SUMMARY  Response property is a kind of liveness property.  The response property problem is defined as follows: Given two activities $\alpha$ and $\beta$, whenever $\alpha$ is executed, is $\beta$ always executed after that? In this paper, we tackled the problem in terms of Workflow Petri nets (WF-nets for short). Our results are (i) the response property problem for acyclic WF-nets is decidable, (ii) the problem is intractable for acyclic asymmetric choice (AC) WF-nets, and (iii) the problem for acyclic bridge-less well-structured WF-nets is solvable in polynomial time. We illustrated the usefulness of the procedure with an application example.

key words: the response property, Petri net, process tree, computational complexity, polynomial time algorithm

1. Introduction

Response property is a kind of liveness property. The response property problem is defined as follows: Given two activities $\alpha$ and $\beta$, whenever $\alpha$ is executed, is $\beta$ always executed after that? Response property analysis is important to verify the correctness of a workflow process. As a simple example, Fig. 1 shows a Petri net\cite{1} representing a workflow for an online ordering process. There are two cases in this workflow. In case 1, if the order is less than $2000$ (Price less than $2000$), then the ID of the buyer will be checked by the system. Next, the buyer can proceed to order (Make Order). Finally, the system will accept the order (Instant Acceptance). In case 2, if the order is more than $2000$ (Price more than $2000$), then the buyer must proceed directly to order the item (Make Order) and the manager will manually accept the order (Manager Acceptance). In this case, whenever Price more than $2000$ is executed, is Manager Acceptance always executed? The answer is no. This is because even if the price is above $2000$ (Price more than $2000$), then the system can accept the order (Instant Acceptance) instead of the manager. In this situation, the business rule is violated. Hence, it is important to distinguish the execution of each task to verify whether the workflow follows the business rule or not.

In Ref.\cite{2}, Hichami et al. proposed a method to verify response property based on Petri nets. The feature of the method is to introduce abstraction technique in order to reduce the computation time of model checking. However, the method lacks formality and did not take account of two or more firing of transition.

In this paper, we take an analytical approach to the response property problem. We first formalize the response property problem and then reveal the computational complexity of the problem. Then, we propose a polynomial time procedure to solve the problem with a representational bias of Petri net called as process tree. This paper is organized as follows: After the introduction in Sect. 1, Sect. 2 gives the formal definition and properties of Petri nets. In Sect. 3, we give the formal definition of the response property problem in terms of Petri nets. In Sect. 4, we reveal its decidability and computational complexity. Then, in Sect. 5, we give a polynomial time procedure to solve the problem and show an application example of our approach in Sect. 6. Finally, we give the conclusion and the future work.

2. Preliminaries

(1) Petri nets and workflow nets

A Petri net is a three tuple $N = (P, T, A)$, where $P$, $T$, and $A$ ($\subseteq (P \times T) \cup (T \times P)$) are finite sets of places, transitions, and arcs, respectively. Let $x$ be a node of $N$. $\bullet x$ and $x \bullet$ respectively denote $\{y | (y, x) \in A\}$ and $\{y | (x, y) \in A\}$. A marking (or a state) is a mapping $M: P \to \{0, 1, 2, \ldots\}$. We represent $M$ as a bag over $P$: $M=[p^{|M(p)|}]_{p \in P}$, $M(p) < 0$. A transition $t$ is said to be firable in $M$ if $M \geq \bullet t$. Firing $t$ in $M$ results in a new marking $M' = M \cup \{t_{\bullet} \}$. This is denoted by $M[N, t]M'$. A marking $M_n$ is said to be reachable from a marking $M_0$ if there exists a transition firing sequence $\sigma=t_1t_2\cdots t_n$ such that $M_0[N, t_1]M_1[N, t_2]M_2\cdots[M, t_n]M_n$. $M_0$ denotes the ini-

---

Fig. 1 A workflow $N_1$ of an online ordering service.
N is said to be a workflow net (WF-net for short) [4] if (i) N has a single source place p₁ and a single sink place p₀ and (ii) every node is on a path from p₁ to p₀. Each transition represents an action. For any WF-net, M₀ is given as [p₁]. We make N strongly connected by connecting p₀ to p₁ via an additional transition r'. The resulting Petri net is called the short-circuited net of N, and is denoted by \( \overline{N} = (P, T \cup \{r'\}, A \cup \{(p₀, r'), (r', p₁)\}) \). There are four important subclasses of WF-nets used in this paper: well-structured (WS for short), extended free choice (EFC for short), asymmetric choice (AC for short), and process tree based (PTB for short). N is said to be WS if there are neither disjoint paths from a place to a transition nor disjoint paths from a transition to a place in \( \overline{N} \). N is said to be EFC if \( \forall p₁, p₂ \in P : p₁ \cap p₂ \neq \emptyset \Rightarrow p₁ = p₂ \). N is said to be AC if \( \forall p₁, p₂ \in P : p₁ \cap p₂ = \emptyset \Rightarrow p₁ \subseteq p₂ \) or \( p₂ \subseteq p₁ \). A WF-net is a marked graph (MG for short) if \( \forall p \in P, [p] = [p'] = 1 \).

Soundness is a criterion of correctness for WF-nets. A WF-net N is said to be sound if (i) \( \forall M \in R(N, [p₁]) : \exists M' \in R(N, M) : M' \geq [p₀] \); (ii) \( \forall M \in R(N, [p₁]) : M \geq [p₀] \Rightarrow M' = [p₀] \); and (iii) There is no dead transition in \( (N, [p₁]) \). The soundness of EFC WF-nets or WS WF-nets can be solved in polynomial time [4].

(2) Process Tree and Process Tree Based WF-net
A process tree is a tree representation of a process [5]. A process is represented by an action which is defined by an atomic action [6]. Each leaf node and each internal node respectively represents an action and an operator in the process.

Definition 1: [7] The set \( \Pi \) of process trees \( \pi \) is as follows:

(i) If \( \alpha \) is an action, then \( \alpha \in \Pi \).
(ii) If \( \oplus \) is an operator and \( \alpha₁, \alpha₂, \ldots, \alphaₙ \) are actions, then \( \oplus(\alpha₁, \alpha₂, \ldots, \alphaₙ) \in \Pi \).
(iii) If \( \oplus \) is an operator and \( \pi₁, \pi₂, \ldots, \piₙ \in \Pi \), then \( \oplus(\pi₁, \pi₂, \ldots, \piₙ) \in \Pi \).

We use three operators standardized by the Workflow Management Coalition (WFMC for short) [4], [8]: sequence (\( \rightarrow \)), exclusive-choice (\( \times \)), and parallel (\( \wedge \)). Each operator can be translated to a part of a WF-net (See Fig. 2).

Next, we define a WF-net that can be converted into process tree called as Process Tree-based WF-net (PTB WF-net for short) [7], [9]. In PTB WF-net, each transition represents only unique action.

Definition 2 (PTB WF-net [7]): For any process tree \( \pi \), let \( N \) be the WF-net itself and \( N_i \ (i=1, 2, \ldots, n) \) be the subnet in \( N \).

(i) If \( \pi \) is an action, a WF-net \( N \) which consists of a transition representing the action and its input and output places is PTB.
(ii) If \( \pi \) is \( \oplus(\pi₁, \pi₂, \ldots, \piₙ) \), then let \( N₁, N₂, \ldots, Nₙ \) be respectively PTB WF-nets representing sub-process trees \( \pi₁, \pi₂, \ldots, \piₙ \).

a. If \( \oplus \) is sequence, then a WF-net constructed by concatenating \( N₁, N₂, \ldots, Nₙ \) which connects the sink place of \( N_i \) with the source place of \( N_{i+1} \ (1 \leq i < n) \) is PTB.

b. If \( \oplus \) is exclusive choice, then a WF-net constructed by bundling PTB WF-nets \( N₁, N₂, \ldots, Nₙ \) which forms a selection of concurrent paths between their source places and sink places is PTB.

c. If \( \oplus \) is parallel, then a WF-net constructed by joining respectively all source places with a transition \( t_i \) and sink places with a transition \( t_O \) of PTB WF-net \( N₁, N₂, \ldots, Nₙ \) is PTB. □

Noted that a process tree of PTB WF-net is an ordered
tree. Soundness and well-structuredness are necessary condition to represent process tree [9].

3. Response Property

Response property is a kind of liveness property that given two transitions \( t \) and \( u \), if \( t \) is fired, \( u \) has to be eventually fired after that. We restricted the analysis of response property to acyclic WF-net. This restriction is reasonable because acyclic WF-nets are applicable to analyze most actual workflows.

Let \( L(N, [p_I]) \) be the set of any firing sequence that transforms \([p_I]\) to any dead marking. Note that in any acyclic WF-net \( N=(P, T, A) \), each marking in \( R(N, [p_I]) \) eventually reaches a dead marking.

**Definition 3** (Response Property): For a transition pair \( t \) and \( u \) in an acyclic WF-net \( N \), \( u \) is said to respond to \( t \) if for all \( \sigma \in L(N, [p_I]) \) : \( \forall i \in \{1, 2, \ldots, |\sigma|\} : (\sigma(i) = t \Rightarrow \exists j \in \{(i + 1), (i + 2), \ldots, |\sigma|\} : \sigma(j) = u) \).

**Definition 4** (Response Property Problem): Instance: Acyclic WF-net \( N \), Transitions \( t \) and \( u \) of \( N \).
Question: Does \( u \) respond to \( t \)?

Let us consider two instances of the response property problem as examples. The first instance is shown in Fig. 1. In this instance, every transition fires at most once. The second instance is shown in Fig. 3. In this instance, some of transitions fire twice or more. The instance in Fig. 1 has been considered in Sect. 1. We concretely discuss the reason in this section.

**Instance 1:**

Instance: Acyclic WF-net \( (N_1, [p_I]) \), transitions \( t_1^1 \) and \( t_1^6 \) (See Fig. 1).
Question: Does \( t_1^6 \) respond to \( t_1^1 \)?

The answer for Instance 1 is no, because \( t_1^6 \) does not always fire for \([p_I]_1[\sigma, N_1][p_0^1]\) because there exist firing sequences \( t_1^1 t_1^2 t_1^4 \) or \( t_1^1 t_1^2 t_1^3 t_1^5 \) transforming \([p_I]\) to \([p_0^1]\). Existence of place branching at \( p_1^1 \) allows the \( t_1^4 \) to be skipped as \( t_1^3 \) is fired. Hence, \( t_1^6 \) does not respond to \( t_1^1 \). This means that Petri net structure plays an important role to the response property and the reachability of marking is important to the analysis of response property as discussed in Sect. 1.

**Instance 2:**

Instance: Acyclic WF-net \( (N_2, [p_I^2]) \), transitions \( t_2^5 \) and \( t_2^6 \) (See Fig. 3).
Question: Does \( t_2^6 \) respond to \( t_2^5 \)?

The answer for Instance 2 is yes. \( t_2^5 \) fires once for \([p_I^2]_2[\sigma, N_2][p_0^2, p_0^3]\) when there are firing sequences \( t_2^5 t_2^1 t_2^3 t_2^4 \) or \( t_2^5 t_2^1 t_2^3 t_2^4 t_2^6 \) in \( L(N_2, [p_I]) \). The value of \( i \) of \( t_2^5 \) is 5 which is lower than the value of \( j \) of \( t_2^6 \) that is 7 for both firing sequences. Then, transition \( t_2^6 \) fires only after transition \( t_2^5 \) fires. Concretely, the order of transition firing plays an important role in deciding the response property.

As a transition \( t \) can fire many times, it is important to check each firing of transition. However, we can simplify the checking process with the property below:

**Property 1:** For a transition pair \( t \) and \( u \) in an acyclic WF-net \( N \), \( u \) is said to respond to \( t \) if for all \( \sigma \in L(N, [p_I]) \) : \( \forall i \in \{1, 2, \ldots, |\sigma|\} : (\sigma(i) = t \Rightarrow \exists j \in \{(i + 1), (i + 2), \ldots, \sigma(|\sigma|)) : \sigma(j) = u) \).

**Proof:** Let \( \sigma \) be any firing sequence in \( L(N, [p_I]) \). The “only-if” part is obvious. We only have to prove the “if” part. We divide the proof into two cases where \( t \notin \sigma \) and \( t \in \sigma \) of \( \sigma \).

Let \( k \) be any position of \( \sigma \). \( k \leq l_{\text{max}} \) is obvious. \( t \) at position \( k \) is followed by the last one. From the assumption \( (\exists j \in \{(l_{\text{max}} + 1), (l_{\text{max}} + 2), \ldots, \sigma(|\sigma|)) : \sigma(j) = u) \), the last one is followed by \( u \). From the transitive law, we have that \( t \) at any position is followed by \( u \).

Q.E.D.

4. Computational Complexity and Intractability

Let us consider the decidability of the response property problem.

**Theorem 1:** The response property problem is decidable for acyclic WF-nets.

**Proof:** Let \( N \) be any acyclic WF-net. Since \( N \) is acyclic, \( L(N, [p_I]) \) is obviously finite. To check whether a transition \( \alpha \) responds to another transition \( \beta \) in \( N \), we have only to check the precedence relation between \( \alpha \) and \( \beta \) for each sequence \( \sigma \) of \( L(N, [p_I]) \). That is, if \( \sigma \) includes \( \alpha \), then \( \beta \) must appear at a later position than the last \( \alpha \). Q.E.D.

Let us consider the computational complexity of the response property problem for a subclass of acyclic WF-nets, called acyclic AC WF-nets.

We show that an NP-complete problem, called 3-conjunctive normal form boolean satisfiability problem (3-CNF-SAT for short), can be transformed to the response property problem of acyclic AC WF-nets.

**Definition 5** (3-CNF-SAT):

Instance: Expression \( E \) of 3-conjunctive normal form that has \( n \) boolean variables and \( m \) clauses.

**Question:** Is there an assignment of variables satisfying

\[^1\{i, (i + 1), \ldots, j\} \text{denotes the set of integers from } i \text{ to } j.\]
Let us consider an acyclic AC WF-nets shown in Fig. 4. We need to check does $\beta$ respond to $\alpha$? The constructed WF-net shows that if $E=\text{true}$ then marking $[p_2, p_4]$ is reachable from $[p_1]$. However, $\beta$ will not always fire because of the conflict with $\gamma$. In this case, $\beta$ does not respond to $\alpha$. We give the following theorem on the complexity:

**Theorem 2:** The response property problem is co-NP hard for acyclic AC WF-nets.

**Proof:** We prove the NP-hardness by a reduction from 3-CNF-SAT. Let $E$ be an expression of 3-CNF-SAT which has $n$ boolean variables $x_1, x_2, \cdots, x_n$ and $m$ clauses $c_1, c_2, \cdots, c_m$. A literal $\ell_i$ is either a variable $x_i$ or its negation $\overline{x_i}$. Without loss of generality, it can be assumed that $E$ has all of $x_i$'s and $\overline{x_i}$'s ($1 \leq i \leq n$), and $m \geq 3$.

We construct a WF-net $N_E = (P_E, T_E, A_E)$ with two transitions $\alpha$ and $\beta$, and show that $E$ is satisfiable if and only if $\beta$ does not respond to $\alpha$ in $(N_E, [p_1])$. $N_E = (P_E, T_E, A_E)$ is given as follows:

$$P_E = \{p_1, p_2, p_3, p_4, p_0\} \cup \bigcup_{i=1}^{m} \{q_i\} \cup \bigcup_{j=1}^{n} \{c_j\}$$

$$T_E = \{t_1, t_2, t_3, \alpha, \beta, \gamma\} \cup \bigcup_{i=1}^{m} \{x_i, \overline{x_i}\}$$

$$A_E = \{(t_1, t_2), (t_1, p_1), (p_1, t_2), (t_2, p_4), (p_2, \gamma), (t_1, p_3), (p_3, \alpha), (\alpha, p_4), (p_4, \gamma), (\gamma, p_0), (p_4, \beta), (\beta, p_0)\}$$

$$\cup \bigcup_{i=1}^{n} \{(t_2, q_i), (q_i, x_i), (q_i, \overline{x_i})\}$$

$$\cup \bigcup_{k=1}^{m} \bigcup_{j=1}^{n} \{(t_k, c_j) | \text{\ell}_k \text{ is the k-th literal of clause } c_j\}$$

$$\cup \bigcup_{j=3}^{m} \{(c_j, t_3)\}$$

$N_E$ is an AC WF-net because places $p_2$ and $p_4$ share an output transition $\gamma$ while $p_1$ has another output transition $\beta$; Places $c_1, c_2, \cdots, c_m$ share only one output transition $t_3$, and the other places share no output transition. $N_E$ can be constructed in polynomial time, because it consists of $(n+m+6)$ places, $(2n+6)$ transitions, and $(3n+4m+12)$ arcs.

The proof of “if” part: Let $\mu$ denote an assignment of variables satisfying $E=\text{true}$, and let $\ell_1, \ell_2, \cdots, \ell_n$ be the literals mapped to true by $\mu$. By the construction of $N_{E_1}$, we have

$$[p_1] [N_E, \ell_1 t_2] [p_1, p_3]$$

$$[N_E, \ell_1 t_2 \cdots \ell_n] M.$$

Since $\mu$ satisfies $E$, for each clause $c_j$ ($1 \leq j \leq m$), there exists a literal $\ell_j$ ($1 \leq i \leq n$) in $c_j$. Therefore place $c_j$ is marked by firing $\ell_j$ i.e. $M \geq [c_1, c_2, \cdots, c_m, p_3]$. Let $M' = M - [c_1, c_2, \cdots, c_m, p_3]$.

$$M = M' \cup [c_1, c_2, \cdots, c_m, p_3]$$

$$[N_E, \alpha] M' \cup [p_2, p_3]$$

$$[N_E, \alpha \gamma] M' \cup [p_0]$$

$\beta$ is dead in $(N_E, M' \cup [p_0])$. Thus, $\beta$ does not respond to $\alpha$.

The proof of “only-if” part: Let $\mu$ denote any assignment of variables satisfying $E=\text{false}$. Since $\mu$ does not satisfy $E$, there exists a clause $c_j$ ($c_1(c_1, c_2, \cdots, c_m)$) mapped to $\text{false}$ by $\mu$. Let $\ell_1', \ell_2', \ell_3'$ denote the literals in $c_j$. Since the corresponding transitions $\ell_1', \ell_2', \ell_3'$ do not fire, their common output place, i.e. place $c_j$, is never marked. $c_j$ is an input place of transition $t_3$, so $t_3$ is dead. We have

$$[p_1] [N_E, \ell_1] [p_1, p_3]$$

$$[N_E, \star] [p_3] \cup M$$

$$[N_E, \alpha] [p_4] \cup M$$

$$[N_E, \star] [p_4] \cup M'$$

$$[N_E, \beta] [p_0] \cup M'$$

Thus $\beta$ responds to $\alpha$.

Q.E.D.

This means that the original problem is intractable for acyclic AC WF-nets. For example, let us consider the following boolean expression:

$$E_1 = (x_1 \lor x_2 \lor x_3) \land (x_1 \lor x_2 \lor x_3) \land (x_1 \lor x_2 \lor x_3) \land (x_1 \lor x_2 \lor x_3)$$

$E_1$ is satisfiable by choosing $x_1=\text{true}$, $x_2=\text{true}$, $x_3=\text{true}$. Figure 4 shows the Petri net $N_{E_1}$ constructed from $E_1$. $\beta$ does not respond to $\alpha$, because

$$[p_1] [N_E, \ell_1 t_2] [p_1, p_3]$$

$$[N_E, x_1 x_2 x_3] [c_1, c_2, c_3, c_4, c_5, p_3]$$

$$[N_E, t_3] [c_1, c_2, c_3, c_4, p_2, p_3]$$

$$[N_E, \alpha] [c_1, c_3, c_5, p_2, p_4]$$

$$[N_E, \gamma] [c_1, c_3, c_5, p_0].$$

5. Polynomial-Time Verification of Response Property

In this section, we propose a polynomial time procedure to decide response property.

5.1 Structural Analysis for Response Property Problem

A transition with two or more output places is called a
transition-split, and a transition with two or more input places is called a transition-join. Similarly, a place with two or more output transitions is called a place-split, and a place with two or more input transitions is called a place-join [10].

Acyclic bridge-less WS WF-net has balance structure of transition/place-splits with transition/place-joins. Two disjoint paths initiated by a transition-split are joined by a place-join. Two disjoint paths initiated by a place-split are joined by a transition-join. The structure of a transition-split with a transition-join is called the transition split-join structure and the structure of a place-split with a place-join is called the place split-join. Acyclic bridge-less WS WF-nets have nesting structure of transition split-join structure and place split-join structure [10]. This enables us to check the response property not in the node-level but in those split-join structure-level.

Let us consider two instances of acyclic bridge-less WS WF-net shown in Fig. 5 to solve the response property problem. Both instances have one transition split-join structure and one place split-join structure. In $N_1$, $\beta$ responds to $\alpha$. $\alpha$ appeared in the disjoint path between place-split $p_1$ and place-join $p_2$. $\beta$ appeared in the disjoint path between transition-split $t_3$ and transition-join $t_5$. Since $\beta$ is included within a transition split-join structure, $\beta$ always fire after $\alpha$. In $N_2$, $\beta$ does not respond to $\alpha$. $\alpha$ appeared in the disjoint path between transition-split $t_1$ and transition-join $t_3$. $\beta$ appeared in the disjoint path between place-split $p_5$ and place-join $p_6$. Place-join $p_2$ allows $\beta$ to be skipped when $t_4$ is fired. The result of the response property is changed by the combination of transition/place split-join structure and the position of $\alpha$ and $\beta$. Therefore, we need to consider the combination of disjoint path and position of transitions within the WF-net.

\[ NCA \text{ is an exclusive-choice operator (\times) and not the parent of } \alpha. \text{ Therefore, we need to consider the combination of disjoint path and position of transitions within the WF-net.} \]

5.2 Utilization of Process Tree

Acyclic bridge-less WS WF-net is known as PTB WF-net [9]. Figure 6 shows the general form of process tree of PTB WF-net. We can represent transition split-join as parallel operator ($\lor$). We can also represent place split-join as exclusive-choice operator ($\times$). Let $\text{v}_\text{NCA}$ be the nearest common ancestor node of $\alpha$ and $\beta$ in a process tree $\pi$ of a WF-net $N$ where there exists the shortest path from $\alpha$ via $\text{v}_\text{NCA}$ to $\beta$ (See Fig. 6).

First, we consider two cases when $\text{v}_\text{NCA}$ is an exclusive-choice operator ($\times$) or a parallel operator ($\lor$). If $\text{v}_\text{NCA}$ is an exclusive-choice operator ($\times$), $\beta$ does not respond to $\alpha$. We can give the following Lemma:

**Lemma 1:** In a PTB WF-net, if $\text{v}_\text{NCA}=\times$, $t_j$ does not respond to $t_i$ (See Fig. 2 (b)).

**Proof:** See Appendix A.1. \( \text{Q.E.D.} \)

If $\text{v}_\text{NCA}$ is a parallel operator ($\lor$), $\beta$ does not respond to $\alpha$. We can give the following Lemma:

**Lemma 2:** In a PTB WF-net, if $\text{v}_\text{NCA}=\lor$, $t_j$ does not respond to $t_i$ (See Fig. 2 (c)).

**Proof:** See Appendix A.2. \( \text{Q.E.D.} \)

Next, if $\text{v}_\text{NCA}$ is a sequence operator ($\rightarrow$) and the parent of $\alpha$ and $\beta$, $\beta$ responds to $\alpha$. However, if $\text{v}_\text{NCA}$ is a sequence operator ($\rightarrow$) and not the parent of $\alpha$ and $\beta$, $\beta$ does not always respond to $\alpha$. Also, if the position of $\alpha$ is on the left of $\beta$ in $\pi_N$, then $\beta$ responds to $\alpha$ because $\alpha$ will fire before $\beta$ in the sequence construct in $N$. However, if $\alpha$ is on the right of $\beta$, then $\beta$ does not respond to $\alpha$ because $\beta$ will fire before $\alpha$.

Let us consider again the two instances of acyclic bridge-less WS WF-net shown in Fig. 5. We converted both instances into process tree (See the right side of Fig. 5). Let us focus on the dotted line on both process tree. For process tree of $N_1$, we find that if parallel operator ($\lor$) exists between $\text{v}_\text{NCA}$ and $\beta$, then $\beta$ responds to $\alpha$. Otherwise, for process tree of $N_2$, we find that if exclusive-choice operator ($\times$) exists between $\text{v}_\text{NCA}$ and $\beta$, then $\beta$ does not respond to $\alpha$. We can give the following theorem:

**Theorem 3:** For a PTB WF-net $N$ with transitions $\alpha$ and $\beta$, $\beta$ responds to $\alpha$ iff in the process tree $\Pi_N$.
(i) The nearest common ancestor \( v_{NCA} \) of \( \alpha \) and \( \beta \) is a sequence operator \((\rightarrow)\);
(ii) The position of \( \alpha \) is on the left side of \( \beta \)^1; and
(iii) There is no exclusive-choice operator \((\times)\) on the path between \( v_{NCA} \) and \( \beta \).

\( \square \)

**Proof :** The proof of the “if” part: Conditions (i), (ii) and (iii) can be illustrated by the process tree \( \Pi_N \) shown in Fig. 6. \( \Theta_\alpha \) denotes the ancestor nodes of \( \alpha \) and the descendents of \( v_{NCA} \). \( \Theta_\beta \) is defined similarly as \( \Theta_\alpha \). \( \Pi_N \) can be illustrated with the WF-net \( N \) shown in Fig. 7. \( N \) consists of \( N_A \), \( N_B \), and \( N_C \). If \( v_{NCA} \) is a non-parent node, then \( \alpha \) and \( \beta \) is in a different connected subnets in \( N \). \( \alpha \) is in \( N_A \), \( \beta \) is in \( N_B \), and \( N_C \) connects \( N_A \) and \( N_B \). If \( v_{NCA} \) can always fire after \( \alpha \) and \( \beta \) can always be enabled after \( t_i \) since \( N_B \) has no choice. WF-net \( N \) is sound. Therefore, \( N \) can be regarded as an interconnection of MG components and from the property of MG, there exists a path connecting \( \alpha \) and \( \beta \). Hence, \( \beta \) responds to \( \alpha \) in \( N \).

The proof of the “only-if” part: We show the proof using a process tree \( \Pi_N \) which has a subtree \( \pi_\phi \) with root \( v_{NCA} \) where \( \alpha \) and \( \beta \) are the leaf nodes.

In Condition (i), if \( v_{NCA}=x \), based on Lemma 1, the firing is selective where only either transition in the subtree \( \pi_\alpha \) or \( \pi_\beta \) will fire. If \( v_{NCA}=x \), based on Lemma 2, there exist partial firing sequences \( \alpha \sigma \beta \) and \( \beta \sigma \alpha \) where there is a case when \( \beta \) can fire before \( \alpha \).

In Condition (ii), if \( \alpha \) is on the right of \( \beta \) then \( i>j \). Therefore, there exists a case where \( \beta \) fires before \( \alpha \) in \( N \). \( N \) is acyclic, therefore the firing of each transitions is only once where there exists a firing sequence that satisfies \( \forall \ell_{max} \in \{1,2,\ldots,|\sigma|\} : \sigma|_{\ell_{max}}=\alpha \Rightarrow \exists j \in \{(\ell_{max}+1), (\ell_{max}+2), \ldots, |\sigma|\} : \sigma|_j=\beta \) once \( \beta \) fires before \( \alpha \).

In Condition (iii), if there exists an exclusive-choice operator between the path from \( v_{NCA} \) to \( \beta \) such that \( \Theta_\beta=x \), then \( \beta \) is in an exclusive-choice construct. Therefore, there exists a case where \( \beta \) will not fire.

Theorem 3 implies the condition to decide the response property problem. So we construct a polynomial time procedure based on Theorem 3. The process tree can be traversed with Breadth-first Search (BFS) [11]. We give the procedure as follows:

\( \ll \text{Decision of Response Property} \gg \)

**Input:** PTB WF-net \( N = (P, T, A) \), transitions \( \alpha, \beta \in T \).

**Output:** Does \( \beta \) respond to \( \alpha \)?

1° Convert \( N \) into process tree \( \Pi_N \) with \( \ll \text{Process Tree Conversion Algorithm} \gg \) [9].

2° Check Conditions (i), (ii) and (iii) of Theorem 3.

2-1° Check Condition (i).

Let the \( \rho_\alpha \) be the path from root to \( \alpha \) and \( \rho_\beta \) be the path from root to \( \beta \). Let \( \rho_C \) be the common part of \( \rho_\alpha \) and \( \rho_\beta \). If the last node of \( \rho_C \) is not sequence operator \((\rightarrow)\), then output no and stop.

2-2° Check Condition (ii).

Let \( i \) be the position of \( \alpha \) and \( j \) be the position of \( \beta \). If \( i<j \), then output no and stop.

2-3° Check Condition (iii).

Backtrack from \( \beta \) to \( v_{NCA} \). If a visited node is exclusive-choice operator \((\times)\), then output no and stop.

3° Output yes and stop.

**Property 2:** The following problem can be solved in polynomial time: Given a PTB WF-net \( N \) with two transitions \( \alpha \) and \( \beta \), to decide whether \( \beta \) responds to \( \alpha \). \( \square \)

**Proof :** The \( \ll \text{Process Tree Conversion Algorithm} \gg \) takes \( O(|P| + |T|) \) and response property check takes \( O(|T|) \) based on BFS.

Q.E.D.

6. Application Example

In this section, we show an application example by showing the response property analysis of a secure online shopping process (see Fig. 1). We assume that we need to check the responsiveness between these two important activities of the online shopping process which is “Price more than \$2000” and “Manager Acceptance”. Let us verify the instance using our procedure.

In Step 1°, we convert \( N_1 \) to process tree \( \Pi_{N_1} \) (See Fig. 8). In Step 2°, we check the Condition (i), (ii) and (iii) of Theorem 3. In Step 2-1°, we check the nearest common node \( v_{NCA} \) of \( \alpha \) and \( \beta \). We obtained that \( \rho_{\alpha}=t_1^1 \rightarrow \), \( \rho_{\beta}=t_2^1 \times \rightarrow \), and \( \rho_C=\rightarrow \), hence \( v_{NCA}=\rightarrow \). In Step 2-2°, we check Condition (ii). From \( i=3 \) and \( j=6 \), we obtained that

\[ \star \]

\[ \rightarrow \]

\[ t_2 \]

\[ t_3 \]

\[ t_4 \]

\[ t_5 \]

\[ t_6 \]

\[ t_7 \]

Fig. 7 Illustration of the proof of “if” part of Theorem 3.

\[ ^{1} \text{Noted that a process tree is an ordered tree.} \]
\(i < j\). \(t'_i\) is on the left side of \(t'_j\). Then, in Step 2-3 we check Condition (iii). We backtrack from \(t'_0\) to \(v_{WCA}\) and found \(x \rightarrow\) on the path. There is a \(x\) on the path. In \(N_1\), there is a place \(p_1\) which constructs exclusive-choice operator (\(\times\)). Either \(t'_2\) or \(t'_0\) will be fired. So, \(t_6\) does not always fire. Finally, in Step 3', the procedure outputs no. Thus, we obtained that \(t'_0\) does not respond to \(t'_1\).

7. Conclusion

In this paper, we gave the formal definition of the response property problem. We also showed that the problem is decidable. We also revealed the intractability for AC WF-nets. We constructed a polynomial time procedure for PTB WF-nets based on the given theorem. Response property analysis is important in terms of analyzing the execution of an action whenever another action is executed in order to adapt to specifications in systems and business processes. In our future work, we plan to investigate the response property problem for cyclic WF-nets.

Acknowledgements

This research was partially supported by Interface Corporation, Japan and Malaysia-Japan Higher Education Program.

Appendix

A.1 Proof of Lemma 1: If \(v_{WCA} = x\), then the PTB WF-net \(N\) is constructed by bundling \(N_1, N_2, \cdots, N_n\) which forms a selective concurrent paths between their source places and sink places. In \(N_n, p_1 = p_1^{(1)} = p_1^{(2)} = \cdots = p_1^{(n)}\) and \(p_0 = p_0^{(1)} = p_0^{(2)} = \cdots = p_0^{(n)}\). Note that only one path will allow the transition \(t_n\) in \(N_n\) to fire. Therefore there exists different firing sequences \(\sigma_1, \sigma_2, \cdots, \sigma_n\) for each path in \(N_1, N_2, \cdots, N_n\) for \([p_1][N_n, \sigma_n][p_0]\). Hence, only either \(t_i\) or \(t_j\) will fire.

Q.E.D.

A.2 Proof of Lemma 2: If \(v_{NCA} = x\), then the PTB WF-net \(N\) is constructed by joining respectively all source places of the concurrent paths with a transition \(t_i\) and sink places with a transition \(t_0\) of PTB WF-net \(N_1, N_2, \cdots, N_n\). Since \(N_1, N_2, \cdots, N_n\) and \(N_0\) are sound, \([p_1^{(1)}], p_1^{(2)} = \cdots = p_1^{(n)}\) is reachable from \([p_1^{(1)}], p_1^{(2)} = \cdots = p_1^{(n)}\], \(p_0^{(1)} = p_0^{(2)} = \cdots = p_0^{(n)}\) are connected to \(p_0\) via another additional transition \(t_0\). Therefore there exists partial firing sequences \(\sigma = t_i\sigma_1\) and \(\sigma = t_j\sigma_1\), where \(\sigma_1\) is all possible firing sequences of transitions in \(T\) for \([p_1^{(1)}], p_1^{(2)} = \cdots = p_1^{(n)}][N_n, \sigma][p_0^{(1)} = p_0^{(2)} = \cdots = p_0^{(n)}].\) Hence, based on \(\sigma, t_j\) will not always fire after \(t_i\).

Q.E.D.

References

[1] T. Murata, “Petri nets: Properties, analysis and applications,” Proc. IEEE, vol.77, no.4, pp.541–580, 1989.
[2] O.E. Hichami, M.A. Achhab, I. Berrada, R. Oucheikh, B.E.E. Mohajir, “An approach of optimization and formal verification of workflow Petri nets,” Journal of Theoretical and Applied Informatics, vol.61, no.3, pp.431–495, 2014.
[3] S. Yamaguchi, “Polynomial time verification of reachability in sound extended free-choice workflow net,” IEICE Trans. Fundamentals, vol.E97-A, no.2, pp.468–475, Feb. 2014.
[4] W.M.P. van der Aalst and K.M. van Hee, Workflow Management: Models, Methods, and Systems, The MIT Press, 2002.
[5] W. van der Aalst, J. Buijs, and B. van Dongen, “Towards improving the representational bias of process mining,” Data-Driven Process Discovery and Analysis, Lecture Notes in Business Information Processing, vol.116, pp.39–54, Springer-Verlag, Berlin, 2012.
[6] R. De Nicola, “Process algebras,” Encyclopedia of Parallel Computing, pp.1624–1636, 2011.
[7] M.A.B. Ahmadon and S. Yamaguchi, “State number calculation problem of workflow nets,” IEICE Trans. Inf. & Syst., vol.E98-D, no.6, pp.1128–1136, June 2015.
[8] W.M.P. van der Aalst, “Verification of workflow nets,” Application and Theory of Petri Nets 1997, Lecture Notes in Computer Science, vol.1248, pp.407–426, Springer, 1997.
[9] M.A.B. Ahmadon and S. Yamaguchi, “Convertibility and conversion algorithm for well-structured workflow net to process tree,” Proc. CANDAR 2013, pp.122–127, 2013.
[10] N. Qu, S. Yamaguchi, and Q.-W. Ge, “Parallel degree of well-structured workflow nets,” IEICE Trans. Fundamentals, vol.E93-A, no.12, pp.2730–2739, Dec. 2010.
[11] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Introduction to Algorithms, Second Edition, pp.998–1003, The MIT Press, 2001.

Muhammad Syaﬁq Bin Ab Malek received his B.E. degree from Yamaguchi University, Japan in 2016. He is a graduate student in the Graduate School of Sciences and Technology for Innovation, Yamaguchi University, Japan. His research interest includes Petri net and its application.

Mohd Anuaruddin Bin Ahmadon received his B.E., M.E. and D.E. degrees from Yamaguchi University, Japan in 2014, 2015 and 2017, respectively. He is currently an Assistant Professor in the Graduate School of Sciences and Technology for Innovation, Yamaguchi University, Japan. His research interest includes Petri net and its application to system analysis. He is a member of IEEE.
Shingo Yamaguchi received the B.E., M.E. and D.E. degrees from Yamaguchi University, Japan, in 1992, 1994 and 2002, respectively. He was a Visiting Scholar in the Department of Computer Science at University of Illinois at Chicago, United States, in 2007. He is currently a Professor in the Graduate School of Sciences and Technology for Innovation, Yamaguchi University, Japan. His research interests are in the area of net theory and its applications. He is a senior member of IPSJ and IEEE.