Property Analysis of Petri Net Reduction

Chuanliang Xia
School of Computer Science and Technology, Shandong Jianzhu University, Jinan, China
State Key Laboratory of Computer Science, Institute of Software, Academy of Sciences, Beijing, China
chuanliang_xia@126.com

Abstract—Petri net reduction can avoid the state exploration problem by guaranteeing the correctness. For system specified in Petri nets, this paper proposes reduction methods. A group of sufficient conditions or sufficient and necessary conditions of boundedness preservation and liveness preservation are proposed. These results are useful for studying dynamic properties of Petri nets.

Index Terms—Petri nets, analysis, property preservation, boundedness, liveness

I. INTRODUCTION

Petri nets are well known for their graphical and analytical capabilities in specification and verification, especially for concurrent systems. Research in reduction methods began with simple pattern modifications on Petri nets [1, 2]. Desel [2] showed that a live and safe FC (free choice) net without frozen tokens can be reduced either to a live and safe marked graph or to a live and safe state machine. A well-known result is the preservation of well-formedness and Commoner’s property under the merge of places within a free choice net or an asymmetric-choice net [3]. In order to improve the Petri net based representation for embedded systems verification efficiency, Xia [4] proposed a set of reduction rules, these reduction rules preserve total-equivalence. Murata [5] presented six reduction rules to reduced ordinary Petri nets; these rules preserved liveliness, safeness and boundedness. Sloan [6] introduced a notion of equivalence guarantees that crucial timing and concurrency properties were preserved. Most reductions are quite specific, such as merging a few places or transitions [7, 8], reducing individual places or transitions [9] or very specific subnets. Esparza [10] provided reduction rules for LTL model-checking of 1-safe Petri nets. Huang [11] proposed some new rules to detect the existence of structural conflicts. Based on Delay Time Petri Net (DTPN), Jiang [12] transformed a Time Petri Net (TPN) component to DTPN model in order to preserve such properties as synchronization, conflict and concurrency during the reduction. In order to improve the analysis efficiency, Shen [13] reduced a large digital system to a simpler one by using three kinds of reduction rules.

This paper investigates one type of transformations and its property-preserving approach for verification. Two kinds of subnet reduction methods are proposed. Conditions of boundedness and liveness preservation of ordinary Petri reduction net are proposed.

This paper is organized as follows. Section II presents basic definitions. Section III investigates the first reduction method. Section IV studies the second reduction method. Section V gives an example of manufacturing system verification. Section VI concludes this paper.

II. BASIC DEFINITIONS

A weighted net is denoted by $N = (P, T; F, W)$ where $P$ is a non-empty finite set of places, $T$ is a non-empty finite set of transitions with $P \cap T = \emptyset$. $F \subseteq (P \times T) \cup (T \times P)$ is a flow relation and $W$ is a weight function defined on the arcs, i.e., $W : F \rightarrow \{1, 2, 3, \ldots\}$. $N_1 = (P_1, T_1; F_1, W_1)$ is called a subnet of $N$ if $P_1 \subseteq P$, $T_1 = T$, $P_1 \neq \emptyset$, $T_1 \neq \emptyset$, $F_1 = (P_1 \times T_1) \cup (T_1 \times P_1)$ and $W_1 = W | F_1$, i.e., the restriction of $W$ on $F_1$.

A Petri net is a couple $(N, M_0)$, where $N$ is a net and $M_0$ is the initial marking of $N$. A transition $t$ is said to be live in $(N, M_0)$ iff for any $M \in R(M_0)$, there exists $M' \in R(M)$ such that $t$ can be fired at $M'$. $(N, M_0)$ is said to be live iff every transition of $N$ is live. A place $p$ is said to be bounded in $(N, M_0)$ iff there exists a constant $k$ such that $M(p) \leq k$ for all $M \in R(M_0)$. $(N, M_0)$ is bounded iff every place of $N$ is bounded.

III. THE FIRST REDUCTION METHOD

In this section we present the first reduction operation. This operation preserves boundedness and liveness.

Definition 3.1 A net $N_0 = (P_0, T_0; F_0, W_0)$ is said to be a p-subnet of $N = (P, T; F, W)$ iff:

(1) $N_0$ is a subnet of $N$,
(2) $\ast T_0 \cup T_0^* \subseteq P_0$,
(3) $N_0$ is connected, $\{p_i, p_o\} \subseteq P_0$ and $p_i$ is the only input place of $N_0$, $p_o$ is the only output place of $N_0$.

Supposition 3.1 A p-subnet satisfies:

(1) $p_i$ is the only place which can contain the initial marking (token(s)).
(2) In a process (tokens from outside flow into \( p_i \), pass \( N_0 \) and then flow out from \( p_o \)), the number of tokens flowing into \( p_i \) is equal to the number of tokens flowing out from \( p_o \).

**Definition 3.2** p-subnet reduction operation: a reduced net \( N' = (P', T', F', W') \) is obtained from original Petri net \( N = (P, T; F, W) \) by using \( \bar{p} \) to replace a p-subnet \( N_p = (P_p, T_p; F_p, W_p) \), where

\[
\begin{align*}
\text{(1)} & \quad P' = \mathcal{P} - P_p, \\
\text{(2)} & \quad T' = T - T_p, \\
\text{(3)} & \quad F' = F - \{(t, p) \mid t \in \mathcal{P} - \{P_p\} \}.
\end{align*}
\]

**Definition 3.3** \((N', M_0')\) obtained from \((N, M_0)\) by p-subnet reduction operation comprises net \( N' \) and marking \( M_0' \) where

\[
M_0' = \begin{cases} 
&M_p(p) = 0 \\
&M_p(p) = n \quad (n \geq 1)
\end{cases}
\]

\((M_p(p))\) is obtained from \( M \) by deleted the vector corresponding to \( p \).

**Definition 3.4** A net \((\bar{N}_p, \bar{M}_p)\) is said to be a closed p-subnet if adding a transition set \( T_p = \{t_p \mid t_p \text{ corresponding to } t \in \mathcal{P} \} \) and are set \( \{(p, t_p) \mid t \in T_p \} \) to \((N_p, M_p)\), and preserving the marking of \((N_p, M_p)\).

Note that in this section, let \((N', M_0')\): the original net; \((N_p, M_p)\): the p-subnet; \((N_p, M_p)\): the p-subnet system; \((\bar{N}_p, \bar{M}_p)\): the closed p-subnet system; \((N, M_0)\): the reduced net.

**Theorem 3.1** Suppose that \((N, M_0)\) is obtained from \((N', M_0)\) by p-subnet reduction operation. Then \((N', M_0')\) is bounded iff \((N, M_0)\) and \((\bar{N}_p, \bar{M}_p)\) are bounded.

**Proof.** (1) Since \((N, M_0)\) is bound, then \( \forall p \in P \), \( \exists k_1 > 0 \), such that \( M(p) \leq k_1 \), \( \forall M \in R(M_0) \).

Obviously, \( \forall p \in P - \{\bar{p} \} \), \( M(p; \bar{p}) \leq k_1 \). Since \((\bar{N}_p, \bar{M}_p)\) is bound, then \( \forall p \in P_p \), \( \exists k_2 > 0 \), such that \( M_p(p) \leq k_2 \), \( \forall M_p \in R(M_p) \). Let \( k = k_1 + k_2 \), by Supposition 3.1, \( \forall p \in P' \), \( M'(p) = [M_p(p; \bar{p}), M_p(p)] \leq k \), \( \forall M' \in R(M_0') \), so \((N', M_0')\) is bound.

(2) Suppose that \((N, M_0)\) is unbound, then \( \exists p \in P \), \( \forall k > 0 \), \( \exists M \in R(M_0) \) and \( M(p) > k \).

By Supposition 3.1 and Definition 3.1-3.4, \( \forall k > 0 \), \( \exists M' \in R(M_0') \) and \( M'(p) > k \). This contradicts with the fact that \((N', M_0')\) is bounded.

**Theorem 3.2** Suppose that \((N, M_0)\) is obtained from \((N', M_0')\) by p-subnet reduction operation. If \((N', M_0')\) is live and \( p \in \{p \mid (p \in P) \land (M_0'(p) > 0)\} \), then \((N, M_0)\) and \((\bar{N}_p, \bar{M}_p)\) are live.

**Theorem 3.3** Suppose that \((N, M_0)\) is obtained from \((N', M_0')\) by p-subnet reduction operation. If \((N, M_0)\) and \((\bar{N}_p, \bar{M}_p)\) are live, then \((N', M_0')\) is live.

### IV. The Second Reduction Method

In this section we present the second reduction operation. This operation preserves boundedness and liveness.

**Definition 4.1** A net \( N_0 = (P_0, T_0; F_0, W_0) \) is said to be a t-subnet of \( N = (P, T; F, W) \) iff,

\[
\begin{align*}
\text{(1)} & \quad P_0 \cup P \subseteq T_0, \\
\text{(2)} & \quad N_0 \text{ is connected, } \{t, t_o\} \subseteq T_0 \text{ and } t_i \text{ is the only input transition of } N_0, t_o \text{ is the only output transition of } N_0.
\end{align*}
\]

**Supposition 4.1** A t-subnet system \((N_0, M_{i0})\) contains t-subnet \( N_0 \) and initial marking \( M_{i0} \), satisfy

(1) In a process (tokens from outside flow into \( t_i \), pass \( N_0 \) and then flow out from \( t_o \)), \( t_o \) is fired, iff \( t_i \) is fired.

(2) Before \( t_i \) is fired and after \( t_o \) is fired, \( \forall t \in T_0 - \{t_i, t_o\}, t \) can not be enabled.

(3) If \( P_0 \) does not contain token in initial state, \( P_0 \) does not contain token after a process; If \( P_0 \) contains token(s) in initial state, the token(s) will come back to the initial state after a process.

**Definition 4.2** t-subnet reduction operation: a reduced net \( N' = (P', T'; F', W') \) is obtained from original Petri net \( N = (P, T; F, W) \) by using \( \bar{t} \) to replace a t-subnet \( N_t = (P_t, T_t; F_t, W_t) \), where

\[
\begin{align*}
\text{(1)} & \quad P' = P - P_t;
\end{align*}
\]
(2) \( T' = T \cup \{ \tilde{t} \} - T_\iota \);  
(3) \( F' = F \cup \{ (p, \tilde{t}) \mid p \in^* t_\iota \} \cup \{ (\tilde{t}, p) \mid p \in t_\iota^* \} \) 
- \{ (p, t_\iota) \mid p \in^* t_\iota \} - \{ (t_\iota, p) \mid p \in t_\iota^* \} - F_\iota.

**Definition 4.3** \((N', M_0')\) obtained from \((N, M_0)\) by t-subnet reduction operation comprises net \(N'\) and marking \(M_0'\), where \(M_0' = M_{(p,P)0} \) (where \(M_{(p,P)}\) is obtained from \(M\) by deleted the vector corresponding to \(P\)).

**Definition 4.4** A net \((\overline{N}, \overline{M}_0)\) is said to be a t-closed net if we add a transition \(t_\iota\) and arcs \(\{ (p, t_\iota) \mid p \in t_\iota^* \} \cup \{ (t_\iota, p) \mid p \in t_\iota^* \} \) to t-subnet \((N', M_0')\) and the marking of \((\overline{N}, \overline{M}_0)\) is preserved.

Note that in this section, let \((N', M_0')\): the original net; \(N_i = (P_i, T_i; F_i, W_i)\): the p-subnet; \((N, M_0)\): the p-subnet system; \((\overline{N}, \overline{M}_0)\): the closed t-subnet system; \((N, M_0)\): the reduced net.

**Theorem 4.1** Suppose that \((N, M_0)\) is obtained from \((N', M_0')\) by t-subnet reduction operation. Then \((N', M_0')\) is bounded iff \((N, M_0)\) and \((\overline{N}, \overline{M}_0)\) are bounded.

**Proof.** (1) Since \((N, M_0)\) is bounded, then \(\forall p \in P\), \(\exists k_1 > 0\) such that \(\forall M \in R(M_0)\), \(M(p) \leq k_1\). Since \((\overline{N}, \overline{M}_0)\) is bounded, then \(\forall p \in P\), \(\exists k_2 > 0\) such that \(\forall M_\iota \in R(M_0)\), \(M_\iota(p) \leq k_2\).

Let \(k = k_1 + k_2\), \(\forall p \in P\), \(\forall M \in R(M_0)\) such that \(M'(p) \leq k\). So, \((N', M_0')\) is bounded.

(2) Suppose that \((N, M_0)\) is unbounded, then \(\exists p \in P\), \(\exists k > 0\), \(\exists M \in R(M_0)\) such that \(M(p) > k\). That is \(\forall k > 0\), \(\exists M' \in R(M_0')\) such that \(M'(p) > k\). This contradicts with the fact that \((N', M_0')\) is bounded.

**Theorem 4.2** Suppose that \((N, M_0)\) is obtained from \((N', M_0')\) by t-subnet reduction operation. If \((N', M_0')\) is live and \(\star t_\iota \subseteq \{ p \mid p \in P \land (M'(p) > 0) \}\), then \((N, M_0)\) and \((\overline{N}, \overline{M}_0)\) are live.

**Theorem 4.3** Suppose that \((N, M_0)\) is obtained from \((N', M_0')\) by t-subnet reduction operation. If \((N, M_0)\) and \((\overline{N}, \overline{M}_0)\) are live, then \((N', M_0')\) is live.

---

**V. Applications**

In this section we apply results of Section III and Section IV to reduce a flexible manufacturing system.

The manufacturing system consists of one workstation \(WS\) for assembly work and two machining centers for machining. Machining center_1 and \(WS\) share robot \(R_1\).

Machining center_2 and \(WS\) share robot \(R_2\). The system runs as follows:

In the machining center_1, the intermediate parts are machined by machine \(M_1\). Each part is fixtured to a pallet and loaded into the machine \(M_1\) by robot \(R_1\).

After processing, robot \(R_1\) unloads the final product, defixtures it and returns the fixture to \(M_1\).

In the machining center_2, parts are machined first by machine \(M_3\) and then by machine \(M_4\). Each part is automatically fixtured to a pallet and loaded into the machine. After processing, robot \(R_2\) unloads the intermediate part from \(M_3\) into buffer B. At machining station \(M_4\), intermediate parts are automatically loaded into \(M_4\) and processed. When \(M_4\) finishes processing a part, the robot \(R_2\) unloads the final product, defixtures it and returns the fixture to \(M_4\).

When workstation \(WS\) is ready to execute the assembly task, it requests robot \(R_1\), robot \(R_2\) and machine \(M_2\) and acquires them if they are available.

When the workstation starts an assembly task, it cannot be interrupted until it is completed. When \(WS\) completes, it releases the robot \(R_1\) and robot \(R_2\).

Firstly, we give the Petri-net based model of the manufacturing system. Secondly, a reduced net system is obtained by p-subnet reduction method and t-subnet reduction method. Thirdly, we will analysis property preservation of the reduced net system.

The Petri-net based model \((\overline{N}, \overline{M}_0)\) of the original manufacturing system is illustrated in Fig. 5.1.
The reduction process consists of the following three steps.

Step 1: \( (N_1, M_1) \) (Fig. 5.2) is obtained by p-subnet reduction method. By Theorem 3.1-3.3, \( (N_1, M_0) \) is bounded and live, iff \( (N_1, M_1) \) is bounded and live.

Step 2: \( (N_2, M_2) \) (Fig. 5.3) is obtained by t-subnet reduction method. By Theorem 4.1-4.3, \( (N_1, M_1) \) is bounded and live, iff \( (N_2, M_2) \) is bounded and live.

Step 3: \( (N_3, M_3) \) (Fig. 5.4) is obtained by deleted places \( p_{r1} \) and \( p_{r2} \). Since \( p_{r1} = p_{r1}^* \), \( M_2(p_{r1}) > 0 \) and \( p_{r2} = p_{r2}^* \), \( M_2(p_{r2}) > 0 \), then \( (N_2, M_2) \) is bounded and live, iff \( (N_3, M_3) \) is bounded and live.

It is easy to see that \( (N_3, M_3) \) is bounded, live ([5]). By Theorem 3.1-3.3 and Theorem 4.1-4.3, the original net system \( (N, M_0) \) is bounded and live.

VI. APPLICATIONS

In this paper we investigate property preservations of Petri reduction net. Two Petri net reduction methods are proposed, which are the key methods to ensure the reduced net preserving well behaved properties.

ACKNOWLEDGMENT

This work was financially supported by the National Natural Science Foundation of China under Grant No. 60573012, 60672180 and 60721061, the National Grand Fundamental Research 973 Program of China under Grant No. 2002cb312200, and CAS Key Laboratory of Computer Sciences (SYSKF0903).

REFERENCES