Token Multiplicity in Reversing Petri Nets Under the Individual Token Interpretation

Anna Philippou Kyriaki Psara
Department of Computer Science, University of Cyprus
Nicosia, Cyprus
annap@ucy.ac.cy kpsara01@ucy.ac.cy

Reversing Petri nets (RPNs) have recently been proposed as a net-based approach to model causal and out-of-causal order reversibility. They are based on the notion of individual tokens that can be connected together via bonds. In this paper we extend RPNs by allowing multiple tokens of the same type to exist within a net based on the individual token interpretation of Petri nets. According to this interpretation, tokens of the same type are distinguished via their causal path. We develop a causal semantics of the model and we prove that the expressive power of RPNs with multiple tokens is equivalent to that of RPNs with single tokens by establishing an isomporphism between the Labelled Transition Systems (LTSs) capturing the reachable parts of the respective RPN models.

1 Introduction

Reversible computation is a form of computing where transitions can be executed in both the forward and the reverse direction, allowing systems to return to past states. It has been attracting increasing attention due to its application in a variety of fields such as low-power computing, biological modelling, quantum computation, robotics, and distributed systems.

In the sequential setting reversibility is generally understood as the ability to execute past actions in the exact inverse order in which they occurred, a process referred to as *backtracking*. However, in the concurrent setting matters are less clear. Indeed, various approaches have been investigated within a variety of formalisms [8, 26, 16, 33, 24, 20]. One of the most well-studied approaches considered suitable for a wide variety of concurrent systems is that of *causal-consistent reversibility* advocating that a transition can be undone only if all its effects, if any, have been undone beforehand [7]. The study of reversibility also extends to *out-of-causal-order reversibility*, a form of reversing where executed actions can be reversed in an out-of-causal order [28, 27, 15] most notably featured in biochemical systems.

In this work, we focus on Reversing Petri Nets [24] (RPNs), a reversible model inspired by Petri nets that allows the modelling of reversibility as realised by backtracking, causal-order, and out-of-causal-order reversing. A key challenge when reversing computations in Petri nets is handling *backward conflicts*. These conflicts arise when tokens occur in a certain place due to different causes making unclear which transitions ought to be reversed. To handle this ambiguity, RPNs introduce the notion of a *history* of transitions, which records causal information of executions. Furthermore, inspired by biochemical systems as well as other resource-aware applications, the model employs named tokens that can be connected together to form bonds, and are preserved during execution.

A restriction in RPNs is that each token is unique and in order to model a system with multiple items of the same type, it is necessary to employ a distinct token for each item, at the expense of the net size. In the current paper we consider an extension of RPNs, which allows multiple tokens of the same type. The introduction of multiple identical tokens creates further challenges involving backward conflicts and requires to extend the RPN machinery for extracting the causal dependencies between transitions.

V. Castiglioni and C. A. Mezzina (Eds): Combined Workshop on Expressiveness in Concurrency and Structural Operational Semantics 2022 (EXPRESS/SOS 2022).
EPTCS 368, 2022, pp. 131–150, doi:10.4204/EPTCS.368.8

We note that formalizing causal dependencies is a well-studied problem in the context of Petri nets, where various approaches have been proposed to reason about causality [11, 13, 32]. In this work we draw inspiration from the so-called individual token and collective interpretations of Petri nets [12, 10]. The collective token philosophy considers all tokens of a certain type to be identical, which results in ambiguities when it comes to causal dependencies. In contrast, in an individual token interpretation, tokens are distinguished based on their causal path. This approach leads to more complicated semantics since to achieve token individuality requires precise correspondence between the token instances and their past. However, it enables backward determinism, which is a crucial property of reversible systems.

Contribution. In this paper we extend RPNs to support multiple tokens of the same type following the individual token interpretation. As such, tokens are associated with their causal history and, while tokens of the same type are equally eligible to fire a transition when going forward, when going backwards they are able to reverse only the transitions they have previously fired. In this context, we define a causal semantics for the model, based on the intuition that a causal link exists between two transitions if a token produced by one was used to fire the other. This leads to the observation that a transition may reverse in causal order only if it was the last transition executed by all the tokens it has involved. We note that this approach allows a causal-order reversible semantics that, unlike the original RPN model, does not require any global history information. In fact, all information necessary for reversal is available locally within the history of tokens. Subsequently, we turn to study the expressiveness of the presented model in comparison to RPNs with single tokens. To do this we employ Labelled Transition Systems (LTSs) capturing the state space of RPN models. We show that for any RPN with multiple tokens there exists an RPN with single tokens with an isomorphic LTS, thereby confirming our conjecture that RPNs with single tokens are as expressive as RPNs with multiple tokens.

Related Work. The first study of reversible computation within Petri nets was proposed in [4, 5], where the authors investigated the effects of adding *reversed* versions of selected transitions by reversing the directions of a transition's arcs. Unfortunately, this approach to reversibility violates causality. Towards examining causal consistent reversibility the work in [21] investigates whether it is possible to add a complete set of effect-reverses for a given transition without changing the set of reachable markings, showing that this problem is in general undecidable. In another line of work [20] propose a causal semantics for P/T nets by identifying the causalities and conflicts of a P/T net through unfolding it into an equivalent occurrence net and subsequently introducing appropriate reverse transitions to create a coloured Petri net (CPN) that captures a causal-consistent reversible semantics. On a similar note, [19] introduces the notion of reversible occurrence nets and associates a reversible occurrence net to a causal reversible prime event structure, and vice versa. Finally, [6] introduces a reversible approach to Petri nets following the individual token interpretation. This work is similar to our approach though it refers to a basic PN model, which does not contain named tokens nor bonds, and it does not support backtracking and out-of-causal reversibility.

The modelling of bonding in the context of reversibility was first considered within reversible processes and event structures in [29], where its usefulness was illustrated with examples taken from software engineering and biochemistry. Reversible frameworks that feature bonds as first-class entities, like RPNs, also include the Calculus of Covalent Bonding [15], which supports causal and out-of-causal-order reversibility in the context of chemical reactions, as well as the Bonding Calculus [1], a calculus developed for modeling covalent bonds between molecules in biochemical systems. In fact, the latter two frameworks and RPNs were reviewed and compared for modeling chemical reactions in [14] with

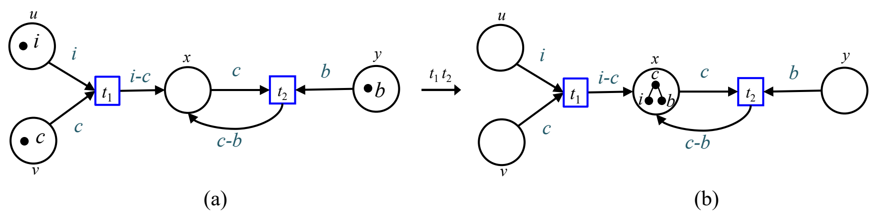


Figure 1: RPN example of a pen assembly/dissassembly

case study the autoprotolysis of water.

This paper extends a line of research on reversing Petri nets, initially introduced for acyclic nets [22] and subsequently for nets with cycles [24]. The usefulness of the framework was illustrated in a number of examples including the modelling of long-running transactions with compensation and a signal-passing mechanism used by the ERK pathway. The RPN framework has been extended to control reversibility in [25] with an application to Massive MIMO. Introducing multiple tokens in RPNs was also examined in [23] by allowing multiple tokens of the same type to exist within a net following the collective interpretation and yielding a locally-controlled, out-of-causal-order reversibility semantics. RPNs have been translated to Answer Set Programming (ASP), a declarative programming framework with competitive solvers [9], and to bounded Coloured Petri Nets [2, 3].

2 Reversing Petri Nets with Multiple tokens

In our previous works we introduced Reversing Petri Nets, a net-based formalism, which features individual tokens that can be connected together via bonds [24]. An assumption of RPNs is that tokens are pairwise distinct. To relax this restriction, subsequent work [23] introduced token multiplicity whereby a model may contain multiple tokens of the same type. It was observed that the possibility of firing a transition multiple times using different sets of tokens, may introduce nondeterminism, also known as backward conflict, when going backwards. Furthermore, two approaches were identified to define reversible semantics in the presence of such backward conflicts, inspired by the individual token and the collective token interpretations [10, 12], defined to reason about causality in Petri nets. In the individual token approach, multiple tokens of the same type residing in the same place are distinguished based on their causal path, whereas in the collective token interpretation they are not distinguished. In [23] the model of RPNs with multiple tokens was investigated under the collective token approach, yielding an out-of-causal-order form of reversibility. In this work, we instead apply the individual token interpretation to define a causal semantics, and we establish that in fact the addition of multiple tokens does not add to the expressiveness of the model, in that for any RPN with multiple tokens there exists an equivalent RPN with only a single token of each type.

To appreciate the challenges induced through the introduction of multiple tokens and the difference between the individual and the collective token interpretations, let us consider the example in Fig. 1(a). In this example we may see an RPN model of an assembly/disassembly of a pen. The product consists of the ink, the cup, and the button of the pen, modelled by tokens i, c, and b, respectively. We may observe that transitions, in addition to transferring tokens between places, have the capacity of creating bonds. Thus, the process of manufacturing the pen requires the ink to be fitted inside the cup, modelled by the creation of the bond $i-c$ by transition t_1 and, subsequently, the fitting of the button on the cup to complete

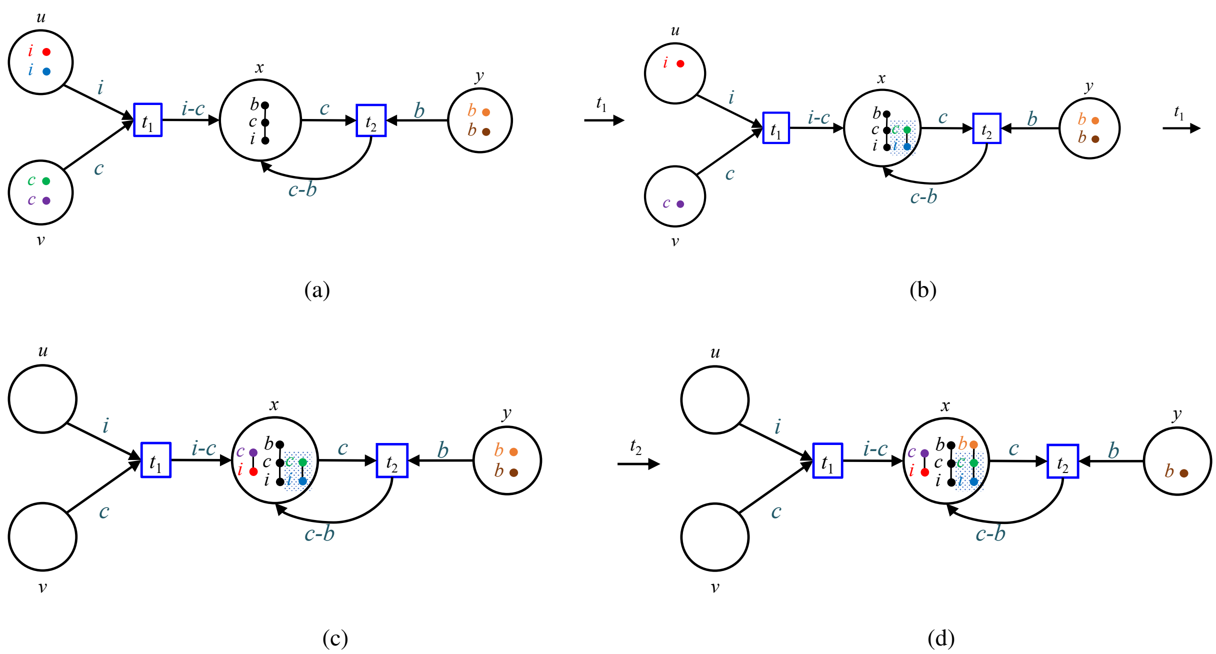


Figure 2: Executing transition t_1 in the net (a) may yield the net in (b). Different selections of tokens could have been made. In net (b) transition t_1 is executed with the (only) available tokens leading to net (c), whereby execution of t_2 with the component produced by the first execution of t_1 yields net (d).

the assembly, modelled as the creation of the bond $c-b$ by transition t_2 (RPN in Fig. 1(b)). The effect of reversing a transition in RPNs is to break the bonds created by the transition (if any) and returning the tokens/bonds from the outgoing places to the incoming places of the transition. In [22, 24] machinery has been developed in order to model backtracking, causal, and out-of-causal-order reversibility for the model. In particular, in the example of Fig. 1(b) reversing transition t_2 will result in the destruction of bond $c-b$ and the return of token b to place y.

Suppose we wish to extend the model of Fig. 1(a) for the assembly of two pens. Given that in RPNs tokens are unique, it would be necessary to introduce three new and distinct tokens and clone the transitions while renaming their arcs to accommodate for the names of the new tokens to be employed, resulting in a considerable expansion of the model for each new pen to be produced. Thus, a natural extension of the formalism involves relaxing this restriction and allowing multiple tokens of the same type to exist within a model. To this effect consider the scenario of Fig. 2(a) presenting a system with an already assembled/sample pen in place x and two items of each of the ink, cup, and button components.

An issue arising in this new setting is that due to the presence of multiple tokens of the same type, the phenomenon of backwards nondeterminism occurs when transitions are reversed. For instance, after execution of transition t_1 twice and t_2, two assembled pens will exist in place x and well as a component $i-c$, as seen in Fig. 2(d). Suppose that in this state transition t_1 is reversed. In the collective token interpretation, all instances of the bond $i-c$ are considered identical. As a result, any of these bonds could be destroyed during the reversal of transition t_1. However, in the individual token interpretation the various ink and cup tokens are distinguished based on their causal path. Therefore, the first execution of transition t_1 yielding the net in Fig. 2(b) and involving the shaded component of tokens in the figure, is considered to have caused the execution of transition t_2. Given this causal relationships between the transitions, under a causal reversibility semantics, the specific $i-c$ component should not be decomposed until transition t_2 is reversed. Similarly, the pre-existing pen should not be broken down into its parts as it was not the created by any of the transitions. Instead, reversing transition t_1 in the RPN of Fig. 2(d) should break the bond in the component consisting the single bond $i-c$. Note that this is compatible with the understanding that disassembly of the product would not allow the separation of the ink from the inside of the cup before the button is removed, since this is enclosed within the pair of the cup and the button.

As a result we observe that following the individual token interpretation, reversing a computation requires keeping track of past behavior – in the context of the example, distinguishing the tokens involving the pre-existing pen and the tokens used to fire each transition. In the following sections we implement this approach for introducing multiple tokens and we study its properties in the context of causal-order reversibility. Furthermore, we establish a correspondence between this model and RPNs with single tokens.

3 Multi Reversing Petri Nets

We present multi reversing Petri nets, an extension of RPNs with multiple tokens of the same type that allow transitions to be reversed following the individual token interpretation. Formally, they are defined as follows:

Definition 1 A *multi reversing Petri net* (MRPN) is a tuple $(P, T, \mathscr{A}, \mathscr{A}_V, \mathscr{B}, F)$ where:

1. P is a finite set of *places* and T is a finite set of *transitions*.
2. $\mathscr{A}$ is a finite set of *base* or *token types* ranged over by $A, B, \ldots$
3. $\mathscr{A}_V$ is a finite set of *token variables* ranged over by $a, b, \ldots$ We write $\mathsf{type}(a)$ for the type of variable a and assume that $\mathsf{type}(a) \in \mathscr{A}$ for all $a \in \mathscr{A}_V$.
4. $\mathscr{B} \subseteq \mathscr{A} \times \mathscr{A}$ is a finite set of undirected *bond types* ranged over by $\beta, \gamma, \ldots$ We assume $\mathscr{B}$ to be a symmetric relation and we consider the elements (A, B) and (B, A) to refer to the same bond type, which we also denote by $A-B$. Furthermore, we write $\mathscr{B}_V \subseteq \mathscr{A}_V \times \mathscr{A}_V$, assuming that (a, b) and (b, a) represent the same bond, also denoted as $a-b$.
5. $F : (P \times T \cup T \times P) \to \mathscr{P}(\mathscr{A}_V \cup \mathscr{B}_V)$ defines a set of directed labelled *arcs* each associated with a subset of $\mathscr{A}_V \cup \mathscr{B}_V$, where $(a, b) \in F(x, y)$ implies that $a, b \in F(x, y)$. Moreover, for all $t \in T$, $x, y \in P$, $x \neq y$, $F(x, t) \cap F(y, t) = \emptyset$.

A multi reversing Petri net is built on the basis of a set of *token types*. Multiple occurrences of a token type, referred to as *token instances*, may exist in a net. Tokens of the same type have identical capabilities on firing transitions and can participate only in transitions with variables of the same type.

As standard in net-based frameworks, places and transitions are connected via labelled directed arcs. These labels are derived from $\mathscr{A}_V \cup \mathscr{B}_V$. They express the requirements and the effects of transitions based on the type of tokens consumed. Thus, collections of tokens corresponding to the same types and connections as the variables on the labelled arc are able to participate in the transition. More precisely, if $F(x, t) = X \cup Y$, where $X \subseteq \mathscr{A}_V$, $Y \subseteq \mathscr{B}_V$, the firing of t requires a distinct token instance of type $\mathsf{type}(a)$ for each $a \in X$, such that the overall selection of tokens are connected together satisfying the restrictions posed by Y. Similarly, if $F(t, x) = X \cup Y$, where $X \subseteq \mathscr{A}_V$, $Y \subseteq \mathscr{B}_V$, this implies that during the forward execution of the transition for each $a \in X$ a token instance of type $\mathsf{type}(a)$ will be transmitted to place x by the transition, in addition to the bonds specified by Y, some of which will be created as an effect of the transition. We make the assumption that if $(a,b) \in Y$ then $a,b \in X$ and the same variable cannot be used on two incoming arcs of a transition.

We introduce the following notations. We write $\circ t = \{x \in P \mid F(x,t) \neq \emptyset\}$ and $t\circ = \{x \in P \mid F(t,x) \neq \emptyset\}$ for the incoming and outgoing places of transition t, respectively. Furthermore, we write $\mathsf{pre}(t) = \bigcup_{x\in P} F(x,t)$ for the union of all labels on the incoming arcs of transition t, and $\mathsf{post}(t) = \bigcup_{x\in P} F(t,x)$ for the union of all labels on the outgoing arcs of transition t.

We restrict our attention to well-formed MRPNs, which satisfy the conservation property [18] in the sense that the number of tokens in a net remains constant during execution. In fact, as we will prove in the sequel, in well-formed nets individual tokens are conserved.

Definition 2 An MRPN $(P,T,\mathscr{A},\mathscr{A}_V,\mathscr{B},F)$ is *well-formed* if for all $t \in T$:

1. $\mathscr{A}_V \cap \mathsf{pre}(t) = \mathscr{A}_V \cap \mathsf{post}(t)$ and
2. $F(t,x) \cap F(t,y) = \emptyset$ for all $x,y \in P$, $x \neq y$.

Thus, a well-formed MRPN satisfies (1) whenever a variable exists in the incoming arcs of a transition then it also exists on its outgoing arcs, and vice versa, which implies that transitions neither create nor erase tokens, and (2) tokens/bonds cannot be cloned into more than one outgoing place.

In the context of token multiplicity, a mechanism is needed in order to distinguish between token instances with respect to their causal path. For instance, consider the MRPN in Fig. 2(d). In this state, three connected components of tokens are positioned in place x, where tokens of the same type, e.g. the three c tokens have distinct connections and causal histories. To capture this, we distinguish between token instances, as follows:

Definition 3 Given an MRPN $(P,T,\mathscr{A},\mathscr{A}_V,\mathscr{B},F)$ a *token instance* has the form (A,i,xs) where xs is a (possibly empty) list of triples $[(k_1,t_1,v_1),\ldots,(k_n,t_n,v_n)]$ with $n \geq 0$, where $i \geq 1$, $A \in \mathscr{A}$, and for all i, $k_i \in \mathbb{N}$, $t_i \in T$, and $v_i \in \{*\} \cup \mathscr{A}_V$. We write $\mathscr{A}_I$ for the set of token instances ranged over by $A_1, A_2, \ldots$, and we define the set of bond instances $\mathscr{B}_I$ by $\mathscr{B}_I = \mathscr{A}_I \times \mathscr{A}_I$. Furthermore, given $A_i = (A,i,[(k_1,t_1,v_1),\ldots,(k_n,t_n,v_n)])$, we write

$$
\begin{array}{rcl}
\mathsf{type}(A_i) & = & A \\
A_i\!\downarrow & = & (A,i) \\
\mathsf{cpath}A_i & = & [(t_1,v_1),\ldots,(t_n,v_n)] \\
\mathsf{last}(A_i) & = & (k_n,t_n,v_n) \\
A_i + (k,t,v) & = & (A,i,[(k_1,t_1,v_1),\ldots,(k_n,t_n,v_n),(k,t,v)]) \\
\mathsf{init}(A_i) & = & (A,i,[(k_1,t_1,v_1),\ldots,(k_{n-1},t_{n-1},v_{n-1})])
\end{array}
$$

The set of token instances $\mathscr{A}_I$ corresponds to the basic entities that occur in a system. In the initial state of a net, tokens have the form $(A,i,[])$ where i is a unique identifier for the specific token instance of type A. As computation proceeds the tokens evolve to capture their causal path. If a transition t is executed in the forward direction, with some token instance $(A,i,[(k_1,t_1,v_1),\ldots,(k_n,t_n,v_n)])$ substituted for a variable v, then the token evolves to $(A,i,[(k_1,t_1,v_1),\ldots,(k_n,t_n,v_n),(k,t,v)])$, where k is an integer that characterizes the executed transition, as we will formally define in the sequel.

In a graphical representation, tokens instances are indicated by $\bullet$ associated with their description, places by circles, transitions by boxes, and bonds by lines between tokens. Note that token variables $a \in F(x,t) \cap \mathscr{A}_V$ with $\mathsf{type}(a) = A$ are denoted by $a : A$ over the corresponding arc $F(x,t)$. An example of an MRPN can be seen in Fig. 3. In this example, we have $\mathscr{A} = \{I,C,B\}$, $\mathscr{A}_V = \{i,c,b\}$, and the set of token instances in the specific state are $\{(I,i,[]),(B,i,[]),(C,i,[]) \mid i \in \{1,2,3\}\}$.

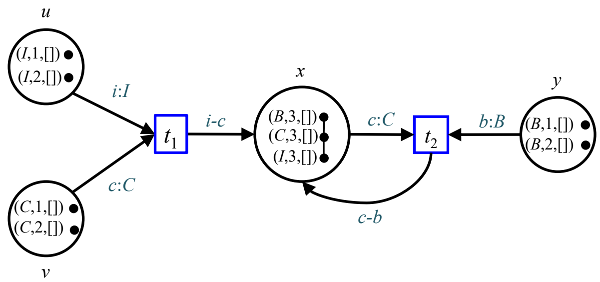


Figure 3: The net of Fig. 2(a) presented as an MRPN.

As with RPNs the association of token/bond instances to places is called a *marking* such that $M : P \rightarrow 2^{\mathscr{A}_I \cup \mathscr{B}_I}$, where we assume that if $(A_i,B_i) \in M(x)$ then $A_i,B_i \in M(x)$. In addition, we employ the notion of a *history*, which assigns a memory to each transition $H : T \rightarrow 2^{\mathbb{N}}$. Intuitively, a history of $H(t) = \emptyset$ for some $t \in T$ captures that the transition has not taken place, or every execution of it has been reversed, and a history such that $k \in H(t)$, captures that the transition had a firing with identifier k that was not reversed. Note that $|H(t)| > 1$ may arise due to cycles but also due to the consecutive execution of the transition by different token instances. A pair of a marking and a history, $\langle M,H\rangle$, describes a *state* of an MRPN with $\langle M_0,H_0\rangle$ the initial state, where $H_0(t) = \emptyset$ for all $t \in T$ and if $A_i \in M_0(x)$, $x \in P$, then $A_i = (A,i,[])$, and $A_i \in M_0(y)$ implies that $x = y$.

Finally, we define $\mathsf{con}(A_i,W)$, where $A_i \in \mathscr{A}_I$ and $W \subseteq \mathscr{A}_I \cup \mathscr{B}_I$, to be the tokens connected to A_i as well as the bonds creating these connections according to set W:

$$\begin{array}{rcl}\mathsf{con}(A_i,W) & = & (\{A_i\} \cap W) \\ & \cup & \{x \mid \exists w \text{ s.t. } \mathsf{path}(A_i,w,W), (B_i,C_i) \in w, x \in \{(B_i,C_i),B_i,C_i\}\}\end{array}$$

where $\mathsf{path}(A_i,w,W)$ if $w = \langle \beta_1,\ldots,\beta_n\rangle$, and for all $1 \leq i \leq n$, $\beta_i = (x_{i-1},x_i) \in W \cap \mathscr{B}_I$, $x_i \in W \cap \mathscr{A}_I$, and $x_0 = A_i$. For example, consider the net in Fig. 3 and let W represent the set of token and bond instances in place x. Then, $\mathsf{con}((I,3,[]),W) = \{(I_3,C_3,B_3,(I_3,C_3),(C_3,B_3)\}$, where $I_3 = (I,3,[])$, $B_3 = (B,3,[])$, and $C_3 = (C,3,[])$.

3.1 Forward Execution

During the forward execution of a transition in an MRPN, a set of token and bond instances, as specified by the incoming arcs of the transition, are selected and moved to the outgoing places of the transition, possibly forming and/or destroying bonds. Precisely, for a transition t we define $\mathsf{eff}^+(t)$ to be the bonds that occur on its outgoing arcs but not the incoming ones and by $\mathsf{eff}^-(t)$ the bonds that occur in the incoming arcs but not the outgoing ones:

$$\mathsf{eff}^+(t) = \mathsf{post}(t) - \mathsf{pre}(t) \qquad \mathsf{eff}^-(t) = \mathsf{pre}(t) - \mathsf{post}(t)$$

Due to the presence of multiple instances of the same token type, it is possible that different token instances are selected during the transition's execution. To enable such a selection of tokens we define the following:

Definition 4 An injective function $\mathscr{W} : V \rightarrow \mathscr{A}_I$, where $V \subseteq \mathscr{A}_V$, is called a *type-respecting assignment* if for all $a \in V$, if $\mathscr{W}(a) = A_i$ then $\mathsf{type}(a) = \mathsf{type}(A_i)$.

We extend the above notation and write $\mathscr{W}(a,b)$ for $(\mathscr{W}(a),\mathscr{W}(b))$ and, given a set $L \subseteq \mathscr{A}_V \cup \mathscr{B}_V$, we write $\mathscr{W}(L) = \{\mathscr{W}(x) \mid x \in L\}$.

Based on the above we define the following:

Definition 5 Given an MRPN $(P,T,\mathscr{A},\mathscr{A}_V,\mathscr{B},F)$, a state $\langle M,H\rangle$, and a transition t, we say that t is *forward-enabled* in $\langle M,H\rangle$ if there exists a type-respecting assignment $\mathscr{S}$: $\mathsf{pre}(t) \cap \mathscr{A}_V \to \mathscr{A}_I$ such that:

1. $\mathscr{S}(F(x,t)) \subseteq M(x)$ for all $x \in \circ t$.
2. If $a,b \in F(x,t)$ for some $x \in \circ t$ and $(a,b) \in \mathsf{eff}^+(t)$, then $\mathscr{S}(a,b) \notin M(x)$.
3. If $a \in F(t,y_1)$ and $b \in F(t,y_2)$, for some $y_1,y_2 \in t\circ$, $y_1 \neq y_2$, then $\mathsf{con}(\mathscr{S}(a),\mathsf{comp_f}(t,\mathscr{S},M)) \neq \mathsf{con}(\mathscr{S}(b),\mathsf{comp_f}(t,\mathscr{S},M))$.

where $\mathsf{comp_f}(t,\mathscr{S},M) = (\bigcup_{x\in\circ t} M(x) \cup \mathscr{S}(\mathsf{eff}^+(t))) - \mathscr{S}(\mathsf{eff}^-(t))$.

Thus, t is forward-enabled in state $\langle M,H\rangle$ if there exists a type-respecting assignment $\mathscr{S}$ of token instances to the variables on the incoming edges of t, which we will refer to as a *forward-enabling assignment* of t, such that (1) the token instances and bonds required by the transition's incoming edges, according to $\mathscr{S}$, are available from the appropriate input places, (2) if the selected token instances to be transferred by the transition are to be bonded together by the transition then they should not be already bonded in an incoming place of the transition (thus the bonds that occur only on the outgoing arcs of a transition are the bonds being created by the transition), and (3) if two token instances are transferred by a transition to different outgoing places then these tokens should not be connected. This is to ensure that connected components are not cloned. Note that $\mathsf{comp_f}(t,\mathscr{S},M) = (\bigcup_{x\in\circ t} M(x) \cup \mathscr{S}(\mathsf{eff}^+(t))) - S(\mathsf{eff}^-(t))$ denotes the set of token and bond instances that occur in the incoming places of t ($\bigcup_{x\in\circ t} M(x)$), including the new bond instances created by t ($\mathscr{S}(\mathsf{eff}^+(t))$), and removing the bonds destroyed by it ($\mathscr{S}(\mathsf{eff}^-(t))$). Intuitively, $\mathsf{comp_f}(t,\mathscr{S},M)$ contains the components that are moved forward by the transition.

To execute a transition t according to an enabling assignment $\mathscr{S}$, the selected token instances along with their connected components are relocated to the outgoing places of the transition as specified by the outgoing arcs, with bonds created and destroyed accordingly. An additional effect is the update of the affected token and bond instances to capture the executed transition in their causal path. To capture this update we define where k is an integer associated with the specific transition instance:

$$A_i \oplus (\mathscr{S},t,k) = \begin{cases} A_i + (k,t,a) & \text{if } \mathscr{S}(a) = A_i \\ A_i + (k,t,*) & \text{if } \mathscr{S}^{-1}(A_i) = \bot \end{cases}$$

Note that A_i may not belong to the range of $\mathscr{S}$, i.e. $\mathscr{S}^{-1}(A_i) = \bot$, if A_i was not specifically selected to instantiate a variable in $\mathsf{pre}(t)$ but, nonetheless, belonged to a connected component transferred by the transition. This is recorded in the causal path of the token instance via the triple $(k,t,*)$. Moreover, we write $(A_i,B_j) \oplus (\mathscr{S},t,k)$ for $(A_i \oplus (\mathscr{S},t,k), B_j \oplus (\mathscr{S},t,k))$ and, given $L \subseteq \mathscr{A}_I \cup \mathscr{B}_I$, we write $L \oplus (\mathscr{S},t,k) = \{x \oplus (\mathscr{S},t,k) \mid x \in L\}$. Finally, the history of the executed transition is updated to include the next unused integer. Given the above we define:

Definition 6 Given an MRPN $(P,T,\mathscr{A},\mathscr{A}_V,\mathscr{B},F)$, a state $\langle M,H\rangle$, a transition t that is enabled in state $\langle M,H\rangle$, and an enabling assignment $\mathscr{S}$, we write $\langle M,H\rangle \xrightarrow{(t,S)} \langle M',H'\rangle$ where for all $x \in P$:

$$\begin{array}{rcl} M'(x) & = & (M(x) - \bigcup_{a\in F(x,t)} \mathsf{con}(\mathscr{S}(a),M(x))) \\ & & \cup \bigcup_{a\in F(t,x)} \mathsf{con}(\mathscr{S}(a),\mathsf{comp_f}(t,\mathscr{S},M)) \oplus (\mathscr{S},t,k) \end{array}$$

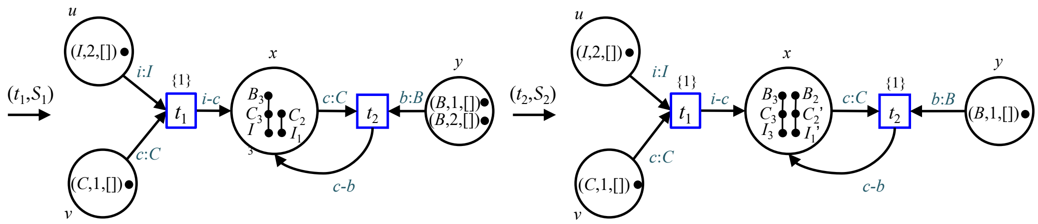


Figure 4: The effect of executing t_1 and t_2 in the net of Fig. 3, where $B_3 = (B,3,[])$, $C_3 = (C,3,[])$, $I_3 = (I,3,[])$, $I_1 = (I,1,[(1,t_1,i)])$, $C_2 = (C,2,[(1,t_1,c)]$, $I_1' = (I,1,[(1,t_1,i),(1,t_2,*)])$, $C_2' = (C,2,[(1,t_1,c),(1,t_2,c)])$, and $B_2 = (B,2,[(1,t_2,b)])$.

where $k = max(\{0\} \cup H(t)) + 1$ and

$$H'(t') = \begin{cases} H(t') \cup \{k\}, & \text{if } t' = t \\ H(t'), & \text{otherwise} \end{cases}$$

Fig. 4 shows the result of consecutively firing transitions t_1 and t_2 from the MRPN in Fig. 3 with enabling assignments $\mathscr{S}_1$, where $\mathscr{S}_1(i) = (I,1,[])$, $\mathscr{S}_1(c) = (C,2,[])$, and $\mathscr{S}_2$, where $\mathscr{S}_2(b) = (B,2,[])$, $\mathscr{S}_2(c) = (C,2,[(1,t_1,c)])$. We note the non-empty histories of the transitions depicted in the graphical representation, as well as the updates in the causal paths of the tokens.

3.2 Causal-order Reversing

We now move on to consider causal-order reversibility for MRPNs. In this form of reversibility, a transition can be reversed only if all its effects (if any), i.e. transitions that it has caused, have already been reversed. As argued in [24], two transition occurrences are causally dependent, if a token produced by the one was subsequently used to fire the other. Since token instances in MPRNs are associated with their causal path, we are able to identify the transitions that each token has participated in by observing its memory. Furthermore, if $\mathsf{last}(A_i) = (k,t,a)$ then the last transition that the token instance A_i has participated in was transition t and specifically its occurrence with history k.

Based on this observation, a transition occurrence t can be reversed in a certain state if the token/bonds instances it has employed have not engaged in any further transitions. Thus, we define causal reverse enabledness as follows.

Definition 7 Consider an MRPN $(P,T,\mathscr{A},\mathscr{A}_V,\mathscr{B},F)$, a state $\langle M,H \rangle$, and a transition t. We say that t is *co-enabled* in $\langle M,H \rangle$ if there exists a type-respecting assignment $\mathscr{R} : \mathsf{post}(t) \cap \mathscr{A}_V \to \mathscr{A}_I$ such that:

1. $\mathscr{R}(F(t,x)) \subseteq M(x)$ for all $x \in t\circ$, and
2. there exists $k \in H(t)$ such that for all $(A,i,xs) \in \bigcup_{x \in P} M(x)$ with $(k,t,b) \in xs$ for some b, $(k,t,b) = \mathsf{last}(A_i)$.

We refer to $\mathscr{R}$ as the *co*-reversal enabling assignment for the k^{th} occurrence of t.

Thus, a transition t is *co*-enabled in $\langle M,H \rangle$ for a specific occurrence k if there exists a type-respecting assignment of token instances on the variables of the outgoing arcs of the transition, which gives rise to a set of token and bond instances that are available in the relevant out-places and, additionally, these token/bond instances were last employed for the firing of the specific occurrence of the transition.

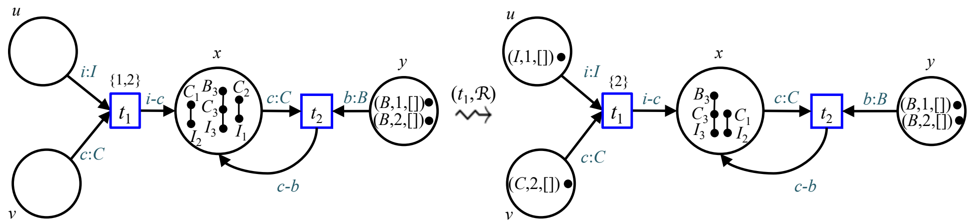


Figure 5: The effect of reversing transition t_1 with enabling assignment $\mathscr{R}(i) = I_1$, $\mathscr{R}(c) = C_2$, in a state following the execution of t_1 twice from the net in Fig. 3, first with enabling assignment $\mathscr{S}_1(i) = (I,1,[])$, $\mathscr{S}_1(c) = (C,2,[])$, and next with enabling assignment $\mathscr{S}_2(i) = (I,2,[])$, $\mathscr{S}_2(c) = (C,1,[])$ where we write $I_1 = (I,1,[(1,t_1,i])$, $I_2 = (I,2,[(2,t_1,i])$, $C_1 = (C,1,[(2,t_1,c])$, and $C_2 = (C,2,[(1,t_1,c])$.

To implement the reversal of a transition t according to a *co*-reversal enabling assignment $\mathscr{R}$, the selected token instances are relocated from the outgoing places of t to its incoming places, with bonds created and destroyed accordingly. The occurrence of the reversed transition is removed from its history.

Definition 8 Given an MRPN $(P,T,\mathscr{A},\mathscr{A}_V,\mathscr{B},F)$, a state $\langle M,H\rangle$, a transition t that is *co*-enabled with *co*-reversal enabling assignment $\mathscr{R}$ for the k^{th} occurrence of t, we write $\langle M,H\rangle \overset{(t,\mathscr{R})}{\rightsquigarrow} \langle M',H'\rangle$ where for all $x \in P$:

$$M'(x) = (M(x) - \bigcup_{a\in F(t,x)} \mathsf{con}(\mathscr{R}(a),M(x))) \cup \bigcup_{a\in F(x,t)} \mathsf{init}(\mathsf{con}(\mathscr{R}(a),\mathsf{comp_r}(t,\mathscr{R},M)))$$

and

$$H'(t') = \begin{cases} H(t') - \{k\}, & \text{if } t' = t \\ H(t'), & \text{otherwise} \end{cases}$$

where $\mathsf{comp_r}(t,\mathscr{R},M) = (\bigcup_{x\in t\circ} M(x) \cup \mathscr{R}(\mathsf{eff}^-(t))) - \mathscr{R}(\mathsf{eff}^+(t))$.

In Fig. 5 we may observe the causal-order reversal of transition t_1. We note that the history information of the affected components is updated by removing the occurrence of the reversed transition and the history information of transition t_1 reflects that occurrence with identifier 1 has been reversed.

Let us now consider executions of both forward and backward moves and write $\longmapsto$ for $\longrightarrow \cup \rightsquigarrow$. We define the reachable states of an MRPN as follows.

Definition 9 Given an MRPN $(P,T,\mathscr{A},\mathscr{A}_V,\mathscr{B},F)$ and an initial state $\langle M_0,H_0\rangle$ we say that state $\langle M,H\rangle$ is *reachable*, if there exist $\langle M_i,H_i\rangle$, $i \le n$ for some $n \ge 0$, such that $\langle M_0,H_0\rangle \overset{(t_1,\mathscr{W}_1)}{\longmapsto} \langle M_1,H_1\rangle \overset{(t_2,\mathscr{W}_2)}{\longmapsto} \dots \overset{(t_n,\mathscr{W}_n)}{\longmapsto} \langle M_n,H_n\rangle = \langle M,H\rangle$.

Furthermore, given a type A, an integer i, and a marking M, we write $\mathsf{num}(A,i,M)$ for the number of token instances of the form (A,i,xs) in M, defined by $\mathsf{num}(A,i,M) = |\{(x,A_i) \mid \exists x \in P, A_i \in M(x), A_i{\downarrow} = (A,i)\}|$. Similarly, for a bond instance $\beta_i \in \mathscr{B}_I$, we define $\mathsf{num}(\beta_i,M) = |\{x \in P \mid \beta_i \in M(x)\}|$. The following result confirms that in an execution beginning in the initial state of an MRPN, token instances are preserved, at most one bond instance may occur at any time, and a bond instance may be created/destroyed during a forward/reverse execution of a transition that features the bond as its effect.

Proposition 1 Given an MRPN $(P,T,\mathscr{A},\mathscr{A}_V,\mathscr{B},F)$, a reachable state $\langle M,H\rangle$, and a transition firing $\langle M,H\rangle \xrightarrow{(t,\mathscr{W})} \langle M',H'\rangle$, the following hold:

1. For all A, i, $\mathsf{num}(A,i,M') = \mathsf{num}(A,i,M)$=1.
2. For all $\beta_i \in \mathscr{B}_I$,
   (a) $0 \leq \mathsf{num}(\beta_i,M') \leq 1$,
   (b) if t is executed in the forward direction with forward enabling assignment $\mathscr{S}$ and $\beta_i \in \mathscr{S}(\mathsf{eff}^+(t))$ then $\mathsf{num}(\beta_i,M') = 1$; if instead $\beta_i \in \mathscr{S}(\mathsf{eff}^-(t))$ then $\mathsf{num}(\beta_i,M') = 0$, otherwise $\mathsf{num}(\beta_i,M) = \mathsf{num}(\beta_i,M')$.
   (c) if t is executed in the reverse direction with reverse enabling assignment $\mathscr{R}$ and $\beta_i \in \mathscr{R}(\mathsf{eff}^+(t))$ then $\mathsf{num}(\beta_i,M') = 0$; if instead $\beta_i \in \mathscr{R}(\mathsf{eff}^-(t))$ then $\mathsf{num}(\beta_i,M') = 1$, otherwise $\mathsf{num}(\beta_i,M) = \mathsf{num}(\beta_i,M')$.

Proof: The proof follows by induction on the length of the execution reaching state $\langle M,H\rangle$. If this is the initial state the result (i.e. clauses 1 and 2(a)) follows by our assumption on the initial state. For the induction step, let us assume that $\langle M,H\rangle$ satisfies the conditions of the proposition.

Let us begin with clause (1) and suppose $\xrightarrow{(t,\mathscr{W})} = \xrightarrow{(t,\mathscr{S})}$, where $\mathscr{S}$ is the forward-enabling assignment for the transition, and let $A_i = (A,i,xs) \in \mathscr{A}_I$. Two cases exist:

1. $A_i \in \mathsf{con}(B_j,M(x))$ for some B_j, $\mathscr{S}(a) = B_j$, $a \in F(x,t)$. Note that x is unique by the assumption that $\mathsf{num}(A,i,M) = 1$. To discern the location of A_i in M' two cases exist.
   - Suppose $A_i \in \mathsf{con}(B_j,\mathsf{comp_f}(t,\mathscr{S},M))$. We observe that, by Definition 2(1), $a \in \mathsf{post}(t)$. Thus, there exists $y \in t\circ$, such that $a \in F(t,y)$. Note that this y is unique by Definition 2(2). As a result, by Definition 6, $\mathsf{con}(B_j,\mathsf{comp_f}(t,\mathscr{S},M)) \subseteq M'(y)$, which implies that $A_i \in M'(y)$.
   - Suppose $A_i \notin \mathsf{con}(B_j,\mathsf{comp_f}(t,\mathscr{S},M'))$ and consider $w = \langle(A_{i_1},A_{i_2}),\ldots,(A_{i_n},B_j)\rangle$, $A_i = A_{i_1}$, $n \geq 1$, such that $\mathsf{path}(A_i,w,M(x))$. Since $A_i \notin \mathsf{con}(B_j,\mathsf{comp_f}(t,\mathscr{S},M'))$ it must be that for some k, $(A_{i_{k-1}},A_{i_k}) \in \mathscr{S}(\mathsf{eff}^-(t))$ and $A_i \in \mathsf{con}(A_{i_k},M(x) - \mathscr{S}(\mathsf{eff}^-(t)))$. Using the same argument as in the previous case for A_{i_k} instead of B_j, we may conclude that $A_i \in M(y)$ such that $\mathscr{S}(b) = A_{i_k}$ and $b \in F(t,y)$.

   Now suppose that $A_i \in \mathsf{con}(C_k,\mathsf{comp_f}(t,\mathscr{S},M))$, $C_k = \mathscr{S}(b)$ for some $b \neq a$, $b \in F(t,y')$. Then it must be that $y = y'$. As a result, we have that $\mathsf{num}(A,i,M') = \mathsf{num}(A,i,M) = 1$ and the result follows.
2. $A_i \notin \mathsf{con}(\mathscr{S}(b),M(x))$ for all $b \in F(x,t)$, $x \in P$. This implies that $\{x \in P \mid A_i \in M'(x)\} = \{x \in P \mid A_i \in M(x)\}$ and the result follows.

Now suppose $\xrightarrow{(t,\mathscr{W})} = \overset{(t,\mathscr{R})}{\rightsquigarrow}$ where $\mathscr{R}$ is the reverse-enabling assignment of the transition. Consider $A_i = (A,i,xs) \in \mathscr{A}_I$. Two cases exist:

1. $A_i \in \mathsf{con}(B_j,M(x))$ for some B_j, $\mathscr{R}(a) = B_j$, $a \in F(t,x)$. Note that x is unique by the assumption that $\mathsf{num}(A,i,M) = 1$. To discern the location of A_i in M' two cases exist.
   - Suppose $A_i \in \mathsf{con}(B_j,\mathsf{comp_r}(t,\mathscr{R},M'))$. We observe that, by Definition 2(1), $a \in \mathsf{pre}(t)$. Thus, there exists $y \in \circ t$, such that $a \in F(y,t)$. Note that this y is unique by Definition 2(3). As a result, by Definition 8,

   $$M'(y) = M(x) - \bigcup_{a \in F(t,x)} \mathsf{con}(\mathscr{R}(a),M(x))) \cup \bigcup_{a \in F(x,t)} \mathsf{init}(\mathsf{con}(\mathscr{R}(a),\mathsf{comp_r}(t,\mathscr{R},M)))$$

   Since $a \in F(y,t) \cap F(t,x)$, $A_i \in \mathsf{con}(\mathscr{R}(a),M(x) \cup F(y,t))$, which implies that $a \in M'(y)$.

- Suppose $A_i \notin \mathsf{con}(B_j, \mathsf{comp_r}(t,\mathscr{R},M'))$ and consider $w = \langle (A_{i_1}, A_{i_2}), \ldots, (A_{i_n}, B_j)\rangle$, $A_i = A_{i_1}$, $n \geq 1$, such that $\mathsf{path}(A_i, w, M(x))$. Since $A_i \notin \mathsf{con}(B_j, \mathsf{comp_r}(t,\mathscr{R},M'))$ it must be that for some k, $(A_{i_{k-1}}, A_{i_k}) \in \mathscr{R}(\mathsf{eff}^+(t))$ and $A_i \in \mathsf{con}(A_{i_k}, M(x) - \mathscr{R}(\mathsf{eff}^+(t)))$. Using the same argument as in the previous case for A_{i_k} instead of B_j, we may conclude that $A_i \in M(y)$ such that $\mathscr{S}(b) = A_{i_k}$ and $b \in F(y,t)$.

   Now suppose that $A_i \in \mathsf{con}(C_k, \mathsf{comp_r}(t,\mathscr{R},M))$, $C_k = \mathscr{R}(b)$ for some $a \neq b$, $b \in F(y',t)$. Then it must be that $y = y'$. As a result, we have that $\{z \in P \mid A_i \in M'(z)\} = \{y\}$ and the result follows.

2. $A_i \notin \mathsf{con}(\mathscr{R}(a), M(x))$ for all $a \in F(t,x)$, $x \in P$. This implies that $\{x \in P \mid A_i \in M'(x)\} = \{x \in P \mid A_i \in M(x)\}$ and the result follows.

The proof of clause 2 follows similar arguments. □

We may now proceed to establish the causal consistency of our semantics. We begin with defining when two states of an MRPN are considered to be causally equivalent. Intuitively, states $\langle M,H\rangle$ and $\langle M',H'\rangle$ are causally equivalent whenever the executions that have led to them contain the same causal paths. Note that these causal paths refer to different independent threads of computation, possibly executed through different interleavings in the executions leading to $\langle M,H\rangle$ and $\langle M',H'\rangle$. In our setting, we can enunciate this requirement by observing the causal histories of token instances and requiring that for each token instance of some type A in one of the two states there is a token instance of the same type that has participated in the exact same sequence of transitions in the other state:

Definition 10 Consider MRPN $(P,T,\mathscr{A},\mathscr{A}_V,\mathscr{B},F)$ and reachable states $\langle M,H\rangle$, $\langle M',H'\rangle$. Then the states are *causally equivalent*, denoted by $\langle M,H\rangle \asymp \langle M',H'\rangle$, if for each $x \in P$, $A_i \in M(x)$ there exists $A_j \in M'(x)$ with $\mathsf{cpath}(A_i) = \mathsf{cpath}(A_j)$, and vice versa.

We may now establish the Loop Lemma for our model.

Lemma 1 (Loop) For any forward transition $\langle M,H\rangle \xrightarrow{(t,\mathscr{S})} \langle M',H'\rangle$ there exists a backward transition $\langle M',H'\rangle \overset{(t,\mathscr{R})}{\rightsquigarrow} \langle M,H\rangle$ and for any backward transition $\langle M,H\rangle \overset{(t,\mathscr{R})}{\rightsquigarrow} \langle M',H'\rangle$ there exists a forward transition $\langle M',H'\rangle \xrightarrow{(t,\mathscr{S})} \langle M'',H''\rangle$ where $\langle M,H\rangle \asymp \langle M'',H''\rangle$.

Proof: Suppose $\langle M,H\rangle \xrightarrow{(t,\mathscr{S})} \langle M',H'\rangle$. Then t is clearly reverse-enabled in $\langle M',H'\rangle$ with reverse-enabling assignment $\mathscr{R}$ such that if $\mathscr{S}(a) = (A,i,xs)$, then $\mathscr{R}(a) = (A,i,xs + (t,k,a))$, where k is the maximum element of $H(t)$. Furthermore, $\langle M',H'\rangle \overset{t,\mathscr{R}}{\rightsquigarrow} \langle M'',H''\rangle$ where $H'' = H$. In addition, all token and bond instances involved in transition t (except those in $\mathsf{eff}^+(t)$) will be returned from the outgoing places of transition t back to its incoming places. At the same time, all destroyed bonds (those in $\mathsf{eff}^-(p)$) will be re-formed, according to Proposition 1. Specifically, for all $A_i \in \mathscr{A}_I$, it is easy to see by the definition of $\rightsquigarrow$ that $A_i \in M''(x)$ if and only if $A_i \in M(x)$. Similarly, for all $\beta_i \in \mathscr{B}_I$, $\beta_i \in M''(x)$ if and only if $\beta_i \in M(x)$. The opposite direction can be argued similarly, with the distinction that when a transition is executed immediately following its reversal, it is possible that the transition instance is assigned a different key, thus giving rise to a state $\langle M'',H''\rangle$ distinct but causally equivalent to $\langle M,H\rangle$. □

We now proceed to define some auxiliary notions. Given a transition $\langle M,H\rangle \overset{(t,\mathscr{W})}{\longmapsto} \langle M',H'\rangle$, we say that the *action* of the transition is $(t,\mathscr{W})$ if $\langle M,H\rangle \xrightarrow{(t,\mathscr{W})} \langle M',H'\rangle$ and $(\underline{t},\mathscr{W})$ if $\langle M,H\rangle \overset{(t,\mathscr{W})}{\rightsquigarrow} \langle M',H'\rangle$ and we may write $\langle M,H\rangle \overset{(t,\mathscr{W})}{\longmapsto} \langle M',H'\rangle$. We write Act_N for the set of all actions in an MRPN N. We use α to range over $\{t,\underline{t} \mid t \in T\}$ and write $\underline{\underline{t}} = t$. Given an execution $\langle M_0,H_0\rangle \overset{(\alpha_1,\mathscr{W}_1)}{\longmapsto} \ldots \overset{(\alpha_n,\mathscr{W}_n)}{\longmapsto} \langle M_n,H_n\rangle$, we say that the *trace* of the execution is $\sigma = \langle(\alpha_1,\mathscr{W}_1),(\alpha_2,\mathscr{W}_2),\ldots,(\alpha_n,\mathscr{W}_n)\rangle$, and write $\langle M,H\rangle \overset{\sigma}{\longmapsto}$

$\langle M_n, H_n\rangle$. Given $\sigma_1 = \langle(\alpha_1,\mathscr{W}_1),\ldots,(\alpha_k,\mathscr{W}_k)\rangle$, $\sigma_2 = \langle(\alpha_{k+1},\mathscr{W}_{k+1}),\ldots,(\alpha_n,\mathscr{W}_n)\rangle$, we write $\sigma_1;\sigma_2$ for $\langle(\alpha_1,\mathscr{W}_1),\ldots,(\alpha_n,\mathscr{W}_n)\rangle$. We may also use the notation $\sigma_1;\sigma_2$ when σ_1 or σ_2 is a single transition. A central concept in what follows is causal equivalence on traces, a notion that employs the concept of concurrent transitions:

Definition 11 Consider an MRPN $(P,T,\mathscr{A},\mathscr{A}_V,\mathscr{B},F)$, a reachable state $\langle M,H\rangle$ and actions $(\alpha_1,\mathscr{W}_1)$ and $(\alpha_2,\mathscr{W}_2)$. Then $(\alpha_1,\mathscr{W}_1)$ and $(\alpha_2,\mathscr{W}_2)$ are said to be *concurrent* in state $\langle M,H\rangle$, if for all $u,v \in \mathscr{A}_V$, if $\mathscr{W}_1(u) = A_i$ and $\mathscr{W}_2(v) = B_j$, $A_i, B_j \in M(x)$ then $\mathsf{con}(A_i,M(x)) \neq \mathsf{con}(B_j,M(x))$.

Thus, two actions are concurrent when they employ different token instances. This notion captures when two actions are independent, i.e. the execution of the one does not preclude the other. Indeed, we may prove the following results.

Proposition 2 (Square Property) Consider an MRPN $(P,T,\mathscr{A},\mathscr{A}_V,\mathscr{B},F)$, a reachable state $\langle M,H\rangle$ and concurrent actions $(\alpha_1,\mathscr{W}_1)$ and $(\alpha_2,\mathscr{W}_2)$ in $\langle M,H\rangle$, such that $\langle M,H\rangle \stackrel{(\alpha_1,\mathscr{W}_1)}{\longmapsto} \langle M_1,H_1\rangle$ and $\langle M,H\rangle \stackrel{(\alpha_2,\mathscr{W}_2)}{\longmapsto}$ $\langle M_2,H_2\rangle$. Then $\langle M_1,H_1\rangle \stackrel{(\alpha_2,\mathscr{W}_2)}{\longmapsto} \langle M',H'\rangle$ and $\langle M_2,H_2\rangle \stackrel{(\alpha_1,\mathscr{W}_1)}{\longmapsto} \langle M'',H''\rangle$, where $\langle M',H'\rangle \asymp \langle M'',H''\rangle$.

Proof: It is easy to see that since the two transitions involve distinct tokens then they can be executed in any order. If, additionally, $\alpha_1 \neq \alpha_2$ or $\alpha_1 = \alpha_2$ and they are both reverse transitions, then the effects imposed on the histories and the tokens of the transitions will be independent and the same in both cases, i.e. $\langle M',H'\rangle = \langle M',H'\rangle$. If instead $\alpha_1 = \alpha_2$ and α_1, α_2 are not both reverse transitions, then it is possible that distinct tokens will be assigned to the forward transition(s). Nonetheless, the sequence of actions executed by each token instance will be the same in both interleavings and, thus, the resulting states are causally equivalent. □

Proposition 3 (Reverse Transitions are Independent) Consider an MRPN $(P,T,\mathscr{A},\mathscr{A}_V,\mathscr{B},F)$, a state $\langle M,H\rangle$ and enabled reverse actions $(\underline{t_1},\mathscr{R}_1)$ and $(\underline{t_2},\mathscr{R}_2)$ where $(\underline{t_1},\mathscr{R}_1) \neq (\underline{t_2},\mathscr{R}_2)$. Then,$(\underline{t_1},\mathscr{R}_1)$ and $(\underline{t_2},\mathscr{R}_2)$ are concurrent.

Proof: It is straightforward to see that two distinct reverse transitions employ different tokens. This is because a token instance may only reverse the last transition occurrence in its history. Therefore $(\underline{t_1},\mathscr{R}_1)$ and $(\underline{t_2},\mathscr{R}_2)$ satisfy the requirement for being concurrent. □

We also define two transitions to be opposite in a certain state as follows:

Definition 12 Consider an MRPN $(P,T\mathscr{A},\mathscr{A}_V,\mathscr{B},F)$ and actions $(\alpha_1,\mathscr{W}_1)$ and $(\alpha_2,\mathscr{W}_2)$. Then $(\alpha_1,\mathscr{W}_1)$ and $(\alpha_2,\mathscr{W}_2)$ are said to be *opposite* if $\underline{\alpha_1} = \alpha_2$ and, if $\alpha_i = t$ for some t, for all $a \in \mathsf{pre}(t)$, $\mathsf{init}(\mathscr{W}_i(a)) = \mathscr{W}_{3-i}(a)$.

Note that this may arise exactly when the two actions are forward and reverse executions of the same transition and using the same token instances. We are now ready to define when two traces are causally equivalent.

Definition 13 Consider a reachable state $\langle M,H\rangle$. Then *causal equivalence on traces with respect to* $\langle M,H\rangle$, denoted by $\sigma_1 \asymp_{\langle M,H\rangle} \sigma_2$, is the least equivalence relation on traces such that (i) $\sigma_1 = \sigma;(\alpha_1,\mathscr{W}_1);(\alpha_2,\mathscr{W}_2);\sigma'$ where $\langle M,H\rangle \stackrel{\sigma}{\longmapsto} \langle M',H'\rangle$ and if $(\alpha_1,\mathscr{W}_1)$ and $(\alpha_2,\mathscr{W}_2)$ are concurrent in $\langle M',H'\rangle$ then $\sigma_2 = \sigma;(\alpha_2,\mathscr{W}_2);(\alpha_1,\mathscr{W}_1);\sigma'$, and (ii) if $(\alpha_1,\mathscr{W}_1)$ and $(\alpha_2,\mathscr{W}_2)$ are opposite transitions then $\sigma_2 = \sigma;\varepsilon;\sigma'$.

We may now establish the Parabolic Lemma, which states that causal equivalence allows the permutation of reverse and forward transitions that have no causal relations between them. Therefore, computations are allowed to reach for the maximum freedom of choice going backward and then continue forward.

Lemma 2 (Parabolic Lemma) Consider an MRPN $(P,T,\mathscr{A},\mathscr{A}_V,\mathscr{B},F)$, a reachable state $\langle M,H\rangle$, and an execution $\langle M,H\rangle \xmapsto{\sigma} \langle M',H'\rangle$. Then there exist traces r,r' both forward such that $\sigma \asymp_{\langle M,H\rangle} \underline{r};r'$ and $\langle M,H\rangle \xmapsto{\underline{r};r'} \langle M'',H''\rangle$ where $\langle M',H'\rangle \asymp \langle M'',H''\rangle$.

Proof Following [17], given the satisfaction of the Square Property (Proposition 2) and the independence of reverse transitions (Proposition 3), we conclude that the lemma holds. A proof from first principles may also be found in [30]. □

We conclude with Theorem 1 stating that two computations beginning in the same state lead to equivalent states if and only if the two computations are causally equivalent. This guarantees the consistency of the approach since reversing transitions in causal order is in a sense equivalent to not executing the transitions in the first place. Reversal does not give rise to previously unreachable states, on the contrary, it gives rise to causally-equivalent states due to different keys being possibly assigned to concurrent transitions.

Theorem 1 Consider an MRPN $(P,T,\mathscr{A},\mathscr{A}_V,\mathscr{B},F)$, a reachable state $\langle M,H\rangle$, and traces σ_1, σ_2 such that $\langle M,H\rangle \xmapsto{\sigma_1} \langle M_1,H_1\rangle$ and $\langle M,H\rangle \xmapsto{\sigma_2} \langle M_2,H_2\rangle$. Then, $\sigma_1 \asymp_{\langle M,H\rangle} \sigma_2$ if and only if $\langle M_1,H_1\rangle \asymp \langle M_2,H_2\rangle$.

Proof: Following [17], given the satisfaction of the Parabolic Lemma and the fact that the model does not allow infinite reverse computations, we conclude that the theorem holds. A proof from first principles may also be found in [30]. □

4 Multi Tokens versus Single Tokens

We now proceed to define Single Reversing Petri Nets as MRPNs where each token type corresponds to exactly one token instance.

Definition 14 A *Single Reversing Petri Net* (SRPN) $(P,T,\mathscr{A},\mathscr{A}_V,\mathscr{B},F)$ is an MRPN where for all $A \in \mathscr{A}$, $|A| = 1$.

Forward and causal-order reversal for SRPNs is defined as for MRPNs. Consequently, SPRNs are special instances of MRPNs. In the sequel, we will show that for each MRPN there is an "equivalent" SRPN. To achieve this, similarly to [31], we will employ Labelled Transition Systems defined as follows:

Definition 15 A labelled transition system (LTS) is a tuple $(Q,E,\rightarrow,I)$ where:

- Q is a countable set of states,
- E is a countable set of actions,
- $\rightarrow \subseteq Q \times E \times Q$ is the step transition relation, where we write $p \xrightarrow{u} q$ for $(p,u,q) \in \rightarrow$, and
- $I \in Q$ is the initial state.

For the purposes of our comparison, we will employ LTSs in the context of isomorphism of reachable parts:

Definition 16 Two LTSs $L_1 = (Q_1,E_1,\rightarrow_1,I_1)$ and $L_2 = (Q_2,E_2,\rightarrow_2,I_2)$ are isomorphic, written $L_1 \cong L_2$, if they differ only in the names of their states and events, i.e. if there are bijections $\gamma : Q_1 \rightarrow Q_2$ and $\eta : E_1 \rightarrow E_2$ such that $\gamma(I_1) = I_2$, and, for $p,q \in Q_1$, $u \in E_1$: $\gamma(p) \xrightarrow{\eta(u)}_2 \gamma(q)$ iff $p \xrightarrow{u}_1 q$.

The set $\mathscr{R}(Q)$ of reachable states in $L = (Q, E, \rightarrow, I)$ is the smallest set such that I is reachable and whenever p is reachable and $p \xrightarrow{u} q$ then q is reachable. The reachable part of L is the LTS $\mathscr{R}(L) = (R(Q), E, \rightarrow_{\mathscr{R}}, I)$, where $\rightarrow_{\mathscr{R}}$ is the part of the transition relation restricted to reachable states. We write $L_1 \cong_{\mathscr{R}} L_2$ if $\mathscr{R}(L_1)$ and $\mathscr{R}(L_2)$ are isomorphic. To check $L_1 \cong_{\mathscr{R}} L_2$ it suffices to restrict to subsets of Q_1 and Q_2 that contain all reachable states, and construct an isomorphism between the resulting LTSs.

We proceed to give a translation from MRPNs to SPRNs. First, we present how an LTS can be associated with an MRPN/SRPN structure.

Definition 17 Let $N = (P, T, \mathscr{A}, \mathscr{A}_V, \mathscr{B}, F)$ be an MRPN (or SRPN) with initial marking M_0. Then $\mathscr{H}(N, M_0) = ((P \rightarrow 2^{\mathscr{A}_I \cup \mathscr{B}_I}) \times (T \rightarrow 2^{\mathbb{N}}), Act, \longmapsto, \langle M_0, H_0 \rangle)$ is the LTS associated with N.

We may now establish that for any MRPN there exists an SPRN with an isomorphic LTS.

Theorem 2 For every MRPN $N = (P, T, \mathscr{A}, \mathscr{A}_V, \mathscr{B}, F)$ with initial marking M_0 there exists an SRPN $N' = (P, T', \mathscr{A}', \mathscr{A}'_V, \mathscr{B}', F')$ with initial marking M'_0 such that $\mathscr{H}(N, M_0) \cong_{\mathscr{R}} \mathscr{H}(N', M'_0)$.

Proof: Let $N = (P, T, \mathscr{A}, \mathscr{A}_V, \mathscr{B}, F)$ be an MRPN with initial state $\langle M_0, H_0 \rangle$. We introduce the notation $\mathscr{W}\!\downarrow$ where for any type-respecting assignment $\mathscr{W}$, $\mathscr{W}\!\downarrow\!(a) = \mathscr{W}(a)\!\downarrow$, that is $\mathscr{W}$ assigns to a variable in the range of $\mathscr{W}$ the token instance associated to it by $\mathscr{W}$ but with its history removed. Furthermore, if $f = \mathscr{W}\!\downarrow$ we write $f_s(a) = a_i$ if $f(a) = (A, i)$. We construct $N' = (P, T', \mathscr{A}', \mathscr{A}'_V, \mathscr{B}', F')$ with initial state $\langle M'_0, H'_0 \rangle$ as follows:

$$\begin{array}{rcl}
\mathscr{A}' & = & \{A_i \mid \exists (A, i, []) \in M_0(x) \text{ for some } x \in P\} \\
\mathscr{A}'_V & = & \{a_i \mid A_i \in \mathscr{A}'\} \\
\mathscr{B}' & = & \{(A_i, B_j) \mid A_i, B_j \in \mathscr{A}', (A, B) \in \mathscr{B}\} \\
T' & = & \{t_{\mathscr{W}\downarrow} \mid t \in T, \mathscr{W} : \mathsf{pre}(t) \rightarrow \mathscr{A}_I \text{ is a type-respecting assignment } \} \\
F'(x_1, x_2) & = & \{f_s(a) \mid \exists i \in \{1,2\}, x_i = t_f \in T', \text{ and } a \in \mathsf{pre}(t)\} \\
 & \cup & \{(f_s(a), f_s(b)) \mid \exists i \in \{1,2\}, x_i = t_f \in T', (a,b) \in \mathsf{pre}(t)\} \\
M'_0(x) & = & \{(A_i, 1, []) \mid (A, i, []) \in M_0(x)\},\ \forall x \in P \\
H'_0(t) & = & \emptyset,\ \forall t \in T'
\end{array}$$

The above construction, projects each type A in N to a set of types A_i in N' such that, $A_i \in \mathscr{A}'$ for each instance $(A, i, [])$ of type A in M_0. Type A_i contains exactly one element, initially named $(A_i, 1, [])$. Furthermore, for each transition $t \in T$, we create a set of transitions of the form $t_f \in T'$, to associate all possible ways in which token/bond instances may be taken as input by t with a distinct transition that takes as input the combination of types projected to by the instances.

We now proceed to define bijections γ and η for establishing the homomorphism between the two LTSs. To simplify the proof, we assume that during the execution of transitions the enabling assignment is recorded both in the transition histories, i.e. given a transition t we have $(k, \mathscr{W}) \in H(t)$ signifying that the k^{th} occurrence of t was executed with enabling assignment $\mathscr{W}$ and also in a token instance (A, i, xs) elements of xs have the form $(k, t, v, \mathscr{W}\!\downarrow)$ again recording the assignment that enabled the specific execution of the transition occurrence. In this setting, it is easy to associate each token instance of N to a token instance of N' as follows, where we write $st(A_i)$ for the equivalent token instance of A_i in the SPRN N':

$st((A, i, xs)) = (A_i, 1, ys)$

where, if $xs = [(k^i, t^i, v^i, f^i)]_{1 \leq i \leq n}$ then $ys = [(|\{(k, t^i, v, f) \mid \exists k, v, f \text{ s.t.} (k, t^i, v, f) \in ys\}|, t^i_f, f_s(a))]_{1 \leq i \leq n}$.

For any reachable state $\langle M,H\rangle$ in LTS $\mathscr{H}(N,M_0)$, we define $\gamma(\langle M,H\rangle)=\langle M',H'\rangle$ such that for all $x\in P$ and $t_f\in T'$

$$\begin{array}{rcl} M'(x) & = & \{st(A_i)\mid A_i\in M(x)\}\cup\{(st(A_i),st(B_j))\mid (A_i,B_j)\in M(x)\} \\ H'(t_f) & = & \{1,\ldots,k\mid k=|\{(i,\mathscr{R})\in H(t)\mid \mathscr{R}\!\downarrow = f\}|\} \end{array}$$

Furthermore, given an action $(t,\mathscr{W})$, we write

$$\eta((t,\mathscr{W}))=(t_{\mathscr{W}\downarrow},\mathscr{W}')$$

where if $\mathscr{W}(a)=A_i$ then $\mathscr{W}'(a_i)=st(A_i)$.

Based on these, we may confirm that there exists an isomorphism between the LTSs $\mathscr{H}(N,M_0)$ and $\mathscr{H}(N',M_0')$ as follows. Suppose $\langle M_m,H_m\rangle$ is a reachable state of $\mathscr{H}(N,M_0)$ with $\gamma(\langle M_m,H_m\rangle)=\langle M_s,H_s\rangle$. Two cases exist:

- Suppose $\langle M_m,H_m\rangle \xrightarrow{(t,\mathscr{S})} \langle M_m',H_m'\rangle$. This implies that t is a forward-enabled transition with forward-enabling assignment $\mathscr{S}$. Consider $\eta(t,\mathscr{S})=(t_{\mathscr{S}\downarrow},\mathscr{S}')$, as defined above. It is easy to see that $t_{\mathscr{S}\downarrow}$ is also a forward-enabled transition in $\langle M_s,H_s\rangle$ with forward-enabling assignment $\mathscr{S}'$. Furthermore, if $\langle M_s,H_s\rangle \xrightarrow{(t_{\mathscr{S}\downarrow},\mathscr{S}')} \langle M_s',H_s'\rangle$, then

$$\begin{array}{rcl} M_s'(x) & = & (M_s(x)-\bigcup_{a\in F'(x,t_{\mathscr{S}\downarrow})}\mathsf{con}(\mathscr{S}'(a),M'(x))) \\ & \cup & \bigcup_{a\in F'(t_{\mathscr{S}\downarrow},x)}\mathsf{con}(\mathscr{S}'(a),\mathsf{comp_f}(t_{\mathscr{S}\downarrow},\mathscr{S}',M_s))\oplus(\mathscr{S}',t_{\mathscr{S}},k) \\ & = & (M_s(x)-\bigcup_{a\in F(x,t)}\{st(A_i),(st(A_i),st(B_j))\mid A_i,(A_i,B_j)\in\mathsf{con}(\mathscr{S}(a),M_m(x)))\\ & \cup & \bigcup_{a\in F(t,x)}\{st(A_i),(st(A_i),st(B_j))\mid A_i,(A_i,B_j)\in \\ & & \qquad \mathsf{con}(\mathscr{S}(a),\mathsf{comp_f}(t,\mathscr{S},M_m))\oplus(\mathscr{S},t,k)\} \end{array}$$

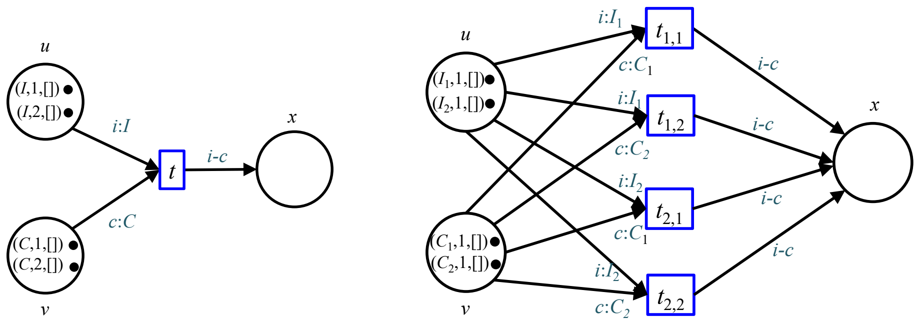
  where $k=max(\{0\}\cup\{k'|k'\in H(t)\})+1$ and

$$H'(t)=\begin{cases} H(t)\cup\{k\}, & \text{if } t=t_{\mathscr{S}\downarrow} \\ H(t'), & \text{otherwise}\end{cases}$$

  We may see that $\gamma(\langle M_m,H_m\rangle)=\langle M_s,H_s\rangle$, and the result follows. Reversing the arguments, we may also prove the opposite direction.

- Suppose $\langle M_m,H_m\rangle \overset{(t,\mathscr{R})}{\rightsquigarrow} \langle M_m',H_m'\rangle$. This implies that t is a reverse enabled transition with enabling assignment $\mathscr{R}$. Consider $\eta(t,\mathscr{R})=(t_{\mathscr{R}\downarrow},\mathscr{R}')$, as defined above. It is easy to see that $t_{\mathscr{R}\downarrow}$ is also a reverse-enabled transition in $\langle M_s,H_s\rangle$ with reverse-enabling assignment $\mathscr{R}'$. Furthermore, if $\langle M_s,H_s\rangle \overset{(t_{\mathscr{R}\downarrow},\mathscr{R}')}{\rightsquigarrow} \langle M_s',H_s'\rangle$, then using similar arguments as in the previous case we may confirm that $\gamma(\langle M_m,H_m\rangle)=\langle M_s,H_s\rangle$. The same holds for the opposite direction. This completes the proof. □

In Fig. 6 we present an MRPN N and its respective SRPN N'. From N we obtain N' by constructing the new token types I_1,I_2,C_1,C_2 and exactly one token instance of each of these types. The places are the same in both RPN models. The transitions required for the SRPN are dependent on the types of the



(a) MRPN N (b) Equivalent SRPN N'

Figure 6: Translating MRPNs to SRPNs

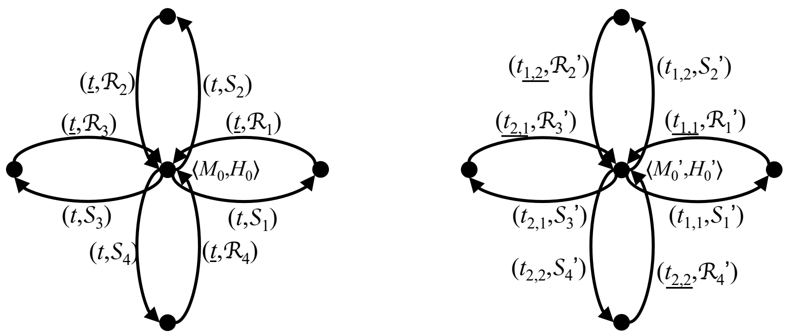
variables required for each MRPN transition and the token instances representing that type. Specifically for each token-instance combination that may fire a transition in the MRPN, a respective transition is required in the SRPN. In the example, two token instances of type I can be instantiated to variable i and two token instances of type C can be instantiated to variable v. This yields four combinations of token instances resulting in four different transitions.

In Fig. 7 we may see the isomorphic LTSs of the two RPNs, where

$$
\begin{aligned}
t_{1,1} &= t_{\mathscr{S}_1\downarrow} & \underline{t_{1,1}} &= \underline{t_{\mathscr{R}_1\downarrow}} \\
t_{1,2} &= t_{\mathscr{S}_2\downarrow} & \underline{t_{1,2}} &= \underline{t_{\mathscr{R}_2\downarrow}} \\
t_{2,1} &= t_{\mathscr{S}_3\downarrow} & \underline{t_{2,1}} &= \underline{t_{\mathscr{R}_3\downarrow}} \\
t_{2,2} &= t_{\mathscr{S}_4\downarrow} & \underline{t_{2,3}} &= \underline{t_{\mathscr{R}_4\downarrow}}
\end{aligned}
$$

and the enabling assignments of the actions in the two LTSs are

$$
\begin{aligned}
\mathscr{S}_1(i) &= (I,1,[]), & \mathscr{S}_1(c) &= (C,1,[]) \\
\mathscr{R}_1(i) &= (I,1,[(t,1,i)]), & \mathscr{R}_1(c) &= (C,1,[(t,1,c)]) \\
\mathscr{S}_2(i) &= (I,1,[]), & \mathscr{S}_2(c) &= (C,2,[]) \\
\mathscr{R}_2(i) &= (I,1,[(t,1,i)]), & \mathscr{R}_2(c) &= (C,2,[(t,1,c)]) \\
\mathscr{S}_3(i) &= (I,2,[]), & \mathscr{S}_3(c) &= (C,1,[]) \\
\mathscr{R}_3(i) &= (I,2,[(t,1,i)]), & \mathscr{R}_3(c) &= (C,1,[(t,1,c)]) \\
\mathscr{S}_4(i) &= (I,2,[]), & \mathscr{S}_4(c) &= (C,2,[]) \\
\mathscr{R}_4(i) &= (I,2,[(t,1,i)]), & \mathscr{R}_4(c) &= (C,2,[(t,1,c)])
\end{aligned}
$$



(a) LTS of net of Fig. 6(a)

(b) LTS of net of Fig. 6(b)

Figure 7: Isomorphic LTSs of an MRPN and its SPRN translation

and

$$
\begin{aligned}
\mathscr{S}_1'(i) &= (I_1,1,[]), & \mathscr{S}_1(c) &= (C_1,1,[])\\
\mathscr{R}_1(i) &= (I_1,1,[(t,1,i)]), & \mathscr{R}_1(c) &= (C_1,1,[(t,1,c)])\\
\mathscr{S}_2(i) &= (I_1,1,[]), & \mathscr{S}_2(c) &= (C_2,1,[])\\
\mathscr{R}_2(i) &= (I_1,1,[(t,1,i)]), & \mathscr{R}_2(c) &= (C_2,1,[(t,1,c)])\\
\mathscr{S}_3(i) &= (I_2,1,[]), & \mathscr{S}_3(c) &= (C_1,1,[])\\
\mathscr{R}_3(i) &= (I_2,1,[(t,1,i)]), & \mathscr{R}_3(c) &= (C_1,1,[(t,1,c)])\\
\mathscr{S}_4(i) &= (I_2,1,[]), & \mathscr{S}_4(c) &= (C_2,1,[])\\
\mathscr{R}_4(i) &= (I_2,1,[(t,1,i)]), & \mathscr{R}_4(c) &= (C_2,1,[(t,1,c)])
\end{aligned}
$$

5 Conclusions

This paper presents an extension of RPNs with multiple tokens of the same type based on the individual token interpretation. The individuality of tokens is enabled by recording their causal path, while the semantics allows identical tokens to fire any eligible transition when going forward, but only the transitions they have been previously involved in when going backward. We have presented a semantics for causal-order reversibility, which unlike the semantics presented in [24] is purely local and requires no global control. Another contribution of the paper is a result illustrating that introducing multiple tokens in the model does not increase its expressive power. Indeed, for every MRPN we may construct an equivalent SRPN, which preserves its computation. In related work [30], MRPNs have also been associated with backtracking and out-of-causal-order semantics and it was shown that in all settings MRPNs are equivalent to the original RPN model.

In our current work we are developing a tool for simulating and verifying RPN models [9], which we aim to apply towards the analysis of resource-aware systems. Our experience in applying RPNs in the context of wireless communications [25] has illustrated that resource management can be studied and understood in terms of RPNs since, along with their visual nature, they offer a number of features, such as token persistence, that is especially relevant in these contexts. In future work, we would like to further apply our framework in the specific fields as well as in the field of long-running transactions.

References

[1] Bogdan Aman & Gabriel Ciobanu (2018): *Bonding calculus*. Natural Computing 17(4), pp. 823–832, doi:10.1007/s11047-018-9709-7.

[2] Kamila Barylska, Anna Gogolinska, Lukasz Mikulski, Anna Philippou, Marcin Piatkowski & Kyriaki Psara (2018): *Reversing Computations Modelled by Coloured Petri Nets*. In: Proceedings of ATAED 2018, CEUR Workshop Proceedings 2115, pp. 91–111. Available at `http://ceur-ws.org/Vol-2115/ATAED2018-91-111.pdf`.

[3] Kamila Barylska, Anna Gogolinska, Lukasz Mikulski, Anna Philippou, Marcin Piatkowski & Kyriaki Psara (2022): *Formal Translation from Reversing Petri Nets to Coloured Petri Nets*. In: Proceedings of RC 2022, LNCS 13354, Springer, pp. 172–186, doi:10.1007/978-3-031-09005-9_12.

[4] Kamila Barylska, Maciej Koutny, Lukasz Mikulski & Marcin Piatkowski (2018): *Reversible computation vs. reversibility in Petri nets*. Science of Computer Programming 151, pp. 48–60, doi:10.1016/j.scico.2017.10.008.

[5] Kamila Barylska, Lukasz Mikulski, Marcin Piatkowski, Maciej Koutny & Evgeny Erofeev (2016): *Reversing Transitions in Bounded Petri Nets*. In: Proceedings of CS&P 2016, CEUR Workshop Proceedings 1698, CEUR-WS.org, pp. 74–85.

[6] Adel Benamira (2020): *Causal Reversibility in Individual Token Interpretation of Petri Nets*. The Computer Science Journal 21(4), doi:10.7494/csci.2020.21.4.3728.

[7] Vincent Danos & Jean Krivine (2004): *Reversible Communicating Systems*. In: Proceedings of CONCUR 2004, LNCS 3170, Springer, pp. 292–307, doi:10.1007/978-3-540-28644-8_19.

[8] Vincent Danos & Jean Krivine (2005): *Transactions in RCCS*. In: Proceedings of CONCUR 2005, LNCS 3653, Springer, pp. 398–412, doi:10.1007/11539452_31.

[9] Yannis Dimopoulos, Eleftheria Kouppari, Anna Philippou & Kyriaki Psara (2020): *Encoding Reversing Petri Nets in Answer Set Programming*. In: Proceedings of RC 2020, LNCS 12227, Spinger, pp. 264–271, doi:10.1007/978-3-030-52482-1_17.

[10] Rob J. van Glabbeek (2005): *The Individual and Collective Token Interpretations of Petri Nets*. In: Proceedings of CONCUR 2005, LNCS 3653, Springer, pp. 323–337, doi:10.1007/11539452_26.

[11] Rob J. van Glabbeek, Ursula Goltz & Jens-Wolfhard Schicke (2021): *On Causal Semantics of Petri Nets*. CoRR abs/2103.00729. Available at `https://arxiv.org/abs/2103.00729`.

[12] Rob J. van Glabbeek & Gordon D. Plotkin (1995): *Configuration Structures*. In: Proceedings of LICS 1995, IEEE Computer Society, pp. 199–209, doi:10.1109/LICS.1995.523257.

[13] Jetty Kleijn & Maciej Koutny (2002): *Causality semantics of Petri nets with weighted inhibitor arcs*. In: Proceedings of CONCUR 2002, LNCS 2421, Springer, pp. 531–546, doi:10.1007/3-540-45694-5_35.

[14] Stefan Kuhn, Bogdan Aman, Gabriel Ciobanu, Anna Philippou, Kyriaki Psara & Irek Ulidowski (2020): *Reversibility in Chemical Reactions*. In: Reversible Computation: Extending Horizons of Computing - Selected Results of the COST Action IC1405, LNCS 1270, Spinger, pp. 151–176, doi:10.1007/978-3-030-47361-7_7.

[15] Stefan Kuhn & Irek Ulidowski (2018): *Local reversibility in a Calculus of Covalent Bonding*. Science of Computer Programming 151(Supplement C), pp. 18–47, doi:10.1016/j.scico.2017.09.008.

[16] Ivan Lanese, Claudio Antares Mezzina & Jean-Bernard Stefani (2016): *Reversibility in the higher-order π-calculus*. Theoretical Computer Science 625, pp. 25–84, doi:10.1016/j.tcs.2016.02.019.

[17] Ivan Lanese, Iain C. C. Phillips & Irek Ulidowski (2020): *An Axiomatic Approach to Reversible Computation*. In: Proceedings of FOSSACS 2020, LNCS 12077 12077, Springer, pp. 442–461, doi:10.1007/978-3-030-45231-5_23.

[18] Y. Edmund Lien (1976): *A note on transition systems*. Information Sciences 10(4), pp. 347–362, doi:10.1016/0020-0255(76)90054-2.

[19] Hernán C. Melgratti, Claudio Antares Mezzina, Iain Phillips, G. Michele Pinna & Irek Ulidowski (2020): *Reversible Occurrence Nets and Causal Reversible Prime Event Structures*. In: *Proceedings of RC 2020*, LNCS 12227, Spinger, pp. 19–36, doi:10.1007/978-3-030-52482-1_2.

[20] Hernán C. Melgratti, Claudio Antares Mezzina & Irek Ulidowski (2020): *Reversing Place Transition Nets*. *Logical Methods in Computer Science* 16(4). Available at `https://lmcs.episciences.org/6843`.

[21] Lukasz Mikulski & Ivan Lanese (2019): *Reversing Unbounded Petri Nets*. In: *Proceedings of PETRI NETS 2019*, LNCS 11522, Springer, pp. 213–233, doi:10.1007/978-3-030-21571-2_13.

[22] Anna Philippou & Kyriaki Psara (2018): *Reversible Computation in Petri Nets*. In: *Proceedings of RC 2018*, LNCS 11106, Springer, pp. 84–101, doi:10.1007/978-3-319-99498-7_6.

[23] Anna Philippou & Kyriaki Psara (2022): *A collective interpretation semantics for reversing Petri nets*. *Theoretical Computer Science* 924, pp. 148–170, doi:10.1016/j.tcs.2022.05.016.

[24] Anna Philippou & Kyriaki Psara (2022): *Reversible computation in nets with bonds*. *Journal of Logical and Algebraic Methods in Programming* 124, p. 100718, doi:10.1016/j.jlamp.2021.100718.

[25] Anna Philippou, Kyriaki Psara & Harun Siljak (2019): *Controlling Reversibility in Reversing Petri Nets with Application to Wireless Communications*. In: *Proceedings of RC 2019*, LNCS 11497, Springer, pp. 238–245, doi:10.1007/978-3-030-21500-2_15.

[26] Iain Phillips & Irek Ulidowski (2007): *Reversing algebraic process calculi*. *Journal of Logic and Algebraic Programming* 73(1-2), pp. 70–96, doi:10.1016/j.jlap.2006.11.002.

[27] Iain Phillips & Irek Ulidowski (2015): *Reversibility and asymmetric conflict in event structures*. *Journal of Logical and Algebraic Methods in Programming* 84(6), pp. 781–805, doi:10.1016/j.jlamp.2015.07.004.

[28] Iain Phillips, Irek Ulidowski & Shoji Yuen (2012): *A Reversible Process Calculus and the Modelling of the ERK Signalling Pathway*. In: *Proceedings of RC 2012*, LNCS 7581, Springer, pp. 218–232, doi:10.1007/978-3-642-36315-3_18.

[29] Iain Phillips, Irek Ulidowski & Shoji Yuen (2013): *Modelling of Bonding with Processes and Events*. In: *Proceedings of RC 2013*, LNCS 7947, Springer, pp. 141–154, doi:10.1007/978-3-642-38986-3_12.

[30] Kyriaki Psara (2020): *Reversible Computation in Petri Nets*. Ph.D. thesis, Department of Computer Science, University of Cyprus. Available at `https://arxiv.org/abs/2101.07066`.

[31] Wolfgang Reisig (1985): *Petri Nets with Individual Tokens*. *Theoretical Computer Science* 41, pp. 185–213, doi:10.1016/0304-3975(85)90070-2.

[32] Grzegorz Rozenberg & Joost Engelfriet (1996): *Elementary net systems*. In: *Advanced Course on Petri Nets*, LNCS 1491, Springer, pp. 12–121, doi:10.1007/3-540-65306-6_14.

[33] Irek Ulidowski, Iain Phillips & Shoji Yuen (2014): *Concurrency and Reversibility*. In: *Proceedings of RC 2014*, LNCS 8507, Springer, pp. 1–14, doi:10.1007/978-3-319-08494-7_1.