Construction and SAT-Based Verification of Contextual Unfoldings

Stefan Schwoon and César Rodríguez

LSV, ENS de Cachan & CNRS, INRIA
61 avenue du Président Wilson, 94230 Cachan, France
{schwoon,rodrigue}@lsv.ens-cachan.fr

Abstract. Unfoldings succinctly represent the set of reachable markings of a Petri net. Here, we shall consider the case of contextual nets, which extend Petri nets with read arcs, and which are more suitable to represent the case of concurrent read access. We discuss the problem of (efficiently) constructing unfoldings of such nets. On the basis of these unfoldings, various verification problems can be encoded as satisfiability problems in propositional logic.

1 Introduction

Petri nets are generally recognized as an adequate formal model for concurrent, distributed systems. The unfolding of a Petri net is, essentially, an acyclic version of the net in which the loops have been unrolled. While in general infinite, one can construct a finite "complete" prefix of it that represents all its behaviours, and whose acyclic structure permits easier analysis. The resulting object is typically larger than the net, but much smaller than the number of its reachable markings. The idea of using unfoldings for verifying properties of Petri nets was first employed by McMillan [7] and subsequently greatly expanded by many other works, see [3] for a survey. On 1-safe Petri nets, for instance, the reachability problem is PSPACE-complete, whereas the corresponding problem is only NP-complete for complete prefixes. Notwithstanding the fact that the size of the unfolding is rather larger than the net itself, this opens avenues for efficient reachability checking [5].

The success of unfolding-based techniques is due to the fact that unfoldings exploit the inherently concurrent nature of the underlying system; loosely speaking, the more concurrency there is in the net, the more advantages unfoldings have when compared to reachability-graph techniques.

However, Petri nets are not expressive enough to adequately model concurrent read accesses to the same resource, that is, if multiple actions at the same time require non-exclusive access to one common resource. Consequently, the unfolding technique becomes inefficient in the presence of such situations.

Contextual nets address this problem. They extend Petri nets with *read arcs*, which allow transitions to require the presence of a token on certain places without actually consuming them. They have been used, e.g., to model concurrent

M. Holzer, M. Kutrib, and G. Pighizzini (Eds.): DCFS 2011, LNCS 6808, pp. 34–42, 2011.

database access [9], concurrent constraint programs [8], priorities [6], and asynchronous circuits [12]. Modelling using read arcs allows unfolding-based techniques to take advantage of concurrency in the model, resulting in unfoldings that are up to exponentially smaller in the presence of multiple readers.

This advantage comes at a price: read arcs introduce a phenomenon known as *asymmetric conflict*, meaning that an event e may not be necessary for another event e' to happen, but if both e and e' happen, then e must happen first. (This phenomenon is absent in the Petri net case.) Asymmetric conflicts complicate the theoretical foundations by no small amount, which explains why contextual unfolding techniques were slow to be developed. A first approach by Vogler, Semenov, and Yakovlev proposed an unfolding procedure for a restricted subclass [12]. Winkowski proposed a general but non-constructive procedure [13]. A constructive, general solution was finally given in [2], and steps towards a concrete implementation were undertaken in [1] and more recently in [11], resulting in the tool Cunf [10].

Experiments in [11] show that contextual unfoldings are not only more succinct but can generally be constructed more quickly then Petri unfoldings. Moreover, as we shall see, the reachability problem remains in NP, and we shall discuss an encoding of this and related problems in SAT.

The rest of the paper is structured as follows. In Section 2 we define contextual nets and their unfoldings. In Section 3 we summarize the salient points about their construction, and in Section 4, we discuss SAT-based approaches to some verification problems.

2 Contextual Nets and Their Unfoldings

In this section, we introduce contextual nets and their unfoldings. We also discuss an example showing the greater succinctness of contextual unfoldings when compared to normal unfoldings. A more expanded treatment of the subject can be found, e.g., in [2] or [11].

2.1 Contextual Nets

A *contextual net (c-net)* is a tuple $N = \langle P, T, F, C, m_0 \rangle$, where P and T are disjoint sets of *places* and *transitions*, $F \subseteq (P \times T) \cup (T \times P)$ is the *flow relation*, and $C \subseteq P \times T$ is the *context relation*. A pair $(p, t) \in C$ is called *read arc*. Any function $m \colon P \to \mathbb{N}$ is called a *marking*, and m_0 is the *initial marking*. A *Petri net* is a c-net without any read arcs.

For $x \in P \cup T$, we call ${}^\bullet x := \{\, y \in P \cup T \mid (y, x) \in F \,\}$ the *preset* of x, and $x^\bullet := \{\, y \in P \cup T \mid (x, y) \in F \,\}$ the *postset* of x. The *context* of a place p is defined as $\underline{p} := \{\, t \in T \mid (p, t) \in C \,\}$, and the context of a transition t as $\underline{t} := \{\, p \in P \mid (p, t) \in C \,\}$. These notions are extended to sets in the usual fashion.

A marking m is *n-safe* if $m(p) \leq n$ for all $p \in P$. A set $A \subseteq T$ of transitions is *enabled* at a marking m if for all $p \in P$,

$$m(p) \geq |p^\bullet \cap A| + \begin{cases} 1 & \text{if } \underline{p} \cap A \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

Such A may *fire*, leading to a new marking m', where $m'(p) = m(p) - |p^\bullet \cap A| + |{}^\bullet p \cap A|$ for all $p \in P$. We call $\langle m, A, m' \rangle$ a *step* of N.

A finite sequence of transitions $\sigma = t_1 \dots t_n \in T^*$ is a *run* if there exist markings $m_1, \dots, m_n$ such that $\langle m_{i-1}, \{t_i\}, m_i \rangle$ is a step for $1 \leq i \leq n$, and m_0 is the initial marking of N; if such a run exists, m_n is said to be *reachable*. A c-net N is said to be n-safe if every reachable marking of N is n-safe.

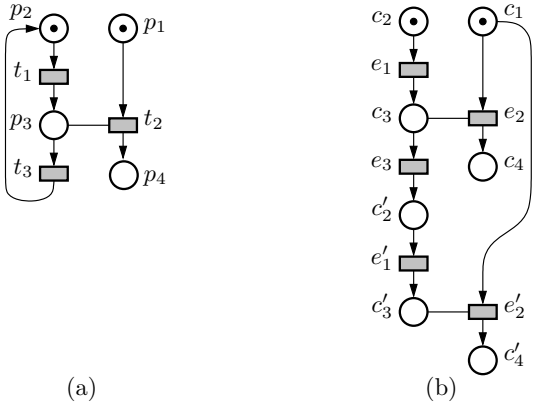

Fig. 1. (a) A 1-safe c-net; (b) One of its unfolding prefixes (right)

Fig. 1 (a) depicts a 1-safe c-net. Read arcs are drawn as undirected lines. For t_2, we have $\{p_1\} = {}^\bullet t_2$, $\{p_3\} = \underline{t_2}$ and $\{p_4\} = t_2^\bullet$.

In this treatment, we restrict our interest to finite 1-safe c-nets and treat markings as sets of places. Notice, however, that the general theory of contextual unfolding developed in [2] extends to semi-weighted nets.

2.2 Unfoldings

Let $N = \langle P, T, F, C, m_0 \rangle$ be a c-net. Intuitively, the unfolding of N is a 1-safe acyclic c-net where loops of N are "unrolled".

Definition 1. *The unfolding of N, written $\mathcal{U}_N$, is a c-net $(B, E, G, D, \widehat{m}_0)$ equipped with a mapping $f\colon (B \cup E) \to (P \cup T)$, which we extend to sets and sequences in the usual way. We call the elements of B* conditions*, and those of E* events*; f maps conditions to places and events to transitions.*

Conditions will take the form $\langle p, e' \rangle$, where $p \in P$ and $e' \in E \cup \{\perp\}$, and events will take the form $\langle t, M \rangle$, where $t \in T$ and $M \subseteq B$. We shall assume $f(\langle p, e' \rangle) = p$ and $f(\langle t, M \rangle) = t$, respectively. A set M of conditions is called concurrent *iff $\mathcal{U}_N$ has a reachable marking M' s.t. $M' \supseteq M$.*

$\mathcal{U}_N$ is the smallest net containing the following elements:

- *if* $p \in m_0$*, then* $\langle p, \bot \rangle \in B$ *and* $\langle p, \bot \rangle \in \widehat{m}_0$*;*
- *for any* $t \in T$ *and disjoint pair of sets* $M_1, M_2 \subseteq B$ *such that* $M_1 \cup M_2$ *is concurrent,* $f(M_1) = {}^\bullet t$*,* $f(M_2) = \underline{t}$*, we have* $e := \langle t, M_1 \cup M_2 \rangle \in E$*, and for all* $p \in t^\bullet$*, we have* $\langle p, e \rangle \in B$*. Moreover,* G *and* D *are such that* ${}^\bullet e = M_1$*,* $\underline{e} = M_2$*, and* $e^\bullet = \{ \langle p, e \rangle \mid p \in t^\bullet \}$*.*

For example, Fig. 1 (b) shows the beginning of the unfolding of the c-net from Fig. 1 (a). In general, unfoldings are infinite. We shall study finite portions of them that contain all information about reachable markings:

Definition 2. *Let* x, y *be nodes. We write* $x < y$ *if either* $(x, y) \in G$*, or* x, y *are events such that* $x^\bullet \cap \underline{y} \neq \emptyset$*, or* (x, y) *is in the transitive closure of the first two cases. We define* $[x] := \{ e \in E \mid e \leq x \}$ *as the set of* causes *of* x*. A set* $X \subseteq E$ *is called* causally closed *if* $[e] \subseteq X$ *for all* $e \in X$*. A* prefix *of* $\mathcal{U}_N$ *is a net* $\mathcal{P} = \langle B', E', G', D', \widehat{m}_0 \rangle$ *such that* $E' \subseteq E$ *is causally closed,* $B' = \widehat{m}_0 \cup (E')^\bullet$*, and* G', D' *are the restrictions of* G, D *to* $(B' \cup E')$*.*

In Fig. 1 (b), we have, e.g., $c_2 < e_1$, $e_1 < e_2$, and $c_2 < e_2$. If e, e' are two events with $e < e'$, then e must occur before e' in any run that fires e'. A prefix is a causally-closed subnet of $\mathcal{U}_N$. We are interested in prefixes that have the same markings as N itself, modulo f.

Definition 3. *A prefix* $\mathcal{P}$ *is called* complete *if for all markings* m*,* m *is reachable in* N *iff there exists a marking* $\widehat{m}$ *reachable in* $\mathcal{P}$ *such that* $f(\widehat{m}) = m$*.*

2.3 Succintness of Contextual vs. Petri Net Unfoldings

For Petri nets, there are methods of constructing complete finite prefixes that may be exponentially smaller than the reachability graph, and at worst no larger than it [4]. Here, we shall make another observation: the unfolding of a c-net may be exponentially more succinct than the unfolding of a Petri net that has the same set of reachable markings.

Consider the c-net N depicted in Fig. 2 (a). The place p has two transitions b, c in its context. This models a situation where, e.g., two processes are read-accessing a common resource modelled by p. Notice that $\{b, c\}$ can fire in N. Fig. 2 (b) shows a Petri net N' in which read arcs have been replaced by read/write loops. Evidently, N' has the same set of reachable markings as N.

However, $\{b, c\}$ cannot fire in N': the encoding step does not preserve the concurrency between b and c. As a result, the unfolding of N' enumerates both the sequence "first b, then c" and the inverse, whereas the unfolding of N is actually identical to N itself, see Fig. 3. One easily sees that if b and c were replaced by n transitions reading from p, the unfolding of N' would become exponentially larger than the one of N.

We note in passing that this blowup can be mitigated, but not completely avoided, by a less naïve transformation into Petri nets provided by Vogler et al [12].

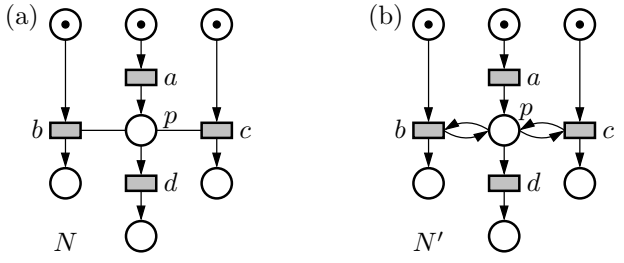

Fig. 2. C-net N and Petri net N'

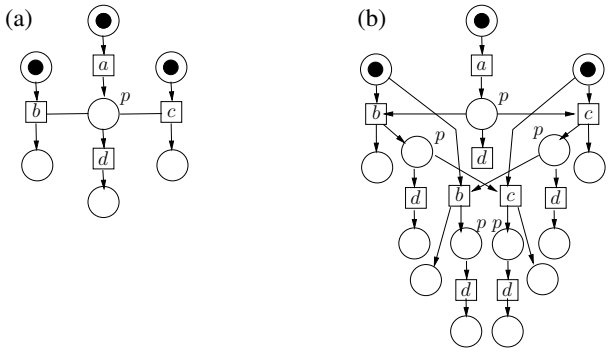

Fig. 3. Unfoldings of N and N' from Fig. 2

3 Constructing Contextual Unfoldings

We sketch the main ideas for the algorithms behind efficiently constructing complete finite unfolding prefixes. Again, an expanded treatment can be found in [2,11]. Notice that Definition 7 and Proposition 8 follow a new and slightly more elegant approach than was put forward in [11].

For the rest of the section, let us fix a 1-safe c-net $N = \langle P, T, F, C, m_0 \rangle$ and its unfolding $\mathcal{U}_N = \langle B, E, G, D, \widehat{m}_0 \rangle$. We first elaborate on the notion of asymmetric conflict that is characteristic for c-nets.

Definition 4. *Two events $e, e' \in E$ are in* asymmetric conflict*, written $e \nearrow e'$, iff (i) $e < e'$, or (ii) $\underline{e} \cap {}^\bullet e' \neq \emptyset$, or (iii) $e \neq e'$ and ${}^\bullet e \cap {}^\bullet e' \neq \emptyset$. For a set of events $X \subseteq E$, we write $\nearrow_X$ to denote the relation $\nearrow \cap\, X \times X$. A* configuration *of $\mathcal{U}_N$ is a finite, causally closed set of events $\mathcal{C}$ such that $\nearrow_{\mathcal{C}}$ is acyclic. A* history *of an event e is a configuration H such that $e'(\nearrow_H)^* e$ for all $e' \in H$. For a configuration $\mathcal{C}$ we define $\mathsf{Cut}(\mathcal{C}) := (\widehat{m}_0 \cup \mathcal{C}^\bullet) \setminus {}^\bullet\mathcal{C}$, and its* marking *as its image through f: $\mathsf{Mark}(\mathcal{C}) := f(\mathsf{Cut}(\mathcal{C}))$.*

Asymmetric conflict can be thought of as a scheduling constraint: if both e, e' occur in a run, then e must occur first (in case (iii) this is vacuously the case, as e, e' cannot both occur). The notion of configuration captures sets of events

that can be arranged to form a run that respects $\nearrow$; naturally, such a set must be free of cyclic scheduling constraints. A history of e is a configuration in which e necessarily fires last.

Definition 5. *Configurations $\mathcal{C}$, $\mathcal{C}'$ are said to be in* conflict, *written $\mathcal{C} \# \mathcal{C}'$, if there exists some $e \in \mathcal{C}$ and $e' \in \mathcal{C}' \setminus \mathcal{C}$ such that $e' \nearrow e$, or the inverse condition holds.*

Intuitively, $\mathcal{C} \# \mathcal{C}'$ means that $\mathcal{C}$ and $\mathcal{C}'$ represent "diverging" runs of the unfolding; from either $\mathcal{C}$ or $\mathcal{C}'$ it is no longer possible to obtain a run that contains the elements of $\mathcal{C} \cup \mathcal{C}'$.

Definition 6. *Let c be a condition. A* generating history *of c is $\emptyset$ if $c \in \widehat{m}_0$, or H if $\{e\} = {}^\bullet c$ and H is a history of e. A* reading history *of c is any H such that $e \in \underline{c}$ and H is a history of e. A* history *of c is any of its generating or reading histories or $H_1 \cup H_2$, where H_1 and H_2 are histories of c verifying $\neg(H_1 \# H_2)$.*

The key idea of [2], in order to construct a complete prefix of $\mathcal{U}_N$, is to consider pairs $\langle e, H \rangle$, where H is a history of e. Definition 7 in combination with Proposition 8 gives a unique characterization of these pairs.

Definition 7. *A pair $\langle c, H \rangle$, where H is a history of c, is called an* enriched condition. *The pair $\rho = \langle c, H \rangle$ is said to be* asymmetrically concurrent *to $\rho' = \langle c', H' \rangle$ iff (i) $\neg(H \# H')$, (ii) $c, c' \in \mathsf{Cut}(H \cup H')$, and (iii) $\underline{c} \cap H' \subseteq H$. If this is the case, we write $\rho \mathbin{//} \rho'$.*

Proposition 8. *Let e be an event of $\mathcal{U}_N$ such that ${}^\bullet e = \{c_1, \ldots, c_k\}$ and $\underline{e} = \{c_{k+1}, \ldots, c_n\}$. Then H is a history of e iff there exist enriched conditions $\rho_i = \langle c_i, H_i \rangle$, for $i = 1, \ldots, n$, such that*

1. *$H = \{e\} \cup \bigcup_{i=1}^{n} H_i$;*
2. *$H_{k+1}, \ldots, H_n$ are generating histories of $c_{k+1}, \ldots, c_n$;*
3. *$\rho_i \mathbin{//} \rho_j$ for $i = 1, \ldots, k$ and $j = 1, \ldots, n$;*
4. *$\rho_i \mathbin{//} \rho_j$ or $\rho_j \mathbin{//} \rho_i$ for $i, j \in k+1, \ldots, n$.*

The binary relation // is efficiently computable [11,10] and allows to discover algorithmically the histories and therefore the events that constitute $\mathcal{U}_N$. Moreover, [2,11] provide criteria for determining which event and histories need to be considered when constructing a complete prefix of $\mathcal{U}_N$; experimental results with Cunf are reported in [11].

4 Verifying Properties about Contextual Nets Using SAT

In this section we briefly give some examples showing how the complete prefix of a c-net unfolding can be used to answer questions about the c-net itself. For this, we adapt the SAT-based reductions of [3] to the contextual case.

We start by recalling that the following problem is NP-complete, see e.g. [3]:

> Given a complete prefix of $\mathcal{U}_N$, where N is a 1-safe *Petri* net, and a marking m of N, is m reachable in N?

It is straightforward to see that the result extends to the case where N is a general c-net (with read arcs).

This result suggests that the reachability problem can be encoded as a satisfiability problem in propositional logic, using a formula whose size is proportional to that of $\mathcal{U}_N$. The key idea is to first construct a formula ϕ_N that characterizes the configurations and markings of $\mathcal{U}_N$. More precisely, for every condition c and event e of $\mathcal{U}_N$, ϕ_N will contain a variable $\mathbf{c}$ and $\mathbf{e}$, respectively; the models of ϕ_N will be those assignments in which the event variables with value true correspond to some configuration $\mathcal{C}$ and the true condition variables to $\mathsf{Mark}(\mathcal{C})$.

ϕ_N will be a conjunction of formulae ensuring that $\mathcal{C}$ is indeed a configuration, that is, causally closed and free of asymmetric-conflict cycles; variable $\mathbf{c}$ will be true iff condition c is produced but not consumed by $\mathcal{C}$. For instance, suppose that condition c has ${}^\bullet c = \{e\}$, $c^\bullet = \{f_1, \ldots, f_m\}$, and $\underline{c} = \{g_1, \ldots, g_k\}$. Then we define

$$\phi_c := \left(\bigvee_{i=1}^{m} \mathbf{f}_i \vee \bigvee_{i=1}^{k} \mathbf{g}_i \right) \rightarrow \mathbf{e} \quad \wedge \quad \mathbf{c} \leftrightarrow \left(\mathbf{e} \wedge \bigwedge_{i=1}^{m} \neg \mathbf{f}_i \right)$$

Moreover, let $\mathsf{Cycles}(\mathcal{U}_N) := \{\, e_1, \ldots, e_n \mid e_1 \nearrow \cdots \nearrow e_n \nearrow e_1 \,\}$ and

$$\phi_C := \bigwedge_{e_1, \ldots, e_n \in \mathsf{Cycles}(\mathcal{U}_N)} \neg(e_1 \wedge \cdots \wedge e_n)$$

We now set $\phi_N := \phi_C \wedge \bigwedge_{c \in B} \phi_c$.

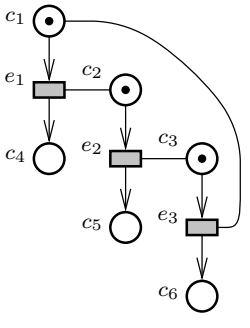

Fig. 4. e_1, e_2, e_3 form an asymmetric-conflict cycle of length 3

The main difference between ϕ_N and the corresponding construction for Petri nets in [3] is the treatment of asymmetric-conflict cycles. In Petri nets, all conflicts are symmetric and between pairs of events e, e' with ${}^\bullet e \cap {}^\bullet e' \neq \emptyset$. In contrast, conflict cycles in c-nets can be of arbitrary length as exemplified by Fig. 4. Nonetheless, they give rise to the same type of constraint in the SAT encoding.

Using ϕ_N, and following the example of [3], one can encode many questions about the set of reachable markings of N in terms of propositional logic, for instance:

- Is there any reachable marking in N that contains both places p and q?
  Let f is the mapping from Definition 1, and suppose that $c_1, \ldots, c_m \in B$ are those conditions with $f(c_i) = p$, for $i = 1, \ldots, m$, and $d_1, \ldots, d_k \in B$ those with $f(d_i) = q$, for $i = 1, \ldots, k$. Let $\phi_p = \mathbf{c}_1 \vee \cdots \vee \mathbf{c}_m$ and $\phi_q = \mathbf{d}_1 \vee \cdots \vee \mathbf{d}_k$. Then, a marking that contains both p and q exists in N iff
  $$\phi_N \wedge \phi_p \wedge \phi_q$$
  is satisfiable.
- Is $\{p, q\}$ a P-invariant of N?
  This means that every reachable marking puts exactly one token into either p or q. Under the same assumptions as above, this is the case iff
  $$\phi_N \rightarrow (\phi_p \oplus \phi_q)$$
  is valid, i.e., its negation is unsatisfiable.
- Does N contain a deadlock?
  Suppose that t is a transition of N with ${}^\bullet t = \{p, q\}$, and otherwise make the same assumptions as above. Let $\phi_t = \phi_p \wedge \phi_q$, then any model of $\phi_N \wedge \phi_t$ corresponds to a marking in which t is enabled. Assuming that we construct corresponding formulae for all other transitions of N, then a deadlock exists iff
  $$\phi_N \wedge \bigwedge_{t \in T} \neg \phi_t$$
  is satisfiable.

References

1. Baldan, P., Bruni, A., Corradini, A., König, B., Schwoon, S.: On the computation of mcMillan"s prefix for contextual nets and graph grammars. In: Ehrig, H., Rensink, A., Rozenberg, G., Schürr, A. (eds.) ICGT 2010. LNCS, vol. 6372, pp. 91–106. Springer, Heidelberg (2010)
2. Baldan, P., Corradini, A., König, B., Schwoon, S.: McMillan's complete prefix for contextual nets. In: Jensen, K., van der Aalst, W.M.P., Billington, J. (eds.) Transactions on Petri Nets and Other Models of Concurrency I. LNCS, vol. 5100, pp. 199–220. Springer, Heidelberg (2008)
3. Esparza, J., Heljanko, K.: Unfoldings - A Partial-Order Approach to Model Checking. EATCS Monographs in Theoretical Computer Science. Springer, Heidelberg (2008)
4. Esparza, J., Römer, S., Vogler, W.: An improvement of McMillan's unfolding algorithm. Formal Methods in System Design 20, 285–310 (2002)
5. Heljanko, K.: Combining Symbolic and Partial-Order Methods for Model-Checking 1-Safe Petri Nets. Ph.D. thesis. Helsinki University of Technology (2002)

6. Janicki, R., Koutny, M.: Invariant semantics of nets with inhibitor arcs. In: Groote, J.F., Baeten, J.C.M. (eds.) CONCUR 1991. LNCS, vol. 527, pp. 317–331. Springer, Heidelberg (1991)
7. McMillan, K.L.: Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits. In: Probst, D.K., von Bochmann, G. (eds.) CAV 1992. LNCS, vol. 663, pp. 164–177. Springer, Heidelberg (1993)
8. Montanari, U., Rossi, F.: Contextual occurrence nets and concurrent constraint programming. In: Ehrig, H., Schneider, H.-J. (eds.) Dagstuhl Seminar 1993. LNCS, vol. 776. Springer, Heidelberg (1994)
9. Ristori, G.: Modelling systems with shared resources via Petri nets. Ph.D. thesis. University of Pisa (1994)
10. Rodríguez, C.: The Cunf tool, `http://www.lsv.ens-cachan.fr/~rodriguez/tools/cunf/`
11. Rodríguez, C., Schwoon, S., Baldan, P.: Efficient contextual unfolding. Tech. rep., LSV, ENS de Cachan (2011)
12. Vogler, W., Semenov, A.L., Yakovlev, A.: Unfolding and finite prefix for nets with read arcs. In: Sangiorgi, D., de Simone, R. (eds.) CONCUR 1998. LNCS, vol. 1466, pp. 501–516. Springer, Heidelberg (1998)
13. Winkowski, J.: Reachability in contextual nets. Fundamenta Informaticae 51(1-2), 235–250 (2002)