Reversing P/T Nets

Hernán Melgratti[1](✉), Claudio Antares Mezzina[2,3](✉), and Irek Ulidowski[2](✉)

[1] University of Buenos Aires - Conicet, Buenos Aires, Argentina
hmelgra@dc.uba.ar
[2] University of Leicester, Leicester, England
iu3@leicester.ac.uk
[3] Dipartimento di Scienze Pure e Applicate, Università di Urbino, Urbino, Italy
claudio.mezzina@uniurb.it

Abstract. Petri Nets are a well-known model of concurrency and provide an ideal setting for the study of fundamental aspects in concurrent systems. Despite their simplicity, they still lack a satisfactory causally reversible semantics. We develop such semantics for Place/Transitions Petri Nets (P/T nets) based on two observations. Firstly, a net that explicitly expresses causality and conflict among events, e.g., an occurrence net, can be straightforwardly reversed by adding reversal for each of its transitions. Secondly, the standard unfolding construction associates a P/T net with an occurrence net that preserves all of its computation. Consequently, the reversible semantics of a P/T net can be obtained as the reversible semantics of its unfolding. We show that such reversible behaviour can be expressed as a finite net whose tokens are coloured by causal histories. Colours in our encoding resemble the causal memories that are typical in reversible process calculi.

1 Introduction

Reversible computing is attracting interest for its applications in many fields including hardware design and quantum computing [30], the modelling of biochemical reactions [12,25,26], parallel discrete event simulation [27] and program reversing for debugging [8,11,16].

A model for reversible computation features two computation flows: the standard forward direction and the reverse one, which allows to reach back any past state of the computation. Reversibility is well understood in a sequential setting in which executions are totally ordered sets of events (see [17]): a sequential computation can be reversed by successively undoing the last not yet undone event. Reversibility becomes more challenging in a concurrent setting because there is no natural way for totally ordering events. Often concurrency models account for the causal dependencies among events, which are reflected as a partial order. Reversing an execution consisting of a partially ordered set of events reduces to successively undoing one of the maximal events not yet undone. This is at the basis of the *causally-consistent reversibility* [6,15,23], which relates reversibility

© IFIP International Federation for Information Processing 2019
Published by Springer Nature Switzerland AG 2019
H. Riis Nielson and E. Tuosto (Eds.): COORDINATION 2019, LNCS 11533, pp. 19–36, 2019.
https://doi.org/10.1007/978-3-030-22397-7_2

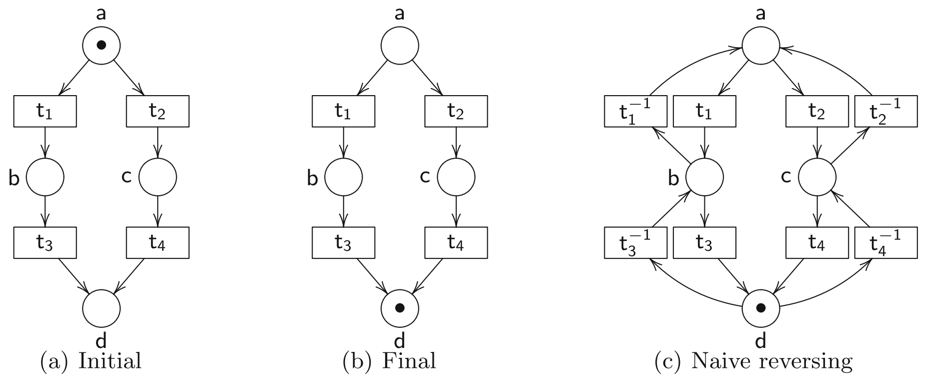


Fig. 1. Backward conflict and naive reversing.

with causality. Intuitively, this notion stipulates that any event can be undone provided that all its consequences, if any, are undone beforehand. Reversibility in distributed systems such as in checkpoint/rollback protocols [29] and in transactions [7,13] can be modelled by causal-consistent reversibility. The interplay between reversibility and concurrency has been widely studied in process calculi [4,6,14,19,23], event structures [5,9,24,28] and lately Petri Nets [1,21]. Despite being a very basic model of concurrency, Petri nets still lack a satisfactory causally-consistent reversible semantics. For instance, no current models are able to handle cyclic nets.

A key point when reversing computation in Petri nets is to handle backward conflicts, i.e., the fact that a token can be generated in a place because of different causes. Consider the net in Fig. 1(a) showing the initial state of a system that can either perform t_1 followed by t_3, or t_2 followed by t_4. The final state of a complete computation is depicted in Fig. 1(b). The information in that state is not enough to deduce whether the token in d has been produced because of t_3 or t_4. Even worse, if we "naively" reverse the net by just adding transitions in the reverse direction, as shown in Fig. 1(c), the reverse transition will do more than undoing the computation. In fact, the token in d can be put back either in b or c regardless of the previous computation.

Analogous problems arise when a net is cyclic. Previous approaches [1,21] to reversing Petri nets tackle backward conflicts by relying on a new kind of tokens, called *bonds* that keep track of the execution history. Bonds are rich enough for allowing other approaches to reversibility, such as *out-of-order* reversibility [12], but they cannot cope with cyclic nets. We propose here a reversible model for P/T nets that can handle cyclic nets by relying on standard notions in Petri net theory. We first observe that a Petri Net can be mapped via the standard unfolding construction to an occurrence net, i.e., an acyclic net that does not have backward conflicts and makes causal dependencies explicit. Then, an occurrence net can be "simply" reversed by reversing each of its transitions. Such construction gives a model that features causally-consistent reversibility.

This is shown by proving that each reachable marking in the reversible version of the occurrence net is a marking that can be reached by just forward computational steps. We observe that the unfolding construction could produce an infinite occurrence net. However, the unfolding can be seen as the definition of a coloured net, where colours account for causal histories. Such interpretation associates a P/T net with an equivalent coloured P/T net, which can be reversed in the "simple" way. The correctness of the construction is shown by exhibiting a one-to-one correspondence of its executions with the ones of the reversible version of the unfolding. Interestingly, the colours used by the construction resemble the memories common in reversible calculi [6,14].

We remark that our proposal deals with reversing (undoing) computation in a Petri net and not with the classical problem of reversibility [3] which requires every computation to be able to reach back the initial state of the system (but not necessary by undoing the previous events). In this sense, the problem of making a net reversible equates to adding a minimal amount of transitions that make a net reversible [2]. Reversibility is a global property while reversing a computation is a local one, as discussed in [2].

2 Background

2.1 Petri Nets

Petri nets are built up from *places* (denoting, e.g., resources and message types), which are repositories for *tokens* (representing instances of resources), and *transitions*, which fetch and produce tokens. We consider the infinite sets $\mathcal{P}$ of places and $\mathcal{T}$ of transitions, and assume that they are disjoint, i.e., $\mathcal{P} \cap \mathcal{T} = \emptyset$. We let $\mathsf{a}, \mathsf{a}', \ldots$ range over $\mathcal{P}$ and $\mathsf{t}, \mathsf{t}', \ldots$ over $\mathcal{T}$. We write $x, y, \ldots$ for elements in $\mathcal{P} \cup \mathcal{T}$.

A *multiset* over a set S is a function $m : S \to \mathbb{N}$ (where $\mathbb{N}$ denotes the natural numbers including zero). We write $\mathbb{N}^S$ for the set of multisets over S. For $m \in \mathbb{N}^S$, $\mathit{supp}(m) = \{x \in S \mid m(x) > 0\}$ is the *support* of m, and $|m| = \sum_{x \in S} m(x)$ stands for its *cardinality*. We write $\emptyset$ for the empty multiset, i.e., $\mathit{supp}(\emptyset) = \emptyset$. The union of $m_1, m_2 \in \mathbb{N}^S$, written $(m_1 \oplus m_2)$, is defined such that $(m_1 \oplus m_2)(x) = m_1(x) + m_2(x)$ for all $x \in S$. Note that $\oplus$ is associative and commutative, and has $\emptyset$ as identity. Hence, $\mathbb{N}^S$ is the free commutative monoid $S^\oplus$ over S. We write x for a singleton multiset, i.e., $\mathit{supp}(x) = \{x\}$ and $m(x) = 1$. Moreover, we write $x_1 \ldots x_n$ for $x_1 \oplus \ldots \oplus x_n$. Let $f : S \to S'$, we write f also for its obvious extension to multisets, i.e., $f(x_0 \ldots x_n) = f(x_0) \ldots f(x_n)$. We avoid writing $\mathit{supp}(\_)$ when applying set operators to multisets, e.g., we write $x \in m$ or $m_1 \cap m_2$ instead of $x \in \mathit{supp}(m)$ or $\mathit{supp}(m_1) \cap \mathit{supp}(m_2)$.

Definition 1 (Petri Net). *A* net N *is a 4-tuple* $N = (S_N, T_N, {}^\bullet\_{}_N, \_{}^\bullet_N)$ *where* $S_N \subseteq \mathcal{P}$ *is the (nonempty) set of places,* $T_N \subseteq \mathcal{T}$ *is the set of transitions and the functions* ${}^\bullet\_{}_N, \_{}^\bullet_N : T_N \to 2^{S_N}$ *assign source and target to each transition such that* ${}^\bullet\mathsf{t} \neq \emptyset$ *and* $\mathsf{t}^\bullet \neq \emptyset$ *for all* $\mathsf{t} \in T_N$*. A marking of a net* N *is a multiset over* S_N*, i.e.,* $m \in \mathbb{N}^S$*. A Petri net is a pair* (N, m) *where* N *is a net and* m *is a marking of* N*.*

We denote $S_N \cup T_N$ by N, and omit the subscript N if no confusion arises. We abbreviate a transition $\mathsf{t} \in T$ with *preset* ${}^\bullet t = s_1$ and *postset* $t^\bullet = s_2$ as $s_1[\rangle s_2$. Hereafter, we only consider nets whose transitions have non-empty presets. The pre and postset of a place $\mathsf{a} \in S$ are defined respectively as ${}^\bullet\mathsf{a} = \{\mathsf{t} \mid \mathsf{a} \in \mathsf{t}^\bullet\}$ and $\mathsf{a}^\bullet = \{\mathsf{t} \mid \mathsf{a} \in {}^\bullet\mathsf{t}\}$. We let ${}^\circ N = \{x \in N \mid {}^\bullet x = \emptyset\}$ and $N^\circ = \{x \in N \mid x^\bullet = \emptyset\}$ denote the sets of *initial* and *final elements* of N respectively. Note that we only consider nets whose initial and final elements are places since transitions have non-empty pre and postsets, i.e., ${}^\bullet\mathsf{t} \neq \emptyset$ and $\mathsf{t}^\bullet \neq \emptyset$ holds for all t.

Definition 2 (Net morphisms). *Let N, N' be nets. A pair $f = (f_S : S_N \to S_{N'}, f_T : T_N \to T_{N'})$ is a* net morphism *from N to N' (written $f : N \to N'$) if $f_S({}^\bullet\mathsf{t}_N) = {}^\bullet(f_T(\mathsf{t}))_{N'}$ and $f_S(\mathsf{t}^\bullet_N) = (f_T(\mathsf{t}))^\bullet_{N'}$ for any t. Moreover, we say N and N' are* isomorphic *if f is bijective.*

The operational (interleaving) semantics of a Petri net is given by the least relation on Petri nets satisfying the following inference rule:

$$\text{(FIRING)}\quad \frac{\mathsf{t} = m\ [\rangle\ m' \in T_N}{(N, m \oplus m'') \xrightarrow{\mathsf{t}} (N, m' \oplus m'')}$$

which describes the evolution of the state of a net (represented by the marking $m \oplus m''$) by the firing of a transition $m[\rangle m'$ that consumes the tokens m in its preset and produces the tokens m' in its postset. We sometimes omit t in $\xrightarrow{\mathsf{t}}$ when the fired transition is uninteresting.

According to Definition 1, transitions consume and produce at most one token in each place. On the other hand, P/T nets below fetch and consume multiple tokens by defining the pre- and postsets of transitions as multisets.

Definition 3 (P/T net). *A* Place/Transition Petri net *(P/T net) is a 4-tuple $N = (S_N, T_N, {}^\bullet\_{}_N, \_{}^\bullet_N)$ where $S_N \subseteq \mathcal{P}$ is the (nonempty) set of places, $T_N \subseteq \mathcal{T}$ is the set of transitions and the functions ${}^\bullet\_{}_N, \_{}^\bullet_N : T_N \to \mathbb{N}^{S_N}$ assign source and target to each transition.A marking of a net N is multiset over S_N, i.e., $m \in \mathbb{N}^S$. A marked P/T net is a pair (N, m) where N is a P/T net and m is a marking of N.*

The notions of pre- and postset, initial and final elements, morphisms and operational semantics are straightforwardly extended to P/T nets. Note that Petri nets can be regarded as a P/T net whose arcs have unary weights.

Next, we introduce some notation for sequences of transitions. Let ';' denote concatenation of such sequences. For the sequence $s = \mathsf{t}_1; \mathsf{t}_2; \ldots; \mathsf{t}_n$, we write $(N, m_0) \xrightarrow{s} (N, m_n)$ if $(N, m_0) \xrightarrow{\mathsf{t}_1} (N, m_1) \xrightarrow{\mathsf{t}_2} \ldots \xrightarrow{\mathsf{t}_n} (N, m_n)$; we call s a firing sequence. We write $(N, m_0) \to^* (N, m_n)$ if there exists s such that $(N, m_0) \xrightarrow{s} (N, m_n)$, and ϵ_m for the empty sequence.

Definition 4. *Let (N, m) be a P/T net. The set of* reachable markings *$reach(N, m)$ is defined as $\big\{m' \mid (N, m) \to^* \big(N, m'\big)\big\}$.*

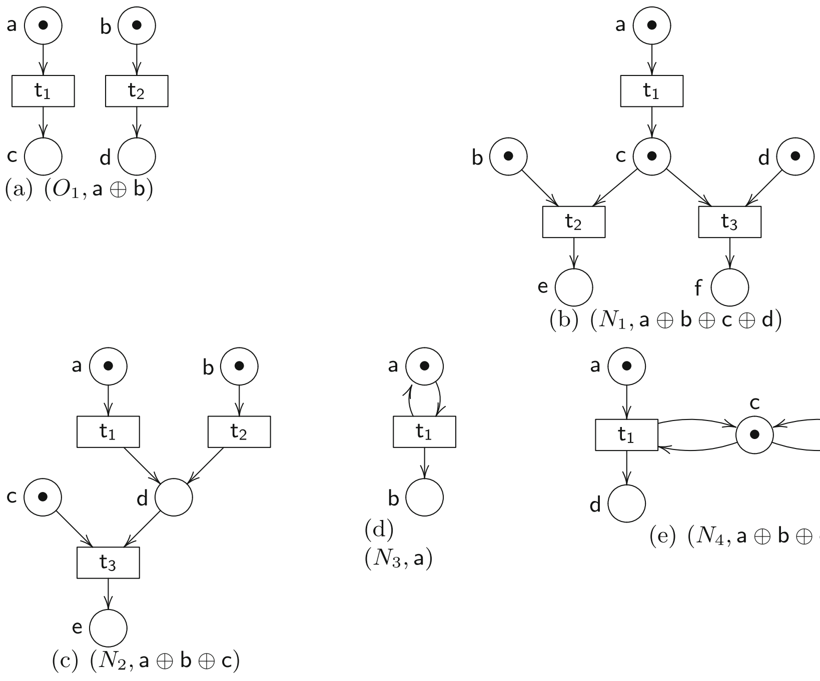

a
b
t1
t2
c
d
(a) $(O_1, \mathsf{a} \oplus \mathsf{b})$
a
t1
b
c
d
t2
t3
e
f
(b) $(N_1, \mathsf{a} \oplus \mathsf{b} \oplus \mathsf{c} \oplus \mathsf{d})$
a
b
t1
t2
c
d
t3
e
(c) $(N_2, \mathsf{a} \oplus \mathsf{b} \oplus \mathsf{c})$
a
t1
b
(d) (N_3, a)
a
b
c
t1
t2
d
e
(e) $(N_4, \mathsf{a} \oplus \mathsf{b} \oplus \mathsf{c})$

Fig. 2. P/T nets

We say a marked P/T net (N, m) is *(1-)safe* if every reachable marking is a set, i.e., $m' \in \mathit{reach}(N, m)$ implies $m' \in 2^{S_N}$.

Example 5. Figure 2 shows different P/T nets, which will be used throughout the paper. As usual, places and transitions are represented by circles and boxes, respectively. The nets O_1 and N_4 are Petri nets, and N_1, N_2 and N_3 are P/T nets which, when executing, may produce multiple tokens in some places.

2.2 Unfolding of P/T Nets

Our approach to reversing Petri nets relies on their occurrence net semantics, which explicitly exhibit the causal ordering, concurrency, and conflicts among events. We start by introducing several useful notions and notations. First, we shall describe a flow of causal dependencies in a net with the relation $\prec$:

Definition 6. *Let* $\prec$ *be* $\{(\mathsf{a}, \mathsf{t}) | \mathsf{a} \in S_N \wedge \mathsf{t} \in \mathsf{a}^\bullet\} \cup \{(\mathsf{t}, \mathsf{a}) | \mathsf{a} \in S_N \wedge \mathsf{t} \in {}^\bullet\mathsf{a}\}$. *We write* $\preceq$ *for the reflexive and transitive closure of* $\prec$.

Consider Fig. 2. We have $\mathsf{a} \prec \mathsf{t}_1$ and $\mathsf{t}_1 \prec c$ in O_1 as well as $\mathsf{t}_1 \preceq \mathsf{t}_2$ in N_1.

Two transitions t_1 and t_2 are in an *immediate conflict*, written $\mathsf{t}_1 \#_0 \mathsf{t}_2$, when $\mathsf{t}_1 \neq \mathsf{t}_2$ and ${}^\bullet\mathsf{t}_1 \cap {}^\bullet\mathsf{t}_2 \neq \emptyset$. For example, t_1 and t_2 in N_4 in Fig. 2 are in an immediate conflict since they share a token in the place c. Correspondingly, for

t_2 and t_3 in N_1. The *conflict* relation $\#$ is defined by letting $x\#y$ if $x \neq y$ and there are $\mathsf{t}_1, \mathsf{t}_2 \in T$ such that $\mathsf{t}_1 \preceq x$, and $\mathsf{t}_2 \preceq y$, and $\mathsf{t}_1 \#_0 \mathsf{t}_2$.

We are now ready to give the definition of an occurrence net following [10,20].

Definition 7 (Occurrence net). *A net* (N, m) *is an* occurrence net *if*

1. *N is* acyclic*;*
2. *N is a* (1-)safe *net, i.e, any reachable marking is a set;*
3. *$m = {}^{\circ}N$, i.e., the initial marking is identified with the set of initial places;*
4. *there are no* backward conflicts*, i.e.,* $|{}^{\bullet}\mathsf{a}| \leq 1$ *for all a in* S_N*;*
5. *there are no* self-conflicts*, i.e,* $\neg(\mathsf{t}\#\mathsf{t})$ *for all* t *in* T_N*.*

We use O to range over occurrence nets.

Example 8. The net O_1 in Fig. 2 is an occurrence net, while the remaining nets are not. N_1 is not an occurrence net since there is a token in place c and c is not an initial place of the net. N_2 has a backward conflict since two transitions produce tokens on the place d. N_3 is cyclic, and N_4 is cyclic and has a backward conflict on c.

The absence of backward conflicts in occurrence nets ensures that each place appears in the postset of at most one transition. Hence, pre- and postset relations can be interpreted as a causal dependency. So, $\preceq$ represents causality.

We say $x, y \in N$ are *concurrent*, written $x \; co \; y$, if $x \neq y$ and $x \not\preceq y$, $y \not\preceq x$, and $\neg x\#y$. A set $X \subseteq N$ is concurrent, written $CO(X)$, if $\forall x, y \in X : x \neq y \Rightarrow x \; co \; y$, and $|\{\mathsf{t} \in T_N \mid \exists x \in X, \mathsf{t} \preceq x\}|$ is finite. For example, the set $\{\mathsf{t}_1, \mathsf{t}_2\}$ of firings in O_1 of Fig. 2 is concurrent, so we can write $CO(\{\mathsf{t}_1, \mathsf{t}_2\})$.

Two transitions are *coinitial* if they start with the same marking, and *cofinal* if they end up in the same marking. We now have a simple version of the Square Lemma [6] for forward concurrent transitions. It will be helpful in proving our Lemma 16 in the next section.

Lemma 9. *Let* t *and* t' *be coinitial concurrent transitions. Then, there exist transitions* t_1 *and* t_1' *such that* $\mathsf{t}; \mathsf{t}_1'$ *and* $\mathsf{t}'; \mathsf{t}_1$ *are cofinal.*

The lemma says that if transitions t and t' originate from one corner of a square, and if they represent independent (concurrent) events, then the square completes with two other independent transitions (t_1 and t_1') meeting at the opposite corner of the square. The order in which concurrent transitions are executed in a firing sequence does not matter. Indeed, the order which should be preserved among firings in a sequence is the causal order. We then consider sequences equivalent up to the swapping of concurrent transitions. This corresponds to considering the set of Mazurkiewicz traces induced by *co* as the independence relation.

Formally, trace equivalence $\equiv$ is the least congruence over firing sequences s such that $\forall \mathsf{t}_1, \mathsf{t}_2 : \mathsf{t}_1 \; co \; \mathsf{t}_2 \implies \mathsf{t}_1; \mathsf{t}_2 \equiv \mathsf{t}_2; \mathsf{t}_1$. The equivalence classes of $\equiv$ are the (Mazurkiewicz) *traces*. We use ω to range over such traces. We also will use ϵ for the empty trace, and ; for the concatenation operator.

$$\frac{m(\mathsf{a}) = n}{\{\mathsf{a}(\emptyset, i) \mid 1 \leq i \leq n\} \subseteq S}\ (\text{INI-MK})$$

$$\frac{H = \{\mathsf{a}_j(h_j, i_j) \mid j \in J\} \subseteq S \qquad Co(H) \qquad \mathsf{t} \in T_N \quad {}^\bullet\mathsf{t}_N = \oplus_{j \in J}\mathsf{a}_j}{\mathsf{t}(H) \in T, \quad {}^\bullet(\mathsf{t}(H)) = H}\ (\text{PRE})$$

$$\frac{x = \mathsf{t}(H) \in T}{Q = \{\mathsf{a}(\{x\}, i) \mid 1 \leq i \leq \mathsf{t}^\bullet_N(\mathsf{a})\} \subseteq S, \quad x^\bullet = Q}\ (\text{POST})$$

Fig. 3. Unfolding rules.

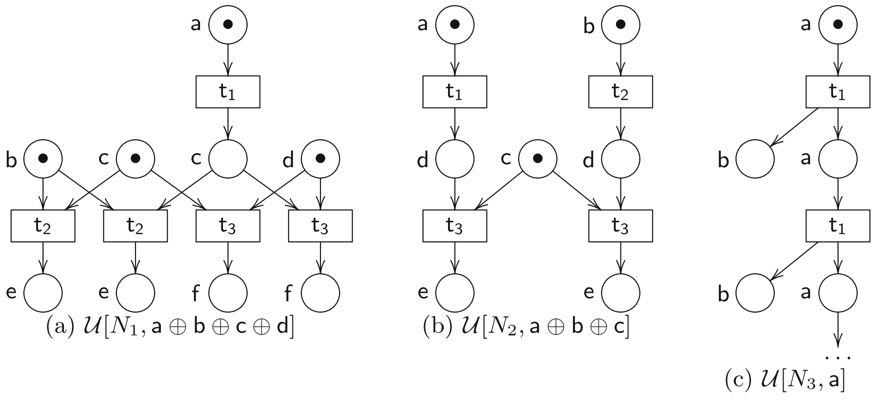

a
t₁
b
c
c
d
t₂
t₂
t₃
t₃
e
e
f
f
(a) $\mathcal{U}[N_1, \mathsf{a} \oplus \mathsf{b} \oplus \mathsf{c} \oplus \mathsf{d}]$
a
b
t₁
t₂
d
c
d
t₃
t₃
e
e
(b) $\mathcal{U}[N_2, \mathsf{a} \oplus \mathsf{b} \oplus \mathsf{c}]$
a
t₁
b
a
t₁
b
a
...
(c) $\mathcal{U}[N_3, \mathsf{a}]$

Fig. 4. Unfoldings of P/T nets

For occurrence nets we have this standard property:

$$s_1 \equiv s_2 \ \textit{iff} \ (O, m_0) \xrightarrow{s_1} (O, m_n) \iff (O, m_0) \xrightarrow{s_2} (O, m_n) \tag{1}$$

Two traces are *coinitial* if they start with the same marking, and *cofinal* if they end up in the same marking. Hence, Eq. (1) tells us that two traces that are *coinitial* and *cofinal* are then trace equivalent.

The unfolding of a net N is the least occurrence net that can account for all the possible computations of N and makes explicit causal dependencies, conflicts and concurrency between firings [20].

Definition 10 (Unfolding). *Let (N, m) be a P/T net. The unfolding of N is the occurrence net $\mathcal{U}[N, m] = (S, T, \delta_0, \delta_1)$ generated inductively by the inference rules in Fig. 3 and the folding morphism $(f_S, f_T) : \mathcal{U}[N, m] \to N$ defined such that $f_S(\mathsf{a}, \_, \_) = \mathsf{a}$ and $f_T(\mathsf{t}, \_) = \mathsf{t}$.*

Places are named by triples $\mathsf{a}(H, i)$ where: a is a place of N where tokens reside; H is the set of immediate causes (i.e., the history of tokens); and i is a positive integer used to disambiguate tokens with the same history. Transitions (or events) are encoded as $\mathsf{t}(H)$, where H is as above and t is the fired transition.

Example 11. The unfoldings of the nets $(N_1, \mathsf{a} \oplus \mathsf{b} \oplus \mathsf{c} \oplus \mathsf{d})$, $(N_2, \mathsf{a} \oplus \mathsf{b} \oplus \mathsf{c})$ and (N_3, a) in Fig. 2 are shown in Fig. 4. Note that since O_1 is an occurrence net its unfolding is isomorphic to O_1, thus it is omitted. Consider the occurrence net $\mathcal{U}[N_1, \mathsf{a} \oplus \mathsf{b} \oplus \mathsf{c} \oplus \mathsf{d}]$. The leftmost transition t_2 is different from the other transition t_2 since they have different histories: the leftmost t_2 is caused by the tokens in b and c (which are available in the initial marking), whereas the other t_2 is caused only by the token in b and the token that is produced by the firing of t_1. Correspondingly, for the two transitions labelled t_3. Consider $\mathcal{U}[N_2, \mathsf{a} \oplus \mathsf{b} \oplus \mathsf{c}]$. After the transitions t_1 and t_2 have fired, there is a token in each of the places labelled d. The token in the leftmost d has the history t_1 and the token in the other d has the history t_2. Once t_3 has fired, we can tell the copies of t_3 apart by inspecting their histories: the leftmost t_3 is caused by a token in d with the history t_1 (as well as the token in c), whereas the other t_3 is caused by d with the history t_2 and by c.

3 Reversing Occurrence Nets

Definition 12. *Let O be an occurrence net. The reversible version of O is $\overleftarrow{O} = (S_{\overleftarrow{O}}, T_{\overleftarrow{O}}, {}^{\bullet}\_{}_{\overleftarrow{O}}, \_{}^{\bullet}_{\overleftarrow{O}})$ defined such that*

$$S_{\overleftarrow{O}} = S_O \qquad\qquad T_{\overleftarrow{O}} = T_O \cup \{\, \overleftarrow{\mathsf{t}} \mid \mathsf{t} \in T_O \}$$

$$^{\bullet}\mathsf{t}_{\overleftarrow{O}} = \begin{cases} ^{\bullet}\mathsf{t}_O & \text{if } \mathsf{t} \in T_O \\ \mathsf{t}^{\bullet}_O & \text{otherwise} \end{cases} \qquad\qquad \mathsf{t}^{\bullet}_{\overleftarrow{O}} = \begin{cases} \mathsf{t}^{\bullet}_O & \text{if } \mathsf{t} \in T_O \\ ^{\bullet}\mathsf{t}_O & \text{otherwise} \end{cases}$$

Given a transition t we write $\overleftarrow{\mathsf{t}}$ for a transition that reverses t. We shall call transitions like $\overleftarrow{\mathsf{t}_1}$ and $\overleftarrow{\mathsf{t}_2}$ in Fig. 5 *reverse* (or backwards) transitions (or firings), and use t, t_1 and t_2 to denote transitions or reverse transitions.

For $\overleftarrow{O}$, we write $(\overleftarrow{O}, m) \xrightarrow{\mathsf{t}} (\overleftarrow{O}, m')$ for a forward firing when $\mathsf{t} \in T_O$, and $(\overleftarrow{O}, m) \overset{\mathsf{t}}{\rightsquigarrow} (\overleftarrow{O}, m')$ for the reverse (or backward) firing when $\mathsf{t} \notin T_O$. We also let $\xrightarrow{\mathsf{t}}$ be $\overset{\mathsf{t}}{\twoheadrightarrow} \cup \overset{\mathsf{t}}{\rightsquigarrow}$. We will often refer to a firing $(\overleftarrow{O}, m) \xrightarrow{\mathsf{t}} (\overleftarrow{O}, m')$ as t. Given a firing t we indicate with $\overleftarrow{\mathsf{t}}$ its inverse that is

$$(\overleftarrow{O}, m) \overset{\overleftarrow{\mathsf{t}}}{\twoheadrightarrow} (\overleftarrow{O}, m') \quad \text{if} \quad (\overleftarrow{O}, m') \overset{\mathsf{t}}{\rightsquigarrow} (\overleftarrow{O}, m)$$

$$(\overleftarrow{O}, m) \overset{\overleftarrow{\mathsf{t}}}{\rightsquigarrow} (\overleftarrow{O}, m') \quad \text{if} \quad (\overleftarrow{O}, m') \overset{\mathsf{t}}{\twoheadrightarrow} (\overleftarrow{O}, m)$$

Hence, we have $\overleftarrow{\overleftarrow{\mathsf{t}}} = \mathsf{t}$. We shall work with sequences of transitions and reverse transitions, ranged over by s, s_1 and s_2. We say that a sequence is a *forward* (resp. *backward*) *sequence* when all its firings are forward (resp. backward).

Next, we extend the notions of causality, conflict and concurrency to transitions and reverse transitions in reverse versions of occurrence nets. We extend

$\prec$ in Definition 6 to cover reverse transitions in an obvious way using Definition 12. As a result, we obtain $\mathsf{t} \preceq \overleftarrow{\mathsf{t}}$ and $\overleftarrow{\mathsf{t}} \preceq \mathsf{t}$. As for the conflict relation, we define an immediate conflict between different $\overleftarrow{\mathsf{t}_1}$ and $\overleftarrow{\mathsf{t}_2}$ as ${}^\bullet\overleftarrow{\mathsf{t}_1} \cap {}^\bullet\overleftarrow{\mathsf{t}_2} \neq \emptyset$. This is $\mathsf{t}_1{}^\bullet \cap \mathsf{t}_2{}^\bullet \neq \emptyset$, meaning t_1 and t_2 are in backward conflict, which is ruled out in occurrence nets. Hence, the immediate conflict relation is empty between reverse transitions, and so is the conflict relation. The immediate conflict relation between t and $\overleftarrow{\mathsf{t}'}$ is defined as ${}^\bullet\mathsf{t} \cap {}^\bullet\overleftarrow{\mathsf{t}'} \neq \emptyset$. This is equivalent to ${}^\bullet\mathsf{t} \cap \mathsf{t}'^\bullet \neq \emptyset$, which means $\mathsf{t}' \preceq \mathsf{t}$. Consequently, the conflict relation on transitions in $\overleftarrow{O}$ is given by the conflict relation on the forward transitions, and can be defined using the causality relation for pairs of a transition and reverse transition. This allows us to define concurrent transitions in $\overleftarrow{O}$. We say $t \; co \; t'$ if (a) $t \; co \; t'$ for $t, t' \in T_O$, (b) $t \not\preceq t'$ and $t' \not\preceq t$ if t, t' are reverse transitions, and (c) $t \not\preceq t', t' \not\preceq t$ and $\overleftarrow{t'} \not\preceq t$ if t is a transition and t' is a reverse transition.

Next, we show that $\overleftarrow{O}$ is a conservative extension of O.

Lemma 13. $(O, m) \xrightarrow{\mathsf{t}} (O, m')$ *iff* $(\overleftarrow{O}, m) \overset{\mathsf{t}}{\twoheadrightarrow} (\overleftarrow{O}, m')$.

In general, a reversible occurrence net is not an occurrence net. This is because adding reverse transitions may introduce backward conflict for these transitions. Consider N_1 in Fig. 2. We notice that initially t_1 and t_2 are in conflict. Then, in $\overleftarrow{N}_1$ in Fig. 5, the place c with a token has two reverse transitions in its preset, namely $\overleftarrow{\mathsf{t}_2}$ and $\overleftarrow{\mathsf{t}_3}$, hence there is a backward conflict.

4 Properties

We now study the properties of the reversible versions of occurrence nets.

An important property of a *fully* reversible system is the Loop Lemma stating that any reduction can be undone. Formally:

Lemma 14 (Loop Lemma). $(\overleftarrow{O}, m) \overset{\mathsf{t}}{\twoheadrightarrow} (\overleftarrow{O}, m')$ *iff* $(\overleftarrow{O}, m') \overset{\overleftarrow{\mathsf{t}}}{\rightsquigarrow} (\overleftarrow{O}, m)$.

We can generalise the result of the Loop Lemma to sequences as follows:

Corollary 15. $(\overleftarrow{O}, m) \rightarrow^* (\overleftarrow{O}, m')$ *iff* $(\overleftarrow{O}, m') \rightarrow^* (\overleftarrow{O}, m)$.

Next, we have a lemma which is instrumental for the proof of causal-consistent reversibility in reversible calculi [6,14]. Note that t and t' can be either forward or reverse transitions.

Lemma 16 (Square Lemma). *Let t and t' be coinitial concurrent transitions. Then, there exist transitions t_1 and t'_1 such that $t; t'_1$ and $t'; t_1$ are cofinal.*

In order to prove causal consistency we first define a notion of equivalence on sequences of transitions and reverse transitions in reversible occurrence nets. By following Lévy's approach [18], we define the notion of *reverse equivalence* on such sequences as the least equivalence relation $\asymp$ which is closed under composition with ; such that the following hold (recall that t, t' are transitions or reverse transitions):

$$t; t' \asymp t'; t \quad \text{if } t \; co \; t' \qquad \mathsf{t}; \overleftarrow{\mathsf{t}} \asymp \epsilon \qquad \overleftarrow{\mathsf{t}}; \mathsf{t} \asymp \epsilon$$

Reversible equivalence $\asymp$ allows us to swap the order of t and t' in an execution sequence as long as t, t' are concurrent. Moreover, it allows cancellation of a transition and its inverse. We have that $\equiv \subset \asymp$. The equivalence classes of $\asymp$ are called *traces*; it is clear that they contain the Mazurkiewicz traces. Hence, we shall use ω, ω_1 and ω_2 to range over such traces.

The following lemma says that, up to reverse equivalence, one can always reach for the maximum freedom of choice, going backward, and only then going forwards.

Lemma 17 (Parabolic Lemma). *Let ω be a trace. There exist two forward traces ω_1 and ω_2 such that $\omega \asymp \overleftarrow{\omega_1}; \omega_2$.*

Proof. By lexicographic induction on length of ω and on the distance between the beginning of ω and the earliest pair of opposing firings in ω. The analysis uses both the Loop Lemma (Lemma 14) and the Square Lemma (Lemma 16).

The following lemma says that, if two traces ω_1 and ω_2 are coinitial and cofinal (e.g. they start from the same marking and end in the same marking) and ω_2 is a forward only trace, then ω_1 has some forward firings and their reverse ones that cancel each other. And this implies that ω_1 is causally equivalent to a forward trace in which all those pairs of fairing are cancelled out.

Lemma 18 (Shortening Lemma). *Let $\omega_1 \asymp \omega_2$ with ω_2 forward. Then, $|\omega_2| \leq |\omega_1|$.*

Proof. The proof is by induction on length of ω_1, using Lemma 16 and Lemma 17. In the proof, the forward trace ω_2 is the main guideline for shortening ω_1 into a forward trace. Indeed, the proof relies crucially on the fact that ω_1 and ω_2 share the same source and target and that ω_2 is a forward trace.

Theorem 19 (Causal Consistency). *Two traces ω_1 and ω_2 are reversible equivalent iff they are coinitial and cofinal, namely*

$$\omega_1 \asymp \omega_2 \; \textit{iff} \; (\overleftarrow{O}, m_0) \xrightarrow{\omega_1} (\overleftarrow{O}, m_n) \iff (\overleftarrow{O}, m_0) \xrightarrow{\omega_2} (\overleftarrow{O}, m_n).$$

Proof. The "if" direction follows by definition of reverse equivalence and trace composition. The "only if" direction exploits the properties the Square, Parabolic and Shortening Lemmas.

With Theorem 19 we proved that the notion of causal consistency characterises a space for admissible rollbacks which are: (1) consistent (in the sense that they do not lead to previously unreachable configurations) and (2) flexible enough to allow rearranging of undo actions. This implies that starting from an initial marking, all the markings reached by mixed computations are markings that could be reached by performing only forward computations. Hence, we have:

Theorem 20. *Let O be an occurrence net and m_0 an initial marking. Then,*

$$(\overleftarrow{O}, m_0) \rightarrow^* (\overleftarrow{O}, m') \iff (\overleftarrow{O}, m_0) \twoheadrightarrow^* (\overleftarrow{O}, m').$$

5 Reversing P/T Nets

This section takes advantage of the classical unfolding construction for P/T nets and the reversible semantics of occurrence nets to add causally-consistent reversibility to P/T nets.

Definition 21. *Let (N, m) be a marked P/T net and $\mathcal{U}[N, m]$ its unfolding. The reversible version of (N, m), written $\overleftarrow{(N, m)}$, is $\overleftarrow{\mathcal{U}[N, m]}$.*

Example 22. The reversible version of the nets in Fig. 2 are shown in Fig. 5. We remark that they are the reversible versions of the nets in Fig. 4, which are the unfoldings of the original nets.

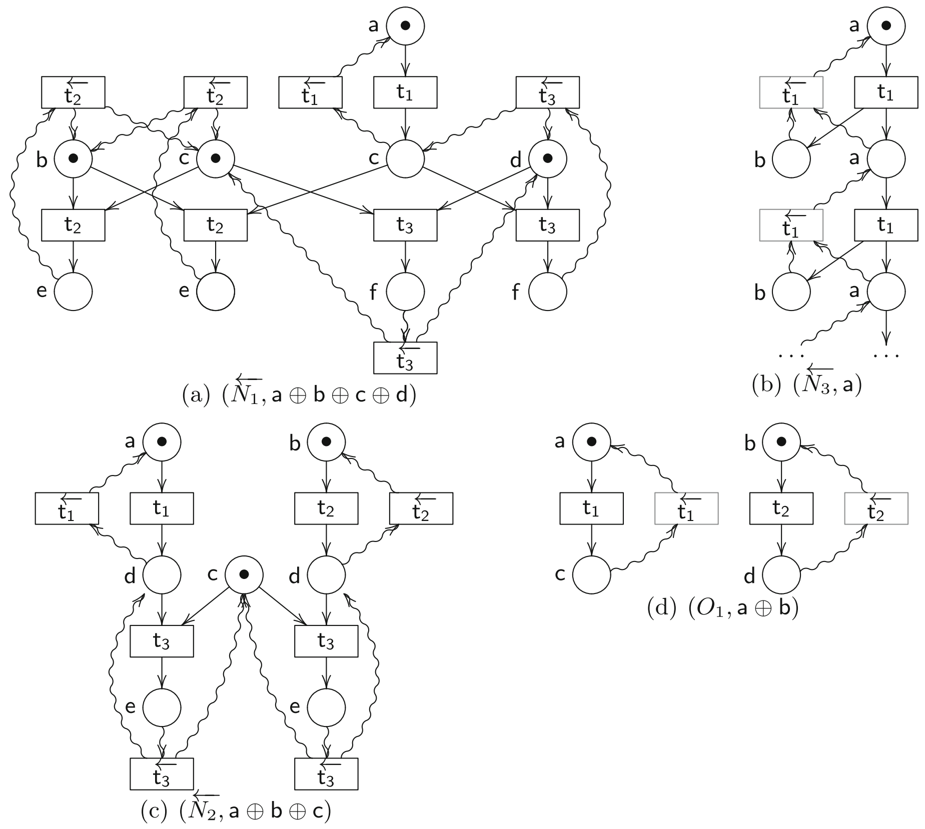
The following result states that a reversible net is a conservative extension of its original version, i.e., reversibility does not change the set of reachable markings. The result is a direct consequence of Lemma 13 and the fact that unfoldings preserve reductions up-to the folding morphism $\mathcal{U}$.

Lemma 23. *$(N, m) \rightarrow^* (N, m')$ iff $\overleftarrow{(N, m)} \twoheadrightarrow^* (\overleftarrow{O}, m'')$ and $m' = f_s(m'')$, where $(f_s, f_t) : \mathcal{U}[N, m] \rightarrow N$, defined such that $f_S(\mathsf{a}, \_, \_) = \mathsf{a}$ and $f_T(\mathsf{t}, \_) = \mathsf{t}$, is the folding morphism.*

We remark that the reversible version of a P/T is defined as the reversible version of an occurrence net (i.e., its unfolding). Consequently, all properties shown in the previous section apply to the reversible semantics of P/T nets. In particular, Lemma 23 combined with Theorem 20 ensures that all markings reachable by the reversible semantics are just the reachable markings of the original P/T net.

6 Finite Representation of Reversible P/T Nets

As shown in Fig. 5(c), the reversible version of a finite net may be infinite. In this section we show how to represent reversible nets in a compact, finite way by using coloured Petri nets. We assume infinite sets $\mathcal{X}$ of variables and $\mathcal{C}$ of colours, defined such that $\mathcal{X} \subset \mathcal{C}$. For $c \in \mathcal{C}$, we write $\mathit{vars}(c)$ for the set of variables in c. With abuse of notation we write $\mathit{vars}(m)$ for the set of variables in a multiset $m \in \mathbb{N}^{\mathcal{P} \times \mathcal{C}}$. Let $\sigma : \mathcal{X} \rightarrow \mathcal{C}$ be a partial function and c a colour (also, $m \in \mathbb{N}^{\mathcal{P} \times \mathcal{C}}$), we write $c\sigma$ (resp., $m\sigma$) for the simultaneous substitution of each variable x in c (resp., m) by $\sigma(x)$.


a
b
c
d
e
f
t_1
t_2
t_3
$\overleftarrow{\mathsf{t}_1}$
$\overleftarrow{\mathsf{t}_2}$
$\overleftarrow{\mathsf{t}_3}$
. . .

(a) $(\overleftarrow{N_1}, \mathsf{a} \oplus \mathsf{b} \oplus \mathsf{c} \oplus \mathsf{d})$ (b) $(\overleftarrow{N_3}, \mathsf{a})$

(c) $(\overleftarrow{N_2}, \mathsf{a} \oplus \mathsf{b} \oplus \mathsf{c})$ (d) $(O_1, \mathsf{a} \oplus \mathsf{b})$

Fig. 5. Reversible P/T and Petri nets

Definition 24 (C-P/T net). *A* coloured place/transition net *(*C-P/T *net) is a 4-tuple $N = (S_N, T_N, {}^\bullet\_{}_N, \_{}^\bullet_N)$ where $S_N \subseteq \mathcal{P}$ is the (nonempty) set of places, $T_N \subseteq \mathcal{T}$ is the set of transitions and the functions ${}^\bullet\_{}_N, \_{}^\bullet_N : T_N \to \mathbb{N}^{S_N \times \mathcal{C}}$ assign source and target to each transition defined such that $vars(\mathsf{t}^\bullet) \subseteq vars({}^\bullet\mathsf{t})$. A marking of a* C-P/T *net N is multiset over $S_N \times \mathcal{C}$ that does not contain variables, i.e., $m \in \mathbb{N}^{S \times \mathcal{C}}$ and $vars(m) = \emptyset$. A marked* C-P/T *net is a pair (N, m) where N is a* P/T *net and m is a marking of N.*

C-P/T nets generalise P/T nets by extending markings to multisets of coloured tokens, and transitions to patterns that need to be instantiated with appropriate colours for firing, as formally stated by the firing rule below.

$$\textsc{(coloured-firing)}\quad \frac{\mathsf{t} = m \; [\rangle \; m' \in T_N}{(N, m\sigma \oplus m'') \xrightarrow{\mathsf{t}} (N, m'\sigma \oplus m'')}$$

The firing of a transition $t = m\ [\rangle\ m'$ requires to instantiate m and m' by substituting variables by colours, i.e., the firing of t consumes the instance $m\sigma$ of the preset m and produces the instance $m'\sigma$ of the postset of m'.

We now introduce an encoding that associates each P/T net N with an equivalent C-P/T net $[\![N]\!]$, whose tokens carry their execution history. We rely on the set of colours $\mathcal{C}$ defined as the least set that contains $\mathcal{X}$ and it is closed under the following rules.

$$\text{(TOKEN)}\ \frac{h \in 2^{\mathcal{C}} \quad n \in \mathbb{N}}{(h, n) \in \mathcal{C}} \qquad \text{(ELEM)}\ \frac{x \in \mathcal{T} \cup \mathcal{P} \quad h \in 2^{\mathcal{C}}}{x(h) \in \mathcal{C}}$$

Colours resemble the unfolding construction (Fig. 3): the colours for tokens are (h, n), where h denotes its (possible empty) set of causes and n is a natural number used for distinguishing tokens with identical causal history. Causal histories are build from coloured versions of transitions ($\mathsf{t}(h)$) and places ($\mathsf{a}(h)$).

Definition 25 (P/T as C-P/T). *Let $N = (S_N, T_N, {}^\bullet\_{}_N, \_^\bullet_N)$ be a P/T net. Then, $[\![N]\!]$ is the C-P/T defined such that $[\![N]\!] = (S_N, T_N, {}^\bullet\_{}_{[\![N]\!]}, {}^\bullet\_{}_{[\![N]\!]})$ and*

- ${}^\bullet\mathsf{t}_{[\![N]\!]} = \mathsf{a}_1(x_1) \oplus \ldots \oplus \mathsf{a}_\mathsf{n}(x_n)$ *where* ${}^\bullet\mathsf{t}_N = \mathsf{a}_1 \ldots \mathsf{a}_\mathsf{n}$ *and* $\forall 1 \leq i \leq n. x_i \in \mathcal{X}$.
- $\mathsf{t}^\bullet_{[\![N]\!]} = \{\mathsf{a}(\{\mathsf{t}(h)\}, i) \mid \mathsf{a} \in \mathit{supp}(\mathsf{t}^\bullet_N) \ \wedge\ 1 \leq i \leq \mathsf{t}^\bullet_N(\mathsf{a}) \ \wedge\ h = {}^\bullet\mathsf{t}_{[\![N]\!]}\}$.

A marked net (N, m) is encoded as $[\![(N, m)]\!] = ([\![N]\!], [\![m]\!])$ where $[\![m]\!] = \{\mathsf{a}(\emptyset, i) \mid \mathsf{a} \in \mathit{supp}(m) \ \wedge\ 1 \leq i \leq m(\mathsf{a})\}$.

The encoding does not alter the structure of a net; it only adds colours to its tokens. In fact, an encoded net has the same places and transitions as the original net, and pre- and postsets of each transition have the same support. Added colours do not interfere with firing because the preset of each transition uses different colour variables for different tokens. The colour $\{\mathsf{t}(h)\}$ assigned to each token produced by the firing of t describes the causal history of the token, i.e., it indicates that the token has been produced by t after consuming the tokens in the preset of t, which is denoted by h. The natural number i is used for distinguishing multiple tokens produced by the same firing. Tokens in the initial marking are coloured as $(\emptyset, i)$, i.e., they have empty causal history.

Example 26. The encoding of the nets in Fig. 2 are shown in Fig. 6. We comment on the encoding of N_1. The transition $\mathsf{t}_1 = \mathsf{a}[\rangle\mathsf{c}$ in N_1 is encoded as $\mathsf{a}(x)[\rangle\mathsf{c}(\mathsf{t}_1(a(x)), 1)$, i.e., the firing of t_1 that consumes a token with colour h from place a generates a token in c with colour $(\mathsf{t}_1(a(h)), 1)$. The transition $\mathsf{t}_2 = \mathsf{b} \oplus c[\rangle\mathsf{e}$ has two places in the preset and uses two variables x and y in its encoded form $\mathsf{b}(x) \oplus c(y)[\rangle\mathsf{e}(\mathsf{t}_2(\mathsf{b}(x) \oplus \mathsf{c}(y)), 1)$. Note that the colour of the token produced in c carries the information of the tokens consumed from both places b and c. The encoding for t_3 is defined analogously.

We illustrate a sequence of firings of $[\![(N_1, \mathsf{a} \oplus \mathsf{b} \oplus \mathsf{c} \oplus \mathsf{d})]\!]$.

$$\begin{aligned}
&([\![N_1]\!], \mathsf{a}(\emptyset, 1) \oplus \mathsf{b}(\emptyset, 1) \oplus \mathsf{c}(\emptyset, 1) \oplus \mathsf{d}(\emptyset, 1))\\
&\xrightarrow{\mathsf{t}_1} ([\![N_1]\!], \mathsf{b}(\emptyset, 1) \oplus \mathsf{c}(\mathsf{t}_1(\mathsf{a}(\emptyset, 1)), 1) \oplus \mathsf{c}(\emptyset, 1) \oplus \mathsf{d}(\emptyset, 1))\\
&\xrightarrow{\mathsf{t}_2} ([\![N_1]\!], \mathsf{e}(\mathsf{t}_2(\mathsf{b}(\emptyset, 1) \oplus \mathsf{c}(\mathsf{t}_1(\mathsf{a}(\emptyset, 1)), 1)), 1) \oplus \mathsf{c}(\emptyset, 1) \oplus \mathsf{d}(\emptyset, 1))
\end{aligned}$$

The firing of t_1 consumes the token $(\emptyset, 1)$ from a and produces the token $(\mathsf{t}_1(\mathsf{a}$ $(\emptyset, 1)), 1)$ in place c. The causal history of the token $\mathsf{t}_1(\mathsf{a}(\emptyset, 1)$ indicates that the token has been produced by the firing of t_1 that consumed the token $(\emptyset, 1)$ from a. The second reduction takes place because of the firing of t_2. By inspecting the causal history of the token produced in the place e we can conclude that t_2 has consumed the token previously generated by t_1.

The following result shows that there is a tight correspondence between the semantics of the coloured version of a P/T net and its unfolding.

Lemma 27. *Let (N, m) be a marked* P/T *net and $\mathcal{U}[N, m] = (O, m')$ its unfolding. Then, $[\![N, m]\!] \xrightarrow{s} ([\![N]\!], m'')$ iff $(O, m') \xrightarrow{s} (O, m'')$.*

Proof. The *if* part follows by induction on the length of the reduction. The base case follows by taking $m'' = m'$ and noting that $[\![N, m]\!] = ([\![N]\!], m')$. The inductive step $s = s'; \mathsf{t}$ follows by applying inductive hypothesis on s' to conclude that $[\![N, m]\!] \xrightarrow{s'} ([\![N]\!], m''')$ iff $(O, m) \xrightarrow{s'} (O, m''')$. *If)* $([\![N]\!], m''') \xrightarrow{\mathsf{t}} ([\![N]\!], m'')$ implies $m''' = {}^\bullet\mathsf{t}_{[\![N]\!]} \oplus m''''$ and $m'' = \mathsf{t}^\bullet_{[\![N]\!]} \oplus m''''$. Since $(O, m) \xrightarrow{s'} (O, m''')$, $CO({}^\bullet\mathsf{t})$. Then, by the unfolding construction we conclude $(O, m''') \xrightarrow{\mathsf{t}} (O, m'')$. The *only if* follows analogously.

The reversible version of $[\![N]\!]$ is defined as for occurrence nets, by adding transitions that are the swapped versions of the ones in N.

Definition 28 (Reversible P/T net). *Let N be a* P/T *net. The reversible version of N is $\overleftarrow{[\![N]\!]}$. The reversible version of a marked* P/T *net (N, m) is the marked* C-P/T *net $(\overleftarrow{[\![N]\!]}, [\![m]\!])$.*

Example 29. The net $[\![\overleftarrow{N_2}]\!]$, the reversible version of $[\![N_2]\!]$ from Fig. 6, is shown in Fig. 7. We now illustrate the execution of $[\![\overleftarrow{N_2}]\!]$.

$$\begin{aligned}
&([\![\overleftarrow{N_2}]\!], \mathsf{a}(\emptyset, 1) \oplus \mathsf{b}(\emptyset, 1) \oplus \mathsf{c}(\emptyset, 1) \oplus \mathsf{d}(\emptyset, 1))\\
\xrightarrow{\mathsf{t}_1}&([\![\overleftarrow{N_2}]\!], \mathsf{b}(\emptyset, 1) \oplus \mathsf{b}(\mathsf{t}_1(\mathsf{a}(\emptyset, 1)), 1) \oplus \mathsf{c}(\emptyset, 1) \oplus \mathsf{d}(\emptyset, 1))\\
\xrightarrow{\mathsf{t}_2}&([\![\overleftarrow{N_2}]\!], \mathsf{b}(\mathsf{t}_1(\mathsf{a}(\emptyset, 1)), 1) \oplus \mathsf{c}(\emptyset, 1) \oplus \mathsf{e}(\mathsf{b}(\emptyset, 1)\mathsf{c}(\emptyset, 1), 1), \mathsf{d}(\emptyset, 1))\\
\overset{\overleftarrow{\mathsf{t}_1}}{\rightsquigarrow}&([\![\overleftarrow{N_2}]\!], \mathsf{a}(\emptyset, 1) \oplus \mathsf{e}(\mathsf{b}(\emptyset, 1)\mathsf{c}(\emptyset, 1), 1) \oplus \mathsf{d}(\emptyset, 1))\\
\overset{\overleftarrow{\mathsf{t}_2}}{\rightsquigarrow}&([\![\overleftarrow{N_2}]\!], \mathsf{a}(\emptyset, 1) \oplus \mathsf{b}(\emptyset, 1) \oplus \mathsf{c}(\emptyset, 1) \oplus \mathsf{d}(\emptyset, 1))
\end{aligned}$$

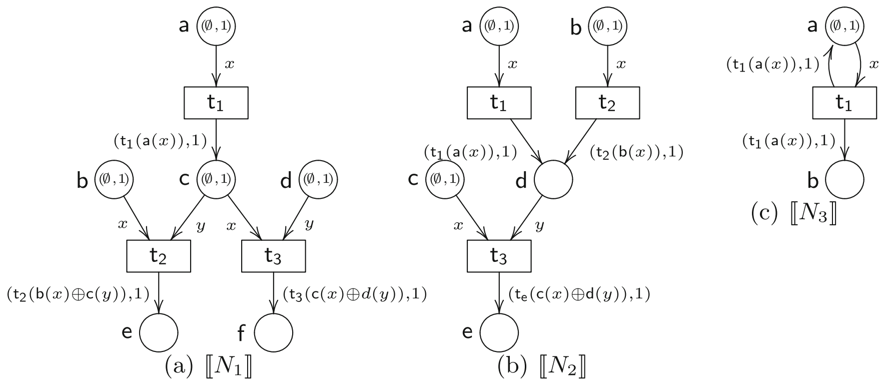


Fig. 6. P/T nets as C-P/T nets

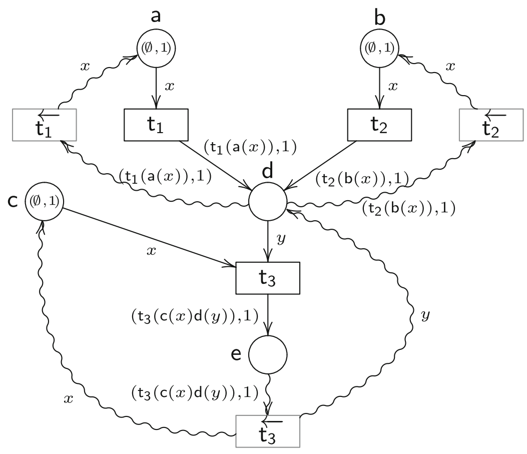


Fig. 7. Reversible coloured net $[\![\overleftarrow{N_2}]\!]$.

In the example above, the firing t_2 can choose to consume either the token $\mathsf{b}(\emptyset, 1)$ or the token $\mathsf{b}(\mathsf{t}_1(\mathsf{a}(\emptyset, 1)), 1)$. Since the first one is chosen, then after t_2 it is still possible to undo t_1. If t_2 chose the second token, then in order to undo t_1 we would first undo t_2, since firing $\overleftarrow{\mathsf{t}_1}$ is not enabled by the token $\mathsf{a}(\emptyset, 1)$.

The following result states that the reductions of the reversible C-P/T of a net are in one-to-one correspondence with the reductions of its reversible unfolding.

Theorem 30 (Correctness). *Let (N, m) be a marked* P/T *net and $\mathcal{U}[N, m] = (O, m')$ its unfolding. Then, $[\![\overleftarrow{N}, [\![m]\!]]\!] \xrightarrow{s} ([\![N]\!], m'')$ iff $(\overleftarrow{O}, m') \xrightarrow{s} (\overleftarrow{O}, m'')$.*

7 Conclusions

We have presented a causally reversible semantics for Place/Transitions Petri Nets (P/T nets) based on two observations. First, occurrence net can be straightforwardly reversed by adding for each transition its reverse. Second, the standard unfolding construction associates a P/T net with an occurrence net that preserves all of its computation. Consequently, the reversible semantics of a P/T net can be obtained as the reversible semantics of its unfolding. We have showed that reversibility in reversible occurrence net is causal-consistent, that is it preserves causality. The unfolding of an occurrence net can be infinite (e.g., it the original P/T net is not acyclic). Therefore we have shown that the reversible behaviour of reversible occurrence nets can be expressed as a finite net whose tokens are coloured by causal histories. Colours in our encoding resemble the causal memories that are typical in reversible process calculi [6,14].

Occurrence nets have a direct mapping into prime event structures. We shall investigate in the future the relation between reversible event structures [5,9,24,28] and our reversible occurrence nets. There is an alternative method for proving causally-consistent reversibility in a reversible model of computation. It is based on showing other properties than those in Sect. 4, mainly the well-foundedness (lack of infinite reverse sequences) and Reverse Diamond properties [22,23]. It would be worthwhile to prove the alternative properties for our reversible nets, and compare the two approaches.

Acknowledgment. Research partly supported by the EU H2020 RISE programme under the Marie Skłodowska Curie grant agreement No 778233. Partly supported by UBACyT projects 20020170100544BA and 20020170100086BA, and CONICET project PIP 11220130100148CO. Also, we thank COST Action IC1405 on Reversible Computation for partial support. The second author acknowledges the support of the Marie Skłodowska Curie Action Individual Fellowship RCADE No 794405.

References

1. Barylska, K., Gogolińska, A., Mikulski, Ł., Philippou, A., Piątkowski, M., Psara, K.: Reversing computations modelled by coloured Petri nets. In: van der Aalst, W.M.P., Bergenthum, R., Carmona, J. (eds.) ATED, vol. 2115, pp. 91–111. CEUR-WS.org (2018)
2. Barylska, K., Koutny, M., Mikulski, Ł., Piątkowski, M.: Reversible computation vs. reversibility in Petri nets. Sci. Comput. Program. **151**, 48–60 (2018). https://doi.org/10.1016/j.scico.2017.10.008
3. Cardoza, E., Lipton, R., Meyer, A.R.: Exponential space complete problems for Petri nets and commutative semigroups (preliminary report). In: Proceedings of STOC, pp. 50–54. ACM (1976). https://doi.org/10.1145/800113.803630
4. Cristescu, I., Krivine, J., Varacca, D.: A compositional semantics for the reversible π-calculus. In: Symposium on Logic in Computer Science, LICS, pp. 388–397. IEEE Computer Society (2013)
5. Cristescu, I.D., Krivine, J., Varacca, D.: Rigid families for CCS and the π-calculus. In: Leucker, M., Rueda, C., Valencia, F.D. (eds.) ICTAC 2015. LNCS, vol. 9399, pp. 223–240. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25150-9_14

6. Danos, V., Krivine, J.: Reversible communicating systems. In: Gardner, P., Yoshida, N. (eds.) CONCUR 2004. LNCS, vol. 3170, pp. 292–307. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28644-8_19
7. Danos, V., Krivine, J.: Transactions in RCCS. In: Abadi, M., de Alfaro, L. (eds.) CONCUR 2005. LNCS, vol. 3653, pp. 398–412. Springer, Heidelberg (2005). https://doi.org/10.1007/11539452_31
8. Giachino, E., Lanese, I., Mezzina, C.A.: Causal-consistent reversible debugging. In: Gnesi, S., Rensink, A. (eds.) FASE 2014. LNCS, vol. 8411, pp. 370–384. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54804-8_26
9. Graversen, E., Phillips, I., Yoshida, N.: Event structure semantics of (controlled) reversible CCS. In: Kari, J., Ulidowski, I. (eds.) RC 2018. LNCS, vol. 11106, pp. 102–122. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99498-7_7
10. Hayman, J., Winskel, G.: The unfolding of general Petri nets. In: Hariharan, R., Mukund, M., Vinay, V. (eds.) FSTTCS. LIPIcs, vol. 2, pp. 223–234 (2008). https://doi.org/10.4230/LIPIcs.FSTTCS.2008.1755
11. Hoey, J., Ulidowski, I., Yuen, S.: Reversing imperative parallel programs with blocks and procedures. In: Proceedings of EXPRESS/SOS (2018)
12. Kuhn, S., Ulidowski, I.: A calculus for local reversibility. In: Devitt, S., Lanese, I. (eds.) RC 2016. LNCS, vol. 9720, pp. 20–35. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40578-0_2
13. Lanese, I., Lienhardt, M., Mezzina, C.A., Schmitt, A., Stefani, J.-B.: Concurrent flexible reversibility. In: Felleisen, M., Gardner, P. (eds.) ESOP 2013. LNCS, vol. 7792, pp. 370–390. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37036-6_21
14. Lanese, I., Mezzina, C.A., Stefani, J.: Reversibility in the higher-order π-calculus. Theor. Comput. Sci. **625**, 25–84 (2016). https://doi.org/10.1016/j.tcs.2016.02.019
15. Lanese, I., Mezzina, C.A., Tiezzi, F.: Causal-consistent reversibility. Bull. EATCS **114** (2014)
16. Lanese, I., Nishida, N., Palacios, A., Vidal, G.: CauDEr: a causal-consistent reversible debugger for Erlang. In: Gallagher, J.P., Sulzmann, M. (eds.) FLOPS 2018. LNCS, vol. 10818, pp. 247–263. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-90686-7_16
17. Leeman Jr., G.B.: A formal approach to undo operations in programming languages. ACM Trans. Program. Lang. Syst. **8**(1), 50–87 (1986). https://doi.org/10.1145/5001.5005
18. Lévy, J.: An algebraic interpretation of the $\lambda\beta K$-calculus; and an application of a labelled λ-calculus. Theor. Comput. Sci. **2**(1), 97–114 (1976). https://doi.org/10.1016/0304-3975(76)90009-8
19. Medic, D., Mezzina, C.A., Phillips, I., Yoshida, N.: A parametric framework for reversible π-calculi. In: Pérez, J.A., Tini, S. (eds.) Proceedings of EXPRESS/SOS. EPTCS, vol. 276, pp. 87–103 (2018)
20. Nielsen, M., Plotkin, G.D., Winskel, G.: Petri nets, event structures and domains, part I. Theor. Comput. Sci. **13**, 85–108 (1981). https://doi.org/10.1016/0304-3975(81)90112-2
21. Philippou, A., Psara, K.: Reversible computation in Petri nets. In: Kari, J., Ulidowski, I. (eds.) RC 2018. LNCS, vol. 11106, pp. 84–101. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99498-7_6
22. Phillips, I., Ulidowski, I.: Reversibility and models for concurrency. In: Proceedings of SOS 2007. ENTCS, vol. 192, pp. 93–108 (2007)
23. Phillips, I., Ulidowski, I.: Reversing algebraic process calculi. J. Log. Algebr. Program. **73**(1–2), 70–96 (2007). https://doi.org/10.1016/j.jlap.2006.11.002

24. Phillips, I., Ulidowski, I.: Reversibility and asymmetric conflict in event structures. J. Log. Algebr. Meth. Program. **84**(6), 781–805 (2015). https://doi.org/10.1016/j.jlamp.2015.07.004
25. Phillips, I., Ulidowski, I., Yuen, S.: A reversible process calculus and the modelling of the ERK signalling pathway. In: Glück, R., Yokoyama, T. (eds.) RC 2012. LNCS, vol. 7581, pp. 218–232. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36315-3_18
26. Pinna, G.M.: Reversing steps in membrane systems computations. In: Gheorghe, M., Rozenberg, G., Salomaa, A., Zandron, C. (eds.) CMC 2017. LNCS, vol. 10725, pp. 245–261. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73359-3_16
27. Schordan, M., Oppelstrup, T., Jefferson Jr., D., Barnes, P.D.: Generation of reversible C++ code for optimistic parallel discrete event simulation. New Gener. Comput. **36**(3), 257–280 (2018). https://doi.org/10.1007/s00354-018-0038-2
28. Ulidowski, I., Phillips, I., Yuen, S.: Reversing event structures. New Gener. Comput. **36**(3), 281–306 (2018). https://doi.org/10.1007/s00354-018-0040-8
29. Vassor, M., Stefani, J.-B.: Checkpoint/rollback vs causally-consistent reversibility. In: Kari, J., Ulidowski, I. (eds.) RC 2018. LNCS, vol. 11106, pp. 286–303. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99498-7_20
30. Vos, A.D., Baerdemacker, S.D., Rentergem, Y.V.: Synthesis of Quantum Circuits vs. Synthesis of Classical Reversible Circuits. Synthesis Lectures on Digital Circuits and Systems, Morgan & Claypool Publishers (2018). https://doi.org/10.2200/S00856ED1V01Y201805DCS054