Expressiveness of ν-*lsPN*

María Martos-Salgado, Fernando Rosa-Velardo*

Technical Report 10/13
Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid
E-mail: `mrmartos@estumail.ucm.es, fernandorosa@sip.ucm.es`

Abstract. Locally synchronous processes are dynamic networks of infinite-state timed processes in which each process carries a single real valued clock. Here, we prove that locally synchronous processes are strictly more expressive than Timed-Arc Petri nets, using coverability languages to compare classes of WSTS. Therefore, we conclude that up to our knowledge, locally synchronous processes is the most expressive class of WSTS among those whose relative expressive power has been studied.

1 Introduction

Petri nets are one of the best known models for concurrent and distributed systems. They have been extended with discrete or continuous time in many works [17, 15, 18, 16, 5]. In [8] an exhaustive comparison of these models is done, and the class of Petri nets with time relative to arcs is proved to be the most expressive one. Among them, in Timed-Arc Petri Nets (*TdPN*) [5] tokens are endowed with a real-valued clock, that can be dynamically created.

Under the so called counting abstraction, each token in a place s of a Petri net represents a process in state s. Hence, Petri nets can be seen as networks (or products) of finite-state automata. With this intuition in mind, *TdPN* can be seen as dynamic networks of finite-state process, each carrying a real-valued clock. Therefore, *TdPN* extend Timed Automata [6] in that they can be used to verify parameterized systems of finite-state timed processes.

In [14] we extend the work in [5] by allowing each process to be infinite-state in turn. Hence, our model manages infinitely-many timed processes, each of which is infinite-state (a potentially unbounded Petri net). In this way we can for example easily model dynamic networks of processes communicating asynchronously via unbounded (unordered) buffers or sharing some global resources, that can be potentially unbounded. In particular, our model can serve as a basis for the parameterized verification of such processes. As a starting point for our work we consider an untimed model we have developed in previous works [19, 20], called ν-*PN*. In ν-*PN* tokens are names, that can be created fresh and matched with other names. Names can be understood as the identifier of processes that can be spawned and synchronize with each other.

* Authors supported by the Spanish projects STRONGSOFT TIN2012-39391-C04-04, DESAFIOS10 TIN2009-14599-C03-01 and PROMETIDOS S2009/TIC-1465.

Then, we consider that each process has a single real-valued clock (as in [5] under the counting abstraction). We say these are locally-synchronous processes, and call them *locally synchronous* ν-*PN* (ν-*lsPN*). Each transition specifies which are the possible ages of the processes involved, and how this age is updated. We allow read-only constraints, so that the age of some processes may not change when firing a transition.

In [14] we successfully apply the theory of regions of [3]. More precisely, working with regions we prove that ν-*lsPN* belong to the class of Well-Structured Transition Systems [10, 1], for which coverability is decidable. This proves that control-state reachability (which can be reduced to coverability) is decidable for them. Moreover, safety properties can be reduced to control-state reachability with standard techniques.

Here, we compare the expressiveness of ν-*lsPN* with other well-structured models. In [11] coverability languages (those obtained with coverability as acceptance condition) are proposed as a measure to compare the expressiveness of WSTS. In [2, 11, 7] Petri nets (*PN*), Petri nets with transfers and resets (*AWN*), ν-*PN* and Data Nets (DN), an extension of ν-*PN* with ordered data, are compared, proving the following strict relations: $PN \prec AWN \prec \nu\text{-}PN \prec DN$. Moreover, *DN* and *TdPN* are proved to be equivalent in [7]. We complete this picture by proving that $TdPN \prec \nu\text{-}lsPN$. Therefore, we prove that ν-*lsPN* is the most expressive model within the WSTS class, up to our knowledge, out of those whose relative expressive power has been studied.[1]

Outline Section 2 gives notations and results we use throughout the paper. In Section 3 we define ν-*lsPN* and in Section 4 we study its expressiveness. Finally, in Section 5 we present our conclusions.

2 Preliminaries

Let $\mathbb{N} = \{0, 1, 2, ...\}$ and for $n \in \mathbb{N}$ let $n^+ = \{1, ..., n\}$ and $n^* = \{0, ..., n\}$. $(X, \leq)$ is a *partial order* (po) if $\leq$ is a reflexive, transitive and antisymmetric binary relation on X.

Multisets. A (finite) *multiset* m over X is a mapping $m : X \to \mathbb{N}$ such that $\{x \in X \mid m(x) > 0\}$ is finite. We denote by $X^{\oplus}$ the set of multisets over X. For $m_1, m_2 \in X^{\oplus}$ we define $m_1 + m_2 \in X^{\oplus}$ by $(m_1 + m_2)(x) = m_1(x) + m_2(x)$ and $m_1 \subseteq m_2$ if $m_1(x) \leq m_2(x)$ for every $x \in X$. When $m_1 \subseteq m_2$ we can define $m_2 - m_1 \in X^{\oplus}$ by $(m_2 - m_1)(x) = m_2(x) - m_1(x)$. We denote by $\emptyset$ the empty multiset, that is, $\emptyset(a) = 0$ for every $a \in A$, and $|m| = \sum_{x \in X} m(x)$. We use set notation for multisets with repetitions to account for multiplicities.

Transition systems. A *transition system* is a tuple $\mathcal{S} = \langle X, \to, x_0 \rangle$ where X is the set of states, $x_0 \in X$ is the initial state and $\to \subseteq X \times X$ is the transition relation. We write $x \to x'$ instead of $(x, x') \in \to$ and we denote by $\to^*$ the

[1] The expressive powers of WSTS based on trees or graphs have not been compared with others using the techniques in [11], up to our knowledge.

reflexive and transitive closure of $\rightarrow$. We say $A \subseteq X$ is *reachable* if $x_0 \rightarrow^* x$ for some $x \in A$. If X is a po, we can define the *coverability problem*, that of deciding, given U upward closed, whether U is reachable.

ν-Petri Nets. We fix infinite sets *Id* of names, *Var* of variables and a subset of special variables $\Upsilon \subset Var$ for fresh name creation. A ν-*PN* [20] is a tuple $N = \langle P, T, F, H \rangle$, where P and T are finite disjoint sets, and $F, H : T \rightarrow (P \times Var)^{\oplus}$ are the input and output functions, respectively. We say that $x \in Var(t)$ iff there is $p \in P$ with $(p, x) \in F(t) + H(t)$. A *marking* is a finite multiset over $P \times Id$. A *mode* is an injection $\sigma : Var(t) \rightarrow Id$. Modes are extended homeomorphically to $(P \times Var(t))^{\oplus}$. A transition t is *enabled* with mode σ for a marking M if $\sigma(F(t)) \subseteq M$ and for every $\nu \in \Upsilon$, $(p, \sigma(\nu)) \notin M$ for any p. The last condition is used to create new names, not in the current marking. Then we have $M \xrightarrow{t} M'$, where $M' = (M - \sigma(F(t)) + \sigma(H(t))$ and $M \rightarrow M'$ if $M \xrightarrow{t} M'$ for some $t \in T$.

3 Locally synchronous ν-*PN*

Now we define the class of locally synchronous ν-*PN* (ν-*lsPN*). In ν-*lsPN* each instance has a single clock. Moreover, we will allow read-only contraints.

Definition 1 (Locally synchronous ν-*PN*). *A* locally synchronous ν-*PN* *(ν-lsPN) is a tuple $N = \langle P, T, F, H, \mathcal{G} \rangle$, where:*

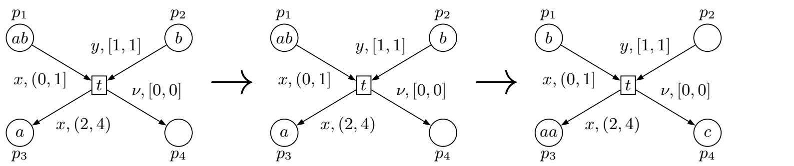
- *P and T are finite disjoint sets,*
- *for $t \in T$, $F_t, H_t : Var \rightarrow P^{\oplus}$ are the input and output functions of t,*
- *for $t \in T$, $\mathcal{G}_t : Var \rightarrow \mathcal{I} \times (\mathcal{I} \cup \{ro\})$ is the time constraints function of t.*

Definition 2 (Markings). *A* marking *M of a ν-lsPN is an expression of the form $a_1{:}(m_1, r_1), ..., a_n{:}(m_n, r_n)$, where $Id(M) = \{a_1, ..., a_n\} \subset Id$ are pairwise different names, and for each $i \in n^+$, $\emptyset \neq m_i \in P^{\oplus}$ and $r_i \in \mathbb{R}_{\geq 0}$.*

We treat markings of ν-*lsPN* as multisets over elements of the form $a{:}(m, r)$, which we call *instances*. Hence, $a : (m, r)$ is an instance with name a, tokens according to m, and age r. We assume that each m_i in each instance is not empty. We use M, M',... to range over markings. We say a marking M *marks* $p \in P$ if there is $a{:}(m, r) \in M$ such that $p \in m$.

Definition 3 (Time delay). *Given $M = a_1{:}(m_1, r_1), ..., a_n{:}(m_n, r_n)$ and $d \in \mathbb{R}_{\geq 0}$, we write M^{+d} to denote the marking $a_1{:}(m_1, r_1 + d), ..., a_n{:}(m_n, r_n + d)$, in which the age of every instance has increased by d. We write $M \xrightarrow{d} M^{+d}$.*

Now we define the firing of transitions, for which we need the following notations. We denote by $\mathcal{G}_t^1(x)$ and $\mathcal{G}_t^2(x)$ the first and second component of $\mathcal{G}_t(x)$, respectively. Intuitively, for a transition to fire the instance corresponding to x must have an age in $\mathcal{G}_t^1(x)$. This age is set to any value in $\mathcal{G}_t^2(x)$, except when x is read-only (when $\mathcal{G}_t^2(x) = ro$), in which case its age does not change. If $\nu \in \Upsilon$ we assume $\mathcal{G}_t^2(\nu) \neq ro$. For each $t \in T$ we define $Var(t) = \{x \in Var \mid F_t(x) + H_t(x) \neq$



$M_1 = a{:}(\{p_1p_3\}, 0),$ $b{:}(\{p_1p_2\}, 0.5)$ $\quad M_2 = a{:}(\{p_1p_3\}, 0.5),$ $b{:}(\{p_1p_2\}, 1)$ $\quad M_3 = a{:}(\{p_3p_3\}, 3), b{:}(\{p_1\}, 1),$ $c{:}(\{p_4\}, 0)$

Fig. 1. Firing of a transition in a *ν-lsPN*.

$\emptyset\}$, which is assumed to be finite, and we split it into $nfVar(t) = Var(t) \setminus \Upsilon$ and $fVar(t) = Var(t) \cap \Upsilon$. We say M' is an $\emptyset$-*expansion* of a marking M (or M is the $\emptyset$-*contraction* of M') if M' is obtained by adding instances $a{:}(\emptyset, r)$ to M.

Definition 4 (Firing of transitions). *Let $t \in T$ with $nfVar(t) = \{x_1, ..., x_n\}$ and $fVar(t) = \{\nu_1, ..., \nu_k\}$. We say t is* enabled *at marking M if:*

- $M = a_1 : (m_1, r_1), ..., a_n : (m_n, r_n) + \overline{M}$,
- *for each $i \in n^+$, $F_t(x_i) \subseteq m_i$ and $r_i \in \mathcal{G}_t^1(x_i)$.*

Then, t can be fired, *and taking*

- $\{b_1, ..., b_k\}$ *pairwise different names not in $Id(M)$,*
- $m'_i = (m_i - F_t(x_i)) + H_t(x_i)$ *for all $i \in n^+$,*
- $m''_j = H_t(\nu_j)$ *for all $j \in k^+$,*
- $r'_i = r_i$ *if $\mathcal{G}_t^2(x_i) = ro$, or any value in $\mathcal{G}_t^2(x_i)$, otherwise, for all $i \in n^+$,*
- r''_j *any value in $\mathcal{G}_t^2(\nu_j)$, for all $j \in k^+$,*

we can reach M', denoted by $M \xrightarrow{t} M'$, where M' is the $\emptyset$-contraction of

$$a_1{:}(m'_1, r'_1), ..., a_n{:}(m'_n, r'_n), b_1{:}(m''_1, r''_1), ..., b_k{:}(m''_k, r''_k) + \overline{M}$$

We write $M \to M'$ if $M \xrightarrow{t} M'$ for some $t \in T$ or $M \xrightarrow{d} M'$ for some $d \in \mathbb{R}_{\geq 0}$, and we implicitly assume an initial marking M_0, thus obtaining the transition system induced by N. Again, the semantics of *ν-lsPN* is a weak semantics, since time elapsing may disable transitions. The *control-state reachability problem* for *ν-lsPN* is defined as reachability of the set of states that mark a given place.

Example 1. Fig. 1 depicts a *ν-lsPN* with three different markings. In the first marking the transition t is not fireable, because no instance with an age in $[1, 1]$ has a token in place p_2. However, after waiting 0.5 units of time, the marking M_2 is reached, and t becomes enabled. Then, we can fire t reaching, for example, the marking M_3 in the figure.

In [14] we use the theory of regions to obtain a finitary transition system over a countable domain, which will be a WSTS, so that we can solve the control-state reachability problem by reducing it to a coverability problem.

Proposition 1. *ν-lsPN are WSTS.*

4 Expressiveness result

In this section we prove that *ν-lsPN* are strictly more expressive than *TdPN*. We compare classes of WSTS by comparing the families of coverability languages they accept, as advocated for instance in [11, 2].

A labeled WSTS is a WSTS in which each transition is endowed with a label taken from $\Sigma \cup \{\epsilon\}$, where Σ is a finite alphabet. A label ϵ denotes a silent transition.be ϵ). We assume that (labeled) WSTS are endowed with a final state. If x_0 and x_f are the initial and final states of $\mathcal{S}$, we can define the coverability language of $\mathcal{S}$ by $L(\mathcal{S}) = \{u \in \Sigma^{\circledast} \mid x_0 \xrightarrow{u} x,\ x \geq x_f\}$.

For two classes of WSTS, $\mathbf{S}_1$ and $\mathbf{S}_2$, we write $\mathbf{S}_1 \preceq \mathbf{S}_2$ whenever for every $\mathcal{S}_1 \in \mathbf{S}_1$ there is $\mathcal{S}_2 \in \mathbf{S}_2$ such that $L(\mathcal{S}_1) = L(\mathcal{S}_2)$. We write $\mathbf{S}_1 \simeq \mathbf{S}_2$ when $\mathbf{S}_1 \preceq \mathbf{S}_2$ and $\mathbf{S}_2 \preceq \mathbf{S}_1$, and we write $\mathbf{S}_1 \prec \mathbf{S}_2$ if $\mathbf{S}_1 \preceq \mathbf{S}_2$ and $\mathbf{S}_2 \not\preceq \mathbf{S}_1$. In [2, 11, 7] the following relations between classes of WSTS are proved:

$$PN \prec AWN \prec \nu\text{-}PN \prec DN \simeq TdPN$$

We complete this picture by proving that $TdPN \prec \nu\text{-}lsPN$. We first prove that *ν-lsPN* extend *TdPN* by simulating a *TdPN* by means of a *ν-lsPN*.

Proposition 2. $TdPN \preceq \nu\text{-}lsPN$

Proof (Sketch). We simulate a token in p with age x by an instance with a single token in p, and with age x. Each transition is simulated by a transition (having the same label) that (i) removes instances/tokens with clocks with the proper values and (ii) creates *fresh* instances, again with clocks with the proper values. If the initial marking of the *TdPN* is $p_1(x_1), ..., p_n(x_n)$ then we consider $a_1 : (p_1, x_1), ..., a_n : (p_n, x_n)$ (for arbitrary names $a_1, ..., a_n$) as initial marking of the *ν-lsPN*(and analogously for the final marking). □

In [7] a framework for the strict comparison of WSTS is developed. This framework is based on two concepts: *reflections* and *witness* languages. A mapping $\varphi : X \to Y$ is a reflection if $\varphi(x) \leq \varphi(x')$ implies $x \leq x'$ for all $x, x' \in X$. A reflection is an *isomorphism* if it is bijective and $x \leq x'$ implies $\varphi(x) \leq \varphi(x')$. We write $X \sqsubseteq_{refl} Y$ if there is a reflection from X to Y. We extend the relation $\sqsubseteq_{refl}$ to classes of wpo by $\mathbf{X} \sqsubseteq_{refl} \mathbf{X}'$ if for any $X \in \mathbf{X}$, there exists $X' \in \mathbf{X}'$ such that $X \sqsubseteq_{refl} X'$.

Witness languages represent the capability of a WSTS to recognize a state space. They are useful to prove strict relations between classes of WSTS because they can be proven *not* to be recognizable by some class of WSTS.

Given an alphabet $\Sigma = \{\mathbf{a}_1, ..., \mathbf{a}_k\}$, we consider a disjoint copy $\overline{\Sigma} = \{\overline{\mathbf{a}_1}, ..., \overline{\mathbf{a}_k}\}$. This notation is extended to words and languages, as expected. A *Σ-representation* of a wpo X is any surjective partial function $\gamma : \Sigma^{\circledast} \to X$. Intuitively, every $u \in \Sigma^{\circledast}$ with $\gamma(u) = x$ is a possible encoding or representation of $x \in X$. We denote by $dom(\gamma)$ the domain of γ. For a Σ-representation γ of X, we define $L_\gamma = \{u\overline{v} \mid u, v \in dom(\gamma) \text{ and } \gamma(v) \leq \gamma(u)\}$, and we say L_γ is a witness of

X.[2] The fact that a WSTS can recognize such L_γ *witnesses* that it is capable of representing the structure of X: it is capable of accepting all words starting with some u (representing some state $\gamma(u)$), followed by some v that represents $\gamma(v) \leq \gamma(u)$. In particular, it must be able to accept $u\overline{u}$ for any $u \in dom(\gamma)$.

In order to be able to apply the general framework to two classes of WSTS we must prove that both classes are *self-witnessing*. Intuitively, a class of WSTS $\mathbf{S}$ is self-witnessing if it can accept encodings (over some alphabet) of their state spaces. Formally, if $\mathbf{X}$ is a class of wpos and $\mathbf{S}$ is a class of WSTS whose state spaces are included in $\mathbf{X}$, $(\mathbf{X}, \mathbf{S})$ is *self-witnessing* if, for all $X \in \mathbf{X}$, there exists $\mathcal{S} \in \mathbf{S}$ that recognizes a witness of X.

Proposition 3 ([7]). *Let* $(\mathbf{X}, \mathbf{S})$ *and* $(\mathbf{X}', \mathbf{S}')$ *be self-witnessing WSTS classes. If* $\mathbf{S} \preceq \mathbf{S}'$ *then* $\mathbf{X} \sqsubseteq_{refl} \mathbf{X}'$.

In the rest of the section for any model $\mathcal{M}$ we denote by $\mathbf{X}_{\mathcal{M}}$ the set of state spaces used in $\mathcal{M}$. In [7] it is proven that *DN* (and therefore *TdPN*) are self-witnessing. Thus, in order to apply Prop. 3 to prove ν-*lsPN* $\not\preceq$ *TdPN* we only need to see that ν-*lsPN* is self-witnessing and that $\mathbf{X}_{\nu\text{-}lsPN} \not\sqsubseteq_{refl} \mathbf{X}_{TdPN}$.

To prove that ν-*lsPN* is self-witnessing, given the set X of regions with n places and having bound max, we need to find a ν-*lsPN* which accept a witness of X.

Proposition 4. ν-*lsPN is self-witnessing.*

Proof. As seen in the previous section, we can use the set of regions as state space of ν-*lsPN*. Then, given a state space X we have to prove that there is a ν-*lsPN* N that accepts a witness of X. If P is the set of places in X, we can write X as $X^{\oplus}_{\max^*} \times (X^{\oplus}_{(\max-1)^*})^{\circledast} \times X^{\oplus}_{\{\max+1\}}$, where for every $I \subseteq (\max+1)^*$, $X_I = P^{\oplus} \times I$.

Take $\Sigma = P \cup (\max+1)^* \cup \{*, \#, \$\}$. We define auxiliary functions $\gamma^I_1 : \Sigma^{\circledast} \to X_I$, $\gamma^I_2 : \Sigma^{\circledast} \to X^{\oplus}_I$ and $\gamma^I_3 : \Sigma^{\circledast} \to (X^{\oplus}_I)^{\circledast}$ as follows: $\gamma^I_1(p_1...p_nk) = (\{p_1, ..., p_n\}, k)$, with $p_i \in P$ and $k \in I$; $\gamma^I_2(u_1\#...\#u_n) = \{\gamma^I_1(u_1), ..., \gamma^I_1(u_n)\}$, with $u_i \in dom(\gamma^I_1)$ for every i; and $\gamma^I_3(v_1 * ... * v_n) = \gamma^I_2(v_1)...\gamma^I_2(v_n)$, where $v_i \in dom(\gamma^I_2)$ for every i. Finally, let $\gamma : \Sigma^{\circledast} \to X$ be defined as $\gamma(u\&v\&w) = (\gamma_2^{max^*}(u), \gamma_3^{(max-1)^*}(v), \gamma_2^{\{max+1\}}(w))$, where u, v and w are in the domain of the corresponding mappings.

Clearly, γ is a Σ-representation of X. Let us now build a ν-*lsPN* N that accepts L_γ (which is a witness of X).

N operates in two phases: the first phase generates u with $\gamma(u) = R = A_0 * A_1 * ... * A_n * A_\infty$, and the second one recognizes any $\overline{v}$ with $\gamma(v) \leq R$. In turn, each of the phases has three consecutive sub-phases, dealing with A_0, $A_1 * ... * A_n$ and A_∞, respectively. We use control places to move from one (sub)phase to the next (with transitions labeled by & or $\overline{\&}$). In order to differentiate between phases, we say that we generate words in the first one, but we recognize them in the second.

[2] Actually, the definition in [7] is slightly more general.

We explain the generation of $A_1 * ... * A_n$ (the other phases are simpler). Let $A_i = \{(m_1^i, k_1^i), ..., (m_{n_i}^i, k_{n_i}^i)\}$. We use a different name to represent each instance. Moreover, instances in the same A_i have the same age. We use a place *now* that holds the name (with age 0) of the instance currently being generated. For a given i, we start by firing transitions of the form t_p, labeled by p, that copy the name in *now* to a place p, followed by the firing of some t_k with $k \in (\max -1)^*$, labeled by k, which copies the name in *now* to a place p_k. Therefore, a word u with $\gamma_1^{(\max -1)^*}(u) = (m_1^i, k_1^i)$ can be produced. Next, the name in *now* is moved to a place *all*, and replaced by a fresh name, with age 0 (transition $t_\#$, labeled by $\#$). These actions can be repeated to generate (the encoding of) any element in $X^{\oplus}_{(\max -1)^*}$. Notice that they all demand that the instance involved has age 0. At any point, instead of firing $t_\#$ we can fire t_* (labeled by $*$). This transition has the same effect as $t_\#$, but it is only enabled if the instance in *now* has a non-null age, so that some time must elapse. Hence, we start accepting the instances with a higher fractional part, in A_{i+1}.

After this phase, there is any number of pairwise different names (each representing an instance) in *all*, some of which have the same age (those instances with an age with the same fractional part). Moreover, for any name a in *all*, a belongs to some of the places in P, and exactly to one place p_k. The transitions in this second phase demand that the age of the instances involved is exactly 1. Moreover, they are all labeled with symbols in $\overline{\Sigma}$.

This phase starts by taking any name in *all* and putting it back to *now*. Then, transitions of the form $\bar{t}_p$ (labeled by $\overline{p}$) can be fired, each consuming a name from p matching the name in *now*. At any point, a transition of the form $\bar{t}_k$, labeled by $\overline{k}$ (with $k \in (\max -1)^*$) can be fired, and consumes from p_k a name matching the one in *now*. Thus, if the current name represented an instance (m, k) then (an encoding of) any (m', k) with $m' \leq m$ can now be recognized. Then, the name in *now* can be replaced by a name taken from *all* (transition $\bar{t}_\#$ labeled by $\overline{\#}$) in order to recognize the next instance.

At any point, time can elapse, so that another instance in *now* reaches age 1. Then, $\bar{t}_*$ can be fired, labeled by $\overline{*}$, with the same effect as $\bar{t}_\#$. Notice that when time elapses, all the names with age greater than 1 are lost (the encodings of the instances they represent cannot be recognized). This is consistent with the fact that we must recognize (the encoding of) a state which is *less or equal* than the one we generated. Notice also that even in the first phase, names with ages older than 1 become garbage. However, it is possible to generate all the names in the first phase with an age smaller than 1, so that the same state can be recognized.

Even though the order between instances is not preserved within each A_i (this is not demanded by the order in $X^{\oplus}_{(\max -1)^*}$), this order *is* preserved between different A_i's, because older instances reach the age of 1 before. To conclude, we consider as final marking the one with a token in the control-state marked in the second phase (the recognizing one). □

In order to prove $\mathbf{X}_{\nu\text{-}lsPN} \not\sqsubseteq_{refl} \mathbf{X}_{TdPN}$ we will use ordinal theory (see Prop. 5 below). Let us explain the needed concepts about ordinals. For more details see [7]. Each ordinal α is equal to the set of ordinals $\{\beta \mid \beta < \alpha\}$ below it, and the class of ordinals is totally ordered by inclusion. Every total well order $(X, \leq_X)$ is isomorphic to a unique ordinal $ot(X, \leq_X)$, called the *order type* of X.

In the context of ordinals, we define $0 = \emptyset$, $n = \{0, ..., n-1\}$ and $\omega = \mathbb{N}$, ordered by the usual order. Moreover, given α and α' ordinals, we define $\alpha + \alpha'$ as the order type of $(\{0\} \times \alpha) \cup (\{1\} \times \alpha')$ ordered by $\leq_{lex}$, the lexicographic order. In the same way, $\alpha * \alpha'$ is defined as the order type of $\alpha' \times \alpha$ ordered by $\leq_{lex}$.The definitions of $+$ and $*$ coincide with the usual operations on $\mathbb{N}$ for ordinals below ω, and we have $\alpha + \overset{k}{\cdots} + \alpha = \alpha * k$. Let α^β be the order type of the set of functions from β to α ordered by $\leq_{lex}$ defined by:

$$f <_{lex} g \iff \exists x \in \beta. \begin{cases} f(x) < g(x) \\ \forall y < x.\ f(y) = g(y) \end{cases}$$

The ordinals below ε_0 (those that can be bounded by a tower $\omega^{\omega^{\cdot^{\cdot^{\cdot^{\omega}}}}}$) can be represented by the hierarchy of ordinals in Cantor Normal Form (CNF), that is recursively given by the following rules:

$C_0 = \{0\}$.
$C_{n+1} = \{\omega^{\alpha_1} + ... + \omega^{\alpha_p} \mid p \in \mathbb{N}, \alpha_1, ..., \alpha_p \in C_n \text{ and } \alpha_1 \geq ... \geq \alpha_p\}$ ordered by $\omega^{\alpha_1} + \cdots + \omega^{\alpha_p} \leq \omega^{\alpha'_1} + \cdots + \omega^{\alpha'_q}$ iff $(\alpha_1, \ldots, \alpha_p) \leq_{lex} (\alpha'_1, \ldots, \alpha'_q)$

Each ordinal below ε_0 has a unique CNF. If $\alpha = \omega^{\beta_1} + \cdots + \omega^{\beta_n}$, we denote by $Cantor(\alpha)$ the multiset $\{\beta_1, \ldots, \beta_n\}$.

A *linearization* of a po $\leq_X$ is a total order $\leq'_X$ on X such that $x \leq_X y \implies x \leq'_X y$. A linearization of a wpo is a well total order, hence isomorphic to an ordinal.

Let $(X, \leq_X)$ be a wpo. The *maximal order type* (shortly: order type) of $(X, \leq_X)$ is $ot(X, \leq_X) = sup\ \{ot(X, \leq'_X) \mid \leq'_X \text{ linearization of } \leq_X\}$.

The existence of the *sup* comes from ordinal theory.

We need the following definitions of *natural addition* (denoted $\oplus$) and *natural multiplication* (denoted $\otimes$) on ordinals, to characterize the order types of $X \uplus Y$ and $X \times Y$:

$Cantor(\alpha \oplus \alpha') = Cantor(\alpha) + Cantor(\alpha')$
$Cantor(\alpha \otimes \alpha') = \{\beta \oplus \beta' \mid \beta \in Cantor(\alpha), \beta' \in Cantor(\alpha')\}$

Let us now prove that $\mathbf{X}_{\nu\text{-}lsPN} \not\sqsubseteq_{refl} \mathbf{X}_{TdPN}$. The following result states that we can do it comparing their *ordinal types* (ot) [9, 21].

Proposition 5 ([22, 7]). *Let X and Y be two wpos. If $X \sqsubseteq_{refl} Y$ then $ot(X) \leq ot(Y)$.*

Using [9, 21, 22], we can compute the order type of products, domains of finite words or finite multisets.

Lemma 1. $ot((\mathbb{N}^m)^{\circledast}) = \omega^{\omega^{\omega^m}}$ *and* $ot((\mathbb{N}^{\oplus})^{\circledast}) = \omega^{\omega^{\omega^\omega}}$

Proof. Let X be a wpo with $\omega \leq ot(X) < \varepsilon_0$, that is, bounded by a tower $\omega^{\omega^{\cdot^{\cdot^{\cdot^{\omega}}}}}$. In [9, 21] it is proved that $ot(X^{\circledast}) = \omega^{\omega^{ot(X)}}$, and [22] proves that $ot(X^{\oplus}) =$

$\omega^{ot(X)}$. Moreover, $ot(\mathbb{N}^m) = \omega^m$ for $m \geq 1$. This allows us to compute the ordinals in the lemma. □

Proposition 6. $\mathbf{X}_{\nu\text{-}lsPN} \not\sqsubseteq_{refl} \mathbf{X}_{TdPN}$

Proof. We have to see that there is $X \in \mathbf{X}_{\nu\text{-}lsPN}$ such that for any $X' \in \mathbf{X}_{TdPN}$ we have $X \not\sqsubseteq_{refl} X'$. Since $TdPN \simeq DN$, we can work with DN. The state space of a DN with set of places P is $(\mathbb{N}^{|P|})^{\circledast}$ [13]. Let $P = \{p\}$ and $\max = 1$, and take $X = (P^{\oplus} \times \{0,1\})^{\oplus} \times ((P^{\oplus} \times \{0\})^{\oplus})^{\circledast} \times (P^{\oplus} \times \{2\})^{\oplus}$. Since $|P| = 1$, $P^{\oplus} \times \{0\}$ is isomorphic to $\mathbb{N}$. By Lemma 1 we have that $ot(X) \geq ot((\mathbb{N}^{\oplus})^{\circledast}) = \omega^{\omega^{\omega^{\omega}}}$. Let any $X' = (\mathbb{N}^m)^{\circledast} \in \mathbf{X}_{TdPN}$. By the previous lemma, $ot(X') = \omega^{\omega^{\omega^m}}$. Then $ot(X) \not\leq ot(X')$, and by Prop. 5 we conclude. □

Corollary 1. *TdPN* $\prec$ *ν-lsPN*

5 Conclusions

In this report, we have compared *ν-lsPN* with other classes of WSTS, proving that they are the most expressive of the studied classes. In particular, we have proved that *TdPN* $\prec$ *ν-lsPN* by applying the framework in [7].

As future work, we plan to study the expressive power of models in which a fixed number (possibly greater than one) of clocks is allowed. Also, it would be useful to compare the models yielded by bounded Petri nets with *TdPN* or NTA, in order to profit from the numerous works existing for the latter. In a different line, in our works we have assumed that processes (or their identifiers) are not ordered in any way. It would be interesting to see whether our work scales in the case of ordered processes, which amounts to extend *DN* with time.

Regarding complexity, since *ν-lsPN* are more expressive than DN or *TdPN*, the complexity of the control-state reachability problem can be proved to be non-primitive recursive. It would be interesting to obtain a finer-grained complexity analysis, as done in [12].

Other directions for further study include other properties, as the existence of Zeno behaviors [4], or liveness properties, although the negative results in the untimed case are discouraging [20].

References

1. Parosh Aziz Abdulla, Karlis Cerans, Bengt Jonsson, and Yih-Kuen Tsay. Algorithmic analysis of programs with well quasi-ordered domains. *Inf. Comput.*, 160(1-2):109–127, 2000.
2. Parosh Aziz Abdulla, Giorgio Delzanno, and Laurent Van Begin. A classification of the expressive power of well-structured transition systems. *Inf. Comput.*, 209(3):248–279, 2011.
3. Parosh Aziz Abdulla and Bengt Jonsson. Verifying networks of timed processes (extended abstract). In Bernhard Steffen, editor, *TACAS*, volume 1384 of *Lecture Notes in Computer Science*, pages 298–312. Springer, 1998.

4. Parosh Aziz Abdulla, Pritha Mahata, and Richard Mayr. Dense-timed Petri nets: Checking zenoness, token liveness and boundedness. *Logical Methods in Computer Science*, 3(1), 2007.
5. Parosh Aziz Abdulla and Aletta Nylén. Timed Petri nets and bqos. In José Manuel Colom and Maciej Koutny, editors, *ICATPN*, volume 2075 of *Lecture Notes in Computer Science*, pages 53–70. Springer, 2001.
6. Rajeev Alur, Costas Courcoubetis, and Thomas A. Henzinger. The observational power of clocks. In Bengt Jonsson and Joachim Parrow, editors, *CONCUR*, volume 836 of *Lecture Notes in Computer Science*, pages 162–177. Springer, 1994.
7. Rémi Bonnet, Alain Finkel, Serge Haddad, and Fernando Rosa-Velardo. Ordinal theory for expressiveness of well structured transition systems. *Inf. Comput.*, 2012.
8. Marc Boyer and Olivier H. Roux. On the compared expressiveness of arc, place and transition time Petri nets. *Fundam. Inform.*, 88(3):225–249, 2008.
9. D. H. J. de Jongh and R. Parikh. Well partial orderings and hierarchies. In *Indagationes Mathematicae*, volume 80, pages 195–207, 1977.
10. Alain Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theor. Comput. Sci.*, 256(1-2):63–92, 2001.
11. Gilles Geeraerts, Jean-François Raskin, and Laurent Van Begin. Well-structured languages. *Acta Inf.*, 44(3-4):249–288, 2007.
12. Serge Haddad, Sylvain Schmitz, and Philippe Schnoebelen. The ordinal-recursive complexity of timed-arc Petri nets, data nets, and other enriched nets. In *LICS*, pages 355–364. IEEE, 2012.
13. Ranko Lazic, Tom Newcomb, Joël Ouaknine, A. W. Roscoe, and James Worrell. Nets with tokens which carry data. *Fundam. Inform.*, 88(3):251–274, 2008.
14. María Martos-Salgado and Fernando Rosa-Velardo. Dynamic networks of infinite-state timed processes. Technical Report 9/13, DSIC Universidad Complutense de Madrid. `http://antares.sip.ucm.es/~frosa/`, 2013.
15. P. Merlin and D. Farber. Recoverability of communication protocols–implications of a theoretical study. *Communications, IEEE Transactions on*, 24(9):1036 – 1043, sep 1976.
16. Mauro Pezzè. Time Petri nets. In *Proceedings of the Multi-Workshop on Formal Methods in Performance Evaluation and Applications*, 1999.
17. C. Ramchandani. Analysis of asynchronous concurrent systems by timed Petri nets. Technical report, Cambridge, MA, USA, 1974.
18. Rami R. Razouk and Charles V. Phelps. Performance analysis using timed Petri nets. In Yechiam Yemini, Robert E. Strom, and Shaula Yemini, editors, *PSTV*, pages 561–576. North-Holland, 1984.
19. Fernando Rosa-Velardo and David de Frutos-Escrig. Name creation vs. replication in Petri net systems. *Fundam. Inform.*, 88(3):329–356, 2008.
20. Fernando Rosa-Velardo and David de Frutos-Escrig. Decidability and complexity of Petri nets with unordered data. *Theor. Comput. Sci.*, 412(34):4439–4451, 2011.
21. D. Schmidt. Well-partial orderings and their maximal order types, 1979. Habilitationsscrift.
22. Andreas Weiermann. A computation of the maximal order type of the term ordering on finite multisets. In Klaus Ambos-Spies, Benedikt Löwe, and Wolfgang Merkle, editors, *CiE*, volume 5635 of *Lecture Notes in Computer Science*, pages 488–498. Springer, 2009.