Liveness for Synchronized Choice Petri Nets

DANIEL Y. CHAO[1] AND JOSE A. NICDAO[2]

[1]*Department of Management and Information Science, National Cheng Chi University, Taipei, Taiwan, ROC*
[2]*Institute of Information Industry, Taipei, Taiwan*
Email: yaw@mis.nccu.edu.tw

Unlike traditional classification by output conditions of places, synchronized choice nets were defined as a new class of nets characterized by local structures. This paper investigates its liveness. The time required to examine local structures is less than that required to examine global structures (via structure objects). Thus polynomial time algorithms can be developed to verify the liveness property of Petri nets.

Received 11 August 1999; revised 1 December 2000

1. INTRODUCTION

Proving liveness or, equivalently, solving the reachability problem for general Petri nets (PNs) is a difficult problem and takes exponential time and space [1]. Traditionally, PNs are classified into marked graphs (MGs), state machines (SMs), free-choice (FC) nets,[2] extended free-choice (EFC) nets[3] and asymmetric-choice (AC) nets[4] based on local structures such as input and output sets of transitions or places. Net behavior such as liveness (no blocking) and boundedness (e.g. no buffer overflow), however, is correlated to special structural objects such as circuits (loops of nodes) for MGs [2], transition–place (TP) and place–transition (PT) handles and bridges [3], deadlocks or siphons, and traps [4] for FC nets which are more global than the above local structures that classify them.

Roughly speaking, a 'handle' is an alternate disjoint path between two nodes. A PT handle starts with a place and ends with a transition while a TP handle starts with a transition and ends with a place. A TP bridge is a path from a transition in the handle to a place in the circuit, while a PT bridge is a path from a place in the handle to a transition in the circuit. These bridges help 'synchronize' the workings of the bridge and the handle.

Both siphons and traps are a set of places. Roughly, once a token leaves a siphon (enters a trap) then it will never return to the siphon (leak out from the trap). If tokens continue to leave, and eventually the siphon is empty of tokens, no input transition of a place in the siphon can be fired again. Hence the net is dead. Thus, deadlocks and traps have been proved to be very useful in analyzing FC but do not appear to be meaningful when the PN synthesis is considered. Esparza and Silva in [3] complemented them with other objects, handles and bridges, which led to results having a clear intuitive meaning. It was shown that in the absence of TP and PT handles a FC is both live and bounded.

However, for general cases of FC, EFC, AC and beyond, the global nature of handles and bridges makes efficient analysis unattainable. We solve the problem by localizing all handles and bridges. That is, if two handles have identical end nodes, then we consider only the case where there is at most one bridge from one handle to another. By investigating all possible variations of such local structures, we are able to find the liveness conditions for synchronized choice (SNC) which is a new class of nets [5] not covered by AC and has some constraints on the above local structures.

Another new structural approach is synthesis-directed analysis. Synthesis [6, 7, 8, 9, 10, 11] expands a net N^1 (solid lines in Figures 1–4) following a set of rules such that after each expansion (dashed lines), the net N^2 remains well behaved (both live and bounded). For instance, in the knitting technique [5], a larger net can be constructed from a simple circuit by continuously adding a set of new paths of handles and bridges at each synthesis step according to transition–transition (TT) and place–place (PP) rules. That is, each new path involved in a synthesis step must be a TT (from transition to transition) or a PP (from place to place) path. Depending on the structural relationship (concurrent, exclusive or sequential) between the two end nodes of the new path, an appropriate rule should be applied. For example, a rule may be such as to forbid the new path generation, or to add more. The net will be complete when all new path generations are permitted while keeping the net well behaved. The rules involve nodes in a local fashion. They provide clues on structural constraints for a net to be well behaved and thus offer a new avenue for net analysis.

Examining the synthesis rules presented in [6, 7, 8, 9], we find that synthesized nets and SNC nets are closely related.

[1]The former name of the author is Yuh Yaw, which appeared in some of his earlier papers.
[2]$\forall p_1, p_2 \in P, p_1 \bullet \cap p_2 \bullet \neq 0 \Rightarrow |p_1 \bullet| = |p_2 \bullet| = 1$.
[3]$\forall p_1, p_2 \in P, p_1 \bullet \cap p_2 \bullet \neq 0 \Rightarrow p_1 \bullet = p_2 \bullet$.
[4]$\forall p_1, p_2 \in P, p_1 \bullet \cap p_2 \bullet \neq 0 \Rightarrow p_1 \bullet \subseteq p_2 \bullet$ or $p_1 \bullet \supseteq p_2 \bullet$.

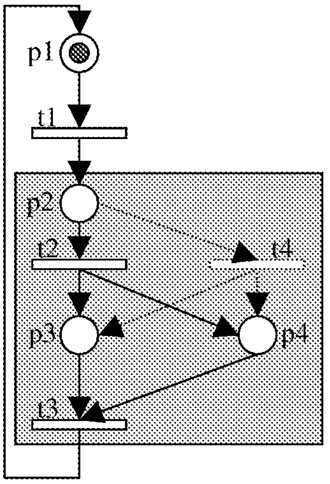

FIGURE 1. An example of live and reversible SNC with no inconsistent pair.

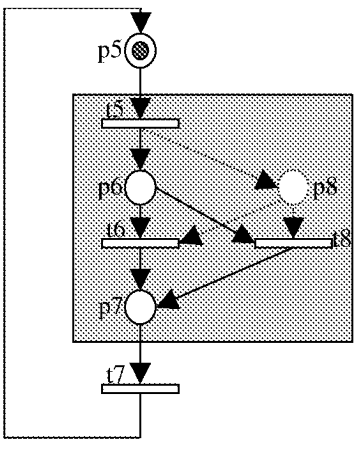

FIGURE 2. Dual of the net in Figure 1. This net is live and reversible without an inconsistent pair.

This is because both involve handles and bridges; the former are constructed by adding new handles and bridges while the latter are based on local structures of handles and bridges.

These local structures play a role not only in the classification but also in the behavior characterization of SNC. Since they are local, they could be searched in polynomial time and efficient algorithms for verifying a net to be SNC and its liveness could be developed—it takes less time to look into a local than a global structure. These local structures determine not only [6] whether an SNC suffers from deadlocks but also other properties. They also simplify the synthesis rules. More importantly, the new synthesis rules can generate more SNC; in other words, the old rules [5] are enhanced.

After introducing the preliminaries in Section 2 and second-order structures in Section 3, we will be able to define SNC more clearly and explain the differences between our approach and that of others. Section 3 also compares SNC with two other classes of nets. Section 4 finds liveness conditions and two kinds of minimal deadlocks. One of them is not a trap in the presence of an inconsistent pair of places (Section 5) and is the sole cause of SNC not being live. We will then be able to provide a formal proof (Section 6) of liveness conditions based on the concept of deadlocks and traps. An integrated algorithm is presented for verification of SNC and liveness in Section 7.

2. PRELIMINARIES

We assume that readers are familiar with the various terminology of PNs; references for these can be found in [2].

DEFINITION 1. *Let* $P = \{p_1, p_2, \ldots, p_a\}$, $T = \{t_1, t_2, \ldots, t_b\}$, *with* $P \cup T \neq \varnothing$ *and* $P \cap T = \varnothing$; $F \subseteq (P \times T) \cup (T \times P)$ *then* $N = (P, T, F)$ *is a net.* $p_i (1 \leqq i \leqq a)$ *is called a place,* t_i $(1 \leqq i \leqq b)$ *a transition, and* M_0 *is an initial marking whose ith component,* $M_0(p_i)$*, represents the number of tokens in place* p_i. $I(U)$ $[O(U)]$ *denotes the set of input (output) nodes of all nodes in* U *which is a node or a set of nodes in* N. t_i *is fireable if each place in* $I(t)$ *holds more than one token.*

DEFINITION 2. *Let* $N = (P, T, F)$ *be a net,* (N, M_0) *be a marked net and* $R(M)$ *the set of markings reachable from* M,

1. *A transition* $t \in T$ *is live under* M_0 *iff* $\forall M \in R(M_0)$, $\exists M \in R(M)$, *t is firable under* M'.
2. *A transition* $t \in T$ *is dead under* M_0 *iff* $\nexists M \in R(M_0)$ *where t is firable.*
3. *A net PN is live under* M_0 *iff* $\forall t \in T$, *t is live under* M_0. *It is bounded if* $\forall M \in R(M_0)$, $\forall p \in P$, *the marking at p,* $M(p)$ *is bounded.*
4. *A net N is structurally live (SL, live for a certain initial state) iff* $\exists M_0$ *such that PN is live under* M_0. *N is structurally bounded (SB, bounded for all initial states) if it is bounded for any finite initial marking* M_0.
5. *A net N is said to be reversible if, for each marking M in* $R(M_0)$, M_0 *is reachable from M; i.e.* $M_0 \in R(M)$. *N is structurally reversible if* $\forall M_0$ *such that N is reversible.*

DEFINITION 3. *A node x in* $N = (P, T, F)$ *is either a* $p \in P$ *or a* $t \in T$. *The post-set of node x is* $x\bullet = \{y | \exists$ *an arc* $(x, y)\}$, *and its pre-set* $\bullet x = \{y | \exists$ *an arc* $(y, x)\}$. *An elementary directed path* Γ *in N is a sequence of nodes:* $\Gamma = [n_1, n_2, \ldots, n_k]$, $k \geq 1$, *such that* $n_i \in \bullet n_{i+1} 1 \leq i < k$ *if* $k > 1$, *and* $n_i = n_j$ *which implies that* $i = j$, $\forall 1 \leq i, j \leq k$. *A path is (non) virtual if it contains only (more than) two nodes. An elementary cycle in N is* $\Gamma = [n_1, n_2, \ldots, n_k]$, $k > 1$ *such that* $n_i = n_j$, $1 \leq i \leq j \leq k$, *implies that* $i = 1$ *and* $j = k$.

In this paper, we consider only strongly connected nets where there exist directed paths between any pair of nodes.

DEFINITION 4. n_a *is a terminal node of a subnet* N' *if either* $\bullet n_a \cap N' = \varnothing$ *or* $n_a \bullet \cap N' = \varnothing$. $P_{N'}(\tau_{N'})$ *is the set of (terminal) places in* N'. $P^1_{N'} = P_{N'} \backslash \tau_{N'}$ *is the set of internal places in* N'.

When X and Y are each a set of places, $X \backslash Y$ is the set of places in X but not in Y. $X \backslash Y$ (Definition 5) is, however, defined differently when X and Y are subnets.

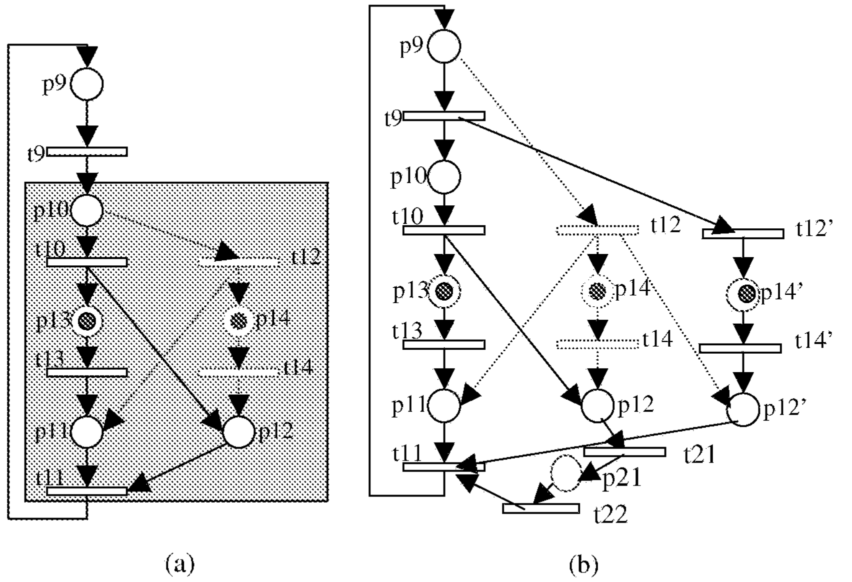

FIGURE 3. (a) Irreversible SNC with PT-inconsistent pair $(p13, p14)$. (b) $H = [t10, p21]$, $N_1 = N\backslash H$; (rule 2) before: $\neg(p10 \leftrightarrow p21)$; after: $p10 \leftrightarrow p21$. (Rule 3) before: $n_s^{13,21} = p9$; after: $n_s^{13,21} = t10$. (Rule 4) before: $n_s^{14',21} = p9$; after: $n_s^{14',21} = t9$. (c) No longer irreversible. One $n_s^{13,21} = t12'$; $(p13, p14)$ no longer a PT-inconsistent pair.

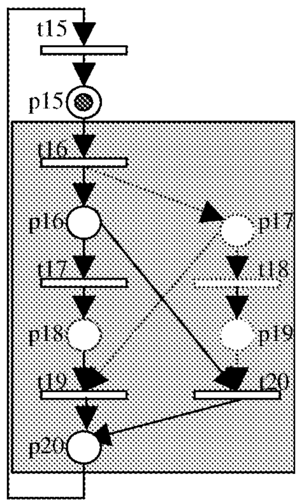

FIGURE 4. Dual of the net in Figure 3a. The SNC is not live with TP-inconsistent pair $(p18, p19)$.

DEFINITION 5. *Let $N' = X \cup Y$. If $X \cap Y$ is a set of terminal nodes, then $N'\backslash X = Y$ and $N'\backslash Y = X$.*

DEFINITION 6. ([12]) *For a PN (N, M), a non-empty subset D of places is called a deadlock if $\bullet D \subseteq D\bullet$, i.e. every transition having an output place in D has an input place in D. If $M(D) = \sum_{p \in D} M(p) = 0$, D is called a token-free deadlock at M. A minimal deadlock does not contain a deadlock as a proper subset. It is called a trap if $\bullet D \subseteq D\bullet$, i.e. every transition having an input place in D has an output place in D.*

For a token-free deadlock, the following lemma was obtained in [12].

LEMMA 1. ([12]) *For a PN (N, M_0), if there does not exist any firable transition, then there exists a token-free deadlock at M_0.*

DEFINITION 7. ([12]) *A subnet $N_i = (P_i, T_i, F_i)$ of N is an I-subnet (O-subnet) of N generated by P_i, if $P_i \subseteq P$, $T_i = \bullet P_i (P_j \bullet)$ and $F_i = F \cap (P_i \times T_i) \cup (T_i \times P_i)$.*

An I-subnet of a minimal deadlock is structurally characterized by the following lemma.

LEMMA 2. ([13]) *The I-subnet of a PN generated by a minimal deadlock D in the PN is strongly connected.*

COMMONER'S PROPERTY. ([14]) Let (N, M_0) be a marked net. (N, M_0) satisfies the Commoner property, iff the following two conditions hold:

(1) every minimal deadlock of N contains a trap;
(2) the maximal trap (not properly contained in any other trap) of each minimal deadlock is marked for M_0.

Condition (1) is referred to as the structural Commoner's property. We will show that a live SNC satisfies the structural deadlock–trap property.

3. HANDLES, BRIDGES, FIRST- AND SECOND-ORDER STRUCTURES

We follow [5] for the definitions of handles, bridges, XY-handles, and XY-bridges where X and Y can be T or P.

DEFINITION 8. *Let $N = (P, T, F)$ and N_1, N_2 be partial subnets of N. An elementary directed path $H = [n_s, n_1, n_2, \ldots, n_k, n_e]$, $n_i \in P \cup T$, $i = 1, 2, \ldots, k$, is called a handle of N_1 if $H \cap N_1 = \{n_s, n_e\}$; n_s and n_e*

are called the start and the end nodes of the handle H, respectively. Note that n_s and n_e may be identical. An elementary directed path $B = [n_1, \ldots, n_r]$, $r \geqq 2$, is a bridge from N_1 to N_2 iff $B \cap (P_1 \cup T_1) = \{n_1\}$ and $B \cap (P_2 \cup T_2) = \{n_r\}$. $p_1 \leftrightarrow p_2$ if p_1 and p_2 are on an elementary circuit. $n_1 \rightarrow n_2$ if $n_1 \leftrightarrow n_2$ and there is an elementary directed path from n_1 to n_2.

DEFINITION 9. *Handles H_1 and H_2 are said to be mutually complementary if they share the same start node n_s and end node n_e; i.e. $H_1 \cap H_2 = \{n_s, n_e\}$. Let $p_i \in H_i$ ($i = 1$ or 2), $p_i \neq n_s$ and $p_i \neq n_e$, then define $\mu_i \subset H_i$ as a directed path on H_i from n_s to p_i; n_s is the nearest start node of (p_1, p_2); i.e. $n_s^{1,2} = n_s$, if there exists at least one μ_i on a certain H_i that contains no other $n^s_{1,2}$ of another H_j; the nearest end node of (p_1, p_2), $n_e^{1,2}$ can be defined in a dual fashion and $n_e^{1,2} = n_e$ if there exists at least one ν_i on a certain H_i that contains no other $n_e^{1,2}$ of another H_j, where $\nu_i \subset H_i$ is a directed path on H_i from p_i to n_e. (p_1, p_2) is called a TP-inconsistent pair of places if $\exists n_s^{1,2}$ is a transition and $\exists n_e^{1,2}$ is a place. (p_1, p_2) is called a PT-inconsistent pair of places if $\exists n_s^{1,2}$ is a place and $\exists n_e^{1,2}$ is a transition. Let $\Upsilon = H_1 \cup H_2$, $H_1 \subset N_1$ and $H_2 \subset N_2$, H_1 (H_2) is a prime handle to H_2 (H_1). (i) If there are no bridges B between H_1 and H_2 and Υ is defined to be a first-order structure (FOS). If $n_s \in X$ and $n_e \in Y$ where $X, Y = T$ or P, then $\Gamma(H, B)$ is said to be an XY-path (XY-handle, XY-bridge). If $X = Y$, then the FOS Υ is said to be symmetrical; otherwise it is asymmetrical (AFOS). (ii) If B_{12} (B_{21}) is the only bridge from H_1 to H_2 (H_2 to H_1), then $\varphi = H_1 \cup H_2 \cup B_{12} \cup B_{21}$ is defined to be a second-order structure (SOS) (see the shaded area in Figures 1–4). (iii) A strongly connected net is SNC if it satisfies the two requirements R1 and R2 where R1 (R2) is: every TP- (PT-) handle to a certain circuit has a PT- (TP-) bridge from its complementary TP-handle to itself.*

In Figure 11c, $n_e^{6,8} = \{t_6, p_7\}$; p_7 is a nearest end node $n_e^{6,8}$ because there are no other end nodes n_e on $\nu = [p8, t8, p7]$ (on $H = [t5, p8, t8, p7]$).

p_1 and p_2 in Definition 9 are inconsistent because they are concurrent (exclusive) and the tokens in them will flow to a set of mutually exclusive (concurrent) places. In Figure 3a, $n_s^{12,11} = \{t10, t12\}$ and $n_e^{12,11} = \{t11\}$; $p10$ is not an $n_s^{12,11}$ because for each path from $p10$ to $p11$ or $p12$, it contains other end nodes $t10$ or $t12$. $(p18, p19)$ in Figure 4 is a TP-inconsistent pair because $n_s^{18,19} = t16$ and $n_e^{18,19} = p20$. Note that $n_s^{1,2}$ and $n_e^{1,2}$ do not exist if $p1 \leftrightarrow p2$.

Note that for a PT-inconsistent pair, it must be that every $n_s^{1,2}$ is a place. Otherwise, it may no longer be irreversible (Figures 3a and c; Figure 3c is not an AC). For a TP-inconsistent pair, however, as long as there exists a transition $n_s^{1,2}$, it is not live.

Figures 1 to 4 are examples of SNC where the shaded areas cover the structures involving R1 or R2. Note the net in Figure 4 is neither a FC nor an EFC. In Figure 1, the only two PT-handles $H_1 = [p_2, t_4, p_4, t_3]$ and $H_2 = [p_2, t_2, p_3, t_3]$ start from the same place p_2 but they join at a transition t_3. To satisfy R1, there is a TP-bridge $B_{12} = [t_4, p_3]$ from H_1 to H_2 and a TP-bridge $B_{21} = [t_2, p_4]$ from H_2 to H_1. In Figure 2, the only two TP-handles $H_1 = [t_5, p_6, t_6, p_7]$ and $H_2 = [t_5, p_8, t_8, p_7]$ start from the same transition t_5 but they join at a place p_7. To satisfy R2, there is a PT-bridge $B_{12} = [p_6, t_8]$ from H_1 to H_2 and a PT-bridge $B_{21} = [p_8, t_6]$ from H_2 to H_1.

In the dining philosopher model in Figure 5, the AFOS with two handles [Put1, Fork1, Tk2, Eat2, Put2, Fork2] and [Put1, Fork0, Tk0, Eat0, Put0, Fork3, Tk3, Eat3, Put3, Fork2] has no bridges across them violating R2. Hence it is not an SNC.

$[p_2, t_2, p_3]$ and $[p_2, t_4, p_3]$ in Figure 1 are two prime handles complementary to one another; $n_s = p_2$ and $n_e = p_3$. Note that there are no bridges interconnecting them; hence, they together form a FOS. Since $X = Y = P$, it is symmetrical.

The following theorem helps to show that a SL and SB FC net is also an SL and SB SNC; the converse, however, is not true.

THEOREM 1. ([3]) *A FC is SL and SB iff:*

(1) *N is strongly connected;*
(2) *no circuit has TP handles; and*
(3) *every PT handle, H_1, of a circuit is bridged to its complementary H_2 through a TP-bridge, B.*

A SL and SB FC is also a SL and SB SNC. This is because any SNC is strongly connected; (3) is equivalent to R1 and R2 need not apply. However, the converse is not true. For instance, the net in Figure 4 is a SL and SB SNC, but not a FC. This is because $p16\bullet = \{t17, t20\}$, $\bullet p19 = \{t20\}$, $\bullet p19 \subset p16$ (neither is an EFC, but an AC). Hence, we have the following lemma.

LEMMA 3. *The set of SL and SB FC is a proper subset of SL and SB SNC.*

Thus, Esparza's work is extended to more general classes of nets. We discovered [5] that R2 and its dual R1 are sufficient for any net to be both consistent (Ct; roughly, there exists a firing sequence to return to initial state M_0) and conservative (Cv; the weighted marking is constant for all states) which implies SB but not necessarily SL.

Recall that synthesized nets and SNC nets are closely related. R1 and R2 involve nodes in a global fashion; the synthesis rules, nevertheless, involve nodes in a local fashion. Thus one can view the rules as a localization of the two requirements, which reduces the complexity of analysis. On one hand, the rules provide local conditions for a net to be SNC, similarly to that for a FC net; on the other hand, they are better than the two requirements by [3] which are global conditions. Note that any Ct and Cv FC is a SNC but not *vice versa*. An arbitrary SNC net may not be SL. However, any SNC net that is a FC, is SL which is not true for AC.

Since SNC and synthesized nets are closely related and it is rather difficult to verify R1 and R2 for arbitrary handles of

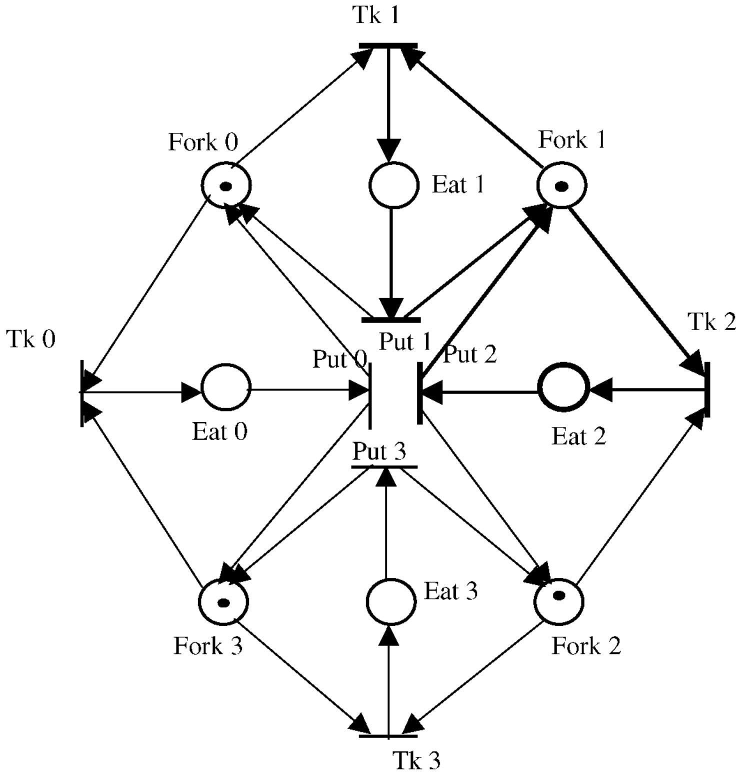

FIGURE 5. [1]. The PN model of dining philosophers is not an SNC due to the AFOS with two handles: [Put1, Fork1, Tk2, Eat2, Put2, Fork2] and [Put1, Fork0, Tk0, Eat0, Put0, Fork3, Tk3, Eat3, Put3, Fork2] with no bridges across them violating R2.

an SNC, we should reduce the global requirements R1 and R2 to simple local ones. It would be simpler to verify them for a class of nets where the B in R1 (R2) is the only one that is bridged to Ω for each PT- (TP-) handle. We prove that this class of net is equivalent to SNC. Conditions for liveness and reversibiliy can be studied by investigating all such possible simple structures and their reachability.

SNC is SB [5] and according to [15], for classes of *bounded* nets such as SNC that can be covered by non-negative T-invariants (a vector with component x_i; if each t_i fires x_i times, it will return to M_0), liveness can be efficiently decided by checking deadlocks with polynomial time complexity. This is because boundedness limits the number of reachable markings to check liveness. Barkaoui *et al.* [14] introduced a new class of extended non-self-controlling (ENSeC) nets to extend EFC and non-self-controlling nets. However, it does not cover all SNC. Any net containing the structure in Figure 7a is not an ENSeC net where $t1$ and $t2$ are in conflict and there is a conflict-free path from $t1$ to $t2$ and *vice versa*. Further, the structures of minimal deadlocks are not correlated to handles and bridges and hence it is not clear and intuitive as to what they look like in structures. Although they have polynomial algorithms for deciding liveness for only subclasses (elementary and loop-free) of ENSeC, there are no efficient algorithms, as stated in [14], to verify a net to be ENSeC.

An AFOS with $n_s \in T(P)$ and $n_e \in P(T)$ may result in unboundedness (non-liveness). To fix the problems, bridges must be inserted into the structure. This results in a SOS which contains two handles, H_1 and H_2, and two bridges, B_{12} from H_1 to H_2 and B_{21} from H_2 to H_1.

It would be easier to search all FOSs rather than all TP-handles to verify the requirements. $\forall n_l \in N$, consider it as an n_s if $\exists n_j \in n_l\bullet$ and $n_k \in n_l\bullet$, $j \neq k$; find an n_e such that a FOS contains n_l, n_e, n_j and n_k. Note that it could be that $n_e = n_l$ and a FOS always exists. Thus, any strongly connected net can be decomposed into a FOS. Furthermore, we may be able to identify the properties of N by investigating all these FOSs. FOSs can characterize all nets in Figures 1–4. The net in Figure 1 is well behaved; however, if we delete the arc from $t2$ to $p4$, it is not *live* even though it can still be decomposed into a symmetrical FOS. This prompts us to investigate SOSs.

Since the SNC requirements are to maintain well behavedness, SOSs must also satisfy the same requirements for the net to be SNC. If we can prove (see Theorem 2) that a net all of whose SOSs satisfy the SNC requirements is a SNC, then we can just search all SOSs and check the requirements without looking into higher-order structures. This considerably simplifies the matter. In addition, we can discover conditions for liveness, irreversibility, reachability, deadlocks, traps, etc. for SNC by investigating all possible conditions of TP- and PT-SOSs.

Prior to proving Theorem 2, the following definitions are helpful.

DEFINITION 10. *Let* $\Upsilon' = H_1' \cup H_2'$ *and* $\Upsilon = H_1 \cup H_2$. Υ' *is said to be embedded in* Υ *on* H_i *if* $n_s' = n_s$, $H_i' \subset H_i$,

TABLE 1. The four FOSs embedded on the PT SOS of N in Figure 1 and their H_i and B (in Definition 10).

	FOS Υ'	H_i	B
1	$[p_2, t_2, p_3]$ and $[p_2, t_4, p_3]$	$H_2 = [p_2, t_2, p_3, t_3]$	$B_{12} = [t_4, p_3]$
2	$[t_2, p_3, t_3]$ and $[t_2, p_4, t_3]$	H_2	$B_{21} = [t_2, p4]$
3	$[p_2, t_2, p_4]$ and $[p_2, t_4, p_4]$	$H_1 = [p_2, t_4, p_4, t_3]$	B_{21}
4	$[t_4, p_3, t_3]$ and $[t_4, p_4, t_3]$	H_1	B_{12}

$i = 1$ or 2, $H'_j \cap H_j \neq \emptyset$, $j \neq i$, $\exists$ a bridge B of Υ and $B \subset H'_j$.

DEFINITION 11. *Υ^m is defined to be an m-order structure on H_i if there are $m - 1$ Υs embedded on H_i, $i = 1$ or 2. H_i, H_j $(i \neq j)$, n_s and n_e are denoted as H_i^m, H_j^m, n_s^m and n_e^m respectively. The corresponding B in Definition 10 is denoted as B_m^k for Υ^k embedded in Υ^m. Υ^m is a TP (PT) m-order structure if its $n_s \in T$ (P) and $n_e \in P$ (T).*

Let SNC_2 denote a net satisfying the SNC requirements for any SOS. In the following, we prove SNC = SNC_2 by showing that an arbitrary m-order structure (for any m) in an SNC_2 meets the requirements in SNC.

THEOREM 2. SNC = SNC_2.

Proof. $(\rightarrow)$ obvious. $(\leftarrow)$ We want to show that for an arbitrary m-order structure, SNC_2 satisfies the requirements in SNC. We approach the proof by induction. When $m = 2$, it is true by definition. Assume it is true for $v = m - 1$. We will show that it is also true for $v = m$. Consider a TP m-order structure Υ^m. The case of a PT m-order structure can be proved in a dual fashion. Let Υ^{m-1} be an $(m - 1)$-order structure embedded in Υ^m on H_1^m. Then the B_m^{m-1} on H_2^{m-1} is a directed elementary path from a node n_l on H_2^m to n_e^{m-1}. We claim that $n_l \notin T$ because the AFOS (with $n_s = n_l$ and $n_e = n_e^m$ containing B_m^{m-1}, the directed elementary path on H_1^m from n_e^{m-1} to n_e^m, and the directed elementary path on H_2^m from n_l to n_e^m) does not have bridges violating the requirements of SNC_2.

If B_m^{m-1} is a PT-path, then it satisfies the requirements for SNC and we are done. If B_m^{m-1} is a PP-path, then Υ^{m-1} is a TP $(m - 1)$-order structure and it must have a PT-bridge B from a node n_f on H_2^{m-1} to a node on H_1^{m-1} which is also on H_1^m. If n_f is on H_2^m, then we are done. If n_f is on B_m^{m-1}, then the directed path on B_m^{m-1} from n_l to n_f plus the above B form a PT-bridge from H_2^m to H_1^m and the theorem is proved. □

Note that the two bridges in a SOS should not be in an elementary circuit.

LEMMA 4. *$\forall SNC_2$, $\neg(B_{ij}^e \rightarrow B_{ji}^s)$, $i, j = 1$ or 2, $i \neq j$, where B_{ij}^e is the end node of bridge B_{ij} from H_i to H_j and B_{ji}^s the start node of bridge B_{ji} from H_j to H_i.*

Proof. Consider a TP SOS. The case of a PT SOS can be proved in a dual manner. Assume the contrary, then there exists a PT FOS whose $n_s = B_{ji}^s$ and $n_e = B_{ij}^e$, violating the requirements of SNC_2. □

Any TP (PT) SOS can be decomposed into four FOSs. An example is shown in Table 1.

In Table 1, $\varphi = H_1 \cup H_2 \cup B_{12} \cup B_{21}$ is a SOS. FOS1 Υ' contains two handles: $H'_2 = [p_2, t_2, p_3]$ and $H'_1 = [p_2, t_4, p_3]$ and $\Upsilon = H_1 \cup H_2$. $B = [t_4, p_3]$. Υ' is embedded in Υ on H_i $(= H_2)$ based on Definition 10 by setting $i = 2$, $j = 1$ and noting that $H'_2 \subset H_2$, $H'_1 \cap H_1 \neq \phi$, $j \neq i$, and $B \subset H'_1$. Υ' is also embedded on φ by Definition 11.

4. INCONSISTENT PAIR AND LIVENESS CONDITIONS

Examining all possible FOSs and SOSs (Figures 1–4), we find that a SNC is not live only if it has a TP SOS. However, the converse is not true. That is, some SNCs that have TP SOSs may be live. They are not live only if the TP SOS involves a TP-inconsistent pair of places as defined below.

We will show that an SNC is not live if it has at least one TP-inconsistent pair. An example is shown in Figure 4 where $p18$ and $p19$ is such a TP-inconsistent pair (with $n_s^{18,19} = t16$, $n_e^{18,19} = p20$). Deadlock occurs for the following firing sequence: $\sigma = t16, t17, t18$. A token gets trapped in each $p18$ and $p19$ and their output transitions, $t19$ and $t20$, cannot fire. Note that after t_s fires, each of the two handles (mutually complementary) $[t16, p16, t17, p18, t19, p20]$ and $[t16, p17, t18, p19, t20, p20]$ from n_s $(= t16)$ to n_e $(= p20)$ get one token. If the tokens could flow along the two handles freely to p_e $(= p20)$, p_e would get two tokens. Continuing the above firing process, the net eventually became unbounded. This violates the fact that any SNC is bounded.

Hence the above two tokens cannot flow freely along the two handles. Instead, they are trapped at input places of two transitions, $t19$ and $t20$, which are mutually exclusive and both cannot fire. Ideally, these two tokens should join at and fire a transition. But now they are diverted to the input places of two transitions that are mutually exclusive. This results in a shortage of tokens for the net to be live. The above token trapping would not occur in the simple structure (shown in Figure 4) if there are no TP-inconsistent pairs of places in the simple structure, as can be checked by all reachable markings after the firing of t_s $(= t16)$ in Figure 4. This observation results in Theorem 4 in Section 6. Below, we will prepare and provide a formal proof of Theorem 4 based on the concept of deadlocks and traps. We will show that minimal deadlocks can be categorized as two kinds. Type I minimal deadlock is also an S-component (Definition 12) and a minimal trap as shown in the following section.

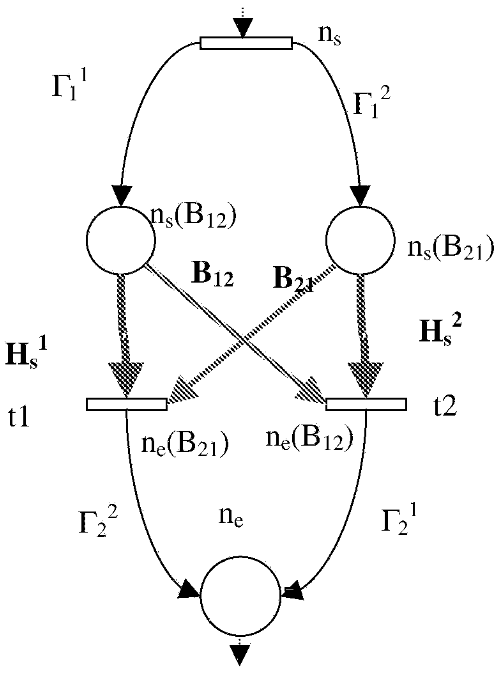

FIGURE 6. The bridge B and the subhandle H_s.

In the following, we examine all possible TP SOSs to find liveness conditions. Prior to that, the following paragraph provides definitions for H_s^1 and H_s^2 used in the conditions.

The two bridges in the structure (Figure 6, a subnet of an SNC) divide the two handles into subhandles. Let H_s^1 (H_s^2) denote the subhandles whose start and end nodes coincide with those of B_{12} (B_{21}) and B_{21} (B_{12}) respectively.

LIVENESS CONDITIONS (LCs). Let $t1$ and $t2$ be the two end nodes of the two bridges. All inputs of $t1$ or $t2$ referred here are one of B_{12}, B_{21}, H_s^1, H_s^2 (see Figure 6).

1. If both $t1$ and $t2$ have exactly one non-VP (virtual path, see Definition 3) input, then the n_s of these two non-VP inputs must be a place.
2. If exactly one of $t1$ and $t2$ has both its inputs being non-VP, the other t must have no non-VP inputs.

Figures 7 and 8 (subnets of SNC) demonstrate cases (1) and (2) of LCs of SNC respectively. Note in Figures 7b and 8b, the nets violate the LCs (1) and (2) respectively and will have a deadlock.

Note that token trapping occurs when two places whose n_s is a transition and whose n_e is a place are marked. Such a pair of places was defined earlier to be a TP-inconsistent pair of places. The LC is equivalent to the absence of a TP-inconsistent pair of places.

5. S-COMPONENTS AND TWO TYPES OF MINIMAL DEADLOCKS

Liveness is closely related to minimal deadlocks that contain no traps. Tokens in a minimal deadlock may get completely unloaded and thereafter it remains unmarked. Hence output transitions of places in the minimal deadlock are dead.

DEFINITION 12. *Given a net N in which N' is a subnet of it and N' ($\subseteq N$) is an S-component of N iff N' is strongly connected state machine and $T' = \bullet P' \cup P'\bullet$.*

We will show that minimal deadlocks in a SNC can be categorized into only two kinds. One is the place set P of an S-component; the other appears only when there exist inconsistent pairs of places.

DEFINITION 13. *A net $N = (P, T, F)$ is S-component decomposable iff $\exists$ a collection $N_i, i = 1, 2, \ldots, k$, of S-components such that $P = \cup P_i, T = \cup T_i, F = \cup F_i$. $\{N_1, N_2, \ldots, N_k\}$ is called a cover of N or N is covered by $\{N_1, N_2, \ldots, N_k\}$.*

The corresponding definitions for a T-component can be defined in a dual fashion [5]. We showed [5] that SNC is a better classification than FC such that SNC and TS-decomposable PNs are equivalently proved by developing a method of constructing T-components. Its S-components can be obtained using its reverse dual. We start from a circuit and add a TT-handle. We then add a TT-handle to the T-component being constructed. Continue in this fashion until we can find no TT-handles to the T-component being constructed. The resulting subnet is a T-component. A new T-component can be found by selecting a circuit passing through one node that is not in the existing T-component and repeating the above procedure. Repeat this process until all nodes are in some T-components.

In the rest of this paper, we will refer to N_i as the ith S-component and its P_i as S_i. An S-component can never gain tokens from or lose tokens to places exterior to an S-component; hence we have the following lemma.

LEMMA 5. *S_i of every S-component N_i in an SNC is both a minimal deadlock and a minimal trap.*

Proof. Throughout this proof, we will consider only nodes in N_i. $\forall p \in S_i$, because N_i is strongly connected, $(p\bullet)\bullet \in S_i$; hence, $(S_i\bullet) \subseteq (\bullet S_i)$. Similarly, we can prove that $(\bullet S_i) \subseteq (S_i\bullet)$. Thus S_i is both a deadlock and a trap. They are both minimal because N_i is a SM. Otherwise, let $D \subset S_i$ be a deadlock, then $\exists p \in S_i$, $p \notin D$, $p' \in D$, $p \in (p'\bullet)\bullet$. There exists no $p_j \in D$, $p_j \in (p'\bullet)\bullet$ since N_i is an SM. Because N_i is strongly connected, there exists a path from p to a $p'' \in D$ and there exists no $p_k \in D$, $p_k \in \bullet(\bullet p'')$ since N_i is a SM. Thus, $p'\bullet \not\subset \bullet D$ and $\bullet p'' \not\subset D\bullet$. Hence D is not a deadlock—a contradiction. Thus S_i is a minimal deadlock. Similarly, S_i can be proved to be a minimal trap. □

There are only two S-components (Figure 9) for the net in Figure 4 and each is both an I- and O-subnet (recall Definition 7) of S_i. Each S_i is both a minimal deadlock and trap. One question, however: is there any other minimal deadlock besides S_i that is not a trap? Since the place set P of a SOS is a subset of $S_1 \cup S_2$, to find such a kind of minimal deadlock, we can only take part of each S_i and combine them. Figure 10b shows such a minimal deadlock, which is no longer a state machine and is associated with a TP-handle. There are no other kinds of minimal deadlocks.

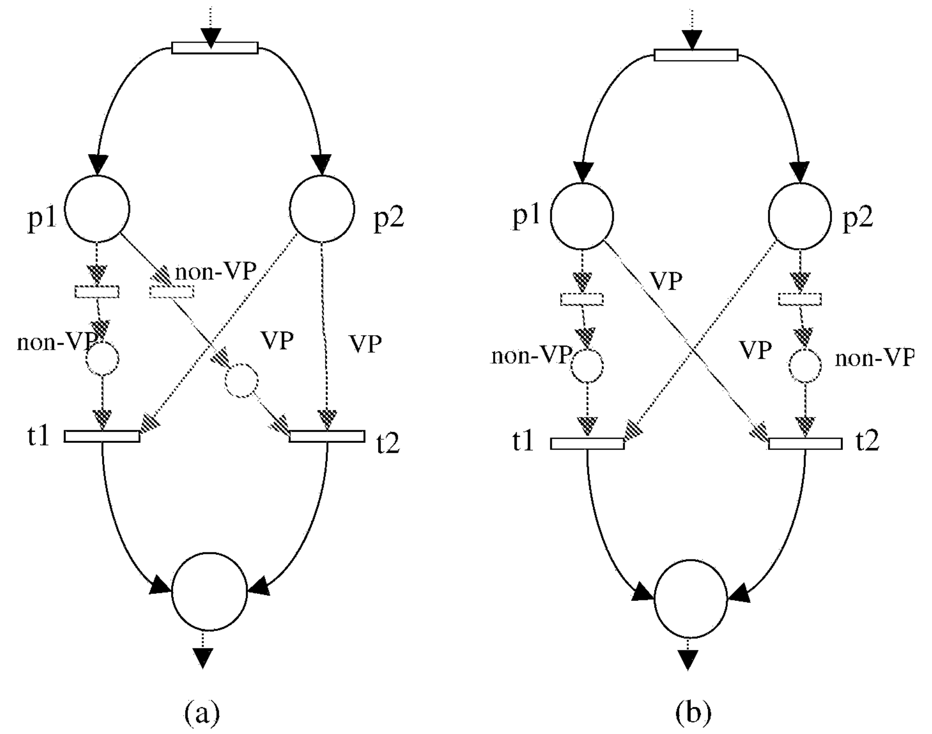

FIGURE 7. (a) Case (1) of liveness condition of SNC. (b) Violates the LC and there is a deadlock.

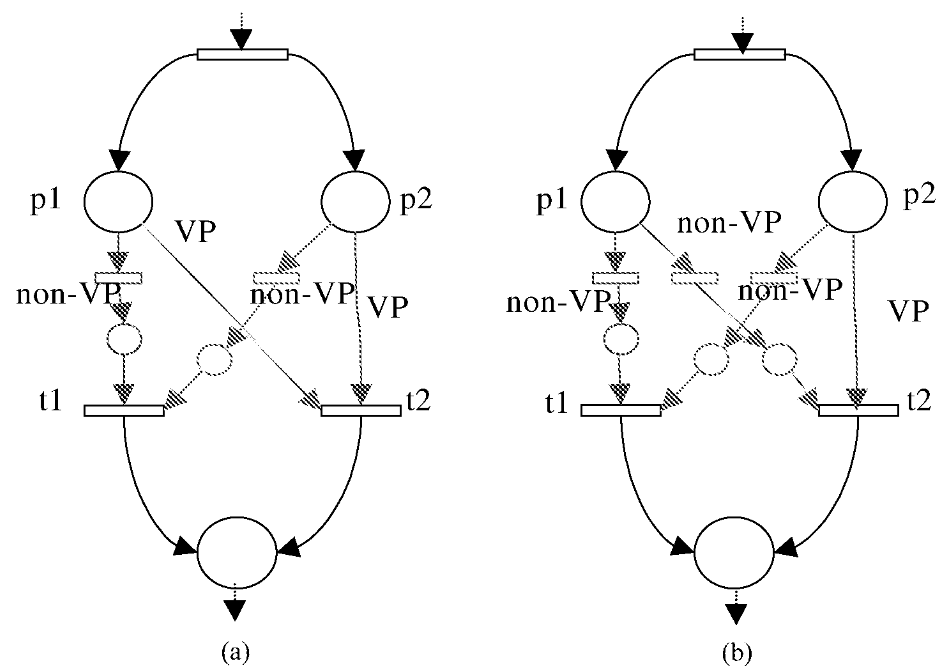

FIGURE 8. (a) Case (2) of liveness condition of SNC. (b) Violates the LC and there is a deadlock.

LEMMA 6. *If D is a non-minimal deadlock and trap and H is a set of non-virtual PT handles to $(D\backslash P_H)^I$ (the I-subnet generated from the set places in D and not in H; i.e. $D\backslash P_H$), then $(D\backslash P_H)^I$ is also a deadlock, but not a trap.*

Proof. Let $p \in (D\backslash P_H)^I$, then if $p \neq n_s$ of H, both $\bullet p$ and $p\bullet$ remain unchanged. If $p = n_s$ of H, then $\exists t \in p\bullet$, $t \in H$ and $t \notin \bullet(D\backslash P_H)^I$ since the PT is non-virtual. We have $\bullet(D\backslash P_H)^I \subset (D\backslash P_H)^I\bullet$. Thus $(D\backslash P_H)^I$ remains a deadlock, but is no longer a trap. □

In Figure 4, $D = S_1 \cup S_2$ is a non-minimal deadlock and trap and H is a set of two non-virtual PT handles: $[p16, t17, p18, t19]$ and $[p17, t18, p19, t20]$. $(D\backslash P_H)^I$ is

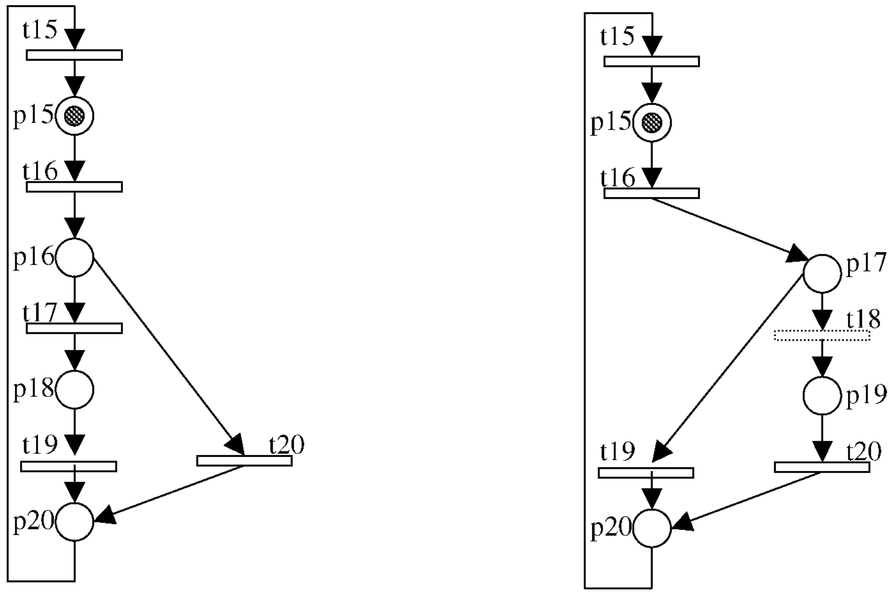

FIGURE 9. S-components of Figure 4.

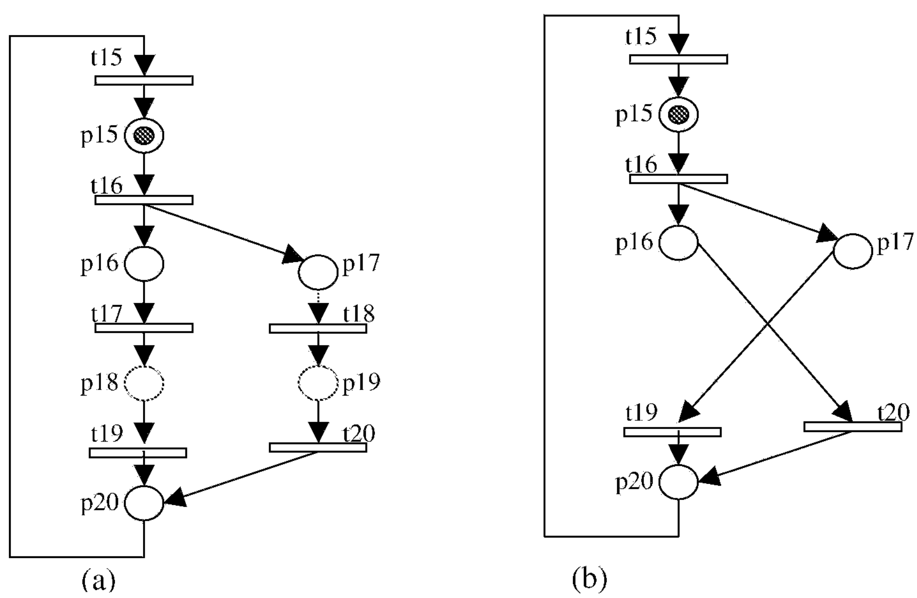

FIGURE 10. The only two type II deadlocks for the net in Figure 4, (a) is non-minimal; (b) is minimal.

indeed a deadlock. When D is an S_i, then all of the above H in N_i are not PT- but PP-handles. Hence, consider the case where the I-subnet (see Definition 7) generated from D, denoted as I_i, contains only two TP-handles as shown in Figure 10. Such a D is called a type II deadlock.

DEFINITION 14. *D_{II} (D_{I}) is a type II (I) deadlock if its I-subnet, I_i, generated from D_{II} (D_{I}) (does not) contains TP-handles.*

By the requirement R1 for SNC, there is one bridge from one TP-handle to the other and *vice versa*. It depends on the structures of the two bridges as to whether or not the deadlock is both minimal and a trap. The two TP-handles (see Figure 10) plus the two bridges form a SOS which may involve another minimal deadlock besides the above D_{I}. Hence, we once again examine the SOS. First we note that (Figure 10a) if either of the two bridges is a VP, then D is no longer minimal and it contains a D_{I} which is an S_i. In Figure 10a, D equals the place set P of the net and one of the two PT-bridges $[p17, t19]$ is a VP and D contains $S_2 = \{p15, p17, p19, p20\}$.

Hence, both bridges must be non-VP (Figure 10b). Some output transitions of the n_s of the bridges do not belong to the input set of transitions of places in D; hence D_{II} is not a trap (see also Theorem 3) and the SNC is not structurally live. Because each bridge is a non-VP, each contains at least a place. These two places form a TP-inconsistent pair with their n_s and n_e identical to that of the above two TP-handles.

Thus we have the following lemma.

LEMMA 7. *An SNC contains a minimal D_{II} iff it contains a TP-inconsistent pair of places.*

Note that a D_{II} (unlike D_I) is a minimal deadlock that does not contain a trap. In the reverse net N^r (by reversing all arcs in N), a trap becomes a deadlock and *vice versa*. Based on this dual property and the fact [6] that the reverse of an SNC is also an SNC (i.e. by interchanging places with transitions of an SNC), if we consider a PT SOS, we will find a type II minimal trap which is not a deadlock. However, this has nothing to do with the liveness of SNC. Hence we will not deal with PT SOS in the following.

6. FORMAL PROOF OF LIVENESS CONDITIONS

We now provide some lemmas to support the formal proof of LCs. In general, $N_1 \cup N_2$ may contain more than one SOS. To simplify the matter, in the following, we consider only one SOS. The case of multiple SOSs in $N_1 \cup N_2$ can be dealt with in a similar manner. Note that $S_1 \cup S_2$ is not a minimal deadlock. Hence, D_{II} can be obtained by deleting some PT-handles in $N_1 \cup N_2$ (from Lemma 6). D_{II} includes only the two complementary prime handles in a SOS and does not contain the two PT-bridges in the R2 requirement. Hence, deleting the above two PT-handles (they are bridges in the SOS) in $N_1 \cup N_2$ and taking the place set of the resulting subnet forms D_{II}.

In the following, we show that if both the above PT-bridges are non-VP, then it is indeed a D_{II}. We then show that the absence of D_{II} (or TP-inconsistent pair) is both a sufficient and necessary condition for liveness. Denoting these two bridges as $B1$ and $B2$, we have the following theorem.

THEOREM 3. *If both $B1$ and $B2$ are non-VP, then $(S_1 \cup S_2)\backslash(P_{B1} \cup P_{B2})$ is a deadlock, but not a trap.*

Proof. Since $S_1 \cup S_2$ is both a deadlock and a trap, the theorem then follows from Lemma 6.

In Figure 4, $B1 = [p16, t17, p18, t19]$ and $B2 = [p17, t18, p19, t20]$. $S_1 = \{p15, p16, p18, p20\}$, $S_2 = \{p15, p17, p19, p20\}$, $P_{B1} = \{p16, p18\}$, $P_{B2} = \{p17, p19\}$, and $D = (S_1 \cup S_2)\backslash(P_{B1} \cup P_{B2}) = \{p15, p16, p17, p20\}$ is the set of all places for the subnet in Figure 10b. It is a deadlock, but not a trap because $t17 \in p16\bullet$, $t18 \in p17\bullet$, $p16 \in D$, $p17 \in D$, and $\neg(\exists p \in D$, such that $t17 \in \bullet p, t18 \in \bullet p)$. □

LEMMA 8. *If $n_s^{1,2} \in T$ and the net is live, then $\exists M \in R(N, M_0)$, $M(p1) = M(p2) = 1$.*

Proof. After $n_s^{1,2}$ fires, $\exists\sigma$ such that $M(p1) = M(p2) = 1$ since there are no bridges across the two handles from $n_s^{1,2}$ to $p1$ and $p2$ respectively to prevent σ from occurring. □

LEMMA 9. *If $\exists$ a TP-inconsistent pair of places in an SNC, then N is not live.*

Proof. Let t_s $(= t16$ in Figure 4) and p_e $(= p20)$ be the n_s and n_e of TP FOS respectively. Assume the contrary and N is live. $\exists M \in R(N, M_0)$, t_s is firable. After t_s fires, subsequently, $\exists\sigma$ such that all ps $(= p16, p17)$ of the non-virtual PT handles in Theorem 3 will get a token. These tokens will flow into an inconsistent pair of places ($p18$ and $p19$) in the PT handles and the deadlock in Theorem 3 will be empty. According to the definition of deadlock, all transitions of the deadlock will no longer be firable and N is not live. □

THEOREM 4. *An SNC is structurally live iff there are no TP-inconsistent pairs of places in the SNC.*

Proof. $(\rightarrow)$ It immediately follows from Lemma 9. $(\leftarrow)$ Consider the case where every S-component (hence every deadlock) contains at least one token. Since there are no TP-inconsistent pairs, every deadlock is also a trap. Hence, no deadlock can ever be empty. But this is impossible if it is not live, because then there exists, as proved below, an empty deadlock containing the input place of a dead t. Let t be in an S-component ζ in N. There are two cases: ζ (1) does not or (2) does share some TT-paths with another ζ'. (1) ζ is a SM and contains at least one token. No t in ζ can be dead. The set of places in ζ is a deadlock containing a $p \in \bullet t$. (2) Find an S-component ζ' such that after deleting some TT-paths in ζ', the resulting N' remains a SNC. By applying the recurrence hypothesis for t on N' with marking M', a subset of M, there exists an empty D_m in M'' reachable from M'. D_m contains a place $p \in \bullet t$. By the definition of the S-component, $\zeta \cap \zeta'$ is at most a set of TT-paths. D_m remains a minimal deadlock in N. □

We now prove that SNC satisfies the structural Commoner's property.

THEOREM 5. *A SNC is structurally live iff it satisfies the structural Commoner's property.*

Proof. Since a D_{II} (unlike D_I) is a minimal deadlock that does not contain a trap, by Lemma 7 and Theorem 4, every minimal deadlock contains a trap. □

THEOREM 6. *An SNC is live iff it satisfies the Commoner's property.*

Proof. If it does not satisfy Commoner's property (2), then transitions in an empty deadlock can never be fired and the SNC is not live. Hence, by Theorem 5, if an SNC is live, it satisfies Commoner's property (2). On the reverse side, if it is dead, then there exists an empty deadlock (Lemma 15 in [15]). The maximal trap inside the empty deadlock is also empty, against Commoner's property (2). □

7. ALGORITHMS FOR VERIFICATION OF SNC AND LIVENESS

Barkaoui *et al.* [14] confessed that they could not offer an algorithm for the verification of the class ENSeC beyond their algorithm for verification of liveness, so did Lautenbach and Ridder [15] with their class of bounded nets. Rather than developing two algorithms for verification of SNC and liveness, we present an integrated algorithm (Algorithm 1) for both.

Let $N_1 = \varnothing, N_2 = N$.

A. Find an elementary circuit c. Go to step C.
B. Find a handle H of N_1 with n_s and n_e.
C. Enter new entries or update old entries per the following rules. n' denotes updated n'_s or n'_e for a certain pair of nodes.
    1. If $n_c \leftrightarrow n_d$, then never update the entry.
    2. If n_c and n_d are on an elementary circuit containing H, then $n_c \leftrightarrow n_d$.
    3. $\forall n_s \leftarrow n_c \leftarrow n_e^{s,e}, \forall \Gamma = n_s \rightarrow n_d \rightarrow n_e^{s,e}, H \subseteq \Gamma, n_s'^{c,d} = n_s, n_e'^{c,d} = n_e^{s,e}$.
    3^r: $\forall n_s^{s,e} \rightarrow n_c \rightarrow n_e, \forall \Gamma = n_e^{s,e} \rightarrow n_d \rightarrow n_e, H \subseteq \Gamma, n_s'^{c,d} = n_s^{s,e}, n_e'^{c,d} = n_e$.
    4. $\forall n_c \in N_1, \forall n_d \in H \rightarrow n_f$, if $n_s^{c,d} \rightarrow n_s^{s,c} \rightarrow n_c, n_s'^{c,d} = n_s^{s,c}$.
    4^r: $\forall n_c \in N_1, \forall n_d \in n_f \rightarrow H$, if $n_c \rightarrow n_e^{e,c} \rightarrow n_e^{c,d}, n_e'^{c,d} = n_e^{e,c}$.
D. $N_1 = N_1 \cup H, N_2 = N \backslash H$. Repeat step B until $N_2 = \varnothing$.

ALGORITHM 1. Algorithm for constructing S-matrix.

Recall that $\varphi = H_1 \cup H_2 \cup B_{12} \cup B_{21}$. In an arbitrary net, there are three abnormal cases: (1) AFOS, (2) exactly one B is missing in φ; i.e. $B = \emptyset$, (3) $B_{12} \leftrightarrow B_{21}$. Case (3) is reduced to case (1) since there is an AFOS inside φ as can be seen from the proof of Lemma 4. We consider any two output nodes of n_s of H_1. For case (1), they are either inconsistent or in an elementary circuit. For case (2), they are inconsistent. Thus we can examine all pairs of output nodes. If they are inconsistent, then the net is not a SNC. Since liveness of SNC can also be checked by inconsistency, we have an integrated algorithm.

In Figure 4, if $B = [p16, t20]$ is missing, then $(p16, p17)$ is inconsistent because $p20$ is a n_e. The above discussion leads to the following.

LEMMA 10. *A net is a SNC, iff all pairs of output nodes of a certain node are consistent.*

To find an inconsistent pair, we construct a $|P| \times |P|$ two-dimensional S-matrix recording the n_s and n_e for all pairs of places. Each entry is a pointer to a record holding the information about the n_s and n_e of a certain pair of p_1 and p_2. To see whether (p_1, p_2) is an inconsistent pair, we examine the corresponding entry in the S-matrix. If there exist different types (places or transitions) of n_s and n_e in the entry, then (p_1, p_2) is an inconsistent pair.

Initially, all records are *null* pointers. Instead of searching all places and finding their n_s and n_e in a global fashion, we follow an incremental approach. First, we find an elementary circle. All entries in the S-matrix are '$\leftrightarrow$'. Next we find a handle to the circle and update the S-matrix. The n_s (n_e) of a pair of places on the pair of mutually complementary handles respectively is the same as that of the handles. We continue the process until all places in the net have been included in the S-matrix. In general, the new path to be added is a handle to the subnet whose places have been included in the S-matrix.

The above process, adding new paths, is similar to the synthesis process in the knitting technique [7]. It is easy to find the structural relationship between places on the handle and places on the same cycle with n_s or n_e, but it is not obvious for places far away from (no direct connection with) the handle.

We need to determine new entries for places on the handle and update old entries to either '$\leftrightarrow$' (Rule 2) or to new n_s or n_e (Rules 3 and 4). There are two cases: (1) $n_s \leftrightarrow n_e$ and (2) $\neg(n_s \leftrightarrow n_e)$. Only (2) involves updates of old entries. There are three possibilities at new or updated entry cd (of row c and column d): (a) '$\leftrightarrow$' ($p_5 \leftrightarrow p_8$ in Figure 11b), (b) new $n'_s = n_s$ ($n'_e = n_e$) ($n_s'^{6,8} = t5 = n_s$ in Figure 11b) and (c) new $n'_s = n_s^{s,c}$ ($n'_e = n_e^{e,c}$) (in Figure 11b if path $[p5, t5', p6', t6', p7]$ is added before, then $n_s'^{(8,6')} = p5 = n_s$ of $t5$ and $p6'$). Note that in Figure 11b where $n_s \leftrightarrow n_e$, the entries for pairs of places that are both in N_1 need no updates. In Figure 11c where $\neg(n_s \leftrightarrow n_e)$, $n_e^{6,8}$ is updated to have a new $n_e = t_6$.

In Figure 6, prior to the addition of B_{12}, the internal nodes of the Γ_1^1 are not sequential to that of Γ_1^2, while they are mutually sequential afterwards. Hence, the corresponding entries must be updated to '$\leftrightarrow$'. Some entry for a pair of places (one from internal nodes of Γ_1^1 or Γ_2^1, the other from N_1) may need to be updated to have new n_s.

Each rule i^r is formed from rule i by applying rule i to N^r, the reverse of N where all arcs in N are reversed. Hence, interchanging n_e with n_s and reversing the '$\rightarrow$' to '$\leftarrow$' leads to the reverse rule i^r. Each time H is obtained, new entries must be entered and some old entries must be updated to either '$\leftrightarrow$' (rule 2) or to new n_s or n_e (rules 3 and 4). Rule 2 finds all elementary circuits containing H and for any two nodes on such a circuit, the entry is '$\leftrightarrow$'. Rule 3 (3^r) fills entries with a new $n'_s = n_s$ ($n'_e = n_e$). Rule 4 (4^r) fills entries with a new $n'_s = n_s^{s,c}$ ($n'_e = n_e^{e,c}$), but only for nodes n_c that have not been treated in rules 2 and 3.

EXAMPLE. In Figure 3b, let $H = [t10, p21]$, $N_1 = N \backslash H$. (Rule 2) before: $\neg(p10 \leftrightarrow p21)$; after: $p10 \leftrightarrow p21$. (Rule 3) before: $n_s^{13,21} = p9$; after: $n_s^{13,21} = t10$. (Rule 4) before: $n_s^{14',21} = p9$; after: $n_s^{14',21} = t9$. Figure 11 shows an example of constructing the S-matrix for the net in Figure 2.

Since there are $O(|P|^2)$ entries in the S-matrix and for each entry there are $O(|T| + |P|)$ n_s and n_e, hence the

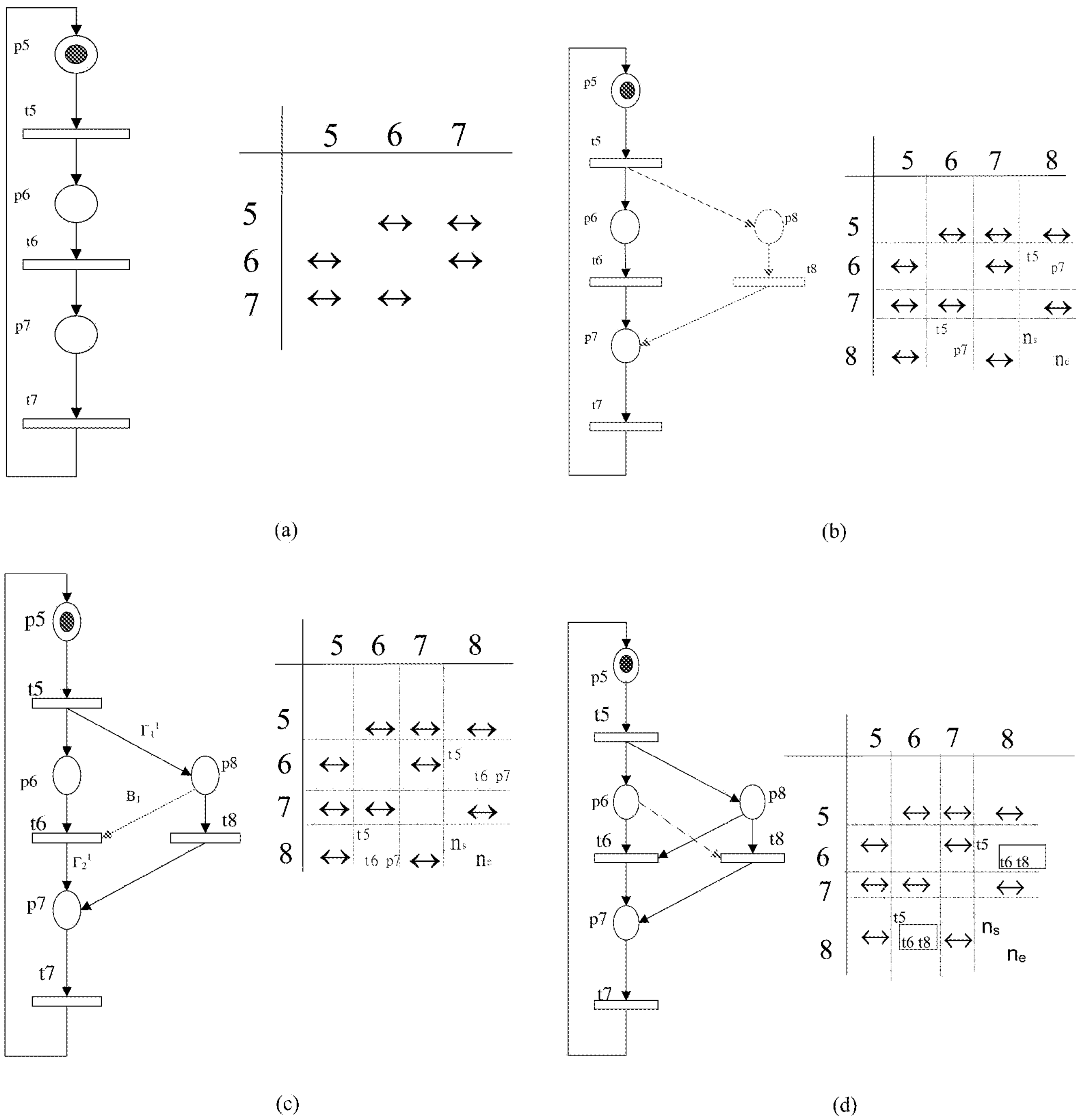

FIGURE 11. An example of constructing S-matrix for N in Figure 2. (a) A circuit and its S-matrix. (b) Adding a new handle $[t5, p8, t8, p7]$, and its updated S-matrix. (c) Adding a new handle $[p8, t6]$, and its updated S-matrix. (d) Adding another new handle $[p6, t8]$, and the final updated S-matrix.

total time complexity is $O(|P|^2(|T| + |P|))$. Once the S-matrix is constructed, we can find inconsistent pairs by searching the entries in the matrix with time complexities of $O(|P|^2(|T| + |P|))$. Hence, the total time complexity for verification of SNC and liveness is $O(|P|^2(|T| + |P|))$.

8. CONCLUSION

We proved that SNC_2 is equivalent to SNC, which covers well behaved FC and yet is not included in AC [5]. That is, a net is a SNC if all SOSs satisfy R1 and R2. Based on these simple SOSs, we have derived the simple conditions plus an integrated algorithm for verification of SNC and liveness. They have also been applied to the reachability and irreversible problem [6]. Any live SNC without a PT-inconsistent pair of places has been proved to be reversible. We have also developed a new systematic technique to find bad siphons (i.e. not traps) and applied the SNC model to flexible manufacturing system deadlock prevention [16].

Thus, unlike traditional structures, the simple structures presented in this paper not only classify, but also characterize properties (such as liveness and irreversibility) of SNC. Unlike other classes of nets, this new classification extracts most nearly well behaved nets among all possible nets into the SNC. As long as a net is in the new class, it is bounded and its LC is simple and verification is efficient. More importantly, it helps to simplify and enhance our synthesis rules in the knitting technique [5]. It is interesting to investigate whether this would extend to nets with a slight variation of the simple structures. Future work should be directed to the construction of legal firing sequences and the shortest path problem which, as we suspect, is also not hard for SNC due to the simplicity of reachability problem.

Note that without type II minimal deadlock, every siphon is a trap. If every siphon is marked, the SNC (like a FC) is live. Thus, Commoner's property remains valid for a larger class of well behaved SNC than the class of well behaved FC.

ACKNOWLEDGEMENTS

This work is supported by the National Science Council under research grant number NSC87-2213-E-004-001. Professor M. C. Zhou's comments improved the quality of the paper.

REFERENCES

[1] Lipton, R. J. (1976) *The Reachability Problem Requires Exponential Space*. Research Report 62, Department of Computer Science, Yale University, New Haven, CT.

[2] Murata, T. (1989) Petri nets: properties, analysis and applications. *Proc. IEEE*, **77**, 541–580.

[3] Esparza, J. and Silva, M. (1991) Circuits, handles, bridges, and nets. In *Advances in Petri Nets. Lecture Notes in Computer Science*, **524**, 210–242. Springer, Berlin.

[4] Hack, M. H. T. (1972) *Analysis of Production Schemata by Petri Nets*. MS Thesis, Department of Electrical Engineering, MIT, Cambridge MA.

[5] Chao, D. Y. and Wang, D. T. (1997) Two theoretical and practical aspects of knitting technique—invariants and a new class of Petri net. *IEEE Trans. Syst., Man Cyber.*, **27**, 962–977.

[6] Chao, D. Y. and Tseng, J.-H. (1999) Some properties of synchronized choice ordinary Petri net. Invited Session paper. In *1999 International Federation of Automatic Control*, Beijing China, July 5–9, pp. 193–197.

[7] Chao, D. Y. (1998) Application of a synthesis algorithm to flexible manufacturing system. *J. Inf. Sci. Eng.*, **14**, 409–447.

[8] Chao, D. Y. and Wang, D. (1994) A synthesis technique for general Petri nets. *J. Syst. Integration*, **4**, 67–102.

[9] Chao, D. Y., Zhou, M. C. and Wang, D. T. (1994) Extending knitting technique to Petri net synthesis of automated manufacturing systems. *Comp. J.*, **37**, 1–10.

[10] Chen, Y., Tsai, W. T. and Chao, D. Y. (1993) Dependency analysis—a compositional technique for building large Petri nets. *IEEE Trans. Par. Distrib. Syst.*, **4**, 414–426.

[11] Zhou, M. C., DiCesare, F. and Desrochers, A. A. (1992) A hybrid methodology for Petri net synthesis of manufacturing systems. *IEEE Trans. Robotics Automation*, **8**, 350–361.

[12] Lee, D.-I., Kumagai, S. and Kodama, S. (1995) Handles and reachability analysis of free choice nets. In *Application and Theory of Petri Nets. Lecture Notes in Computer Science*, **935**, 298–316. Springer, Berlin.

[13] Lee, D.-I., Kumagai, S. and Kodama, S. (1991) Complete structural characterization of state machine allocatable nets. *IEICE Trans.*, Special Issue on Petri Nets and Discrete Event Systems, E 74, **10,** 3115–3123.

[14] Barkaoui, K., Couvreur, J. M. and Dutheillet, C. (1995) On liveness in extended non self-controlling nets. In *Application and Theory of Petri Nets. Lecture Notes in Computer Science*, **935**, 25–44. Springer, New York.

[15] Lautenbach, K. and Ridder, H. (1994) Liveness in bounded Petri nets which are covered by t-invariants. In Valette, R. (ed.), *Application and Theory of Petri Nets*, Zaragoza, Spain, June 1994. *Lecture Notes in Computer Science*, **815**, 358–378. Springer, Berlin.

[16] Ezpeleta, J., Colom, J. M. and Martinez, J. (1995) A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Trans. Robotics Automation*, **11**, 173–184.