Forward Analysis for Petri Nets with Name Creation[*]

Fernando Rosa-Velardo and David de Frutos-Escrig

Dpto. de Sistemas Informáticos y Computación
Universidad Complutense de Madrid
{fernandorosa,defrutos}@sip.ucm.es

Abstract. Pure names are identifiers with no relation between them, except equality and inequality. In previous works we have extended P/T nets with the capability of creating and managing pure names, obtaining ν-APNs and proved that they are strictly well structured (WSTS), so that coverability and boundedness are decidable. Here we use the framework recently developed by Finkel and Goubault-Larrecq for forward analysis for WSTS, in the case of ν-APNs, to compute the cover, that gives a good over approximation of the set of reachable markings. We prove that the least complete domain containing the set of markings is effectively representable. Moreover, we prove that in the completion we can compute least upper bounds of simple loops. Therefore, a forward Karp-Miller procedure that computes the cover is applicable. However, we prove that in general the cover is not computable, so that the procedure is non-terminating in general. As a corollary, we obtain the analogous result for Transfer Data nets and Data Nets. Finally, we show that a slight modification of the forward analysis yields decidability of a weak form of boundedness called width-boundedness.

1 Introduction

Pure names have been extensively studied in the fields of security and mobility, because they can be used to represent different entities widely used in them. For instance, names can represent communicating channels in π-calculus terms, computing boundaries in the Ambient Calculus or ciphering keys in the spi Calculus [13]. In previous works we have extended P/T nets with a primitive to create fresh names, defining ν-APNs. Names are represented as tokens, that are no longer indistinguishable. These tokens can move along the places of the net and be used to restrict the firing of some transitions, imposing for instance that two certain names at the preconditions match.

In [17] we proved that ν-APNs are Well Structured Transition Systems (WSTS). For WSTS it is possible to perform a backward analysis that computes the set $\uparrow Pre^*(\uparrow M)$ [1,8], the set of predecessors of an upward-closed set

[*] Work partially supported by the Spanish projects DESAFIOS10 TIN2009-14599-C03-01, UCM-BSCH GR58/08/910606 and PROMETIDOS S2009/TIC-1465.

J. Lilius and W. Penczek (Eds.): PETRI NETS 2010, LNCS 6128, pp. 185–205, 2010.

$\uparrow M$. An effective representation of that set allows us to decide the coverability problem, by checking whether the initial marking $M_0 \in \uparrow Pre^*(\uparrow M)$. However, the construction of such sets is extremely expensive, with a non primitive recursive complexity [20].

Very recently, Finkel and Goubault-Larrecq have laid the foundation of a theory supporting forward analysis of WSTS [10,11], computing $\downarrow Post^*(\downarrow M_0)$, the so called *cover* of the transition system. The cover provides a good over approximation of the set of reachable states, and its construction is generally more efficient in practice than that of $\uparrow Pre^*(\uparrow M)$. However, it is not always possible to obtain an effective representation of the cover [3]. The paper [10] establishes a theory for the completion of *well quasi orders* (wqos), so that we can always represent downward-closed sets by means of their least upper bounds. There it is proved that the least completion of X (that contains an adequate domain of limits, in the sense of [12]) is the so called ideal completion of X, or equivalently, the sobrification of X [14].

We will see here that the ideal completion of the set of markings can be effectively represented by mapping markings to the domain $\mathcal{MS}(\mathcal{MS}(P))$ of finite multisets of finite multisets of places. For that purpose we introduce the domain of ω-markings (analogous to the classical notion of ω-markings for P/T nets). In an ω-marking, not only some identifiers may appear an unbounded number of times in some places, as happens in classical ω-markings, but also an unbounded number of different identifiers may occur in a marking.

Assuming a complete domain (thus containing an adequate domain of limits), a generic Karp-Miller procedure to compute the cover is presented in [11]. This procedure is correct provided the WSTS is ∞-effective, which intuitively means that we can accelerate simple loops (flat loops, in the sense of [4]). We will see that ν-APNs are ∞-effective when we restrict the non-determinism arising in loops, so that we can apply to them the generic Karp-Miller procedure. Unfortunately, when applied to this kind of systems, the procedure is not guaranteed to terminate. We will see that this is unavoidable, since we can reduce the problem of boundedness for reset nets, which is known to be undecidable [7], to the computation of the cover.

Data nets [16] are Petri nets in which tokens are taken from a linearly ordered and dense domain, and capable of performing whole place operations, such as transfers or resets. Transfer Data Nets is the subclass of Data nets in which no resets are allowed, and Petri Data Nets is the subclass of Data Nets (and of Transfer Data Nets) in which no whole-place operation is allowed. Petri Data nets subsume ν-APNs [16], so that as a corollary, there cannot be an algorithm computing (a finite basis of) the cover of a Petri Data net, and therefore neither for a Transfer Data net, thus answering negatively to a question posed in [11].

But even if there is no algorithm for the computation of the cover, we can use a slight modification of the forward Karp-Miller procedure to decide width-boundedness of ν-APNs [18,5]. A net is width-bounded if only a bounded number of different names appear in each reachable marking. The paper [5] also establishes the decidability of width-boundedness (called m-boundedness there), but we claim that the algorithm presented there does not properly work in all the cases. This is because the algorithm stops whenever unboundedness is detected. However, width-unbounded nets may be bounded or not, so that we need to further explore the reachability graph to decide width-boundedness. For that purpose, we need ways to finitely represent downward-closed sets of reachable markings, our ω-markings. We already knew [18] that width-boundedness is decidable, but we obtain the result here as a simple application of our forward analysis.

The rest of the paper is structured as follows. Section 2 introduces our notations and some basic concepts. In Section 3 we present ν-APNs. In Section 4 we show how ν-APNs fit in the general framework for forward analysis of WSTS in [10,11]. Section 5 contains our main results: the viability of a forward Karp-Miller procedure for ν-APNs, non-computability of the cover and decidability of width-boundedness. Finally, Section 6 presents our conclusions and some directions for further work.

2 Preliminaries

wqos, dcpos. A quasi order $\leq$ is a reflexive and transitive binary relation on a set X. A partial order is an antisymmetric quasi order. A poset is a set endowed with a partial order. We write $a < b$ if $a \leq b$ and $b \not\leq a$. A quasi order is simply said well (wqo) [9], if for every infinite sequence $a_0, a_1, \ldots$ there are i and j with $i < j$ such that $a_i \leq a_j$. Equivalently, an order is a wqo if every sequence has an increasing subsequence.

The downward closure $\downarrow E$ of $E \subseteq X$ is $\{y \in X \mid y \leq x \text{ for some } x \in E\}$. A set is downward closed iff $\downarrow E = E$. A basis of a downward closed set E is a set A such that $\downarrow A = E$. An element $x \in X$ is an upper bound of E if $y \leq x$ for all $y \in E$. We write $lub(E)$ to denote the least upper bound of E, when it exists. An element $x \in E$ is maximal if $x = y$ whenever $x \leq y \in E$; $Max E$ is the set of maximal elements of E. A subset D of X is said to be directed if $lub(\{x, y\})$ exists for all $x, y \in D$. A poset is *directed complete* (dcpo) if every directed subset has a least upper bound. For an arbitrary subset E, $Lub(E) = \{lub(D) \mid D \text{ directed}, D \subseteq E\}$. The set $Lub(E)$ can be thought of as E together with all its limits. For a dcpo X, we write $x \ll y$ whenever $y \leq lub(D)$ implies $x \leq z$ for some $z \in D$, for all directed subset D. X is continuous if for all $x \in X$, $x = lub\{y \in X \mid y \ll x\}$.

WSTS. A labelled transition system is a tuple $N = (X, \rightarrow, Act)$ with a set X of states, Act a set of actions and a transition relation $\rightarrow = \bigcup_{a \in Act} \xrightarrow{a}$, with $\xrightarrow{a} \subseteq X \times X$. We denote by $\xrightarrow{a}^*$ (resp. $\rightarrow^*$) the reflexive and transitive closure of $\xrightarrow{a}$ (resp. $\rightarrow$). $Post_{a,N}(M)$ (or just $Post_a(M)$) is the set $\{M' \mid M \xrightarrow{a} M'\}$ of immediate a-successors of M. $Post^*(M) = \{M' \mid M \rightarrow^* M'\}$ is the set of reachable states. Both $Post_a$ and $Post^*$ are extended pointwise to sets of states. A Well Structured Transition System (WSTS) is a tuple $N = (X, \rightarrow, Act, \leq)$, where $(X, \rightarrow, Act)$ is a labelled transition system, and $(X, \leq)$ is a wqo, satisfying the following monotonicity condition[1]: $M_1 \geq M_2 \xrightarrow{a} M_2'$ implies the existence of M_1' such that $M_1 \xrightarrow{a} M_1' \geq M_2'$. Given a state M, the *cover* of M is the set $\downarrow Post^*(M)$ (or equivalently, $\downarrow Post^*(\downarrow M)$ because of monotonicity), and we will denote it by $Cover_N(M)$ (or just $Cover(M)$ if there is no confusion). Given an initial state M_0, the cover of N is the cover of M_0. N is said to be *effective* if $Post_a(M)$ is finite and computable for all M, and $\leq$ is decidable. A WSTS $(X, \rightarrow, Act, \leq)$ is complete whenever $(X, \leq)$ is a continuous dcpo and for every $a \in Act$, $Post_a(Lub(E)) = Lub(Post_a(E))$ for every set E.

An ideal is a downward closed directed subset. The ideal completion $\overline{X}$ of a wqo X is the set of ideals of X, ordered by inclusion. Given a WSTS $N = (X, \rightarrow, Act, \leq)$, the ideal completion of N is the transition system $\overline{N} = (\overline{X}, \mapsto, Act)$, where $F \overset{a}{\mapsto} F' = \downarrow \{s' \mid s \xrightarrow{a} s',\ s \in F\}$. $(\overline{X}, \subseteq)$ is a continuous dcpo. However, $\overline{N}$ is not a WSTS in general. A wqo is an ω^2-wqo if it does not contain the Rado's structure, and an ω^2-WSTS is a WSTS with an underlying ω^2-wqo [15]. Then, $\overline{N}$ is a WSTS iff N is a ω^2-WSTS [11].

Multisets. Given an arbitrary set A, we will denote by $\mathcal{MS}(A)$ the set of finite multisets of A, that is, the mappings $m : A \rightarrow \mathbb{N}$. When needed, we identify each set with the multiset defined by its characteristic function, and use set notation for multisets when convenient. We denote by $S(m)$ the support of m, that is, the set $\{a \in A \mid m(a) > 0\}$ and by $|m| = \sum\limits_{a \in S(m)} m(a)$ the cardinality of m. Given two multisets $m_1, m_2 \in \mathcal{MS}(A)$ we denote by $m_1 + m_2$ the multiset defined by $(m_1 + m_2)(a) = m_1(a) + m_2(a)$. We will write $m_1 \subseteq m_2$ if $m_1(a) \leq m_2(a)$ for every $a \in A$. Then, we can define $m_2 - m_1$, taking $(m_2 - m_1)(a) = m_2(a) - m_1(a)$. We will denote by $\emptyset \in \mathcal{MS}(A)$ the empty multiset. If $f : A \rightarrow B$ and $m \in \mathcal{MS}(A)$, we define $f(m) \in \mathcal{MS}(B)$ by $f(m)(b) = \sum_{f(a)=b} m(a)$.

Every partial order $\leq$ defined over A induces a partial order $\sqsubseteq$ in the set $\mathcal{MS}(A)$, given by $\{a_1, \ldots, a_n\} \sqsubseteq \{b_1, \ldots, b_m\}$ if there is an injective function $\iota : \{1, \ldots, n\} \rightarrow \{1, \ldots, m\}$ such that $a_i \leq b_{\iota(i)}$ for all i. If we do not demand ι to be injective we obtain the powerdomain order $\leq_\exists^\forall$. We write $\sqsubseteq_\iota$ and $\leq_\iota^\forall$ to stress the use of the mapping ι. It is well known that if $\leq$ is a wqo then so is $\sqsubseteq$.

3 ν-APNs

In this section we present ν-APNs; the reader is referred[2] to [19] for more details. In ν-APNs names can be created, communicated and matched. We can use this mechanism to deal with authentication issues [17], correlation or instance isolation [6]. We formalize name management by replacing ordinary tokens by distinguishable tokens. We fix a set *Id* of names, that can be carried by tokens of any ν-APN. In order to handle these colors, we need matching variables labelling

[1] Different monotonicity notions are considered in [9].

[2] We present here a more general version, that allows weights in arcs and check for inequality. The results in [17,19,18] can be easily transferred to this extended version.

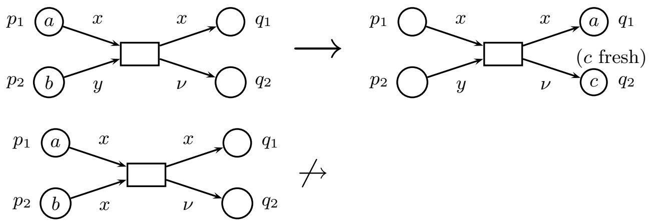

Fig. 1. Two simple ν-APN

the arcs of the nets, taken from a fixed set *Var*. Moreover, we add a primitive capable of creating new names, formalized by means of special variables in a set $\Upsilon \subset \mathit{Var}$, ranged by $\nu, \nu_1, \ldots$, that can only be instantiated to fresh names.

As an example, the net in the top of Fig. 1 is a simple ν-APN with a single transition. When fired, it moves one token from p_1 to q_1 (because of variable x labelling both arcs), removes a token from p_2 (variable y does not appear in any outgoing arc) and a new name is created in q_2 (because of variable ν). Instead, the net in the bottom of Fig. 1 uses the same variable x to label the two arcs incoming its only transition. In that case, the transition must take two tokens carrying the same name from p_1 and p_2, so that the transition is not enabled.

Definition 1. *A ν-APN is a tuple $N = (P, T, F)$, where P and T are finite disjoint sets, $F : (P \times T) \cup (T \times P) \to \mathcal{MS}(\mathit{Var})$ is such that for every $t \in T$, $\Upsilon \cap \mathit{pre}(t) = \emptyset$ and $\mathit{post}(t) \setminus \Upsilon \subseteq \mathit{pre}(t)$, where $\mathit{pre}(t) = \bigcup_{p \in P} S(F(p,t))$ and $\mathit{post}(t) = \bigcup_{p \in P} S(F(t,p))$.*

The set of pairs (x, y) such that $F(x, y) \neq \emptyset$ defines the set of arcs of N. We also take $\mathit{Var}(t) = \mathit{pre}(t) \cup \mathit{post}(t)$, $\mathit{fVar}(t) = \mathit{Var}(t) \cap \Upsilon$ and $\mathit{nfVar}(t) = \mathit{Var}(t) \setminus \mathit{fVar}(t)$. To avoid tedious definitions, along the paper we will consider a fixed ν-APN $N = (P, T, F)$.

Definition 2. *A marking of N is a function $M : P \to \mathcal{MS}(\mathit{Id})$. We denote by $\mathit{Id}(M)$ the set of names in M, that is, $\mathit{Id}(M) = \bigcup_{p \in P} S(M(p))$.*

Like for other classes of higher-order nets, transitions are fired with respect to a mode, that chooses which tokens are taken from the preconditions and which are put in the postconditions. Given a transition t of N, a mode for t is an injection $\sigma : \mathit{Var}(t) \to \mathit{Id}$ that instantiates each variable to a different identifier. Thus, by using the same variable we force the equality of names taken from preconditions, and because modes are injections, we also check the inequality of names by using different variables. We will use $\sigma, \sigma', \sigma_1 \ldots$ to range over modes.

Definition 3. *Let M be a marking, t a transition and σ a mode for t. We say t is enabled with mode σ if for all $p \in P$, $\sigma(F(p,t)) \subseteq M(p)$ and $\sigma(\nu) \notin \mathit{Id}(M)$ for all $\nu \in \mathit{fVar}(t)$. The reached state after the firing of t with mode σ is the marking M', given by $M'(p) = (M(p) - \sigma(F(p,t))) + \sigma(F(t,p))$ for all $p \in P$.*

In the definition of firing we demand that $\sigma(\nu) \notin Id(M)$, for every special variable ν, that is, that every such ν is instantiated to a different fresh name, not in the current marking. Moreover (and unlike in [19]) we demand modes to be injective, which amounts to being able to check for inequality of names (not only for equality, by using the same variable in different arcs). We will write $M \xrightarrow{t} M'$, $M \xrightarrow{t(\sigma)} M'$, $M \rightarrow M'$ and $M \xrightarrow{\tau} M'$ with $\tau = t_1(\sigma_1)\cdots t_n(\sigma_n)$, saying that τ is a transition sequence, with their obvious meanings.

Let us now define the natural order between markings, that induces the coverability problem in ν-APN. We define $M_1 \sqsubseteq_\alpha M_2$ if there is an injection $\iota : Id(M_1) \rightarrow Id(M_2)$ such that $\iota(M_1(p)) \subseteq M_2(p)$, for all $p \in P$. We take $\equiv_\alpha$ as $\sqsubseteq_\alpha \cap {}_\alpha\sqsupseteq$ and identify markings up to $\equiv_\alpha$, that allows renaming of names. The relation $\sqsubseteq_\alpha$ is a wqo [17]. We will sometimes write $M_1 \sqsubseteq_\iota M_2$ to emphasize the use of ι.

4 Forward Analysis for ν-APNs

The state space of a P/T net is the set $\mathbb{N}^k$. However, that set is not complete. For instance, the increasing chain $(n)_{n=1}^{\infty}$ does not have a least upper bound in $\mathbb{N}$. For that purpose, the classical Karp-Miller construction for P/T nets works instead with the domain $(\mathbb{N} \cup \{\omega\})^k$, which is the completion of $\mathbb{N}^k$. In particular, the least upper bound of the previous chain is just ω. In general, a generic Karp-Miller procedure needs to work with the completion of the domain of the WSTS, in case it is not already complete.

In this section we build the completion of the transition system defined by a ν-APN. In [10] it is proved that the ideal completion[3] of a poset is effective (ideals can be finitely represented, and inclusion is decidable) whenever the poset is built up from some basic data type constructions, among which are finite domains, with any order, and multisets of elements in a domain with effective ideal completion. Let us see that we can build our markings using these two constructions.

The behavior of ν-APNs is invariant under $\equiv_\alpha$ [17]. When working modulo $\equiv_\alpha$ we can represent markings as multisets of multisets of places, where each multiset represents the projection of the marking over some identifier. For instance, the marking M given by $M(p) = \{a\}$ and $M(q) = \{a, b\}$ can be equivalently represented by the multiset $\{\{p, q\}, \{q\}\}$ in $\mathcal{MS}(\mathcal{MS}(P))$. In general, for a marking M, its multiset representation is given by $\{M^a \mid a \in Id(M)\}$, where $M^a(p) = M(p)(a)$. We can also denote the previous multiset by the expression $pq + q$, where pq represents the identifier a, which is both in p and in q, and q represents the identifier b, which is only in q. In the following, $\sqsubseteq$ will denote the natural order over $\mathcal{MS}(\mathcal{MS}(P))$ (induced by the equality in P).

Lemma 1. *Let M_1 and M_2 be two markings, and $\overline{M}_1$ and $\overline{M}_2$ their multiset representation. Then we have $M_1 \sqsubseteq_\alpha M_2$ iff $\overline{M}_1 \sqsubseteq \overline{M}_2$.*

[3] Actually, the authors work with the equivalent concept of sobrification.

In particular, $M_1 \equiv_\alpha M_2$ iff their multiset representations coincide. Since we are interested in the abstract treatment of pure names, our set of configurations will be just the set of finite multisets of finite multisets of places[4].

Next we define ω-markings, the analogous concept of the classical ω-markings of P/T nets in the case of ν-APN. We use a terminology inspired by the Simple Regular Expressions of [3]. We denote by $\mathbb{N}_\omega$ the set $\mathbb{N} \cup \{\omega\}$, and extend the natural order and the usual arithmetic to $\mathbb{N}_\omega$. Next we will consider a fixed enumeration of the places of the net, $P = \{p_1, \ldots, p_n\}$.

Definition 4. *A* product *is an expression* $p_1^{i_1} \cdots p_n^{i_n}$ *with* $i_1, \ldots, i_n \in \mathbb{N}_\omega$. *A* sum *is an expression of the form* $E_1 + \ldots + E_m$*, where each* E_i *is a product. An* ω-marking *is an expression* $\mathcal{A} + \infty(\mathcal{B})$*, with* $\mathcal{A}$ *and* $\mathcal{B}$ *sums.*

Intuitively, ω-markings are markings (modulo $\equiv_\alpha$) in which some identifiers may appear an unbounded number of times, and also an unbounded number of different identifiers may appear. Notice that each product corresponds to an ordinary ω-marking of a P/T net. For instance, the ω-marking $pq^\omega + \infty(p^\omega)$ represents the marking in which an identifier appears once in p and infinitely often in q, and infinitely many other different identifiers appear infinitely often in p. Clearly, plain markings are a particular class of ω-markings, those in which $\mathcal{B}$ is the empty expression and $E_i = p_1^{i_1} \cdots p_n^{i_n}$ with $i_1, \ldots, i_n \in \mathbb{N}$ for all E_i in $\mathcal{A}$. Sometimes, for an ω-marking $\mathcal{M} = \mathcal{A} + \infty(\mathcal{B})$ we will refer to $\mathcal{A}$ as the bounded part of $\mathcal{M}$ and to $\mathcal{B}$ as the unbounded part of $\mathcal{M}$.

We denote by $\emptyset$ the empty sum, and we will simply write $\mathcal{A}$ instead of $\mathcal{A}+\infty(\emptyset)$ and $\infty(\mathcal{B})$ instead of $\emptyset+\infty(\mathcal{B})$. We will often omit places p with a null exponent, and expand exponential factors, writing for instance qq instead of p^0q^2 (assuming $P = \{p, q\}$).

We define $|p_1^{i_1} \cdots p_n^{i_n}|_\omega = |\{k \mid i_k = \omega\}|$, and $(p_1^{i_1} \cdots p_n^{i_n})^\omega = p_1^{j_1} \cdots p_n^{j_n}$, where $j_k = 0$ if $i_k = 0$, and $j_k = \omega$ otherwise (e.g., $(ppq^\omega)^\omega = p^\omega q^\omega$). Given two products $E_1 = p_1^{i_1} \cdots p_n^{i_n}$ and $E_2 = p_1^{j_1} \cdots p_n^{j_n}$ we take $E_1 \sqsubseteq E_2 \Leftrightarrow i_k \leq j_k$ for all $k \in \{1, \ldots, n\}$, and we define $E_1 \oplus E_2 = p_1^{i_1+j_1} \cdots p_n^{i_n+j_n}$, and whenever $E_2 \sqsubseteq E_1$, $E_1 \ominus E_2 = p_1^{i_1-j_1} \cdots p_n^{i_n-j_n}$, provided $j_k \neq \omega$ for all $k \in \{1, \ldots, n\}$. Finally, $(\mathcal{A} + \infty(\mathcal{B})) + (\mathcal{A}' + \infty(\mathcal{B}'))$ is the ω-marking $(\mathcal{A} + \mathcal{A}') + \infty(\mathcal{B} + \mathcal{B}')$.

Let us now define the order between ω-markings, that extends the natural one for markings.

Definition 5. *Given two ω-markings* $\mathcal{M} = E_1 + \ldots + E_m + \infty(E_{m+1} + \ldots + E_k)$ *and* $\mathcal{M}' = E'_1 + \ldots + E'_{m'} + \infty(E_{m'+1} + \ldots + E_{k'})$ *we define* $\mathcal{M} \sqsubseteq \mathcal{M}'$ *if there is* $\iota : \{1, \ldots, k\} \to \{1, \ldots, k'\}$ *such that:*

- *If* $\iota(i) = \iota(j)$ *and* $\iota(j) \leq m'$ *then* $i = j$ *(it is partially injective),*
- *If* $i > m$ *then* $\iota(i) > m'$,
- $E_i \sqsubseteq E_{\iota(i)}$ *for all* $i \in \{1, \ldots, k\}$.

[4] Notice that $\mathcal{MS}(P)$ is isomorphic to $\mathbb{N}^{|P|}$, so that alternatively we could have considered $\mathcal{MS}(\mathbb{N}^{|P|})$ instead of $\mathcal{MS}(\mathcal{MS}(P))$.

As for multisets, we use a mapping ι to specify which product of $\mathcal{M}'$ is used to bound each product in $\mathcal{M}$. Products in the bounded part of $\mathcal{M}$ can be mapped to products in the bounded or in the unbounded part of $\mathcal{M}'$, though products in the unbounded part of $\mathcal{M}$ can only be mapped to products that are also in the unbounded part of $\mathcal{M}'$. Intuitively, infinitely many copies of a product can only be bounded by an infinite number of products. Products in the bounded part of $\mathcal{M}'$ can only be used once to bound products in $\mathcal{M}$, while this is not the case for products in the unbounded part. Alternatively, we could have defined $\mathcal{A}+\infty(\mathcal{B}) \sqsubseteq \mathcal{A}'+\infty(\mathcal{B}')$ if we can split $\mathcal{A}$ into $\mathcal{A}_1$ and $\mathcal{A}_2$ so that[5] $\mathcal{A}_1 \sqsubseteq \mathcal{A}'$, $\mathcal{A}_2 \leq_{\exists}^{\forall} \mathcal{B}'$, and $\mathcal{B} \leq_{\exists}^{\forall} \mathcal{B}'$. The products in $\mathcal{A}_1$ are mapped to the bounded part, while the ones in $\mathcal{A}_2$ and in $\mathcal{B}$ are mapped to the unbounded part. Notice that, in this case, we are using the order $\leq_{\exists}^{\forall}$ since, intuitively, we have infinitely many copies of the products in $\mathcal{B}'$, so that we can choose any of them to bound as many sums as needed, so that the mapping needs not be injective. For instance, it holds $p+q+qq+\infty(q) \sqsubseteq pq+\infty(qq)$ because $p \sqsubseteq pq$, $q+qq \leq_{\exists}^{\forall} qq$ and $q \leq_{\exists}^{\forall} qq$.

We take $\equiv$ as $\sqsubseteq \cap \sqsupseteq$ and identify ω-markings up to $\equiv$. We take as ω-Markings the set of ω-markings identified up to $\equiv$. As for plain markings, we will also use the notation $\sqsubseteq_{\iota}$. When there is no confusion, we will write $\iota(E_i)$ instead of $E_{\iota(i)}$. For instance, $p+qq+\infty(q) \sqsubseteq_{\iota} p+\infty(qq)$ with $\iota(p)=p$, $\iota(qq)=qq$ and $\iota(q)=qq$. The following equivalences will be used along the rest of the paper.

Lemma 2. *If $E_1 \sqsubseteq E_2$ then $E_1+\infty(E_2) \equiv \infty(E_2)$ and $\infty(E_1+E_2) \equiv \infty(E_2)$.*

Thus, for instance we have that $p+q+\infty(pq) \equiv \infty(pq) \equiv \infty(p+q+pq)$. Though ω-markings can be intuitively seen as markings in which some identifiers appear infinitely often, and in which an infinite number of different identifiers can appear, technically they represent sets of markings, those bounded by them as expressed by their denotations.

Definition 6. *The denotation of a product $E = p_1^{k_1}\cdots p_n^{k_n}$ is the set of multisets of places $[\![E]\!] = \{A \in \mathcal{MS}(P) \mid A(p_i) \leq k_i \text{ for all } i=1,\dots,n\}$. The denotation of a sum $\mathcal{A} = \sum_{i=1}^{m} E_i$ is given by $[\![\mathcal{A}]\!] = \{\{A_i \mid A_i \in [\![E_i]\!], i \in I\} \mid I \subseteq \{1,\dots,m\}\}$. We define the denotation of an ω-marking $\mathcal{M} = \mathcal{A}+\infty(\mathcal{B})$ as the set of markings $[\![\mathcal{M}]\!] = \{M + \sum_{i=1}^{k} M_i \mid k \geq 0, M \in [\![\mathcal{A}]\!], M_i \in [\![\mathcal{B}]\!]\}$.*

Take the ω-marking $pq+\infty(qq)$. The denotation of pq is the set $\{\emptyset, p, q, pq\}$, and $[\![qq]\!] = \{\emptyset, q, qq\}$. Thus, $[\![pq+\infty(qq)]\!]$ is the set of markings of the form $M + \underbrace{q+\ldots+q}_{n_1} + \underbrace{qq+\ldots+qq}_{n_2}$ with $n_1, n_2 \geq 0$ and $M \in [\![pq]\!]$. Notice that $[\![\mathcal{M}]\!]$ is a downward closed and directed set, that is, an ideal.

Proposition 1. *The ideal completion of $(\mathcal{MS}(\mathcal{MS}(P)), \sqsubseteq)$ can be effectively represented as (ω-Markings,$\sqsubseteq$).*

[5] Abusing notation, we are considering sums to be multisets of products.

In particular, given two ω-markings $\mathcal{M}_1$ and $\mathcal{M}_2$ it holds that $\mathcal{M}_1 \sqsubseteq \mathcal{M}_2 \Leftrightarrow [\![\mathcal{M}_1]\!] \subseteq [\![\mathcal{M}_2]\!]$, so that ($\omega$-Markings,$\sqsubseteq$) is a continuous dcpo.

Now we need to lift the transition relation to the completed domain of ω-markings. More precisely, for each ω-marking $\mathcal{M}$ we need to effectively compute the set $\downarrow Post([\![\mathcal{M}]\!])$. First, let us introduce some notations: Given a transition t and a variable x, we will denote by $pre_t(x)$ the product $p_1^{i_1} \cdots p_n^{i_n}$, with $i_k = F(p_k,t)(x)$, and $post_t(x) = p_1^{i_1} \cdots p_n^{i_n}$, with $i_k = F(t,p_k)(x)$. In particular, the products $post_t(\nu)$, that correspond to the special variables $\nu \in \Upsilon$, are the "fresh" products created by the transition t. For instance, the net in the bottom of Fig. 1 satisfies $pre_t(x) = p_1 p_2$, $post_t(x) = q_1$ and $post_t(\nu) = q_2$.

Definition 7. *Let $\mathcal{M} = E_1 + \cdots + E_m + \infty(E_{m+1} + \cdots + E_k)$ be an ω-marking, and t a transition. An ω-mode for t is any mapping $\sigma : nfVar(t) \to \mathbb{N}$ such that:*

- *If $\sigma(x) = \sigma(y)$ and $\sigma(y) \leq m$ then $x = y$, and*
- *$pre_t(x) \sqsubseteq E_{\sigma(x)}$ for all $x \in Var(t)$.*

Then we write $\mathcal{M} \xrightarrow{t(\sigma)} \mathcal{A} + \infty(\mathcal{B})$, where $\mathcal{B} = E_{m+1} + \cdots + E_k$ and

$$\mathcal{A} = \sum_{x \in nfVar(t)} ((E_{\sigma(x)} \ominus pre_t(x)) \oplus post_t(x)) + \sum_{i \notin \sigma(Var(t))} E_i + \sum_{\nu \in fVar(t)} post_t(\nu)$$

We define $\overline{Post_t}(\mathcal{M}) = \{\mathcal{M}' \mid \mathcal{M} \xrightarrow{t(\sigma)} \mathcal{M}' \text{ for some } \sigma\}$, and extend it pointwise to sets of ω-markings.

We will write $\sigma(x) = E$ to denote that the product E is used by variable x in mode σ. For all $x \in Var(t)$, we will write $\nabla_t(x) = (\sigma(x) \ominus pre_t(x)) \oplus post_t(x)$. Notice that for $\nu \in fVar(t)$ then $\nabla_t(\nu)$ is simply $post_t(\nu)$. We will also write $\mathcal{M} \xrightarrow{t} \mathcal{M}'$, $\mathcal{M} \to \mathcal{M}'$, $\mathcal{M} \xrightarrow{\tau} \mathcal{M}'$ and $\mathcal{M} \to^* \mathcal{M}'$ as with plain markings, with their obvious meanings. Moreover, if the product E in $\mathcal{M}$ evolves to E' in $\mathcal{M}'$ we will also write $E \xrightarrow{t(\sigma)} E'$ or $E \to E'$. Notice that whenever $\mathcal{M} \to \mathcal{M}'$ the unbounded part of $\mathcal{M}$ and $\mathcal{M}'$ coincide. However, new products may appear in the bounded part of $\mathcal{M}'$, like those in the unbounded part of $\mathcal{M}$ involved in the firing of the transition. For instance, the net in Fig. 8 can fire $p + \infty(q) \xrightarrow{t_2(\sigma)} p + qq + \infty(q)$ with $\sigma(x) = q$. Intuitively, one of the infinitely many names in q has been chosen, and put twice in q by the transition.

Let us see that we can use $\overline{Post_t}(\mathcal{M})$ to compute $\downarrow Post_t([\![\mathcal{M}]\!])$. For that purpose, we need the following lemma. From now on, we will denote just by $[\![\overline{Post_t}(\mathcal{M})]\!]$ the set $\bigcup_{\mathcal{M}' \in \overline{Post_t}(\mathcal{M})} [\![\mathcal{M}']\!]$.

Lemma 3. *The following conditions hold:*

- *If $M \in [\![\mathcal{M}]\!]$ and $M \xrightarrow{t} M'$ then we have $M' \in [\![\overline{Post_t}(\mathcal{M})]\!]$.*
- *If $M \in [\![\overline{Post_t}(\mathcal{M})]\!]$ then there are $M' \in [\![\mathcal{M}]\!]$ and $M'' \in [\![\overline{Post_t}(\mathcal{M})]\!]$ such that $M \sqsubseteq M''$ and $M' \xrightarrow{t} M''$.*

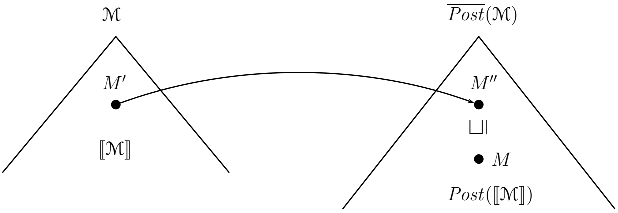

Fig. 2. Computation of $Post([\![\mathcal{M}]\!])$

The first part of the previous lemma states that $Post_t([\![\mathcal{M}]\!]) \subseteq [\![\overline{Post}_t(\mathcal{M})]\!]$. For a better insight of the second part, see Fig. 2. Both allow us to prove the following result.

Proposition 2. $\downarrow Post_t([\![\mathcal{M}]\!]) = [\![\overline{Post}_t(\mathcal{M})]\!]$

Corollary 1. *The completion $\overline{N}$ of a ν-APN N is an effective complete WSTS.*

For a complete WSTS, the *clover* [11] of a state M is defined by $Clover(M) = Max\ Lub(Cover(M))$. The clover of a state is finite because our order is well. It holds that $\downarrow Clover(M) = Lub(Cover(M))$, so that the clover is a finite basis of the cover (together with all the limits). Moreover, if $\overline{N}$ is the completion of $N = (X, \rightarrow, \leq)$ then $Cover_N(M) = Cover_{\overline{N}}(M) \cap X = \downarrow Clover_{\overline{N}}(M) \cap X$, so that the clover of the completion is a basis of the cover (once we remove the limits by intersecting with X).

Now let us see that we can apply a forward Karp-Miller algorithm to compute the clover of N (although, as we will see, it will not terminate in general). For that purpose, we will need to compute the least upper bounds of all the ω-markings produced in a loop, that is, we need to accelerate loops.

5 Accelerations

In the previous section we have mostly seen how ν-APNs fit in the general framework of [10,11]. In the classic construction of the Karp-Miller tree for P/T nets, every time a transition sequence τ such that $M \xrightarrow{\tau} M'$ with $M(p) \leq M'(p)$ for all p and $M(q) < M'(q)$ for some q, we know that the transition sequence τ can be repeated arbitrarily often, so that the number of tokens in q can be considered to be unbounded. In other words, we can replace M' by the least upper bound of the markings obtained by repeating τ an arbitrary number of times.

In order to translate the Karp-Miller procedure to ν-APNs, we need to prove that the completion of a ν-APN is ∞-effective, meaning that we can compute the least upper bound of the markings obtained by repeating a transition sequence, that is, that we can accelerate loops. In the previous section we have shown how we can effectively represent the completed domains, so that the limit of an increasing chain (and more generally, of a directed set) always exists. However,

$$\begin{tikzcd}
Id(\mathcal{M}_1) & Id(\mathcal{M}_2) & Id(\mathcal{M}_3) & Id(\mathcal{M}_4) \\
a \arrow[r, "\iota_1"] & b \arrow[r, "\iota_2"] & c \arrow[r, "\iota_3"] & d \\
b \arrow[r] & c \arrow[r] & d \arrow[r] & e
\end{tikzcd}
\qquad
\begin{tikzcd}
Id(\mathcal{M}_1) & Id(\mathcal{M}_2) & Id(\mathcal{M}_3) & Id(\mathcal{M}_4) \\
a \arrow[r, "\iota_1"] & b \arrow[r, "\iota_2"] & a \arrow[r, "\iota_3"] & b \\
b \arrow[r] & a \arrow[r] & b \arrow[r] & a \\
c \arrow[r] & d \arrow[r] & e \arrow[r] & f
\end{tikzcd}$$

Fig. 3. Example of construction of the sequences $(\iota_i)_{i=1}^{\infty}$

the *double infiniteness* in ω-markings makes the task of computing those limits a non trivial one. We now specify what will it mean in our setting to repeat a transition sequence.

We will discuss the case in which τ is a single transition t, because the general case would only obscure the presentation. Later we will see how the general case can also be considered. Let us suppose that $\mathcal{M}_1 \overset{t(\sigma_1)}{\rightarrow} \mathcal{M}_2$ and $\mathcal{M}_1 \sqsubseteq_{\iota_1} \mathcal{M}_2$. Intuitively, because of monotonicity we can repeat the firing of t in $\mathcal{M}_2$. However, the occurrence of a token a in p is bounded by the occurrence of $\iota_1(a)$ in p. Therefore, if t used a token a because $\sigma_1(x) = a$ for some variable x, then now t must use $\iota(a)$ instead, thus taking $\sigma_2(x) = \iota(a)$. We define the sequences $(\sigma_i)_{i=1}^{\infty}$, $(\mathcal{M}_i)_{i=1}^{\infty}$ and $(\iota_i)_{i=1}^{\infty}$ of ω-modes, ω-markings and mappings, respectively, as follows:

- $\sigma_{i+1}(x) = \iota_i(\sigma_i(x))$, for $i \geq 1$,
- $\mathcal{M}_i \overset{t(\sigma_i)}{\rightarrow} \mathcal{M}_{i+1}$ for $i \geq 1$, and
- $\iota_{i+1}(E) = \begin{cases} E' \text{ if } F' \overset{t(\sigma_i)}{\rightarrow} E \text{ and } \iota_i(F') \overset{t(\sigma_{i+1})}{\rightarrow} E' \text{ for } F' \text{ in } \mathcal{M}_i \\ \quad E \text{ and } E' \text{ in the bounded part,} \\ E \text{ otherwise} \end{cases}$ for $i \geq 1$.

σ_{i+1} is defined following the previous intuitions: if a variable x is first instantiated by a product E, in the next step it is instantiated by $\iota_i(E)$. $\mathcal{M}_{i+1}$ is simply obtained by letting $\mathcal{M}_i$ evolve with mode σ_i. The definition of the mappings ι_i require further explanations. The mappings ι_i map products to products, but perhaps their definition is better understood by considering not the products themselves, but the identifier that each product represents. Consider the left handside of the diagram in Fig. 3, where a is mapped to b by ι_1, and b is mapped to a fresh identifier c. The definition of ι_2 above simply states that now (the product representing) b is mapped to (the product representing) c, because b was mapped to c by ι_1. Accordingly, since ι_1 mapped b to a fresh identifier (represented by a product $E = post_t(\nu)$ for some $\nu \in \Upsilon$), ι_2 must map c to another fresh identifier (which is represented by the same product $E = post_t(\nu)$). Finally, if E is in the unbounded part of $\mathcal{M}_i$ then it is also in the unbounded part of $\mathcal{M}_{i+1}$, and $\iota_{i+1}(E) = E$.

We will denote by $t(\sigma_1)_{\iota}^{k}$ the sequence $t(\sigma_1) \cdots t(\sigma_k)$. In general, for a transition sequence τ we can define as above the sequences of ω-modes, ω-markings and mappings. This is because we can always simulate the effect of the firing of a transition sequence using some given modes with the firing of a single transition.

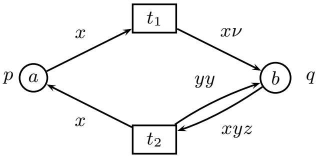

Fig. 4. From transition sequences to transitions

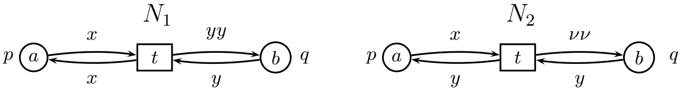

Fig. 5. w-accelerations and d-accelerations

Lemma 4. *Let τ be a transition sequence of a ν-APN $N = (P, T, F)$. Then there is a ν-APN $N' = (P, \{\bar{t}\}, F')$ such that $\mathcal{M}_1 \xrightarrow{\tau} \mathcal{M}_2$ in N if and only if $\mathcal{M}_1 \xrightarrow{\bar{t}(\sigma)} \mathcal{M}_2$ in N' for some mode σ of $\bar{t}$.*

We will call τ-*contraction* of N to the net N' given by the previous result. We will also write τ_ι^k to denote the transition sequence $\bar{t}(\sigma)_\iota^k$, where $\bar{t}(\sigma)$ is the only transition of its τ-contraction. Consider for instance the net in Fig. 4 and the transition sequence $\tau = t_1(\sigma_1)t_2(\sigma_2)$, where $\sigma_1(x) = a$, $\sigma_1(\nu) = c$, $\sigma_2(x) = b$, $\sigma_2(y) = c$ and $\sigma_2(z) = a$. The τ-contraction of that net is the net N_2 depicted in Fig. 5. Notice that the modes σ_1 and σ_2 are such that $\sigma_1(x) = \sigma_2(z)$. Accordingly, since t_1 puts once $\sigma_1(x)$ in q, and t_2 removes $\sigma_2(z)$ from q, in N_2 the token a is neither put nor removed from q.

We are now ready to define in our setting what it means to accelerate a simple loop. The sequence $(\mathcal{M}_i)_{i=1}^{\infty}$ is an increasing sequence, so that the following definition makes sense.

Definition 8. *Let $\overline{N}$ be the completion of a ν-APN N. We say $\overline{N}$ is ∞-effective if it is effective and whenever $\mathcal{M}_1 \xrightarrow{\tau} \mathcal{M}_2$ with $\mathcal{M}_1 \sqsubseteq_\iota \mathcal{M}_2$ we can compute*

$$acc_\iota(\mathcal{M}_1 \xrightarrow{\tau} \mathcal{M}_2) = lub\{\mathcal{M} \mid \mathcal{M}_1 \xrightarrow{\tau_\iota^n} \mathcal{M},\ n > 0\}$$

Let us see that we can compute that least upper bound. In the first place, we can compute the τ-contraction of the net, and work with it instead. Therefore, we can always assume that we want to accelerate a single transition. Let us consider the nets N_1 and N_2 in Fig. 5. Notice that both nets can fire the run $p+q \xrightarrow{t} p+qq$, and $p+q \sqsubseteq_\iota p+qq$ with $\iota(p) = p$ and $\iota(q) = qq$. However, the result of an acceleration in both cases is very different: for N_1, every marking of the form $p + q^n$ is reachable; for N_2, every marking $p + qq + q + \cdots + q$ is reachable. Intuitively, the difference between both situations is that in N_1 each product is mapped to itself (the product p evolves to $\iota(p) = p$ and the product q evolves to $\iota(q) = qq$). However, that is not the case for N_2, where the product q evolves to $\iota(p) = p$. If we consider not products, but the identifiers they represent, then the difference becomes clearer. In N_1 both a and b are mapped to themselves by ι, while in N_2, a is mapped to b, and b is mapped to a fresh identifier. We formalize the behavior of N_1 in the following definition.

Definition 9. *We say $\mathcal{M}_1 \xrightarrow{t(\sigma)} \mathcal{M}_2$ is properly increasing if $\mathcal{M}_1 \sqsubseteq_\iota \mathcal{M}_2$ and for all products E_2 in $\mathcal{M}_2$ there are no different products E_1 and E_1' in the bounded part of $\mathcal{M}_1$ such that $E_1 \xrightarrow{t(\sigma)} E_2$ and $\iota(E_1') = E_2$.*

The firing $p + q \xrightarrow{t} p + qq$ is properly increasing in N_1, but not in N_2, because there is a product p in $p + qq$, and two different products in $p + q$, namely p and q, such that p is mapped to p by ι_1 and q evolves to p. However, every increasing firing can be unrolled into a properly increasing one. Indeed, consider again the diagrams in Fig. 3. In both parts of the diagrams, there is a natural k so that each identifier is mapped in k steps either to itself, or to a fresh identifier. In the left handside, after two steps, both a and b are mapped to fresh identifiers. In the right handside, after three steps, both a and b are mapped to themselves, but c is mapped to a fresh identifier. This happens in general, as we will see in the next lemma.

Lemma 5. *If $\mathcal{M} \xrightarrow{t(\sigma)} \mathcal{M}'$ and $\mathcal{M} \sqsubseteq_\iota \mathcal{M}'$ then there is $k > 0$ such that the firing of the $\bar{t}(\sigma)^k_\iota$-contraction of N is properly increasing.*

We call order of ι, that we will denote as $o(\iota)$, to the natural k given by the previous result, which can be effectively computed. Moreover, we will write $\bar{t}(\sigma)$ instead of $t(\sigma)^{o(\iota)}_\iota$, when it is clear from the context. Clearly, $acc_\iota(\mathcal{M}_1 \xrightarrow{t(\sigma)} \mathcal{M}_2) = acc_\iota(\mathcal{M}_1 \xrightarrow{t(\sigma)^k_\iota} \mathcal{M})$ for any $k > 0$ so that, in particular, we can take $k = o(\iota)$. Moreover, by Lemma 4 we can work with the $\bar{t}(\sigma)$-contraction of the net instead.

As an example, consider the nets in Fig. 5. In N_1 after one step each identifier can be mapped to itself, that is, $\mathcal{M}_1 \xrightarrow{t(\sigma)} \mathcal{M}_2$ with $\mathcal{M}_1 \sqsubseteq_\iota \mathcal{M}_2$, $\iota(a) = a$ and $\iota(b) = b$, so that $o(\iota) = 1$. In N_2 we find the situation in the left of Fig. 3, so that $o(\iota) = 2$. Thus, we need to consider the transition sequence $\tau = t(\sigma)t(\sigma')$, with $\sigma'(x) = b$ and $\sigma'(y) = \sigma(\nu)$. In turn, in order to compute the acceleration we can consider its τ-contraction, depicted in Fig. 6.

d-acceleration. Using properly increasing sequences has the advantage that whenever a product E_x (with $\sigma(x) = E_x$) evolves to some E_x' in the range of ι, then necessarily $E_x \sqsubseteq E_x'$. Then, by repeating the firing of t we will obtain products of the form $E_x \oplus \Delta_t(x)^k$, for some *increment* $\Delta_t(x)$, with least upper bound $E_x \oplus \Delta_t(x)^\omega$. This is the situation for N_1 in Fig. 5 and $p + q \xrightarrow{t} p + qq$, that is properly increasing. Using the previous notations, $E_x = p$ and $E_y = q$, so that $\Delta_t(x) = \emptyset$ and $\Delta_t(y) = q$. Therefore, $acc_\iota(p + q \xrightarrow{t(\sigma)} p + qq) = p + q^\omega$.

w-acceleration. However, in N_2 we cannot apply the previous acceleration. In this case the $\bar{t}(\sigma)$-contraction of N_2 is given by the net in Fig. 6. In it, every

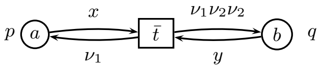

Fig. 6. Contraction of the net N_2 in Fig. 5

product is mapped to a fresh one, and every marking of the form $pq+qq+q+\ldots+q$ is reachable. If we take $\Delta^\iota_t(x) = post_t(\nu_1) \ominus pre_t(x)$ and $\Delta^\iota_t(y) = post_t(\nu_2) \ominus pre_t(y)$ then $acc_\iota(p + q \xrightarrow{t(\sigma)} p + qq) = pq + qq + \infty(q) \equiv post_t(\nu_1) + post_t(\nu_2) + \infty(\Delta^\iota_t(x) + \Delta^\iota_t(y))$.

A simpler case in which a w-acceleration can be applied appears in the net in Fig. 8. The first firing that takes place is $p \xrightarrow{t_1} p+q$, so that $p \sqsubseteq_\iota p+q$ with $\iota(p) = p$. Notice that there is a fresh product, namely q, not in the range of ι, so that any marking of the form $p + q + \ldots + q$ is reachable, and $acc_\iota(p \xrightarrow{t_1} p + q) = p + \infty(q)$.

Following the previous intuitions, if $\mathcal{M}_1 \xrightarrow{t(\sigma)} \mathcal{M}_2$ is properly increasing we partition $nfVar(t)$ as follows:

$$\begin{aligned}
V_{un} &= \{x \in nfVar(t) \mid \sigma(x) \text{ in the unbounded part}\},\\
V_d &= \{x \in nfVar(t) \mid \sigma(x) \xrightarrow{t(\sigma)} \iota(\sigma(x))\},\\
V^\nu_w &= \{x \in nfVar(t) \mid \iota(x) = post_t(\nu_x) \text{ for } \nu_x \in fVar(t)\},\\
V^{un}_w &= \{x \in nfVar(t) \mid \iota(\sigma(x)) = \nabla_t(y_x) \text{ for some } y_x \in V_{un}\}.
\end{aligned}$$

Moreover, there are two injections: $h_\nu : V^\nu_w \to fVar(t)$ and $h_{un} : V^{un}_w \to V_{un}$ given by $h_\nu(x) = \nu_x$ and $h_{un}(x) = y_x$. Let us write $V^\nu_r(t) = fVar(t) \setminus h_\nu(V^\nu_w)$, $V^{un}_r = V_{un} \setminus h_{un}(V^{un}_w)$, $V_w = V^\nu_w \cup V^{un}_w$, $V_b = V_d \cup V_{un}$ and $V_r = V^\nu_r \cup V^{un}_r$.

For all $x \in V_{un}$, $\sigma(x)$ is a product in the unbounded part. For all $x \in V_d$, the products $\sigma(x)$ are mapped to themselves by ι, so that ∇_x will be used instead in the following firing of t. They will be responsible for d-accelerations. Products $\sigma(x)$ with $x \in V^\nu_w$ are those mapped by ι to fresh products. Therefore, $post_t(\nu_x)$ will be used instead in the next firing, so that it will leave some garbage that will cause a w-acceleration. Other products of the form $post_t(\nu)$ will not be used later, those with $\nu \in V^\nu_r$, so that they will also contribute to the w-acceleration.

Variables in V^{un}_w and V^{un}_r have an effect analogous to those in V^ν_w and V^ν_r. Products $\sigma(x)$ with $x \in V^{un}_w$ are mapped by ι to a product ∇_{y_x} that has evolved from a product in the unbounded part. As before, ∇_{y_x} will be used instead in the next firing, leaving again some garbage. Moreover, some products ∇_y that come from a product in the unbounded part (those with $y \in V^{un}_r$) will also remain and contribute to the w-acceleration.

Definition 10. *Let $\mathcal{M}_1 \xrightarrow{t(\sigma)} \mathcal{M}_2$ be a properly increasing sequence. We define the following products:*

- *For all $x \in V_d$, $\Delta_t(x)$ is any product such that $\sigma(x) \xrightarrow{t(\sigma)} \sigma(x) \oplus \Delta_t(x)$,*
- *For all $x \in V^\nu_w$, $\Delta^\iota_t(x) = post_t(h_\nu(x)) \ominus pre_t(x)$,*
- *For all $x \in V^{un}_w$, $\Delta^\iota_t(x) = \nabla_t(h_{un}(x))) \ominus \sigma(x)$.*

```
Procedure Clover($M_0$)
$\Theta \leftarrow \{M_0\}$
while $\overline{Post}(\Theta) \not\sqsubseteq \Theta$ do
    Choose fairly $\mathcal{M} \in \Theta$, $\tau$ and $\iota$
    such that $\mathcal{M} \xrightarrow{\tau} \mathcal{M}'$
    if $\mathcal{M} \not\sqsubseteq_\iota \mathcal{M}'$ then
        $\Theta \leftarrow \Theta \cup \{\mathcal{M}'\}$
    else
        $\Theta \leftarrow \Theta \cup \{acc_\iota(\mathcal{M} \xrightarrow{\tau} \mathcal{M}')\}$
return $Max\ \Theta$
```

```
Procedure width-Clover($M_0$)
$\Theta \leftarrow \{M_0\}$, bounded ←true
while $\overline{Post}(\Theta) \not\sqsubseteq \Theta$ and bounded do
    Choose fairly $\mathcal{M} \in \Theta$, $\tau$ and $\iota$ such that $\mathcal{M} \xrightarrow{\tau} \mathcal{M}'$
    if $\mathcal{M} \not\sqsubseteq_\iota \mathcal{M}'$ then
        $\Theta \leftarrow \Theta \cup \{\mathcal{M}'\}$
    else
        $\mathcal{M}' \leftarrow acc_\iota(\mathcal{M} \xrightarrow{\tau} \mathcal{M}')$
        if x-bounded($\mathcal{M}'$) then
            $\Theta \leftarrow \Theta \cup \{\mathcal{M}'\}$
        else
            bounded ←false
return (bounded,$Max\ \Theta$)
```

Fig. 7. Karp-Miller procedure (left) and algorithm deciding width-boundedness (right)

Proposition 3. *If* $\mathcal{M}_1 \xrightarrow{t(\sigma)} \mathcal{M}_2$ *is properly increasing with* $\mathcal{M}_1 \sqsubseteq_\iota \mathcal{M}_2$ *then there is* $\mathcal{M}$ *such that* $\mathcal{M}_1 \equiv \sum_{x\in V_b} \sigma(x) + \infty(\sum_{x\in V_{un}} \sigma(x)) + \mathcal{M}$*, and* $acc_\iota(\mathcal{M}_1 \xrightarrow{t(\sigma)} \mathcal{M}_2)$ *is*

$$\sum_{x\in V_d}(\sigma(x)\oplus\Delta_t(x)^\omega) + \sum_{x\in V_w^\nu}(post_t(h_\nu(x))) + \sum_{x\in V_w^{un}}(\sigma(x)\oplus\Delta_t^\iota(x)) + \sum_{x\in V_w}\nabla_t(x) +$$

$$\infty(\sum_{x\in V_w^\nu}(post_t(x)\oplus\Delta_t^\iota(x))) + \sum_{x\in V_w^{un}}(\nabla_t(x)\oplus\Delta_t^\iota(x)) + \sum_{x\in V_r}\nabla_t(x) + \sum_{x\in V_{un}}\sigma(x)) + \mathcal{M}$$

Morevoer, the computation of the acceleration does not depend on the increment $\Delta_t(x)$ *chosen.*

Corollary 2. *The completion of a* ν*-APN is an* ∞*-effective (complete) WSTS.*

Because it is ∞-effective, it makes sense to apply the Karp-Miller procedure `Clover`(M_0) in the left of Fig. 7. Fairness in the choosing of the tuples $(\mathcal{M}, \tau, \iota)$ ensures that in every infinite run, every such tuple will be eventually chosen at a later stage. We know that the cover is effectively representable, so that there is a finite set of ω-markings Θ such that $\downarrow Post^*(\downarrow M_0) = \bigcup_{\mathcal{M}\in\Theta} [\![\mathcal{M}]\!]$.

Example 1. Let us see with detail how the algorithm behaves for the ν-APN N_2 in Fig. 5. The initial ω-marking is $\mathcal{M}_0 = p + q$, that is, $\Theta = \{\mathcal{M}_0\}$. The only possible mode that enables t is given by $\sigma_1(x) = p$ and $\sigma(y) = q$, which produces the ω-marking $p + qq$. Notice that:

- $p + q \sqsubseteq_{\iota_1} p + qq$, with $\iota_1(p) = p$ and $\iota_1(q) = qq$,
- the product p in $\mathcal{M}_0$ disappears,
- the product q in $\mathcal{M}_0$ evolves to p,
- the product qq in $p + qq$ is fresh.

Therefore, the firing is not properly increasing, because there is a product p in $p + qq$, and two different products in $\mathcal{M}_0$, namely p and q, such that p is mapped

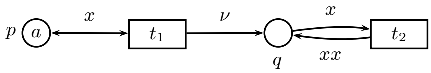

Fig. 8. dw-accelerations

to p by ι_1 and q evolves to p. Actually, we are exactly in the situation of the diagram in the left of Fig. 3. Therefore, we need to unroll the transition sequence $t(\sigma_1)$, with contraction depicted in Fig. 6. There, the firing $p + q \to pq + qq$ can take place, which is properly increasing. Moreover, $V_d = V_{un} = V_w^{un} = \emptyset$ and $V_w^\nu = \{x, y\}$ with $h_\nu(x) = \nu_1$ and $h_\nu(y) = \nu_2$.

- $\Delta_t(x) = post_t(\nu_1) \ominus pre_t(x) = q$,
- $\Delta_t(y) = post_t(\nu_2) \ominus pre_t(y) = q$,
- $\nabla_t(x) = (\sigma(x) \ominus pre_t(x)) \oplus post_t(x) = \emptyset$,
- $\nabla_t(y) = (\sigma(y) \ominus pre_t(y)) \oplus post_t(y) = \emptyset$.

Then, according to Prop. 3, the acceleration is $pq + qq + \infty(q + q) \equiv pq + qq + \infty(q) = \mathcal{M}_1$, so that $\Theta = \{\mathcal{M}_0, \mathcal{M}_1\}$. Starting from $\mathcal{M}_1$ we could fire $t(\sigma)$ with $\sigma(x) = pq$ and $\sigma(y) = qq$, that produces again the ω-marking $\mathcal{M}_1$. We can also fire t from $\mathcal{M}_1$ with a different mode σ, with $\sigma(x) = pq$ and $\sigma(y) = q$, which yields the ω-marking $\mathcal{M}_2 = p + qq + qq + \infty(q)$. Since $\mathcal{M}_1 \not\sqsubseteq \mathcal{M}_2$ no acceleration is performed, and $\Theta = \{\mathcal{M}_0, \mathcal{M}_1, \mathcal{M}_2\}$.

Let us now see what happens if we fire the transition starting from $\mathcal{M}_2$. We could fire it using a mode such that $\sigma(x) = p$ and $\sigma(y) = qq$. The corresponding firing is increasing, but not properly increasing. As happened before, the order of the mapping ι is 2, and the contraction of the unrolling is given again by Fig. 6. The acceleration is analogous to the one obtained from $\mathcal{M}_0$, and produces again the ω-marking $\mathcal{M}_1$.

The other way in which t can be fired from $\mathcal{M}_2$ is more interesting, namely in a mode σ with $\sigma(x) = p$ and $\sigma(y) = q$. Notice that y is instantiated to a product in the unbounded part of $\mathcal{M}_2$. Using that mode, the firing $p + qq + \infty(q) \to p + qq + qq + \infty(q)$ can happen. Moreover, that firing is properly increasing. Indeed, $\iota(p) = p$, $\iota(qq) = qq$ (for both occurrences of qq) and $\iota(q) = q$ and, although the product $\iota(p) = p$ is the result of the evolution of a product different from p, namely q, that product is in the unbounded part. Now we have $V_d = V_w^\nu = V_r^{un} = \emptyset$, $V_{un} = \{y\}$, $V_w^{un} = \{x\}$ and $V_r^\nu = \{\nu\}$, with $h_{un}(x) = y$. Moreover, $\Delta_t^\iota(x) = \nabla_t(x) = \emptyset$, so that the acceleration is $(p \oplus \emptyset) + \emptyset + \infty(\emptyset \oplus \emptyset + qq + q) + qq + qq \equiv p + \infty(qq) = \mathcal{M}_3$.

Similarly, from $\mathcal{M}_3$ we can obtain the ω-marking $\mathcal{M}_4 = pq + \infty(qq)$, and obtain $\Theta = \{\mathcal{M}_0, \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4\}$. From Θ, no other ω-marking can be obtained. Thus, the algorithm returns the maximal ω-marking $\mathcal{M}_4$ (because $\mathcal{M}_i \sqsubseteq \mathcal{M}_4$ for all i), so that the cover is the set of markings $[\![pq + \infty(qq)]\!]$. In particular, every reachable marking M has one identifier a and a (possibly empty) set of identifiers $\{b_1, \ldots, b_m\}$ such that $M^a \subseteq \{p, q\}$ and $M^{b_i} \subseteq \{q, q\}$.

It is easy to see that the procedure `Clover`(M_0) does not terminate in general. Consider the net in Fig. 8. First, $p \xrightarrow{t_1} p + q$ and we can apply a w-acceleration as previously explained, thus obtaining $p + \infty(q)$. Now we can fire transition t_2,

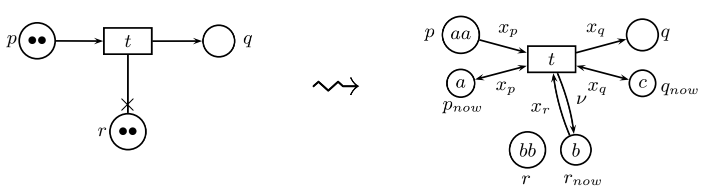

Fig. 9. Simulation of reset nets

$p+\infty(q)\xrightarrow{t_2}p+qq+\infty(q)$. Notice that $p+\infty(q)\sqsubseteq_\iota p+qq+\infty(q)$ with $\iota(p)=p$ and $\iota(q)=q$. The algorithm could then replace $p+qq+\infty(q)$ by its acceleration $p+\infty(qq)$. In the same way, all the ω-markings $p+\infty(q^n)$ are produced by the algorithm.

We could consider yet another type of acceleration, that we could call *dw-acceleration*. Instead of firing t_2 again using one of the infinitely many q's, we could fire it using qq. If we repeat this process, every marking $p+\sum_{i=1}^{m}q^i+\infty(q)$ becomes reachable, and their least upper bound is $p+\infty(q^w)$.

It is true that dw-accelerations give a better approximation of the clover. However, they are not enough, neither any other acceleration we could imagine, since, in general, it is not possible to compute the clover.

Proposition 4. *There is a ν-APN for which the clover is not computable.*

Proof (sketch). Let us suppose that the clover is always computable. Let us see that, in that case, we could decide boundedness of reset nets, which we know to be undecidable [7]. Given a reset net N we build a ν-APN N^* that simulates it. For each place p of N we consider a new place p_{now} in N^*. The construction of N^* is such that p_{now} contains a single token at any time. The firing of any transition ensures (by matching) that the token being used in p coincides with that in p_{now}. Every time a transition resets a place p, the content of p_{now} is replaced by a fresh token, so that no token remaining in p can be used. In this way, our simulation introduces some garbage tokens, that once become garbage, always stay like that. Fig. 9 depicts a simple reset net and its simulation.

Moreover, if M_0 is the initial marking of N, we consider a different identifier a_p for each place p of N. Then, we define the initial marking of N^* as $M_0^*(p_{now})=\{a_p\}$ and $M_0^*(p)=\{a_p,\overset{M_0(p)}{\ldots},a_p\}$, for each $p\in P$. Let us suppose that we can compute the clover Θ. Then N is unbounded iff there is $\mathcal{M}\in\Theta$ containing a product $p_{now}p^\omega$, for some $p\in P$, and boundedness would turn decidable. □

In particular, since Petri Data Nets [16] subsume ν-APN, there is no procedure computing the cover of a Petri Data Net, neither for a Transfer Data Net, thus answering negatively to a question posed in [11].

Accelerations and non-determinism. In Def. 8 we are fixing by means of the mapping ι the relation between names in $\mathcal{M}_1$ and names in $\mathcal{M}_2$. In particular, we

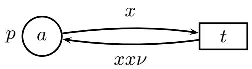

Fig. 10. Accelerations and non-determinism

are choosing among one of such possible relations, and forcing that the chosen relation is kept between all the markings in the generated increasing chain. Thus, we are removing part of the inherent non-determinism in ν-APN that arises in the non-deterministic choosing of consumed names by transitions. For instance, the net in Fig. 10 can fire its only transition $p\xrightarrow{t}p+pp$ and $p \sqsubseteq p+pp$, but we can choose two different ways to map products to products, namely $\iota_1(p)=p$ and $\iota_2(p)=pp$. In the first case, the result of accelerating is the ω-marking $\mathcal{M}_1=\infty(pp)$ (we are always consuming the just created name), while in the second case we obtain $\mathcal{M}_2=p^\omega+\infty(p)$ (we are always taking the name that appeared already in the initial marking).

If we do not impose any particular relation between the names, then at any point any token could be chosen, so that starting from the initial marking, any marking of the form $p^{n_1}+\ldots+p^{n_k}$ can be reached, with least upper bound $\infty(p^\omega)$. Therefore, any acceleration schema that does not impose any mapping ι relating names should compute $\infty(p^\omega)$ as acceleration.

In general, if we can choose between several mappings $\iota_1,\ldots,\iota_k$, because of monotonicity, in each of the limits $\mathcal{M}_1,\ldots,\mathcal{M}_k$ we can again choose between those mappings. Actually, if we choose again the mapping ι_i starting from $\mathcal{M}_i$, the obtained marking is again $\mathcal{M}_i$, by definition of acceleration. However, we could use a different ι_j to accelerate starting from $\mathcal{M}_i$, with $i\neq j$. In our previous example, we can again accelerate starting from $\mathcal{M}_1=\infty(pp)$ and from $\mathcal{M}_2=p^\omega+\infty(p)$. In the case of $\mathcal{M}_1$, we reach $\mathcal{M}_3=ppp+\infty(pp)$, and ppp is obtained from one of the infinitely-many pp. If we apply a dw-acceleration we obtain $\infty(p^\omega)$. Moreover, this is also what we obtain if we accelerate starting from $\mathcal{M}_2$. Indeed, we could choose to fire t starting from $\mathcal{M}_2$ in a mode $\sigma_1(x)=p^\omega$, but the reached marking would be again $\mathcal{M}_2$. However, if we consider a mode $\sigma_2(x)=p$, then we reach $p^\omega+pp+\infty(p)$, where pp is obtained from one of the p. Thus, we can again apply a dw-acceleration to obtain the same ω-marking $\mathcal{M}_2$.

In the previous example we have managed to accelerate (using dw-accelerations) without restricting ourselves to a given mapping ι. However, it remains to see that we can do it in general, that is, that we can still accelerate any loop even if we remove the hypothesis of accelerating with respect to a given mapping ι.

Width-boundedness. We have proved that the cover is not computable in general. To conclude the section we will use the generic Karp-Miller procedure (or a slight variation) to decide a property related to boundedness, called width-boundedness in [18].

Definition 11. *We say N is* width-bounded *if there is $n\in\mathbb{N}$ such that for all reachable M, $|Id(M)|\leq n$.*

Let us see that the forward analysis, though non-terminating in general, can decide width-boundedness. Let us define the predicate over ω-markings `width-bounded`, given by `width-bounded`($\mathcal{M}$) iff $\mathcal{M} = \mathcal{A} + \infty(\emptyset)$. To detect width-boundedness it is enough to stop whenever an ω-marking $\mathcal{M}$ such that $\neg$`width-bounded`($\mathcal{M}$) is found. In this way we can slightly modify the procedure `Clover`, obtaining the algorithm in the right of Fig. 7, `width-Clover`(M_0). The modified algorithm always terminates, returning true iff the net is width-bounded, in which case the clover is computed.

Proposition 5. *Width-boundedness is decidable and the clover is computable for width-bounded ν-APN.*

6 Conclusions and Future Work

In this paper we have established a forward analysis for ν-APNs, an extension of P/T nets with pure name management and creation, with the goal of computing a finite basis of its cover, that is, of the set $\downarrow Post^*(M_0)$. For that purpose, we have applied the results and techniques developed in [10,11] for WSTS. We have defined a friendly presentation of the completion of a ν-APN by means of ω-markings, a natural extension of the analogous concept for P/T nets. We have seen that the transition relation, lifted to the completion, is effective (we can compute successors) and ∞-effective (we can compute the least upper bounds of the sets of ω-markings produced by simple loops). This ensures that it makes sense to apply a forward Karp-Miller procedure. Unfortunately, we have proved that such procedure cannot terminate in general, or we could decide boundedness in reset nets, which is undecidable. As a corollary, a finite basis of the cover is not computable for the class of Transfer Data Nets, not even for the class of Petri Data Nets. Nevertheless, we can slightly modify that algorithm to get a procedure to decide width-boundedness and to compute a finite basis of the cover of a width-bounded net.

The d-accelerations and w-accelerations in Sect. 5 appear naturally when computing the least upper bound of simple loops. However, the dw-accelerations have been sketched in a rather ad-hoc way. It would be interesting to formalize the type of loops they accelerate, and possibly to characterize a subclass of ν-APNs (larger than width-bounded nets) for which `Clover` terminates. In general, it would be interesting to see if a non-deterministic version of accelerations, in which we do not restrict the modes by the relation between the different names involved represented by the mapping ι, is computable. More precisely, it would be interesting to study the structure of the set of markings $\{\mathcal{M} \mid \mathcal{M}_1 \xrightarrow{t^k} \mathcal{M}\}$ (without restricting the modes), to see if it is a directed set, and computing its least upper bound in that case.

Acknowledgments. The authors would like to thank the anonymous referees for their valuable comments.

References

1. Abdulla, P.A., Cerans, K., Jonsson, B., Tsay, Y.: Algorithmic Analysis of Programs with Well Quasi-ordered Domains. Inf. and Comp. 160(1-2), 109–127 (2000)
2. Abdulla, P.A., Nylén, A.: Better is Better than Well: On Efficient Verification of Infinite-State Systems. In: 15th Annual IEEE Symp. on Logic in Computer Science, LICS 2000, pp. 132–140 (2000)
3. Abdulla, P.A., Collomb-Annichini, A., Bouajjani, A., Jonsson, B.: Using Forward Reachability Analysis for Verification of Lossy Channel Systems. Formal Methods in System Design 25(1), 39–65 (2004)
4. Bardin, S., Finkel, A., Leroux, J., Schnoebelen, P.: Flat Acceleration in Symbolic Model Checking. In: Peled, D.A., Tsay, Y.-K. (eds.) ATVA 2005. LNCS, vol. 3707, pp. 474–488. Springer, Heidelberg (2005)
5. Dietze, R., Kudlek, M., Kummer, O.: Decidability Problems of a Basic Class of Object Nets. In: Fundamenta Informaticae, vol. 79, pp. 295–302. IOS Press, Amsterdam (2007)
6. Decker, G., Weske, M.: Instance Isolation Analysis for Service-Oriented Architectures. In: Int. Conference on Services Computing, SCC 2008, pp. 249–256. IEEE Computer Society, Los Alamitos (2008)
7. Dufourd, C., Finkel, A., Schnoebelen, P.: Reset Nets Between Decidability and Undecidability. In: Larsen, K.G., Skyum, S., Winskel, G. (eds.) ICALP 1998. LNCS, vol. 1443, pp. 103–115. Springer, Heidelberg (1998)
8. Finkel, A., Schnoebelen, P.: Fundamental Structures in Well-Structured Infinite Transition Systems. In: Lucchesi, C.L., Moura, A.V. (eds.) LATIN 1998. LNCS, vol. 1380, pp. 102–118. Springer, Heidelberg (1998)
9. Finkel, A., Schnoebelen, P.: Well-Structured Transition Systems Everywhere! Theoretical Computer Science 256(1-2), 63–92 (2001)
10. Finkel, A., Goubault-Larrecq, J.: Forward analysis for WSTS, Part I: Completions. In: Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science, STACS'09, pp. 433–444 (2009)
11. Finkel, A., Goubault-Larrecq, J.: Forward analysis for WSTS, Part II: Complete WSTS. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009. LNCS, vol. 5556, pp. 188–199. Springer, Heidelberg (2009)
12. Geeraerts, G., Raskin, J.-F., van Begin, L.: Expand, enlarge and check: New algorithms for the coverability problem of WSTS. J. Comp. Sys. Sci. 72(1), 180–203 (2006)
13. Gordon, A.: Notes on Nominal Calculi for Security and Mobility. In: Focardi, R., Gorrieri, R. (eds.) FOSAD 2000. LNCS, vol. 2171, pp. 262–330. Springer, Heidelberg (2001)
14. Goubault-Larrecq, J.: On Noetherian spaces. In: 22nd IEEE Symposium on Logic in Computer Science, LICS 2007, pp. 453–462. IEEE Computer Society, Los Alamitos (2007)
15. Jančar, P.: A note on well quasi-orderings for powersets. Information Processing Letters 72(5-6), 155–160 (1999)
16. Lazic, R., Newcomb, T., Ouaknine, J., Roscoe, A.W., Worrell, J.: Nets with Tokens which Carry Data. Fundamenta Informaticae 88(3), 251–274 (2008)

17. Rosa-Velardo, F., de Frutos-Escrig, D., Marroquín-Alonso, O.: On the expressiveness of Mobile Synchronizing Petri Nets. In: 3rd International Workshop on Security Issues in Concurrency, SecCo 2005. ENTCS, vol. 180(1), pp. 77–94. Elsevier, Amsterdam (2007)
18. Rosa-Velardo, F., de Frutos-Escrig, D.: Name creation vs. replication in Petri Net systems. Fundamenta Informaticae 88(3), 329–356 (2008)
19. Rosa-Velardo, F., de Frutos-Escrig, D.: Name creation vs. replication in Petri Net systems. In: Kleijn, J., Yakovlev, A. (eds.) ICATPN 2007. LNCS, vol. 4546, pp. 402–422. Springer, Heidelberg (2007)
20. Schnoebelen, P.: Verifying lossy channel systems has nonprimitive recursive complexity. Inf. Process. Lett. 83(5), 251–261 (2002)