A Tutorial on the Synthesis of the Maximally Permissive Liveness Enforcing Supervisory Policy in Discrete-Event/Discrete-State Systems modeled by a Class of General Petri Nets

E. Salimi[1], N. Somnath[1] and R.S. Sreenivas[1]

***Abstract*— A *Discrete-Event/Discrete-State* (DEDS) system is in a *livelocked*-state, if *some* process has entered into a state of suspended-animation for perpetuity. If *every* process of the DEDS system is in a state of suspended animation, the DEDS system is *deadlocked*. A livelock-free DEDS system does not have deadlocked-states, but a deadlock-free DEDS system can still experience livelocks. A livelock-prone DEDS system can be regulated by a *liveness enforcing supervisory policy* (LESP) to ensure the supervised-system is livelock-free. A LESP is said to be *maximally permissive*, if the fact that it prevents the occurrence of an event a state leads us to infer that every LESP, irrespective of the implementation-paradigm that is chosen, will do the same.**

We concern ourselves with DEDS systems that are modeled using general *Petri nets* (PNs), and we review key results on the synthesis of the maximally permissive LESP for DEDS systems modeled by PNs using a series of illustrative examples and a software package designed explicitly for this purpose. The paper concludes with an incomplete list of future research topics.

I. INTRODUCTION

We concern ourselves with the problem of synthesizing *liveness enforcing supervisory policies* (LESPs) for a *Petri net* (PN) model of a *Discrete-Event/Discrete-State* (DEDS) system. Essentially, a LESP ensures the *liveness property* (cf. [1]) that irrespective of the past, all events/activities of a DEDS system can occur at some instant in the future. In a PN model of a DEDS system, this translates to the requirement that every transition in the PN model is potentially fireable for all markings that can be reached under the supervision of the LESP. The DEDS system that is modeled by the PN does not experience *livelocks* or *deadlocks* under the supervision of a LESP, which serves as the primary motivation for this endeavor. There are other motivations as well, and we describe a few below.

Oftentimes the correctness of software systems is established by ascertaining the liveness property of an appropriately constructed PN model (cf. [2]), where livelock-freedom in PN models of *VHDL*-code has found use in *asynchronous circuit design*).

Programmable Logic Controllers (PLCs) are programmed predominantly by *Ladder Logic Diagrams* (LLDs). It is a rule-based language (as opposed to a conventional, procedural language). Each LLD consists of several *rungs*. Each rung has a collection of conditions, usually written on the left-hand-side, which when satisfied activates the relevant *coils/relays*, usually written on the right-hand-side of the rung. The rungs are executed sequentially and cyclically. The correctness of an LLD is dependent on the execution order of the rungs and the extant conditions of the sensed-devices. Verification of LLDs involves an investigation into its *safety* and *liveness* properties. Deadlock detection and avoidance in LLDs is a pressing issue (cf. [3]). Attempts at using conventional verification tools to this end have found limited success. These tools are ineffective when the number of discrete states exceeds 10^3 states (cf. [4]). An alternative could be to convert LLD into a PN-equivalent using the extant methods in the literature (cf. [5], [6], for instance). Following this, the extant results in the synthesis of LESPs can be used to enhance the original PN-structure in such a way that the resulting enhanced-PN is live (cf [7]). These structural additions can be interpreted appropriately using the results in the literature on translating PNs to LLDs (cf. [8], [9]). This method is not restricted by size as those mentioned above, as the software reported in references ([10], [11]) can comfortably handle problems with $\approx 10^7$ states. Following this line of reasoning, any medium-size production environment can be comfortably analyzed for correctness.

The examples presented in this paper are intended to reinforce and clarify the theoretical concepts in the synthesis of LESPs for PN models of DEDS systems. It is hoped that the exposition in this paper will serve as a companion to the theoretical results in the references [12], [13], [14], [15], [7], [16]. Additionally, these illustrative examples highlight the capabilities of the software described in references [10], [11].

II. NOTATIONS, DEFINITIONS AND OTHER PRELIMINARIES

We use $\mathcal{N}$ ($\mathcal{N}^+$) to denote the set of non-negative (positive) integers. For $m, n, k \in \mathcal{N}^+$, the set of $m \times n$ matrices (k-dimensional vectors) of non-negative integers is represented as $\mathcal{N}^{m\times n}$ ($\mathcal{N}^k$). The term $card(\bullet)$ denotes the cardinality of the set argument. A *Petri net structure* $N = (\Pi, T, \Phi, \Gamma)$ is an ordered 4-tuple, where $\Pi = \{p_1, \ldots, p_n\}$ is a set of n *places*, $T = \{t_1, \ldots, t_m\}$ is a collection of m *transitions*, $\Phi \subseteq (\Pi \times T) \cup (T \times \Pi)$ is a set of *arcs*, and $\Gamma : \Phi \to \mathcal{N}^+$ is the *weight* associated with each arc. A PN structure is said to be *ordinary* (resp. *general*) if the weight associated with an arc is (resp. not necessarily) unitary. The *initial marking function* (or the *initial marking*) of a PN

[1]Industrial & Enterprise Systems Engineering, and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign {salimi2,somnath1,rsree}@illinois.edu.

structure N is a function $\mathbf{m}^0 : \Pi \to \mathcal{N}$, which identifies the number of *tokens* in each place. We will use the term *Petri net* (PN) and the symbol $N(\mathbf{m}^0)$ to denote a PN structure N along with its initial marking $\mathbf{m}^0$.

A marking $\mathbf{m} : \Pi \to \mathcal{N}$ is sometimes represented by an integer-valued vector $\mathbf{m} \in \mathcal{N}^n$, where the i-th component $\mathbf{m}_i$ represents the token load ($\mathbf{m}(p_i)$) of the i-th place. The weight of an arc is represented by an integer that is placed along side the arc. For brevity, we refrain from denoting the weight of those arcs $\phi \in \Phi$ where $\Gamma(\phi) = 1$.

For a string of transitions $\sigma \in T^*$, $\mathbf{x}(\sigma)$ denotes the *Parikh vector* of σ. That is, the i-th entry, $\mathbf{x}_i(\sigma)$, corresponds to the number of occurrences of transition t_i in σ.

We define the sets ${}^\bullet x := \{y \mid (y,x) \in \Phi\}$ and $x^\bullet := \{y \mid (x,y) \in \Phi\}$. A transition $t \in T$ is said to be *enabled* at a marking $\mathbf{m}^i$ if $\forall p \in {}^\bullet t, \mathbf{m}^i(p) \geq \Gamma((p,t))$. The set of enabled transitions at marking $\mathbf{m}^i$ is denoted by the symbol $T_e(N, \mathbf{m}^i)$. An enabled transition $t \in T_e(N, \mathbf{m}^i)$ can *fire*, which changes the marking $\mathbf{m}^i$ to $\mathbf{m}^{i+1}$ according to $\mathbf{m}^{i+1}(p) = \mathbf{m}^i(p) - \Gamma(p,t) + \Gamma(t,p)$.

A set of markings $\mathcal{M} \subseteq \mathcal{N}^n$ is said to be *right-closed* (cf. [17]) if $((\mathbf{m}^1 \in \mathcal{M}) \wedge (\mathbf{m}^2 \geq \mathbf{m}^1) \Rightarrow (\mathbf{m}^2 \in \mathcal{M}))$. Every right-closed set of vectors $\mathcal{M} \subseteq \mathcal{N}^n$ contains a finite set of minimal-elements, $min(\mathcal{M}) \subset \mathcal{M}$. The set $\min(\mathcal{M})$ serves as a proxy for $\mathcal{M}$.

A. Supervisory Control of PNs

The paradigm of supervisory control of PNs assumes a subset of *controllable transitions*, denoted by $T_c \subseteq T$, which 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 $T_u = T - T_c$. The controllable (resp. uncontrollable) transitions are represented as filled (resp. unfilled) boxes in graphical representation of PNs.

A *supervisory policy* $\mathcal{P} : \mathcal{N}^n \times T \to \{0,1\}$, is a function that returns a 0 or 1 for each transition and each reachable marking. The supervisory policy $\mathcal{P}$ permits the firing of transition t_j at marking $\mathbf{m}^i$, iff $\mathcal{P}(\mathbf{m}^i, t_j) = 1$. If $t_j \in T_e(N, \mathbf{m}^i)$ for some marking $\mathbf{m}^i$, we say the transition t_j is *state-enabled* at $\mathbf{m}^i$. If $\mathcal{P}(\mathbf{m}^i, t_j) = 1$, we say the transition t_j is *control-enabled* at $\mathbf{m}^i$. A transition has to be state- 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 $\forall \mathbf{m}^i \in \mathcal{N}^n, \mathcal{P}(\mathbf{m}^i, t_j) = 1$, if $t_j \in T_u$. This is implicitly assumed of any supervisory policy in this paper.

A string of transitions $\sigma = t_1 t_2 \cdots t_k$, where $t_j \in T (j \in \{1, 2, \ldots, k\})$ is said to be a *valid firing string* starting from the marking $\mathbf{m}^i$, if, (1) $t_1 \in T_e(N, \mathbf{m}^i), \mathcal{P}(\mathbf{m}^i, t_1) = 1$, and (2) for $j \in \{1, 2, \ldots, k-1\}$ the firing of the transition t_j produces a marking $\mathbf{m}^{i+j}$ and $t_{j+1} \in T_e(N, \mathbf{m}^{i+j})$ and $\mathcal{P}(\mathbf{m}^{i+j}, t_{j+1}) = 1$.

The set of reachable markings under the supervision of $\mathcal{P}$ in N from the initial marking $\mathbf{m}^0$ is denoted by $\Re(N, \mathbf{m}^0, \mathcal{P})$. If $\mathbf{m}^{i+k}$ results from the firing of $\sigma \in T^*$ starting from the initial marking $\mathbf{m}^i$, we represent it symbolically as $\mathbf{m}^i \xrightarrow{\sigma} \mathbf{m}^{i+k}$.

A transition t_k is *live* under the supervision of $\mathcal{P}$ if $\forall \mathbf{m}^i \in \Re(N, \mathbf{m}^0, \mathcal{P}), \exists \mathbf{m}^j \in \Re(N, \mathbf{m}^i, \mathcal{P})$ such that $t_k \in T_e(N, \mathbf{m}^j)$ and $\mathcal{P}(\mathbf{m}^j, t_k) = 1$.

A policy $\mathcal{P}$ is a *liveness enforcing supervisory policy* (LESP) for $N(\mathbf{m}^0)$ if all transitions in $N(\mathbf{m}^0)$ are live under $\mathcal{P}$. The policy $\mathcal{P}$ is said to be *maximally permissive* if for every LESP $\widehat{\mathcal{P}} : \mathcal{N}^n \times T \to \{0,1\}$ for $N(\mathbf{m}^0)$, the following condition holds $\forall \mathbf{m}^i \in \mathcal{N}^n, \forall t \in T, \mathcal{P}(\mathbf{m}^i, t) \geq \widehat{\mathcal{P}}(\mathbf{m}^i, t)$. The next section presents an informal description of the key results in the synthesis of LESPs for PN models of DEDS systems.

III. Results

For any PN structure N, we define the set of initial markings $\mathbf{m}^0$ of N for which there is a LESP for $N(\mathbf{m}^0)$, as follows

$$\Delta(N) = \{\mathbf{m}^0 \in \mathcal{N}^n \mid \exists \text{a LESP for } N(\mathbf{m}^0)\}.$$

The set $\Delta(N)$ is *control invariant* with respect to N. That is, if $\mathbf{m}^1 \in \Delta(N), t_u \in T_e(N, \mathbf{m}^1) \cap T_u$ and $\mathbf{m}^1 \xrightarrow{t_u} \mathbf{m}^2$ in N, then $\mathbf{m}^2 \in \Delta(N)$. Alternately, only the firing of a controllable transition at a marking in $\Delta(N)$ can result in a new marking that is not in $\Delta(N)$. The PN $N(\mathbf{m}^0)$ has a LESP if and only if $\mathbf{m}^0 \in \Delta(N)$. This leads us to the first result regarding the structure of the maximally permissive LESP.

Result 1: (Lemma 5.9, [12]) The supervisory policy that control-disables a transition only if its firing at a marking in $\Delta(N)$ would result in a new marking that is not in $\Delta(N)$, is the maximally permissive LESP for $N(\mathbf{m}^0)$ for $\mathbf{m}^0 \in \Delta(N)$.

As an illustration, consider the PN structure N_1 shown in figure 1(a). This structure belongs to a class of PNs called *General Free Choice*[1] PNs. It is not hard to show that $\Delta(N_1) = \{\mathbf{m} \in \mathcal{N}^5 \mid (\mathbf{m}(p_1) + \mathbf{m}(p_2) + \mathbf{m}(p_3) + \mathbf{m}(p_4) \geq 1) \vee (\mathbf{m}(p_5)_{mod2} = 1)\}$. The control invariance of $\Delta(N_1)$ with respect to N_1 follows from the fact that each firing of the uncontrollable transition t_6 at marking $\mathbf{m}^1 \in \Delta(N_1)$ will take two (i.e. an even number of) tokens out of p_5, if $\mathbf{m}^1 \xrightarrow{t_6} \mathbf{m}^2$ in N_1, then (i) if $\mathbf{m}^1(p_1)+\mathbf{m}^1(p_2)+\mathbf{m}^1(p_3)+\mathbf{m}^1(p_4) \geq 0$, then $\mathbf{m}^2(p_1)+\mathbf{m}^2(p_2)+\mathbf{m}^2(p_3)+\mathbf{m}^2(p_4) \geq 0$, and $\mathbf{m}^2 \in \Delta(N_1)$, or (b) if $\mathbf{m}^1(p_1) + \mathbf{m}^1(p_2) + \mathbf{m}^1(p_3) + \mathbf{m}^1(p_4) = 0$ then it must be that $\mathbf{m}^1(p_5)_{mod2} = 1$, consequently $(\mathbf{m}^2(p_5)_{mod2} = 1)$ and $\mathbf{m}^2 \in \Delta(N_2)$. The maximally permissive LESP for $N_1(\mathbf{m}_1^0)$ for any $\mathbf{m}_1^0 \in \Delta(N_1)$ is shown in figure 1(b).

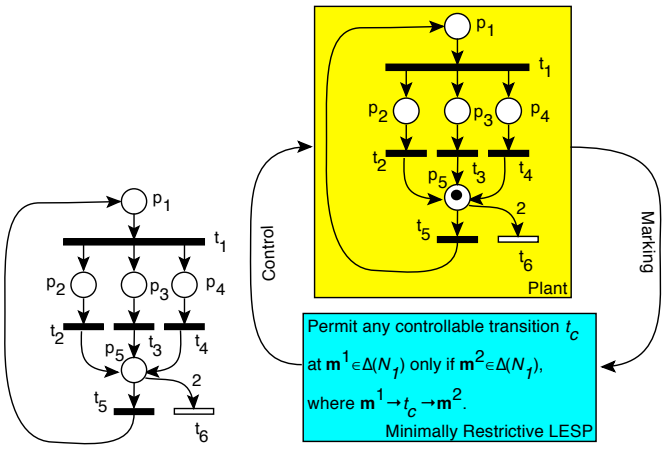
We call attention to the fact that the set $\Delta(N_1)$ is not right-closed as $(0\ 0\ 0\ 0\ 1)^T \in \Delta(N_1)$, but $(0\ 0\ 0\ 0\ 2)^T (\geq (0\ 0\ 0\ 0\ 1)^T) \notin \Delta(N_1)$.

The set $\Delta(N)$ is not computable in general.

Result 2: (Theorems 3.1 and 3.2, [12]) For an arbitrary PN $N(\mathbf{m}^0)$, and for an integral vector $\mathbf{m} \in \mathcal{N}^n$, neither "$\mathbf{m} \in \Delta(N)$?" nor "$\mathbf{m} \notin \Delta(N)$?" is semi-decidable[2]

[1] A PN structure $N = (\Pi, T, \Phi, \Gamma)$ is *Free-Choice* (FC) if $\forall p \in \Pi, (card(p^\bullet) > 1 \Rightarrow {}^\bullet(p^\bullet) = \{p\})$, where $card(\bullet)$ denotes the cardinality of the set argument. The PN structure N_1 is a *general* PN as $\Gamma(p_5, t_6) = 2$.

[2] That is, there is no program P that takes $\mathbf{m}$ as input, and eventually halts if and only if $\mathbf{m} \in \Delta(N)$ (resp. $\mathbf{m} \notin \Delta(N)$).



(a) N_1 (b) Maximally permissive LESP

Fig. 1. *General Free Choice* PN structures (a) $N_1 = (\Pi_1, T_1, \Phi_1, \Gamma_1)$, (b) The maximally permissive LESP for $N_1(\mathbf{m}_1^0)$ for any $\mathbf{m}_1^0 \in \Delta(N_1)$.

There are two consequences of result 2:

(*Consequence #1*) Neither membership, nor non-membership, of the set $\{N(\mathbf{m}^0) \mid \exists$ a LESP for $N(\mathbf{m}^0)\}$ is semi-decidable, which has discouraging, but interesting, consequences. For this we must digress into the realm of *random sets*. An n-bit binary-string $\omega \in \{0,1\}^n$ is said to be *random* if it is its own shortest description [18]. Since there are 2^n-many n-bit strings, and only $(2^n - 1)$-many "shorter-than-n" strings, there is at least one random n-bit string. Therefore, for each n, there is an n-bit random string (it is just that we can never be sure which one among the 2^n-many strings is random; because if its identity is known, it ceases to be random). If we were to collect all these random strings into a set, Ω, for reasons just described, neither membership, nor non-membership of the set Ω is semi-decidable. Getting back to the set of PNs that have LESPs – after taking away all subsets of this set where we have theoretical support for the existence/non-existence of policies, we will be left with a set that is random. Specifically, for PNs in this random set, there is no "rule-rhythm-or-rhyme" that accounts for why there is, or there is no, LESP. Following Chaitin [19], the property of liveness-enforcement by supervision that we seek is true/untrue by "*pure-accident*" as far as this set is concerned.

(*Consequence #2*) Any heuristic procedure for computing a LESP for an arbitrary PN $N(\mathbf{m}^0)$, will hang indefinitely for at least one instance where there is a LESP, and another instance for which there is no LESP. We must restrict attention to a class of problem instances if we are to automate the process of LESP synthesis. We choose PN structures for which the set $\Delta(N)$ is right-closed for this reason.

The following result identifies a large-family of PNs for which $\Delta(N)$ is right-closed.

Result 3: The set $\Delta(N)$ is right-closed if

1) all transitions in the PN structure N are controllable [20] (i.e $T_c = T$),
2) If N is an ordinary Free-Choice PN [12],
3) if N belongs to the family of general Free-Choice PNs $\mathcal{F}$, that is identified in reference [14],
4) if N belongs to the family of ordinary PNs $\mathcal{G}$ identified in reference [13], or
5) if N belongs to the family of general PNs $\mathcal{H}$ identified in reference [15].

The right-closure of $\Delta(N)$ for each of these families of PN structures is established by showing that if there is an LESP $\mathcal{P}$ for $N(\mathbf{m}^0)$ for some N that belongs to one the classes listed above, then using an explicit construction, it is shown that there is a LESP $\widehat{\mathcal{P}}$ for $N(\widehat{\mathbf{m}}^0)$ for any $\widehat{\mathbf{m}}^0 \geq \mathbf{m}^0$. The right-closed $\Delta(N)$ is denoted by its (finite set of) minimal elements $\min(\Delta(N))$.

Result 4: (Appendix, [20]) If N is any PN structure where all of its transitions are controllable (i.e. $T_c = T$), the finite set $\min(\Delta(N))$ can be computed.

As a consequence, the (set of minimal elements of the right-closed) set

$$\Delta_f(N) = \{\mathbf{m}^0 \in \mathcal{N}^n \mid \exists \text{a LESP for } N(\mathbf{m}^0) \text{ if } T_c = T\},$$

is computable for any N (and consequently, for any N that belongs to the families of PN structures identified in result 3). Additionally, $\Delta(N) \subseteq \Delta_f(N)$, for any N, and the set $\Delta_f(N)$ can be used as an initial starting-point for an iterative algorithm to compute $\Delta(N)$ in a few instances. The set $\Delta_f(N_1)$ for the PN shown in figure 1(a) is identified by the minimal elements $\{(1\ 0\ 0\ 0\ 0)^T, (0\ 1\ 0\ 0\ 0)^T, (0\ 0\ 1\ 0\ 0)^T, (0\ 0\ 0\ 1\ 0)^T, (0\ 0\ 0\ 0\ 1)^T\}$.

Result 5: (Lemma 5.10; figure 8, [12]) The largest subset of any right-closed set of marking $\mathcal{M} \subseteq \mathcal{N}^n$, that is control invariant with respect to a PN structure N can be computed.

The right-closed set $\mathcal{M} \subseteq \mathcal{N}^n$ is not control invariant with respect to N only when $\exists t_u \in T_u$, $\exists \mathbf{m} \geq \mathbf{m}^i$, where $\mathbf{m}^i \in \min(\mathcal{M})$, such that $\mathbf{m} \stackrel{t_u}{\rightarrow} \widehat{\mathbf{m}}$ in N and $\widehat{\mathbf{m}} \notin \mathcal{M}$ (cf. equation 4, [12]). In this case, the minimal element $\mathbf{m}^i$ is replaced by a set of new (and larger) minimal elements such that the firing of t_u at any marking in the new (and smaller) right-closed set will always result in a marking that is within the new (and smaller) right-closed set (cf. steps 4 and 5, figure [12]).

This brings us to the following result on the decidability of the existence of a LESP for $N(\mathbf{m}^0)$ for PN structures where $\Delta(N)$ is right-closed.

Result 6: If N is a PN structure such that $\Delta(N)$ is right-closed, then "$\mathbf{m}^0 \in \Delta(N)$?" is decidable.

This result uses the fact that (1) $\Delta(N)$ is control invariant with respect to N, and (2) there is a path condition on the *coverability graph* of the supervised PN of $N(\widetilde{\mathbf{m}})$, where $\widetilde{\mathbf{m}} \in \min(\Delta(N))$ (cf. [12] for details).

The algorithm that decides "$\mathbf{m}^0 \in \Delta(N)$?" proceeds in an iterative fashion. At the i-th stage of the iteration, it uses a right-closed closed set $\widehat{\Delta}_i(N)$, where $\Delta(N) \subseteq \widehat{\Delta}_i(N)$. Initially, $\widehat{\Delta}_0(N) = \Delta_f(N)$, and $\mathbf{m}^0 \notin \widehat{\Delta}_i(N) \Rightarrow \mathbf{m}^0 \notin \Delta(N)$.

If $\mathbf{m}^0 \in \widehat{\Delta}_i(N)$ and (1) $\widehat{\Delta}_i(N)$ is control invariant with respect to N, and (2) each member of $\min(\widehat{\Delta}(N))$ satisfies

the path-condition listed above, then the algorithm terminates with $\Delta(N) = \widehat{\Delta}_i(N)$.

If $\mathbf{m}^0 \in \widehat{\Delta}_i(N)$, but $\widehat{\Delta}_i(N)$ is not control invariant with respect to N, then $\widehat{\Delta}_i(N)$ is replaced by $\widehat{\Delta}_{i+1}(N)$, which is the largest subset of $\widehat{\Delta}_i(N)$ that is control invariant with respect to N (cf. result 5), and the process is repeated again.

If $\mathbf{m}^0 \in \widehat{\Delta}_i(N)$, and some $\mathbf{m}^i \in \min(\widehat{\Delta}(N))(\subseteq \mathcal{N}^n)$ violates the the path-condition listed above, then $\mathbf{m}^i$ is elevated by n-many unit-vectors, which in turn defines $\widehat{\Delta}_{i+1}(N)$, and $\widehat{\Delta}_{i+1}(N) \subset \widehat{\Delta}_i(N)$. The process is repeated again. This procedure forms the crux of the software package described in references [10], [11].

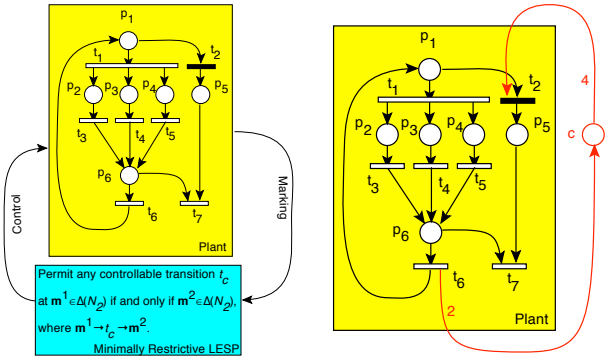
An alternate approach to liveness enforcement involves augmenting the structure of the PN N by the addition of extra places (i.e. *monitors*) and arcs between the transitions in the PN and the monitors. The objective is to find an appropriate augmentation, along with an initial assignment of tokens to the monitors such that the resulting augmented PN is live. It is also mandatory to ensure that no uncontrollable transition is ever prevented from firing due a lack of sufficient tokens in any monitor place that is connected to it. The control action effected by this construction implicitly defines a supervisory policy. A liveness enforcing monitor construction is deemed minimally restrictive if this implicitly defined supervisory policy is maximally permissive.

There is a significant amount of literature on a class of monitors called *invariant-based monitors* (cf. [21], [22], [23]), where the extra arcs and places that are added to the PN structure N ensure that for any reachable marking $\mathbf{m}$ of the augmented PN, a matrix inequality of the form $\mathbf{Lm} \geq \mathbf{a}$ is satisfied (i.e the property $\mathbf{Lm} \geq \mathbf{a}$ is *invariant* for all reachable markings). The following result identifies a necessary and sufficient condition for the existence of a liveness enforcing monitor construction that is maximally permissive.

Result 7: [7] There is an invariant-based, maximally permissive, liveness enforcing monitor construction for a PN $N(\mathbf{m}^0)$ if and only if $\Delta(N)$ is convex.

$\Delta(N_1)$ is not convex, as a result there can be no invariance-based monitor that is equivalent to the maximally permissive LESP of figure 1(b). It is not known if there is an invariance-based monitor that can enforce liveness in $N_1(\mathbf{m}_1^0)$ even when $\mathbf{m}_1^0$ belongs some (strict) subset of $\Delta(N_1)$.

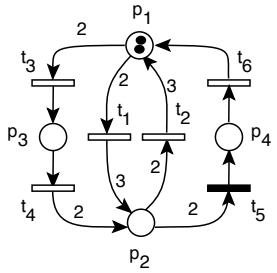
The PN structure N_2 shown in figure 2(a) (and 2(b)) has all transitions but t_2, as uncontrollable transitions. It is an *Asymmetric-Choice* PN structure (cf. [24] for details). The set $\Delta(N_2) = \Delta_f(N_2) - \{\mathbf{m} \in \mathcal{N}^6 \mid \mathbf{m}(p_5) \geq 3\mathbf{m}(p_1)+\mathbf{m}(p_2)+\mathbf{m}(p_3)+\mathbf{m}(p_4)+\mathbf{m}(p_6)\} = \{\mathbf{m} \in \mathcal{N}^6 \mid 3\mathbf{m}(p_1) + \mathbf{m}(p_2) + \mathbf{m}(p_3) + \mathbf{m}(p_4) + \mathbf{m}(p_6) - \mathbf{m}(p_5) \geq 1\}$. Additionally, $\Delta(N_2)$ is not right-closed. The maximally permissive LESP for $N_2(\mathbf{m}_2^0)$ for any $\mathbf{m}_2^0 \in \Delta(N_2)$ is shown in figure 2(a). Since $\Delta(N_2)$ is convex, it follows that there is an invariance-based monitor that is equivalent to the maximally permissive LESP of figure 2(a), and it is shown in figure 2(b). The monitor place c, with weighted arcs $\{(t_6, c), (c, t_2)\}$ represents the monitor place construction. This construction enforces liveness for any initial marking $\mathbf{m}_2^0 \in \Delta(N_2)$ (i.e. $3\mathbf{m}_2^0(p_1)+\mathbf{m}_2^0(p_2)+\mathbf{m}_2^0(p_3)+\mathbf{m}_2^0(p_4)+\mathbf{m}_2^0(p_6) - \mathbf{m}_2^0(p_5) \geq 1$).



(a) Maximally permissive LESP (b) Invariance-Based Monitor

Fig. 2. (a) The maximally permissive LESP for $N_2(\mathbf{m}_2^0)$ for any $\mathbf{m}_2^0 \in \Delta(N_2)$ (b) An invariance-based monitor that is equivalent to the maximally permissive LESP of figure 2(a). The initial token load of the monitor place c is given by the expression $3\mathbf{m}_2^0(p_1) + \mathbf{m}_2^0(p_2) + \mathbf{m}_2^0(p_3) + \mathbf{m}_2^0(p_4) + \mathbf{m}_2^0(p_6) - \mathbf{m}_2^0(p_5) - 1$.

The input file for the software described in reference [11] that describes the PN $N_3(\mathbf{m}_3^0)$ of figure 3(a) is shown in figure 3(b). Figure 4 shows the output generated by the software, which lists the five minimal elements of $\min(\Delta(N_3))$.



(a) $N_3(\mathbf{m}_3^0)$

```
4 6
2 0 2 0 0 0
0 2 0 0 2 0
0 0 0 1 0 0
0 0 0 0 0 1
0 3 0 0 0 1
3 0 0 2 0 0
0 0 1 0 0 0
0 0 0 0 1 0
2 0 0 0
0 0 0 0 1 0
```

(b) Input File

Fig. 3. (a) The general PN $N_3(\mathbf{m}_3^0)$, where $N_3 \in \mathcal{F}$ ($\Rightarrow \Delta(N_3)$ is right-closed)[14]. (b) The input file for the software described in [11] that describes $N(\mathbf{m}_3^0)$. There are $n = 4$ places and $m = 6$ transitions in this PN, which is represented by the first line. This is followed by the $n \times m$ **IN** and $n \times m$ **OUT** matrices. The n-long line that follows these matrices represents the initial marking $\mathbf{m}_3^0$, which places two tokens in p_1. The m-long line in the end of the file indicates, through a sequence of 1's (0's) the controllable (uncontrollable) transitions in N. t_5 is only controllable transition.

Testing the convexity of an arbitrary $\mathcal{M} \subseteq \mathcal{N}^n$ can only be done in an approximate-sense (cf. [25]). However, if $\Delta(N)$ is right-closed, its convexity can be tested in an exact-sense.

Result 8: [7] The convexity of a right-closed set $\mathcal{M} \subseteq \mathcal{N}^n$ is decidable.

This procedure involves checking the minimal elements of the convex hull of the set $\mathcal{V} = \min(\mathcal{M}) \cup \{\mathbf{m}^i + \mathbf{1}_j \mid \mathbf{m}^i \in \min(\mathcal{M}), \mathbf{1}_j$ is an unit-vector$\}$, which is the minimal elements of $\mathcal{M}$ along with their elevated counterparts. The set $\mathcal{M}$ is convex if and only if $\min(\mathcal{M}) = \min(Int(conv(\mathcal{V})))$,

```
Input File = "pn1"

Incidence Matrix :

   T  1  2  3  4  5  6
P
  1  -2  3 -2  .  .  1
  2   3 -2  .  2 -2  .
  3   .  .  1 -1  .  .
  4   .  .  .  .  1 -1

Initial Marking : ( 2 0 0 0 )

There is an LESP for this (fully controlled) PN

_____________________________________________


Minimal Elements of the fully controlled Net
--------------------------------------------

 1: ( 0 0 1 0 )
 2: ( 1 0 0 1 )
 3: ( 0 0 0 2 )
 4: ( 0 2 0 0 )
 5: ( 2 0 0 0 )


List of Controllable Transitions
--------------------------------
t5


(Final) Minimal Elements of the control-invariant set
-----------------------------------------------------

 1: ( 0 0 1 0 )
 2: ( 1 0 0 1 )
 3: ( 0 0 0 2 )
 4: ( 0 2 0 0 )
 5: ( 2 0 0 0 )

This is An LESP
```

Fig. 4. The output file generated from the input file of figure 3(b).

where $conv(\bullet)$ denotes the convex-hull of the point-set argument, and $Int(\bullet)$ denotes the set of integral vectors in the polyhedron argument.

The convexity of $\Delta(N_3)$, whose minimal elements are identified in figure 4, can be computed using the convex hull of the set $\mathcal{V}_4$, which consists of 25 $(= 5 + (5 \times 4))$ integral vectors. There are 27 integral vectors in $conv(\mathcal{V})$, which are shown in the Polymake[3] output of figure 5. The minimal elements of the 27 vectors shown in this figure can be picked by visual inspection. The members of $\min(Int(conv(\mathcal{V})))$, are $\{(0\ 0\ 0\ 2),\ (0\ 0\ 1\ 0),\ (0\ 1\ 0\ 1),\ (0\ 2\ 0\ 0),\ (1\ 0\ 0\ 1),\ (1\ 1\ 0\ 0),\ (2\ 0\ 0\ 0)\}$ $(\neq \min(\Delta(N_3)))$. Therefore, $\Delta(N_3)$ is not convex, which can be verified as follows

$$\underbrace{(0\ 1\ 0\ 1)}_{\in conv(\Delta(N_3))-\Delta(N_3))} = \frac{1}{2} \times \underbrace{(0\ 2\ 0\ 0)}_{\in\Delta(N_3)} + \frac{1}{2} \times \underbrace{(0\ 0\ 0\ 2)}_{\in\Delta(N_3)},$$

As a consequence of result 8 we know that *there is no maximally permissive, invariant-based monitor that enforces liveness for this PN*. Stated differently – every invariant-based monitor that enforces liveness in $N_3(\mathbf{m}_3^0)$ will be more restrictive than the one that is based on $\Delta(N_3)$, which is shown in figure 6(a). This maximally permissive LESP can also be represented equivalently using a *Disjunctive Normal Form* (DNF) formula on the markings, as shown in figure 6(b). This is generalized in the following result, and one of its consequences is that if $\Delta(N)$ is right-closed, then the maximally permissive LESP can be implemented using *threshold sensors*[4] in places.

[3]www.polymake.org

[4]A α-threshold sensor in place $p \in \Pi$ outputs a "1" if the marking satisfies the inequality $\mathbf{m}(p) \geq \alpha$. That is, it detects if there are at least α-many tokens in p.

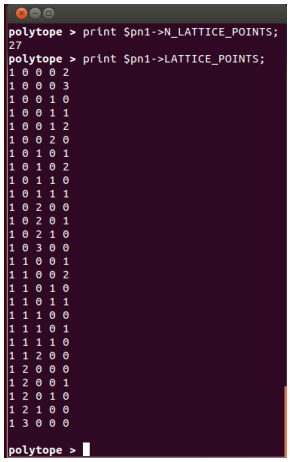


Fig. 5. Polymake output that shows there are 27 integral vectors in the polytope identified by the 25 integral vectors of $\mathcal{V}$, which was constructed from $\min(\Delta(N))$ and the five unit-vectors.

Result 9: [16] If $\Delta(N)$ is right-closed for N, then the maximally permissive LESP can be characterized using a collection of boolean expressions $\{\Theta_{t_c}(N)\}_{t_c \in T_c}$. This maximally permissive LESP permits $t_c \in T_c$ at a marking $\mathbf{m}$ iff $\Theta_{t_c}(N)$ evaluates to "$TRUE$" at $\mathbf{m}$.

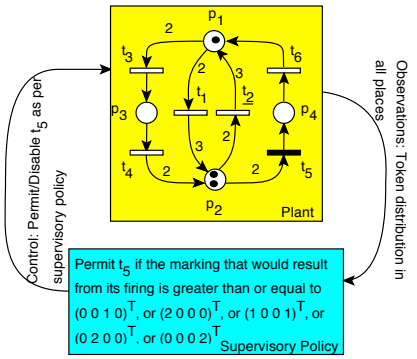
As noted above, this result presents a sufficient information structure for the implementation of a maximally permissive LESP in those cases where $\Delta(N)$ is right-closed. This result also provides the basis for monitor based LESPs that are not necessarily invariant-based, which as noted in result 7, has its limitations. For instance, there can be no invariant-based monitor that enforces liveness that is equivalent to the DNF-based LESP shown in figure 6(b).

Unlike monitor-based implementations, which require the sensing of transitions whose firing can change the token-load of monitor places, the DNF-based implementation of the maximally permissive LESP determines the control-action based on *just* the outputs of the threshold sensors in select places. This implementation advantage makes it the preferred choice over monitors. The connections between these LESPs and other techniques for enforcing disjunctive specifications [26] is suggested as a future research topic.

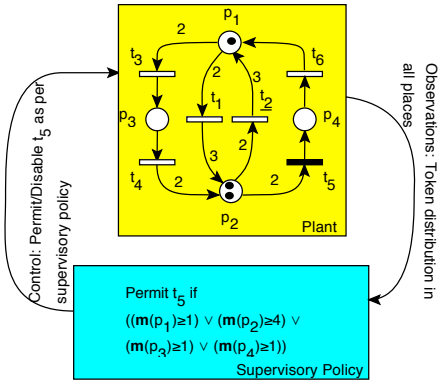
IV. Conclusion

In this paper we consolidated results that are pertinent to *liveness enforcing supervisory policies* (LESPs) in PNs. These results were illustrated using a series of examples in a semi-formal setting, and were intended to clarify the theoretical concepts in references [12], [13], [14], [15], [7]. This expository treatment via examples was possible with the help of the software described in references [11].

Our ongoing software development activities involve the synthesis of invariant-based monitors that are equivalent to LESPs that use the $\Delta(N)$-set described in this paper. Additionally, we are exploring automating "divide-and-conquer" approaches (cf. [27], [28], for example) to the software that computes the members of $\min(\Delta(N))$ when N belongs to a family of PN structures where $\Delta(N)$ is known to be



(a) Maximally permissive LESP for $N_3(\mathbf{m}_3^0)$



(b) The DNF-based maximally permissive LESP for $N_3(\mathbf{m}_3^0)$

Fig. 6. (a) The maximally permissive LESP for $N_3(\mathbf{m}_3^0)$ for any $\mathbf{m}_3^0 \in \Delta(N_3)$. There is no invariant-based monitor that is equivalent to this LESP, as $\Delta(N_3)$ is not convex. (b) A DNF-based LESP for $N_3(\mathbf{m}_3^0)$, where $\mathbf{m}_0^3 \in \Delta(N_3)$, that is equivalent to the LESP shown in figure 6(a).

right-closed. On the theoretical front, we are engaged in identifying additional families of PN structures for which the $\Delta(N)$-set is right-closed.

REFERENCES

[1] B. Alpern and F. B. Schneider, "Defining liveness," *Information Processing Letters*, vol. 21, no. 4, 1985.

[2] S. Little, N. Seegmiller, D. Walter, C. Myers, and T. Yoneda, "Verification of analog/mixed-signal circuits using labeled hybrid petri nets," in *Proceedings of the ICCAD-2006*, 2006.

[3] Z. Aspar, M. Khalil-Hani, and N. Shaikh-Husin, "Deadlock detection and avoidance using signal interpreted petri nets," in *2012 IEEE International Conference on Circuits and Systems (ICCAS)*, 2012, pp. 150–155.

[4] F. Garcia and A. Sanchez, "Formal verification of safety and liveness properties for logic controllers. a tool comparison," in *3rd International Conference on Electrical and Electronics Engineering*, 2006, pp. 1–2.

[5] K. Venkatesh, M. Zhou, and R. Caudill, "Comparing ladder logic diagrams and petri nets for sequence controller design through a discrete manufacturing system," *IEEE Transactions on Industrial Electronics*, vol. 41, no. 6, pp. 611–619, 1994.

[6] J.-S. Lee and P.-L. Hsu, "An improved evaluation of ladder logic diagrams and petri nets for the sequence controller design in manufacturing systems," *Int J. Adv. Manuf. Technol.*, vol. 24, pp. 279–287, 2004.

[7] E. Salimi and R. Sreenivas, "On invariant-based monitors that enforce liveness in a class of partially controlled general petri nets," *IEEE Transactions on Automatic Control*, October 2014, conditionally accepted.

[8] M. Gomaa, "Petri net to ladder logic diagram converter and a batch process simulation," *ARPN Journal of Engineering and Applied Sciences*, vol. 6, no. 2, pp. 67–72, February 2011.

[9] M. Uzam, A. Jones, and N. Ajlouni, "Conversion of petri net controllers for manufacturing systems into ladder logic diagrams," *IEEE Conference on Emerging Technologies and Factory Automation (EFTA 96)*, vol. 2, pp. 649–655, 1996.

[10] S. Chandrasekaran and R. Sreenivas, "On the automatic generation of the minimally restrictive liveness enforcing supervisory policy for manufacturing- and service-systems modeled by a class of general free choice petri nets," in *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC-13)*, Paris, France, April 2013, session WeC01.3.

[11] S. Chandrasekaran, N. Somnath, and R. Sreenivas, "A Software Tool for the Automatic Synthesis of Minimally Restrictive Liveness Enforcing Supervisory Policies for a class of General Petri Net models of Manufacturing- and Service-Systems," *Journal of Intelligent Manufacturing*, 2014, to appear (DOI 10.1007/s10845-014-08885).

[12] R. Sreenivas, "On the existence of supervisory policies that enforce liveness in partially controlled free-choice petri nets," *IEEE Transactions on Automatic Control*, vol. 57, no. 2, pp. 435–449, February 2012.

[13] ——, "On a decidable class of partially controlled petri nets with liveness enforcing supervisory policies," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 5, pp. 1256–1261, August 2013.

[14] N. Somnath and R. Sreenivas, "On Deciding the Existence of a Liveness Enforcing Supervisory Policy in a Class of Partially-Controlled General Free-Choice Petri Nets," *IEEE Transactions on Automation Science and Engineering*, vol. 10, pp. 1157–1160, October 2013.

[15] E. Salimi, N. Somnath, and R. Sreenivas, "Some observations on the maximally permissive liveness enforcing supervisory policy for a class of weighted petri nets," *Automatica*, November 2013, submitted.

[16] V. Deverakonda and R. Sreenivas, "On a Sufficient Information Structure for Supervisory Policies that Enforce Liveness in a Class of General Petri Nets," *IEEE Transactions on Automatic Control*, 2014, to appear.

[17] R. Valk and M. Jantzen, "The residue of vector sets with applications to decidability problems in Petri nets," *Acta Informatica*, vol. 21, pp. 643–674, 1985.

[18] M. Li and P. Vitanyi, *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 1997.

[19] G. Chaitin, *The Limits of Mathematics: A Course on Information Theory and the Limits of Formal Reasoning*. Springer, 2002.

[20] R. Sreenivas, "On the existence of supervisory policies that enforce liveness in discrete-event dynamic systems modeled by controlled Petri nets," *IEEE Transactions on Automatic Control*, vol. 42, no. 7, pp. 928–945, July 1997.

[21] A. Giua, "Petri nets as discrete event models for supervisory control," Ph.D. dissertation, ECSE Dept., Rensselaer Polytechnic Institute, Troy, NY., 1992.

[22] J. Moody and P. Antsaklis, *Supervisory Control of Discrete Event Systems using Petri Nets*. MA: Kluwer Academic Publishers, 1998.

[23] M. Iordache and P. Ansaklis, *Supervisory control of Concurrent Systems: A Petri net Structural Approach*. MA: Kulwer Academic Publishers, 2006.

[24] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[25] L. Rademacher and S. Vempala, "Testing geometric convexity," in *Proceedings of FSTTCS 2004*, IMSc, Chennai, 2004.

[26] M. V. Iordache, P. Wu, F. Zhu, and P. J. Antsaklis, "Efficient design of petri-net supervisors with disjunctive specifications," in *Proceedings of the 9th IEEE International Conference on Automation Science and Engineering (CASE 2-13)*, Madison, WI, 2013.

[27] R. Sreenivas, "On supervisory policies that enforce liveness in completely controlled petri nets with directed cut-places and cut-transitions," *IEEE Transactions on Automatic Control*, vol. 44, no. 6, pp. 1221–1225, June 1999.

[28] ——, "On supervisory policies that enforce liveness in a class of completely controlled petri nets obtained via refinement," *IEEE Transactions on Automatic Control*, vol. 44, no. 1, pp. 173–177, January 1999.