On the Computation of McMillan's Prefix for Contextual Nets and Graph Grammars*

Paolo Baldan[1], Alessandro Bruni[1], Andrea Corradini[2], Barbara König[3], and Stefan Schwoon[4]

[1] Dipartimento di Matematica Pura e Applicata, Università di Padova, Italy
[2] Dipartimento di Informatica, Università di Pisa, Italy
[3] Abteilung für Informatik und Angewandte Kognitionswissenschaft, Universität Duisburg-Essen, Germany
[4] LSV, ENS Cachan & CNRS, INRIA Saclay, France

Abstract. In recent years, a research thread focused on the use of the unfolding semantics for verification purposes. This started with a paper by McMillan, which devises an algorithm for constructing a finite complete prefix of the unfolding of a safe Petri net, providing a compact representation of the reachability graph. The extension to contextual nets and graph transformation systems is far from being trivial because events can have multiple causal histories. Recently, we proposed an abstract algorithm that generalizes McMillan's construction to bounded contextual nets without resorting to an encoding into plain P/T nets. Here, we provide a more explicit construction that renders the algorithm effective. To allow for an inductive definition of concurrency, missing in the original proposal and essential for an efficient unfolding procedure, the key intuition is to associate histories not only with events, but also with places. Additionally, we outline how the proposed algorithm can be extended to graph transformation systems, for which previous algorithms based on the encoding of read arcs would not be applicable.

1 Introduction

Partial-order semantics are used to alleviate the state-explosion problem when model checking concurrent systems. A thread of research started by McMillan [10, 11] proposes the *unfolding* semantics for the verification of finite-state systems, modelled as Petri nets. The unfolding of a Petri net [12] is a nondeterministic process of the net that completely expresses its behaviour; it is an acyclic but usually infinite net. McMillan's algorithm constructs a finite *prefix* of the unfolding, which is *complete*, i.e., each marking reachable in the original net is represented in the prefix. Computing a complete prefix can be seen as a preprocessing step for reachability analysis of a Petri net: a complete prefix is generally larger than the original Petri net, but smaller than the reachability

* Supported by the MIUR Project `SisteR`, by the project `AVIAMO` of the University of Padova and by the DFG project SANDS.

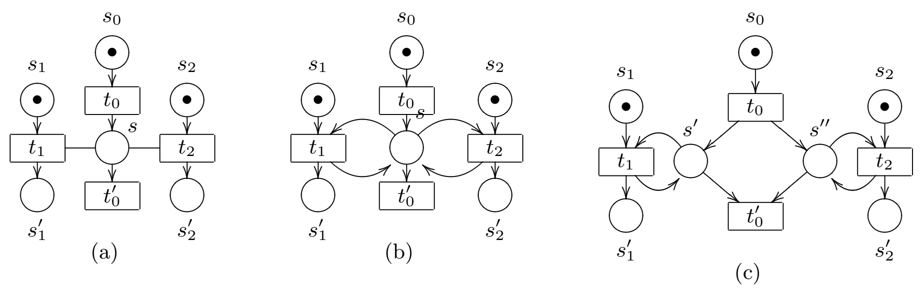


Fig. 1. (a) A safe contextual net; (b) its encoding by replacing read arcs with consume/produce loops; (c) its concurrency-preserving PR-encoding.

graph, and reachability queries can be answered efficiently on the prefix. Other applications of unfoldings are, e.g. in fault diagnosis [7] and in planning [8].

The unfolding construction has been generalized to other rule-based formalisms such as contextual nets [14, 4] and graph grammars [5]. However, concerning McMillan's construction, problems arise because these formalisms allow to preserve part of the state in a rewriting step. This has been observed originally for contextual nets by Vogler et al. [14]: they showed that for such nets the prefix generated by McMillan's algorithm might not be complete because events can have more than one causal history.

A solution to this problem is to encode contextual nets into an "equivalent" P/T net, to which McMillan's algorithm is then applied. Consider the net in Fig. 1 (a), where read arcs are drawn as undirected lines. Net (b) is obtained by replacing each read arc of (a) with a consume/produce loop, obtaining a net with the same reachability graph, but whose unfolding could grow exponentially due to the sequentialization imposed on readers. Net (c) is obtained like net (b), but first creating "private copies" of the read places for each reader: for safe nets this *Place-Replication (PR) encoding* preserves concurrency, and the size of the unfolding is comparable to that of the original net.

Unfortunately, such approaches are not viable for graph grammars nor, in general, for rewriting formalisms where states have a structure richer than multisets. Sticking to standard graph rewriting, if a rule preserves a node both encodings would transform it into a rule that deletes and creates again that node. Such two rules are not equivalent neither in the DPO approach (where because of the dangling condition, if there is an edge incident to the node then the second rule might not be applicable to a valid match for the first one) nor in SPO (where edges incident to the node would be deleted by the second rule as a side-effect), just to mention two of the most popular graph transformation approaches.

Another solution is to adapt McMillan's procedure; i.e., one generalizes it in such a way that the unfolding of a contextual net is itself a contextual net. In [14] this was done for the subclass of *read-persistent* contextual nets, for which a slight modification of McMillan's algorithm works, essentially because the restriction guarantees that each event has a single causal history. This approach has been successfully generalized to graph grammars in [2], by identifying a corresponding class of read-persistent, finite-state grammars, and showing how the finite complete prefix computed with a variation of McMillan's algorithm could be used for verifying properties of the original grammar.

However, read-persistency can be a strong restriction. Recently, we have proposed an algorithm that works for the whole class of bounded contextual nets [3]. The main idea is to equip events with causal histories. Instead of building the prefix by adding one event at a time, we add one pair *(event, history)* at a time.

While [3] provided the theoretical foundations of the finite complete prefixes of contextual nets, important practical concerns were left unresolved. E.g., [3] did not address how to actually compute the pairs that are added to the unfolding. Here, we refine the algorithm and design an effective, concrete procedure for computing the unfolding prefix. This is based on the idea of associating histories not only to events, but also to places. This eases the inductive computation of the relation of concurrency, and thus the construction of the prefix of the unfolding.

Moreover, we argue that the proposed algorithm can be adapted smoothly to graph grammars as well, even if, because of space constraints, the technical details are not worked out.

2 Contextual nets and their unfolding

In this section we review contextual nets and their unfoldings. We refer to [3] for a fuller treatment with more examples illustrating the definitions.

2.1 Contextual nets

Let A be a set; a *multiset* of A is a function $M : A \to \mathbb{N}$ where $\{a \in A : M(a) > 0\}$ is finite. The set of multisets of A is denoted by $A^{\oplus}$. Usual operations such as multiset union $\oplus$ or difference $\ominus$ are used. A function $f : A \to B$ induces a function on multisets denoted $f^{\oplus} : A^{\oplus} \to B^{\oplus}$. We write $M \leq M'$ if $M(a) \leq M'(a)$ for all $a \in A$, and $a \in M$ for $M(a) > 0$. For $n \in \mathbb{N}$, we denote by $\bar{n}$ the constant function that yields n for all arguments, i.e. $\bar{n}(a) = n$ for all a.

Definition 1 (contextual net). *A* contextual Petri net (c-net) *is a tuple* $N = \langle S, T, {}^{\bullet}(.), (.)^{\bullet}, \underline{(.)}, m \rangle$*, where* S *is a set of* places*,* T *is a set of* transitions*, and* ${}^{\bullet}(.), (.)^{\bullet} : T \to S^{\oplus}$*,* $\underline{(.)} : T \to \mathcal{P}(S)$ *are functions which provide the pre-set, post-set, and context of a transition;* $m \in S^{\oplus}$ *is the* initial marking*. We assume* ${}^{\bullet}t \neq \bar{0}$ *for each transition* $t \in T$*.*

In the following when considering a c-net N, we will implicitly assume $N = \langle S, T, {}^{\bullet}(.), (.)^{\bullet}, \underline{(.)}, m \rangle$. Given a place $s \in S$ we define ${}^{\bullet}s = \{t \in T : s \in t^{\bullet}\}$, $s^{\bullet} = \{t \in T : s \in {}^{\bullet}t\}$, $\underline{s} = \{t \in T : s \in \underline{t}\}$.

An example of a contextual net, inspired by [14], is depicted in Fig. 2(a). For instance, referring to transition t_1 we have ${}^{\bullet}t_1 = \{s_1\}$, $t_1{}^{\bullet} = \{s_3\}$ and $\underline{t_1} = \{s_2\}$.

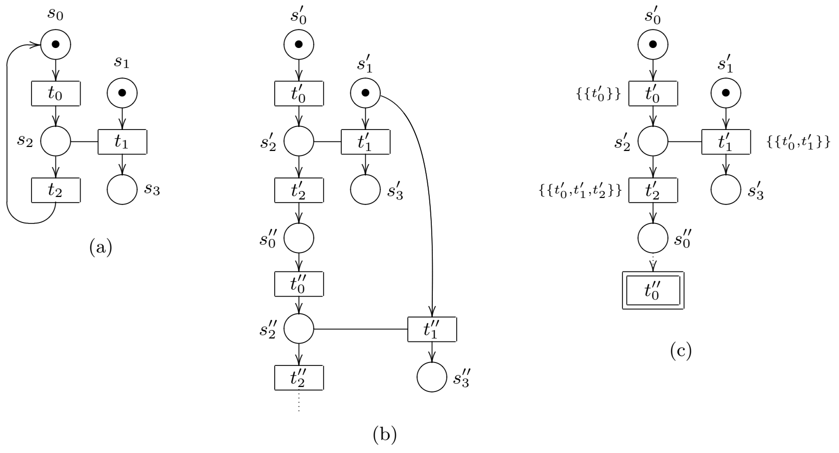


Fig. 2. (a) A c-net N_0, (b) its unfolding $\mathcal{U}_a(N_0)$ and (c) a complete enriched prefix.

Definition 2 (firing). *Let N be a c-net. A transition $t \in T$ is* enabled *at a marking $M \in S^{\oplus}$ if ${}^{\bullet}t \oplus \underline{t} \leq M$. In this case, its firing produces the marking $M' = M \ominus {}^{\bullet}t \oplus t^{\bullet}$, written as $M\,[t\rangle\,M'$.*

A marking M of a c-net N is called *reachable* if there is a finite sequence of firings leading from the initial marking to M, i.e., $m\,[t_1\rangle\,M_1\,[t_2\rangle\,M_2 \ldots [t_n\rangle\,M$.

Definition 3 (bounded, safe and semi-weighted nets). *A c-net N is called n-*bounded *if every reachable marking M satisfies $M \leq \bar{n}$. It is called* safe *if it is 1-bounded and, for any $t \in T$, ${}^{\bullet}t$, $t^{\bullet}$ are sets (rather than general multisets). A c-net N is called* semi-weighted *if the initial marking m is a set and, for any $t \in T$, $t^{\bullet}$ is a set.*

We restrict to semi-weighted nets in order to simplify the presentation. The treatment of general nets would lead to some technical complications in the definition of the unfolding (Definition 9), related to the fact that an occurrence of a place would not be identified uniquely by its causal history.

2.2 Occurrence c-nets

We will introduce two relations among transitions: causality and asymmetric conflict. Occurrence c-nets are safe c-nets where these relations satisfy certain acyclicity and well-foundedness requirements.

Causality is defined as for ordinary nets, with an additional clause stating that transition t causes t' if t generates a token in a context place of t'. Intuitively, $t < t'$ if t must happen before t' can happen.

Definition 4 (causality). *Let N be a safe c-net. The* causality relation $<$ *is the least transitive relation on $S \cup T$ such that (i) if $s \in {}^{\bullet}t$ then $s < t$; (ii) if*

$s \in t^\bullet$ then $t < s$; (iii) if $t^\bullet \cap \underline{t'} \neq \emptyset$ then $t < t'$. Given $x \in S \cup T$, we write $\lfloor x \rfloor$ for the set of causes *of x in T, defined as $\lfloor x \rfloor = \{t \in T : t \leq x\} \subseteq T$, where $\leq_N$ is the reflexive closure of $<$.*

We say that a transition t is in *asymmetric conflict* with t', denoted $t \nearrow t'$, if whenever both t and t' fire in a computation, t fires before t'. The paradigmatic case is when transition t' consumes a token in the context of t, i.e., when $\underline{t} \cap {}^\bullet t' \neq \emptyset$, as for transitions t'_1 and t'_2 in Fig. 2(b) (see [13, 9, 14]).

Definition 5 (asymmetric conflict). *Let N be a safe c-net. The* asymmetric conflict relation *$\nearrow$ is the binary relation on T defined as*

$$t \nearrow t' \qquad \text{if} \qquad \underline{t} \cap {}^\bullet t' \neq \emptyset \;\; \text{or} \;\; (t \neq t' \;\wedge\; {}^\bullet t \cap {}^\bullet t' \neq \emptyset) \;\; \text{or} \;\; t < t'.$$

For $X \subseteq T$, $\nearrow_X$ denotes the restriction of $\nearrow$ to X, i.e., $\nearrow_X = \nearrow \cap (X \times X)$.

An occurrence c-net is a safe c-net that exhibits an acyclic behaviour, satisfying suitable conflict-freeness requirements.

Definition 6 (occurrence c-nets). *A c-net N is called* occurrence c-net *if*

- *N is safe, and for any $s \in S$, $|{}^\bullet s| \leq 1$ (no backward conflicts)*
- *$<$ is a strict partial order and $\lfloor t \rfloor$ is finite for any $t \in T$;*
- *$m = \{s \in S : {}^\bullet s = \emptyset\}$ (the initial marking is the set of minimal places);*
- *$\nearrow_{\lfloor t \rfloor}$ is acyclic for all $t \in T$.*

The last condition of the definition corresponds to the requirement of irreflexivity for the conflict relation in ordinary occurrence nets. An example of an occurrence c-net can be found in Fig. 2(b). From now on, consistently with the literature, we shall often call the transitions of an occurrence c-net *events*.

Definition 7 (configurations). *Let N be an occurrence c-net. A finite set of events $C \subseteq T$ is called a* configuration *if*

1. *$\nearrow_C$ is acyclic;*
2. *C is left-closed w.r.t. $\leq$, i.e. for all $t \in C$, $t' \in T$, $t' \leq t$ implies $t' \in C$.*

The marking produced by a configuration C is $C^\bullet = m \cup \bigcup_{t \in C} t^\bullet - \bigcup_{t \in C} {}^\bullet t$. A finite set M of places is called concurrent*, written $conc(M)$, if there exists a configuration C such that $M \subseteq C^\bullet$.*

We denote by $Conf(N)$ the set of all configurations of N. They are equipped with the ordering defined as $C_1 \sqsubseteq C_2$, if $C_1 \subseteq C_2$ and $\neg(t_2 \nearrow t_1)$ for all $t_1 \in C_1$, $t_2 \in C_2 \setminus C_1$. Furthermore two configurations C_1, C_2 are said to be in conflict*, written $C_1 \# C_2$, when there is no $C \in Conf(N)$ such that $C_1 \sqsubseteq C$ and $C_2 \sqsubseteq C$.*

Configurations characterise the possible (concurrent) computations of an occurrence c-net. The relation $\sqsubseteq$ is a computational order of configurations: $C \sqsubseteq C'$ if C can evolve and become C'. Differently from non-contextual occurrence nets this order is not simply subset inclusion among configurations.

Given a configuration C and an event $t \in C$, the *history of* t *in* C is the set of events that *must* precede t in the (concurrent) computation represented by C. For ordinary nets the history of an event t coincides with the set of causes $\lfloor t \rfloor$, independently of the configuration where t occurs. For c-nets, the presence of asymmetric conflicts implies that an event may have several histories.

Definition 8 (history). *Let N be an occurrence net. Given a configuration C and an event $t \in C$, the* history *of t in C, denoted by $C[\![t]\!]$, is defined as $C[\![t]\!] = \{t' \in C : t'(\nearrow_C)^* t\}$. The set of possible histories of an event t, namely $\{C[\![t]\!] : C \in \mathit{Conf}(N) \ \wedge \ t \in C\}$ is denoted by $\mathit{Hist}(t)$.*

For example, in the contextual net shown in Fig. 2(b), the event t'_2 has two possible histories: $\{t'_0, t'_2\}$ and $\{t'_0, t'_1, t'_2\}$.

2.3 Unfolding of contextual nets

Given a semi-weighted c-net N, an *unfolding* construction allows us to obtain an occurrence c-net $\mathcal{U}_a(N)$ that describes the behaviour of N [4, 14]. The unfolding can be constructed inductively by starting from the initial marking of N and then by adding, at each step, an occurrence of each transition of N which is enabled by (the image of) a concurrent subset of the places already generated.

Definition 9 (unfolding). *Let $N = \langle S, T, {}^\bullet(.), (.)^\bullet, \underline{(.)}, m\rangle$ be a semi-weighted c-net. The unfolding $\mathcal{U}_a(N) = \langle S', T', {}^\bullet(.), (.)^\bullet, \underline{(.)}, m'\rangle$ of the net N is the occurrence c-net generated by the following inference rules, where $M_p, M_c \subseteq S'$; $M_p \cap M_c = \emptyset$; and $\pi_2(\langle x, y\rangle) = y$.*

$$\frac{s \in m}{\langle \emptyset, s\rangle \in S'} \qquad \frac{t' = \langle M_p, M_c, t\rangle \in T' \quad s \in t^\bullet}{\langle t', s\rangle \in S'}$$

$$\frac{t \in T \quad {\pi_2}^{\oplus}(M_p) = {}^\bullet t \quad {\pi_2}^{\oplus}(M_c) = \underline{t} \quad \mathit{conc}(M_p \cup M_c)}{\langle M_p, M_c, t\rangle \in T'}$$

The initial marking is $m' = \{\langle \emptyset, s\rangle : s \in m\}$, and given $t' = \langle M_p, M_c, t\rangle$

$${}^\bullet t' = M_p \qquad \underline{t'} = M_c \qquad t'^\bullet = \{\langle t', s\rangle : s \in t^\bullet\}$$

The folding morphism $f_N = \langle f_T, f_S\rangle : \mathcal{U}_a(N) \to N$ is a pair of mappings $f_T : T' \to T$ and $f_S : S' \to S$ defined by $f_T(t') = t$ for $t' = \langle M_p, M_c, t\rangle$ and $f_S(s') = s$ for $s' = \langle x, s\rangle$.

Places and events in the unfolding of a c-net represent respectively tokens and firing of transitions in the original net. Each place in the unfolding is a pair recording the "history" of the token and the corresponding place in the original net. Each event is a triple recording the precondition and context used in the firing, and the corresponding transition in the original net. A new place with empty history $\langle \emptyset, s\rangle$ is generated for each place s in the initial marking. Moreover, a new event $t' = \langle M_p, M_c, t\rangle$ is inserted in the unfolding whenever we can find a concurrent set of places (precondition M_p and context M_c) that corresponds, in the original net, to a marking that enables t. For each place s in the post-set of such t, a new place $\langle t', s\rangle$ is generated, belonging to the post-set of t'. The folding morphism f maps each place (event) of the unfolding to the corresponding place (transition) in the original net.

An initial part of the unfolding of the net N_0 in Fig. 2(a) is represented in Fig. 2(b). The folding morphism from $\mathcal{U}_a(N_0)$ to N_0 is implicitly represented by the name of the items in the unfolding.

Proposition 1 (completeness of the unfolding). *Let N be a c-net and let $\mathcal{U}_a(N) = \langle S', T', {}^{\bullet}(.), (.)^{\bullet}, \underline{(.)}, m'\rangle$ be its unfolding. A marking $M \in S^{\oplus}$ is coverable in N iff there exists a concurrent subset $X \subseteq S'$ such that $M = {f_S}^{\oplus}(X)$.*

Proposition 1 captures the sense in which the unfolding is *complete* w.r.t. the original net. This notion of completeness is slightly weaker than that of [10, 14], for example, as it is concerned with markings only, and not with transitions.

The result also suggests a method for checking reachability of a marking M of the original net: find the subsets X such that $M = {f_S}^{\oplus}(X)$ and check whether at least one is concurrent. Note that in (non-contextual) Petri net unfoldings this amounts to checking whether $\lfloor X\rfloor$ does not contain a (symmetric) conflicts, which takes linear time in $|\lfloor X\rfloor|$. For contextual nets, one needs to check the absence of cycles in the asymmetry relation, which is equally possible in linear time. Hence the complexity of checking the concurrency of a set X is the same in ordinary and contextual unfoldings, and since a contextual unfolding has in general fewer conditions and events than its PR-encoding, fewer concurrency checks need to be answered.

3 Computing the prefix as an enriched occurrence net

In this section we first recall the notion of enriched occurrence net, which is an occurrence net which records a subset of histories for each involved event. Next we describe an algorithm for computing a finite complete prefix of the full unfolding of a c-net N, which is a mild variation of that in [3]. The construction starts from the initial marking, and iteratively adds new extended events representing occurrences of transitions of N. The prefix will actually be an *enriched* occurrence net, where only histories which are considered "useful to produce new markings" are recorded.

Definition 10 (enriched occurrence net, extended event). *An* enriched occurrence net *is a pair $E = \langle N_E, \chi_E\rangle$, where $N_E = \langle S_E, T_E, {}^{\bullet}(.), (.)^{\bullet}, \underline{(.)}, m_E\rangle$ is an occurrence net and $\chi_E : T_E \to \mathcal{P}(\mathcal{P}(T_E))$ is a function such that*

- *for any $t \in T_E$, $\emptyset \neq \chi_E(t) \subseteq Hist(t)$*
- *for all $t, t' \in T_E$, for any $C \in \chi_E(t)$ if $t' \in C$ then $C[\![t']\!] \in \chi_E(t')$.*

An extended event *for an occurrence net N_E is a pair $\epsilon = \langle t, H_t\rangle$, where $t \in T_E$ and $H_t \in Hist(t)$. We say that $\langle t, H_t\rangle$* covers *another extended event*

$\langle t', H_{t'} \rangle$ when $H_{t'} \sqsubseteq H_t$. An enriched occurrence net $E = \langle N_E, \chi_E \rangle$ contains the extended event $\epsilon = \langle t, H_t \rangle$, written $\epsilon \in E$, if $t \in T_E$ and $H_t \in \chi_E(t)$.

From now on, $N = \langle S, T, {}^\bullet(.), (.)^\bullet, \underline{(.)}, m \rangle$ is a fixed semi-weighted c-net, $\mathcal{U}_a(N) = \langle S', T', {}^\bullet(.), (.)^\bullet, \underline{(.)}, m' \rangle$ is its unfolding, and $f_N : \mathcal{U}_a(N) \to N$ is the folding morphism.

Definition 11 (enriched prefix). *An* enriched prefix *of the unfolding $\mathcal{U}_a(N)$ is any enriched occurrence net E such that N_E is a prefix of $\mathcal{U}_a(N)$.*

An example of an enriched prefix of $\mathcal{U}_a(N_0)$ in Fig. 2(b) is given in Fig. 2(c). For any event t the set of histories $\chi_E(t)$ is written next to the event.

During the construction of a complete prefix of the unfolding, at each step we add a *possible extension* to the current prefix, that is, an extended event whose pre-set and context are already there, and whose history is compatible with the histories contained in the current prefix for the involved events.

Definition 12 (possible extension). *Let E be an enriched prefix of $\mathcal{U}_a(N)$. A* possible extension *of E is an extended event $\epsilon = \langle t, H_t \rangle$ of $\mathcal{U}_a(N)$ such that $\epsilon \notin E$ and for any $t' \in H_t - \{t\}$ it holds that $\langle t', H_t[\![t']\!] \rangle \in E$.*

Note that $\epsilon = \langle t, H_t \rangle$ is a possible extension when $E' \overset{def}{=} E \cup \epsilon$, obtained from E by inserting transition t and its post-set (if it is not already there) and by adding H_t to the set of histories of t, is an enriched prefix of $\mathcal{U}_a(N)$. E.g., for the prefix in Fig. 2(c) (assuming that t''_0 is not there) two possible extensions are $\epsilon = \langle t''_0, \{t'_0, t'_1, t'_2, t''_0\} \rangle$ and $\epsilon' = \langle t'_2, \{t'_0, t'_2\} \rangle$. Instead, $\langle t''_0, \{t'_0, t'_2, t''_0\} \rangle$ is not a possible extension, since t'_2 does not have the history $\{t'_0, t'_2\}$.

A configuration of $\mathcal{U}_a(N)$ represents a computation in the unfolding, which in turn maps, via the folding morphism, to a computation of N. Hence we can define the marking of N after a configuration of the unfolding.

Definition 13 (marking after a configuration). *Let $C \in Conf(\mathcal{U}_a(N))$ be a configuration. The marking of N after C is defined as $mark(C) = {f_S}^{\oplus}(C^\bullet)$.*

In [10] a *cut-off* is defined as an event of the unfolding that can be omitted safely because there exists another event with a smaller causal history generating the same marking. In our setting a cut-off is defined as an *extended* event, thus taking histories explicitly into account.

Definition 14 (cut-off). *Let E be an enriched prefix of the unfolding. An extended event $\langle t, H_t \rangle$ in E is called a* cut-off *if either $mark(H_t) = m$, the initial marking of N, or there is another extended event $\langle t', H_{t'} \rangle$ of E satisfying (i) $mark(H_t) = mark(H_{t'})$ and (ii) $|H_{t'}| < |H_t|$.*

The algorithm shown in Fig. 3 computes a complete finite prefix. It is analogous to the standard algorithm for ordinary Petri nets, but applied to extended events: during the construction, each event t of *Fin*, the currently built part of

```
$Fin := m'$, $\chi_{Fin} := \emptyset$
$pe := \{\langle t', \emptyset\rangle : t'$ enabled by $m'\}$
while $pe \neq \emptyset$ do
  – take $\epsilon = \langle t, H\rangle \in pe$ such that $|H|$ is minimal.
  – $pe = pe - \{\epsilon\}$
  – if $\epsilon$ would be a cut-off in $Fin$, do nothing, else insert $\epsilon$ in $Fin$, i.e.,
      • if $t$ is already present in $Fin$ then add the history $H$ to $\chi_{Fin}(t)$;
      • otherwise add $t$ and the places in $t^\bullet$ to $Fin$, and set $\chi_{Fin}(t) := \{H\}$.
      • $pe := pe \cup PE(Fin, \epsilon)$
```

Fig. 3. Algorithm for computing the finite prefix

the prefix, has associated also a set of histories $\chi_{Fin}(t)$, thus making the prefix under construction an enriched occurrence net.

The algorithm is similar to, but more abstract than, the algorithm in [3]. In fact, the present algorithm uses the function PE which, applied to the enriched prefix Fin and to an extended event ϵ, is expected to return a set of extended events. In [3], instead, the set of extended events added at each iteration was described (in a pretty abstract, non-constructive way) in the algorithm itself.

The main result shows that the algorithm terminates and correctly computes a complete finite prefix of the unfolding, provided that the set returned by $PE(Fin, \epsilon)$ contains at least all the possible extensions of Fin which cover ϵ.

Theorem 1 (correctness and completeness). *If the net N is finite and n-bounded, and if function PE applied to any enriched prefix Fin and to an extended event ϵ returns a set of extended events containing at least all the possible extensions of Fin covering ϵ, then the algorithm in Fig. 3 terminates, and the resulting enriched prefix Fin_0 is complete.*

As an example, consider the net N_0 and its unfolding $\mathcal{U}_a(N_0)$ in Fig. 2. The algorithm (as detailed in the next section) would produce the enriched prefix depicted in Fig. 2(c). Note that it includes the event t'_2. In fact t'_2 has two possible histories: the minimal history $H_2 = \lfloor t'_2 \rfloor = \{t'_0, t'_2\}$ and $H'_2 = \{t'_0, t'_1, t'_2\}$. While $\langle t'_2, H_2\rangle$ is a cut-off, the pair $\langle t'_2, H'_2\rangle$ is not, and thus it is included.

4 Computing the possible extensions

During the construction of the complete prefix, we must compute the possible extensions of the current prefix. The key idea to do this efficiently and incrementally is to equip places with histories.

4.1 Extended places

A first simple observation is that the histories of an event t in an enriched occurrence net E are obtained from the histories of the events that are in direct

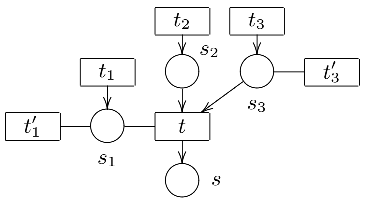


Fig. 4. Predecessors w.r.t. asymmetric conflict of an event t.

asymmetric conflict with t. Referring to Fig. 4, the histories for t can be constructed by taking one history for every direct cause of t (i.e., t_1, t_2 and t_3), and possibly one for t'_3 that is in direct asymmetric conflict with t. In contrast, a transition that merely reads from the context of t, such as t'_1, is not in direct asymmetric conflict with t and therefore not considered. If these histories are "consistent" they can be joined to form a history of t. Once this history is added to the prefix, it can be used to generate new histories for the events (not depicted) that use the post-set s or that consume the context s_1. Therefore the generation of a new history for an event has to be "propagated" according to the structure of the net, because it can entail new histories for other events.

In order to concretely realize this sort of propagation in the algorithm we will rely on the notion of *extended place*, which is a place with an associated history. Hereafter, $E = \langle N, \chi \rangle$ denotes a fixed enriched occurrence net.

Definition 15 (histories for places). *The* causal histories *of a place $s \in S$, denoted $\chi_c(s)$, are the histories of the transition that generates s (i.e., $\chi_c(s) = \chi(^{\bullet}s)$) if $s \notin m$. If $s \in m$, then $\chi_c(s) = \{\emptyset\}$. The* read histories *of s, denoted $\chi_r(s)$, are the histories of the events which read s, i.e., $\chi_r(s) = \bigcup_{t \in \underline{s}} \chi(t)$.*

In words, a place inherits histories from the (unique) event which generates the place (causal histories) and from events which read the place (read histories).

Definition 16 (extended places). *An* extended place *is a pair $\sigma = \langle s, H \rangle$ where $s \in S$ and $H \in \chi_c(s) \cup \chi_r(s)$. It is called a* causal extended place *if $H \in \chi_c(s)$ and a* read extended place *if $H \in \chi_r(s)$.*

Let $\langle s, H \rangle$ be a read extended place. It can be shown that there is a unique causal history $H' \in \chi_c(s)$ such that $H' \sqsubseteq H$. The extended place $\langle s, H' \rangle$ is denoted $\langle s, H \rangle^{\uparrow}$. We write π_H and π_S for the projections of an extended place to its components, i.e., given $\sigma = \langle s, H \rangle$, $\pi_H(\sigma) = H$ and $\pi_S(\sigma) = s$.

More intuition can be obtained from Fig. 5, where events and places are annotated with their histories. Place s_3 is associated with three extended places: one causal extended place $\langle s_3, \emptyset \rangle$ and two read extended places $\langle s_3, \{t_1\} \rangle$, $\langle s_3, \{t_2\} \rangle$; we do not consider $\langle s_3, \{t_1, t_2\} \rangle$ which would represent two readings of place s_3. Now, according to Definition 8, there are four histories for transition t_3, i.e.,

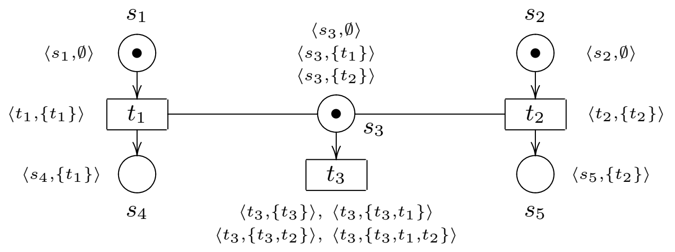

s_1
$\langle s_1,\emptyset\rangle$
$\langle t_1,\{t_1\}\rangle$
t_1
$\langle s_4,\{t_1\}\rangle$
s_4
$\langle s_3,\emptyset\rangle$
$\langle s_3,\{t_1\}\rangle$
$\langle s_3,\{t_2\}\rangle$
s_3
t_3
$\langle t_3,\{t_3\}\rangle$, $\langle t_3,\{t_3,t_1\}\rangle$
$\langle t_3,\{t_3,t_2\}\rangle$, $\langle t_3,\{t_3,t_1,t_2\}\rangle$
s_2
$\langle s_2,\emptyset\rangle$
t_2
$\langle t_2,\{t_2\}\rangle$
$\langle s_5,\{t_2\}\rangle$
s_5

Fig. 5. Illustration of extended transitions and places

$\{t_3\}$, $\{t_3, t_1\}$, $\{t_3, t_2\}$ and $\{t_3, t_1, t_2\}$. Interestingly, all of them can be obtained as a suitable combination of the extended places associated with s_3 (without considering the extended events corresponding to t_1 and t_2). Note that if we had n readers instead of just 2, we would have 2^n histories but only n read places. Thus, ignoring multiple readings of a place avoids a combinatorial explosion.

For each place we have, conceptually, one extended place for each possible history of the (only) event which generates the place, and one for each event that reads the place. In this way a new history for a transition t can be obtained by looking only at the extended places in its pre-set and context, without checking the transitions in asymmetric conflict with t, thus making the construction more local. The generation of a history for t in turn produces new histories (extended places) for the places in the post-set and in the context. This can be formalized elegantly by defining a notion of pre-set and post-set for extended events.

Definition 17 (pre-set and post-set of extended events). *Given an extended event $\epsilon = \langle t, H\rangle$, we define*

– $^\bullet\epsilon = \{\langle s, H'\rangle \mid s \in {}^\bullet t \ \wedge \ H' \in \chi_c(s) \cup \chi_r(s) \ \wedge \ H' \sqsubseteq H\}$
– $\hat{\underline{\epsilon}} = \{\langle s, H'\rangle \mid s \in \underline{t} \ \wedge \ H' \in \chi_c(s) \ \wedge \ H' \sqsubseteq H\}$
– $\epsilon^\bullet = \{\langle s, H\rangle \mid s \in t^\bullet\}$
– $\check{\underline{\epsilon}} = \{\langle s, H\rangle \mid s \in \underline{t}\}$

Note in particular, that an extended event has *two* "context sets": $\hat{\underline{\epsilon}}$, the extended places which are read (these must be *causal* extended places, because reading a place should be causally unrelated to other concurrent events reading the same place) and $\check{\underline{\epsilon}}$, the extended places which are generated by the readings of ϵ (these are read extended places). For instance, referring to Fig. 5, we can consider the extended event $\epsilon = \langle t_1, \{t_1\}\rangle$. Then we have $^\bullet\epsilon = \langle s_1, \emptyset\rangle$, $\hat{\underline{\epsilon}} = \langle s_3, \emptyset\rangle$, $\epsilon^\bullet = \langle s_4, \{t_1\}\rangle$, and $\check{\underline{\epsilon}} = \langle s_3, \{t_1\}\rangle$.

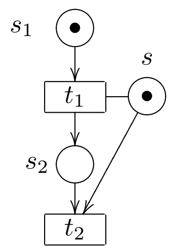
In order to formalize the intuition that the extended places used when generating a new extended event are "consistent", we introduce two relations on extended places: concurrency and subsumption.

Definition 18 (concurrency). *Let $\sigma_1 = \langle s_1, H_1\rangle$, $\sigma_2 = \langle s_2, H_2\rangle$ be extended places, $\sigma_1 \neq \sigma_2$. We say that they are* concurrent, *written $\sigma_1 \frown \sigma_2$, if $\neg(H_1 \# H_2)$ (thus $H_1 \cup H_2$ is a configuration) and $s_1, s_2 \in (H_1 \cup H_2)^\bullet$.*

In words, σ_1 and σ_2 are concurrent when the histories H_1 and H_2 associated with the two places are compatible, hence their union is a configuration C, and after executing C both places are marked.

We will see in Lemma 1 that, as expected, each pair of extended places in the pre-set of an extended event is concurrent. But the pre-set of an extended event also satisfies an important closure property:

Consider the net to the right. After the execution of t_1, for s we have one causal history $\langle s, \emptyset \rangle$ and one read history, namely $\langle s, \{t_1\} \rangle$. Now, transition t_2 could, in principle, be fired using the causal histories $\langle s_2, \{t_1\} \rangle$ and $\langle s, \emptyset \rangle$ and no read history. However, observe that the extended place $\langle s, \{t_1\} \rangle$ is implicitly there, since the inclusion of $\langle s_2, \{t_1\} \rangle$ implies that s has been read by t_1: we will say that $\langle s_2, \{t_1\} \rangle$ *subsumes* $\langle s, \{t_1\} \rangle$. Notice that the fact that $\langle s, \{t_1\} \rangle$ is not mentioned explicitly does not affect the new history that we are building for t_2, but it causes serious problems when computing the concurrency relation.



Definition 19 (subsumption relation). *For $\sigma = \langle s, H \rangle$, $\sigma' = \langle s', H' \rangle$ extended places, we write $\sigma \propto \sigma'$ (σ* subsumes *σ') when $s' \in H^{\bullet}$ and there exists $t \in H$ such that $s' \in \underline{t}$ and $H' = H[\![t]\!]$.*

In the example above $\langle s_2, \{t_1\} \rangle \propto \langle s, \{t_1\} \rangle$. Note that if $\sigma \propto \sigma'$, then σ' is necessarily a read extended place.

Lemma 1 provides a characterisation of extended events which will be used in the algorithm when looking for possible extensions of the current prefix. It says that an extended event $\epsilon = \langle t, H \rangle$ can be generated from a set of extended places which compose the pre-set and the context of t, and which have "consistent" histories, whose union will be H. Then ϵ generates a causal extended place for each event in the post-set of t and one read extended place for each event in the context of t. In particular, we note that the pre-set of an extended event is automatically closed under the subsumption relation. For this reason, in the algorithm, when looking for possible extensions we will take a set of extended places X which "enables" a transition and which satisfies a closure condition w.r.t. subsumption analogous to that in item 4 below.

Lemma 1 (events). *Let $\epsilon = \langle t, H \rangle$ be an extended event. If we let $X_p = {}^{\bullet}\epsilon$ and $X_c = \underline{\hat{\epsilon}}$, and $X = X_p \cup X_c$, then*

1. *$\pi_S(X_p) = {}^{\bullet}t$ and $\pi_S(X_c) = \underline{t}$;*
2. *for any place $s \in \pi_S(X)$ there is a unique causal history $H' \in \chi_c(s)$ such that $\langle s, H' \rangle \in X$;*
3. *for any place $s \in \pi_S(X)$, if there is a read history $H' \in \chi_r(s)$ such that $\langle s, H' \rangle \in X$ then $s \in {}^{\bullet}t$;*
4. *for any $\sigma \in X$, if $\sigma \propto \sigma'$ and ${\sigma'}^{\uparrow} \in X_p$ then $\sigma' \in X_p$ (subsumption-closedness)*
5. *X is pairwise concurrent*

and $H = \bigcup \pi_H(X) \cup \{t\}$.

The next two lemmata provide an inductive characterisation of the concurrency and subsumption relations that will be pivotal for identifying the possible extensions of the prefix.

Lemma 2 (concurrency). *Let $\sigma = \langle s, H\rangle$ and $\sigma' = \langle s', H'\rangle$ be extended places in E, with $|H| \geq |H'|$. Then $\sigma \frown \sigma'$ if and only if $H = H' = \emptyset$ or $\sigma \in \epsilon^{\bullet} \cup \check{\underline{\epsilon}}$ for some ϵ and one of the following holds*

1. *$\sigma' \in \epsilon^{\bullet} \cup \check{\underline{\epsilon}}$, $\sigma \neq \sigma'$*
2. *$\sigma' \in \hat{\underline{\epsilon}}$*
3. *(a) for all $\sigma'' \in {}^{\bullet}\epsilon \cup \hat{\underline{\epsilon}}$, it holds $\sigma'' \frown \sigma'$ and (b) if $\sigma' \propto \sigma'''$, with $\sigma'''^{\uparrow} \in {}^{\bullet}\epsilon$ then $\sigma''' \in {}^{\bullet}\epsilon$.*

Lemma 3 (subsumption). *Let σ and σ' be extended places in E. Then $\sigma \propto \sigma'$ if and only if $\sigma \in \epsilon^{\bullet} \cup \check{\underline{\epsilon}}$ for some ϵ and one of the following holds*

1. *$\sigma' \in \check{\underline{\epsilon}}$ or*
2. *there is $\sigma'' \in {}^{\bullet}\epsilon \cup \hat{\underline{\epsilon}}$ such that $\sigma'' \propto \sigma'$ and $\sigma \frown \sigma'$.*

4.2 Computing the possible extensions

With the notions introduced so far, finally we can show how to generate the possible extensions of a given enriched prefix. We will refer to the construction of the prefix of the unfolding of a fixed semi-weighted c-net $N = \langle S, T, {}^{\bullet}(.), (.)^{\bullet}, \underline{(.)}, m\rangle$.

According to Lemma 1, in order to generate a new extended event, we can choose a concurrent set of extended places which "enables" the event. For any place in the pre-set and context we take a single causal history (condition 2 below). Only for places in the pre-set we can additionally take some read histories (condition 3 below). The fact that read histories can be consumed but not read corresponds to the fact that, referring to Fig. 4, to build a history for t we take additional histories only for transitions which read from consumed places (such as t'_3) and not for those that read from context places (such as t'_1).

Proposition 2 (possible extensions). *A possible extension of an enriched prefix Fin is an extended event $\epsilon = \langle t, H_t\rangle$ obtained as follows. Find sets of extended places X_p, X_c such that, if we denote by $X = X_p \cup X_c$ then*

1. *$f(\pi_S(X_p)) = {}^{\bullet}t_N$ and $f(\pi_S(X_c)) = \underline{t_N}$ for some transition t_N of the original net; the corresponding event in the unfolding is thus $t = \langle \pi_S(X_p), \pi_S(X_c), t_N\rangle$;*
2. *for any place $s \in \pi_S(X)$ there is exactly a single* causal *history $H \in \chi_c(s)$ such that $\langle s, H\rangle \in X$;*
3. *for any $s \in \pi_S(X)$ if there is a* read *history $\langle s, H\rangle \in X$ for some $H \in \chi_r(s)$ then $s \in {}^{\bullet}t$;*
4. *for any extended place $\sigma \in X$, if there is a σ' such that $\sigma \propto \sigma'$ and $\sigma'^{\uparrow} \in X_p$ then $\sigma' \in X$.*
5. *X is pairwise concurrent*

The history for t is defined as $H_t = \{t\} \cup \bigcup \pi_H(X)$.

Points 4 and 5 of Proposition 2 require to check the subsumption and the concurrency relations between certain pairs of extended places. The next result, that can be proved directly from the characterization of the two relations given in Lemmata 2 and 3, shows that both the concurrency and the subsumption relations can be computed inductively during the construction of the prefix.

Proposition 3 (concurrency and subsumption, inductively). *Let Fin be a finite enriched prefix of the unfolding obtained, according to the algorithm of Fig. 3, by starting with the initial marking m' and adding extended events $\epsilon_0, \ldots, \epsilon_n$. Then for each pair σ and σ' of extended places of Fin, $\sigma \frown \sigma'$ ($\sigma \propto \sigma'$, respectively) if and only if this can be deduced using the following rules:*

1. *[Base case]: For all $s_1, s_2 \in m'$ (initial marking) $\langle s_1, \emptyset\rangle \frown \langle s_2, \emptyset\rangle$.*
2. *[Inductive case]: Assume that the extended event $\epsilon_i = \langle t, H\rangle$ has been added, using the set $X = X_p \cup X_c$ of extended places, thus $t = \langle \pi_S(X_p), \pi_S(X_c), t_N\rangle$. Adding ϵ_i produces a set of extended places $Y = Y_p \cup Y_c$, where $Y_p = t^\bullet \times \{H\}$ are causal extended places and $Y_c = \underline{t} \times \{H\}$ are read extended places.*
   Then the concurrency relation can be extended to pairs including at least one new extended place using the following inference rules:

$$\frac{\sigma, \sigma' \in Y \quad \sigma \neq \sigma'}{\sigma \frown \sigma'} \qquad \frac{\sigma \in Y \quad \sigma' \in X_c}{\sigma \frown \sigma'}$$

$$\frac{\sigma \in Y \quad \forall \sigma'' \in X.\, \sigma'' \frown \sigma' \quad \forall \sigma'' \in X_p.\, (\sigma'' = \sigma'''^{\uparrow} \wedge \sigma' \propto \sigma''' \to \sigma''' \in X_p)}{\sigma \frown \sigma'}$$

   Similarly, the subsumption relation can be extended to pairs including at least one new extended place, using rule (new) *for the subsumptions induced by ϵ_i, and rule* (inh) *for inheriting the subsumptions of the premises, if not consumed:*

$$\frac{\sigma \in Y \quad \sigma' \in Y_c}{\sigma \propto \sigma'}\,(new) \qquad\qquad \frac{\sigma \in Y \quad \sigma'' \in X \quad \sigma'' \propto \sigma' \quad \sigma \frown \sigma''}{\sigma \propto \sigma'}\,(inh)$$

We conclude by stating the correctness of the definition of the function PE based on Propositions 2 and 3.

Corollary 1 (definition and correctness of PE). *Let PE be the function that, when applied to an enriched prefix Fin and to an extended event ϵ, returns all the possible extensions of Fin, as defined in Proposition 2, built using at least one extended place generated by ϵ (i.e., using a set of extended places X such that $X \cap (\epsilon^\bullet \cup \underline{\check{\epsilon}}) \neq \emptyset$). Then PE satisfies the requirements of Theorem 1.*

5 Finite prefix for graph transformation systems

As mentioned in the introduction, the adequate treatment of read arcs and of the "reading" of items in general, is particularly important for graph transformation systems (GTSs). In fact, for GTSs—due to the presence of dangling conditions—the preservation of a node cannot be simulated by its deletion and subsequent re-creation. This means that the use of a "direct" algorithm which is not based on an encoding of read arcs as consume-create loops, while being an option for contextual nets, becomes mandatory for GTSs.

It has been observed earlier that in the absence of inhibiting conditions, which occur in DPO, the unfolding of graph transformation systems can be defined analogously to the unfolding of contextual nets. In [2, 1] we have given the relevant definitions of occurrence graph grammars and we have generalized relations such as causality and asymmetric conflict to that setting. The role of places is played by the type graph, the initial graph represents the initial marking of the net and the rules correspond to transitions. In [2] the absence of inhibiting conditions was ensured by forbidding the deletion of nodes, whereas in [1] we used a more general setting, namely the SPO approach. Nevertheless, the more constrained format of rules allowed in [2] is balanced by the fact that one can take isomorphisms of graphs "up to isolated nodes", which leads to a more liberal notion of a "finite-state" graph transformation system.

Space limitations do not allow a detailed exposition of this development, but all constructions of the paper and especially the technical issues of Section 4—including the characterisation of extended events and the computation of the causality and subsumption relations—can be performed also for single-pushout rewriting. This would no longer be true if we switched to double-pushout rewriting where the presence of inhibiting conditions ("a node can not be deleted unless all adjacent edges are deleted") leads to a formalism which generalizes inhibitor nets, for which the unfolding semantics becomes much more involved.

In order to obtain *finite* complete McMillan prefixes for GTSs we clearly need to restrict to finite-state GTSs, where the number of reachable graphs is finite (up to isomorphism). More technically, this amounts to requiring that for every reachable graph, typed over the type graph T, the size of the preimage of an element of T is bounded by a constant k (this is analogous to the boundedness condition for Petri where one requires that for any reachable marking the number of tokens in each place is bounded by a fixed constant).

We also believe that this work can be generalized to rewriting in adhesive categories if we use a sesqui-pushout-like setting, as described in [6]. This would require to single out for a given rewriting system, a set of "atomic" subobjects from which any rewritten object can be built, playing the role of places.

6 Conclusions

We have described a worked-out procedure for computing McMillan prefixes for (bounded) contextual Petri nets and we argued that it can be extended to the computation of the prefix of the unfolding of finite-state graph transformation systems. Such prefixes are very valuable for the partial order verification of highly concurrent systems, where the use of unfolding techniques sometimes leads to an exponential gain in efficiency. While a part of the theoretical basis was already described in [3] it turned out that for aiming at an efficient implementation we needed more concepts such as extended places and the subsumption relation. A first prototype implementation for contextual nets has been realised. Although not yet optimised, it provides reassuring results on the feasibility of the approach.

As shown in this paper, the computation of finite complete prefixes for contextual nets is quite involved, but it is unclear whether it is possible to avoid the high complexity. The proposed technique becomes particularly interesting for graph transformation systems and related formalisms (with a structured state and preservation of items in a computational step), where approaches based on the encoding of item preservation by deletion and re-creation are not viable.

References

1. Baldan, P., Chatain, T., Haar, S., König, B.: Unfolding-based diagnosis of systems with an evolving topology. In: Proc. of CONCUR '08. LNCS, vol. 5201, pp. 203–217. Springer (2008)
2. Baldan, P., Corradini, A., König, B.: Verifying finite-state graph grammars: an unfolding-based approach. In: Gardner, P., Yoshida, N. (eds.) Proc. of CONCUR'04. LNCS, vol. 3170, pp. 83–98. Springer (2004)
3. Baldan, P., Corradini, A., König, B., Schwoon, S.: McMillan's complete prefix for contextual nets. LNCS Transactions on Petri Nets and Other Models of Concurrency (ToPNoC) 5100, 199–220 (2008)
4. Baldan, P., Corradini, A., Montanari, U.: An event structure semantics for P/T contextual nets: Asymmetric event structures. In: Nivat, M. (ed.) Proc. of FoSSaCS'98. LNCS, vol. 1378, pp. 63–80. Springer (1998)
5. Baldan, P., Corradini, A., Montanari, U., Ribeiro, L.: Unfolding Semantics of Graph Transformation. Information and Computation 205, 733–782 (2007)
6. Baldan, P., Corradini, A., Heindel, T., König, B., Sobociński, P.: Unfolding grammars in adhesive categories. In: Proc. of CALCO'09. LNCS, vol. 5728, pp. 350–366. Springer (2009)
7. Benveniste, A., Haar, S., Fabre, E., Jard, C.: Distributed monitoring of concurrent and asynchronous systems. In: Amadio, R.M., Lugiez, D. (eds.) Proc. of CONCUR'03. LNCS, vol. 2761, pp. 1–26. Springer (2003)
8. Bonet, B., Haslum, P., Hickmott, S.L., Thiébaux, S.: Directed unfolding of Petri nets. LNCS Transactions on Petri Nets and Other Models of Concurrency (ToPNoC) 5100, 172–198 (2008)
9. Langerak, R.: Transformation and Semantics for LOTOS. Ph.D. thesis, Department of Computer Science, University of Twente (1992)
10. McMillan, K.: Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits. In: Proc. of CAV'92. LNCS, vol. 663, pp. 164–174. Springer (1992)
11. McMillan, K.: Symbolic Model Checking. Kluwer (1993)
12. Nielsen, M., Plotkin, G., Winskel, G.: Petri Nets, Event Structures and Domains, Part 1. Theoretical Computer Science 13, 85–108 (1981)
13. Pinna, G.M., Poigné, A.: On the nature of events: another perspective in concurrency. Theoretical Computer Science 138(2), 425–454 (1995)
14. Vogler, W., Semenov, A., Yakovlev, A.: Unfolding and finite prefix for nets with read arcs. In: Proc. of CONCUR'98. LNCS, vol. 1466, pp. 501–516. Springer (1998)