Non-sequential behaviour of dynamic nets ⋆

Roberto Bruni and Hernán Melgratti

Dipartimento di Informatica, Università di Pisa, Italia.
{bruni,melgratt}@di.unipi.it

Abstract. Dynamic nets are an extension of Petri nets where the net topology may change dynamically. This is achieved by allowing (i) tokens to be *coloured* with place names (carried on as data), (ii) transitions to designate places where to spawn new tokens on the basis of the *colours* in the fetched tokens, and (iii) firings to add fresh places and transitions to the net. Dynamic nets have been given step or interleaving semantics but, to the best of our knowledge, their non-sequential truly concurrent semantics has not been addressed in the literature. To fill this gap, we extend the ordinary notions of processes and unfolding to dynamic nets, providing two different constructions: (i) a specific process and unfolding for a particular initial marking, and (ii) processes and unfolding patterns that abstract away from the colours of the token initially available.

1 Introduction

Petri nets, introduced in [13], have became a reference model for studying concurrent systems, mainly due to their simplicity and the intrinsic concurrent nature of their behaviours. In addition to the classical "token game" operational semantics, several alternative approaches have appeared in the literature for characterising the semantics of Petri nets, notably non-sequential processes, unfolding constructions and algebraic models. In particular, non-sequential processes have played a very important rôle when studying the non-interleaved semantics of Petri nets. In essence, non-sequential processes provide a full-fledged account for the causal relations among the steps of a computation, i.e., they provide a full explanation about the causes that led to the firing of a transition.

Recently, the basic Petri net model has been extended to account for mobility, giving birth to the so called *Mobile nets* [1] and *Dynamic nets* [1,5]. The difference between the two is that the structure of dynamic nets is slightly more constrained so to enforce the locality principle: tokens in a place can be consumed only by local transitions (that were spawned together with that place). Dynamic nets are an extension of coloured nets (since tokens carry on information) where token colours are the names of places in the net, transitions may use the information carried on by tokens to designate places where to spawn new tokens, and transitions may add fresh places and transitions when they fire.

⋆ Research supported by the EU FET-GC2 IST-2004-16004 Integrated Project Sensoria.

The behaviour of dynamic nets has been defined by providing their step or interleaving semantics but, to the best of our knowledge, there is no proposal for their non-sequential truly concurrent semantics. In this work, we pursue this line of research by extending the classical notions of processes and unfolding to dynamic nets. In particular, we provide two different kind of constructions: (i) a specific process and unfolding for a particular initial marking (where the information carried on by tokens of the initial marking is essential), and (ii) a notion of general process and unfolding pattern that do not depend on the information carried on by tokens of the initial marking.

It is worth remarking that the unfolding construction has proved very helpful to define the correct notion of processes for dynamic nets, as otherwise it would have been very difficult to deal with the changes in the net topology due to the introduction of fresh subnets.

The results presented here can find an interesting application in defining net models and causal semantics for distributed mobile calculi and programming languages. More specifically we are thinking of the join calculus [7] and those languages like JoCaml [6] and C-omega [3] whose designs have been strongly influenced by the join paradigm. In fact it has been shown in [5] that join processes may coherently be viewed as dynamic nets (and vice versa).

Structure of the paper. In Section 2 we recall the preliminary definitions from the literature, aiming to keep the paper self-contained. It is worth noting that while we expect the reader to have some confidence with the material in Sections 2.1 and 2.2 (nets, step semantics, causal processes and unfolding), we have chosen to give an extensive introduction to dynamic nets (Sections 2.3), which could be a less familiar subject for many readers. The unfolding construction for marked dynamic nets is carried on in Section 3 accompanied by the definition of deterministic process of dynamic nets. The main result establishes a strong correspondence between the two notions. Section 4 introduces unfolding patterns and process patterns as a framework to give more compact and abstract representations of net behaviour: the same pattern can be instantiated to many different concrete computations. Conclusions and directions for future work are given in Section 5.

2 Background

2.1 P/T Petri Nets

Petri nets are built up from *places* (denoting resources, type of messages), which are repositories of *tokens* (representing instances of resources), and *transitions*, which fetch and produce tokens. In the following we shall consider an infinite set $\mathcal{P}$ of resource names.

Definition 2.1 (Net). *A* net N *is a 4-tuple* $N = (S_N, T_N, \delta_{0N}, \delta_{1N})$ *where* $S_N \subseteq \mathcal{P}$ *is the (nonempty) set of places,* $\mathtt{a}, \mathtt{a}', \ldots$, T_N *is the set of transitions,* $\mathtt{t}, \mathtt{t}', \ldots$ *(with* $S_N \cap T_N = \emptyset$*), and the functions* $\delta_{0N}, \delta_{1N} : T_N \to \wp_{\mathrm{f}}(S_N)$ *assign finite sets of places, called respectively source and target, to each transition.*

We denote $S_N \cup T_N$ by N, and omit the subscript N if no confusion arises. We abbreviate a transition $\mathtt{t} \in T$ with *preset* ${}^\bullet\mathtt{t} = \delta_0(\mathtt{t}) = s_1$ and *postset* $\mathtt{t}^\bullet = \delta_1(\mathtt{t}) = s_2$ as $s_1[\rangle s_2$. Similarly for any place $\mathtt{a} \in S$, the preset ${}^\bullet\mathtt{a} = \{\mathtt{t}|\mathtt{a} \in \mathtt{t}^\bullet\}$ of $\mathtt{a}$ is the set of all transitions of which $\mathtt{a}$ is target and the postset $\mathtt{a}^\bullet = \{\mathtt{t}|\mathtt{a} \in {}^\bullet\mathtt{t}\}$ of $\mathtt{a}$ is the set of all transitions of which $\mathtt{a}$ is source. We consider only nets whose transitions have a non-empty preset. If ${}^\bullet\mathtt{a} \cup \mathtt{a}^\bullet = \emptyset$ the place $\mathtt{a}$ is *isolated*. Moreover, we let ${}^\circ N = \{x \in N | {}^\bullet x = \emptyset\}$ and $N^\circ = \{x \in N | x^\bullet = \emptyset\}$ denote the sets of *initial* and *final elements* of N respectively.

While according to Definition 2.1 transitions can consume and produce at most one token in each state, in P/T nets (see Definition 2.2) transitions can fetch and produce several tokens in a particular place, i.e., the pre- and postsets of transitions are multisets instead of sets.

Given a set S, a *multiset* over S is a function $m : S \to \mathbb{N}$ (where $\mathbb{N}$ denotes the set of natural numbers with zero). Let $dom(m) = \{\mathtt{s} \in S \mid m(\mathtt{s}) > 0\}$. The set of all finite multisets (i.e., with finite domain) over S is written $\mathcal{M}_S$. The empty multiset (i.e., with $dom(m) = \emptyset$) is written $\emptyset$. The multiset union $\oplus$ is defined as $(m_1 \oplus m_2)(\mathtt{s}) = m_1(\mathtt{s}) + m_2(\mathtt{s})$ for any $\mathtt{s} \in S$. Given a multiset m, $|m| = \sum_{\mathtt{s} \in S} m(\mathtt{s})$ denotes the *size* of m.

Note that $\oplus$ is associative and commutative, and $\emptyset$ is the identity for $\oplus$. Hence, $\mathcal{M}_S$ is the free commutative monoid $S^\oplus$ over S. We write $\mathtt{s}$ for a singleton multiset m such that $dom(\mathtt{s}) = \{\mathtt{s}\}$ and $m(\mathtt{s}) = 1$. Moreover, we write $\{[\mathtt{s_1}, \ldots, \mathtt{s_n}]\}$ for $\mathtt{s_1} \oplus \ldots \oplus \mathtt{s_n}$. By abusing notation we will apply functions (i.e., f_S) over (multi)sets, meaning the multiset obtained by applying the function element-wise: $f_S(\{[\mathtt{a_0}, \ldots, \mathtt{a_n}]\}) = f_S(\mathtt{a_0}) \oplus \ldots \oplus f_S(\mathtt{a_n})$. Also, we shall use set operators over multisets to denote the operation over the domain of the multiset, e.g. $s \in m$ and $m_1 \cap m_2$ in place of $s \in dom(m)$ and $dom(m_1) \cap dom(m_2)$.

Definition 2.2 (P/T net). *A marked place / transition Petri net (P/T net) is a tuple $N = (S_N, T_N, \delta_{0N}, \delta_{1N}, m_{0N})$ where $S_N \subseteq \mathcal{P}$ is a set of places, T_N is a set of transitions, the functions $\delta_{0N}, \delta_{1N} : T_N \to \mathcal{M}_{S_N}$ assign respectively, source and target to each transition, and $m_{0N} \in \mathcal{M}_{S_N}$ is the initial marking.*

The notions of pre- and postset, initial and final elements, and isolated places are straightforwardly extended to consider multisets instead of sets. Note that a net can be regarded as a P/T net whose arcs have unary weights.

The operational semantics of P/T nets is given by (the least relation inductively generated by) the inference rules in Figure 1. Given a net N, the proof for $m \to_T m'$ means that a marking m evolves to m' under a *step*, i.e., the concurrent firing of several transitions. We omit the subscript T whenever the set of transitions is clear from the context. Rule (FIRING) describes the evolution of the state of a net (represented by the marking $m \oplus m''$) by applying a transition $m[\rangle m'$, which consumes the tokens m corresponding to its preset and produces the tokens m' corresponding to its postset. The multiset m'' represents idle resources, i.e., the tokens that persist during the evolution. Rule (STEP) stands for the parallel composition of computations. The sequential composition

$$\frac{\overset{\text{(FIRING)}}{m\ [\rangle\ m' \in T \quad m'' \in \mathcal{M}_S}}{m \oplus m'' \to_T m' \oplus m''} \qquad \frac{\overset{\text{(STEP)}}{m_1 \to_T m_1' \quad m_2 \to_T m_2'}}{m_1 \oplus m_2 \to_T m_1' \oplus m_2'}$$

Fig. 1. Operational semantics of P/T nets.

of computations is the reflexive and transitive closure of $\to$, which is written $\to^*$, i.e., $m \to^* m'$ denotes the evolution of m to m' under a (possibly empty) sequence of steps.

2.2 Unfolding and Process Semantics of P/T nets

The definition of the processes and unfolding semantics of P/T nets rely on the notions of occurrence nets and deterministic causal nets, which are defined below. (We report here on the presentation given in [12])

Definition 2.3 (Occurrence net). *A net N is an occurrence net if*

- *for all* $\mathtt{a} \in S_N$, $|{}^\bullet\mathtt{a}| \leq 1$
- *the* causal dependency *relation $\prec$ is irreflexive, where $\prec$ is the transitive closure of the* immediate cause *relation*

$$\prec^1 = \{(\mathtt{a},\mathtt{t}) | \mathtt{a} \in S_N \wedge t \in \mathtt{a}^\bullet\} \cup \{(\mathtt{t},\mathtt{a}) | \mathtt{a} \in S_N \wedge t \in {}^\bullet\mathtt{a}\};$$

  moreover, $\forall \mathtt{t} \in T_N$, the set $\{\mathtt{t}' \in T_N | \mathtt{t}' \prec \mathtt{t}\}$ is finite and the reflexive closure of $\prec$ is denoted by $\preceq$;
- *the* binary conflict *relation $\#$ on $T_N \cup S_N$ is irreflexive, where $\#$ is defined in terms of the* binary direct conflict *relation $\#_m$ as below:*

$$\forall \mathtt{t}_1,\mathtt{t}_2 \in T_N, \quad \mathtt{t}_1 \#_m \mathtt{t}_2 \Leftrightarrow \delta_{0N}(\mathtt{t}_1) \cap \delta_{0N}(\mathtt{t}_2) \neq \emptyset \wedge \mathtt{t}_1 \neq \mathtt{t}_2$$
$$\forall x,y \in S_N \cup T_N, \quad x \# y \Leftrightarrow \exists \mathtt{t}_1,\mathtt{t}_2 \in T_N : \mathtt{t}_1 \#_m \mathtt{t}_2 \wedge \mathtt{t}_1 \preceq x \wedge \mathtt{t}_2 \preceq y.$$

Given $x, y \in T_N \cup S_N$ s.t. $x \neq y$, x and y are *concurrent*, written $x\ co\ y$, when $x \not\prec y$, $y \not\prec x$, and $\neg x \# y$. A set $X \in T_N \cup S_N$ is concurrent, written $CO(X)$, if $\forall x, y \in X : x \neq y \Rightarrow x\ co\ y$, and $|\{\mathtt{t} \in T_N | \exists x \in X, \mathtt{t} \preceq x\}|$ is finite.

Definition 2.4 (Causal Net). *A net $K = (S_K, T_K, \delta_{0K}, \delta_{1K})$ is a* causal net *(also called* deterministic occurrence net*) if it is an occurence net and*

$$\forall \mathtt{a} \in S_K, |\mathtt{a}^\bullet| \leq 1.$$

It is worth noting that for any causal net K the conflict relation is empty. Consequently, a causal net is an acyclic net where the presets (resp. postsets) of transitions do not share places.

Occurrence nets can represent non-sequential computations: their places represent tokens and their transitions represents events, i.e., firings. The "typing" of tokens and events in the occurence net over places and transitions of the executed net can be expressed as net morphisms, mapping tokens to the places where they have been stored and events to the triggered transitions.

$$\frac{m_N(\mathtt{a}) = n}{\{(\emptyset, \mathtt{a})\} \times [n] \subseteq S} \quad \text{(INI-MK)}$$

$$\frac{B = \{(\epsilon_j, b_j, i_j) | j \in J\} \subseteq S,\ Co(B),\ \mathtt{t} \in T_N,\ \delta_{0N}(\mathtt{t}) = \oplus_{j\in J} b_j}{(B, \mathtt{t}) \in T, \quad \delta_0(B, \mathtt{t}) = B} \quad \text{(PRE)}$$

$$\frac{x = (B, \mathtt{t}) \in T}{Q = \{(\{x\}, b, i) \mid 1 \le i \le \delta_{1N}(\mathtt{t})(\mathtt{b})\} \subseteq S, \quad \delta_1(x) = Q} \quad \text{(POST)}$$

Fig. 2. Unfolding rules.

Definition 2.5 (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 $f_S(\delta_{iN}(\mathtt{t})) = \delta_{iN'}(f_T(\mathtt{t}))$ for any $\mathtt{t}$ and $i = 0, 1$. Moreover, N and N' are said to be* isomorphic*, and thus equivalent, if f is bijective.*

The above definition extends in the obvious way to the cases in which N and N' are P/T nets.

Definition 2.6 (Deterministic Causal Process). *Let N be* P/T *net. A* deterministic causal process *for N is a net morphism P from a causal net K to N such that $P({}^\circ K) = m_{0N}$,i.e., P maps the implicit initial marking of K (i.e., the minimal elements ${}^\circ K$) to the initial marking of N.*

Roughly, a deterministic process represents just a set of causally equivalent computations [8]. Differently, the unfolding of a net N is the least occurrence net that can account for all the possible computations over N, making explicit the causal dependencies, conflicts and concurrency between firings.

Definition 2.7 (Unfolding). *Let N be a* P/T *net. The occurrence net $\mathcal{U}[N] = (S, T, \delta_0, \delta_1)$ generated inductively by the inference rules in Figure 2 is said the* unfolding *of N.*

In the case of the unfolding $\mathcal{U}$, it can be readily verified that the mapping over N is just the projection of places and transition names to their second element. In fact tokens are encoded as triples $(H, \mathtt{a}, i)$ where H is the set of immediate causes (determining the history of the token), $\mathtt{a}$ is the place of N where the token resides and i is a positive integer used to disambiguate tokens in the same place and with the same history, while events are encoded as pairs $(H, \mathtt{t})$, where H is the set of immediate causes and $\mathtt{t}$ is the fired transition.

2.3 Dynamic nets

Different formulations for dynamic nets have been proposed in the literature [1,5]. The definition we give here is based on [1]. We consider an infinite set of place names $\mathcal{P}$ ranged over by $\mathtt{a}, \mathtt{b}, \ldots$ and an infinite set of variable names $\mathcal{X}$, ranged over by $x, y, \ldots$. We require also variable names to be different from place names, i.e., $\mathcal{X} \cap \mathcal{P} = \emptyset$. We will use $\mathcal{C} = \mathcal{P} \cup \mathcal{X}$, ranged over by $c_1, c_2, \ldots.$, to refer both to place and variable names.

Similar to high-level nets, tokens in dynamic nets carry on information (or colours as they are usually known). Although colours can be thought of as data structures of any type, for the sake of simplicity we will assume colours to be sequences of names. Let C be a set of names, C^* stands for the set of all finite (possible empty) sequences of C, i.e., $C^* = \{(c_1, \ldots, c_n) \mid \forall i \ s.t.\ 0 \leq i \leq n : c_i \in C\}$. The empty sequence, i.e. a token that carries no information, is denoted by $\bullet$ (by analogy with ordinary tokens), and the underlying set of a sequence $(c_1, \ldots, c_n)$ is written:

$$\overline{(c_1, \ldots, c_n)} = \bigcup_i \{c_i\}$$

Then, markings of a dynamic net are just coloured multisets.

Definition 2.8 (Coloured Multiset). *Given two sets S and C, a* coloured multiset *over S and C is a function $m : S \to C \to \mathbb{N}$. Let $dom(m) = \{(s, c) \in S \times C \mid m(s)(c) > 0\}$. The set of all finite (coloured) multisets over S and C^* is written $\mathcal{M}_{S,C}$. The multiset union is defined as $(m_1 \oplus m_2)(s)(c) = m_1(s)(c) + m_2(s)(c)$.*

We write $s(c)$ for a multiset m such that $dom(m) = \{(s, c)\}$ and $m(s)(c) = 1$. Additionally, $(s, c) \in m$ is a shorthand for $(s, c) \in dom(m)$, while $s \in m$ means $(s, c) \in m$ for some c.

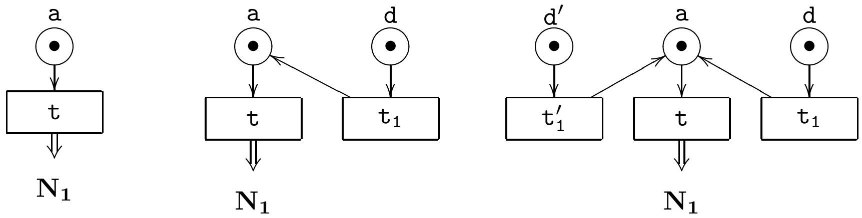
Definition 2.9 (DN). *The set* DN *is the least set satisfying the following equation (when considering the set of colours $\mathcal{C}$):*

$$\begin{aligned} \mathcal{N} = \{ & (S_N, T_N, \delta_{0N}, \delta_{1N}, m_{0N}) \mid \\ & S_N \subseteq \mathcal{P} \ \wedge \ \delta_{0N} : T_N \to \mathcal{M}_{S_N,\mathcal{C}} \ \wedge \ \delta_{1N} : T_N \to \mathcal{N} \ \wedge \ m_{0N} \in \mathcal{M}_{\mathcal{C},\mathcal{C}}\} \end{aligned}$$

For $(S_N, T_N, \delta_{0N}, \delta_{1N}, m_{0N}) \in$ DN, S_N is the set of places, T_N is the set of the transitions, δ_{0N} and δ_{1N} are the functions assigning the pre- and postset to every transition, and m_{0N} is the initial marking. Note that $\mathcal{N}$ is a domain equation [9] defining the recursive type of dynamic nets. The simplest elements in $\mathcal{N}$ are markings, i.e., the tuples $(\emptyset, \emptyset, \emptyset, \emptyset, m)$ with $m \in \mathcal{M}_{\mathcal{C},\mathcal{C}}$. Then, nets are defined recursively, because the postset of any transition (given by δ_{1N}) is another element of $\mathcal{N}$. The set DN is defined as the least fixed point of the recursive equation above.

As usual, we denote $S_N \cup T_N$ by N, and omit subscript N whenever no confusion arises. Moreover, we abbreviate a transition $\mathtt{t} \in T$ such that $\delta_0(\mathtt{t}) = m$ and $\delta_1(\mathtt{t}) = N$ as $m[\rangle N$, and refer to m as the *preset* of $\mathtt{t}$ (written ${}^\bullet\mathtt{t}$) and N as the *postset* of $\mathtt{t}$ (written $\mathtt{t}^\bullet$).

Note that the initial marking m_{0N} is not required to be a multiset over the places of the net, i.e., the initial marking can put tokens in places that are not



(a) Initial state. (b) After firing t. (c) After two firings of t.

Fig. 3. A simple dynamic net.

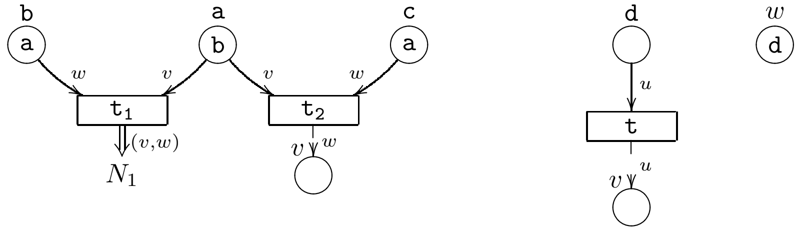
defined by the net. In fact, the initial marking m_{0N} is a multiset in $m_{0N} \in \mathcal{M}_{\mathcal{C},\mathcal{C}}$ and not over the places of the net S_N (e.g., in $\mathcal{M}_{S_N,\mathcal{C}}$). A trivial example is to consider the coloured transition $\mathtt{a}(x)[\rangle(\emptyset,\emptyset,\emptyset,\emptyset,\mathtt{b}(\bullet))$, where $\mathtt{b}$ does not belong to the places of the subnet $(\emptyset,\emptyset,\emptyset,\emptyset,\mathtt{b}(\bullet))$. We usually abbreviate transitions as the previous one, where the postset does not define new places, by writing just the initial marking fixed by the postset, i.e., $\mathtt{a}(x)[\rangle\mathtt{b}(\bullet)$.

Names defined in S_N act as binders on N. Therefore, nets are considered up-to α-conversion on S_N. For instance, the two nets $(\{a\},\emptyset,\emptyset,\emptyset,a(\bullet))$ and $(\{b\},\emptyset,\emptyset,\emptyset,b(\bullet))$ are α-equivalent, while $(\emptyset,\emptyset,\emptyset,\emptyset,a(\bullet))$ and $(\emptyset,\emptyset,\emptyset,\emptyset,b(\bullet))$ are not. Analogously, the names of transitions in T_N are binders, and hence we consider nets up-to α-conversion on transition names.

Example 2.1. Consider the net N in Figure 3(a), where circles are places, boxes are transitions, and solid lines connect transitions to their pre and postset, and tokens are represented by their colours. The double-lined arrow indicates the dynamic transition $\mathtt{t} = \mathtt{a}(\bullet)[\rangle N_1$, which creates an instance of the subnet N_1 when fired. We allow the initial marking of N_1 and the postset of transitions in T_{N_1} to generate tokens in $\mathtt{a}$. Therefore, the following is a valid definition for N_1: $S_{N_1} = \{\mathtt{d}\}$, $T_{N_1} = \{\mathtt{t_1}\}$, $m_{0N_1} = \mathtt{a}(\bullet)\oplus\mathtt{d}(\bullet)$ and $\mathtt{t_1} = \mathtt{d}(\bullet)[\rangle\mathtt{a}(\bullet)$. A firing of $\mathtt{t}$ will lead to (a net isomorphic to) the net shown in Figure 3(b). (Firings are formally defined in Figure 5.) A fresh place $\mathtt{d}$ and a transition $\mathtt{t_1}$ (whose pre- and postset are $\mathtt{d}(\bullet)$ and $\mathtt{a}(\bullet)$, resp.) have been added to the net. Also two tokens have been produced: one in $\mathtt{a}$ and the other in $\mathtt{d}$, accordingly to the initial marking of N_1. This marking enables $\mathtt{t}$, which can be fired again. The new activation of $\mathtt{t}$ will create a new subnet containing a new place and a new transition whose names are different from all other names already present in the net (Figure 3(c)).

The colours appearing in the preset of a transition are intended to be the formal parameters of that transition, which are substituted by the actual colours of consumed tokens when a transition is fired. The set of formal parameters (or received names) of a transition is defined below.

Definition 2.10 (Received names of a transition). *The* set of colours of a multiset $m \subseteq \mathcal{M}_{S,C}$ *is defined as* $col(m) = \cup_{(a,c)\in m}\ \overline{c}$*, the* set of constants *is* $col_{\mathcal{P}}(m) = col(m) \cap \mathcal{P}$*, and the* set of variables *or* received names of a multiset



(a) N. (b) N_1.

Fig. 4. Graphical representation of N and N_1.

is $rn(m) = col_{\mathcal{X}}(m) = col(m) \cap \mathcal{X}$*. Given a transition* $t = m[\rangle N$*, the set of* received names *of* t *is given by* $rn(t) = col_{\mathcal{X}}(m)$*.*

Note that $col_{\mathcal{P}}(m)$ stands for the set of constant (not variable) colours used as formal parameters of a transition. This feature provides a basic mechanism for pattern matching, since a transition will be enabled only when consumed tokens carry on the same constant colours as those specified by the preset of the transition. Differently, $col_{\mathcal{X}}(m)$ is the set of all variables, which will be instantiated when the transition fires.

Definition 2.11 (Defined and Free names). *The set of* defined names in a marking m *is* $dn(m) = \{\mathtt{a} | \exists \mathtt{c}.(\mathtt{a}, \mathtt{c}) \in m\}$*, i.e., names appearing in place position. Given* $N = (S_N, T_N, \delta_{0N}, \delta_{1N}, m_{0N}) \in \text{DN}$*, the* set of defined *(dn) and* free *(fn)* names *of transitions, sets of transitions, and nets are defined as follow:*

$dn(m_1[\rangle N_1) = dn(m_1)$
$dn(T_N) = \bigcup_{t \in T_N} \ dn(t)$
$dn(N) = S_N$
$fn(m_1[\rangle N_1) = dn(m_1) \ \cup \ col_{\mathcal{P}}(m_1) \cup \ (fn(N_1) \setminus rn(m_1))$
$fn(T_N) = \bigcup_{t \in T_N} fn(t) \setminus dn(T_N)$
$fn(N) = (fn(T_N) \cup dn(m_{0N}) \cup col(m_{0N})) \setminus S_N$

Example 2.2. Consider N_1 and N defined as follow and depicted in Figure 4 (Note places v and w in the representation of N_1 are not defined places by N_1):

$S_N = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$	$T_N = \{\mathtt{t}_1, \mathtt{t}_2\}$	$S_{N_1} = \{\mathtt{d}\}$
$\delta_{0N}(\mathtt{t}_1) = \mathtt{a}(v) \oplus \mathtt{b}(w)$	$\delta_{1N}(\mathtt{t}_1) = N_1$	$T_{N_1} = \{\mathtt{t}\}$
$\delta_{0N}(\mathtt{t}_2) = \mathtt{a}(v) \oplus \mathtt{c}(w)$	$\delta_{1N}(\mathtt{t}_2) = v(w)$	$\delta_{0N_1}(\mathtt{t}) = \mathtt{d}(u)$
$m_N = \mathtt{a}(\mathtt{b}) \oplus \mathtt{b}(\mathtt{a}) \oplus \mathtt{c}(\mathtt{a})$		$\delta_{1N_1}(\mathtt{t}) = v(u)$
		$m_{N_1} = w(\mathtt{d})$

The sets of defined, received and free names of N and N_1 are as follow

$$\begin{array}{llll} rn(\mathtt{t}_1) = \{v, w\} & rn(\mathtt{t}_2) = \{v, w\} & & rn(\mathtt{t}) = \{u\} \\ dn(\mathtt{t}_1) = \{\mathtt{a}, \mathtt{b}\} & dn(\mathtt{t}_2) = \{\mathtt{a}, \mathtt{c}\} & & dn(\mathtt{t}) = \{\mathtt{d}\} \\ fn(\mathtt{t}_1) = \{\mathtt{a}, \mathtt{b}\} & fn(\mathtt{t}_2) = \{\mathtt{a}, \mathtt{c}\} & & fn(\mathtt{t}) = \{\mathtt{d}, v\} \\ dn(T_N) = dn(N) = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\} & & & dn(T_{N_1}) = dn(N_1) = \{\mathtt{d}\} \\ fn(T_N) = \emptyset & & & fn(T_{N_1}) = \{v\} \\ fn(N) = \emptyset & & & fn(N_1) = \{v, w\} \end{array}$$

Definition 2.12 (Dynamic Net). *$N \in$ DN is a* dynamic net *if $fn(N) = \emptyset$.*

The above definition states that a dynamic net is closed, i.e., it does not generate tokens in places that do not belong to it. The condition $fn(N) = \emptyset$ assures tokens to be generated always in places of the net, since markings are bound to places defined by the net, which are guaranteed to be different from places in other nets.

Remark 2.1. If N is a dynamic net then $m_{0N} \in \mathcal{M}_{S_N,S_N}$.

The net N presented in Example 2.2 is closed, and hence dynamic, even though N_1 is not. In fact, the names v and w are not bound in N_1.

Remark 2.2. As variables in a transition are used to describe parameters, we consider only dynamic nets whose transitions $\mathtt{t} = m[\rangle N$ satisfy the condition $(fn(N) \cap \mathcal{X}) \subseteq rn(\mathtt{t})$. This restriction requires all free variables of the postset of a transition to be bound to some variable in the preset. Note that this is always the case when a net is closed.

Similarly to coloured nets, the firing of a transition $\mathtt{t}$ requires the instantiation of the received colours of $\mathtt{t}$, i.e., the formal parameters $rn(\mathtt{t})$ of the transition $\mathtt{t}$ have to be replaced by the actual parameters corresponding to the colours of the consumed tokens. Hence, we need suitable notions of substitution and instantiation of nets.

Definition 2.13 (Substitution). *Let $\sigma : \mathcal{X} \rightharpoonup \mathcal{X} \cup \mathcal{P}$ be a partial function. The* substitution σ *on a multiset $m \in \mathcal{M}_{\mathcal{C},\mathcal{C}}$ is given by*

$$(m\sigma)(\mathtt{a}_1)(c_1) = \textstyle\sum_{\mathtt{a} \in \{\mathtt{a}' | \mathtt{a}'\sigma = \mathtt{a}_1\}} \sum_{c \in \{c' | c'\sigma = c_1\}} \; m(\mathtt{a})(c)$$

Definition 2.14 (Instantiation of transitions and nets). *Let $\sigma : \mathcal{X} \rightharpoonup \mathcal{P} \cup \mathcal{X}$ be a substitution. The* instantiation of a transition *$\mathtt{t} = m_1[\rangle N_1$ with σ is the transition $\mathtt{t}\sigma = m_1\sigma[\rangle N_1\sigma$. Given a dynamic net $N = (S_N, T_N, \delta_{0N}, \delta_{1N}, m_{0N})$, the* instantiation of N *with σ s.t. $range(\sigma) \cap S_N = \emptyset$ is defined as $N\sigma = (S_N, T_N, \delta_{0N}, \delta_{1N}\sigma, m_{0N}\sigma)$, where $\delta_{1N}\sigma(\mathtt{t}) = (\delta_{1N}(\mathtt{t}))\sigma$ if $rn(\mathtt{t}) \cap (dom(\sigma) \cup range(\sigma)) = \emptyset$.*

Remark 2.3. (i) The recursive definition given above is well-founded because it is recursive on the structure of a net $N \in$ DN, which is well-founded. (ii) The conditions $range(\sigma) \cap S_N = \emptyset$ and $rn(\mathtt{t}) \cap (dom(\sigma) \cup range(\sigma)) = \emptyset$ avoid name

$$\frac{\texttt{t} = m\,[\rangle\,N_1 \in T \qquad m'' \in \mathcal{M}_{S,\mathcal{C}}}{(S,T,m\sigma \oplus m'') \to (S,T,m'') \otimes N_1\sigma}\ \text{(DYN-FIRING)} \quad \begin{array}{l} rn(\texttt{t}) \subseteq dom(\sigma) \text{ and} \\ range(\sigma) \subseteq S \end{array}$$

$$\frac{(S,T,m_1) \to (S,T,m'_1) \otimes N_1 \quad (S,T,m_2) \to (S,T,m'_2) \otimes N_2}{(S,T,m_1 \oplus m_2) \to (S,T,m'_1 \oplus m'_2) \otimes (N_1 \oplus N_2)}\ \text{(DYN-STEP)}$$

Fig. 5. Operational semantics of dynamic nets.

clashes. When such condition is not satisfied, α-conversion either on the places of the net or on the received names of the transition (as defined below) can be applied before the instantiation. (*iii*) Note that σ is not applied on δ_{0N}, since all variables appearing in the preset of a transition are local.

Definition 2.15 (α-equivalence). *Two transitions* $\texttt{t}_1$ *and* $\texttt{t}_2$ *are* α*-convertible if there exists an injective substitution* $\sigma : \mathcal{X} \rightharpoonup \mathcal{X}$*, with* $rn(\texttt{t}_1) \subseteq dom(\sigma)$*, s.t.* $\texttt{t}_1\sigma = \texttt{t}_2$*. As usual,* α*-conversion is a equivalence relation denoted by* $\equiv_\alpha$*. Two nets* N *and* N' *are* α*-convertible if there exist bijective substitutions* $\sigma_S : S_N \to S_{N'}$ *and* $\sigma_T : T_N \to T_{N'}$ *s.t.* ${}^\bullet\texttt{t}\sigma_S[\rangle\texttt{t}^\bullet\sigma_S \equiv_\alpha {}^\bullet(\texttt{t}\sigma_T)[\rangle(\texttt{t}\sigma_T)^\bullet$ *and* $m_{0N}\sigma_S = m_{0N'}$*. We shall always consider transitions and nets up-to* α*-equivalence.*

The following definition introduces two different ways for composing nets, which will be used to formalize the operational semantics of dynamic nets.

Definition 2.16 (Composition of nets). *Let* N_1*,* $N_2 \in$ DN *s.t.* $N_1 \cap N_2 = \emptyset$ *(i.e., they share neither places nor transitions) and* $fn(N_1) \cap S_{N_2} = \emptyset$ *(i.e.,* N_2 *does not define the free names of* N_1*). Then, the* addition *of* N_2 *to* N_1 *(written* $N_1 \otimes N_2$*) is defined as* $N_1 \otimes N_2 = (S_{N_1} \uplus S_{N_2}, T_{N_1} \uplus T_{N_2}, \delta_{0N_1} \uplus \delta_{0N_2}, \delta_{1N_1} \uplus \delta_{1N_2}, m_{0N_1} \oplus m_{0N_2})$*. The addition* $N_1 \otimes N_2$ *is said the* parallel composition *of* N_1 *and* N_2 *(written* $N_1 \oplus N_2$*) if also* $fn(N_2) \cap S_{N_1} = \emptyset$*.*

Observe that the side conditions required by parallel composition avoid free names of one net to be captured by the transitions defined by the other. Nevertheless, when a subnet N_2 is added to a net N_1 ($N_1 \otimes N_2$) we allow the free names of N_2 to be captured by the definitions in N_1. We remind that we are considering nets up-to α-conversion on the name of places and transitions. Hence, it is always possible to choose N'_2 α-equivalent to N_2 s.t. $N_1 \cap N'_2 = \emptyset$.

Rules defining the operational semantics of dynamic nets is in Figure 5. Note that the state of a computation considers not only markings but also the structure of the net. For simplicity we write (S,T,m) as a shorthand for $(S,T,\delta_0,\delta_1,m)$. Rule (DYN-FIRING) stands for the firing of $\texttt{t}$ when the marking contains an instance of the preset of $\texttt{t}$ (for a suitable substitution σ). The resulting net consists of the original net, where the consumed tokens have been removed, and a new instance of N_1 (i.e., the postset of $\texttt{t}$) has been added. The composition $\otimes$ of nets assures the names of the added components to be fresh. The side condition $rn(\texttt{t}) \subseteq dom(\sigma)$ assures all formal parameters of $\texttt{t}$ to be instantiated, while $range(\sigma) \subseteq S$ guarantees all consumed tokens to be coloured with names defined by the net. Rule (DYN-STEP) stands for the parallel composition of computations when the initial marking contains enough tokens for executing them independently. Note that both concurrent steps operate over the same net structure, in fact both start from a net whose places and transitions are S and T. Those steps can add new elements (i.e., N_1 and N_2), which by definition are fresh. Moreover, the new components can be chosen to assure new elements to be disjoint, i.e., such that $(N_1 \oplus N_2)$ is defined.

Remark 2.4 (Subject reduction). When starting from a dynamic net, the application of rules (DYN-FIRING) and (DYN-STEP) generates dynamic nets.

Example 2.3. Consider the dynamic net presented in Example 2.1 (see Figure 3(a)) but with the initial marking $m' = \mathtt{a}(\bullet) \oplus \mathtt{a}(\bullet)$. In what follows we show a computation that fires concurrently two instances of $\mathtt{t}$. We have

$$\frac{\dfrac{\mathtt{a}(\bullet)[\rangle N_1}{(\{\mathtt{a}\},\{\mathtt{t}\},\mathtt{a}(\bullet)) \to (\{\mathtt{a}\},\{\mathtt{t}\},\emptyset) \otimes N_1'} \quad \dfrac{\mathtt{a}(\bullet)[\rangle N_1}{(\{\mathtt{a}\},\{\mathtt{t}\},\mathtt{a}(\bullet)) \to (\{\mathtt{a}\},\{\mathtt{t}\},\emptyset) \otimes N_1''}}{(\{\mathtt{a}\},\{\mathtt{t}\},\mathtt{a}(\bullet) \oplus \mathtt{a}(\bullet)) \to (\{\mathtt{a}\},\{\mathtt{t}\},\emptyset) \otimes (N_1' \oplus N_1'')}$$

where $N_1' = (\{\mathtt{d}\},\{\mathtt{t}_1\},\mathtt{a}(\bullet) \oplus \mathtt{d}(\bullet))$ and $N_1'' = (\{\mathtt{d}'\},\{\mathtt{t}_1'\},\mathtt{a}(\bullet) \oplus \mathtt{d}'(\bullet))$. Note that $N_1' \oplus N_1'' = (\{\mathtt{d},\mathtt{d}'\},\{\mathtt{t}_1,\mathtt{t}_1'\},\mathtt{a}(\bullet) \oplus \mathtt{d}(\bullet) \oplus \mathtt{a}(\bullet) \oplus \mathtt{d}'(\bullet))$. And finally,

$$(\{\mathtt{a}\},\{\mathtt{t}\},\emptyset) \otimes (N_1' \oplus N_1'') = (\{\mathtt{a},\mathtt{d},\mathtt{d}'\},\{\mathtt{t},\mathtt{t}_1,\mathtt{t}_1'\},\mathtt{a}(\bullet) \oplus \mathtt{d}(\bullet) \oplus \mathtt{a}(\bullet) \oplus \mathtt{d}'(\bullet)).$$

3 Unfolding Dynamic nets

In this section we characterise the unfolding of dynamic nets. Different from P/T nets, the unfolding of a dynamic net should consider, not only the evolution of markings but also, the changes on the structure of the net. Intuitively, the unfolding of a dynamic net gives two different structures: (i) the places and transitions of the unfolded dynamic net, and (ii) the occurrence net describing the evolution of the states (i.e., markings).

Definition 3.1 (Dynamic net unfolding). *Let $N = (S_N, T_N, \delta_{0N}, \delta_{1N}, m_N)$ be a dynamic net. The unfolding $\mathcal{U}[N] = (S, T, \delta_0, \delta_1, \mathcal{S}, \mathcal{T}, \xi_0, \xi_1)$ is the joint combination of an occurrence net $(S, T, \delta_0, \delta_1)$ and a dynamic net $(\mathcal{S}, \mathcal{T}, \xi_0, \xi_1, \emptyset)$, which are defined inductively by the rules in Figure 6.*

In the above definition $(S, T, \delta_0, \delta_1)$ is the causal net describing the evolution of the marking of the net, while $(\mathcal{S}, \mathcal{T}, \xi_0, \xi_1)$ gives the structure of the unfolded dynamic net. Note that both structures are inductively defined starting from the original dynamic net N: At the beginning (by rules INI-PL and INI-TR) the unfolded structure of the dynamic net $(\mathcal{S}, \mathcal{T}, \xi_0, \xi_1)$ coincides with the definition of N, i.e., with (S_N, T_N, δ_{0N} and δ_{1N}). Initially, the causal net contains the places corresponding to the initial marking (rule INI-MK), i.e., for any token $\mathtt{a}(c)$

$$
\frac{\mathtt{a} \in S_N}{\mathtt{a} \in \mathcal{S}}\ \text{(INI-PL)} \qquad \frac{\mathtt{t} \in T_N}{\mathtt{t} \in \mathcal{T}, \quad \xi_0(\mathtt{t}) = \delta_{0N}(\mathtt{t}), \quad \xi_1(\mathtt{t}) = \delta_{1N}(\mathtt{t})}\ \text{(INI-TR)} \qquad \frac{m_N(\mathtt{a})(c) = n}{\{(\emptyset, \mathtt{a}(c))\} \times [n] \subseteq S}\ \text{(INI-MK)}
$$

$$
\text{(PRE)}\quad \frac{B = \{(\epsilon_j, b_j, i_j) | j \in J\} \subseteq S,\ Co(B),\ \mathtt{t} \in \mathcal{T},\ \xi_0(\mathtt{t})\sigma = \oplus_{j\in J} b_j}{(B, \sigma, \mathtt{t}) \in T, \quad \delta_0(B, \sigma, \mathtt{t}) = B}
$$

$$
\text{(POST)}\quad \frac{x = (B, \sigma, \mathtt{t}) \in T,\ \ \xi_1(\mathtt{t}) = N_1}{\begin{array}{c} Q = \{(\{x\}, b(c), i) \mid 0 < i \leq m_{N_1}\rho_x\sigma(\mathtt{b})(c)\} \subseteq S, \quad \delta_1(x) = Q, \quad S_{N_1}\rho_x \subseteq \mathcal{S}, \\ T_{N_1}\rho_x \subseteq \mathcal{T}, \quad \textit{for } \mathtt{t} \in T_{N_1} : \xi_0(\mathtt{t}\rho_x) = \delta_{0N_1}(\mathtt{t})\rho_x, \xi_1(\mathtt{t}\rho_x) = \delta_{1N_1}(\mathtt{t})\rho_x\sigma \end{array}}
$$

Fig. 6. Definition of $\mathcal{U}[N]$

in m_{0N}, there is one place in $\mathcal{U}[N]$ named $(\emptyset, \mathtt{a}(c), i)$ for some i. Note that the name of a place contains a set (in this case $\emptyset$) that records the history (i.e., the immediate causes) of the particular token associated to that place. In this case $\emptyset$ means that the associated token is part of the initial marking of N (i.e., it has not been generated by a transition). The part $\mathtt{a}(c)$ of the name provides the correspondence with the original token (it identifies the corresponding place and colour in N), while the number i univocally identifies one token $\mathtt{a}(c)$ when the initial marking contains several copies.

Then, the rules PRE and POST unfold the net by defining the transitions $x = (B, \sigma, \mathtt{t})$ of the causal structure. In fact, rule PRE identifies the set of places B corresponding to concurrent events (i.e., $CO(B)$) in the causal structure that conform an instance (for a substitution σ) of the preset of some transition $\mathtt{t}$ in the dynamic structure $\mathcal{T}$. For any such B, a new transition $(B, \sigma, \mathtt{t})$ is added to the causal net. Note that the preset of $(B, \sigma, \mathtt{t})$ is B. We assume w.l.o.g. that $dom(\sigma) = rn(\mathtt{t})$, i.e., we consider only the relevant part of substitutions.

Rule POST updates both the dynamic structure and the causal net. Consider transition $x = (B, \sigma, \mathtt{t})$ s.t. the postset of $\mathtt{t}$ is N_1 (i.e., $\xi_1(\mathtt{t}) = N_1$). Rule POST adds a *fresh* instance of N_1 to the dynamic structure and the events for the initial marking of N_1 to the causal net. Freshness is obtained by applying to every element of S_{N_1} and T_{N_1} the renaming ρ_x, which is defined as follow

$$
\forall z \in S_{N_1} \cup T_{N_1} : \rho_x(z) = z_x
$$

Note this renaming adds the name of the fired transition x to any element generated by x (i.e., its history). Hence, by applying ρ_x we assure new elements in the generated instance of N_1 to be fresh (and unique).

Then, the causal structure is modified as follows: for any place in the initial marking of N_1 a new place is generated in S. Note the name of such places carries $\{x\}$ as its history (i.e., the name of the transition that generates them), the name of the token $\mathtt{b}(c)$ corresponding to the tokens in the initial marking of N_1 suitably renamed by ρ_x (i.e., the names of the fresh instance of N_1) and

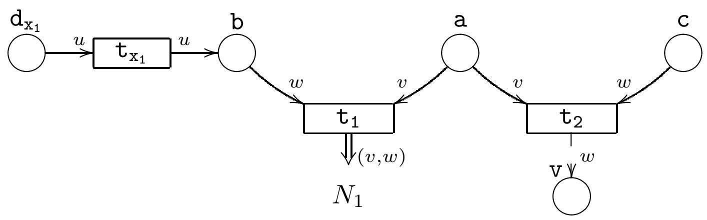


Fig. 7. The dynamic part of the unfolding $\mathcal{U}[N]$: ($\mathcal{S}$,$\mathcal{T}$,ξ_0,ξ_1).

instantiated by σ (i.e., the colour of the consumed tokens). Moreover, the postset of x is defined as the set Q of all generated places.

Furthermore, rule POST refreshes the structure of the dynamic net by adding the elements of the fresh instance of N_1 (i.e., $S_{N_1}\rho_x$ and $T_{N_1}\rho_x$), and updating the flow relations ξ_i accordingly. In particular, note that σ is also used when defining ξ_1. This is necessary in cases as the following: consider the transition $\mathtt{t} = \mathtt{a}(x)[\rangle N_1$, s.t. N_1 defines the transition $\mathtt{t_1} = \mathtt{b}(y)[\rangle x(y)$. Since x is free in $\mathtt{t_1}$ (and hence in N_1), x is bound to the occurrence of x in the preset of $\mathtt{t}$. Consequently, if $\mathtt{t}$ is fired with the substitution σ, then σ has to be applied to the transitions in N_1. In contrast with this, we do not use σ for ξ_0, since the substitution has no effect on the presets of transitions.

Remark 3.1. (i) The unfolding construction works since transitions generated dynamically can be seen as resources that are concurrent to tokens. When a new transition is added to $\mathcal{T}$ by using rule POST, we are sure that such transition cannot fetch tokens from already existing places. Hence, all elements generated previously are still valid. (ii) The unfolding is unique since the result is influenced neither by the order in which productions are applied nor by the multiple applications of a same rule that produce always the same elements.

Example 3.1. Consider the dynamic net N defined in Example 2.2. Its unfolding $\mathcal{U}[N]$ is depicted in Figures 7 and 8.

Proposition 3.1. *Let N be a dynamic net and $\mathcal{U}[N] = (S, T, \delta_0, \delta_1, \mathcal{S}, \mathcal{T}, \xi_0, \xi_1)$ the unfolding of N. Then $(S, T, \delta_0, \delta_1)$ is an occurrence net.*

Proof. By induction on the structure of the proof $x \in T$.

Proposition 3.2. *Let N be a dynamic net and $\mathcal{U}[N] = (S, T, \delta_0, \delta_1, \mathcal{S}, \mathcal{T}, \xi_0, \xi_1)$ the unfolding of N. Then $(\mathcal{S}, \mathcal{T}, \xi_0, \xi_1, \emptyset)$ is a dynamic net.*

Proof. By induction on the structure of the proof that $\mathtt{t} \in \mathcal{T}$.

The unfolding for the case of an ordinary P/T net (when regarded as a dynamic net) essentially coincides with the ordinary notion of unfolding, since (i) tokens carry on the colour $\bullet$, which determines a unique possible substitution $\sigma = \emptyset$ in rule PRE, and (ii) rule POST does not modify the dynamic structure.

The dynamic net unfolding $\mathcal{U}[N] = (S, T, \delta_0, \delta_1, \mathcal{S}, \mathcal{T}, \xi_0, \xi_1)$ defines an implicit morphism from the occurrence net $(S, T, \delta_0, \delta_1)$ to $(\mathcal{S}, \mathcal{T}, \xi_0, \xi_1, \emptyset)$, which

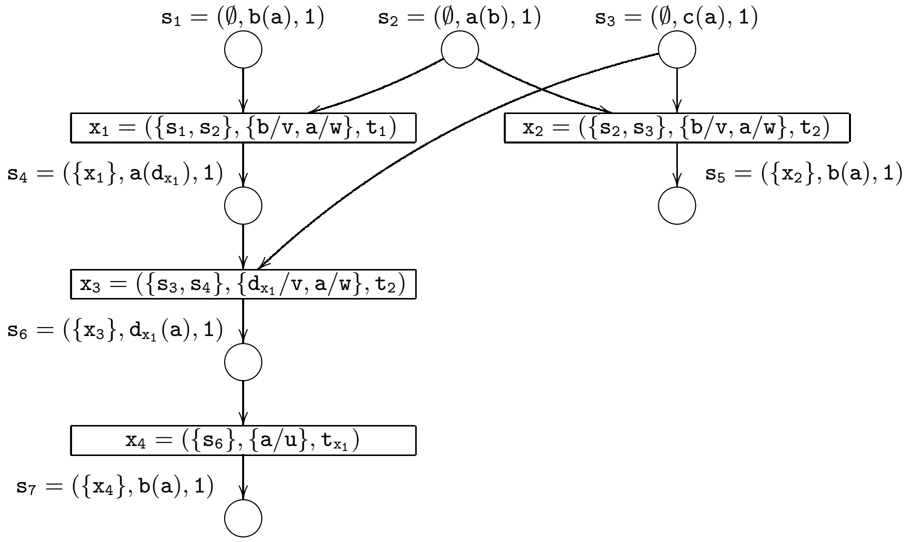


Fig. 8. The causal structure of the unfolding $\mathcal{U}[N]$: (S,T,δ_1,δ_2).

is given by the projections of places (ϵ, b, i) to the second component b and of transitions $(B, \sigma, \mathtt{t})$ to the third component $\mathtt{t}$. Exploiting this fact, we can define the notion of deterministic processes of a dynamic net as below.

Definition 3.2 (Process of a dynamic net). *A* deterministic process *for a dynamic net N (written $P : K \rightsquigarrow N$) is a net morphism P from a causal net K to $C = (S, T, \delta_0, \delta_1)$ s.t. $P(^\circ K) = {}^\circ C$, where $\mathcal{U}[N] = (S, T, \delta_0, \delta_1, \mathcal{S}, \mathcal{T}, \xi_0, \xi_1)$.*

We note the set of origins and destinations of $P : K \to N$ respectively by $O(P) = {}^\circ K$ and $D(P) = K^\circ \cap S_K$. Moreover, $pre(P)$ and $post(P)$ stand for the multisets of initial and final markings of P, i.e., $pre(P) = \bigoplus_{(x,b,i)\in P(O(P))} b$ and $post(P) = \bigoplus_{(x,b,i)\in P(D(P))} b$.

The following result relates the operational semantics of dynamic nets with their non-sequential semantics. In particular, we show that, although processes capture more information about the behaviour of a net than reductions, computations in both cases are the same. (We recall that dynamic nets are considered up-to α-conversion).

Theorem 3.1 (Correspondence). *Let N, N' be dynamic nets. Then $N \to^* N'$ iff there exists a process $P : K \rightsquigarrow N$ such that:*

(i) $pre(P) = m_{0N}$ and $post(P) = m_{0N'}$, and

(ii) all places and transitions in N' are either in N or they are added by the computation described by P, i.e., $N \oplus \bigoplus_{x=(B,\sigma,\mathtt{t})\in P(T_K)} \mathtt{t}^\bullet(\rho_x, \sigma) = (S_{N'}, T_{N'}, \delta_{0N'}, \delta_{1N'}, m)$, where substitution $\mathtt{t}^\bullet(\rho, \sigma)$ for $\mathtt{t}^\bullet = N_i$ is defined as $(S_i, T_i, \delta_{i0}, \delta_{i1}, m_{i0})(\rho, \sigma) = (S_i\rho, T_i\rho, \delta_{i0}\rho, \delta_{i0}\rho\sigma, m_{i0}\rho\sigma)$.

(iii) Moreover, for the unfolding $\mathcal{U}[N] = (S, T, \delta_0, \delta_1, \mathcal{S}, \mathcal{T}, \xi_0, \xi_1)$ of N, we have $S_{N'} \subseteq \mathcal{S}, T_{N'} \subseteq \mathcal{T}, \delta_{0N'} \subseteq \xi_0, \delta_{1N'} \subseteq \xi_1$.

Proof. $\Rightarrow$) It follows by induction on the length n of the derivation $N \rightarrow^n N'$.

- **Base case ($n = 0$):** Hence $N = N'$. Conditions $(i)-(iii)$ hold by taking $K = (S_K, \emptyset, \emptyset, \emptyset)$ with $S_K = \bigcup_{\mathtt{a}(c) \in dom(m_{0N}))} (\emptyset, \mathtt{a}(c)) \times [m_{0N}(\mathtt{a})(c)]$, and $P : K \rightarrow (S, T, \delta_0, \delta_1)$ as the identity on places and transitions.
- **Inductive Step ($n = k+1$):** Then, $N \rightarrow^k N'' \rightarrow N'$. By inductive hypothesis on $N \rightarrow^k N''$, $\exists P'$ satisfying $(i)-(iii)$. The proof follows by showing (by rule induction on the structure of the proof $N'' \rightarrow N'$) that $\exists P$, which is an extension of P', satisfying conditions $(i)-(iii)$.
  - ∗ **Rule** (DYN-FIRING): Then, $N'' \rightarrow N'$ by firing $\mathtt{t} \in T_{N''}$. Note the final elements of P' contains a set M of concurrent elements corresponding to an instance of the preset of $\mathtt{t}$. Since elements in M are concurrent, the unfolding contains an event e corresponding to the firing of $\mathtt{t}$ with preset M. P adds e and its postset to P'.
  - ∗ **Rule** (DYN-STEP): By inductive hypothesis, there are two processes P_1 and P_2 starting from two different markings m_1 and m_2 of the same net N. Although, P_1 and P_2 are processes of two different unfoldings, they can be seen as being processes of the same unfolding of N with initial marking $m_1 \oplus m_2$. W.l.o.g, we assume K_1 and K_2 only to share initial elements that are not in conflict. Then, $K_1 \cup K_2$ is a causal net. Finally, P is defined as the union of P_2 and P_2.

  In both cases, conditions can be verified by straightforward analysis.

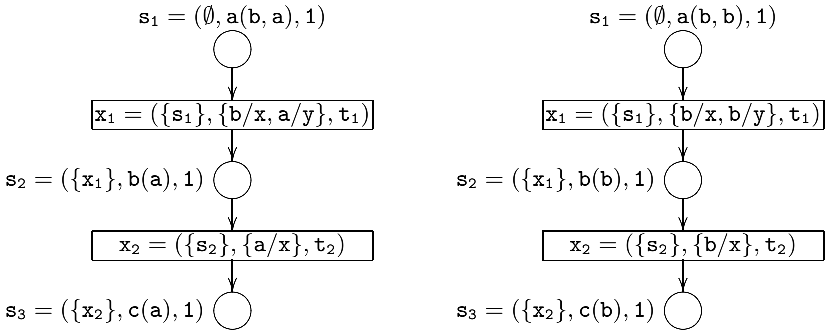
$\Leftarrow$) The proof follows by induction on the number of transitions of K.

- **Base Case** ($|T_K| = 0$). It follows immediately by taking $N' = N$.
- **Inductive Step**. $T_K = T_{K'} \cup \mathtt{t}$ s.t. $\mathtt{t}^\bullet \subseteq K^\circ$ Let K' be the causal net obtained from removing $\mathtt{t}$ and $\mathtt{t}^\bullet$ from K, and P' the restriction of P to the elements of K'. By defining N'' as
  - ∗ $(S_{N''}, T_{N''}, \delta_{0N''}, \delta_{0N''}, m) = N \oplus \bigoplus_{x=(B,\sigma,\mathtt{t}) \in P(T_{K'})} \mathtt{t}^\bullet(\rho_x, \sigma)$, and
  - ∗ $m_{0N''} = post(P') = post((D(P) \cup {}^\bullet\mathtt{t}) \backslash \mathtt{t}^\bullet)$

  and using inductive hypothesis on P', we have $N \rightarrow N''$. By condition (iii) and the definition of P, there exists $\mathtt{t} \in N''$ whose firing makes N'' to become N'.

4 Unfolding pattern

The definitions of unfolding and deterministic processes for dynamic nets given in the previous section depend on the colours carried on by tokens. The main disadvantage of such kind of definitions is that they cannot abstract away from colours that are irrelevant for the computation. Consider the net N, where $S_N = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $T_N = \{\mathtt{t}_1, \mathtt{t}_2\}$ s.t. $\mathtt{t}_1 = \mathtt{a}(x, y)[\rangle x(y)$, $\mathtt{t}_2 = \mathtt{b}(x)[\rangle \mathtt{c}(x)$. When constructing the unfoldings for the initial markings $m_1 = \mathtt{a}(\mathtt{b}, \mathtt{a})$ and $m_2 = \mathtt{a}(\mathtt{b}, \mathtt{b})$, we obtain isomorphic causal nets as shown in Figure 9 (we omit the representation of the dynamic structure because it stays N in both cases). To quotient out the set of all unfoldings with isomorphic causal nets and the same dynamic structure, we generalise the notions given in the previous section by allowing unfolding and processes to be parametric in (some) colours carried on by tokens of the initial marking. We call such constructions *unfolding* and *process patterns*.



(a) $m_1 = \mathtt{a}(\mathtt{b},\mathtt{a})$. (b) $m_2 = \mathtt{a}(\mathtt{b},\mathtt{b})$.

Fig. 9. Causal nets of the Unfolding of N for different initial markings m_1 and m_2.

(INI-PL-PATT)
$$\frac{\mathtt{a} \in S_N}{\mathtt{a} \in \mathcal{S}}$$

(INI-TR-PATT)
$$\frac{\mathtt{t} \in T_N}{\mathtt{t} \in \mathcal{T}, \quad \xi_0(\mathtt{t}) = \delta_{0N}(\mathtt{t}), \quad \xi_1(\mathtt{t}) = \delta_{1N}(\mathtt{t})}$$

(INI-MK-PATT)
$$\frac{m_N(\mathtt{a})(c) = n}{\{(\emptyset, \mathtt{a}(c), \emptyset)\} \times [n] \subseteq S}$$

(PRE-PATT)
$$\frac{\begin{array}{c} B = \{(\epsilon_j, b_j, \mu_j, i_j) | j \in J\} \subseteq S, \quad Co(B), \quad \mathtt{t} \in \mathcal{T}, \quad \xi_0(\mathtt{t})\sigma = \oplus_{j \in J} b_j \mu_{\mathtt{t}}, \\ range(\mu_{\mathtt{t}}) \subseteq S_N, \quad \mu = \mu_{\mathtt{t}} \cup \bigcup_j \mu_j \ well-defined\ substitution \end{array}}{(B, \sigma, \mathtt{t}, \mu) \in T, \quad \delta_0(B, \sigma, \mathtt{t}, \mu) = B}$$

(POST-PATT)
$$\frac{x = (B, \sigma, \mathtt{t}, \mu) \in T, \;\; \xi_1(\mathtt{t}) = N_1}{\begin{array}{c} Q = \{(\{x\}, \mathtt{b}(c), \mu, i) \mid 0 < i \leq m_{N_1}\rho_x\sigma(\mathtt{b})(c)\} \subseteq S, \quad \delta_1(x) = Q, \quad S_{N_1}\rho_x \subseteq \mathcal{S}, \\ T_{N_1}\rho_x \subseteq \mathcal{T}, \quad for\ \mathtt{t} \in T_{N_1} : \xi_0(\mathtt{t}\rho_x) = \delta_{0N_1}(\mathtt{t})\rho_x, \xi_1(\mathtt{t}\rho_x) = \delta_{1N_1}(\mathtt{t})\rho_x\sigma \end{array}}$$

Fig. 10. Definition of $\mathcal{UP}[N]$

Definition 4.1 (Pattern). *A* (marking) pattern *is a multiset* $p \in \mathcal{M}_{\mathcal{P}\cup\mathcal{X},\mathcal{P}\cup\mathcal{X}}$. *Moreover, p is a* colour pattern *if* $p \in \mathcal{M}_{\mathcal{P},\mathcal{P}\cup\mathcal{X}}$, *i.e., variables appear only as colours. A colour pattern p is* linear *if any variable occur at most once in p.*

Definition 4.2 (Unfolding pattern). *Let* $N = (S_N, T_N, \delta_{0N}, \delta_{1N}, m_N) \in$ DN, *s.t. the initial marking m_N is a linear colour pattern. The* unfolding pattern $\mathcal{UP}[N] = (S, T, \delta_0, \delta_1, \mathcal{S}, \mathcal{T}, \xi_0, \xi_1)$ *is defined inductively by the rules in Figure 10.*

The three first rules are quite similar to the definition of unfolding in the previous section. The main difference is that place names (rule INI-MK-PATT) carry on an extra element (the third one) that records all substitutions made on variable colours of the initial pattern. Clearly, for tokens corresponding to the initial marking, the third element is set to the empty substitution.

As far as the unfolding is concerned, it works analogously to the previous case. The main difference is that when identifying the preset B of a transition (in rule PRE-PATT) it may be the case that $\bigoplus_j b_j$ is actually a pattern more general

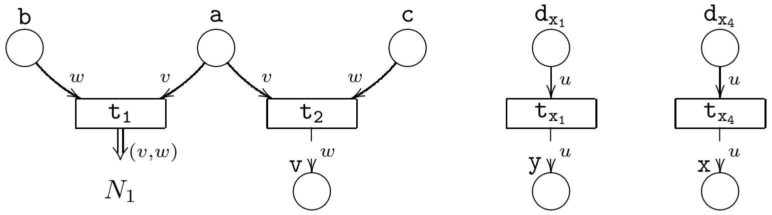


Fig. 11. The dynamic part of the unfolding pattern $\mathcal{UP}[N]$: $(\mathcal{S},\mathcal{T},\xi_0,\xi_1)$.

than the preset of the transition we would like to apply, hence its variables may require to be instantiated in order to conform the preset of a transition. Consider the simple case of N containing only the transition $\mathtt{t} = \mathtt{a}(\mathtt{b}, x)[\rangle\mathtt{b}(\mathtt{a})$. When considering the initial pattern $m_{0N} = \mathtt{a}(v, w)$ the causal net of the unfolding contains the place $\mathtt{s_1} = (\emptyset, \mathtt{a}(v, w), \emptyset, 1)$ (i.e., $\mathtt{s_1} \in S$). Since $\mathtt{t} \in \mathcal{T}$ we would like to fire $\mathtt{t}$ with the token $\mathtt{s_1}$. To make $\mathtt{a}(v, w)$ be an instance of ${}^\bullet\mathtt{t}$ we need to substitute the variable v of $\mathtt{a}(v, w)$ by $\mathtt{b}$, and substitute the variable x in ${}^\bullet\mathtt{t}$ by w. Hence, we take $\mu_\mathtt{t} = \{\mathtt{b}/v\}$ and $\sigma = \{w/x\}$, so that $\mathtt{a}(v, w)\mu_\mathtt{t} = \mathtt{a}(\mathtt{b}, w) = {}^\bullet\mathtt{t}\sigma$.

Moreover, we require $\mu_\mathtt{t}$ to be *minimal* in order for the unfolding to describe the most general pattern of computation. That is, we will not consider substitutions $\mu_\mathtt{t}$ that instantiate more variables than those needed. In the previous example we do not consider the substitutions $\theta_1 = \{\mathtt{b}/v, \mathtt{a}/w\}$ and $\theta_2 = \{\mathtt{b}/v, \mathtt{b}/w\}$ as candidate $\mu_\mathtt{t}$ (although they also satisfy the equation $\mathtt{a}(v, w)\theta_i = {}^\bullet\mathtt{t}\sigma_i$ with $\sigma_1 = \{\mathtt{a}/x\}$ and $\sigma_2 = \{\mathtt{b}/x\}$) because they are not minimal instantiations. In particular, $\mu_\mathtt{t}$ means that there not exists μ' satisfying the instantiation condition and $\mu_\mathtt{t} = \mu' \circ \mu''$, with $\mu'' \neq \emptyset$. Substitutions renaming variables are not possible $\mu_\mathtt{t}$. In such cases the renaming occurs in σ, as in the previous example.

Since colours in any marking refer to places of the net (i.e., names in S_N) and $\mu_\mathtt{t}$ is an instantiation of a pattern corresponding to an initial marking, we require $range(\mu_\mathtt{t}) \subseteq S_N$. Additionally, this condition assures that if a variable appearing in place position is instantiated, then the chosen transition $\mathtt{t}$ belongs to T_N (because transitions generated dynamically cannot fetch messages from already existing places). Consequently, we are also sure that the applied transition is causally independent of the consumed tokens.

Condition $\mu = \mu_\mathtt{t} \cup \bigcup_j \mu_j$ *well − defined substitution* assures that all instantiations made by concurrent computations on pattern variables are consistent.

Example 4.1. Consider the dynamic net N presented in Example 3.1. The unfolding pattern for the marking $\mathtt{a}(y) \oplus \mathtt{b}(x) \oplus \mathtt{c}(z)$ is shown in Figures 11 and 12.

While the unfolding is defined for a concrete marking (i.e., all places and colours are constants), an unfolding pattern defines a set of different unfoldings, one for any possible (compatible) instantiation of the pattern.

Definition 4.3 (Instance of an unfolding pattern). *Let $\mathcal{UP}[N]$ the unfolding pattern of $N = (S_N, T_N, \delta_{0N}, \delta_{1N}, p)$, where p is a linear pattern, and*

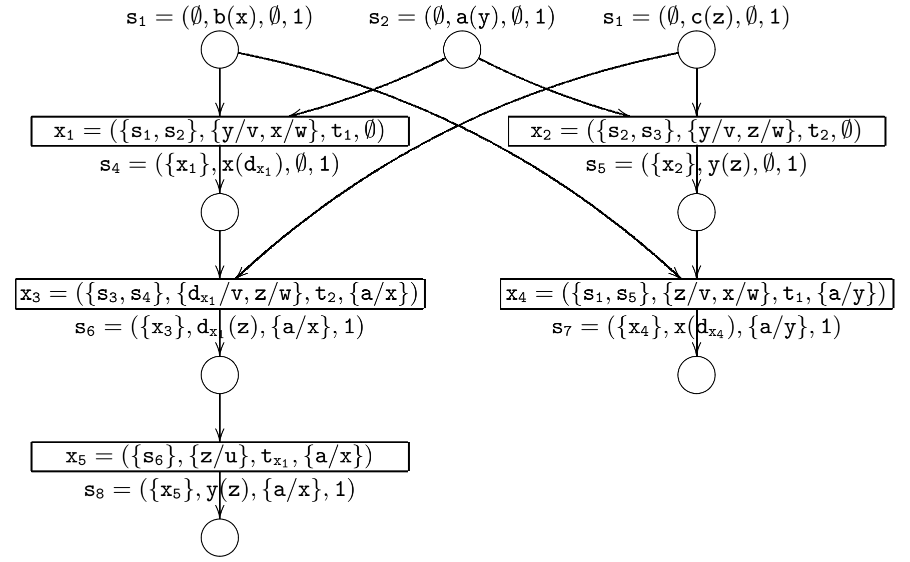


Fig. 12. The causal part of the unfolding pattern $\mathcal{UP}[N]$: (S,T,δ_1,δ_2).

$$\frac{(H, \mathtt{a}(c), \mu, n) \in S,\ (\mu \cup \theta)\ well-defined}{(H, \mathtt{a}(c), \mu, n) \in S\theta} \qquad \frac{\mathtt{a}_x \in \mathcal{S},\ x \in T\theta}{\mathtt{a}_x \in \mathcal{S}\theta} \qquad \frac{\mathtt{x} \in T\theta}{\delta\theta_i(\mathtt{x}) = \delta_i(\mathtt{x})}$$

$$\frac{(B, \sigma, \mathtt{t}, \mu) \in T,\ (\mu \cup \theta)\ well-defined}{(B, \sigma, \mathtt{t}, \mu) \in T\theta} \qquad \frac{\mathtt{t}_x \in \mathcal{T},\ x \in T\theta}{\mathtt{t}_x \in \mathcal{T}\theta} \qquad \frac{\mathtt{t} \in \mathcal{T}\theta}{\xi\theta_i(\mathtt{t}) = \xi_i(\mathtt{t})}$$

Fig. 13. Definition of $\mathcal{UP}[N]\theta$, the θ-instance of $\mathcal{UP}[N]$.

a substitution θ s.t. $dom(\theta) \subseteq col_{\mathcal{X}}(p)$ and $range(\theta) \subseteq S_N$. The instance of $\mathcal{UP}[N]$ by θ, written $\mathcal{UP}[N]\theta = (S\theta, T\theta, \delta\theta_0, \delta\theta_1, \mathcal{S}\theta, \mathcal{T}\theta, \xi\theta_0, \xi\theta_1)$, is given by the inference rules in Figure 13.

Note that $\mathcal{UP}[N]\theta$ is obtained by removing all elements of $\mathcal{UP}[N]$ that are incompatible with θ, i.e., elements instantiating the variables of p differently from θ. The following result relates unfoldings and unfolding patterns by showing that the unfolding corresponding to an initial marking $m_0 = p\theta$ can be obtained by instantiating the unfolding pattern for p with θ.

Lemma 4.1. *Let $\mathcal{U}[N] = (S, T, \delta_0, \delta_1, \mathcal{S}, \mathcal{T}, \xi_0, \xi_1)$ be the unfolding of N, and $\mathcal{UP}[N] = (S_P, T_P, \delta_{0P}, \delta_{1P}, \mathcal{S}_P, \mathcal{T}_P, \xi_{0P}, \xi_{1P})$ be the unfolding pattern of N for the linear pattern p s.t. $m_{0N} = p\theta$, and $\mathcal{UP}[N]\theta$ the corresponding instance of $\mathcal{UP}[N]$. Then, there exists a bijective function $f = (f_S : S \to S_P\theta, f_T : T \to T_P\theta, f_{\mathcal{S}} : \mathcal{S} \to \mathcal{S}_P\theta, f_{\mathcal{T}} : \mathcal{T} \to \mathcal{T}_P\theta)$ such that:*

1. *$\forall \mathtt{t} \in \mathcal{T} : f_{\mathcal{S}}({}^\bullet\mathtt{t})[\rangle f_{\mathcal{S}}(\mathtt{t}^\bullet) \equiv_\alpha {}^\bullet f_{\mathcal{T}}(\mathtt{t})[\rangle f_{\mathcal{T}}(\mathtt{t})^\bullet$, i.e., dynamic structures are isomorphic;*

2. $\forall \mathtt{x} \in T : f_S({}^\bullet\mathtt{x})[\rangle f_S(\mathtt{x}^\bullet) = {}^\bullet f_T(\mathtt{t})[\rangle f_T(\mathtt{t})^\bullet$*, i.e., causal nets are isomorphic;*
3. $\forall \mathtt{x} = (B, \sigma, \mathtt{t}) \in T : f_T(\mathtt{x}) = (B', \sigma', f_{\mathcal{T}}(\mathtt{t}), \mu)$*, i.e., mapped events correspond to the same transition;*
4. $\forall \mathtt{s} = (H, \mathtt{a}(c), i) \in S : f_S(\mathtt{s}) = (H', \mathtt{b}(c'), \mu, j)$ *and* $\mathtt{a}(c) = f_{\mathcal{S}}(\mathtt{b}(c')\theta)$*, i.e., mapped places correspond to the same token.*

Proof. The proof follows by rule induction on the definition of $\mathcal{U}[N]$

Then, starting from the definition of the unfolding pattern, we give the notion of process pattern, which is a process parametric on the colours of the initial tokens. Such definition relies on the following notion of associated instantiation.

Definition 4.4 (Associated instantiation). *Given an* $\mathcal{UP}[N]$*, the instantiation* $inst(s)$ *associated to an element* e *of* $S \cup T$ *is defined as* $inst(H, \sigma, \mu, i) = \mu$ *and* $inst(H, \sigma, \mathtt{t}, \mu) = \mu$*. Given a set* $E \subseteq S \cup T$*,* $inst(E) = \{inst(e) | e \in E\}$*.*

The associated instantiations for a set E of elements belonging to an unfolding pattern collects all the instantiations made by the elements of E to the variables of the initial pattern. Then, a process pattern is defined as follows.

Definition 4.5 (Process Pattern). *A* process pattern *for a dynamic net* N *is a net morphism* PP *from a causal net* K *to the net* $(S, T, \delta_0, \delta_1)$*, where* $\mathcal{UP}[N] = (S, T, \delta_0, \delta_1, \mathcal{S}, \mathcal{T}, \xi_0, \xi_1)$ *for a particular pattern* p *as initial marking s.t.* $\forall \mu_1, \mu_2 \in inst(PP(D(PP))) : \mu_1 \cup \mu_2$ *is a well-defined substitution.*

The above definition is analogous to Definition 3.2 but it also requires the instantiations made by final elements to be consistent. This condition assures concurrent events in a process not to instantiate variables differently.

As for unfolding patterns, we allow process patterns to be instantiated.

Definition 4.6 (Compatible instantiation of a process). *A substitution* θ *is a* compatible instantiation *of a process pattern* PP *if* $\forall \mu \in inst(PP(D(PP)))$*,* $\mu \cup \theta$ *is a well-defined substitution, i.e.,* θ *is compatible with the instantiations of final elements of* P*.*

Finally, we show that process patterns generalise the definition of processes. We start by showing that an instance of a process pattern is a process of the corresponding instance of the unfolding pattern (Lemma 4.2). Then, we show that the processes of a marked net can be obtained as instantiations of suitable process patterns (Theorem 4.1).

Lemma 4.2. *Let* PP *be a process pattern of* N *for the unfolding pattern* $\mathcal{UP}[N]$*, and* θ *a compatible instantiation. Then,* PP *is a process of* $\mathcal{UP}[N]\theta$*.*

Proof. The proof follows from the fact that all elements of the unfolding that are image of PP are consistent with θ. Hence, they are also in $\mathcal{UP}[N]\theta$.

Theorem 4.1. *Let* PP *be a process pattern of* N *for the linear pattern* p *s.t.* $m_{0N} = p\theta$*. Then,* PP *describes also a process of* $\mathcal{U}[N]$*.*

Proof. By Lemma 4.2, PP with θ is a process of $\mathcal{UP}[N]\theta$. By Lemma 4.1, $\mathcal{U}[N]$ and $\mathcal{UP}[N]\theta$ are isomorphic, then there exists P (defined as the composition of PP and the function f of Lemma 4.1), which is a process of $\mathcal{U}[N]$.

5 Concluding Remarks

We have defined the semantic foundations for expressing and analyzing the non-sequential behaviour of dynamic nets. This work extends the line initiated in [4,11], where suitable notions of process and process pattern have been defined for coloured and reconfigurable nets. Here, we have extended the ordinary notion of unfolding and process to the more expressive setting of dynamic nets.

Since the unfolding semantics have been used for checking properties of, e.g., marked P/T nets and graph grammars [10,2], we expect the notions given in this paper will allow for the development of techniques for checking properties of dynamic nets. Moreover, unfolding patterns would also enable the study of properties at a more abstract level than unfoldings, for instance, to prove non reachability for a set of initial markings instead of just for a marking.

Since dynamic nets are in one-to-one correspondence with asynchronous join processes, this work provides a process semantics for an asynchronous name passing calculus. Hence, the notions presented in this paper can serve as a starting point for studying the non-sequential behaviour of asynchronous name passing calculus, for instance, for studying process equivalences by considering the true concurrent semantics of processes.

References

1. A. Asperti and N. Busi. Mobile Petri nets. Technical Report UBLCS 96-10, Computer Science Department, University of Bologna, 1996.
2. P. Baldan, A. Corradini, and B. König. A static analysis technique for graph transformation systems. *Proc. CONCUR'01*, *LNCS* 2154, pp. 381–395. Springer, 2001.
3. N. Benton, L. Cardelli, and C. Fournet. Modern concurrency abstractions for C$^\sharp$. *Proc. ECOOP'02*, *LNCS* 2374, pp. 415–440. Springer, 2002.
4. R. Bruni, H. Melgratti, and U. Montanari. Extending the zero-safe approach to coloured, reconfigurable and dynamic nets. *Lectures on Concurrency and Petri Nets, Advances in Petri Nets*, *LNCS* 3098, pp. 291–327. Springer, 2003.
5. M. Buscemi and V. Sassone. High-level Petri nets as type theories in the Join calculus. *Proc. FoSSaCS'01*, *LNCS* 2030, pp. 104–120. Springer, 2001.
6. S. Conchon and F. Le Fessant. Jocaml: Mobile agents for Objective-Caml. *Proc. ASA'99 / MA'99*, pp. 22–29. IEEE, 1999.
7. C. Fournet and G. Gonthier. The reflexive chemical abstract machine and the Join calculus. *Proc. POPL'96*, pp. 372–385. ACM Press, 1996.
8. U. Goltz and W. Reisig. The non-sequential behaviour of Petri nets. *Inform. and Comput.*, 57:125–147, 1983.
9. C. Gunter and D Scott. Semantic domains. *Handbook of Theoretical Computer Science, Vol. B: Formal Models and Sematics*, pp. 633–674. MIT Press, 1990.
10. K.L. McMillan. *Symbolic Model Checking*. Kluwer, 1993.
11. H. Melgratti. *Models and Languages for Global Computing Transactions*. PhD thesis, Computer Science Department, University of Pisa, 2005.
12. J. Meseguer, U. Montanari, and V. Sassone. Process versus unfolding semantics for place/transition Petri nets. *Theoret. Comput. Sci.*, 153(1-2):171–210, 1996.
13. C.A. Petri. *Kommunikation mit Automaten*. PhD thesis, Institut für Instrumentelle Mathematik, Bonn, 1962.