McMillan's Complete Prefix for Contextual Nets*

Paolo Baldan[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] Institut für Informatik (I7), Technische Universität München, Germany

Abstract. In a seminal paper, McMillan proposed a technique for constructing a finite complete prefix of the unfolding of bounded (i.e., finite-state) Petri nets, which can be used for verification purposes. Contextual nets are a generalisation of Petri nets suited to model systems with read-only access to resources. When working with contextual nets, a finite complete prefix can be obtained by applying McMillan's construction to a suitable encoding of the contextual net into an ordinary net. However, it has been observed that if the unfolding is itself a contextual net, then the complete prefix can be significantly smaller than the one obtained with the above technique. A construction for generating such a contextual complete prefix has been proposed for a special class of nets, called read-persistent. In this paper we propose a new algorithm that works for arbitrary semi-weighted, bounded contextual nets. The construction explicitly takes into account the fact that, unlike ordinary or read-persistent nets, an event can have several different histories in contextual net computations.

1 Introduction

In recent years there has been a growing interest in the use of partial-order semantics to deal with the state-explosion problem when model checking concurrent systems. In particular, a thread of research that started with the seminal work by McMillan [10, 11] proposes the use of the *unfolding* semantics as a basis for the verification of finite-state systems, modelled as Petri nets.

The unfolding of a Petri net, originally introduced in [14], is a safe, acyclic *occurrence* net that completely expresses its behaviour. For non-trivial nets the unfolding can be infinite even if the original net is *bounded*, i.e., it has a finite number of reachable states. McMillan's algorithm constructs a *finite complete prefix*, i.e., a subnet of the unfolding such that each marking reachable in the original net corresponds to some concurrent set of places in such a prefix.

* Research partially supported by EU IST-2004-16004 SENSORIA, MIUR Project ART, DFG project SANDS and CRUI/DAAD VIGONI "Models based on Graph Transformation Systems: Analysis and Verification".

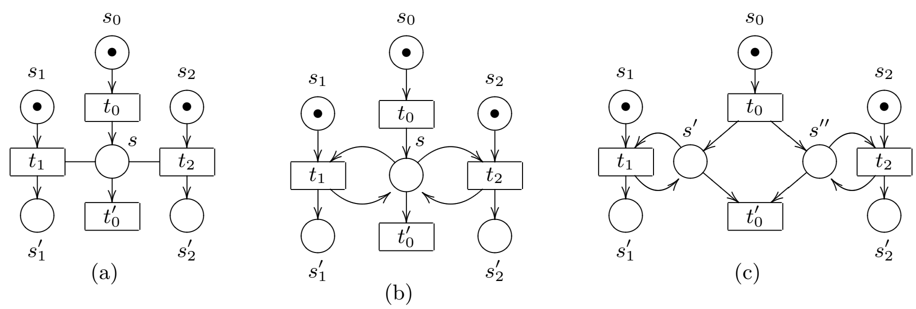


Fig. 1. (a) A safe contextual net; (b) its encoding by replacing read arcs with consume/produce loops; (c) its concurrency-preserving PR-encoding.

Contextual nets [13], also called nets with test arcs [5], activator arcs [8] or read arcs [17], extend ordinary nets with the possibility of checking for the presence of tokens without consuming them. The possibility of faithfully representing concurrent read accesses to resources allows to model in a natural way phenomena like concurrent access to shared data (e.g., reading in a database) [16], to provide concurrent semantics to concurrent constraint programs [12], to model priorities [7] or to conveniently analyse asynchronous circuits [18].

When working with contextual nets, if one is interested only in reachable markings, it is well-known that read arcs can be replaced by consume/produce loops (see Fig. 1(a) and (b)), obtaining an ordinary net with the same reachability graph. However, when one unfolds the net obtained by this transformation, the number of transitions and places might explode due to the sequentialization imposed on readers. A cleverer encoding, proposed in [18] and hereafter referred to as the *place replication encoding* (*PR-encoding*), consists of creating "private" copies of the read places for each reader (see Fig. 1(c)). In this way, for safe nets the encoding does not lead to a loss of concurrency, and thus the explosion of the number of events and places in the unfolding can be mitigated.

A construction that applies to contextual nets and produces an unfolding that is itself a contextual (occurrence) net has been proposed independently by Vogler, Semenov and Yakovlev in [18] and by the first two authors with Montanari in [3]. In particular, the (prefixes of the) unfolding obtained with this construction can be much smaller than in both encodings considered above.

Unfortunately, as discussed in [18], McMillan's construction of the finite complete prefix does not extend straightforwardly to the whole class of contextual nets. The authors of [18] propose a natural generalization of McMillan's algorithm by taking into account some specific features of contextual nets (for example, in the definition of *co-sets*), but they show that their approach only works for contextual nets that are *read-persistent*, i.e., where there is no interference between preconditions and context conditions: any two transitions t_1 and t_2 such that t_1 consumes a token that is read by t_2 cannot be enabled at the same time. Similarly, the algorithm proposed in [2], where McMillan's approach was generalised to graph grammars, is designed for a restricted class of grammars, which are the graph-grammar-theoretical counterpart of read-persistent nets.

The algorithms of [18] and [2] fail on non-read-persistent systems because, in general, a transition of a contextual occurrence net can have more than one possible *causal history* (or *local configuration*, according to [18]): this happens, for example, when a transition consumes a token which could be read by another transition. In this situation, McMillan's original *cut-off* condition (used by the algorithms in [18] and [2]) is not adequate anymore, because it considers a single causal history for each event (see also the example discussed in Section 3).

In this paper we present a generalization of McMillan's construction that applies to arbitrary bounded *semi-weighted* contextual nets, i.e., Place/Transition contextual nets where the initial marking and the post-set of each transition are sets rather than proper multisets: this class of nets strictly includes safe contextual nets. The proposed algorithm explicitly takes into account the possible histories of events, and generates from a finite bounded semi-weighted contextual net a finite complete prefix of its unfolding. The same constructions and results could have been developed for general weighted contextual nets, at the price of some technical (not conceptual) complications.

As in McMillan's original work, the key concept here is that of a cut-off event, which is, roughly, an event in the unfolding that, together with its causal history, does not contribute to generating new markings. We show that the natural generalisation of cut-off that takes into account all the possible histories of each event is theoretically fine, in the sense that the maximal cut-off free prefix of the unfolding is complete. However, this characterisation is not constructive in general, since an event can have infinitely many histories. We then show how this problem can be solved by restricting the attention to a finite subset of "useful" histories for each event, which really contribute to generating new states.

The contribution of this approach is twofold. From a theoretical point of view, the algorithm extends [18] since it applies uniformly to the full class of contextual nets (and, for read-persistent nets, it specialises to [18]). From a practical point of view, with respect to the approach based on the construction of the complete finite prefix of the PR-encoding, we foresee several improvements. For safe nets the proposed technique produces a smaller unfolding prefix (once the histories recorded for generating the prefix are disregarded) and it has a comparable efficiency (we conjecture that the histories considered when unfolding a safe contextual net exactly correspond to the events obtained by unfolding its PR-encoding). Additionally, our technique appears to be more efficient for non-safe nets (see Appendix A) and it looks sufficiently general to be extended to other formalisms able to model concurrent read accesses to part of the state, like graph transformation systems, for which the encoding approach does not seem viable.

The paper is structured as follows. In Section 2 we introduce contextual nets and their unfolding semantics. In Section 3 we characterise a finite complete prefix of the unfolding for finite-state contextual nets, relying on a generalised notion of cut-off and in Section 4 we describe an algorithm for constructing a complete finite prefix. Finally, in Section 5 we draw some conclusions.

2 Contextual nets and their unfolding

In this section we introduce the basics of marked contextual P/T nets [16, 13] and we review their unfolding semantics as defined in [18, 3].

2.1 Contextual nets

We first recall some notation for multisets. Let A be a set; a *multiset* of A is a function $M : A \to \mathbb{N}$. It is called finite if $\{a \in A : M(a) > 0\}$ is finite. The set of finite multisets of A is denoted by $\mu_* A$. The usual operations on multisets, like multiset union $\oplus$ or multiset difference $\ominus$, are used. We write $M \leq M'$ if $M(a) \leq M'(a)$ for all $a \in A$. If $M \in \mu_* A$, we denote by $[\![M]\!]$ the multiset defined, for all $a \in A$, as $[\![M]\!](a) = 1$ if $M(a) > 0$, and $[\![M]\!](a) = 0$ otherwise. A *multirelation* $f : A \leftrightarrow B$ is a multiset of $A \times B$. It is called *finitary* if $\{b \in B : f(a,b) > 0\}$ is a finite set for all $a \in A$. A finitary multirelation f induces in an obvious way a function $\mu f : \mu_* A \to \mu_* B$, defined as $\mu f(M)(b) = \sum_{a \in A} M(a) \cdot f(a,b)$ for $M \in \mu_* A$ and $b \in B$. In the sequel we will implicitly assume that all multirelations are finitary. A *relation* $r : A \leftrightarrow B$ is a multirelation r where multiplicities are bounded by one, namely $r(a,b) \leq 1$ for all $a \in A$ and $b \in B$. Sometimes we shall write simply $r(a,b)$ instead of $r(a,b) = 1$.

Definition 1 ((marked) contextual net). *A* (marked) contextual Petri net (c-net) *is a tuple* $N = \langle S, T, F, C, m \rangle$*, where*

- S *is a set of* places *and* T *is a set of* transitions*;*
- $F = \langle F_{pre}, F_{post} \rangle$ *is a pair of finitary multirelations* $F_{pre}, F_{post} : T \leftrightarrow S$*;*
- $C : T \leftrightarrow S$ *is a finitary relation, called the* context relation*;*
- $m \in \mu_* S$ *is a finite multiset, called the* initial marking.

The c-net is called finite *if* T *and* S *are finite sets. Without loss of generality, we assume* $S \cap T = \emptyset$*. Moreover, we require that for each transition* $t \in T$*, there exists a place* $s \in S$ *such that* $F_{pre}(t,s) > 0$*.*

In the following when considering a c-net N, we will implicitly assume $N = \langle S, T, F, C, m \rangle$.

Given a finite multiset of transitions $A \in \mu_* T$ we write ${}^\bullet A$ for its *pre-set* $\mu F_{pre}(A)$ and $A^\bullet$ for its *post-set* $\mu F_{post}(A)$. Moreover, $\underline{A}$ denotes the *context* of A, defined as $\underline{A} = [\![\mu C(A)]\!]$. The same notation is used to denote the functions from S to the powerset $\mathcal{P}(T)$, e.g., for $s \in S$ we define ${}^\bullet s = \{t \in T : F_{post}(t,s) > 0\}$.

An example of a contextual net, inspired by [18], is depicted in Fig. 2(a). Note that read arcs are drawn as undirected lines. For instance, referring to transition t_1 we have ${}^\bullet t_1 = s_1$, $t_1{}^\bullet = s_3$ and $\underline{t_1} = s_2$.

For a finite multiset of transitions A to be enabled at a marking M, it is sufficient that M contains the pre-set of A and one *additional* token in each place of the context of A. This corresponds to the intuition that a token in a place (like s in Fig. 1(a)) can be used as context concurrently by many transitions; instead, if read arcs are replaced by consume/produce loops (as in Fig. 1(b)) the transitions needing a token in place s can fire only one at a time.

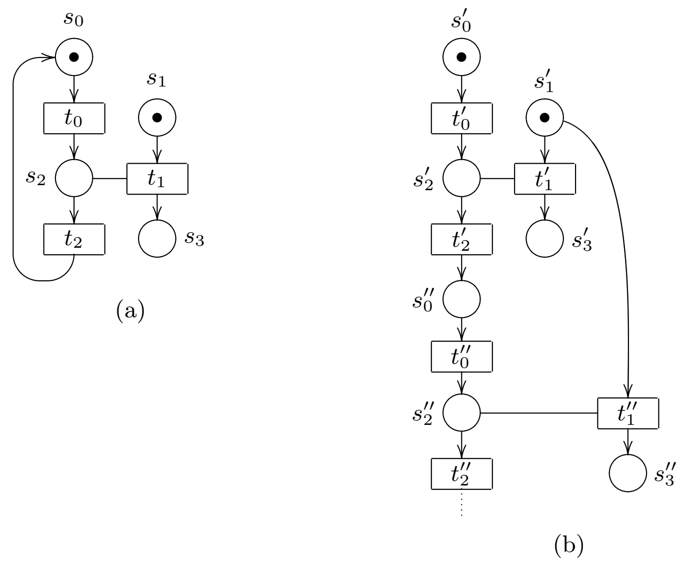


Fig. 2. (a) A contextual net N_0 and (b) its unfolding $\mathcal{U}_a(N_0)$.

Definition 2 (enabling, step). *Let N be a c-net. A finite multiset of transitions $A \in \mu_* T$ is* enabled *at a marking $M \in \mu_* S$ if ${}^{\bullet}A \oplus \underline{A} \leq M$. In this case, the execution of A in M, called a* step *(or a* firing *when it involves just one transition), produces the new marking $M' = M \ominus {}^{\bullet}A \oplus A^{\bullet}$, written as $M\,[A\rangle\, M'$.*

A marking M of a c-net N is called *reachable* if there is a finite sequence of steps leading to M from the initial marking, i.e., $m\,[A_0\rangle\, M_1\,[A_1\rangle\, M_2 \ldots [A_n\rangle\, M$.

Definition 3 (bounded, safe and semi-weighted nets). *A c-net N is called n-*bounded *if for any reachable marking M each place contains at most n tokens, namely $M(s) \leq n$ for all $s \in S$. It is called* safe *if it is 1-bounded and F_{pre}, F_{post} are relations (rather than general multirelations). A c-net N is called* semi-weighted *if the initial marking m is a set and F_{post} is a relation.*

Observe that requiring F_{pre} (resp. F_{post}) to be relations amounts to asking that for any transition $t \in T$, the pre-set (resp. post-set) of t is a set, rather than a general multiset.

We recall that considering semi-weighted nets is essential to characterise the unfolding construction, in categorical terms, as a coreflection [4]. However, in this paper, the choice of taking semi-weighted nets rather than general weighted nets is only motivated by the need of simplifying the presentation: the generalisation would require only some technical complications in the definition of the unfolding (Definition 10), related to the fact that an occurrence of a place would not be completely identified by its causal history.

2.2 Occurrence c-nets

Occurrence c-nets are safe c-nets such that the dependency relations among transitions that we will introduce, causality and asymmetric conflict, satisfy suitable acyclicity and well-foundedness requirements.

Causality is defined as for ordinary nets, with an additional clause stating that transition t causes t' if it generates a token in a context place of t'.

Definition 4 (causality). *Let N be a safe c-net. The* causality relation $<_N$ *is the least transitive relation on $S \cup T$ such that*

1. *if $s \in {}^{\bullet}t$ then $s <_N t$;*
2. *if $s \in t^{\bullet}$ then $t <_N s$;*
3. *if $t^{\bullet} \cap \underline{t'} \neq \emptyset$ then $t <_N t'$.*

Given $x \in S \cup T$, we write $\lfloor x \rfloor$ for the set of causes *of x in T, defined as $\lfloor x \rfloor = \{t \in T : t \leq_N x\} \subseteq T$, where $\leq_N$ is the reflexive closure of $<_N$.*

We say that a transition t is in *asymmetric conflict* with t', denoted $t \nearrow_N t'$, 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 [4, 15, 9, 18]).

Note that the fact that *whenever both t and t' fire, t fires before t'* trivially holds when $t <_N t'$, because t cannot follow t' in a computation, and (with t and t' in interchangeable roles) also when t and t' have a common precondition, since they will never fire in the same computation. For technical convenience the definition of $\nearrow_N$ takes these two situations into account as well.

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

$$t \nearrow_N t' \qquad \textit{iff} \qquad \underline{t} \cap {}^{\bullet}t' \neq \emptyset \;\; \textit{or} \;\; (t \neq t' \;\wedge\; {}^{\bullet}t \cap {}^{\bullet}t' \neq \emptyset) \;\; \textit{or} \;\; t <_N t'.$$

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

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

Definition 6 (occurrence c-nets). *An* occurrence c-net *is a safe c-net N such that*

- *each place $s \in S$ is in the post-set of at most one transition, i.e. $|{}^{\bullet}s| \leq 1$;*
- *the* causal relation $<_N$ *is irreflexive and its reflexive closure $\leq_N$ is a partial order, such that $\lfloor t \rfloor$ is finite for any $t \in T$;*
- *the* initial marking *is the set of minimal places w.r.t. $\leq_N$, i.e., $m = \{s \in S : {}^{\bullet}s = \emptyset\}$;*
- *$\nearrow_{\lfloor t \rfloor}$ is acyclic for all $t \in T$.*

The last condition corresponds to the requirement of irreflexivity for the conflict relation in ordinary occurrence nets. In fact, if a transition t has a $\nearrow_N$ cycle in its causes then it can never fire, since in an occurrence c-net N, the order in which transitions appear in a firing sequence must be compatible with the asymmetric conflict relation. An example of an occurrence c-net can be found in Fig. 2(b).

The notion of concurrency is the natural generalisation of the one for ordinary nets. Note that, because of the presence of contexts, some places that a transition needs in order to fire (the contexts) can be concurrent with the places it produces.

Definition 7 (concurrency relation). *Let N be an occurrence c-net. A finite set of places $M \subseteq S$ is called* concurrent*, written $conc(M)$, if*

1. $\forall s, s' \in M.\ \neg(s < s')$;
2. $\lfloor M \rfloor$ *is conflict-free, i.e.,* $\nearrow_{\lfloor M \rfloor}$ *is acyclic.*

It can be shown that, as for ordinary occurrence nets, a set of places M is concurrent if and only if there is some reachable marking in which all the places of M contain one token.

From now on, consistently with the literature, we shall often call the transitions of an occurrence c-net *events*.

Definition 8 (configuration). *Let N be an occurrence c-net. A set of events $C \subseteq T$ is called a* configuration *if*

1. $\nearrow_C$ *is well-founded;*
2. $\{t' \in C : t' \nearrow t\}$ *is finite for all* $t \in C$;
3. C *is left-closed w.r.t.* $\leq$, *i.e. for all* $t \in C$, $t' \in T$, $t' \leq t$ *implies* $t' \in C$.

We denote by $Conf(N)$ the set of all configurations of N, equipped with the ordering defined as $C \sqsubseteq C'$, if $C \subseteq C'$ and $\neg(t' \nearrow t)$ for all $t \in C, t' \in C' \setminus C$.

Furthermore two configurations C_1, C_2 are said to be in conflict *($C_1 \# C_2$) when there is no $C \in Conf(N)$ such that $C_1 \sqsubseteq C$ and $C_2 \sqsubseteq C$.*

The notion of configuration characterises the possible (concurrent) computations of an occurrence c-net. It can be proved that a subset of events C is a configuration iff the events in C can all be fired, starting from the initial marking, in any order compatible with $\nearrow$. The relation $\sqsubseteq$ is a computational order of configurations: $C \sqsubseteq C'$ if C can evolve and become C'. Remarkably, this order is not simply subset inclusion since a configuration C cannot be extended with an event t' if $t' \nearrow t$ for some $t \in C$, since t' cannot fire after t in a computation. Two configurations are in (symmetric) conflict if they do not have a common extension. More concretely $C_1 \# C_2$ when there exists $t_1 \in C_1$ and $t_2 \in C_2 \setminus C_1$ such that $t_2 \nearrow t_1$ or the symmetric condition holds.

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. Instead, for c-nets, due to the presence of asymmetric conflicts between events, an event t which occurs in more than one configuration may have different histories. The next definition formalises this fact.

Definition 9 (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 all possible histories of an event t, namely $\{C[\![t]\!] : C \in Conf(N) \wedge t \in C\}$ is denoted by $Hist(t)$.

2.3 Unfolding

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 [3, 18]. As for ordinary nets, each event in $\mathcal{U}_a(N)$ represents a particular firing of a transition in N, and places in $\mathcal{U}_a(N)$ represent occurrences of tokens in the places of N. The unfolding is equipped with a mapping to the original net N, sending each place (event) of the unfolding to the corresponding place (transition) in N.

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. We present an equivalent axiomatic definition, in the style of the one proposed by Winskel in [20].

Definition 10 (unfolding). *Let $N = \langle S, T, F, C, m\rangle$ be a semi-weighted c-net. The unfolding $\mathcal{U}_a(N) = \langle S', T', F', C', m'\rangle$ of the net N is the unique occurrence c-net satisfying the following (recursive) equations*

$$\begin{aligned}
m' &= \{\langle \emptyset, s\rangle : s \in m\} \\
S' &= \{m'\} \cup \{\langle t', s\rangle : t' = \langle M_p, M_c, t\rangle \in T' \;\wedge\; s \in t^\bullet\} \\
T' &= \{\langle M_p, M_c, t\rangle : M_p, M_c \subseteq S' \;\wedge\; M_p \cap M_c = \emptyset \;\wedge\; conc(M_p \cup M_c) \;\wedge \\
&\qquad t \in T \;\wedge\; \mu f_S(M_p) = {}^\bullet t \;\wedge\; \mu f_S(M_c) = \underline{t}\}
\end{aligned}$$

$$\begin{aligned}
&F'_{pre}(t', s') && \text{iff} && t' = \langle M_p, M_c, t\rangle \;\wedge\; s' \in M_p \quad (t \in T) \\
&C'(t', s') && \text{iff} && t' = \langle M_p, M_c, t\rangle \;\wedge\; s' \in M_c \quad (t \in T) \\
&F'_{post}(t', s') && \text{iff} && s' = \langle t', s\rangle \quad (s \in S)
\end{aligned}$$

where $f_N = \langle f_T, f_S\rangle : \mathcal{U}_a(N) \to N$ is the folding morphism*, consisting of 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 $s' = \langle x, s\rangle$.*

As said before, 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.

The unfolding is complete with respect to the behaviour of the original net in the following sense.

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

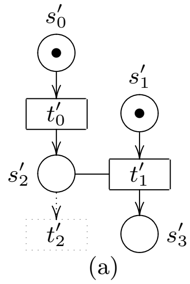
This is the notion of completeness that we will use in the rest of the paper: it is slightly weaker than that of [10, 18], for example, as it is concerned with markings only, and not with transitions.

3 Defining a Complete Finite Prefix

To obtain a finite prefix of the unfolding that is still complete in the sense of Proposition 1, the idea is to avoid to include useless events in the unfolding, where "useless" means events which do not contribute to generating new markings. To this aim McMillan introduced the notion of "cut-off" for ordinary nets, which is roughly an event whose history does not generate a new marking. Then the complete finite prefix is the greatest prefix without cut-offs. This definition of cut-off event has to be adapted to the present framework, since for contextual nets an event may have different histories, or, using McMillan terminology, different local configurations.

Considering only the minimal history of an event, i.e., its set of causes, in the definition of cut-off leads to a finite but not necessarily complete prefix, as observed in [18]. For instance, consider net N_0 in Fig. 2(a). According to the ordinary definition of cut-off, in its unfolding $\mathcal{U}_a(N_0)$ shown in Fig. 2(b) the event t'_2 would be a cut-off since its minimal history $\{t'_0, t'_2\}$ generates a marking corresponding to the initial marking. Thus the largest prefix without cut-offs would be the net O_0 in Fig. 3(a), which is not complete since it does not "represent" the marking $s_0 \oplus s_3$, reachable in N_0.

Considering instead all the possible histories of an event leads to a characterisation of a prefix which is finite and complete, even if this characterisation is not constructive since there can be infinitely many possible histories for a single



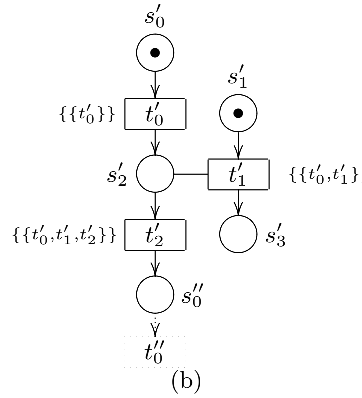


Fig. 3. (a) An incomplete and (b) a complete enriched prefix for the net in Fig. 2.

event (see [2]). In the present paper we suggest to record for each event only a subset of histories which are considered "useful to produce new markings".

To formalise this fact we introduce a notion of occurrence net decorated with possible histories for the involved events.

Definition 11 (enriched occurrence net). *An* enriched occurrence net *is a pair $E = \langle N, \chi \rangle$, where N is an occurrence net and $\chi : T \to \mathcal{P}(\mathcal{P}(T))$ is a function such that for any $t \in T$, $\emptyset \neq \chi(t) \subseteq Hist(t)$.*

The enriched occurrence net E is called closed *if for all $t, t' \in T$, for any $C \in \chi(t)$ if $t' \in C$ then $C[\![t']\!] \in \chi(t')$.*

A configuration of E is a configuration $C \in Conf(N)$ such that $C[\![t]\!] \in \chi(t)$ for all $t \in C$. The set of configurations of E is denoted by $Conf(E)$.

Often, given an enriched occurrence net E we will denote its components by N_E and χ_E. If the enriched net is E_i, we will call its components N_i and χ_i.

A generalisation of the natural prefix ordering over occurrence nets can be defined on enriched occurrence nets.

Definition 12 (prefix ordering). *Given two enriched occurrence nets E_1 and E_2, we say that E_1 is a* prefix *of E_2, written $E_1 \preceq E_2$, if N_1 is a prefix of N_2, and for any $t \in T_1$, $\chi_1(t) \subseteq \chi_2(t)$.*

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

Definition 13 (enriched event, enriched prefix). *An* enriched event *of the unfolding is a pair $\langle t, H_t \rangle$, where $t \in T'$ is an event of the unfolding, and $H_t \in Hist(t)$ is one of its histories. An* enriched prefix *of the unfolding $\mathcal{U}_a(N)$ is any* closed *enriched occurrence net E such that N_E is a prefix of $\mathcal{U}_a(N)$. We will say that the enriched prefix E contains $\langle t, H_t \rangle$ and write $\langle t, H_t \rangle \in E$ if $t \in T_E$ and $H_t \in \chi_E(t)$.*

An example of enriched prefix of $\mathcal{U}_a(N_0)$ in Fig. 2(b) is given in Fig. 3(b). For any event t the set of histories $\chi_E(t)$ is written near to the event itself.

It can be shown that the set of enriched prefixes of $\mathcal{U}_a(N)$ endowed with the prefix ordering $\preceq$ forms a lattice. Given two enriched prefixes E_1 and E_2, their least upper bound is $E_1 \sqcup E_2 = \langle N_E, \chi_E \rangle$, where N_E is the componentwise union of N_1 and N_2, and, for any event t in N, $\chi_E(t) = \bigcup_{\{i : t \in N_i\}} \chi_i(t)$. Moreover, it is not difficult to prove that given two enriched prefixes E_1 and E_2

$$E_1 \preceq E_2 \qquad \text{iff} \qquad \mathit{Conf}(E_1) \subseteq \mathit{Conf}(E_2).$$

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

Definition 14 (marking after a configuration). *Let $C \in \mathit{Conf}(\mathcal{U}_a(N))$ be a finite configuration. We denote by $\mathit{mark}(C)$ the marking of N after C, defined as $\mu f_S(m' \oplus \bigoplus_{t \in C} t^{\bullet} \ominus \bigoplus_{t \in C} {}^{\bullet}t)$.*

The notion of cut-off is now defined for enriched events, thus taking histories explicitly into account.

Definition 15 (cut-off). *An enriched event $\langle t, H_t \rangle$ of the unfolding $\mathcal{U}_a(N)$ is called a* cut-off *if either $\mathit{mark}(H_t) = m$, the initial marking of N, or there is another enriched event $\langle t', H_{t'} \rangle$ of $\mathcal{U}_a(N)$ satisfying*

(1) $\mathit{mark}(H_t) = \mathit{mark}(H_{t'})$ and
(2) $|H_{t'}| < |H_t|$.

Let E be an enriched prefix of the unfolding. We say that E contains a cut-off if some enriched event $\langle t, H_t \rangle \in E$ is a cut-off in the full unfolding $\mathcal{U}_a(N)$. The enriched event $\langle t, H_t \rangle \in E$ is called a local cut-off *in E if either $\mathit{mark}(H_t) = m$ or there is an enriched event $\langle t', H_{t'} \rangle \in E$ satisfying (1) and (2) above.*

A different notion of cut-off which refines the one originally proposed by McMillan by using *adequate orders* over configurations has been introduced in [6]. We are confident that this improvement can be integrated seamlessly into our framework, as mentioned in the conclusions.

Note that the notion of cut-off is based on a quantification over all the enriched events of the full unfolding and as such it is not effective. For an enriched event, being a cut-off is a global property, independent of the specific prefix of the unfolding we are considering. Clearly, every local cut-off in an enriched prefix E is also a cut-off. This simple observation will be used several times in the sequel.

Definition 16 (truncation). *The* truncation *$\mathcal{T}_a(N)$ of the unfolding is an enriched occurrence net defined as the greatest enriched prefix (w.r.t. prefix ordering $\preceq$) of the unfolding which does not contain cut-offs.*

The above definition is well-given thanks to the lattice structure of the set of enriched prefixes ordered by $\preceq$. However, it is not yet constructive. In Section 4 we will present an algorithm for computing a complete finite prefix, possibly larger than the truncation, using the notion of local cut-off.

We say that a configuration C of the unfolding includes a cut-off if for some $t \in C$, the enriched event $\langle t, C[\![t]\!]\rangle$ is a cut-off. The next fundamental lemma shows that configurations of the unfolding containing cut-offs can be disregarded without losing information about the reachable markings.

Lemma 1 (cut-off elimination). *Let $C \in Conf(\mathcal{U}_a(N))$ be a finite configuration. There exists a finite configuration C' without cut-offs such that $mark(C) = mark(C')$.*

Using the lemma above we can show that the truncation is a complete prefix of the unfolding.

Theorem 1 (completeness). *The truncation $\mathcal{T}_a(N)$ is a complete prefix of the unfolding, i.e., for any reachable marking M of N there is a finite configuration C of $\mathcal{T}_a(N)$ such that $mark(C) = M$.*

For finite n-bounded nets the number of reachable states of the net is finite and thus one can prove that the truncation of its unfolding is finite. We get this as a corollary of a more general result which will be useful in proving the termination of the algorithm for the complete prefix.

Theorem 2 (finiteness). *Let N be a finite n-bounded c-net and let E be an enriched prefix of the unfolding free of local cut-offs. Then E is finite.*

Recalling that any local cut-off is a cut-off and thus that $\mathcal{T}_a(N)$ is free from local cut-offs we have the following.

Corollary 1. *Let N be a finite n-bounded net. The truncation $\mathcal{T}_a(N)$ is finite.*

For instance, consider the net N_0 and its unfolding $\mathcal{U}_a(N_0)$ in Fig. 2. The truncation $\mathcal{T}_a(N_0)$ is the enriched prefix depicted in Fig. 3(b). 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 in the truncation.

4 Computing the prefix

The construction builds incrementally a finite prefix of the full unfolding of a semi-weighted c-net N by starting from the initial marking and by iteratively adding new events representing occurrences of transitions of N. For each event t in Fin, the currently built part of the prefix, we also record a current set of histories $\chi_{Fin}(t)$, thus making the prefix under construction an enriched occurrence net. During the construction we record in a set pe the enriched events which are

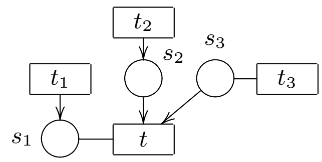


Fig. 4. Predecessors w.r.t. asymmetric conflict of an event t.

candidates for being included in *Fin*, i.e., the pairs $\langle t, H\rangle$ where t is an event enabled in *Fin* and H is one of its current possible histories.

Let us first illustrate how the histories of an event t in a given enriched prefix E can be obtained from the histories of the events that are in direct asymmetric conflict with t. Consider a situation like in Fig. 4, which illustrates a part of the prefix E. A direct predecessor of t w.r.t. asymmetric conflict is either a cause (like t_1, which produces a token that is read, or t_2, which produces a token that is consumed by t) or an event as t_3 that reads a token consumed by t.

A new history for t can be constructed as follows: for every direct cause t_i of t choose any history H_i of t_i, while for every transition t_j that is in direct asymmetric conflict with t (but not a cause) optionally take any history H_j. Whenever such histories are pairwise not in conflict (see Definition 8) then the set $H = \{t\} \cup \bigcup_i H_i$, the union of all such histories (and t), is called a *history for* t *consistent with* E.

Note that $H \in \mathit{Hist}(t)$ and furthermore adding H to E keeps the prefix closed, since for every transition $t' \in H$ the history $H[\![t']\!]$ is already contained in E. This is a consequence of the fact that for any t_i we have $H[\![t_i]\!] = H_i$ since no two histories in the union are in conflict.

The algorithm proceeds as follows. Again we use the notation of Definition 10.

Initialization: Start with $\mathit{Fin} := m'$ and let χ_{Fin} be the empty function. An event $t = \langle M_p, M_c, \hat{t}\rangle$ is enabled in *Fin* whenever $\mathit{conc}(M_p \cup M_c)$. Now let *pe* be the set of all pairs of the form $\langle t, H_t\rangle$, where t is an event enabled in *Fin* and H_t is a history of t consistent with *Fin*. Initially the only history of t is $\{t\}$.

Loop: While $\mathit{pe} \neq \emptyset$ do: Choose a pair $\langle t, H_t\rangle \in \mathit{pe}$ such that $|H_t|$ is minimal. Remove this pair from *pe*.

- If $\langle t, H_t\rangle$ would be a local cut-off in *Fin*, do nothing.
- If $\langle t, H_t\rangle$ is not a local cut-off, then insert it into *Fin*. This means
  - if t is already present in *Fin* then add the history H_t to $\chi_{\mathit{Fin}}(t)$;
  - otherwise add t to *Fin* and set $\chi_{\mathit{Fin}}(t) := \{H_t\}$.

  Consider all events t' contained either in *Fin* or in *pe*: Whenever t' has a new history $H_{t'}$ consistent with the updated prefix *Fin*, arising from the insertion of H_t, then add $\langle t', H_{t'}\rangle$ to *pe*. (Note that a propagation phase is necessary to obtain all new histories.)

If a new transition has been added to *Fin*, update *pe* by adding all events t which have become enabled in *Fin* in the last step together with all their histories consistent with *Fin*. Then perform the next step of the loop.

Note that whenever a new pair $\langle t', H_{t'} \rangle$ is added to *pe*, then the size of $H_{t'}$ is larger than the size of the history H_t under consideration. This is due to the fact that these newly generated histories must include H_t. Observe also that all pairs $\langle t, H \rangle$ with $H \in \mathit{Hist}(t)$ are considered at some point, unless there exists a local cut-off $\langle t', H' \rangle$ such that $t' \in H$ and $H' = H[\![t']\!]$.

An efficient computation of the prefix should be based on suitable data structures. As observed above, a set of direct predecessors is needed for each event in order to update its histories. Furthermore histories should not be stored explicitly, but via pointer structures containing references back to the histories they originated from. In addition, causality and conflict of histories should be computed incrementally. To this aim it would be helpful to keep trace of all the $\nearrow$-sequences $t_1 \nearrow \ldots \nearrow t_n$ in order to support an easy identification of $\nearrow$-cycles.

It can be shown that at every iteration of the algorithm the prefix *Fin* does not contain local cut-offs. This can be used to prove the correctness and termination of the algorithm.

Theorem 3. *If the net N is finite and n-bounded the algorithm terminates and Fin is complete.*

The complete prefix of a c-net can be much smaller than the complete prefix (constructed using McMillan's algorithm) for the net where read arcs are replaced by consume/produce loops. In fact, consider a net N_1^n analogous to the net in Fig. 1(a) but with n readers $t_1, \ldots, t_n$. Let N_2^n be obtained encoding N_1^n as an ordinary net by simply replacing read arcs with a consume/produce loops, as in Fig. 1(b). The unfolding of net N_2^n includes $k_n = n + n(n-1) + \ldots + n!$ events corresponding to the readers, since each event does not only record the occurrence of a transition, but also its entire history, i.e., the sequence of all events occurring before. Similarly, there are $k_n + 1$ copies of event t_0'. Note that none of these events is a cut-off (according to McMillan's definition), since any two events generating the same marking have histories of equal size. Therefore the complete prefix computed for N_2^n is the unfolding itself. Instead, the complete enriched prefix obtained from N_1^n is the net N_1^n itself, thus it has $n+2$ transitions only; among them, $t_0, t_1, \ldots, t_n$ have one history each, while t_0' has 2^n histories. Even if still of exponential size, this prefix is much smaller than the complete prefix of N_2^n, essentially because the order in which the events occurred does not need to be recorded. Moreover, the underlying net obtained by disregarding the histories, which are only auxiliary information needed to construct the prefix, is dramatically smaller in this case.

Now let N_3^n be the PR-encoding of N_1^n, as shown in Fig. 1(c). The unfolding of N_3^n has one occurrence for each of the transitions $t_0, t_1, \ldots, t_n$ and 2^n occurrences of t_0', none of which is a cut-off (hence, also in this case, the complete prefix is the full unfolding). Thus there is a one-to-one correspondence between the histories in the enriched prefix of N_1^n and the events of the unfolding of N_3^n. We conjecture that this is a general fact, i.e., the histories of the complete enriched prefix of a safe c-net N are in one-to-one correspondence with the events of the complete finite prefix of the PR-encoding of N. Still, the size of the prefix of N_3^n is exponential in n while the size of the prefix of N_1^n, once the histories are disregarded, is linear.

It is worth stressing that the size of the complete prefixes can be further reduced using adequate orders [6], as remarked also in the conclusion. This would lead to a smaller prefix, for example, for N_2^n.

5 Conclusions

We have presented an approach for computing finite complete prefixes of general contextual nets, which extends the approach proposed for the class of read-persistent nets in [18] and provides an alternative to the technique based on the PR-encoding of contextual nets as ordinary nets. Our work relies on the idea of dealing explicitly with the multiple histories that events can have in contextual net computations, due to the presence of asymmetric conflicts. Subsets of "useful" histories for events are recorded in the prefix during the construction and, correspondingly, a new notion of cut-off is considered. In the case of read-persistent nets every transition has a single history and hence our approach coincides with the one introduced in [18].

Our work shares some basic ideas with [19], where however the definition of cut-off is non-constructive, since it depends on all the possible histories that an event may have. In order to avoid this problem we introduced the (constructive) notion of local cut-off. Apart from that the notion of cut-off in [19] is stronger than ours, which might lead to larger prefixes.

As witnessed by some examples in the paper, the complete prefix of a contextual net can be significantly smaller than that of an equivalent net where read arcs are replaced by consume/produce loops, and it will never be larger. The ability to generate smaller unfoldings comes with a price, i.e., during the construction of the prefix we have to record and evaluate additional information such as histories and asymmetric conflict. Still, we conjecture that the algorithm will never require more space or time than the ordinary algorithm applied to the PR-encoding of the net. More precisely, for safe nets, as discussed in Section 3 the histories in the prefix should correspond exactly to the events in the unfolding of the PR-encoding, and causality and conflict on histories should be the exact match to causality and conflict for transitions. Furthermore we expect our technique to be strictly more efficient for non-safe nets (see Appendix A), since, in this case, the PR-encoding can lead to concurrent occurrences of the same reader where the occurrences share places in consume/produce loops, leading to a blowup in the size of the unfolding.

From a more methodological perspective, let us stress that our approach can build a complete finite prefix for a large class of c-nets directly, without the need of resorting to an encoding. We think that this feature makes our approach more suitable than others to be extended to other classes of systems exhibiting concurrent read-only accesses, for which an encoding could either not be feasible or could cause a significant loss of concurrency.

In particular, we are interested in graph transformation systems (GTSs), a quite expressive formalism where reading and preserving part of the system state, in this case a graph, is an integral part of the model. We believe that our direct approach will be useful to generalise McMillan's approach to the full class of GTSs, while currently only its read-persistent subclass is dealt with [2]. We are also interested in nets with inhibitor arcs. In this case, an encoding as c-nets would be feasible but it would cause (at least in the non-safe case) a loss of concurrency, and thus a direct approach could be preferable.

We plan to implement and test the algorithm for contextual nets in the framework of the Mole unfolder [1] that currently deals with ordinary nets. At present, with the limited goal of analyzing the size of the produced prefix, we implemented a prototype which given a safe c-net, converts the read arcs into consume/produce loops, builds its finite prefix, and then merges the occurrences of the same context places. A complete implementation of our algorithm is currently in progress. We expect that in order to obtain satisfactory experimental results about the complexity (in time and in space) of our algorithm, in comparison with others, firstly we will need to be able to deal with more refined notions of cut-offs based on adequate orders [6], and secondly we will have to design and implement efficient data structures for recording the sets of histories of an event during the construction of the prefix.

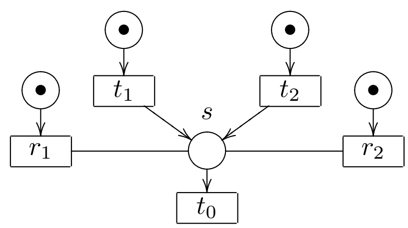
References

1. The Mole unfolder. Available at `http://www.fmi.uni-stuttgart.de/szs/tools/mole`.
2. P. Baldan, A. Corradini, and B. König. Verifying finite-state graph grammars: an unfolding-based approach. In P. Gardner and N. Yoshida, editors, *Proceedings of CONCUR 2004*, volume 3170 of *LNCS*, pages 83–98. Springer Verlag, 2004.
3. P. Baldan, A. Corradini, and U. Montanari. An event structure semantics for P/T contextual nets: Asymmetric event structures. In M. Nivat, editor, *Proceedings of FoSSaCS '98*, volume 1378 of *LNCS*, pages 63–80. Springer Verlag, 1998.
4. P. Baldan, A. Corradini, and U. Montanari. Contextual Petri nets, asymmetric event structures and processes. *Information and Computation*, 171(1):1–49, 2001.
5. S. Christensen and N. D. Hansen. Coloured Petri nets extended with place capacities, test arcs and inhibitor arcs. In M. Ajmone-Marsan, editor, *Applications and Theory of Petri Nets*, volume 691 of *LNCS*, pages 186–205. Springer Verlag, 1993.
6. J. Esparza, S. Römer, and W. Vogler. An improvement of McMillan's unfolding algorithm. *Formal Methods in System Design*, 20:285–310, 2002.
7. R. Janicki and M. Koutny. Invariant semantics of nets with inhibitor arcs. In *Proceedings of CONCUR '91*, volume 527 of *LNCS*. Springer Verlag, 1991.
8. R. Janicki and M. Koutny. Semantics of inhibitor nets. *Information and Computation*, 123:1–16, 1995.
9. R. Langerak. *Transformation and Semantics for LOTOS*. PhD thesis, Department of Computer Science, University of Twente, 1992.

10. K.L. McMillan. Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits. In *Proceedings of CAV '92, Fourth Workshop on Computer-Aided Verification*, volume 663 of *LNCS*, pages 164–174. Springer Verlag, 1992.
11. K.L. McMillan. *Symbolic Model Checking*. Kluwer, 1993.
12. U. Montanari and F. Rossi. Contextual occurrence nets and concurrent constraint programming. In H.-J. Schneider and H. Ehrig, editors, *Proceedings of the Dagstuhl Seminar 9301 on Graph Transformations in Computer Science*, volume 776 of *LNCS*. Springer Verlag, 1994.
13. U. Montanari and F. Rossi. Contextual nets. *Acta Informatica*, 32(6):545–596, 1995.
14. M. Nielsen, G. Plotkin, and G. Winskel. Petri Nets, Event Structures and Domains, Part 1. *Theoretical Computer Science*, 13:85–108, 1981.
15. G. M. Pinna and A. Poigné. On the nature of events: another perspective in concurrency. *Theoretical Computer Science*, 138(2):425–454, 1995.
16. G. Ristori. *Modelling Systems with Shared Resources via Petri Nets*. PhD thesis, Department of Computer Science - University of Pisa, 1994.
17. W. Vogler. Efficiency of asynchronous systems and read arcs in Petri nets. In *Proceedings of ICALP'97*, volume 1256 of *LNCS*, pages 538–548. Springer Verlag, 1997.
18. W. Vogler, A. Semenov, and A. Yakovlev. Unfolding and finite prefix for nets with read arcs. In *Proceedings of CONCUR'98*, volume 1466 of *LNCS*, pages 501–516. Springer Verlag, 1998.
19. J. Winkowski. Reachability in contextual nets. *Fundamenta Informaticae*, 51(1):235–250, 2002.
20. G. Winskel. Event Structures. In *Petri Nets: Applications and Relationships to Other Models of Concurrency*, volume 255 of *LNCS*, pages 325–392. Springer Verlag, 1987.

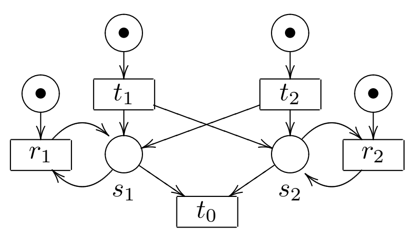
A Unfolding of Non-Safe Nets

Unfolding of a non-safe contextual net N with the algorithm proposed in this paper might lead to an occurrence net smaller than the unfolding of the ordinary net obtained as the PR-encoding of N (see Fig. 1(c)). As an example consider the following net:



The truncation of this net has two occurrences of transition t_0 (either t_0 is caused by t_1 or by t_2), each with four histories (which specify whether r_1 or r_2, or both, or none has been fired before). So in total we have eight histories.

Now consider the corresponding PR-encoding:



Unfolding this net we obtain four occurrences of place s_1 (after firing t_1 or $t_1;r_1$ or t_2 or $t_2;r_1$) and analogously four occurrences of place s_2. All pairs of such places (one representing s_1 and the other s_2) are concurrent. Hence we obtain $4 \cdot 4 = 16$ occurrences of transition t_0.

Intuitively this can be interpreted as follows: the token in s is split into two half-tokens in s_1 and s_2. Then some of the transitions in the unfolding of the encoded net consume "half a token" produced by t_1 and "half a token" produced by t_2.

More generally, consider a net like the one above, but with h writers $t_1, \ldots,$ t_h and k readers $r_1, \ldots, r_k$. Then the truncation of the contextual net has h occurrences of t_0 with a total number of histories $h \cdot 2^k$, since t_0 can consume the token produced by any of the h writers, after it has been read by any subset of the k readers. Instead, the unfolding of the PR-encoding of the net includes $(h \cdot 2)^k$ occurrences of t_0, since each occurrence of t_0 consumes k tokens, and each of these tokens can be produced by any of the h readers and it could have possibly been produced/consumed by the corresponding reader.