Decidability and Complexity of Petri Nets with Unordered Data

Fernando Rosa-Velardo, David de Frutos-Escrig

Dpto. de Sistemas Informáticos y Computación
Universidad Complutense de Madrid

Abstract

We prove several decidability and undecidability results for ν-PN, an extension of P/T nets with pure name creation and name management. We give a simple proof of undecidability of reachability, by reducing reachability in nets with inhibitor arcs to it. Thus, the expressive power of ν-PN strictly surpasses that of P/T nets. We encode ν-PN into Petri Data Nets, so that coverability, termination and boundedness are decidable. Moreover, we obtain Ackermann-hardness results for all our decidable decision problems. Then we consider two properties, width-boundedness and depth-boundedness, that factorize boundedness. Width-boundedness has already been proven to be decidable. Here we prove that its complexity is also non primitive recursive. Then we prove undecidability of depth-boundedness. Finally, we prove that the corresponding "place version" of all the boundedness problems are undecidable for ν-PN. These results carry over to Petri Data Nets.

Key words: Petri nets, pure names, Well Structured Transition Systems, reachability, boundedness

1. Introduction

Pure names are identifiers with no relation between them other than equality [17]. Dynamic name generation has been thoroughly studied, mainly in the field of security and mobility [17] because they can be used to represent channels, as in π-calculus [29], ciphering keys, as in spi-Calculus [2] or computing boundaries, as in the Ambient Calculus [7].

In previous works we have studied a very simple extension of P/T nets [9], that we called ν-PN [33, 30], for name creation and management.[2] Tokens in

Email addresses: fernandorosa@sip.ucm.es (Fernando Rosa-Velardo), defrutos@sip.ucm.es (David de Frutos-Escrig)

Authors partially supported by the Spanish projects DESAFIOS10 TIN2009-14599-C03-01, PROMETIDOS S2009/TIC-1465 and TESIS TIN2009-14312-C02-01.

[2] Actually, we used the term ν-APN, where the A stands for *Abstract*, though we prefer to use this simpler acronym.

ν-PN are pure names, that can be created fresh, moved along the net and be used to restrict the firing of transitions with name matching. They essentially correspond to the minimal OO-nets of [23], where names are used to identify objects.

In this paper we prove several (un)decidability and complexity results for some decision problems in ν-PN. In [23] the author proved that reachability is undecidable for minimal OO-nets, thus proving that the model surpasses the expressive power of P/T nets. The same result was obtained independently in [33] for a model similar to ν-PN. Both undecidability proofs rely on a weak simulation of a Minsky machine that preserves reachability. We present here an alternative and simpler proof of the same result, based on a simulation of Petri nets (thus, with a much smaller representation gap) with inhibitor arcs, for which reachability is undecidable, that preserves reachability.

Then we study decidability of coverability (in terms of which safety properties can be specified), termination (whether there is an infinite run) and boundedness (whether there are infinitely-many reachable states). We prove that ν-PN belong to the class of Well Structured Transition Systems (WSTS) [3, 16]. Instead of directly proving well structuredness, we simulate ν-PN by means of Data Nets [24], which are WSTS. In Data Nets, tokens are not pure in general, but taken from a linearly-ordered infinite domain. Names can be created, but they can only be guaranteed to be fresh by explicitly using the order in the data domain, for instance by taking a datum which is greater than any other that has been used. Thus, in an unordered version of Data Nets, names cannot be guaranteed to be fresh.

Actually, we prove that Petri Data Nets (PDN), a subclass of Data Nets in which whole-place operations are not allowed, are enough to simulate ν-PN. PDN, unlike general Data Nets, are *strict* WSTS. This implies that not only coverability and termination are decidable for them, but also boundedness [3, 16]. Therefore, the three decidability results carry over to ν-PN.

In [24], the authors proved that coverability, termination and boundedness are not primitive recursive for PDN. However, for an unordered version of PDN (in which names cannot be guaranteed to be fresh) the complexity of those problems is non-elementary. Here we prove that it is enough to add the capability of creating fresh names to unordered PDN (hence obtaining ν-PN), to regain the non primitive recursive complexity of those decision problems. We do it by reducing the same problems in reset nets [11], which are Ackermann-hard [34], to the corresponding ones in ν-PN. For that purpose, we reuse the scheme of the simulation of nets with inhibitor arcs used to prove undecidability of reachability.

Finally, we consider several weaker forms of boundedness. ν-PN can represent infinite state systems that can grow in two orthogonal directions: On the one hand, markings may have an unbounded number of different names; On the other hand, each name may appear in markings an unbounded number of times. In the first case we will say the net is width-unbounded, and in the second we will say it is depth-unbounded. In [31] we proved decidability of width-boundedness by performing a forward analysis that, though incomplete in general for the computation of the cover (the downward closure of the reachability set), can decide width-boundedness. In particular, we instantiated the general framework developed in [14, 15] for forward analyses of WSTS in the case of ν-PN. Now we use the previous complexity results to prove that width-boundedness is also non primitive recursive.

Then we prove undecidability of depth-boundedness. Thus, though both boundedness concepts are closely related, they behave very differently. This result can be rather surprising. Actually, the paper [10] erroneously establishes the decidability of depth-boundedness (called t-boundedness there). To conclude, we use this result to obtain undecidability of all the associated place-boundedness problems, which we call place-boundedness (whether a given place is bounded), p-depth-boundedness (whether a given place is depth-bounded) and p-width-boundedness (whether a given place is width-bounded). Therefore, even if boundedness and width-boundedness are decidable, their "place version" are not. In particular, and to the best of our knowledge, ν-PN is the only class of Petri nets, together with transfer nets [12], for which boundedness is decidable and place-boundedness is undecidable. These undecidability results carry over to Petri Data Nets.

Other related work. CMRS [4] and MSR [8] are also Well Structured models that can be seen as extensions of Petri Nets in which tokens carry ordered data, namely integers and reals, respectively. Alternatively, we could have encoded ν-PN into CMRS, instead of encoding them into PDN. Actually, CMRS and PDN are equivalent up to language equivalence, when coverability is used as acceptance condition for words [5]. We have preferred to use PDN due to the smaller representation gap between them and ν-PN.

Other similar models include Object Nets [35], that follow the so called nets-within-nets paradigm. In Object Nets, tokens can themselves be Petri nets that synchronize with the net in which it lies.

Several papers study the expressive power of Object Nets. The paper [20] considers a two level restriction of Object Nets, called Elementary Object Nets (EON), and proves undecidability of reachability for them. This result extends those in [22]. Moreover, some subclasses are proved to have decidable reachability. In [21] it is shown that, when the synchronization mechanism is extended so that object tokens can be communicated, then Turing completeness is obtained. However, in all these models processes (object nets) do not have identities.

Nested Petri Nets [25] also have nets as tokens, that evolve autonomously, move along the system net, synchronize with each other (horizontal synchronization steps) or synchronize with the system net (vertical synchronization steps). Nested nets are more expressive than ν-PN. Indeed, it is possible to simulate every ν-PN by means of a Nested Petri Net which uses only object-autonomous and horizontal synchronization steps. In Nested Petri Nets, reachability and boundedness are undecidable, although other problems, like termination, remain decidable [26]. Thus, decidability of termination can also be obtained as a consequence of [26]. Here we obtain decidability of termination on the way of the proof of decidability for boundedness and coverability.

Outline. The rest of the paper is structured as follows. Section 2 presents some basic results and notations we will use throughout the paper. Section 3 defines ν-PN. Section 4 proves undecidability of reachability. In Sect. 5 we prove decidability of coverability, termination and boundedness. In Sect. 6 we prove that those problems are not primitive recursive. Section 7 presents our results about weaker forms of boundedness and in Section 8 we present our conclusions.

2. Preliminaries

For an arbitrary set A we denote $\bar{A} = \{\bar{a} \mid a \in A\}$, that satisfies $A \cap \bar{A} = \emptyset$. For any $n \in \mathbb{N}$ we write $\lfloor n \rfloor = \{1, ..., n\}$. For a vector $e = (a_1, ..., a_k) \in \mathbb{N}^k$ we write $e(i) = a_i$ and we denote by $\mathbf{0}$ the null vector in any $\mathbb{N}^k$. We simply use $\leq$, $+$ and $-$ for the corresponding component-wise operations in $\mathbb{N}^k$.

wqo. A *quasi order* (qo) is a reflexive and transitive binary relation on a set A. A *partial order* (po) is an antisymmetric quasi order. For a qo $\sqsubseteq$ we write $a \sqsubset b$ if $a \sqsubseteq b$ and $b \not\sqsubseteq a$. The upward closure of a subset B is $\uparrow B = \{a \in A \mid \exists b \in B \text{ st } b \sqsubseteq a\}$. A qo $\sqsubseteq$ is *well (wqo)* [28] if for every infinite sequence $a_0, a_1, \ldots$ there are i and j with $i < j$ such that $a_i \sqsubseteq a_j$. A *well partial order* (wpo) is a po that is well. If $\{(A_i, \sqsubseteq_i)\}_{i=1}^k$ is a finite family of wqo then $(A_1 \times ... \times A_k, \sqsubseteq)$ is also a wqo, where $(a_1, ..., a_k) \sqsubseteq (b_1, ..., b_k)$ iff $a_i \sqsubseteq_i b_i$ for each $i \in \lfloor k \rfloor$. Thus, since the natural order in $\mathbb{N}$ is a wpo, so is the component-wise order in $\mathbb{N}^k$.

Words and multisets. For a set A, a sequence $u = a_1...a_n$ with each $a_i \in A$ is a finite *word* over A, and we write $|u| = n$ and $u(i) = a_i$. We denote by A^* the set of finite words over A. If $\leq$ is a qo over A we can define the qo over A^* given by $a_1...a_n \leq^* b_1...b_m$ iff there is $h : \lfloor n \rfloor \to \lfloor m \rfloor$ increasing such that $a_i \leq b_{h(i)}$. If we do not demand h to be increasing, but only injective, then we obtain the multiset order, which is denoted by $\leq^\oplus$.

Thus, a (finite) *multiset* over A is the equivalence class of a word over A, modulo the kernel of $\leq^\oplus$, that is, the equivalence relation $\leq^\oplus \cap {}^\oplus\!\geq$. We will call *ordering* of a multiset to any of its representatives. We will denote by $A^\oplus$ the set of finite multisets over A. It is well known that if $(A, \leq)$ is a wqo then so are $(A^*, \leq^*)$ and $(A^\oplus, \leq^\oplus)$ [28].

For a multiset $m \in A^\oplus$ we write $m(a)$ to denote the multiplicity of a in m, that is, the number of occurrences of a in m. Notice that m is determined by $m(a)$ for all $a \in A$. We will use set notation for multisets when convenient. We denote by $supp(m)$ the support of m, that is, the set $\{a \in A \mid m(a) > 0\}$ and by $|m| = \sum_{a \in supp(m)} m(a)$ the cardinality of m. Given two multisets $m_1, m_2 \in A^\oplus$ we denote by $m_1 + m_2$ the multiset satisfying $(m_1 + m_2)(a) = m_1(a) + m_2(a)$. We will write $m_1 \subseteq m_2$ if $m_1(a) \leq m_2(a)$ for every $a \in A$. In this case, we can define $m_2 - m_1$, given by $(m_2 - m_1)(a) = m_2(a) - m_1(a)$. We will denote by $\sum$ the extended multiset sum operator and by $\emptyset \in A^\oplus$ the multiset $\emptyset(a) = 0$, for every $a \in A$. If $f : A \to B$ and $m \in A^{\oplus}$, then we define $f(m) \in B^{\oplus}$ by $f(m)(b) = \sum_{f(a)=b} m(a)$.

Transition systems. A *transition system* is a tuple $(S, \to, s_0)$, where S is a (possibly infinite) set of states, $s_0 \in S$ is the initial state and $\to \subseteq S \times S$. We denote by $\to^*$ the reflexive and transitive closure of $\to$.

The *reachability problem* in a transition system consists in deciding for a given states s_f whether $s_0 \to^* s_f$. The *termination problem* consists in deciding whether there is an infinite sequence $s_0 \to s_1 \to s_2 \to \cdots$. The *boundedness problem* consists in deciding whether the set of reachable states is finite. For any transition system $(S, \to, s_0)$ endowed with a qo $\leq$ we can define the *coverability problem*, that consists in deciding, given a state s_f, whether some state $s \in \uparrow \{s_f\}$ is reachable.

WSTS. A *Well Structured Transition System* (WSTS) is a tuple $(S, \to, s_0, \leq)$, where $(S, \to, s_0)$ is a finitely branching transition system and $\leq$ is a decidable wqo compatible with $\to$ (meaning that $s_1' \geq s_1 \to s_2$ implies that there is $s_2' \geq s_2$ with $s_1' \to s_2'$). For WSTS,[3] the coverability and the termination problems are decidable [3, 16]. A WSTS is said to be *strict* if it satisfies the following strict compatibility condition: $s_1' > s_1 \to s_2$ implies that there is $s_2' > s_2$ with $s_1' \to s_2'$. For strict WSTS, also the boundedness problem is decidable [16].

Petri Nets. Next we define *Place/Transition Nets* (P/T nets) in order to set our notations. A P/T net is a tuple $N = (P, T, F)$ where P and T are disjoint finite sets of places and transitions, respectively, and $F : (P \times T) \cup (T \times P) \to \mathbb{N}$. A marking M of N is a finite multiset of places of N, that is, $M \in P^{\oplus}$.

We denote by $t^{\bullet}$ and ${}^{\bullet}t$ the multisets of postconditions and preconditions of t, respectively, that is, $t^{\bullet}(p) = F(t, p)$ and ${}^{\bullet}t(p) = F(p, t)$. A transition t is enabled at marking M if ${}^{\bullet}t \subseteq M$. The reached state of N after the firing of t is $M' = (M - {}^{\bullet}t) + t^{\bullet}$.

We will write $M \xrightarrow{t} M'$ if M' is the reached marking after the firing of t at marking M, and $M \to M'$ if there is some t such that $M \xrightarrow{t} M'$. For a transition sequence $\tau = t_1 \ldots t_m$ we will write $M \xrightarrow{\tau} M'$ to denote the consecutive firing of transitions t_1 to t_m. We write $M \to^* M'$ if there is some τ such that $M \xrightarrow{\tau} M'$.

3. Petri nets with name creation

Let us now extend P/T nets with the capability of name management by defining ν-PN. In a ν-PN names can be created, communicated and matched. We formalize name management by replacing ordinary tokens by distinguishable ones, thus adding colours to our nets. We fix an infinite set *Id* of names, that can be carried by tokens of any ν-PN. In order to handle these colors, we need

[3] Actually, another effectiveness condition, *effective Pred-basis*, is needed for coverability.

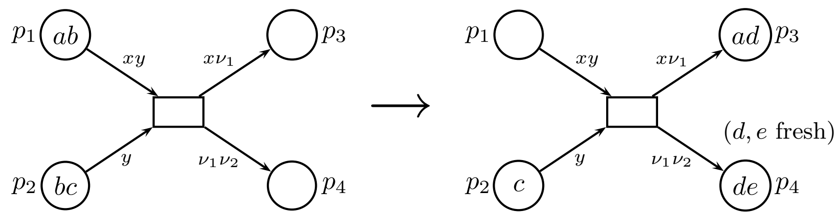

Figure 1: A simple ν-PN

matching variables labelling 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 $\Upsilon \subset Var$, ranged by $\nu, \nu_1, \ldots$ that can only be instantiated to fresh names.

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

We also take $Var(t) = pre(t) \cup post(t)$ and $fVar(t) = Var(t) \setminus \Upsilon$. To avoid tedious definitions, along the paper we will consider an arbitrary ν-PN $N = (P, T, F)$. A ν-PN is depicted as usual: places are drawn as circles, transitions are drawn as boxes and F is represented by arrows labeled in this case by a multiset of variables, that is, we draw an arrow from x to y labeled by $F(x, y)$ whenever $F(x, y) \neq \emptyset$.

Definition 2. A *marking* of N is a mapping $M : P \to Id^{\oplus}$. We denote by $Id(M)$ the set of names in M, that is, $Id(M) = \bigcup_{p \in P} supp(M(p))$.

We will fix an arbitrary initial marking M_0 of N. Like in other classes of high-order nets, transitions are fired with respect to a mode, that chooses which tokens are taken from preconditions and which are put in postconditions. Given a transition t of a net N, a *mode* of t is an injection $\sigma : Var(t) \to Id$, that instantiates each variable to an identifier. We will use $\sigma, \sigma', \sigma_1 \ldots$ to range over modes.

Definition 3. Let M be a marking, $t \in T$ 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 $\sigma(\nu) \notin Id(M)$ for all $\nu \in \Upsilon \cap Var(t)$. The reached state after the *firing* of t with mode σ is the marking M', given by

$$M'(p) = (M(p) - \sigma(F(p, t))) + \sigma(F(t, p)) \quad \textit{for all } p \in P.$$

We will write $M \xrightarrow{t(\sigma)} M'$ to denote that M' is reached from M when t is fired with mode σ, and extend the notation as done for P/T nets. We will denote by

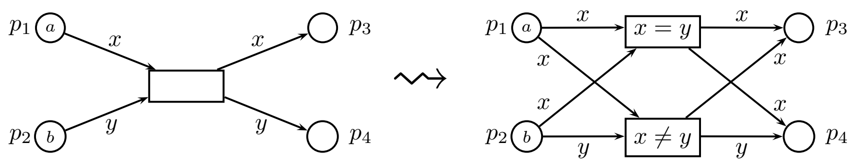

Figure 2: The net on the left cannot check for inequalities (it can fire its transition when $a = b$ or $a \neq b$). The net on the right can fire the transition in the top when $a = b$, and the one in the bottom when $a \neq b$.

$Reach(N)$ the set of reachable markings of N. In order to keep usual notations in P/T nets, we will assume that there is a "distinguished" color $\bullet \in Id$ and a distinguished variable ϵ that can only be instantiated to $\bullet$, which will be omitted in our figures. Thus, we can manage ordinary black tokens with the same notations as in P/T nets.

Example 1. Figure 1 depicts a simple ν-PN with four places and a single transition. This transition moves one token from p_1 to p_3 (because of variable x labelling both arcs), removes a token from p_1 and p_2 provided they carry the same name (variable y appears in both incoming arcs but it does not appear in any outgoing arc), and two different names are created: one appears both in p_3 and p_4 (because of $\nu_1 \in \Upsilon$) and the other appears only in p_4 (because of $\nu_2 \in \Upsilon$).

Notice that we demand modes to be injections (unlike in [30]), which formalizes the fact that we can check for inequality. For instance, in the example in Fig. 1 the two tokens taken from p_1 must carry different names because we are labelling the arc from p_1 to t with two different variables, namely x and y. The capability of checking for inequality among all the names involved in the firing of a transition improves the expressive power of the model (see Fig. 2), so that all the positive results can be transferred to the subclass of ν-PN in which check for inequalities are not allowed.

If a ν-PN has no arc labelled with variables from Υ then only a finite number of identifiers (those in the initial marking) can appear in any reachable marking. It is easy to see that these nets can be expanded to an equivalent P/T net. In particular, reachability is decidable for any such net, as it is for P/T nets [13].

In order to simplify some of our proofs, we will work with a subclass of ν-PN in which transitions can at most create one fresh name.

Definition 4. N is *normal* if there is a variable $\nu \in \Upsilon$ such that for every pair $(x, y) \in (P \times T) \cup (T \times P)$, either $F(x, y) = \{\nu\}$ or $F(x, y) \cap \Upsilon = \emptyset$.

Every ν-PN can be simulated by a normal ν-PN. Intuitively, the simulation splits each transition that creates $k > 1$ names into k transitions that must be fired consecutively, each of which creates a single name. Therefore, from now on we will assume that ν-PN are normal when needed.

4. Undecidability of reachability for ν-PN

Let us now prove that reachability is undecidable for ν-PN. In [23] (and independently in [33]) undecidability of reachability is proved by reducing reachability of the final state with all the counters containing zero in Minsky machines to reachability in ν-PN. In this section we prove that same result in a more simple way, by reducing reachability of inhibitor nets (that allow to check for zero) to reachability in ν-PN.

An *inhibitor net* is a tuple $N = (P, T, F, F_{in})$, where P and T are disjoint sets of places and transitions, respectively, $F \subseteq (P \times T) \cup (T \times P)$, and $F_{in} \subseteq P \times T$. Pairs in F_{in} are inhibitor arcs. For a transition $t \in T$ we write ${}^\bullet t = \{p \in P \mid (p,t) \in F\}$, $t^\bullet = \{p \in P \mid (t,p) \in F\}$ and ${}^i t = \{p \in P \mid (p,t) \in F_{in}\}$. In figures we will draw a circle instead of an arrow to indicate that an arc is an inhibitor arc.

A *marking* of an inhibitor net N is a multiset of places of N. A transition t of N is enabled if $M(p) > 0$ for all $p \in {}^\bullet t$ and $M(p) = 0$ for all $p \in {}^i t$. In that case t can be fired, producing $M' = (M - {}^\bullet t) + t^\bullet$. The reachability problem is undecidable for nets with two inhibitor arcs [13].

Proposition 1. *Reachability is undecidable for ν-PN.*

Proof. Given an inhibitor net $N = (P, T, F, F_{in})$ let us build a ν-PN N^* that simulates it. For each $p \in P$ let us consider a different variable x_p. Then we define $N^* = (P \cup \bar{P}, T, F^*)$ where:

- If $(p,t) \in F$ then $F^*(p,t) = F^*(\bar{p},t) = F^*(t,\bar{p}) = \{x_p\}$,
- If $(t,p) \in F$ then $F^*(t,p) = F^*(\bar{p},t) = F^*(t,\bar{p}) = \{x_p\}$,
- If $(p,t) \in F_{in}$ then $F^*(\bar{p},t) = \{x_p\}$ and $F^*(t,\bar{p}) = \{\nu\}$.
- $F^*(x,y) = \emptyset$ elsewhere.

Let $h : P \to Id$ be an injection. For any marking M of N, we define M^h as the marking of N^* given by $M^h(\bar{p}) = \{h(p)\}$ and $M^h(p) = \{h(p), \overset{M(p)}{\ldots}, h(p)\}$, for each $p \in P$. Moreover, let us consider a fixed injection h_0, so that $M_0^{h_0}$ is the initial marking of N^*, provided M_0 is the initial marking of N.

For each place p of N we are considering a new place $\bar{p}$ in N^*. The construction of N^* is such that $\bar{p}$ contains a single token at any time. The firing of any transition ensures that the token being used in p carries a name that coincides with the name carried by the token in $\bar{p}$. Every time a transition t checks the emptiness of a place p, the content of $\bar{p}$ is replaced by a fresh token. Therefore, our simulation can cheat, firing t even if there are some tokens in p. In that case, those tokens will remain as garbage. Moreover, once a token becomes garbage it can never be removed. Let us see that $M_0 \to^* M$ iff there is h such that $M_0^{h_0} \to M^h$. Moreover, we can take h so that $h(p) = h_0(p)$ or $h(p) = \overline{h_0(p)}$.

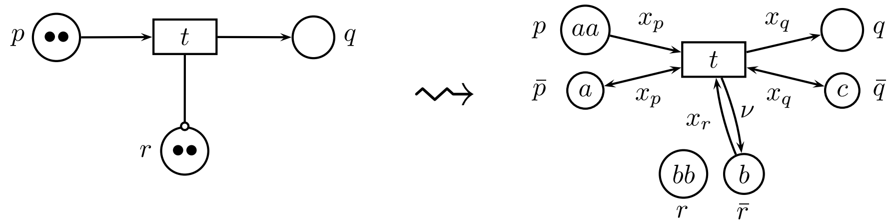

Figure 3: Weak simulation of Petri nets with inhibitor arcs

Let us first assume that $M_0 \xrightarrow{\tau} M$. We take $h(p) = h_0(p)$ if the emptiness of p is not checked in τ, and $h(p) = \overline{h_0(p)}$ otherwise. Then τ is a transition sequence of N^* that can reach M^h from $M_0^{h_0}$. Indeed, if the emptiness of p is not checked then $M^*(\bar{p}) = M_0^{h_0}(p) = h_0(p)$. Otherwise, if t is the last transition to check its emptiness, in N^* we can fire the transition t with a mode σ such that $\sigma(\nu) = \overline{h_0(p)}$.

Conversely, let $M_0^{h_0} \xrightarrow{\tau} M^h$ for some h as above. Since M^h does not contain garbage tokens, then that run has not cheated, so that it is a correct simulation of N and $M_0 \xrightarrow{\tau} M$. Fig. 3 depicts a simple inhibitor net and its simulation. Notice that there are $2^{|P|}$ possible injections.[4] Thus, we have reduced reachability in inhibitor nets, which is undecidable [13], to a finite set of reachability problems in ν-PN.

Notice that the ν-PN built in the previous proof does not need to check for inequality, so that already in the restricted model, in which modes are not necessarily injective, reachability is undecidable.

5. Strict well structuredness of ν-PN

In this section we prove that ν-PN are strict WSTS, so that termination, coverability and boundedness are decidable. Instead of directly proving that they are an instance of WSTS, we will see how they can be encoded within Petri Data Nets, a more general model, that is known to be a WSTS.

In the first place, let us clarify what is the order we are interested in for ν-PN. One could think that the desired order is the following:

$$M_1 \sqsubseteq M_2 \Leftrightarrow M_1(p) \subseteq M_2(p) \text{ for all } p \in P$$

This order is not a wqo: it is enough to consider a ν-PN with a single place p and a sequence of pairwise different identifiers $(a_i)_{i=1}^{\infty}$, and define $M_i(p) = \{a_i\}$ for all $i = 1, 2, \ldots$ which trivially satisfies that for all $i < j$, $M_i \not\sqsubseteq M_j$.

However, this order is too restrictive, since it does not take into account the abstract nature of pure names. Indeed, whenever a new name is created, actually any other fresh name could have been created. Therefore, reachability

[4] Actually, it is enough to consider only the places that can be checked for emptiness.

(or coverability) of a given marking is equivalent to reachability (or coverability) of any marking produced after consistently renaming the new names in it. For homogeneity, we will suppose that we can rename every name, even those appearing in the initial marking (which, after all, are a fixed number of names). To capture these intuitions, we identify markings up to renaming of names.

Definition 5. Two markings M and M' are *α-equivalent,* in which case we write $M \equiv_\alpha M'$, if there is a bijection $\iota : Id(M) \to Id(M')$ such that for all $p \in P$ and $a \in Id(M)$, $M'(p)(\iota(a)) = M(p)(a)$.

We will write $M \equiv_\iota M'$ to stress the use of the particular mapping ι in the previous definition. Moreover, for a marking M and a set of identifiers A, any bijection $\iota : Id(M) \to A$ defines a marking that we denote as $\iota(M)$, given by $\iota(M)(p)(\iota(a)) = M(p)(a)$, which is α-equivalent to M. The following result states that $\equiv_\alpha$ is a forward and backward bisimulation.

Proposition 2. *Let* $M_1 \xrightarrow{t(\sigma)} M'_1$.

- *If* $M_1 \equiv_\alpha M_2$ *then there is* M'_2 *and* σ' *such that* $M'_1 \equiv_\alpha M'_2$ *and* $M_2 \xrightarrow{t(\sigma')} M'_2$.
- *If* $M'_1 \equiv_\alpha M'_2$ *then there is* M_2 *and* σ' *such that* $M_1 \equiv_\alpha M_2$ *and* $M_2 \xrightarrow{t(\sigma')} M'_2$.

PROOF. Let $A = Id(M_1) \setminus Id(M'_1)$ and B the set of names created by $t(\sigma)$. Then, $Id(M'_1) = (Id(M_1) \setminus A) \cup B$.

- Assume $M_1 \equiv_\iota M_2$ and let $\sigma' = \iota \circ \sigma$. Transition t can be fired from M_2 with mode σ', obtaining M'_2 with $Id(M'_2) = (Id(M'_1) \setminus \iota(A)) \cup B'$ for some B' of the same cardinality than B. We define ι' by extending ι to B so that $\iota(B) = B'$, which verifies $M_2 \equiv_{\iota'} M'_2$.
- Assume now that $M'_1 \equiv_{\iota'} M'_2$ and let us define $\iota : Id(M_1) \to Id(M'_2) \cup A$ by $\iota(a) = \iota'(a)$ if $a \in Id(M'_1)$, and $\iota(a) = a$ if $a \in A$. Then $M_2 = \iota(M_1)$ and $\sigma' = \iota \circ \sigma$ satisfy $M_1 \equiv_\iota M_2$ and $M_2 \xrightarrow{t(\sigma')} M'_2$.

Example 2. If we represent a marking M of the net in Fig. 1 by a tuple $(M(p_1), M(p_2), M(p_3), M(p_4))$, then the markings $M_1 = (\{a,b\}, \{b,c\}, \emptyset, \emptyset)$ and $M_2 = (\{a,c\}, \{b,c\}, \emptyset, \emptyset)$ are two α-equivalent markings of that ν-PN. Indeed, $M_1 \equiv_\iota M_2$ with $\iota(a) = a$, $\iota(b) = c$ and $\iota(c) = b$. M_1 can evolve to the marking $M'_1 = (\emptyset, \{c\}, \{a,d\}, \{d,e\})$ when it fires t and M_2 can evolve to $M'_2 = (\emptyset, \{b\}, \{a,e\}, \{d,e\})$. Indeed, $M'_1 \equiv_\alpha M'_2$ holds.

We are now ready to define the order we are interested in for ν-PN.

Definition 6. Let M_1 and M_2 be markings of N. We will write $M_1 \sqsubseteq_\alpha M_2$ if there is a marking M'_1 such that $M'_1 \equiv_\alpha M_1$ and $M'_1 \sqsubseteq M_2$.

Then, $M_1 \sqsubseteq_\alpha M_2$ when there is ι such that $M_1 \equiv_\iota M'_1 \sqsubseteq M_2$, or equivalently, when $\iota(M_1) \sqsubseteq M_2$. We will write $M_1 \sqsubseteq_\iota M_2$ to emphasize on the use of ι. Clearly, $\sqsubseteq_\alpha$ is a decidable quasi order. Moreover, the kernel of $\sqsubseteq_\alpha$ is $\equiv_\alpha$, that is, $M_1 \sqsubseteq_\alpha M_2$ and $M_2 \sqsubseteq_\alpha M_1$ iff $M_1 \equiv_\alpha M_2$.

5.1. From names to multisets

Now let us see that because we are identifying markings up to renaming of names, we can abstract away those names and represent markings as multisets of vectors. As a consequence, $\sqsubseteq_\alpha$ is a wqo. Notice that, in particular, the counterexample we used to prove that $\sqsubseteq$ is not a wqo is no longer valid, since all those markings were α-equivalent. With this representation of markings we are closer to Petri Data Nets, in which markings are words of vectors.

A marking is a mapping $M : P \to (Id \to \mathbb{N})$ that says, for a given place p and an identifier a, how many times the token a can be found in place p. However, we can also currify those mappings as $M : Id \to (P \to \mathbb{N})$. Intuitively, since the behavior of a net is invariant under renaming, as we proved in Prop. 2, we can represent markings (modulo $\equiv_\alpha$) as multisets in $(P \to \mathbb{N})^\oplus$, that is, in $(\mathbb{N}^{|P|})^\oplus$. In this way, we can represent a marking by a multiset with a cardinality that equals the number of different identifiers in it.

Example 3. The marking M_1 in Example 2 can be represented as the multiset $\{(1,0,0,0),(1,1,0,0),(0,1,0,0)\}$, where the first vector stands for the identifier a, the second stands for b, and the third for c. As we will prove later, since M_2 is α-equivalent to M_1, it is represented by that same multiset. The markings M_1' and M_2' are represented by $\{(0,0,1,0),(0,1,0,0),(0,0,1,1),(0,0,0,1)\}$.

Let us see it formally. In the following we will assume that $P = \{p_1,\ldots,p_k\}$.

Definition 7. For a marking M of N, we define the vector $M^a \in \mathbb{N}^k$ as $M^a = (M(p_1)(a),\ldots,M(p_k)(a))$ and $\overline{M} = \{M^a \mid a \in Id(M)\} \in (\mathbb{N}^k)^\oplus$.

Let us denote by $\leq^\oplus$ the canonic order in $(\mathbb{N}^k)^\oplus$, that is, the multiset order induced by the component-wise order in $\mathbb{N}^k$. Since the latter is a wpo, so is $\leq^\oplus$. Moreover, it coincides with $\sqsubseteq_\alpha$, as we prove next.

Lemma 1. *Let M_1 and M_2 be two markings. Then $M_1 \sqsubseteq_\alpha M_2$ iff $\overline{M}_1 \leq^\oplus \overline{M}_2$.*

Proof. Let $\overline{M}_1 = \{A_1,\ldots,A_n\}$ and $\overline{M}_2 = \{B_1,\ldots,B_n\}$ with $A_i = M_i^{a_i}$ and $B_j = M_2^{b_j}$. If $M_1 \sqsubseteq_\iota M_2$ then define $h(i)$ such that $B_{h(i)} \leq M_2^{\iota(a_i)}$. Then $A_i(j) = M_1(p_j)(a_i) \leq M_2(p_j)(\iota(a_i)) = B_{h(i)}(j)$, so that $A_i \leq B_{h(i)}$ and therefore $\overline{M}_1 \leq^\oplus \overline{M}_2$.

Conversely, since $\overline{M}_1 \leq^\oplus \overline{M}_2$, there is $h : \{1,\ldots,n\} \to \{1,\ldots,m\}$ injective such that $A_i \leq B_{h(i)}$. Let us define $\iota : Id(M_1) \to Id(M_2)$ by $\iota(a_i) = b_{h(i)}$. Then we have $M_1(p_j)(a_i) = M_1^{a_i}(p_j) \leq M_2^{b_{h(i)}}(p_j) = M_2^{\iota(a_i)}(p_j) = M_2(p_j)(\iota(a_i))$. Therefore, $M_1(p_j)(a) \leq M_2(p_j)(\iota(a))$ for all $a \in Id(M_1)$ and the thesis follows.

As mentioned before, since $\leq^\oplus$ is a wqo, then so is $\sqsubseteq_\alpha$. Moreover, since $\leq^\oplus$ is a po, two α-equivalent markings are represented by the same multiset. Thanks to Prop. 2 and Lemma 1, a ν-PN can be seen as a rewrite theory, in which we rewrite multisets of vectors. Let us see what is the effect of firing a transition in terms of multisets. For any $x \in Var$, let $pre_t(x) = (F(p_1,t)(x),...,F(p_k,t)(x))$ and $post_t(x) = (F(t,p_1)(x),...,F(t,p_k)(x))$ in $\mathbb{N}^k$.

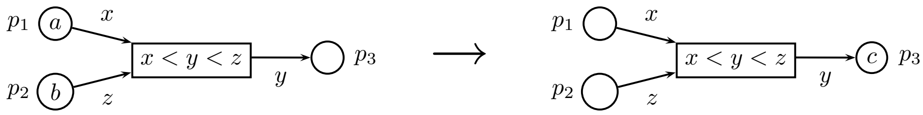

Figure 4: Firing of a Petri Data Net transition (assuming $a < c < b$)

Lemma 2. $M_1 \xrightarrow{t(\sigma)} M_2$ *iff the following holds:*

- $M_2^{\sigma(x)} = (M_1^{\sigma(x)} - pre_t(x)) + post_t(x)$ *for all* $x \in Var(t)$,
- $M_2^a = M_1^a$ *for all* $a \neq \sigma(x)$ *for all* $x \in Var(t)$,
- $M_1^{\sigma(\nu)} = \mathbf{0}$ *if* $\nu \in Var(t)$.

PROOF. Assume $M_1 \xrightarrow{t(\sigma)} M_2$ holds. If $\sigma(x) \neq a$ for every $x \in Var(t)$ then clearly $M_2^a = M_1^a$. Otherwise, let $p = p_j \in P$. We know that $M_2(p) = (M_1(p) - \sigma(F(p,t)) + \sigma(F(t,p))$. Then, $M_2^a(j) = M_2(p)(a) = (M_1(p)(a) - \sigma(F(p,t))(a)) + \sigma(F(t,p))(a) = (M_1^a(j) - F(p_j,t)(x)) + F(t,p_j)(x) = (M_1^a(j) - pre_t(x)(j)) + post_t(x)(j)$. This holds for all $j \in \lfloor k \rfloor$, so that $M_2^a = (M_1^a - pre_t(x)) + post_t(x)$. Finally, since $\sigma(\nu) \notin Id(M_1)$, we have $M_1^{\sigma(\nu)} = \mathbf{0}$.

Conversely, the third item states that $\sigma(\nu) \notin Id(M_1)$. As before, by the first and second items we have that $M_2(p) = (M_1(p) - \sigma(F(p,t))) + \sigma(F(t,p))$ and the thesis follows.

In particular, notice that $M_2^{\sigma(\nu)} = post_t(\nu)$, because $pre_t(\nu)$ is always $\mathbf{0}$.

5.2. Petri Data Nets

We now briefly define Petri Data Nets. For more details see [24]. A *Petri Data Net* (PDN) is a tuple $N = (P, T, pre, post)$ where P is a finite set of places, T is a finite set of transitions and for each $t \in T$, $pre_t, post_t \in (\mathbb{N}^{|P|})^*$ with $|pre_t| = |post_t|$. A marking s of N is a finite sequence of vectors in $\mathbb{N}^{|P|} \setminus \mathbf{0}$. A marking s is the $\mathbf{0}$-contraction of a sequence $s' \in (\mathbb{N}^{|P|})^*$ if it can be obtained by removing all the occurrences of $\mathbf{0}$ from s'. We write $s_1 \xrightarrow{t} s_2$ for $t \in T$ with $|pre_t| = |post_t| = n$ if:

- s_1 is the $\mathbf{0}$-contraction of $u_0 x_1 u_1 \cdots u_{n-1} x_n u_n$ with $u_i \in (\mathbb{N}^{|P|} \setminus \mathbf{0})^*$ and $x_i \in \mathbb{N}^{|P|}$,
- $x_i \geq pre_t(i)$ and $y_i = (x_i - pre_t(i)) + post_t(i)$ for $i \in \{1, \ldots, n\}$,
- s_2 is the $\mathbf{0}$-contraction of $u_0 y_1 u_1 \cdots u_{n-1} y_n u_n$

The transition system of a PDN is a strict WSTS, in taking $\leq^*$ as the word order induced by the component-wise order in $\mathbb{N}^k$. Therefore, coverability, termination and boundedness are all decidable for them.

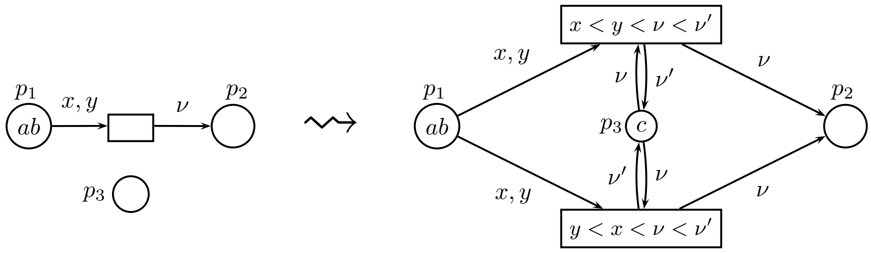

Figure 5: Simulation of ν-PN by a PDN, assuming $a < c$ and $b < c$

A PDN can be graphically depicted similarly as ν-PN, by considering that tokens carry data taken from a linearly ordered and dense domain, and arcs are labelled with variables (not in Υ) that are totally ordered. If $x_1 < ... < x_n$ are all the variables adjacent to a transition t, then $pre_t(i)$ specifies the tokens that must be taken from each place carrying the datum to which x_i is instantiated (and analogously for $post_t$). For instance, Fig. 4 depicts a PDN with a single transition t given by $pre_t = (1,0,0)(0,0,0)(0,1,0)$ and $post_t = (0,0,0)(0,0,1)(0,0,0)$.

5.3. From PDN to ν-PN

The graphical representation of PDN commented above suggests the short gap between ν-PN and them. We have brought ν-PN closer to PDN (and not the other way around, as suggested by that graphical representation), by representing markings as multisets of vectors, similar to the words of vectors in PDN. In order to close the gap all we have to do is to ignore the relative order in tokens in a PDN, and use this order only to simulate the fresh name creation of ν-PN.

Definition 8. We call *ordering* of a transition $t \in T$ of N to any injection $h : \lfloor |fVar(t)| \rfloor \to fVar(t)$. An *ordering* of a marking M is any ordering of $\overline{M}$.

Let us denote by $e \in \mathbb{N}^k$ the tuple $(0, ..., 0, 1)$. In the next definition, in order to simplify notations, we assume that N is normal, and that the place p_k of N is a dummy place (with no arcs coming out of it or going into it). In particular, the k-th component of every $pre_t(x)$ and $post_t(x)$ is null.

Definition 9. We define the PDN $N' = (P, T', pre, post)$, where:

- $T' = \{t^h \mid t \in T,\ h \textit{ ordering of } t\}$,
- Let $t^h \in T'$ with $|fVar(t)| = m$. Then we take

$$\begin{array}{rcllll} pre_{t^h} & = & pre_t(h(1)) \; ... \; pre_t(h(m)) & e & \mathbf{0} \\ post_{t^h} & = & post_t(h(1)) \; ... \; post_t(h(m)) & post_t(\nu) & e \end{array}$$

Since N and N' have the same set of places, any ordering of a marking of N is a marking of N'. However, N' uses the place p_k, unlike N, to hold exactly one token, which carries a datum greater than any other in the marking, represented by the tuple e. Thus, any marking reachable in N' from some marking of the form $s\ e$, where the k-th component of every vector in s is null, is also of that form. For such s, we will write $s' = s\ e$.

Lemma 3. *The following holds:*

- *If $M_1 \xrightarrow{t} M_2$ then for every s_1 ordering of M_1 there is an ordering s_2 of M_2 and h an ordering of t such that $s'_1 \xrightarrow{t^h} s'_2$,*
- *If $s'_1 \xrightarrow{t^h} u$ for some ordering s_1 of M_1 then $u = s'_2$ for an ordering s_2 of some M_2 such that $M_1 \xrightarrow{t} M_2$.*

Proof. First, assume that $M_1 \xrightarrow{t(\sigma)} M_2$, and let s_1 be an ordering of M_1. Define h so that $\sigma(h(i)) = a_i$, with $s_1 = u_1\ M_1^{a_1} u_2 ... u_m M_1^{a_m} u_{m+1}$. The marking s'_1 is the $\mathbf{0}$-contraction of $u_1\ M_1^{a_1} u_2 ... u_m M_1^{a_m} u_{m+1}\ e\ \mathbf{0}$. By Lemma 2, it evolves after firing t^h in N' to the $\mathbf{0}$-contraction of $u_1\ M_2^{a_1} u_2 ... u_m M_2^{a_m} u_{m+1}\ post_t(\nu)\ e$, which is (without the final e) an ordering of M_2.

The second item follows similarly as the previous one, now by using the converse implication in Lemma 2 and by construction of N'.

Lemma 4. *The following holds:*

- *If $s'_1 \leq^* s'_2$ for s_i ordering of M_i, then $M_1 \sqsubseteq_\alpha M_2$.*
- *If $M_1 \sqsubseteq_\alpha M_2$ then for every s_1 ordering of M_1 there is an ordering s_2 of M_2 such that $s'_1 \leq^* s'_2$.*
- *If $M_1 \sqsubseteq_\alpha M_2$ then for all s_2 ordering of M_2 there is s_1 ordering of M_1 such that $s'_1 \leq^* s'_2$.*

Proof. First notice that $s_1 \leq^* s_2$ iff $s'_1 \leq^* s'_2$. If s_i is an ordering of $\overline{M}_i$ for $i \in \{1, 2\}$, then $s_1 \leq^* s_2$ implies $\overline{M}_1 \leq^\oplus \overline{M}_2$, and by Lemma 1 we have $M_1 \sqsubseteq_\alpha M_2$ and the first item follows.

For the second and third items, if $\overline{M}_1 \leq^\oplus \overline{M}_2$ and s_1 is an ordering of $\overline{M}_1$ then there is an ordering of $\overline{M}_2$ with $s_1 \leq^* s_2$. In order to conclude it is enough to consider again the first remark and apply again Lemma 1.

Proposition 3. *Termination, boundedness and coverability are all decidable for ν-PN.*

Proof. In the first place, PDN are strictly well-structured [24], so that the three properties are decidable for them. Prop. 3 states that N terminates iff N' terminates, that is, there is an infinite run in N iff there is an infinite run in N'. More precisely, thanks to the first item, from an infinite run in N we can build an infinite run in N' for every ordering of the initial marking. Conversely, thanks to the second item we can build an infinite run of N from any infinite run in N'.

It also holds that N is bounded iff N' is bounded. If N is unbounded then by König's lemma there is an infinite run in N that contains infinitely many different markings. As above, from that run we can build an infinite run of N' for every ordering of M_0. Moreover, by the first item of Prop. 4, that run contains infinitely many different configurations. Conversely, as above we can obtain an infinite run in N' with infinitely many different configurations. By the second item in Prop. 3, these configurations are of the form $s\ e$, with s ordering of some M, and we can build an infinite run in N. This run has infinitely many different (non α-equivalent) markings, by applying the second item in Prop. 4.

Finally, let us see that we can decide coverability in N by solving a finite number of coverability problems in N. Let us see that a marking M_f can be covered in N iff there is an ordering s_f of M_f such that s'_f can be covered in N'. Since there are *only* $|Id(M)|!$ such orderings we are done. Let us first assume that M_f can be covered in N. Then, there is a run reaching $M_\alpha \sqsupseteq M_f$. By Prop. 3 we can build a run in N' reaching s'. By the third item in Prop. 4 the thesis follows. Conversely, it is enough to apply the second item in Prop. 3 and the first item in Prop. 4.

Let us remark that in the proof of the previous result we decide coverability in ν-PN by solving a number of coverability problems in PDN, which is the factorial of the number of identifiers of the marking we want to cover. Instead, we could have proved directly that ν-PN are strict WSTS by applying the same techniques used in [33, 24, 8], thus obtaining a symbolic backwards reachability algorithm for coverability. For instance, we have already proved that our order $\sqsubseteq_\alpha$ is a wqo. See the technical report [32] for more details.

5.4. Renaming only fresh names

One can think that we have proved decidability of a weak version of the coverability problem, that in which we allow arbitrary renaming of identifiers. For instance, if we consider the net in the left of Fig. 2, and we ask whether the marking M given by $M(p_1) = M(p_2) = \emptyset$, $M(p_3) = \{b\}$ and $M(p_4) = \{a\}$ can be covered, the result would be affirmative, since the marking obtained by exchanging a and b in M (which is α-equivalent to M) is reachable in one step.

However, we can use this apparently weak version to decide a more restricted version of coverability: Let M_0 and M_f be two markings of N, and we want to decide if we can cover M_f from M_0 without allowing renaming of names. Thus, if a name a appears both in M_0 and in M_f we want to reach a marking M such that $M_f \sqsubseteq_\iota M$ with ι satisfying $\iota(a) = a$. Since $R = Id(M_0) \cap Id(M_f)$ contains only a finite number of names, we can add new places in order to ensure the latter. We define the ν-PN $N^* = (P \cup R, T, F)$. For any marking M we define $M^*(p) = M(p)$ if $p \notin R$ and $M^*(r) = \{r\}$ for all $r \in R$. By construction of N^*, places in R are isolated, so that their tokens are never moved or removed.

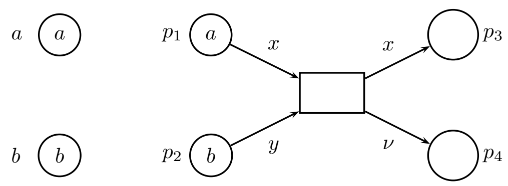

Figure 6: The ν-PN in the left of Fig. 2 extended to decide a restricted version of coverability

In particular, for any reachable M with $M_f \sqsubseteq_\iota M$ it holds $\iota(a) = a$ for every $a \in R$.

Example 4. Let us again consider the example in Fig. 2. Following the previous construction, that can be seen in Fig. 6, we add a place for a and another one for b. When we execute this new net, the reasoning we followed before now fails. In one step we can reach M' with $M'(p_1) = M'(p_2) = \emptyset$, $M'(p_3) = M'(a) = \{a\}$ and $M'(p_4) = M'(b) = \{b\}$. However, thanks to the newly added places, it is not true that M' equals the result of exchanging a and b in M^* (using the notations of the proof of the previous result).

We could ask ourselves whether a lighter version of reachability in which we allow renaming of names, as we are doing with coverability, is decidable. However, this is not true, as the construction in Prop. 1 also works in that case.

6. Complexity of the decision procedures

Now we obtain hardness results for the decision problems shown to be decidable in Prop. 3. We do it by means of a simulation of reset nets by ν-PN. The construction is very similar to the one we used in Sect. 4 to simulate inhibitor nets with ν-PN.

A *reset net* is a tuple $N = (P, T, F, F_r)$, where P and T are disjoint sets of places and transitions, respectively, $F \subseteq (P \times T) \cup (T \times P)$, and $F_r \subseteq P \times T$. Pairs in F_r are reset arcs. For a transition $t \in T$ we write ${}^\bullet t = \{p \in P \mid (p,t) \in F\}$ and ${}^r t = \{p \in P \mid (p,t) \in F_r\}$, and analogously for $t^\bullet$. For simplicity, and without loss of generality, we assume that ${}^r t \cap t^\bullet = \emptyset$ for every $t \in T$, and that each transition can reset at most one place.

A marking of a reset net N is a multiset of places of N. A transition t is enabled in M if $M(p) > 0$ for all $p \in {}^\bullet t$. In that case t can be fired, producing M' defined as[5]

- $M'(p) = (M(p) - F(p,t)) + F(t,p)$ for all $p \notin {}^r t$,
- $M'(p) = 0$ for all $p \in {}^r t$.

[5]Note that we are identifying F with its characteristic function.

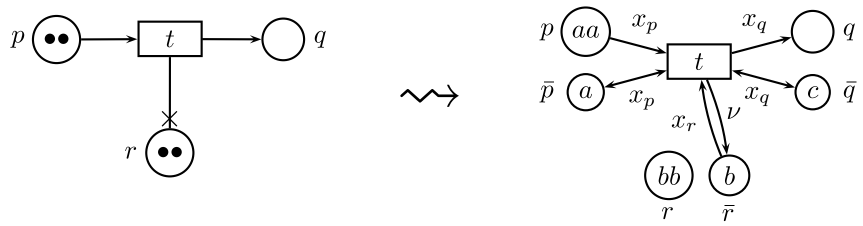

Figure 7: Simulation of reset nets

Proposition 4. *Given a reset net $N = (P, T, F, F_r)$ with initial marking M_0 we can build in polynomial time a ν-PN $N^* = (P \cup \bar{P}, T, F^*)$ with initial marking M_0^* such that:*

- *If M is reachable in N then there is M^* reachable in N^* such that for every $p \in P$ there is $a_p \in Id$ with $M^*(\bar{p}) = \{a_p\}$ and $M^*(p)(a_p) = M(p)$.*
- *If M^* is reachable in N^* then there is M reachable in N and $a_p \in Id$ for every $p \in P$ such that $M^*(\bar{p}) = \{a_p\}$ and $M^*(p)(a_p) = M(p)$.*

In particular,

- *N terminates iff N^* terminates,*
- *Given M we can build M^* such that M can be covered in N iff M^* can be covered in N^*.*

Proof. We consider a different variable x_p for each $p \in P$. Then we define F^* as follows:

- If $(p, t) \in F$ then $F^*(p, t) = F^*(\bar{p}, t) = F^*(t, \bar{p}) = \{x_p\}$,
- If $(t, p) \in F$ then $F^*(t, p) = F^*(\bar{p}, t) = F^*(t, \bar{p}) = \{x_p\}$,
- If $(p, t) \in F_r$ then $F^*(\bar{p}, t) = \{x_p\}$ and $F^*(t, \bar{p}) = \{\nu\}$,
- $F^*(x, y) = \emptyset$ elsewhere.

To define M_0^*, we consider a different identifier a_p for each place p of N. Then, we take $M_0^*(\bar{p}) = \{a_p\}$ and $M_0^*(p) = \{a_p, \overset{M_0(p)}{\ldots}, a_p\}$, for each $p \in P$.

Intuitively, for each place p of N we consider a new place $\bar{p}$ in N^*. The construction of N^* is such that $\bar{p}$ contains a single token at any time. The firing of any transition ensures that the token being used in p coincides with that in $\bar{p}$. Every time a transition resets a place p, the content of $\bar{p}$ is replaced by a fresh token, so that no token remaining in p can be used. Notice that our simulation does not cheat. It can nevertheless introduce some garbage tokens, that once become garbage, always stay like that. Therefore, by construction, the first two items clearly hold. These two items entail that N terminates iff N^* terminates. Finally, given a marking M if is enough to define M^* as done above with M_0 to obtain that M can be covered iff M^* can be covered. Fig. 7 depicts a simple reset net and its simulation.

Proposition 5. *Coverability, boundedness and termination for ν-PN are not primitive recursive.*

Proof. Since coverability and termination are both Ackermann-hard for reset nets [34], the previous construction entails Ackermann-hardness for coverability and termination in ν-PN. This hardness extends to boundedness by means of a very simple reduction: given a ν-PN N it is enough to build N' by adding to N a place in which an ordinary token is put in every firing. Clearly, N terminates iff N' is bounded.

Notice again that the ν-PN built in Prop. 4 does not need to check for inequality, so that the previous hardness result holds even in the subclass of ν-PN in which such checks are not allowed.

7. Weaker forms of boundedness

Let us now discuss weaker forms of boundedness. In the first place, we characterize boundedness (finiteness of the reachability set) in terms of the form of every reachable marking, as is usual in Petri nets.

Lemma 5. *The set of reachable markings of N is finite (up to $\equiv_\alpha$) if and only if there is $n \geq 0$ such that every reachable marking M satisfies $|M(p)| \leq n$ for all $p \in P$.*

Proof. If $Reach(N)$ is finite we can define $s = max\{|Id(M)| \mid M \in Reach\}$ and $k = max\{M(p)(a) \mid M \in Reach,\ p \in P,\ a \in Id(M)\}$. Then, for each reachable M, $|M(p)| = |\sum_{a \in supp(M(p))} M(p)(a)| \leq k \cdot s$. Conversely, if for each n there is a reachable M_n such that $|M_n(p)| > n$ for some p, then there are infinitely many reachable markings.

We will use the previous characterization in order to factorize the property of boundedness. Unlike ordinary P/T nets, that only have one infinite dimension, ν-PN have two different sources of infinity: the number of different identifiers and the number of times each of those appears. Consequently, several different notions of boundedness arise, in one of the dimensions, in the other or in both.

Definition 10. Let N be a ν-PN.

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

Indeed, width and depth-boundedness factorize boundedness.

Proposition 6. *N is bounded iff it is width-bounded and depth-bounded.*

Figure 8: Width-bounded but not depth-bounded ν-PN (left) and vice-versa (right)

Proof. It is enough to consider that $|M(p)| = |\sum_{a \in Id(M)} M(p)(a)| \leq |Id(M)| \cdot max\{M(p)(a) \mid a \in Id\}$. If N is bounded, by Lemma 5 there is $n \in \mathbb{N}$ such that $|M(p)| \leq n$. Then $|\sum_{a \in Id(M)} M(p)(a)| \leq n$, so that every $M(p)(a)$ is at most n. Moreover, since each $M(p)(a) > 0$ for each $a \in supp(M)$, then $|supp(M(p))| \leq n$ for every p, so that $|Id(M)|$ is also bounded. Conversely, let us assume there are n and m such that $|supp(M(p))| \leq n$ and $M(p)(a) \leq m$. From the latter if follows that $max\{M(p)(a) \mid a \in supp(M(p))\} \leq m$. Then, by the previous observation, $|M(p)| \leq n \cdot m$ and the thesis follows.

Thanks to the previous result we know that if a ν-PN is bounded then it is width-bounded and depth-bounded. However, if it is unbounded it could still be the case that it is width-bounded (see left of Fig. 8) or depth-bounded (see right of Fig. 8), though not simultaneously width and depth-bounded.

In [31] we proved decidability of width-boundedness for ν-PN. The proof relies on the results in [14, 15] that establish a framework for forward analysis for WSTS. We do not show the details here, since they are rather involved. However, the hardness results in the previous section can be easily transferred to width-boundedness.

Proposition 7. *Width-boundedness is not primitive recursive.*

Proof. We reduce termination in ν-PN to width-boundedness. Let N be a ν-PN. We obtain N' by adding to N a new place and adding arcs so that every transition puts a fresh name in that new place. Clearly, N terminates iff N' is width-bounded, and we conclude.

Though width and depth-boundedness seem to play a dual role, the proof of decidability of width-boundedness can not be adapted in the case of depth-boundedness. Actually, depth-boundedness turns out to be undecidable, though this fact could be considered to be rather anti intuitive (actually, in the paper [10] there is a wrong decidability proof).

Proposition 8. *Depth-boundedness is undecidable for ν-PN.*

Proof. We reduce boundedness in reset nets to depth-boundedness in ν-PN. Given a reset net N, let us consider the ν-PN N^* built in Prop. 4. It holds that N is bounded iff N^* is depth-bounded. Since boundedness in reset nets is undecidable [11] we can conclude.

We can use this undecidability result in order to prove undecidability of all the "place versions" of the boundedness problems considered. Let us start by defining those problems.

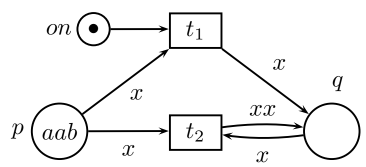

Figure 9: Reducing depth-boundedness of p to boundedness of q

Definition 11. Let p be a place of a ν-PN N with initial marking M_0.

- p is bounded if there is $n \in \mathbb{N}$ such that for every marking M reachable $|M(p)| \leq n$.
- p is width-bounded if there is $n \in \mathbb{N}$ such that for every marking M reachable $|supp(M(p))| \leq n$.
- p is depth-bounded if there is $n \in \mathbb{N}$ such that for every marking M reachable and every $a \in Id$, $M(p)(a) \leq n$.

The *p-boundedness* (*p-width-boundedness*, *p-depth-boundedness*) problem is that of deciding if a given place is bounded (width-bounded, depth-bounded). We prove the following:

Proposition 9. *p-boundedness, p-width-boundedness and p-depth-boundedness are undecidable for ν-PN.*

Proof. Since depth-boundedness is undecidable, clearly p-depth-boundedness is also undecidable. Let us now see that p-boundedness is also undecidable. We reduce p-depth-boundedness to p-boundedness. Let $N = (P, T, F)$ with initial marking M_0 and $p \in P$. Let us define $N' = (P \cup \{on, q\}, T \cup \{t_1, t_2\}, F')$ (see Fig. 9) with initial marking M'_0, where:

- F' extends F as follows:
  - $F'(on, t) = F'(t, on) = \{\epsilon\}$ for all $t \in T$,
  - $F'(on, t_1) = \{\epsilon\}$ and $F'(p, t_1) = F'(t_1, q) = \{x\}$,
  - $F'(p, t_2) = F'(q, t_2) = \{x\}$ and $F'(t_2, q) = \{x, x\}$.
- M'_0 extends M_0 by $M'_0(on) = \{\bullet\}$ and empty in q.

Let us also denote by M' the marking of N' obtained from a marking M of N, as done with M_0 above. The computations of N' start with a computation of N, until t_1 fires from M', moving some name a from p to q. From that point on, only t_2 can fire, each time moving the same name a from p to q, so that there can be at most $M(p)(a)$ tokens in q. Thus, p is depth-bounded in N iff q is bounded in N', and we conclude.

Finally, let us see that p-width-boundedness is also undecidable. We reduce p-boundedness to p-width-boundedness (see Fig. 10). Let N be a ν-PN and p

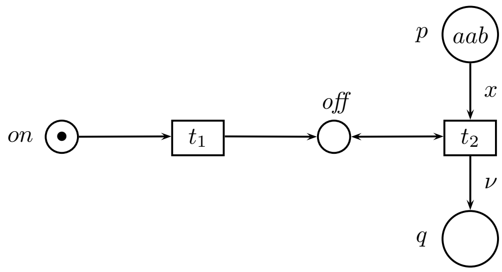

Figure 10: Reducing boundedness of p to width-boundedness of q

a place of N. We add two new transitions, t_1 and t_2, and two control places, *on* and *off*. The new place *on* is a pre/postcondition of every transition of N. Transition t_1 can move a token from *on* to *off*, which is a pre/postcondition of t_2. Transition t_2 can remove any token from p and put a fresh token in a new place q (when *off* is marked). The initial marking of N' extends that of N by putting a black token in *on*. Then, p is bounded in N if and only if q is width-bounded in N', and we conclude.

Let us conclude by remarking that we can define the analogous problems to width and depth-boundedness in PDN, together with their place versions. Then, the construction in Def. 9 and Lemma 3 imply undecidability of depth-boundedness and all the place-boundedness problems.

Corollary 1. *Depth-boundedness, p-boundedness, p-depth-boundedness and p-width-boundedness are undecidable for PDN.*

Thus, since boundedness is decidable for them, PDN have also decidable boundedness and undecidable place-boundedness. Notice that boundedness is already undecidable in Data Nets, where whole-place operations are allowed. Decidability of the width-boundedness problem remains open for PDN, though we conjecture that the same techniques used in [31] could be used for them.

8. Conclusions and Future Work

In this paper we have studied the expressive power of ν-PN, a simple extension of P/T nets with a primitive for fresh name creation. We knew that the expressive power of P/T nets is strictly increased because, unlike for P/T nets, reachability is undecidable. We have presented a simple proof of this result, by showing that ν-PN can weakly simulate nets with inhibitor arcs. This simulation gives a better insight about the expressive power of names and, in particular, can be seen as a starting point for the other negative results we have obtained.

Then we prove that ν-PN can be simulated by Petri Data Nets. Therefore, they are strictly well-structured, so that termination, coverability and boundedness are decidable. Moreover, we tighten the complexity bounds for those problems. More precisely, we obtain non-primitive recursive complexities for them. In [24] such complexity bound was proved for PDN, but in an unordered version of PDN, which in particular cannot create fresh names, the complexity is only non-elementary.

We have also considered two orthogonal notions of boundedness, width- and depth-boundedness, which factorize the latter. We obtained above decidability of boundedness. In previous works we proved decidability of width-boundedness. Now we prove that its complexity is also non primitive recursive. Moreover, we prove that depth-boundedness is undecidable, together with all the related place-boundedness problems.

We plan to use ν-PN as a framework to study concurrent systems in which several (unordered) processes may interact with each other. More precisely, we intend to use ν-PN to describe extensions of Workflow Nets [1], in which some global resources are shared by each instance of the workflow [18, 19]. It is our intention to apply our positive results in this setting, and to use our negative results to obtain theoretical upper bounds in their analysis. We have already obtained some results in this line of work [27].

References

[1] W. M. P. van der Aalst, and K. M. van Hee. Workflow Management: Models, Methods, and Systems. MIT Press (2002)

[2] M. Abadi, and A.D. Gordon. *A Calculus for Cryptographic Protocols: The Spi Calculus.* Fourth ACM Conference on Computer and Communications Security, ACM Press, 1997, 36-47.

[3] P. A. Abdulla, K. Cerans, B. Jonsson, and Y. Tsay. *Algorithmic analysis of programs with well quasi-ordered domains.* Information and Computation 160:109-127. Academic Press Inc., 2000.

[4] P.A. Abdulla, G. Delzanno. *On the coverability problem for constrained multiset rewriting systems*, AVIS 2006 (ETAPS 2006 workshop).

[5] P.A. Abdulla, G. Delzanno, and L. Van Begin. *A classification of the expressive power of well-structured transition systems.* Inf. Comput. 209(3): 248-279 (2011)

[6] P. A. Abdulla, and B. Jonsson. *Verifying Programs with Unreliable Channels.* Information and Computation, 127(2):91-101, 1996.

[7] L. Cardelli, and A.D. Gordon. *Mobile Ambients.* 1st Int. Conf. on Foundations of Software Sciences and Computation Structures, FOSSACS'98. LNCS 1387, pp. 140-155. Springer, 1998.

[8] G. Delzanno. Constraint-based automatic verification of abstract models of multithreaded programs. Theory and Practice of Logic Programming 7(1-2):67-91 (2007)

[9] J. Desel and W. Reisig. *Place/transition petri nets.* Lectures on Petri Nets I: Basic Models, LNCS 1491, pp.122–173. Springer, 1998.

[10] R. Dietze, M. Kudlek and O. Kummer. Decidability Problems of a Basic Class of Object Nets. *Fundamenta Informaticae* 79. IOS Press (2007) 295-302.

[11] C. Dufourd, A. Finkel and Ph. Schnoebelen. *Reset nets between decidability and undecidability.* 25th Int. Col. on Automata, Languages and Programming, ICALP'98. LNCS 1443, pp. 103-115. Springer, 1998.

[12] C. Dufourd, P. Jančar and Ph. Schnoebelen. *Boundedness of Reset P/T Nets.* 26th Int. Col. on Automata, Languages and Programming, ICALP'99. LNCS 1644, pp. 301-310. Springer, 1999.

[13] J. Esparza and M. Nielsen. *Decidability issues for Petri Nets - a survey.* Bulletin of the EATCS 52:244-262 (1994).

[14] A. Finkel and J.Goubault-Larrecq. Forward analysis for WSTS, Part I: Completions. *In Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science, STACS'09* (2009) 433-444.

[15] A. Finkel and J.Goubault-Larrecq. Forward analysis for WSTS, Part II: Complete WSTS. *In 36th International Colloquium on Automata, Languages and Programming, ICALP'09.* LNCS 5556. Springer (2009) 188-199.

[16] A.Finkel, and P.Schnoebelen. *Well-Structured Transition Systems Everywhere!* Theoretical Computer Science 256(1-2):63-92 (2001).

[17] A. Gordon. *Notes on Nominal Calculi for Security and Mobility.* Foundations of Security Analysis and Design, FOSAD'00. LNCS 2171, pp. 262-330. Springer, 2001.

[18] K. van Hee, A. Serebrenik, N. Sidorova and M. Voorhoeve. Soundness of Resource-Constrained Workflow Nets. 26th Int. Conference on Application and Theory of Petri Nets, ICATPN 2005, LNCS vol. 3536, pp. 250-267. Springer, 2005.

[19] G. Juhás, I. Kazlov, and A. Juhásová. Instance Deadlock: A Mistery behind Frozen Programs. PETRI NETS 2010, LNCS vol. 6128, pp. 1-17. Springer, 2010.

[20] M. Köhler. Reachable markings of object Petri nets. Fundamenta Informaticae 79(3-4):401-413 (2007)

[21] M. Köhler, and F. Heitmann. On the expressiveness of communication channels for object nets. Fundamenta Informaticae 93(13):205-219 (2009)

[22] M. Köhler, and H. Rölke. Properties of Object Petri Nets. 25th Int. Conf. on Petri Nets, ICATPN'04. LNCS 3099, pp. 278-297. Springer, 2004.

[23] O. Kummer. *Undecidability in object-oriented Petri nets.* Petri Net Newsletter, 59:18-23, 2000.

[24] R. Lazic, T.C. Newcomb, J. Ouaknine, A.W. Roscoe and J. Worrell. *Nets with Tokens Which Carry Data.* Fundamenta Informaticae 88 (3):251-274. IOS Press, 2008.

[25] I. Lomazova. *Nested petri nets - a formalism for specification and verification of multi-agent distributed systems.* Fundamenta Informaticae 43(1-4):195-214. IOS Press, 2000.

[26] I. Lomazova, and Ph. Schnoebelen. *Some Decidability Results for Nested Petri Nets.* 3rd Int. Andrei Ershov Memorial Conf. on Perspectives of System Informatics, PSI'99. LNCS vol. 1755, pp. 208-220. Springer, 2000.

[27] M. Martos-Salgado, and F. Rosa-Velardo. *Dynamic Soundness in Resource-Constrained Workflow Nets.* In 31st IFIP International Conference on FORmal TEchniques for Networked and Distributed Systems, FORTE'11. LNCS (to appear)

[28] E.C. Milner. Basic WQO- and BQO-theory. In I. Rival (ed.). Graphs and Order. The Role of Graphs in the Theory of Ordered Sets and Its Applications. D. Reidel Publishing Co.. pp. 487-502.

[29] R. Milner, J. Parrow, and D. Walker. *A Calculus of Mobile Processes, I.* Information and Computation 100(1): 1-40 (1992).

[30] F. Rosa-Velardo, D. de Frutos-Escrig. Name Creation vs. Replication in Petri Net Systems. Fundamenta Informaticae 88(3):329-356. IOS Press, 2008.

[31] F. Rosa-Velardo, D. de Frutos-Escrig. Forward analysis for Petri nets with name creation. 31st Int. Conf. on Applications and Theory of Petri nets, PETRI NETS 2010. LNCS vol. 6128, pp. 185-205. Springer, 2010.

[32] F. Rosa-Velardo, D. de Frutos-Escrig. Decision Problems for Petri Nets with names. CoRR abs/1011.3964 (2010) `http://arxiv.org/abs/1011.3964`

[33] F. Rosa-Velardo, D. de Frutos-Escrig, and O. Marroquín-Alonso. *On the expressiveness of Mobile Synchronizing Petri Nets.* SecCo'05. ENTCS 180(1):77-94. Elsevier, 2007.

[34] P Schnoebelen. Revisiting Ackermann-Hardness for Lossy Counter Machines and Reset Petri Nets. In 355h Int. Symposium on Mathematical Foundations of Computer Science, MCFS'10. LNCS vol. 6281, pp. 616-628. Springer (2010)

[35] R. Valk. Petri Nets as Token Objects - An Introduction to Elementary Object Nets. 19th Int. Conf. on Applications and Theory of Petri Nets, ICATPN'98, LNCS 1420, pp. 1-25. Springer, 1998.