-1}, \tilde{t}_{\mathrm{out}}) = \#(\tilde{\sigma}_{2i}, \tilde{t}_{\mathrm{in}}) = 0$, and the resulting string of transitions, $\sigma_1 \tilde{\sigma}_1 \tilde{\sigma}_2 \sigma_2 \cdots \sigma_n \tilde{\sigma}_{2n-1} \tilde{\sigma}_{2n} \sigma_{n+1}$ will be permitted under the supervision of $\hat{\mathcal{P}}$ at the initial-marking $\hat{\mathbf{m}}^0$ in $\hat{M}$. The result follows from the fact that $f(\sigma_1 \tilde{\sigma}_1 \tilde{\sigma}_2 \sigma_2 \cdots \sigma_n \tilde{\sigma}_{2n-1} \tilde{\sigma}_{2n} \sigma_{n+1}) = \sigma$. 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 $\mathcal{P}$ enforces liveness in the CCPN M, then the policy $\hat{\mathcal{P}}$ enforces liveness in the CCPN $\hat{M}$.

Proof: Let $\hat{\sigma} \in \hat{T}^*$ be any firing string such that $\hat{\mathbf{m}}^0 \to \hat{\sigma} \to \hat{\mathbf{m}}^1$ under the supervision of $\hat{\mathcal{P}}$ in $\hat{M}$. We show that any $\hat{t} \in \hat{T}$ can be fired after the valid firing string $\hat{\sigma}$.

Case 1: $(\hat{t} \in T - \{t_0\})$. From Lemma 3.3 we know $\mathbf{m}^0 \to f(\hat{\sigma}) \to \mathbf{m}^1$ in M under the supervision of $\mathcal{P}$. Since $\mathcal{P}$ enforces liveness, $\exists \sigma_1 \in T^*$ such that $f(\hat{\sigma}) \sigma_1 \hat{t}$ is valid under the supervision of $\mathcal{P}$ at the initial-marking $\mathbf{m}^0$ in M. By Lemma 3.4 we know $\exists \hat{\sigma}_1 \in \hat{T}^*$ such that $f(\hat{\sigma}_1 \hat{t}) = \sigma_1 \hat{t}$ and $\hat{\sigma} \hat{\sigma}_1 \hat{t}$ is valid under the supervision of $\hat{\mathcal{P}}$ at the initial-marking $\hat{\mathbf{m}}^0$ in M. Hence every transition in $T - \{t_0\}$ is live under the supervision of $\hat{\mathcal{P}}$ in $\hat{M}$.

Case 2: $(\hat{t} = \tilde{t}_{\mathrm{in}})$. Using the same argument as in Case 1, we establish the existence of a firing string $f(\hat{\sigma}) \sigma_1 t_0$ that is valid under the supervision of $\mathcal{P}$ at the initial-marking $\mathbf{m}^0$ in M. By Lemma 3.4, we infer the firing string $f(\hat{\sigma}) \sigma_1 t_0$ can be simulated by a firing string $\hat{\sigma} \hat{\sigma}_1 \in \hat{T}^*$ in $\hat{M}$, where $f(\hat{\sigma}_1) = \sigma_1 t_0$. From the definition of $f(\bullet)$ we infer $\exists \hat{\sigma}_2$, $\hat{\sigma}_3 \in \hat{T}^*$ such that $\hat{\sigma}_1 = \hat{\sigma}_2 \tilde{t}_{\mathrm{in}} \hat{\sigma}_3$, and $f(\hat{\sigma}_3) = \lambda$. This implies $\tilde{t}_{\mathrm{in}}$ is live under the supervision of $\hat{\mathcal{P}}$ in $\hat{M}$.

Case 3: $(\hat{t} \in \tilde{T} - \{\tilde{t}_{\mathrm{in}}\})$. From Lemma 3.3 we know $\tilde{\mathbf{m}}^0 \to \tilde{f}(\hat{\sigma}) \to \tilde{\mathbf{m}}^1$ in the PN $\tilde{N}$. Since $\tilde{N}$ is live, it is possible to fire $\hat{t}$, although not necessarily immediately, starting at the marking $\tilde{\mathbf{m}}^1$. We consider two subcases.

Case 3a: If there is no token-free path from $\tilde{t}_{\mathrm{in}}$ to $\hat{t}$ at $\tilde{\mathbf{m}}^1$ in $\tilde{N}$, then since $\tilde{N}$ is a live MGPN, $\exists \tilde{\sigma}_1 \in \tilde{T}^*$ such that $\tilde{\sigma}_1 \hat{t}$ is a valid firing string in the PN $\tilde{N}$ at the marking $\tilde{\mathbf{m}}^1$ and $\#(\tilde{\sigma}_1, \tilde{t}_{\mathrm{in}}) = 0$ (cf. [4, Property 3]). From the construction of $\hat{N}$ and the definition of $\hat{\mathcal{P}}$ we infer the firing string $\hat{\sigma} \tilde{\sigma}_1 \hat{t}$ is valid in $\hat{M}$ under the supervision of $\hat{\mathcal{P}}$ at the initial-marking $\hat{\mathbf{m}}^0$.

Case 3b: If there is a token-free path from $\tilde{t}_{\mathrm{in}}$ to $\hat{t}$ at $\tilde{\mathbf{m}}^1$ in $\tilde{N}, \hat{t}$ cannot fire until $\tilde{t}_{\mathrm{in}}$ has fired once in $\tilde{N}$ (cf. [4, Property 3]). Since $\tilde{t}_{\mathrm{in}}$ in live under the supervision of $\hat{\mathcal{P}}$ in $\hat{M}$ (cf. Case 2), $\exists \hat{\sigma}_1 \in \hat{T}^*$ such that $\#(\hat{\sigma}_1, \tilde{t}_{\mathrm{in}}) = 1$ and $\hat{\mathbf{m}}^0 \to \hat{\sigma} \hat{\sigma}_1 \to \hat{\mathbf{m}}^2$ under the supervision of $\hat{\mathcal{P}}$ in $\hat{M}$. From Lemma 3.3 we infer $\tilde{\mathbf{m}}^0 \to \tilde{f}(\hat{\sigma} \hat{\sigma}_1) \to \tilde{\mathbf{m}}^2$ in the PN $\tilde{N}$. Since $\#(\tilde{f}(\hat{\sigma} \hat{\sigma}_1), \tilde{t}_{\mathrm{in}}) = 1$, all paths from $\tilde{t}_{\mathrm{in}}$ to $\hat{t}$ in the PN $\tilde{N}$ will be nonempty at the marking $\tilde{\mathbf{m}}^2$. Since $\tilde{N}$ is live, we infer $\exists \tilde{\sigma}_3 \in \tilde{T}^*$ such that $\#(\tilde{\sigma}_3, \tilde{t}_{\mathrm{in}}) = 0$ and $\#(\tilde{\sigma}_3, \hat{t}) = 1$ and $\tilde{\sigma}_3$ is a valid firing string at the marking $\tilde{\mathbf{m}}^2$ in $\tilde{N}$ (cf. [4, Property 3]). From the construction of $\hat{N}$ and the definition of $\hat{\mathcal{P}}$, we infer $\tilde{\sigma}_3$ is a valid firing string under the supervision of $\hat{\mathcal{P}}$ at the marking $\hat{\mathbf{m}}^2$ in $\hat{M}$. Hence $\hat{t}$ is live under the supervision of $\hat{\mathcal{P}}$ in $\hat{M}$. □

Theorem 3.2 [5], [6]: For a given CCPN $M = (\Pi, T, \Phi, \mathbf{m}^0, C, B)$, with an underlying PN $N = (\Pi, T, \Phi, \mathbf{m}^0)$, there exists a supervisory policy $\mathcal{P} : \mathcal{N}^n \rightarrow \{0,1\}^m$ that enforces liveness, if and only if $\exists$ a valid firing string $\sigma = \sigma_1\sigma_2$, in N, starting from $\mathbf{m}^0$, such that: 1) $\mathbf{m}^2 \geq \mathbf{m}^1$ and 2) all transitions in T appear at least once in the string σ_2, where $\mathbf{m}^0 \rightarrow \sigma_1 \rightarrow \mathbf{m}^1 \rightarrow \sigma_2 \rightarrow \mathbf{m}^2$ 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*, $\max\{\mathrm{card}(\Pi), \mathrm{card}(\mathrm{T})\}$, of the PN N.

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

Proof: If there is a supervisory policy that enforces liveness in $\hat{M}$, from Theorem 3.2 we infer $\exists \hat{\sigma}_1, \hat{\sigma}_2 \in \hat{T}^*$ such that in the PN $\hat{N}$: 1) $\hat{\mathbf{m}}^0 \rightarrow \hat{\sigma}_1 \rightarrow \hat{\mathbf{m}}^1 \rightarrow \hat{\sigma}_2 \rightarrow \hat{\mathbf{m}}^2$; 2) $\hat{\mathbf{m}}^2 \geq \hat{\mathbf{m}}^1$; and 3) all transitions in $\hat{T}$ appear at least once in $\hat{\sigma}_2$.

Let $\mathcal{P}_{\mathrm{trivial}} : \mathcal{N}^{\mathrm{card}(\Pi)} \rightarrow \{1\}^{\mathrm{card}(T)}$ ($\hat{\mathcal{P}}_{\mathrm{trivial}} : \mathcal{N}^{\mathrm{card}(\hat{\Pi})} \rightarrow \{1\}^{\mathrm{card}(\hat{T})}$) be the trivial supervisory policy of permanently control-enabling all transitions in $T(\hat{T})$. It is easy to see that if $\mathcal{P} = \mathcal{P}_{\mathrm{trivial}}$, then $\hat{\mathcal{P}} = \hat{\mathcal{P}}_{\mathrm{trivial}}$. Applying Lemma 3.3 to the case when $\mathcal{P} = \mathcal{P}_{\mathrm{trivial}}$, we infer that in the PN N, $\mathbf{m}^0 \rightarrow f(\hat{\sigma}_1) \rightarrow \mathbf{m}^1 \rightarrow f(\hat{\sigma}_2) \rightarrow \mathbf{m}^2$. Since every transition in $\hat{T}$ appears at least once in $\hat{\sigma}_2$, we conclude every transition in T appears at least once in $f(\hat{\sigma}_2)$ also. From the fact that $\forall \hat{p} \in \hat{\Pi}$, $\hat{\mathbf{m}}^2(\hat{p}) \geq \hat{\mathbf{m}}^1(\hat{p})$, and Lemma 3.2 when $\mathcal{P} = \mathcal{P}_{\mathrm{trivial}}$, we conclude $\forall p \in \Pi$, $\mathbf{m}^2(p) \geq \mathbf{m}^1(p)$. 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 $\hat{M}$ 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 $\tilde{t}_{\mathrm{in}}$ to $\tilde{t}_{\mathrm{out}}$ 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 $\hat{M}$.

We illustrate the utility of the results of this paper by an example. Let $\hat{M}$ be the CCPN with an underlying PN $\hat{N}$ as shown in Fig. 1(c). The PN $\hat{N}$ is obtained by refining the transition t_0 in the PN N shown in Fig. 1(a) by the PN $\tilde{N}$ shown in Fig. 1(b). The PN $\tilde{N}$ is a live MGPN with a simple path $P = \tilde{t}_{\mathrm{in}} p_6 \tilde{t}_{\mathrm{out}}$ from $\tilde{t}_{\mathrm{in}}$ to $\tilde{t}_{\mathrm{out}}$ that is empty at the initial-marking. It is worthwhile to note that the time-complexity of testing if a given PN $\bar{N} = (\bar{\Pi}, \bar{T}, \bar{\Phi}, \bar{\mathbf{m}}^0)$ is an MGPN is $O(\mathit{card}(\bar{\Pi}) \times \mathit{card}(\bar{T}))$. When presented with the refined PN $\hat{N}$, a candidate for $\tilde{N}$ can be identified in at least $O(k^5)$ time, where $k = \max(\mathit{card}(\hat{\Pi}), \mathit{card}(\hat{T}))$. 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 $\hat{N}$ 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 $\hat{N}$ 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 M is easier than the corresponding test for the CCPN $\hat{M}$. There exists a supervisory policy that enforces liveness in the CCPN M. This can be inferred by Theorem 3.2 and the fact that in the PN $N, \mathbf{m}^0 \rightarrow t_2 t_3 t_0 t_1 \rightarrow \mathbf{m}^0$ and all transitions in T appear once in the string $t_2 t_3 t_0 t_1$. The supervisory policy $\mathcal{P}$ 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 $\hat{\mathcal{P}}$ of preventing the firing of t_1 when there is a token in p_2 also enforces liveness in $\hat{M}$.

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 ennuciated 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 $\tilde{N}$ that yields a similar result. Toward this end, it might be worthwhile to investigate the application of the transformations that preserve liveness such 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

[1] G. Berthelot, "Checking properties of nets using transformations," in *Advances in Petri Nets*, Lecture Notes in Computer Science, vol. 222. Germany: Springer-Verlag, pp. 19–40, 1985.

[2] N. D. Jones, L. H. Landweber, and Y. E. Lien, "Complexity of some problems in Petri nets," *Theoretical Computer Sci.*, vol. 4, pp. 277–299, 1977.

[3] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, pp. 541–580, Apr. 1989.

[4] T. Murata, V. B. Le, and D. J. Leu, "A method for realizing the synchronic distance matrix as a marked graph," in *IEEE Int. Conf. Circuits and Systems*, Rome, Italy, May 1982, pp. 609–612.

[5] R. S. Sreenivas, "Enforcing liveness via supervisory control in discrete event dynamic systems modeled by completely controlled Petri nets," in *WODES-96: Int. Workshop on Discrete-Event Systems*, University of Edinburgh, U.K., Aug. 1996, pp. 296–301.

[6] ——, "On the existence of supervisory policies that enforce liveness in discrete event dynamic systems modeled by controlled Petri nets," *IEEE Trans. Automat. Contr.*, vol. 42, pp. 928–945, July 1996.

[7] ——, "On supervisory policies that enforce liveness in a class of completely controlled Petri nets obtained via refinement," Tech. Rep. UILU-ENG-97-2226, Univ. Illinois, vol. DC-182, Sept. 1997.

[8] I. Suzuki and T. Murata, "A method for stepwise refinement and abstraction of Petri nets," *J. Computer and Syst. Sci.*, vol. 27, pp. 51–76, 1983.