Computer Science • 21(4) 2020 https://doi.org/10.7494/csci.2020.21.4.3728

Adel Benamira

CAUSAL REVERSIBILITY IN INDIVIDUAL TOKEN INTERPRETATION OF PETRI NETS

Abstract *Causal reversibility in concurrent systems means that events that the origin of other events can only be undone after undoing its consequences. In opposition to backtracking, events that are independent of each other can be reversed in an arbitrary order; in other words, we have flexible reversibility with respect to a causality relationship. An implementation of individual token interpretation of Petri Nets (IPNs) has been proposed by Rob Van Glabbeek et al.; the present paper investigates a study of causal reversibility within IPNs. Given N as an IPN, by adding an intuitive firing rule to undo transitions according to the causality relationship, the coherence of N is assured; i.e., the set of all reachable states of N in the reversible version and that of the original one are identical. Furthermore, reversibility in N is flexible, and their initial state can be accessible in reverse from any state. In this paper, an approach for controlling causal-reversibility within IPNs is proposed.*

Keywords reversibility, concurrent systems, Petri Nets, causality

Citation Computer Science 21(4) 2020: 489–511

1. Introduction

Reversibility in concurrent systems has received much attention over the last decade. The concept of reversible computing is based on a combination of forward and backward computation (in contrast to traditional forward computation). This has been sprung from [21,28], in which the authors confirm that only irreversible computations need to consume energy; therefore, reversible computation is favorable in low-energy computing. In the literature, there are a surprising number of reversible computation studies that have emerged for modeling systems that are naturally reversible, such as biological system [9,15,40], chemical systems [9,42], and quantum computations [1,10]. Reversibility has also been used in transactions systems [14,29] and in debugging systems [26,32,33], without forgetting space exploration problems [11,17,27,29]. In this kind of system, each one can automatically go backward to a specific state (e.g., a stable state) in case of error or checkpoints.

Reversibility in a sequential context is well-understood [34, 44]. To reverse the execution of a calculus, we can recursively undo the last action performed by this calculus. Since there is no concept of a last action in a concurrent context[1], the definition of reversibility in which context is trickier and more complex to analyse it.

As mentioned by [43], three forms of undoing events exist. Backtracking is the first (and simplest) one in which events are undone in the inverse order that they occurred. Causal reversing (the second form) means that events that cause other events can only be undone after the caused events are undone first; thus, independent events can be undone in any order irrespective of the order that they have actually occurred. Opposite to causal reversing, the third form is out-of-causal reversing.

The foundational studies of causal reversibility in concurrent computations have been largely deployed. The notion of causally reversibility was first introduced in the process calculus RCCS [13]. To this end, the authors associate a memory to each process, which accumulates the necessary information to capture the history of its attached process to backtrack. Alternatively, Phillips and Ulidowski proposed a technique for reversing process calculi without using memories [41]. In this technique, they generated unique identifiers for each new execution of actions. The interesting proprieties must be assured by causal reversibility are resumed by (i) the coherence of a system, which means that the modified system has the same states as the original one, and (ii) flexible reversibility, in which events that are independent of each other can be reversed in an arbitrary order.

In the works discussed above, there are no directives to go forward nor backward. Hence, reversibility is uncontrollable. In [31], the authors have classified controlling reversibility with respect to a causality relationship in three categories: internal control, external control, and semantic control [12, 14, 30, 40].

It is well-known that, in the literature, causal reversibility can distinguish concurrency from non-determinism and discern between instances of the same action

[1] Many actions are executed concurrently.

(auto-concurrency). Hence, it is naturally able to define causal reversibility within causality semantics. Furthermore, it's well-understood [7, 8, 18] and largely defined within Petri nets (which are general formal descriptive models of concurrent systems) and can also be used as an underlying semantics model. In [23,25], causality semantics have been formalized by the notion of an individual token interpretation of Petri nets.

In [23] explains that, in the individual token interpretation of Petri nets, one can distinguish different tokens residing in the same place, keeping track of where they come from. if a transition fires by using a token that has been produced by another transition, there is a causal link between the two. Consequently, the causal relationships between the transitions in a run of net can always be described by means of a partial order. On the other hand, tokens cannot be distinguished in the collective token interpretation.

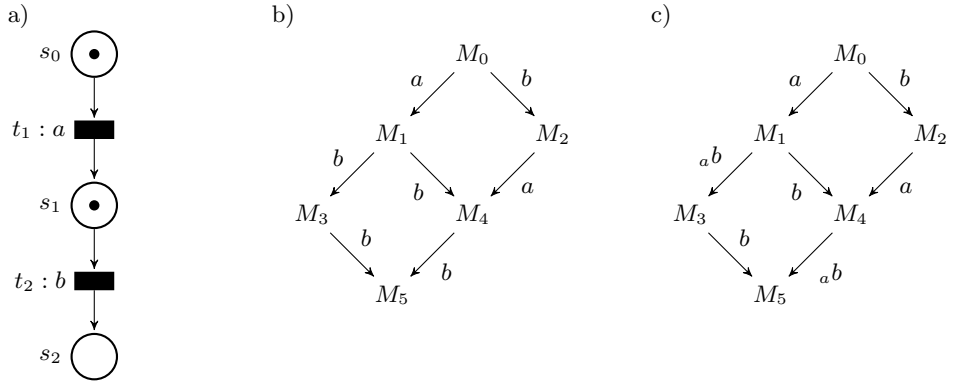

Figure 1. Collective and individual token interpretations of Petri net: a) Petri net N; b) CT-Marking graph of N; c) IT-Marking graph of N

The net of Figure 1 [23] illustrates the difference between the collective token interpretations (see Figure 1b) and the individual token interpretation (see Figure 1c) of given Petri net N in Figure 1a. In this net, the transitions labeled a and b are independent actions; thus, we can fire each once. After a has fired, there are two tokens in the place of s_1. According to the individual token philosophy, it makes a difference which of these tokens is used in firing b. if the token that was already there is used (which must certainly be the case if b happens before the token from a arrives), transitions a and b are causally independent (case 1). If the token that was produced by a is used, b is causally dependent on a and noted by $_ab$ (case 2). In an opposite manner, the collective token interpretation cannot distinguish these two cases.

1.1. Previous work

Few researchers have addressed reversible computation within the Petri net context. The first study was proposed in [3, 5]; more recently, causal reversibility has been introduced within an individual token interpretation [4]. Similar to the Phillips and

Ulidowski approach, a unique identifier for each firing is assured without adding memories. [37] proposes an approach for controlling reversibility by associating transitions with conditions whose satisfaction/violation allows for the execution of transitions in the forward/reverse direction, respectively. These works have been limited to a subclass of Petri nets that are acyclic.

1.2. Our contribution

We believe that the restriction to acyclic Petri nets is due to the mechanism that are used to distinguish tokens; hence, the work reported in the current paper was motivated by the powerful of the manner of [23] in which tokens can be distinguished. Indeed, the present work finds, as in [4], an interesting set of results with any restriction[2].

Precisely, the contribution of this paper is based on causality semantics that uses Glabbeek's presentation of the individual token interpretation of Petri nets [23], noted in the present work by IPN. A study of causal reversibility within IPNs is proposed and in which interesting proprieties have been verified, such as the coherence of a system and flexible reversibility. Furthermore, the initial state of a system can be accessible in reverse from any one. In this paper, a control causal reversibility within IPNs is proposed as well. According to Mazurkiewicz's trace equivalence, flexible reversibility in IPNs is founded on the interesting theorem that is proven in Section.3.

1.3. Paper organization

The paper is organized as follows. The second section defines the individual token interpretation of Petri nets under the interleaving semantics. The third section examines the application of Mazurkiewicz's trace equivalence to IPNs in which an interesting theorem (to hold flexible reversibility) has been proven. The fourth section is the core of the present paper in which the coherence and flexible reversibility of a given IPN are assured and the initial state of a system can be accessible in reverse from any one. Section 5 defines a new model (States-based Control Causal-Reversible IPNs) in which causal reversible is controlled by a rollback specification. This paper is ended by some conclusions of the present work.

2. Preliminaries

In [23], the reader finds a formal presentation of the individual token interpretation of Petri nets using step semantics; however, the contribution of this paper is defined using interleaving semantics. Hence, the present section redefines it under interleaving semantics.

In [23], each token has been identified by a triple (u, k, s) such that s is the place where the token is created and u is the transition firing that brought it there. The n tokens that are created in s by u have been distinguished by giving ordinal

[2]For any cyclic or acyclic Petri net.

numbers $k = 0, 1, 2, ..., n-1$. Hence, the tokens that are defined as tuple (u,k,s) allow us to distinguish them in the following situations: (i) tokens are located in different places (held by s); (ii) tokens are located in the same place but created by different firings (see u); and (iii) tokens are located in the same place and created by the same firing (assured by k). We take $(*, k, s)$ for initial tokens of s, we have $u = *$.

The (X, t) pair introduces the firing of transition t with consuming the set of tokens X. Function β is defined from the tokens to the places where they occur by $\beta(u, k, s) = s$, and η from transition firings u such that $u = (X, t)$ to the transition that fires by $\eta(u) = t$. Function β extends to a function from sets of tokens X to sets of places by $\beta(X) = \{\beta(n) | \forall n \in X\}$.

Definition 2.1. *A (labeled, marked)Petri net is a tuple $N = (S, T, F, I, L)$ where:*

- *S and T are disjoint sets (of places and transitions);*
- *$F : (S \times T \cup T \times S) \to \mathbb{N}$ (the flow relationship including arc weights);*
- *$I : S \to \mathbb{N}$ (the initial marking);*
- *$L : T \longrightarrow A$, for A a set of actions, the labeling function.*

Definition 2.2. *Given Petri net $N = (S, T, F, I, L)$, sets of tokens $S_\bullet$ and transition firing $T_\bullet$ of N are recursively defined by:*

- *$(*, k, s) \in S_\bullet$ for $s \in S$ and $k < I(s)$;*
- *$(u, k, s) \in S_\bullet$ for $s \in S$, $u \in T_\bullet$ and $k < F(\eta(u), s)$;*
- *$(X, t) \in T_\bullet$ for $t \in T$ and $X \subseteq S_\bullet$ such that $\beta(X) =^\bullet t \neq \phi$ where $^\bullet t = \{s | \forall s \in S. F(s, t) \neq 0\}$.*

Labeling function $L_\bullet : T_\bullet \longrightarrow A$ on transition firings is given by $L_\bullet(t) = L(\eta(t))$. An individual marking of N is a subset of tokens $M \subseteq S_\bullet$. The initial individual marking $I_\bullet$ is defined as $\forall (*, k, s) \in S_\bullet$ implies that $(*, k, s) \in I_\bullet$.

Definition 2.3. *For a firing $u \in T_\bullet$ such that $u = (X, t)$, let $^\bullet u = X$ and $u^\bullet = \{(u, k, s) | K < F(\eta(u), s)\}$ be the set of input tokens and the set of output tokens of u. Transition t is enabled under an individual marking $M \in S_\bullet$ if $^\bullet u \subseteq M$. In this case, t can fire under M, yielding $M' = (M \setminus^\bullet u) \cup u^\bullet$, written as $M \xrightarrow{u}_\bullet M'$ or $M[u\rangle M'$.*

For each marking M, $[M\rangle$ is the set of markings reachable from M. Hence, $[I_\bullet\rangle$ is the set of markings reachable from $I_\bullet$, in the other words it's the set of all marking reachable of N.

To explain how IPNs can preserve a causal relationship between actions, given the Petri net N of Figure 1 in which the initial individual marking $I_\bullet$ is the set $\{(*, 0, s_0), (*, 0, s_1)\}$. From this state, transition t_1 (respectively, t_2) is enabled; hence, we have firing $u_1 = (\{(*, 0, s_0)\}, t_1)$ (respectively, $u_2 = (\{(*, 0, s_1)\}, t_2)$). Firing u_1 creates state M_1 such that $M_1 = \{(u_1, 0, s_1), (*, 0, s_1)\}$ from which we have two possible firings: the first one is caused by the initial token $(*, 0, s_1)$, and the second one by the consequence of u_1; i.e., token $(u_1, 0, s_1)$, noted respectively by u_3 and u_4 such that $u_3 = (\{(*, 0, s_1)\}, t_2)$ and $u_4 = (\{(u_1, 0, s_1)\}, t_2)$. These mean that we can differentiate the firing of t_2 that is caused by the execution of t_1 to the one which is caused by the initial token of s_1. In the collective token interpretation, we do not have this possibility (see Figure 1b).

The marking graph of N is given as Figure 2 such that:

- $X_1 = \{(*, 0, s_0)\}$ and $M_1 = \{((X_1, t_1), 0, s_1), (*, 0, s_1)\}$;
- $X_2 = \{(*, 0, s_1)\}$ and $M_2 = \{((X_2, t_2), 0, s_2), (*, 0, s_0)\}\}$;
- $X_3 = \{((X_1, t_1), 0, s_1)\}$ and $M_3 = \{((X_3, t_2), 0, s_2), (*, 0, s_1)\}$;
- $M_4 = \{((X_1, t_1), 0, s_1), ((X_2, t_2), 0, s_2)\}$;
- $M_5 = \{((X_3, t_2), 0, s_2), ((X_2, t_2), 0, s_2)\}$.

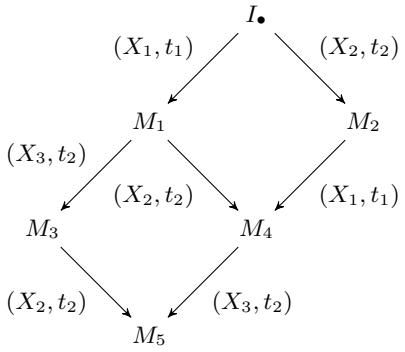

Figure 2. Marking graph within individual token interpretation

3. Trace equivalence in IPNs

Let σ and σ' be two sequences, and let u and u' be two independent actions. By the definition of Mazurkiewicz's trace equivalence [35, 36], sequences $\sigma.u.u'.\sigma'$ and $\sigma.u'.u.\sigma'$ are two equivalent traces. This section examines Mazurkiewicz's trace equivalence within the IPN context and in which it's found that any two sequences are Mazurkiewicz's equivalent traces only if they share a same source and a same target marking; i.e., $M[\sigma.u.u'.\sigma'\rangle M'$ and $M[\sigma.u'.u.\sigma'\rangle M'$.

When we give a sequence to reverse it, two possible reversing sequences can emerge: (i) the reversing of itself (the usual backtracking), and (ii) the reversing of their equivalent sequence. Thus, it can be shown that the equivalence concept is a pile foundation of the flexible reversibility definition.

In this section, the Mazurkiewicz's trace equivalence is redefined in the IPN context; to this end, we begin by defining the independent relationships between the firings in the IPN context. Intuitively, two firings (X, t) and (X', t') are in conflict if and only if $X \cap X' \neq \phi$. We say that a firing causes another one if and only if the second one consumes the tokens that are directly or indirectly created by the first one; i.e., (X, t) causes (X', t') if and only if $(X, t)^\bullet \cap X' \cap \hbar(X') \neq \phi$, the indirect tokens are given by function $\hbar$ (see Definition 3.1). Concerning the independent relationship, we can say that two firings are independent if and only if they are not in conflict and neither causes the other one.

Definition 3.1. *Let $X \subseteq S_\bullet$. The function $\hbar$ is defined recursively as: $\hbar(X) = \{X' \cup \hbar(X') | \forall((X', t), k, s) \in X\}$.*

Definition 3.2. *Let $u, u' \in T_\bullet$ such that $u = (X, t)$ and $u' = (X', t')$. Independent relationship $\Im \subseteq T_\bullet \times T_\bullet$ is defined by $(u, u') \in \Im$ if and only if*

1. *$X \cap X' = \phi$ and*
2. *$u^\bullet \cap X' \cap \hbar(X') = \phi$ and $u'^\bullet \cap X \cap \hbar(X) = \phi$.*

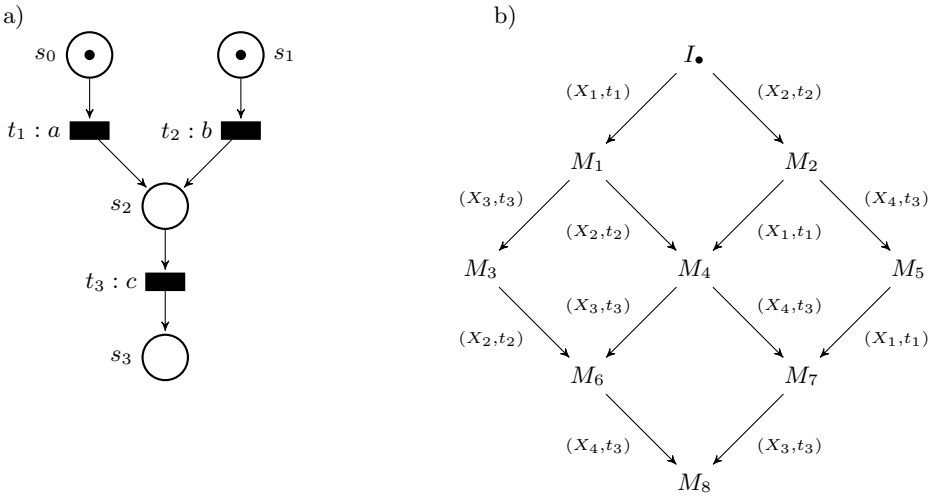

Figure 3. Marking graph of IPN: a) Petri net N; b) the marking graph of N

For instance, let N be an IPN of Figure 3. This net presents an auto-concurrence of action c, which is caused by the parallel execution of actions a and b. The marking graph of N is presented as Figure 3b such that:

- $I_\bullet = \{(*, 0, s_0), (*, 0, s_1)\}$;
- $X_1 = \{(*, 0, s_0)\}$;
- $M_1 = \{(*, 0, s_1), (\{(X_1, t_1)\}, 0, s_2)\}$;
- $X_2 = \{(*, 0, s_1)\}$;
- $M_2 = \{(*, 0, s_0), (\{(X_2, t_2)\}, 0, s_2)\}$;
- $X_3 = \{(\{(X_1, t_1)\}, 0, s_2)\}$;
- $M_3 = \{(*, 0, s_1), (\{(X_3, t_3)\}, 0, s_3)\}$;
- $M_4 = \{(\{(X_1, t_1)\}, 0, s_2), (\{(X_2, t_2)\}, 0, s_2)\}$;
- $X_4 = \{(\{(X_2, t_2)\}, 0, s_2)\}$;
- $M_5 = \{(*, 0, s_0), (\{(X_4, t_3)\}, 0, s_3)\}$;
- $M_6 = \{(\{(X_3, t_3)\}, 0, s_3), (\{(X_2, t_2)\}, 0, s_2)\}$;
- $M_7 = \{(\{(X_1, t_1)\}, 0, s_2), (\{(X_4, t_3)\}, 0, s_3)\}$;
- $M_8 = \{(\{(X_3, t_3)\}, 0, s_3), (\{(X_4, t_3)\}, 0, s_3)\}$.

From this graph, we can take that:

- $((X_1,t_1),(X_2,t_2)) \in \mathfrak{I}$ because $X_1 \cap X_2 = \phi$, $(X_1,t_1)^\bullet \cap X_2 \cap \hbar(X_2) = \phi$ and $(X_2,t_2)^\bullet \cap X_1 \cap \hbar(X_1) = \phi$ such that $(X_1,t_1)^\bullet = \{(\{(X_1,t_1)\},0,s_2)\}$, $\hbar(X_2) = \phi$, $(X_2,t_2)^\bullet = \{(\{(X_2,t_2)\},0,s_2)\}$ and $\hbar(X_1) = \phi$;
- $((X_3,t_3),(X_2,t_2)) \in \mathfrak{I}$ because $X_3 \cap X_2 = \phi$, $(X_3,t_3)^\bullet \cap X_2 \cap \hbar(X_2) = \phi$ and $(X_2,t_2)^\bullet \cap X_3 \cap \hbar(X_3) = \phi$ such that $(X_3,t_3)^\bullet = \{(\{(X_3,t_3)\},0,s_3)\}$, $\hbar(X_2) = \phi$, $(X_2,t_2)^\bullet = \{(\{(X_2,t_2)\},0,s_2)\}$ and $\hbar(X_3) = \{(*,0,s_0)\}$;
- $((X_1,t_1),(X_4,t_3)) \in \mathfrak{I}$ because $X_1 \cap X_4 = \phi$, $(X_1,t_1)^\bullet \cap X_4 \cap \hbar(X_4) = \phi$ and $(X_2,t_2)^\bullet \cap X_1 \cap \hbar(X_1) = \phi$ such that $(X_1,t_1)^\bullet = \{(\{(X_1,t_1)\},0,s_2)\}$, $\hbar(X_4) = \{(*,0,s_1)\}$, $(X_4,t_3)^\bullet = \{(\{(X_4,t_3)\},0,s_3)\}$ and $\hbar(X_1) = \phi$;
- $((X_3,t_2),(X_4,t_3)) \in \mathfrak{I}$ because $X_3 \cap X_4 = \phi$, $(X_1,t_1)^\bullet \cap \hbar(X_4) = \phi$ and $(X_2,t_2)^\bullet \cap \hbar(X_1) = \phi$ such that $(X_3,t_3)^\bullet = \{(\{(X_3,t_3)\},0,s_3)\}$, $\hbar(X_4) = \{(*,0,s_1)\}$, $(X_4,t_3)^\bullet = \{(\{(X_4,t_3)\},0,s_3)\}$ and $\hbar(X_3) = \{(*,0,s_0)\}$.

Mazurkiewicz's trace equivalence is defined in the IPN context as follows:

Definition 3.3. *Let σ,σ' be two sequences and let u,u' be two firings such that $(u,u') \in \mathfrak{I}$. Relationship $\sim \subseteq T_\bullet^* \times T_\bullet^*$ is defined by:*

if $\sigma.u.u'.\sigma', \sigma.u'.u.\sigma' \in T_\bullet^*$, then $\sigma.u.u'.\sigma' \sim \sigma.u'.u.\sigma'$.

Given sequences $\sigma = (X_1,t_1).(X_2,t_2).(X_3,t_3).(X_4,t_3)$ and $\sigma' = (X_2,t_2).(X_4,t_3).(X_1,t_1).(X_3,t_3)$ from Figure 3b,

we can take:

- $\sigma \sim (X_2,t_2).(X_1,t_1).(X_3,t_3).(X_4,t_3)$ from the fact that $((X_1,t_1),(X_2,t_2)) \in \mathfrak{I}$;
- since $((X_3,t_3),(X_4,t_3)) \in \mathfrak{I}$, we have $(X_2,t_2).(X_1,t_1).(X_3,t_3).(X_4,t_3) \sim (X_2,t_2).(X_1,t_1).(X_4,t_3).(X_3,t_3)$;
- $(X_2,t_2).(X_1,t_1).(X_4,t_3).(X_3,t_3) \sim (X_2,t_2).(X_4,t_3.).(X_1,t_1).(X_3,t_3) = \sigma'$; thus, $((X_1,t_1),(X_4,t_3)) \in \mathfrak{I}$.

Hence, $\sigma \sim \sigma'$.

Any one can observe that, in this IPN, for any two sequences σ and σ' such that $M[\sigma\rangle M'$ and $M[\sigma'\rangle M'$, we have $\sigma \sim \sigma'$. In the following, we find that it remains true for any IPN (see Theorem 3.1); however, before this, it is important to report that σ and σ' have exactly the same set of actions.

Lemma 3.1. *Let σ and σ' be two sequences such that $M[\sigma\rangle M'$ and $M[\sigma'\rangle M'$. We can have $\|\sigma\| = \|\sigma'\|$.*

With $\|u_1.u_2...u_n\| = \{u_i | \forall i \in 1..n\}$.

Proof 3.1. *We use proof by contradiction. Suppose the claim is false; this implies that $M[u_1\rangle M_1[u_2\rangle M_2...[u_n\rangle M'$ and $M[u'_1\rangle M'_1[u_2\rangle M'_2...[u'_m\rangle M'$ such that $n \neq m$. So, have $u = (X,t) \in \|\sigma\|$ such that $u \notin \|\sigma'\|$ means that we haven't any state M'_i such that $X \in M'_i$. As a consequence, X does not appear in any token history of M'. By Definition 2.3, this will be impossible.*

Theorem 3.1. *Let $\sigma,\sigma' \in T_\bullet^*$, $\sigma \sim \sigma'$ if and only if $M[\sigma\rangle M'$ and $M[\sigma'\rangle M'$.*

Proof 3.2. *We prove $\sigma \sim \sigma' \Rightarrow M[\sigma\rangle M' \wedge M[\sigma'\rangle M'$ and vice-versa.*

1. *First, we show $\sigma \sim \sigma' \Rightarrow M[\sigma\rangle M' \wedge M[\sigma'\rangle M'$: From the definition of $\sim$, we have $\sigma_1.\sigma_2.\sigma_3 \sim \sigma_1.\sigma_2'.\sigma_3$ such that $\sigma = \sigma_1.\sigma_2.\sigma_3$ and $\sigma' = \sigma_1.\sigma_2'.\sigma_3$. From the fact that $[...\rangle$ is a function and each sequence in an IPN is unique, we take $M[\sigma_1\rangle M_1[\sigma_2\rangle M_2[\sigma_3\rangle M'$ and $M[\sigma_1\rangle M_1[\sigma_2'\rangle M_2[\sigma_3\rangle M'$. Thus, $M[\sigma\rangle M' \wedge M[\sigma'\rangle M'$ is held.*
2. *Now, we show $M[\sigma\rangle M' \wedge M[\sigma'\rangle M' \Rightarrow \sigma \sim \sigma'$: we remember from Lemma 3.1 that $\|\sigma\| = \|\sigma'\|$. We use proof by contradiction. Suppose the claim is false; this implies that:*
   a) *a sub-sequence $u.u_1$ exists in σ and a sub-sequence $u.u_2$ in σ' such that $(u_1, u_2) \notin \mathfrak{I}$; this implies the following: if u_1 causes u_2 in σ (respectively if u_2 causes u_1), then $u_1 \notin \|\sigma'\|$ (respectively $u_2 \notin \|\sigma\|$); this will be impossible, in fact that $\|\sigma\| = \|\sigma'\|$;*
   b) *or, they exist a sub-sequence $u_1.u$ of σ and a sub-sequence $u_2.u$ of σ' such that $(u_1, u_2) \notin \mathfrak{I}$, this implies the following: if u_1 causes u_2 in σ (respectively if u_2 causes u_1), then $u_1 \notin \|\sigma'\|$ (respectively $u_2 \notin \|\sigma\|$); this will be impossible, in fact that $\|\sigma\| = \|\sigma'\|$.*

4. Reversibility in IPNs

Causal reversibility means that an event that causes other events can only be undone after the caused events are undone first. Our contribution consists of modeling this kind of reversibility within IPNs. Given $N = (S, T, F, I, L)$ as an IPN, let $u \in T_\bullet$ such that $u = (X, t)$, the forward firing of t can be shown as the destruction of all tokens of set X and then the production of tokens set $u^\bullet$. Therefore (and opposite), the undoing (backward) of u consumes $u^\bullet$ and produces X. In this section, we show how this intuitive undoing vision can assure both the coherence of a system and flexible reversibility.

First, the formal definitions of forward and backward firings within IPNs are given.

Definition 4.1. *For a firing $u \in T_\bullet$ in a Petri net such that $u = (X, t)$, let ${}^\bullet u = X$ and $u^\bullet = \{(u, k, s) | K < F(\eta(u), s)\}$ be the set on input tokens and the set of output tokens of u, respectively.*

1. *forward firing: transition t is enabled under an individual marking $M \in S_\bullet$ if ${}^\bullet u \subseteq M$; in this case, t can fire under M, yielding $M' = (M \setminus {}^\bullet u) \cup u^\bullet$, written as $M \xrightarrow{u}_{\bullet f} M'$ or $M[u\rangle_f M'$.*
2. *backward firing: transition t can be undone under an individual marking $M' \in S_\bullet$ if $u^\bullet \subseteq M'$; in this case, the undo of t can fire under M', yielding $M = (M' \setminus u^\bullet) \cup^\bullet u$, written as $M' \xrightarrow{u}_{\bullet b} M$ or $M'[u\rangle_b M$.*

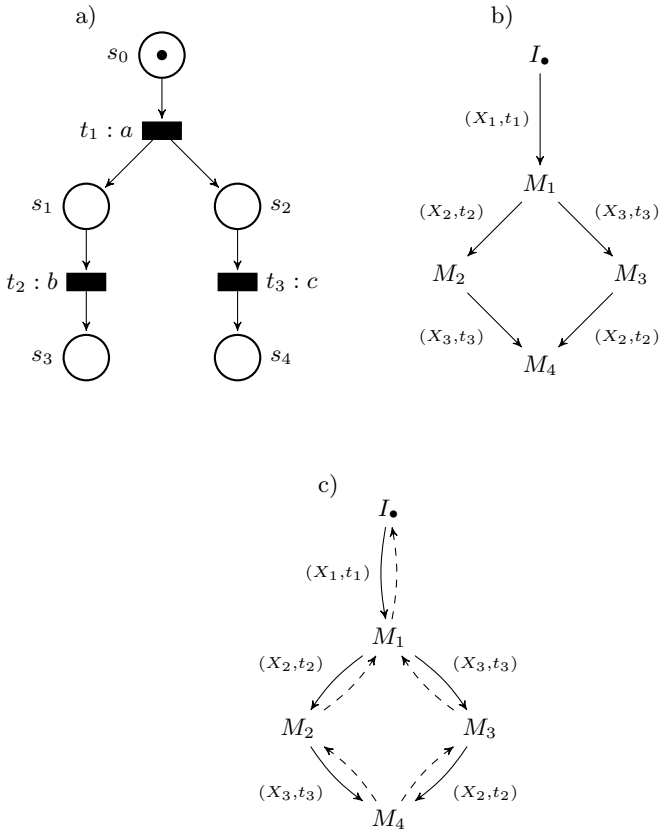

Figure 4. Reversibility in individual token interpretation Petri net: a) Petri net N; b) usual marking graph; c) marking graph with causal reversibility

For example, let N be the IPN of Figure 4, and let $I_\bullet = \{(*, 0, s_0)\}$ be the initial marking state of N. Marking $I_\bullet$ allows us to only perform (X_1, t_1) such that $X_1 = \{(*, 0, s_0)\}$. By Definition 4.1.1, we have $I_\bullet[(X_1, t_1)\rangle_f M_1$ such that $M_1 = \{(\{(X_1, t_1)\}, 0, s_1), (\{(X_1, t_1)\}, 0, s_2)\}$.

At state M_1, we have three possible actions: perform (X_2, t_2), perform (X_3, t_3), or undo (X_1, t_1):

1. $M_1[(X_2, t_2)\rangle_f M_2$ such that $X_2 = \{(\{(X_1, t_1)\}, 0, s_1)\}$ and $M_2 = \{(\{(X_1, t_1)\}, 0, s_2), (\{(X_2, t_2)\}, 0, s_3)\}$;

2. $M_1[(X_2, t_3)\rangle_f M_3$ such that $X_3 = \{(\{(X_1, t_1)\}, 0, s_2)\}$ and $M_3 = \{(\{(X_1, t_1)\}, 0, s_1), (\{(X_3, t_3)\}, 0, s_4)\}$;

3. $M_1[(X_1, t_1)\rangle_b I_\bullet$.

The undoing of (X_2, t_2) or the performing of (X_3, t_3) will be possible from M_2:

1. $M_2[(X_3, t_3)\rangle_f M_4$ such that
   $M_4 = \{(\{(X_2, t_2)\}, 0, s_3), (\{(X_3, t_3)\}, 0, s_4)\}$;
2. $M_2[(X_2, t_2)\rangle_b M_1$.

From M_3, we have:

1. $M_3[(X_2, t_2)\rangle_f M_4$ such that
   $M_4 = \{(\{(X_2, t_2)\}, 0, s_3), (\{(X_3, t_3)\}, 0, s_4)\}$;
2. $M_3[(X_3, t_3)\rangle_b M_1$.

The marking M_4 allows us to undo (X_3, t_3) or (X_2, t_2):

1. $M_4[(X_3, t_3)\rangle_b M_2$;
2. $M_4[(X_2, t_2)\rangle_b M_3$.

All of these will be resumed as the marking graph of Figure 4c, in which a dashed arc means the undoing of an action. The graph of Figure 4b is the usual marking graph of N (without reversibility). Each one can be see that (i) the two previous graphs have exactly the same marking set, and (ii) in Figure 4c, the undoing of sequence $I_\bullet[(X_1, t_1)\rangle_f M_1[(X_2, t_2)\rangle_f M_2[(X_3, t_3)\rangle_f M_4$ is realized either by backtracking $M_4[(X_3, t_3)\rangle_b M_2[[(X_2, t_2)\rangle_b M_1(X_1, t_1)\rangle_b I_\bullet$, or by sequence $M_4[(X_2, t_2)\rangle_b M_3[[(X_3, t_3)\rangle_b M_1(X_1, t_1)\rangle_b I_\bullet$; i.e., by the backtracking of its equivalent sequence $I_\bullet[(X_1, t_1)\rangle_f M_1[(X_3, t_3)\rangle_f M_2[(X_2, t_2)\rangle_f M_4$.

In this instance, it is obvious to affirm both the coherence of the system and flexible reversibility. The question is now to know whether these proprieties will be held for any reversible IPN. In the follow, the coherence and flexible reversibility for a given reversible IPN are proven. Furthermore, the initial marking is reachable by reversibility from any marking state.

Definition 4.2. *Let $u \in T_\bullet$. Relationships $\mathfrak{F}, \mathfrak{B} \subseteq T_\bullet \times S_\bullet \times S_\bullet$ are defined by*

- $(u, M, M') \in \mathfrak{F}$ *if and only if* $M \xrightarrow{u}_{\bullet^f} M'$;
- $(u, M, M') \in \mathfrak{B}$ *if and only if* $M' \xrightarrow{u}_{\bullet^b} M$.

In a reversible IPN, the undoing of a firing (X, t) generates any new marking state. It is just the reverse of the past state of (X, t). To check this, it must be proven that $\mathfrak{B}$ is a partial function that is the inverse function of $\mathfrak{F}$.

Lemma 4.1. *$\mathfrak{F}$ and $\mathfrak{B}$ are partial functions on $T_\bullet \times S_\bullet \longrightarrow S_\bullet$.*

Proof 4.1. *Let $u \in T_\bullet$ such that $u = (X, t)$, and let $M_1, M_2, M_3 \in S_\bullet$. Function $\mathfrak{F}$ is a partial function from $T_\bullet \times S_\bullet$ to $S_\bullet$ if and only if: $\mathfrak{F}(u, M_1) = M_2$ and $\mathfrak{F}(u, M_1) = M_3$, then $M_2 = M_3$.*

With the aim of obtaining a contradiction, we assume that $M_2 \neq M_3$. By the definition of forward firing, we have $M_2 = (M_1 \setminus X) \cup u^\bullet$ and $M_3 = (M_1 \setminus X) \cup u^\bullet$; thus, $M_1 \neq M_1$, $X \neq X$ and $\{(u, k, s) | K < F(\eta(u), s)\} \neq \{(u, k, s) | K < F(\eta(u), s)\}$. Therefore, they are a contradiction.

This is a similar proof for $\mathfrak{B}$.

In the rest, an application $\mathfrak{F}(u, M)$ (respectively, $\mathfrak{B}(u, M)$) is noted by $\mathfrak{F}_u(M)$ (respectively, $\mathfrak{B}_u(M)$).

Proposition 4.1. *Let $u \in T_\bullet$, the function $\mathfrak{B}_u$ is the inverse function of $\mathfrak{F}_u$.*

Proof 4.2. *Function $\mathfrak{B}_u$ is the inverse function of $\mathfrak{F}_u$ if and only if $\mathfrak{B}_u \circ \mathfrak{F}_u(M) = M$. We have $\mathfrak{B}_u \circ \mathfrak{F}_u(M) = ((M \setminus X) \cup u^\bullet) \setminus u^\bullet \cup X = M$. Thus, $\mathfrak{B}_u \circ \mathfrak{F}_u(M) = M$.*

Proposition 4.2. *In a reversible IPN, the undoing of a firing (X, t) generates any new marking state. This is just the reverse of the past marking of (X, t).*

Proof 4.3. *From the fact that $\mathfrak{B}_u$ is the inverse function of $\mathfrak{F}_u$.*

To define our calculus, functions $\mathfrak{F}$ and $\mathfrak{B}$ will be extended by composition to $T_\bullet^*$.

Definition 4.3. *Let $\sigma = u_1.u_2....u_n$ be a sequence in $T_\bullet^*$ such that $M_0 \xrightarrow{u_1}_{\bullet f} M_1 \xrightarrow{u_2}_{\bullet f} M_2... \xrightarrow{u_n}_{\bullet f} M_n$. Function $\mathfrak{F} : T_\bullet^* \times S_\bullet \longrightarrow S_\bullet$ is defined by $\mathfrak{F}_\sigma = \mathfrak{F}_{u_n} \circ \mathfrak{F}_{u_{n-1}} ... \mathfrak{F}_{u_2} \circ \mathfrak{F}_{u_1}$.*

In what follows, we note $M[\sigma\rangle_f M'$ for $\mathfrak{F}_\sigma(M) = M'$.

For any marking M, set $[M\rangle_f$ is the set of markings reachable from M. Hence, $[I_\bullet\rangle_f$ is the set of all markings that are reachable of N.

Definition 4.4. *Let $\sigma = u_1.u_2. \ldots u_n$ be a sequence in $T_\bullet^*$ such that $M_n \xrightarrow{u_n}_{\bullet b} M_{n-1} \xrightarrow{u_{n-1}}_{\bullet b} M_{n-2}... \xrightarrow{u_1}_{\bullet b} M_0$. Function $\mathfrak{B} : T_\bullet^* \times S_\bullet \longrightarrow S_\bullet$ is defined as follows: $\mathfrak{B}_\sigma = \mathfrak{B}_{u_1} \circ \mathfrak{B}_{u_2} ... \mathfrak{B}_{u_{n-1}} \circ \mathfrak{B}_{u_n}$.*

We note $M'[\sigma\rangle_b M$ for $\mathfrak{B}_\sigma(M') = M$.

For any marking M, set $[M\rangle_b$ is the set of markings reachable by reversibility from M; thus, $[I_\bullet\rangle_b = \{M | \forall \sigma \in T_\bullet^* : M[\sigma\rangle_b I_\bullet\}$.

Proposition 4.3. *In a reversible IPN, for any $\sigma \in T_\bullet$ such that $\mathfrak{F}_\sigma(M) = M'$, we have $\mathfrak{B}_\sigma(M') = M$*

Proof 4.4. *Let n be the length of sequence σ. We will use induction on n.*

1. *$n = 1$: this is evident by the fact that $\mathfrak{B}$ is the inverse function of $\mathfrak{F}$.*
2. *Suppose that, if n is held, then $\mathfrak{B}_\sigma(M') = M$. If we take $\mathfrak{F}_{\sigma.u}(M) = M$" such that $u \in T_\bullet$, then we wish to show that $\mathfrak{B}_{\sigma.u}(M") = M$. From Definition 4.3, we have $\mathfrak{F}_u(M') = M$"; therefore, $\mathfrak{B}_u(M") = M'$. As a result, $\mathfrak{B}_\sigma \circ \mathfrak{B}_u(M") = M$.*

Proposition 4.3 confirms that a backtracking of any sequence will be possible. Therefore, the initial marking is reachable by backtracking from any marking state of the system (see Proposition 4.4).

Proposition 4.4. *In a reversible IPN, for any $M \in S_\bullet$, $\sigma \in T_\bullet^*$ exists such that $\mathfrak{B}_\sigma(M) = I_\bullet$.*

Proof 4.5. *$M \in S_\bullet$ means that $\exists \sigma \in T_\bullet^*$ such that $\mathfrak{F}_\sigma(I_\bullet) = M$. Thus, from Proposition 4.3, we have $\mathfrak{B}_\sigma(M) = I_\bullet$*

Now, we can enunciate the theorem of the coherence of the system.

Theorem 4.1. (The coherence of the system)

Given $N = (S, T, F, I, L)$ being an IPN, the set of all reachable states of N and that the reversible N will be identical: $[I_\bullet\rangle = [I_\bullet\rangle_f = [I_\bullet\rangle_b$.

Proof 4.6. *By definition, $[I_\bullet\rangle = [I_\bullet\rangle_f$. From definitions $[I_\bullet\rangle_f$ and $[I_\bullet\rangle_b$ and from Proposition 4.4, we have $[I_\bullet\rangle_f = [I_\bullet\rangle_b$.*

Theorem 4.2. (Flexible Reversibility)

For any $\sigma \in T_\bullet^*$ such that $\mathfrak{F}_\sigma(M) = M'$, if $\sigma' \in T_\bullet^*$ exists such that $\sigma \sim \sigma'$, then $\mathfrak{B}_\sigma(M') = \mathfrak{B}_{\sigma'}(M') = M$.

Proof 4.7. *From Theorem 3.1, we take $\mathfrak{F}_{\sigma'}(M) = M'$, and from Proposition 4.3, we can take $\mathfrak{B}_\sigma(M') = \mathfrak{B}_{\sigma'}(M') = M$.*

This theorem means that the reversibility of σ will be given by backtracking or by the backtracking of its equivalent sequences.

5. States-based controlling causal-reversibility

Unfortunately, controlling reversibility in the distributed system context is trickier and not evident. Furthermore, we do not have a global view of a system. For example, taken the distributed system of Figure 5a, which is composed of two subsystems (Sub_1 and Sub_2), dispersed in two different locations. We note that this vision of distributed systems has been proposed in [24]. The undoing of Sub_2 vis-a-vis their local vision means that it must undo t_{22} and then undo t_{21}; thus, $\overline{t_{22}}.\overline{t_{21}}$ is the associated sequence to this backtracking. This reflect is held if only if the subsystem is independent of the components of the system. However, in the global vision, we must also undo t_{12} of Sub_1 and its consequences; hence, it will be possible to have two equivalent backtracking sequences[3]: $\overline{t_{22}}.\overline{t_{13}}.\overline{t_{12}}.\overline{t_{21}}$ and $\overline{t_{13}}.\overline{t_{22}}.\overline{t_{12}}.\overline{t_{21}}$.

The above example is introduced to explain the difficulty of controlling reversibility using a given sequence. The alternative is the employment of the state space of a system, one can proposes a pair of states (trigger, target) such as when trigger state R is spotted, rollback[4] (R, G) can be used to go back to target state G. However, we have fallen in the same previous problem. Indeed, the (R, G) pair will be defined from the global system; i.e., we must dispose their enumeration states.

To get around this obstacle, this section proposes a control causal-reversibility within IPNs by giving an implicit rollback. Let us go back to our distributed system of Figure 5a. When one wants to undo the second subsystem, the trigger state is the one that contains token $(\_, \_, s_{22})$, and the target state includes token $(*, 0, s_{20})$. Hence, the specification of this rollback can be taken as a pair of predicates as $(\Phi(M), \Psi(M))$ such that $\Phi(M) = (\_, \_, s_{22}) \in M$ and $\Psi(M) = (*, 0, s_{20}) \in M$; i.e., a rollback specification will be defined from the description of the system (e.g., tokens and places) instead of their global state space. A rollback is executed if a marking state exists that satisfies Φ. As a conclusion, our control causal-reversibility approach is defined as a control causal-reversibility using a given enumerate rollback that satisfies the rollback specification.

[3]We remember that reversibility in reversible IPN is flexible.

[4]The rollback can be realized by either backtracking or flexible reversibility.

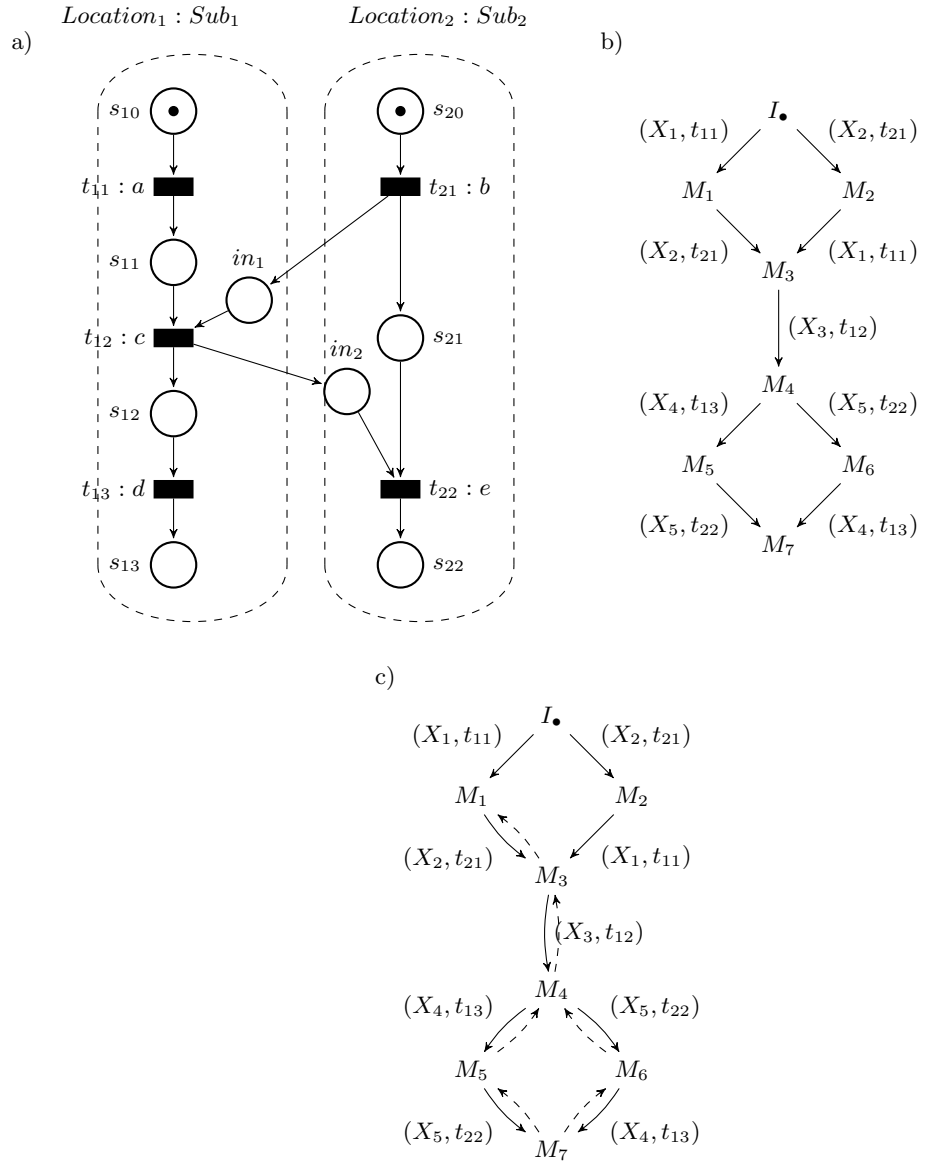

Figure 5. Controlling reversibility: a) distributed system N; b) marking graph of N; c) marking graph of $(N, (M_7, M_1))$

In the following and in the first, the formal definition of control causal-reversibility using an enumerate rollback is given and followed by a presentation of the control causal-reversibility approach using the rollback specification that is noted by states-based controlling causal-reversibility within IPNs (SCCR-IPNs).

To give the intuition behind the formal definition of the control causal-reversibility using an enumerate rollback, let us take the marking graph from Figure 6b, and let the given (M_5, M_1) be a rollback. To execute this rollback, we have two reversible equivalent sequences $\overline{(X_4,t_4)}.\overline{(X_3,t_3)}.\overline{(X_2,t_2)}$ and $\overline{(X_4,t_4)}.\overline{(X_2,t_2)}.\overline{(X_3,t_3)}$. Now, if we take the rollback (M_5, M_2) we also have two sequences $\sigma_1 = \overline{(X_4,t_4)}.\overline{(X_3,t_3)}$ and $\sigma_2 = \overline{(X_4,t_4)}.\overline{(X_2,t_2)}.\overline{(X_3,t_3)}.(X_2,t_2)$. However, the sequence σ_2 is more costly than the first, since both the undoing and the doing of (X_3,t_3) are not needed. It will be possible to control the rollback in order to get out of σ_2; the intuition of this is based on the definition of a partial order over the states (see Definition 5.2). According to this partial order, we only have the $M_2 \preceq M_4 \preceq M_5$ chain between M_5 and M_2. Therefore, the rollback enables the σ_1 sequence.

We show the following: (i) given a finite marking graph, the marking state set is structured as a domain (see Proposition 5.2) over partial order $\preceq$; (ii) the control backward firing is a continued function on this domain. This means that our control causal-reversibility is an optimal and finite calculus.

Definition 5.1. *Let $n, n' \in S_\bullet$ such that $n = (u,k,s)$, $n' = (u',k',s')$ and $u' = (X,t)$. Relationship $\leqslant$: $S_\bullet \longrightarrow S_\bullet$ is defined recursively as follows:*

$n \leqslant n'$ if and only if:

- *$n = n'$,*
- *or $n \in X$: means n directly gives rise to n',*
- *or $\exists n" \in X$ such that $n \leqslant n"$: means n indirectly gives rise to n'.*

Definition 5.2. *Let $M, M' \in S_\bullet$. Relationship $\preceq$: $S_\bullet \longrightarrow S_\bullet$ is defined as follows:*

$M \preceq M'$ if and only if $\forall n \in M$, then $\exists((X,t),k,s) \in M'$ such that:

- *$n = ((X,t),k,s)$: means same token,*
- *or $n \in X$: means n directly gives rise to $((X,t),k,s)$,*
- *or $\exists n' \in X$ such that $n \leqslant n'$: means n indirectly gives rise to $((X,t),k,s)$.*

Proposition 5.1. *Relationship $\preceq$ is a partial order.*

Proof 5.1. *Relationship $\preceq$ is reflexive, transitive, and anti-symmetric.*

1. *Reflexive: by definition.*
2. *Transitive: let $M_1 \preceq M_2$ and $M_2 \preceq M_3$, it will be easy to deduce that $M_1 \preceq M_3$. From the definition of $\preceq$, we can write:*
   - *$M_1 \preceq M_2$ if and only if $\forall n_1 \in M_1$ then $\exists((X_2,t_2),k_2,s_2) \in M_2$ such that $n_1 = ((X_2,t_2),k_2,s_2) \vee n_1 \in X_2 \vee \exists n_2 \in X_2 . n_1 \leqslant n_2$;*
   - *$M_2 \preceq M_3$ if and only if $\forall n_2 \in M_2$ then $\exists((X_3,t_3),k_3,s_3) \in M_3$ such that $n_2 = ((X_3,t_3),k_3,s_3) \vee n_2 \in X_3 \vee \exists n_3 \in X_3 . n_2 \leqslant n_3$.*

   From the fact that $(\forall a \in A . \exists b \in B)$ and $(\forall b \in B . \exists c \in C)$ implies $(\forall a \in A . \exists c \in C)$, we can deduce that $\forall n_1 \in M_1$ then $\exists((X_3,t_3),k_3,s_3) \in M_3$ such that $n_1 = ((X_3,t_3),k_3,s_3) \vee n_1 \in X_3 \vee \exists n_3 \in X_3 . n_1 \leqslant n_3$. So, we have $M_1 \preceq M_3$ as a result.
3. *Anti-symmetric: if $M_1 \preceq M_2$ and $M_2 \preceq M_1$, then $M_1 = M_2$, since the marking graph of IPN is acyclic.*

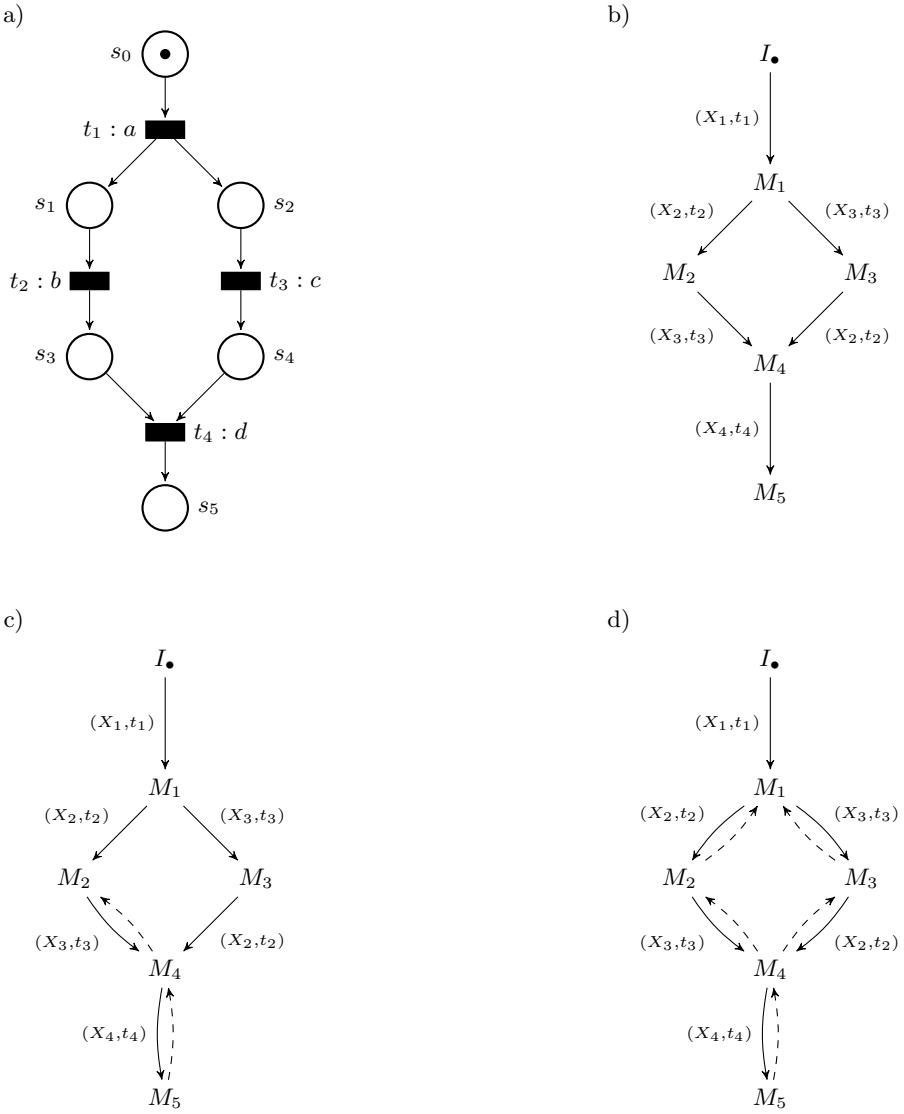

Figure 6. Control Causal-Reversibility using given enumerate rollback: a) reversible IPN N; b) marking graph of N; c) marking graph of $(N,(M_5,M_2))$; d) marking graph of $(N,(M_5,M_1))$

Proposition 5.2. $(S_\bullet,\preceq)$ *is a domain.*

Proof 5.2. $(S_\bullet,\preceq)$ *is a domain if it is a complete partial order CPO with down button* $\perp$*. Let* N *be an IPN, and let* Y *be a chain defined over* $S_\bullet$*; if the marking graph of* N *is a finite graph, then* Y *has a least upper bound (so, it is a CPO). Since* $\forall M \in S_\bullet : I_\bullet \preceq M$*, the down button is the initial marking state,* $\perp = I_\bullet$*.*

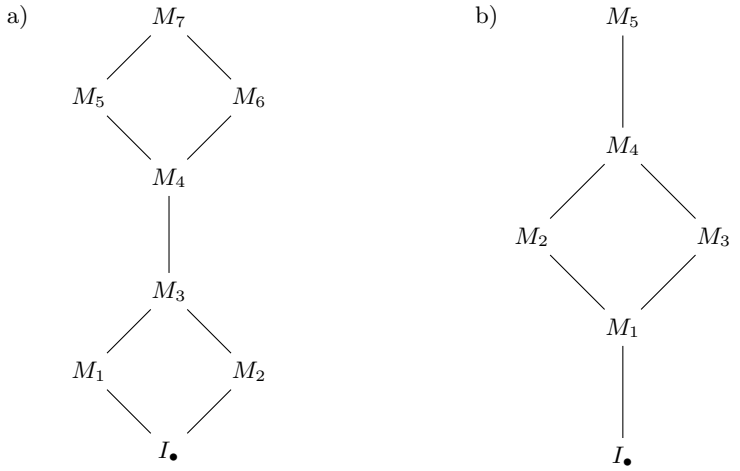

Figure 7. Domain of marking graph: a) domain of Figure 5a; b) domain of Figure 6a

Given a rollback (R, G), in the following, we define the control causal-reversibility within an IPN as a reversible IPN by modifying the backward firing rule (see Definition 5.3.2) with respect to chain $G \preceq .. \preceq R$, written as $(N, (R, G))$.

Definition 5.3. *Let (R, G) be a rollback. For a firing $u \in T_\bullet$ in a Petri net such that $u = (X, t)$, let ${}^\bullet u = X$ and $u^\bullet = \{(u, k, s) | K < F(\eta(u), s)\}$ be the set on input tokens and the set of output tokens of u.*

1. *Forward firing: transition t is enabled under an individual marking $M \in S_\bullet$ if ${}^\bullet u \subseteq M$. In this case, t can fire under M, yielding $M' = (M \setminus {}^\bullet u) \cup u^\bullet$, written as $M \overset{u}{\longrightarrow}_{\bullet f} M'$ or $M[u\rangle_f M'$.*
2. *Controlling backward firing: transition t can be undone under a rollback (R, G) and an individual marking $M' \in S_\bullet$ if $u^\bullet \subseteq M'$, $G \preceq M \preceq R$, and $G \prec M' \preceq R$. In this case, the undo of t can fire under M', yielding $M = (M' \setminus u^\bullet) \cup {}^\bullet u$. written as $M' \overset{u}{\longrightarrow}_{\bullet cb} M$ or $M'[u\rangle_{cb} M$.*

The controlling backward firing rule means that the undoing is possible if an only if both the source and the target of this firing are on chain $G \preceq ... \preceq R$.

For instance, the marking graph of $(N, (M_5, M_2))$ (respectively, $(N, (M_5, M_1))$) is presented as Figure 6c (respectively, Figure 6d). The marking graph of Figure 5c is from rollback (M_7, M_1).

Proposition 5.3. *Controlling backward firing $[..\rangle_{cb}$ is a continued function over $(S_\bullet, \preceq)$.*

Proof 5.3. *Let Y be a chain on $(S_\bullet, \preceq)$ such that $Y = M_1 \preceq M_2 \preceq .. \preceq M_n$. Function $[..\rangle_{cb}$ over rollback (M_n, M_1) is continued if and only if it is monotone and $\bigsqcup(Y) = \bigsqcup\{M' | \forall M \in Y. \exists u \in T_\bullet : M[u\rangle_{cb} M'\}$.*

- *Function* $[..\rangle_{cb}$ *is monotone: for all* $M_1 \preceq M_2$*, we have* $M_1[u\rangle_{cb}M_1'$ *and* $M_2[u\rangle_{cb}M_2'$*. From the definition of* $[..\rangle_{cb}$ *and* $\preceq$*, we have* $M_1' \preceq M_1$ *and* $M_2' \preceq M_2$*. From the fact that* $[..\rangle_{cb}$ *is a function, we have* $M_1 = M_2'$*. So,* $M_1' \preceq M_2'$*.*
- *Function* $[..\rangle_{cb}$ *is continuous: it is a direct consequence of the monotony of* $[..\rangle_{cb}$*. We recall that* $[..\rangle_{cb}$ *is defined over* (M_n, M_1)*; thus,* $\exists u_1 \in T_\bullet$ *such that* $M_2[u_1\rangle_{cb}M_1$*,* $\exists u_2 \in T_\bullet$ *such that* $M_3[u_2\rangle_{cb}M_2$*,...and* $\exists u_n \in T_\bullet$ *such that* $M_{n+1}[u_n\rangle_{cb}M_n$*. We get* $\bigsqcup\{M'|\forall M \in Y.\exists u \in T_\bullet : M[u\rangle_{cb}M'\} = \bigsqcup(\{M_1, M_2, ..., M_n\}) = \bigsqcup(Y)$ *as an outcome.*

In the above, it has been proven that (i) given a finite marking graph, the marking state set is structured as a domain over partial order $\preceq$ and (ii) the control backward firing is a continued function. In other words, this reversibility is a finite calculus, and it is optimal in the way that we cannot redo an action[5].

After given the formal definition of the control causal-reversibility using a given enumerate rollback, the formal definition of a rollback specification is introduced in the following to use in the states-based control reversible IPN definition.

Definition 5.4. *A rollback specification is a* (Φ, Ψ) *pair such that* Φ *and* Ψ *are predicates over* $S_\bullet$*. Let* $(M, M') \in S_\bullet \times S_\bullet$*, we saw that* (Φ, Ψ) *is satisfied by* (M, M') *if and only if* M *satisfies* Φ *and* M' *satisfies* Ψ*, written as* $(M, M') \vDash (\Phi, \Psi)$*.*

Definition 5.5. *The states-based control reversible IPN (noted as SCCR-INP) is a tuple* $(N, (\Phi, \Psi))$ *such that* N *is a Petri net* (S, T, F, I, L) *and* (Φ, Ψ) *is a rollback specification.*

Definition 5.6. *Let* (Φ, Ψ) *be a rollback specification. For a firing* $u \in T_\bullet$ *in a Petri net such that* $u = (X, t)$*, let* ${}^\bullet u = X$ *and* $u^\bullet = \{(u, k, s)|K < F(\eta(u), s)\}$ *be the set of input tokens and the set of output tokens of* u*.*

1. *Forward firing: transition* t *is enabled under an individual marking* $M \in S_\bullet$ *if* ${}^\bullet u \subseteq M$*. In this case,* t *can fire under* M*, yielding* $M' = (M \setminus {}^\bullet u) \cup u^\bullet$*, written as* $M \xrightarrow{u}_{\bullet f} M'$ *or* $M[u\rangle_f M'$*.*
2. *Controlling backward firing: transition* t *can be undone under* (Φ, Ψ) *and an individual marking* $M' \in S_\bullet$ *if* $u^\bullet \subseteq M'$ *and* (R, G) *exists such that* $(R, G) \vDash (\Phi, \Psi)$ *such that* $G \preceq M \preceq R$ *and* $G \prec M' \preceq R$*. In this case, the undo of* t *can fire under* M'*, yielding* $M = (M' \setminus u^\bullet) \cup^\bullet u$*, written as* $M' \xrightarrow{u}_{\bullet cb} M$ *or* $M'[u\rangle_{cb}M$*.*

6. Conclusions

This paper has explained causal reversibility in an individual token interpretation of Petri nets (IPNs). The evidence from this study intimates that causal reversibility in a given IPN assures both its coherence and flexible reversibility; furthermore, its initial state can be accessible in reverse from any state.

[5]This calculus follows a chain of the domain.

In the distributed system context, it will be difficile to control causal-reversibility without being given the global behavior of a system. In the present work, we found that Van Glabbeek's representation of individual token interpretation [24] provides a powerful tool for define a controlling causal-reversibility by giving an implicit rollback that describes the state that, from it, we go back to a consistent one it is concretized by the proposition of the states-based control causal-reversible IPN (SCCR--IPN) definition. Improving the rollback specification language, this model has the potential to be used in different contexts such as biological, chemical, debugging, transaction systems, and in state space exploration problems.

References

[1] Altenkirch T., Grattage J.: A functional quantum programming language. In: *20th IEEE Symposium on Logic in Computer Science (LICS 2005), 26–29 June 2005, Chicago, IL, USA, Proceedings*, pp. 249–258, 2005. https://doi.org/10.1109/LICS.2005.1.

[2] Barylska K., Erofeev E., Koutny M., Mikulski L., Piątkowski M.: Reversing Transitions in Bounded Petri Nets, *Fundamenta Informaticae*, vol. 157(4), pp. 341–357, 2018, https://doi.org/10.3233/FI-2018-1631.

[3] Barylska K., Koutny M., Mikulski L., Piątkowski M.: Reversible Computation vs. Reversibility in Petri Nets. In: Devitt S., Lanese I. (eds.), *Reversible Computation. RC 2016*, Lecture Notes in Computer Science, vol. 9720, pp. 105–118, Springer, Cham, 2016. https://doi.org/10.1007/978-3-319-40578-0_7.

[4] Barylska K., Koutny M., Mikulski L., Piątkowski M.: Reversible computation vs. reversibility in Petri nets, *Science of Computer Programming*, vol. 151, pp. 48–60, 2018. https://doi.org/10.1016/j.scico.2017.10.008.

[5] Barylska K., Mikulski L., Piątkowski M., Koutny M., Erofeev E.: Reversing Transitions in Bounded Petri Nets. In: *Proceedings of the 25th International Workshop on Concurrency, Specification and Programming, Rostock, Germany, September 28–30, 2016*, pp. 74–85, 2016. http://ceur-ws.org/Vol-1698/CS&P2016_08_Barylska&Mikulski&Piatkowski&Koutny&Erofeev_Reversing-Transitions-in-Bounded-Petri-Nets.pdf.

[6] Bednarczyk M.A.: Hereditary history preserving bisimulations or what is the power of the future perfect in program logics. Technical report, Institute of Computer Science, Polish Academy of Sciences, Gdansk, 1991.

[7] Best E., Devillers R.R.: Sequential and concurrent behaviour in Petri net theory, *Theoretical Computer Science*, vol. 55(1), pp. 87–136, 1987, https://doi.org/10.1016/0304-3975(87)90090-9.

[8] Best E., Devillers R.R., Kiehn A., Pomello L.: Concurrent bisimulations in Petri nets, *Acta Informatica*, vol. 28(3), pp. 231–264, 1991, https://doi.org/10.1007/BF01178506.

[9] Cardelli L., Laneve C.: Reversible structures. In: *CMSB'11: Proceedings of the 9th International Conference on Computational Methods in Systems Biology*, pp. 131–140, 2011. https://doi.org/10.1145/2037509.2037529.

[10] Clairambault P., Visme de M., Winskel G.: Concurrent Quantum Strategies. In: Thomsen M., Soeken M. (eds.), *Reversible Computation. RC 2019*, Lecture Notes in Computer Science, vol. 11497, pp. 3–19, Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-21500-2_1.

[11] Clavel M., Durán F., Eker S., Lincoln P., Martí-Oliet N., Meseguer J., Quesada J.F.: Maude: specification and programming in rewriting logic, *Theoretical Computer Science*, vol. 285(2), pp. 187–243, 2002, https://doi.org/10.1016/S0304-3975(01)00359-0.

[12] Cook J.J.: Reverse Execution of Java Bytecode, *The Computer Journal*, vol. 45(6), pp. 608–619, 2002. https://doi.org/10.1093/comjnl/45.6.608.

[13] Danos V., Krivine J.: Reversible Communicating Systems. In: Gardner P., Yoshida N. (eds.), *CONCUR 2004 – Concurrency Theory. CONCUR 2004*, Lecture Notes in Computer Science, vol. 3170, pp. 292–307, Springer, Berlin–Heidelberg, 2004. https://doi.org/10.1007/978-3-540-28644-8_19.

[14] Danos V., Krivine J.: Transactions in RCCS. In: Abadi M., de Alfaro L. (eds.), *CONCUR 2005 – Concurrency Theory. CONCUR 2005*, Lecture Notes in Computer Science, vol. 3653, pp. 398–412, Springer, Berlin–Heidelberg, 2005. https://doi.org/10.1007/11539452_31.

[15] Danos V., Krivine J.: Formal Molecular Biology Done in CCS-R, *Electronic Notes in Theoretical Computer Science*, vol. 180(3), pp. 31–49, 2007, https://doi.org/10.1016/j.entcs.2004.01.040.

[16] Danos V., Krivine J., Sobociński P.: General Reversibility, *Electronic Notes in Theoretical Computer Science*, vol. 175(3), pp. 75–86, 2007, https://doi.org/10.1016/j.entcs.2006.07.036.

[17] Danos V., Krivine J., Tarissan F.: Self-assembling Trees, *Electronic Notes in Theoretical Computer Science*, vol. 175(1), pp. 19–32, 2007, https://doi.org/10.1016/j.entcs.2006.11.017.

[18] Engelfriet J.: Branching processes of Petri nets, *Acta Informatica*, vol. 28(6), pp. 575–591, 1991, https://doi.org/10.1007/BF01463946.

[19] Fecher H.: A completed hierarchy of true concurrent equivalences, *Information Processing Letters*, vol. 89(5), pp. 261–265, 2004, https://doi.org/10.1016/j.ipl.2003.11.008.

[20] Feldman S.I., Brown C.B.: IGOR: A system for program debugging via reversible execution, *ACM SIGPLAN Notices*, vol. 24(1), pp. 112–123, 1988. https://doi.org/10.1145/68210.69226.

[21] Frank M.P.: Physical Foundations of Landauer's Principle. In: Kari J., Ulidowski I. (eds.), *Reversible Computation. RC 2018*, Lecture Notes in Computer Science, vol. 11106, pp. 3–33, Springer, Cham, 2018. https://doi.org/10.1007/978-3-319-99498-7_1.

[22] Glabbeek van R.J.: The Linear Time – Branching Time Spectrum I. In: *Handbook of Process Algebra*, pp. 3–99, 2001. https://doi.org/10.1016/b978-044482830-9/50019-9.

[23] Glabbeek van R.J.: The Individual and Collective Token Interpretations of Petri Nets. In: Abadi M., de Alfaro L. (eds.), *CONCUR 2005 – Concurrency Theory*, Lecture Notes in Computer Science, vol. 3653, pp. 323–337, Springer, Berlin–Heidelberg, 2005. https://doi.org/10.1007/11539452_26.

[24] Glabbeek van R.J., Goltz U., Schicke-Uffmann J.: On Distributability of Petri Nets. In: Birkedal L. (ed.), *Foundations of Software Science and Computational Structures. FoSSaCS 2012*, Lecture Notes in Computer Science, vol. 7213, pp. 331–345, Springer, Berlin–Heidelberg, 2012. https://doi.org/10.1007/978-3-642-28729-9_22.

[25] Goltz U., Reisig W.: The non-sequential behaviour of Petri nets, *Information and Control*, vol. 57(2–3), pp. 125–147, 1983. https://doi.org/10.1016/S0019-9958(83)80040-0.

[26] Hoey J., Ulidowski I.: Reversible Imperative Parallel Programs and Debugging. In: Thomsen M., Soeken M. (eds.), *Reversible Computation. RC 2019*, Lecture Notes in Computer Science, vol. 11497, pp. 108–127, Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-21500-2_7.

[27] Krivine J.: A Verification Technique for Reversible Process Algebra. In: *Reversible Computation, 4th International Workshop, RC 2012, Copenhagen, Denmark, July 2-3, 2012. Revised Papers*, pp. 204–217, 2012. https://doi.org/10.1007/978-3-642-36315-3_17.

[28] Landauer R.: Irreversibility and Heat Generation in the Computing Process, *IBM Journal of Research and Development*, vol. 5(3), pp. 183–191, 1961, https://doi.org/10.1147/rd.53.0183.

[29] Lanese I., Lienhardt M., Mezzina C.A., Schmitt A., Stefani J.-B.: Concurrent Flexible Reversibility. In: Felleisen M., Gardner P. (eds.), *Programming Languages and Systems. ESOP 2013*. Lecture Notes in Computer Science, vol. 7792, pp. 370–390, Springer, Berlin, Heidelberg, 2013. https://doi.org/10.1007/978-3-642-37036-6_21.

[30] Lanese I., Mezzina C.A., Schmitt A., Stefani J.-B.: Controlling Reversibility in Higher-Order Pi. In: Katoen J.P., König B. (eds.), *CONCUR 2011 – Concurrency Theory*, Lecture Notes in Computer Science, vol. 6901, pp. 297–311, Springer, Berlin–Heidelberg, 2011. https://doi.org/10.1007/978-3-642-23217-6_20.

[31] Lanese I., Mezzina C.A., Stefani J.-B.: Controlled Reversibility and Compensations. In: Glück R., Yokoyama T. (eds.), *Reversible Computation. RC 2012*, Lecture Notes in Computer Science, vol. 7581, pp. 233–240, Springer, Berlin–Heidelberg, 2012. https://doi.org/10.1007/978-3-642-36315-3_19.

[32] Lanese I., Nishida N., Palacios A., Vidal G.: CauDEr: A Causal-Consistent Reversible Debugger for Erlang. In: Gallagher J., Sulzmann M. (eds.), *Functional and Logic Programming. FLOPS 2018*, Lecture Notes in Computer Science, vol. 10818, pp. 247–263, Springer, Cham, 2018. https://doi.org/10.1007/978-3-319-90686-7_16.

[33] Lanese I., Palacios A., Vidal G.: Causal-Consistent Replay Debugging for Message Passing Programs. In: Pérez J., Yoshida N. (eds.), *Formal Techniques for Distributed Objects, Components, and Systems. FORTE 2019*, Lecture Notes in Computer Science, vol. 11535, pp. 167–184, Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-21759-4_10.

[34] Leeman G.B.: A formal approach to undo operations in programming languages, *ACM Transactions on Programming Languages and Systems*, vol. 8(1), pp. 50–87, 1986, https://doi.org/10.1145/5001.5005.

[35] Mazurkiewicz A.W.: Trace theory. In: Brauer W., Reisig W., Rozenberg G. (eds.), *Petri Nets: Applications and Relationships to Other Models of Concurrency. ACPN 1986*, Lecture Notes in Computer Science, vol. 255, pp. 279–324, Springer, Berlin–Heidelberg, 1986. https://doi.org/10.1007/3-540-17906-2_30.

[36] Mazurkiewicz A.W.: Basic notions of trace theory. In: de Bakker J.W., de Roever W.P., Rozenberg G. (eds.), *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency. REX 1988*, Lecture Notes in Computer Science, vol. 354, pp. 285–363, Springer, Berlin–Heidelberg, 1988. https://doi.org/10.1007/BFb0013025.

[37] Philippou A., Psara K., Siljak H.: Controlling Reversibility in Reversing Petri Nets with Application to Wireless Communications, *CoRR*, vol. abs/1905.11958, 2019. http://arxiv.org/abs/1905.11958.

[38] Phillips I., Ulidowski I.: A Logic with Reverse Modalities for History-preserving Bisimulations. In: *Proceedings 18th International Workshop on Expressiveness in Concurrency, EXPRESS 2011, Aachen, Germany, 5th September 2011*, pp. 104–118, 2011. https://doi.org/10.4204/EPTCS.64.8.

[39] Phillips I., Ulidowski I.: Reversibility and asymmetric conflict in event structures, *The Journal of Logic and Algebraic Programming*, vol. 84(6), pp. 781–805, 2015. https://doi.org/10.1016/j.jlamp.2015.07.004.

[40] Phillips I., Ulidowski I., Yuen S.: A Reversible Process Calculus and the Modelling of the ERK Signalling Pathway. In: Glück R., Yokoyama T. (eds.), *Reversible Computation. RC 2012*, Lecture Notes in Computer Science, vol. 7581, pp. 218–232, Springer, Berlin–Heidelberg, 2012. https://doi.org/10.1007/978-3-642-36315-3_18.

[41] Phillips I., Ulidowski I.: Reversing algebraic process calculi, *The Journal of Logic and Algebraic Programming*, vol. 73(1–2), pp. 70–96, 2007. https://doi.org/10.1016/j.jlap.2006.11.002.

[42] Soloveichik D., Seelig G., Winfree E.: DNA as a universal substrate for chemical kinetics. In: Goel A., Simmel F.C., Sosík P. (eds.), *DNA Computing. DNA 2008*, Lecture Notes in Computer Science, vol. 5347, pp. 57–69, Springer, Berlin–Heidelberg, pp. 57–69, 2008. https://doi.org/10.1007/978-3-642-03076-5_6.

[43] Ulidowski I., Phillips I., Yuen S.: Concurrency and Reversibility. In: Yamashita S., Minato S. (eds.), *Reversible Computation. RC 2014*, Lecture Notes in Computer Science, vol. 8507, pp. 1–14, Springer, Cham, 2014. https://doi.org/10.1007/978-3-319-08494-7_1.

[44] Zelkowitz M.V.: Reversible Execution, *Communications of the ACM*, vol. 16(9), p. 566, 1973. https://doi.org/10.1145/362342.362360.

Affiliations

Adel Benamira
8 May 1945 Guelma University, Computer Science Department, BP-401, 24000, Guelma, Algeria, benamira.adel@univ-guelma.dz

Received: 15.03.2020
Revised: 17.05.2020
Accepted: 17.05.2020