Information and Computation 224 (2013) 1–22

Contents lists available at SciVerse ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Ordinal theory for expressiveness of well-structured transition systems

Rémi Bonnet [a,*,1], Alain Finkel [a,1], Serge Haddad [a,1], Fernando Rosa-Velardo [b,2]

[a] Ecole Normale Supérieure de Cachan, LSV, CNRS UMR 8643, Cachan, France
[b] Sistemas Informáticos y Computación, Universidad Complutense de Madrid, Facultad de Informática, C/Prof. José García Santesmases, s/n, 28040 Madrid, Spain

ARTICLE INFO

Article history:
Received 5 July 2012
Available online 15 November 2012

Keywords:
Well-structured transition systems
Expressiveness
Coverability languages
Well-partial orders
Ordinals

ABSTRACT

We characterize the importance of resources (like counters, channels, or alphabets) when measuring the expressiveness of Well-Structured Transition Systems (WSTS). We establish, for usual classes of well partial orders, the equivalence between the existence of order reflections (non-monotonic order embeddings) and the simulations with respect to coverability languages. We show that the non-existence of order reflections can be proved by the computation of order types. This allows us to extend the current classification of WSTS, in particular solving some open problems, and to unify the existing proofs.

1. Introduction

WSTS. Infinite-state systems appear in many models and applications: stack automata, counter systems, Petri nets or VASSs, reset/transfer Petri nets, fifo (lossy) channel systems, parameterized systems.... Among these infinite-state systems, some of them, called Well-Structured Transition Systems (WSTS) [1], enjoy two nice properties: there is a well partial ordering (wpo) on the set of states and the transition relation is monotone with respect to this wpo.

The theory of WSTS has been successfully applied to the verification of safety properties of numerous infinite-state models like Lossy Channel Systems (LCS), extensions of Petri nets like reset/transfer and Affine Well Structured Nets (AWN) [2], or broadcast protocols. Most of the positive results are based on the decidability of the coverability problem (whether an upward closed set of states is reachable from the initial state) for WSTS, under natural effectiveness hypotheses. The reachability problem, on the contrary, is undecidable even for the class of Petri nets extended with reset or transfer transitions.

Expressiveness. Well Structured Languages [3] were introduced as a measure of the expressiveness of subclasses of WSTS. More precisely, the language of an instance of a model is defined as the class of *finite* words accepted by it, with *coverability* as accepting condition, that is, generated by traces that reach a state which is greater than a given final state. Convincing arguments show that the class of coverability languages is the right one. For instance, though reachability languages are more precise than coverability languages, the class of reachability languages is RE, the class of Recursively Enumerable languages, for almost all Petri nets extensions containing Reset Petri Nets or Transfer Petri Nets. We would like to answer the two following questions: (1) What are the (proper) inclusions between different subclasses of WSTS like LCS, AWN,

* Corresponding author.
E-mail addresses: remi.bonnet@lsv.ens-cachan.fr (R. Bonnet), alain.finkel@lsv.ens-cachan.fr (A. Finkel), serge.haddad@lsv.ens-cachan.fr (S. Haddad), fernandorosa@sip.ucm.es (F. Rosa-Velardo).

[1] These authors are partially supported by the Agence Nationale de la Recherche, AVERISS (grant ANR-06-SETIN-001), AVERILES (grant ANR-05-RNTL-002) and REACHARD (grant ANR-11-BS02-001).
[2] This author is partially supported by the MEC Spanish project DESAFIOS10 TIN2009-14599-C03-01, and the Comunidad de Madrid program PROMETIDOS S2009/TIC-1465.

0890-5401/$ – see front matter
http://dx.doi.org/10.1016/j.ic.2012.11.003

reset/transfer Petri nets, Data nets [4], identifying some types of languages which are not in some classes. Knowing that a particular kind of languages is not in a given class of WSTS may prevent us from looking for a model which does not exist. (2) Another natural question when confronted to an extension of a model is whether the additional resources actually yield an increase in expressiveness. For instance, for counter machines, it is well known that 1-counter machines, whose set of states is $Q \times \mathbb{N}$, where Q is a finite set, are strictly less powerful than 2-counter machines (i.e., Minsky machines), which operate on $Q \times \mathbb{N}^2$. Another example, if we look at Timed Automata, is that clocks are a strict resource: Timed Automata with k clocks are less expressive that Timed Automata with $k+1$ clocks [5]. Surprisingly, no similar results exist for well-known models like Petri Nets (with respect to the number of places) or Lossy Channel Systems (with respect to the number of channels, or number of symbols in the alphabet) except in some particular recent works [6].

Even if we will not study the following problem, let us mention it is connected to expressiveness: given a case study modelled (for instance) by a Lossy Channel System with p places, communicating through k lossy fifo channels, is there an equivalent (in some sense depending on the property we are interested in) Lossy Channel System with $p' < p$ places communicating through $k' < k$ lossy fifo channels? If we could answer this question, it would be possible to find a *minimal model* for the case study. This is not only of theoretical interest, since minimizing the size of the model (i.e., the number of places and channels in this example) is crucial when we want to verify it.

Finally, let us remark that the expressive power of WSTS comes from two natural sources: from the structure of the state space and from the semantics of the transition relation. These two notions were often extremely intertwined in the proofs. We propose ourselves to separate them in order to have a formal and generalizable method for comparing the expressiveness of WSTS.

Ordinal theory for partial orders. Ordinals are a well-known representation of well-founded total orders. Thanks to de Jongh, Parikh, Schmidt [7,8] and others, this representation has been extended to well partial orders. We are mainly interested in the order type of a wpo, which can be understood as the "size" of the order. The order types of the union, product, and finite words have been determined by de Jongh and Parikh. Recently, Weiermann [9] has completed this view by computing the order type for multisets.

Our contribution. First, we introduce order reflections, a variation of order embeddings that are allowed to be non-monotonic. We define a notion of witnessing, that reflects the ability of a WSTS to recognize a wpo through a coverability language. We establish the equivalence between the existence of order reflections and the simulations with respect to coverability languages, modulo the ability of the WSTS classes to witness their own state space.

Second, we show how to use results from the theory of ordinals, and more precisely the properties of maximal order types, studied by de Jongh, Parikh and Schmidt [7,8], to easily prove the absence of reflections.

Then, we study Lossy Channel Systems and extensions of Petri nets. We show that all the classes of WSTS considered are self-witnessing. In the first place, this allows us to unify and simplify the existing proofs regarding the classification of WSTS. Using our framework, we can easily prove that AWN are strictly less expressive than LCS and ν-Petri nets [10] (an extension of Petri nets with unordered data). Moreover, it allows us to solve the open problem [11] of the relative expressiveness of ν-Petri Nets and Data Nets [4] (an extension of Petri nets with ordered data). Apart from these qualitative results, we obtain new quantitative results stating that the number of unbounded places for these Petri nets extensions, and the size of the alphabet and number of channels for LCS, are relevant resources when considering their expressiveness.

Finally, we complete our view by putting Timed Petri nets [12] in our picture. By a close study of the state space of Timed Petri nets, we conclude that it is isomorphic to the state space of Data nets, even if these two models are quite different. We prove that both models are actually equivalent, hence strengthening our guiding principle about the importance of resources for the expressiveness on WSTS.

Related work. Coverability languages have been used to discriminate the expressive power of several WSTS, like Lossy Channel Systems or several monotonic extensions of Petri Nets. In [3] several pumping lemmas are proved to discriminate between extensions of Petri Nets. In [13–15] the expressive power of Petri Nets is proved to be strictly below that of Affine Well Nets, and Affine Well Nets are proved to be strictly less expressive than Lossy Channel Systems. Similar results are obtained in [11], though some significant problems are left open, like the distinction between ν-Petri Nets [10] and Data Nets [4] that we solve here. Recently, the study of the complexity of WSTS with states spaces equal to $\mathbb{N}^n$ or equal to Σ^* has begun in [16,6]: it consists in measuring, with multiple-recursive functions, the length of bad sequences (i.e., sequences of states that do not contain any increasing subsequence) in the wpos $\mathbb{N}^n$ and Σ^*. These complexity results can be sometimes used for proving strict inclusions between subclasses of WSTS.

Outline. The rest of the paper is organized as follows. In Section 2 we introduce wpos, WSTS and ordinals. Then in Section 3 we develop the study of reflections and its links to expressiveness of WSTS. Afterwards, in Section 4, we apply our result to the classical models of Petri Nets and Lossy Channel Systems. Section 5 presents the extension of our results applicable to more recent models of WSTS. In Section 6 we prove the equivalence between Data Nets and Timed Petri Nets. Finally we conclude and give perspectives to this work in Section 7.

This work is based on [17] and on the research report [18].

2. Preliminaries and WSTS

Well orders. $(X, \leqslant_X)$ is a *quasi-order* (qo) if $\leqslant_X$ is a reflexive and transitive binary relation on X. For a qo we write $x <_X y$ iff $x \leqslant_X y$ and $y \not\leqslant_X x$. A *partial order* (po) is an antisymmetric quasi-order. Given any qo $(X, \leqslant_X)$, the quotient set $X/\equiv_{\leqslant_X}$ is a po where $x \equiv_{\leqslant_X} y$ is defined by $x \leqslant_X y \wedge y \leqslant_X x$. Hence, in all the paper, we will suppose that $(X, \leqslant_X)$ is a po.

The *downward closure* of a subset $A \subseteq X$ is defined as $\downarrow A = \{x \in X \mid \exists x' \in A,\ x \leqslant x'\}$. A subset A is *downward closed* iff $\downarrow A = A$. A po $(X, \leqslant_X)$ is a *well partial order* (wpo) if for every infinite sequence $x_0, x_1, \ldots \in X$ there are i and j with $i < j$ such that $x_i \leqslant x_j$. Equivalently, a po is a wpo when there are no strictly decreasing (for inclusion) sequences of downward closed sets.

We will shorten $(X, \leqslant_X)$ to X when the underlying order is obvious. Similarly, $\leqslant$ will be used instead of $\leqslant_X$ when X can be deduced from the context.

If X and Y are wpos, their Cartesian product, denoted $X \times Y$ is well ordered by $(x, y) \leqslant_{X\times Y} (x', y') \iff x \leqslant_X x' \wedge y \leqslant_Y y'$. Their disjoint union, denoted $X \uplus Y$ is well ordered by:

$$z \leqslant_{X\uplus Y} z' \iff \begin{cases} z, z' \in X \\ z \leqslant_X z' \end{cases} \quad \text{or} \quad \begin{cases} z, z' \in Y \\ z \leqslant_Y z' \end{cases}$$

A po $(X, \leqslant)$ is *total* (or *linear*) if for any $x, x' \in X$ either $x \leqslant x'$ or $x' \leqslant x$. If $(X_i, \leqslant_i)$ are total po for $i \in \mathbb{N}$ we can define the (irreflexive) total order $<_{lex}$ in $\bigcup_k X_1 \times \cdots \times X_k$ by $(x_1, \ldots, x_p) <_{lex} (x'_1, \ldots, x'_q)$ iff there is $i \in \{1, \ldots, min(p, q)\}$ such that $x_j = x'_j$ for $j < i$ and $x_i <_i x'_i$ or $(x_1, \ldots, x_p) = (x'_1, \ldots, x'_p)$ and $q > p$. Then $\leqslant_{lex}$ given by $x \leqslant_{lex} x'$ iff $x = x'$ or $x <_{lex} x'$ is a total order.

Functions. Given a partial function (shortly: function) $f : X \to Y$, the *domain* of f is defined by $dom(f) = \{x \in X \mid \exists y \in Y,\ f(x) = y\}$ and its *range* by $range(f) = \{y \in Y \mid \exists x \in X,\ \ f(x) = y\}$. A function f is *surjective* if $range(f) = Y$ and it is *total* if $dom(f) = X$. Total functions are called *mappings*. A mapping f is *injective* if for all x, x', $f(x) = f(x') \Longrightarrow x = x'$. If X and Y are ordered, a mapping $f : X \to Y$ is *increasing* (resp. *strictly increasing*) if $x \leqslant_X y \Longrightarrow f(x) \leqslant_Y f(y)$ (resp. if $x <_X y \Longrightarrow f(x) <_Y f(y)$); f is an *order embedding* (shortly: embedding) if $f(x) \leqslant_Y f(x') \iff x \leqslant_X x'$. A bijective order embedding is called an *order isomorphism* (shortly: isomorphism).

Multisets. Given a set X, we denote by $X^{\oplus}$ the set of finite multisets of X, that is, the set of mappings $m : X \to \mathbb{N}$ with a finite support $sup(m) = \{x \in X \mid m(x) \neq 0\}$. We use the set-like notation $\{|\ldots|\}$ for multisets when convenient, with $\{|x^n|\}$ describing the multiset containing x n times. We use $+$ and $-$ for multiset addition and subtraction, respectively defined by $(m + m')(x) = m(x) + m'(x)$ and $(m - m')(x) = \max(m(x) - m'(x), 0)$. If X is a wpo then so is $X^{\oplus}$ ordered by $\leqslant_{\oplus}$ defined by $\{|x_1, \ldots, x_n|\} \leqslant_{\oplus} \{|x'_1, \ldots, x'_m|\}$ if there is an injection $h : \{1, \ldots, n\} \to \{1, \ldots, m\}$ such that $x_i \leqslant_X x'_{h(i)}$ for each $i \in \{1, \ldots, n\}$.

Words. Given a set X, any $u = x_1 \cdots x_n$ with $n \geqslant 0$ and $x_i \in X$, for all $i \in \{1, \ldots, n\}$, is a finite word on X. We denote by X^* the set of finite words on X. If $n = 0$ then u is the empty word, which is denoted by ε. We write $X_\varepsilon = X \cup \{\varepsilon\}$. A language L on X is a subset of X^*. Given L and L' two languages on X^*, we define the language $LL' = \{uv \mid u \in L, v \in L'\}$. If X is a wpo then so is X^* ordered by $\leqslant_{X^*}$ which is defined as follows: $x_1 \ldots x_n \leqslant_{X^*} x'_1 \ldots x'_m$ if there is a strictly increasing mapping $h : \{1, \ldots, n\} \to \{1, \ldots, m\}$ such that $x_i \leqslant_X x'_{h(i)}$ for each $i \in \{1, \ldots, n\}$ (Higman's lemma).

WSTS. A *Labelled Transition System* (LTS) is a tuple $\mathcal{S} = \langle X, \Sigma, \to\rangle$ where X is the set of states, Σ is the labelling alphabet and $\to \subseteq X \times \Sigma_\varepsilon \times X$ is the transition relation. We write $x \xrightarrow{a} x'$ to say that $(x, a, x') \in \to$. This relation is extended to $u \in \Sigma^*$ by $x \xrightarrow{u} x' \iff x \xrightarrow{a_1} x_1 \cdots x_{k-1} \xrightarrow{a_k} x'$ and $u = a_1 a_2 \cdots a_k$ (note that some a_i's can be ε). A *Well Structured Transition System* (shortly a WSTS) is a tuple $\mathcal{S} = (X, \Sigma, \to, \leqslant)$, where $(X, \Sigma, \to)$ is an LTS, and $\leqslant$ is a wpo on X, satisfying the following monotonicity condition: for all $x_1, x_2, x'_1 \in X, u \in \Sigma^*$, $x_1 \leqslant x'_1$, $x_1 \xrightarrow{u} x_2$ implies the existence of $x'_2 \in X$ such that $x'_1 \xrightarrow{u} x'_2$ and $x_2 \leqslant x'_2$. For a class $\mathbf{X}$ of wpos, we will denote by $WSTS_{\mathbf{X}}$ the class of WSTS with state space in $\mathbf{X}$, or just $WSTS_X$ for $WSTS_{\{X\}}$.

Coverability and reachability languages. Trace languages, reachability languages and coverability languages are natural candidates for measuring the expressive power of classes of WSTS. Given a WSTS $\mathcal{S}$ and two states x_0 and x_f, the reachability language is $L_R(\mathcal{S}, x_0, x_f) = \{u \in \Sigma^* \mid x_0 \xrightarrow{u} x_f\}$ while the coverability language is $L(\mathcal{S}, x_0, x_f) = \{u \in \Sigma^* \mid x_0 \xrightarrow{u} x,\ x \geqslant x_f\}$. Finally, the trace language is given by $L_T(\mathcal{S}, x_0) = \{u \in \Sigma^* \mid \exists x_f.\ x_0 \xrightarrow{u} x_f\}$. Let us remark that all trace languages are coverability languages in taking $x_f = \bot$ where $\bot$ is the least element of X (if there is one). Also, assuming it is possible to have transitions that test whether the state is greater than some x_f, coverability languages can be obtained from trace languages by intersecting by some regular language (and WSTS are closed under intersection [3]).

To justify our choice of coverability languages as measure of expressiveness, we note that the class of reachability languages is the set of recursively enumerable languages for all Petri nets extensions containing reset Petri nets or transfer Petri nets. Thus, such a criterion does not discriminate sufficiently. One could consider infinite coverability languages. A sensible accepting condition in this case could be repeated coverability, that is, the capacity of covering a given marking infinitely

often, in the style of Büchi automata. However, analogously to what happens with reachability, repeated coverability is generally undecidable (except for some notable exceptions, like Petri nets), which makes ω-languages a bad candidate to study the relative expressive power of WSTS. In conclusion, we will use the class of coverability languages, as in [3,13,14,11]. Because of our remarks on the relation between coverability languages and trace languages, it will generally be possible to translate results on coverability languages to results on trace languages.

For two classes of WSTS $\mathbf{S}_1$ and $\mathbf{S}_2$, we write $\mathbf{S}_1 \preccurlyeq \mathbf{S}_2$ whenever for every language $L(\mathcal{S}_1, x_1, x'_1)$ with $\mathcal{S}_1 \in \mathbf{S}_1$, and x_1, x'_1 two states of $\mathcal{S}_1$, there exists another system $\mathcal{S}_2 \in \mathbf{S}_2$ and two states x_2, x'_2 of $\mathcal{S}_2$ such that $L(\mathcal{S}_2, x_2, x'_2) = L(\mathcal{S}_1, x_1, x'_1)$. When $\mathbf{S}_1 \preccurlyeq \mathbf{S}_2$ and $\mathbf{S}_2 \preccurlyeq \mathbf{S}_1$, one denotes the equivalence of classes by $\mathbf{S}_1 \simeq \mathbf{S}_2$. We write $\mathbf{S}_1 \prec \mathbf{S}_2$ for $\mathbf{S}_1 \preccurlyeq \mathbf{S}_2$ and $\mathbf{S}_2 \not\preccurlyeq \mathbf{S}_1$. Clearly, $\preccurlyeq$ is reflexive and transitive.

The lossy semantics. The *lossy* semantics $\mathcal{S}_l$ of an LTS $\mathcal{S}$ with state space X endowed with a wpo $\leqslant$ is the original system $\mathcal{S}$ completed by all ε-transitions $x \xrightarrow{\varepsilon} y$, for all $x, y \in X$ such that $y < x$. We observe that $\mathcal{S}_l$ satisfies the monotonicity condition, hence $\mathcal{S}_l$ is a WSTS. Moreover, due to the lossy semantics one has: for all $x_1, x_2 \in X, u \in \Sigma^*$, $x_1 \xrightarrow{u} x_2$ implies $x_1 \xrightarrow{u} x'_2$ for all $x'_2 \leqslant x_2$. This implies that $L(\mathcal{S}_l, x_0, x_f) = L_R(\mathcal{S}_l, x_0, x_f)$ for any $x_0, x_f \in X$. Moreover, if $\mathcal{S}$ was already a WSTS, we also have: $L(\mathcal{S}, x_0, x_f) = L(\mathcal{S}_l, x_0, x_f)$.

3. A method for comparing WSTS

In this section we propose a method to compare the expressiveness of WSTS mainly based on their state space. We will prove some results that provide us with tools to establish strict relations between classes of WSTS.

3.1. A new tool: order reflections

Definition 1 *(Order reflections).* Let $(X, \leqslant_X)$ and $(Y, \leqslant_Y)$ be two partially ordered sets. A mapping $\varphi : X \to Y$ is an *order reflection* (shortly: reflection) if $\varphi(x) \leqslant_Y \varphi(x')$ implies $x \leqslant_X x'$ for all $x, x' \in X$.

We will write $X \sqsubseteq Y$ if there is an embedding from X to Y and $X \sqsubseteq_{refl} Y$ if there is a reflection from X to Y. We will use $\not\sqsubseteq$ and $\not\sqsubseteq_{refl}$ for their negation and $\sqsubset$ and $\sqsubset_{refl}$ for their antisymmetric version (i.e. $X \sqsubset Y \iff X \sqsubseteq Y \wedge Y \not\sqsubseteq X$). Here are some basic properties of reflections we will use throughout the paper: for any set X, any injective mapping to $(X, =)$ is a reflection; every reflection is injective; the composition of two reflections is a reflection (so $\sqsubseteq_{refl}$ is transitive).

Furthermore, if φ is an embedding from X to Y then X is isomorphic to $\varphi(X)$ and hence can be identified to it. Clearly, existence of embeddings is a stronger requirement than the existence of reflections. In particular, it can be the case that a wpo X cannot be embedded in another wpo Y, even if there are reflections from X to Y, as implied by the following result.

Proposition 1. *The following properties hold*:

- $\mathbb{N}^k \sqsubseteq_{refl} \mathbb{N}^\oplus$, *for any* $k > 0$.
- $\mathbb{N}^k \not\sqsubseteq \mathbb{N}^\oplus$ *for any* $k \geqslant 3$ (*but* $\mathbb{N}^2 \sqsubseteq \mathbb{N}^\oplus$).

Proof. The proof of $\mathbb{N}^3 \not\sqsubseteq \mathbb{N}^\oplus$ is technical and of little interest for the remainder of the paper. It is available in Appendix A, Proposition 17.

The mapping $\varphi : \mathbb{N}^2 \to \mathbb{N}^\oplus$ given by $\varphi(a, b) = \{|a + 2, 1^b|\}$ is an order-embedding, so the only part remaining is to show that there is an order reflection from $\mathbb{N}^k$ to $\mathbb{N}^\oplus$, for any $k > 0$.

Let us take a fixed $k \in \mathbb{N}$. There are $k!$ possible relative orders of $x_1, \ldots, x_k$. Let us denote $N_k = k!$ and let o_k be a mapping that associates with each tuple $(x_1, \ldots, x_k)$ a number between 0 and $N_k - 1$ such that $o_k(x_1, \ldots, x_k) = o_k(x'_1, \ldots, x'_k)$ means that $x_1, \ldots, x_k$ and $x'_1, \ldots, x'_k$ are in the same relative order.

We define $ac : \{0, \ldots, N_k - 1\} \to \mathbb{N}^\oplus$ by $ac(n) = \{|2N_k - (n + 1), n|\}$. Note that $ac(m)$ and $ac(n)$ are incomparable with respect to the multiset order if m and n are different numbers between 0 and $N_k - 1$.

Now we define φ by:

$$\varphi(x_1, \ldots x_k) = \big\{\big|(2N_k + x_1), (2N_k + x_2), \ldots, (2N_k + x_k)\big|\big\} + ac\big(o_k(x_1, \ldots x_k)\big)$$

We claim this is an order reflection. Indeed, let us take $X = (x_1, \ldots, x_k)$ and $X' = (x'_1, \ldots, x'_k)$ and assume that we have $\varphi(X) \leqslant_{\mathbb{N}^\oplus} \varphi(X')$. Then, there is a bijective mapping $\sigma : \varphi(X) \to \varphi(X')$ with:

$$\varphi(X) = \big\{\big|2N_k + x_1, \ldots, 2N_k + x_k, 2N_k - \big(o_k(X) + 1\big), o_k(X)\big|\big\}$$
$$\varphi\big(X'\big) = \big\{\big|2N_k + x'_1, \ldots, 2N_k + x'_k, 2N_k - \big(o_k\big(X'\big) + 1\big), o_k\big(X'\big)\big|\big\}$$
$$\forall x \in \varphi(X).\ x \leqslant \sigma(x)$$

The cardinalities of $\varphi(X)$ and $\varphi(X')$ are the same, and the elements of the form $2N_k + x_i$ can only be mapped to one of their counterpart, so:

$$\sigma\big(2N_k - \big(o_k(X)+1\big)\big) = 2N_k - \big(o_k\big(X'\big)+1\big)$$
$$\sigma\big(o_k(X)\big) = o_k\big(X'\big)$$

This means that $o_k(X) = o_k(X')$. The components of X and X' are thus in the same relative order. Without loss of generality, we will assume this order is $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_k$. Let us assume by contradiction that $X \not\preccurlyeq X'$, so that there exists i such that $x_j \leqslant x'_j$ for all $j > i$ and $x_i > x'_i$. Then, x_i is mapped by σ to some x'_m, so we have $x_i \leqslant x'_m$ for some m. Two cases may occur:

- $m > i$: Then by cardinality, we have an element x_p in $\{x_{i+1}, \ldots, x_k\}$ that is mapped by σ to an element $x'_{p'}$ with $p' \leqslant i$. Thus, we have $x_i \leqslant x_p \leqslant x'_{p'} \leqslant x'_i$, contradicting our assumption that $x'_i < x_i$.
- $m < i$: Then, we have $x_i \leqslant x'_m \leqslant x'_i$, contradicting again our assumption.

Thus, we have $x_i \leqslant x'_i$ for all i, concluding our demonstration. □

3.2. Expressiveness of WSTS and order reflections

Reflections are more appropriate than embeddings for the comparison of WSTS. In particular, the existence of a reflection implies the relation between the corresponding classes of WSTS.

Theorem 1. *Let X and Y be two wpo. We have*:

$$X \sqsubseteq_{refl} Y \quad \Longrightarrow \quad WSTS_X \preccurlyeq WSTS_Y$$

Proof. Let $L = L(\mathcal{S}, x_0, x_f)$ for some WSTS $\mathcal{S} = \langle X, \Sigma, \to^*, \leqslant\rangle$ with state space X with $x_0, x_f \in X$, respectively. Because a WSTS has the same coverability languages as its lossy version, we can assume that $\mathcal{S}$ is a lossy WSTS.

Let φ be a reflection from X to Y. Since φ is an injection, we can consider the following labelled transition system $\mathcal{S}^\varphi$, of states $\varphi(X) \subseteq Y$, with initial and final states $\varphi(x_0)$ and $\varphi(x_f)$, respectively, and whose transitions $a \in \Sigma$ are defined by:

$$\varphi(x) \xrightarrow{a}_{\mathcal{S}^\varphi} \varphi\big(x'\big) \quad \Longleftrightarrow \quad x \xrightarrow{a}_{\mathcal{S}} x'$$

It holds that $\mathcal{S}^\varphi \in WSTS_Y$. Indeed, if we take $\varphi(x_1)$, $\varphi(x'_1)$ and $\varphi(x_2)$ such that $\varphi(x_1) \xrightarrow{u}_{\mathcal{S}^\varphi} \varphi(x'_1)$ and $\varphi(x_2) \geqslant \varphi(x_1)$, then we have by definition of $\mathcal{S}^\varphi$, and because φ is a reflection, that $x_1 \xrightarrow{u}_{\mathcal{S}} x'_1$ and $x_2 \geqslant x_1$, which means, by well-structure of $\mathcal{S}$, that there exists $x'_2 \geqslant x'_1$ such that $x_2 \xrightarrow{u}_{\mathcal{S}} x'_2$. By the lossiness property of $\mathcal{S}$, we have $x_2 \xrightarrow{u}_{\mathcal{S}} x'_1$, and thus $\varphi(x_2) \xrightarrow{u}_{\mathcal{S}^\varphi} \varphi(x'_1)$. Moreover, $\mathcal{S}$ and $\mathcal{S}^\varphi$ clearly recognize the same language, so that $L = L(\mathcal{S}^\varphi, \varphi(x_0), \varphi(x_f))$ with $\mathcal{S}^\varphi \in WSTS_Y$, which concludes our proof. □

We would like to obtain the converse of the previous result: $X \not\sqsubseteq_{refl} Y \Longrightarrow WSTS_X \not\preccurlyeq WSTS_Y$. First, we only present this result for "simple" state spaces. The case of more complex state spaces will be handled in Subsection 3.3. In both cases, the result makes use of a class of languages, that we call witness languages.

Given an alphabet $\Sigma = \{a_1, \ldots, a_k\}$, we define $\overline{\Sigma}$ by $\overline{\Sigma} = \{\overline{a_1}, \ldots, \overline{a_k}\}$ where $\overline{a_i}$'s are fresh symbols (i.e. $\Sigma \cap \overline{\Sigma} = \emptyset$). This notation is extended to words by $\overline{u} = \overline{a_1} \cdots \overline{a_k}$ for $u = a_1 \cdots a_k \in \Sigma^*$. In the same way, given $L \subseteq \Sigma^*$, we have $\overline{L} = \{\overline{u} \mid u \in L\} \subseteq \overline{\Sigma}^*$.

Definition 2 *(Witness languages).* Let X be a wpo and Σ a finite alphabet. A *Σ-representation* of X is any surjective partial function $\gamma : \Sigma^* \to X$. For a Σ-representation γ of X, we define $L_\gamma = \{u\overline{v} \mid u, v \in dom(\gamma) \text{ and } \gamma(v) \leqslant \gamma(u)\}$. A language $L \in (\Sigma \cup \overline{\Sigma})^*$ is a *γ-witness* (shortly: witness) of X if $L \cap dom(\gamma)\overline{dom(\gamma)} = L_\gamma$.

Intuitively, given a witness L of X, the fact that a WSTS can recognize L *witnesses* that the WSTS can represent 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 $\gamma(v) \leqslant \gamma(u)$. Witness languages are useful for proving strict relations between classes of WSTS:

Theorem 2. *Let L be a witness of X. If $X \not\sqsubseteq_{refl} Y$ then there are no $y_0, y_f \in Y$ and no $\mathcal{S} \in WSTS_Y$ such that $L = L(\mathcal{S}, y_0, y_f)$.*

Proof. Assume by contradiction that L is a γ-witness of X which is the coverability language of a WSTS $\mathcal{S}$ whose state space is Y, with y_0 and y_f as initial and final states, respectively. For each $x \in X$, let us pick a $u_x \in \Sigma^*$ such that $\gamma(u_x) = x$. The word $u_x\overline{u_x}$ is recognized by $\mathcal{S}$, hence we can find y_x and y'_x such that $y_0 \xrightarrow{u_x} y_x \xrightarrow{\overline{u_x}} y'_x \geqslant y_f$.

We define $\varphi(x) = y_x$. Let us prove that φ is an order reflection from X to Y, thus reaching a contradiction. Assume that $\varphi(x) \leqslant \varphi(x')$. We have $y_0 \xrightarrow{u_x} y_x \xrightarrow{\overline{u_x}} y'_x \geqslant y_f$ and $y_0 \xrightarrow{u_{x'}} y_{x'} \xrightarrow{\overline{u_{x'}}} y'_{x'} \geqslant y_f$ with $\varphi(x) = y_x \leqslant y_{x'} = \varphi(x')$. Since $\mathcal{S}$ is a WSTS, the sequence $\overline{u_x}$, which is fireable from y_x, is also fireable from $y_{x'}$, and the state reached by this subsequence is greater or equal than y'_x. Hence, the state reached after $u_{x'}\overline{u_x}$ is greater or equal than the one reached after $u_x\overline{u_x}$, which means that $u_{x'}\overline{u_x} \in L \cap dom(\gamma)\overline{dom(\gamma)} = L_\gamma$. By definition of L_γ, this implies that $\gamma(u_x) = x \leqslant x' = \gamma(u_{x'})$, so that φ is an order reflection. □

The simple state spaces we mentioned before are the ones produced by the following grammar:

$$
\begin{array}{rcll}
\Gamma & ::= & Q & \text{(finite set with equality)} \\
& | & \mathbb{N} & \text{(naturals with the standard order)} \\
& | & \Sigma^* & \text{(words on a finite set with the order defined in Section 2)} \\
& | & \Gamma \times \Gamma & \text{(Cartesian product with the order defined in Section 2)}
\end{array}
$$

As $\mathbb{N}$ is isomorphic to Σ^* when Σ is a singleton, any set produced by Γ is isomorphic to a set $Q \times \Sigma_1^* \times \cdots \times \Sigma_k^*$ where Q and each Σ_i are finite sets.

Proposition 2. *Let X be a set produced by the grammar Γ. Then, there is a witness of X that is recognized by a WSTS of state space X.*

Proof. We have $X = Q \times \Sigma_1^* \times \cdots \times \Sigma_k^*$, ordered by its canonic order $\leqslant_X$ (which is the Cartesian product of equality on Q and subword ordering on Σ_i^* for all i). Without loss of generality, we assume that the Σ_i's are disjoint. We also define $\Sigma_T = \bigcup_{1 \leqslant i \leqslant k} \Sigma_i$ and $\Sigma_Q = \{a_q \mid q \in Q\}$ (Σ_T and Σ_Q disjoint). Finally, we choose arbitrarily a $q_0 \in Q$.

We define a WSTS $\mathcal{S} = \langle X, \Sigma, \to, \leqslant_X \rangle$ by:

- $\Sigma = \Sigma_T \cup \Sigma_Q \cup \overline{\Sigma}_T \cup \overline{\Sigma}_Q$.
- For $a \in \Sigma_T$,

$$
(q, u_1, \ldots, u_k) \xrightarrow{a} (q', u'_1, \ldots, u'_k) \iff \begin{cases} q = q' \\ u'_i = u_i a & \text{if } a \in \Sigma_i \\ u'_j = u_j & \text{otherwise} \end{cases}
$$

- For $\bar{a} \in \overline{\Sigma}_T$,

$$
(q, u_1, \ldots, u_k) \xrightarrow{\bar{a}} (q', u'_1, \ldots, u'_k) \iff \begin{cases} q = q' \\ u_i = a u'_i & \text{if } a \in \Sigma_i \\ u_j = u'_j & \text{otherwise} \end{cases}
$$

- For $a_p \in \Sigma_Q$,

$$
(q, u_1, \ldots, u_k) \xrightarrow{a_p} (q', u'_1, \ldots, u'_k) \iff \begin{cases} q = q_0 \\ q' = p \\ u'_i = u_i \end{cases}
$$

- For $\overline{a_p} \in \Sigma_Q$,

$$
(q, u_1, \ldots, u_k) \xrightarrow{\overline{a_p}} (q', u'_1, \ldots, u'_k) \iff \begin{cases} q = p \\ q' = q_0 \\ u'_i = u_i \end{cases}
$$

- $s \xrightarrow{\epsilon} s' \iff s' \leqslant s$.

Indeed, $\mathcal{S}$ is a WSTS because it is defined by the lossy semantics (last item) of an LTS. We define $\gamma(x) = (q, u_1, \ldots, u_k)$ iff $x \in a_q \| u_1 \| \cdots \| u_k$, where $\|$ denotes the shuffling operation (i.e. $z \in u \| v \iff z = u_1 v_1 u_2 \cdots u_p v_p$ with $u = u_1 u_2 \cdots u_p$ and $v = v_1 v_2 \cdots v_p$, with $u_i, v_i \in \Sigma^*$). γ is a $(\Sigma_T \cup \Sigma_Q)$-representation of X.

We define $L = L(\mathcal{S}, (q_0, \epsilon, \ldots, \epsilon), (q_0, \epsilon, \ldots, \epsilon))$ and we have:

$$
L \cap dom(\gamma)\overline{dom(\gamma)} = \left\{ u\bar{v} \;\middle|\; u, v \in dom(\gamma) \text{ and } \gamma(v) \leqslant \gamma(u) \right\}
$$

This concludes the demonstration. □

WSTS that can recognize witnesses of their own state spaces are especially interesting. Indeed, in this case one can use Theorem 2 to obtain an equivalence between the existence of an order reflection, and inclusion for the sets of recognized languages. In particular:

Proposition 3. *Let X be a wpo produced by Γ and Y any wpo. Then,*

$$X \sqsubseteq_{refl} Y \iff WSTS_X \preccurlyeq WSTS_Y$$

Proof. The direction from left to right is given by Theorem 1. Conversely, let us prove that $X \not\sqsubseteq_{refl} Y \Longrightarrow WSTS_X \not\preccurlyeq WSTS_Y$. We can find a witness L of X recognized by a WSTS of state space X (Proposition 2). By Theorem 2, this language cannot be recognized by a WSTS of state space Y, hence the result. □

3.3. Self-witnessing WSTS classes

The reason we were able to build our equivalence between the existence of a reflection from X to Y and $WSTS_X \preccurlyeq WSTS_Y$ for any wpo X produced by Γ was Proposition 2. However, we conjecture that this result is no longer true for any state space X that embeds $\mathbb{N}^{\oplus}$, that is, that there is no WSTS of state space X that can recognize a witness of X. This prompts us to define a new notion:

Definition 3 *(Self-witnessing).* Let $\mathbf{X}$ be a class of wpos and $\mathbf{S}$ 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.

In particular, Proposition 2 states that $(WSTS_X, \{X\})$ is self-witnessing for any X produced by the grammar Γ. We will shorten $(\mathbf{X}, \mathbf{S})$ as $\mathbf{S}$ when the state space is not explicitly needed. 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 $X \sqsubseteq_{refl} X'$. Next, we prove the result analogous to Proposition 2 for self-witnessing classes.

Proposition 4. *Let $(\mathbf{X}, \mathbf{S})$ be a self-witnessing WSTS class and $\mathbf{S}'$ a WSTS class using state spaces inside $\mathbf{X}'$. Then, $\mathbf{S} \preccurlyeq \mathbf{S}' \Longrightarrow \mathbf{X} \sqsubseteq_{refl} \mathbf{X}'$.*
Moreover, if $\mathbf{S}' = WSTS_{\mathbf{X}'}$, $\mathbf{S} \preccurlyeq \mathbf{S}' \iff \mathbf{X} \sqsubseteq_{refl} \mathbf{X}'$.

Proof. Let us show the first implication. Let $X \in \mathbf{X}$. Since $(\mathbf{X}, \mathbf{S})$ is self-witnessing, there is $\mathcal{S} \in \mathbf{S}$ that recognizes L, a witness of X. Because $\mathbf{S} \preccurlyeq \mathbf{S}'$, there is $\mathcal{S}' \in \mathbf{S}'$ recognizing L. $\mathcal{S}'$ has state space $X' \in \mathbf{X}'$, and by Theorem 2, $X \sqsubseteq_{refl} X'$.

For the second implication, for any $X \in \mathbf{X}$, we have $X' \in \mathbf{X}'$ such that $X \sqsubseteq_{refl} X'$. Because of Theorem 1, $WSTS_X \preccurlyeq WSTS_{X'}$. Hence, $WSTS_{\mathbf{X}} \preccurlyeq WSTS_{\mathbf{X}'}$. Since clearly $\mathbf{S} \preccurlyeq WSTS_{\mathbf{X}}$, we conclude. □

We will see in Section 4 and Section 5 that many usual classes of WSTS, even those outside the algebra Γ, are self-witnessing.

3.4. How to prove the non-existence of reflections?

Because of Proposition 3 and Proposition 4, the non-existence of reflections will be a powerful tool to prove strict relations between WSTS. We will use some results from set-theoretical ordinals to get a simple way of disproving the existence of such reflections. First, we recall a few properties of these objects.

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 $(X, \leqslant_X)$ is isomorphic to a unique ordinal $ot(X, \leqslant_X)$, called the *order type* of X.

In the context of ordinals, we define $0 = \emptyset$, $n = \{0, \dots, n-1\}$ 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 $\leqslant_{lex}$. In the same way, $\alpha * \alpha'$ is defined as the order type of $\alpha' \times \alpha$ ordered by $\leqslant_{lex}$. Note that these operations are not commutative: we have $1 + \omega = \omega \neq \omega + 1$. The definitions of $+$ and $*$ coincide with the usual operations on $\mathbb{N}$ for ordinals below ω, and we have $\alpha + \overset{k}{\cdots} + \alpha = \alpha * k$. We can also define exponentiation by having α^β be the order type of the set of functions from β to α ordered by $\leqslant_{lex}$ defined by:

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

We will work with ordinals below ε_0, that is, those that can be bounded by a tower $\omega^{\omega^{\cdot^{\cdot^{\omega}}}}$. These can be represented by the hierarchy of ordinals in Cantor Normal Form (CNF), that is recursively given by the following rules:
$C_0 = \{0\}$.
$C_{n+1} = \{\omega^{\alpha_1} + \cdots + \omega^{\alpha_p} \mid p \in \mathbb{N},\ \alpha_1, \dots, \alpha_p \in C_n \text{ and } \alpha_1 \geqslant \cdots \geqslant \alpha_p\}$ ordered by:

$$\omega^{\alpha_1} + \cdots + \omega^{\alpha_p} \leqslant \omega^{\alpha'_1} + \cdots + \omega^{\alpha'_q} \iff (\alpha_1, \ldots, \alpha_p) \leqslant_{lex} (\alpha'_1, \ldots, \alpha'_q)$$

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 $\{|\beta_1, \ldots, \beta_n|\}$.

Let us recall that a *linearization* of a po $\leqslant_X$ is a total order $\leqslant'_X$ on X such that $x \leqslant_X y \Longrightarrow x \leqslant'_X y$. A linearization of a wpo is a well total order, hence isomorphic to an ordinal. We extend the definition of order types to non-total wpos:

Definition 4. Let $(X, \leqslant_X)$ be a wpo. The *maximal order type* (shortly: order type) of $(X, \leqslant_X)$ is $ot(X, \leqslant_X) = sup\{ot(X, \leqslant'_X) \mid \leqslant'_X$ linearization of $\leqslant_X\}$.

The existence of the *sup* comes from ordinal theory. Moreover, de Jongh and Parikh [7] even show that this *sup* is actually attained. Let $Down(X)$ be the set of downward closed subsets of X. Then, another well-known characterization of the maximal order type is the following:

Proposition 5. $ot(X) + 1 = sup\{\alpha \mid \exists f : \alpha \to Down(X),\ f$ *strictly increasing*$\}$.

Proof. We first prove that $ot(X) + 1 \leqslant sup\{\alpha \mid \exists f : \alpha \to Down(X),\ f$ strictly increasing$\}$.

Let $\leqslant'$ be a linearization of $\leqslant$ of order type $ot(X)$. Let φ be an isomorphism from $ot(X)$ to $(X, \leqslant')$. We define $f : ot(X) + 1 \to Down(X)$ by:

$$f(\beta) = \left\{x \in X \;\middle|\; x <' \varphi(\beta)\right\} \quad \text{for } \beta < ot(X)$$
$$f\big(ot(X)\big) = X$$

f is strictly increasing, which means that: $ot(X) + 1 \in \{\alpha \mid \exists f : \alpha \to Down(X),\ f$ strictly increasing$\}$ and concludes the first part of the proof.

We now prove that $ot(X) + 1 \geqslant sup\{\alpha \mid \exists f : \alpha \to Down(X),\ f$ strictly increasing$\}$.

Let α be an ordinal and f be a strictly increasing mapping from α to $Down(X)$. We define the quasi-order $\leqslant_f$ on X by:

$$x \leqslant_f y \quad \text{iff} \quad \forall \beta < \alpha, \quad y \in f(\beta) \implies x \in f(\beta)$$

$\leqslant_f$ is clearly reflexive and transitive. Let $\leqslant_{tie}$ be a linearization of $\leqslant_X$. We define the order $\leqslant'_f$ by:

$$x \leqslant'_f y \iff \begin{cases} x \leqslant_f y \wedge y \not\leqslant_f x \quad \text{or,} \\ x \leqslant_f y \wedge y \leqslant_f x \wedge x \leqslant_{tie} y \end{cases}$$

$\leqslant'_f$ is clearly reflexive and antisymmetric. Let us show transitivity. Assume that $x \leqslant'_f y$ and $y \leqslant'_f z$. If they are all three in the same equivalence class (resp. in different equivalence classes) of $\equiv_{\leqslant_f}$, $x \leqslant'_f z$ comes from transitivity of $\leqslant_{tie}$ (resp. $\leqslant_f$). If x and y are $\leqslant_f$-equivalent, and $y <_f z$ we immediately get $x <'_f z$. The last case is similar.

Let us prove that $\leqslant'_f$ is a linear order. Pick any x and y. If they are equivalent w.r.t. $\leqslant_f$, we get the result by linearity of $\leqslant_{tie}$. So assume by symmetry that there exists β, $x \in f(\beta)$ and $y \notin f(\beta)$. Then for any β' such that $y \in f(\beta')$, $\beta < \beta'$ since f is strictly increasing and ordinals are totally ordered. Thus $x \in f(\beta')$. Since β' is arbitrary, this shows that $x \leqslant'_f y$.

Let us prove that $\leqslant'_f$ is a linearization of $\leqslant_X$. Pick any $x \leqslant_X y$ (and thus $x \leqslant_{tie} y$). Because for all β, $f(\beta)$ is downward closed, we have $x \leqslant_f y$, which leads to $x \leqslant'_f y$.

Choose some $x_{\max} \notin X$, and $X' = X \cup \{x_{\max}\}$. We extend $\leqslant'_f$ on X' by $x \leqslant'_f x_{\max}$ for all $x \in X$. We define $\varphi : \alpha \to (X', \leqslant'_f)$ by:

$$\varphi(\beta) = \min_{\leqslant'_f}\left\{x \in X' \;\middle|\; x \notin f(\beta)\right\}$$

The min is defined because X' is well-ordered and at least $x_{\max} \notin f(\beta)$ for any β. Because f is increasing, φ is also monotonic.

Let us show that φ is an order embedding. Assume $\beta < \beta'$. Then there exists y such that $y \in f(\beta')$ and $y \notin f(\beta)$. This means $\varphi(\beta) \leqslant'_f y$. As $y \in f(\beta')$ and $f(\beta')$ is downward closed, $\varphi(\beta) \in f(\beta')$, which implies $\varphi(\beta) < \varphi(\beta')$.

We have an order embedding from α to $(X', \leqslant'_f)$ which means $\alpha \leqslant ot(X') = ot(X) + 1$. □

The reason that order types are particularly useful to prove the absence of order reflections is that these reflections preserve strict inclusions of downward closed sets (and by Proposition 5, we have seen that order types can be defined by strictly increasing sequences of downward closed sets).

Lemma 1. *Let X and Y be two wpos and φ a reflection from X to Y. Let $A \subsetneq X$ with $A = {\downarrow}A$. Then ${\downarrow}\varphi(A) \subsetneq Y$*

Proof. Let us assume that $\downarrow\varphi(A) = Y$. Let us take $x \in X$, $x \notin A$. Since $\varphi(x) \in Y$ and $\downarrow\varphi(A) = Y$, there is $x' \in A$ such that $\varphi(x) \leqslant \varphi(x')$. Since φ is a reflection we have $x \leqslant x'$ and since A is downward closed $x \in A$, hence the contradiction. □

This leads us to the proposition that we use to separate many classes of WSTS (originally found in [9]):

Proposition 6. *(See [9].) Let X and Y be two wpos. Then,*

$$X \sqsubseteq_{refl} Y \quad \Longrightarrow \quad ot(X) \leqslant ot(Y)$$

Proof. Let $\varphi : X \to Y$ be a reflection and let us consider an ordinal α and a mapping $f : \alpha \to Down(X)$, strictly increasing. We define $g : \alpha \to Down(Y)$ by $g(\beta) = \downarrow\varphi(f(\beta))$. By Lemma 1, g is strictly increasing. By the characterization of order types in Proposition 5, we have $ot(X) \leqslant ot(Y)$. □

The order types of the usual state spaces used for WSTS are known. We will recall some classic results on these order types, but we need the following definitions of addition and multiplication on ordinals to be able to characterize the order types of $X \uplus Y$ and $X \times Y$. Remember (Section 2) that an ordinal α below ε_0 is uniquely determined by $Cantor(\alpha)$, hence the validity of the following definition.

Definition 5 *(Hessenberg 1906).* (See [7].) The *natural addition*, denoted $\oplus$, and the *natural multiplication*, denoted $\otimes$, are defined by:

$$Cantor(\alpha \oplus \alpha') = Cantor(\alpha) + Cantor(\alpha')$$
$$Cantor(\alpha \otimes \alpha') = \left\{ \left| \beta \oplus \beta' \;\middle|\; \beta \in Cantor(\alpha),\ \beta' \in Cantor(\alpha') \right| \right\}$$

We already know that the order type of a finite set (with any order) is its cardinality and that the order type of $\mathbb{N}$ is ω. De Jongh and Parikh [7], and Schmidt [8] have shown a way to compose order types with the disjoint union, the Cartesian product, and the Higman ordering. A more recent and difficult result, by Weiermann [9], provides us with the order type of multisets. These results are summed up here:

Proposition 7. *(See [7–9].)*

- $ot(X \uplus Y) = ot(X) \oplus ot(Y)$,
- $ot(X \times Y) = ot(X) \otimes ot(Y)$,
- $$ot(X^*) = \begin{cases} \omega^{\omega^{ot(X)-1}} & \text{if } X \text{ finite} \\ \omega^{\omega^{ot(X)}} & \text{otherwise} \end{cases}$$
  (*for* $ot(X) < \epsilon_0$),
- $ot(X^{\oplus}) = \omega^{\omega^{ot(X)}}$ *for* $ot(X) < \epsilon_0$.

Formulas exist even for $ot(X) \geqslant \epsilon_0$. We refer the interested reader to [7] and [9] for the complete formulas. With these general results we can obtain many strict relations between wpo.

Corollary 1. *The following strict relations hold for any $k > 0$:*

(1) $\mathbb{N}^k \sqsubset_{refl} \mathbb{N}^{k+1}$ (4) $\mathbb{N}^k \sqsubset_{refl} \mathbb{N}^{\oplus}$

(2) $\left(\mathbb{N}^k\right)^{\oplus} \sqsubset_{refl} \left(\mathbb{N}^{k+1}\right)^{\oplus}$ (5) $\mathbb{N}^k \sqsubset_{refl} \Sigma^*$ $\left(\text{for } |\Sigma| > 1\right)$

(3) $\left(\mathbb{N}^k\right)^* \sqsubset_{refl} \left(\mathbb{N}^{k+1}\right)^*$

Proof. The non-strict relations in (1)–(3) are clear, and (4) comes from Proposition 1. For (5), $\varphi(n_1, \ldots, n_k) = a^{n_1} b \ldots b a^{n_k}$ is a reflection. Strictness follows from Proposition 6 and the following order types, obtained according to the previous results: $ot(\mathbb{N}^k) = \omega^k$, $ot((\mathbb{N}^k)^{\oplus}) = \omega^{\omega^k}$, $ot((\mathbb{N}^k)^*) = \omega^{\omega^{\omega^k}}$, and $ot(\Sigma^*) = \omega^{\omega^{|\Sigma|-1}}$. □

4. Petri Nets and Lossy Channel Systems

The state spaces described by Proposition 3 are exactly those of Petri Nets and Lossy Channel Systems. We will look more closely at these systems to see the implication of this result regarding their expressiveness.

4.1. Petri Nets

Definition 6 *(Petri Nets).* A *Petri net* is a tuple $\langle P, T, \mathsf{Pre}, \mathsf{Post}, \Sigma, \lambda\rangle$ where:

- P is a finite set of places,
- T is a finite set of transitions,
- Pre and Post are mappings from $P \times T$ to $\mathbb{N}$,
- Σ is a finite alphabet of labels, and
- $\lambda : T \to \Sigma_\varepsilon$ is the labelling function.

A marking of a Petri Net is usually defined as a mapping from P to $\mathbb{N}$. We assume here that $P = \{1, \ldots, d\}$, and we see this marking as a vector $x \in \mathbb{N}^d$.

A transition t is enabled in the marking x if for all $p \in \{1, \ldots, d\}$, $x(p) \geqslant \mathsf{Pre}(p, t)$. The transition relation $\to$ of the WSTS associated with the Petri Net is defined by $x \xrightarrow{a} y$ if there is a transition $t \in T$ with $\lambda(t) = a$ enabled in x, such that for all $p \in \{1, \ldots, d\}$, $y(p) = x(p) - \mathsf{Pre}(p, t) + \mathsf{Post}(p, t)$.

We denote by PN (PN_k) the class of transition systems associated to Petri Nets (with k places). Then we have the following:

Theorem 3. *For any $k > 0$, $PN_k \not\preccurlyeq WSTS_{\mathbb{N}^{k-1}}$.*

Proof. We remark that the WSTS defined in the proof of Proposition 2 is actually the lossy semantics of a Petri Net when $X = \mathbb{N}^k$. This induces that we can take the non-lossy version of this Petri net, which is still a WSTS. Hence, PN_k is self-witnessing. Since $\mathbb{N}^k \not\sqsubseteq_{refl} \mathbb{N}^{k-1}$, we conclude by Proposition 4. □

Moreover, if we consider *Affine Well Nets* (*AWN*) [2] (an extension of Petri Nets with whole-place operations like transfers or resets), and denote by AWN_k the class of AWN with k unbounded places, we can obtain from the previous result the following simple consequence:

Corollary 2. *$PN_k \prec PN_{k+1} \not\preccurlyeq AWN_k$ for all $k > 0$.*

4.2. Lossy Channel Systems

Communicating Finite State Machines [19], FIFO Petri Nets [20] and FIFO automata [21] are (almost equivalent) models for systems of processes communicating through (perfect) FIFO (First In First Out) channels; all these models may simulate Turing machines by using the FIFO channel to simulate the tape and the transitions of a Turing machine. Given an alphabet (i.e., a finite set) M of messages, let A be the following (finite) set $A = (\{!, ?\} \times M) \cup \{\bot\}$ of (channel) elementary actions which can be of three different types: $(!, m)$, shortly written $!m$ (resp. $(?, m)$, shortly written $?m$) is the sending (resp. receiving) action of message m in (resp. from) a channel c (which will be specified); the action $\bot$ is an internal action which does not modify the channels: at each step, a FIFO automaton is able to make *at least* one action on each channel. Suppose now that there are k channels. A (vector) action $a \in A^k$ is a vector $a = (a_1, \ldots, a_k)$ of k elementary actions a_i, where each a_i is the unique action on channel c_i.

Definition 7 *(FIFO Automata).* A *FIFO Automaton* with k channels is a tuple $(Q, M, T, \delta, \Sigma, \lambda)$ where

- Q is a finite (and non-empty) set of states,
- M is a finite set of messages,
- T is a finite set of transitions,
- $\delta : T \to Q \times A^k \times Q$,
- Σ is a finite alphabet of labels, and
- $\lambda : T \to \Sigma_\varepsilon$ is the labelling function.

The set of configurations of a FIFO Automaton is $Q \times (M^*)^k$. Given two configurations $x = (p, u_1, \ldots, u_k)$ and $y = (q, v_1, \ldots, v_k)$, we may *fire* the transition $t \in T$ from x and we reach y, written $x \xrightarrow{\lambda(t)} y$, iff $\delta(t) = (p, a, q)$ and for every $i \in \{1, \ldots, k\}$, we have: $(a_i = \bot \Longrightarrow u_i = v_i)$, $(a_i = ?m \Longrightarrow u_i = m v_i)$ and $(a_i = !m \Longrightarrow u_i m = v_i)$.

The lossy semantics (as defined in Section 2) of a FIFO automaton is well known under the name of *Lossy Channel System* (LCS) [22]. *Completely specified protocols* [23,24] are a variant of LCS, and are sometimes called *front-lossy* Channel System because the messages of M can only be lost when they reach the front of the fifo channel.

We define $LCS(k, p)$ as the set of transition systems associated to lossy channel systems with k channels and p messages. A classic result is that one can encode many channels into one, as long as an additional character (a separator) becomes available for the channel alphabet.

Proposition 8. *Let $\mathcal{S} \in LCS(k,p)$ and x_0, x_f states of $\mathcal{S}$. Then there is $\mathcal{S}' \in LCS(1,p+1)$ and x'_0, x'_f states of $\mathcal{S}'$ such that $L(\mathcal{S}, x_0, x_f) = L(\mathcal{S}', x'_0, x'_f)$.*

Proof. Let M be the set of messages of $\mathcal{S}$. We order the k channels of $\mathcal{S}$, $C_1, \ldots, C_k$. We recursively define $C_{k+i} = C_i$. We keep a notion of "active channel" through the control states. We pick a message $\# \notin M$. Messages of $\mathcal{S}'$ are $M \cup \{\#\}$ and a state of $\mathcal{S}'$ is $(q, i, u_i \# u_{i+1} \# \ldots \# u_{i+k-1})$ where q is the original control state of $\mathcal{S}$, $1 \leqslant i \leqslant k$ is the current active channel and u_j is the content of channel C_j. Reading a character in C_i requires i to be the active channel; writing a character in C_i requires C_{i+1} to be the active channel.

The system can change the active channel from C_i to C_j $(j > i)$ at any time by iterating $j-i$ times the following sequence of ε-transitions:

- Write #.
- Read a word in M^* and copy it to the end of the channel.
- Read #.

As long as exactly $k-1$ separators # stay in the channel, the described system simulate $\mathcal{S}$. However, one can lose these separators. To remove spurious traces, we add a final checking procedure, starting from the final states of $\mathcal{S}$, that reads $k-1$ symbols # and, if successful, puts the system in its real final state. □

Thanks to our framework, we can sharpen this result by adding strict inclusions:

Theorem 4. $LCS(k,p) \prec LCS(k+1,p) \prec LCS(1,p+1)$.

Proof. $LCS(k,p) \preccurlyeq LCS(k+1,p)$ clearly holds. We have already shown that $LCS(k+1,p) \preccurlyeq LCS(1,p+1)$. For the strictness, we remark again that the WSTS introduced in the proof of Proposition 2 is actually a LCS, that is, given a state space $X = Q \times (M_p^*)^k$, we can find $\mathcal{S}$ in $LCS(k,p)$ and a witness L of X such that $\mathcal{S}$ recognizes L. This implies that $LCS(k,p)$ is self-witnessing. For all k and p, $ot(Q \times (M_p^*)^k) = \omega^{\omega^{p-1} * k} * |Q|$. This implies that $(M_p^*)^{k+1} \not\sqsubseteq_{refl} Q \times (M_p^*)^k$ and $M_{p+1}^* \not\sqsubseteq_{refl} Q \times (M_p^*)^k$ for all Q. To conclude we only need to apply Proposition 4. □

Moreover, in [14] the authors prove that $AWN \prec LCS$. We can easily get back the strictness:

Proposition 9. $LCS(1,2) \not\preccurlyeq AWN$.

Proof. As in the previous result, we remark that $LCS(1,2)$ is self-witnessing. Thus, we only need to apply Proposition 4, considering that for any $k > 0$, $M_2^* \not\sqsubseteq_{refl} \mathbb{N}^k$ (Corollary 1). □

This result is tight: $LCS(0,p) \simeq FA$ (Finite Automata), $LCS(k,1) \simeq PN_k$.

5. Petri Net extensions with data

Many extensions of Petri nets with data have been defined in the literature to gain expressive power for better modelling capabilities. Data Nets (*DN*) [4] are a monotonic extension of Petri nets in which tokens are taken from a linearly ordered and dense domain, and transitions can perform whole place operations like transfers, resets or broadcasts. A similar model, in which tokens can only be compared with equality, is that of ν-Petri Nets (ν-*PN*) [10]. The relative expressive power of *DN* and ν-*PN* has been an open problem since [11]. In this section we prove that ν-$PN \prec DN$. We work with the subclass of *DN* without whole place operations, called *Petri Data Nets* (*PDN*), since $DN \simeq PDN$ [14].

5.1. Petri Data Nets and ν-Petri Nets

5.1.1. Petri Data Nets

We denote by $\mathbf{0}$ the null vector in $\mathbb{N}^k$ for any k, and for a word $w = x_1 \cdots x_n$ we write $|w| = n$ and $w(i) = x_i$.

A *Petri Data Net* (*PDN*) is a Petri net where each token carries an *identity* from a linearly ordered and dense domain $\mathbb{D}$. A marking s of a *PDN* can be seen, e.g., as a multiset of pairs in $\mathbb{D} \times P$, or as a map $s \in (\mathbb{N}^P)^{\mathbb{D}}$. However, two key features of *PDN*s will guide our choice for marking representation:

1. a marking s only has finitely many tokens. Thus, denoting $d_1 < \cdots < d_m$ the identities that occur in s and gathering all tokens carrying the same identity d_i, one obtains a (non-null) place vector v_i in $\mathbb{N}^{|P|}$. Therefore, s can be written $(d_1, v_1) \cdots (d_m, v_m)$, implicitly associating the null vector $\mathbf{0}$ with any $d \in \mathbb{D} \setminus \{d_1, \ldots, d_m\}$;

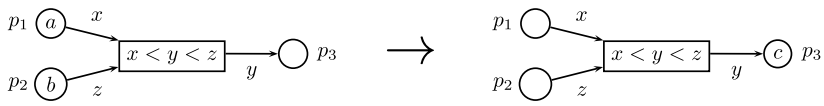

Fig. 1. Firing of a Petri data net transition (assuming $a < c < b$).

2. the concrete identities d_i are irrelevant, and only their relative *order* is useful w.r.t. the dynamics of the net. Thus, s can be safely abstracted as the sequence $v_1 \cdots v_m$ in $(\mathbb{N}^{|P|} \setminus \mathbf{0})^*$. (Also the choice for set $\mathbb{D}$ is irrelevant.)

Every transition t of a *PDN* specifies a sequence of n ordered potential identities and for any such identity specifies the tokens $\mathsf{Pre}(t)$ to be consumed and $\mathsf{Post}(t)$ to be produced. Thus, $\mathsf{Pre}(t)$ and $\mathsf{Post}(t)$ are two sequences of n (possibly null) place vectors.

Definition 8 *(Petri Data Nets).* A *k-dimensional Petri Data Net* (*k-PDN*) is a tuple $\mathcal{N} = (P, T, \mathsf{Pre}, \mathsf{Post}, \Sigma, \lambda)$, where:

- P is a finite set of $k = |P|$ places,
- T is a finite set of transitions with $P \cap T = \emptyset$,
- for every t in T, $\mathsf{Pre}(t)$ and $\mathsf{Post}(t)$ are finite sequences in $(\mathbb{N}^k)^*$ with $|\mathsf{Pre}(t)| = |\mathsf{Post}(t)|$,
- Σ is a finite alphabet, and
- $\lambda : T \to \Sigma_\epsilon$ is the labelling function.

Consider now a marking $s \in (\mathbb{N}^k \setminus \mathbf{0})^*$. In order to fire a transition t with $|\mathsf{Pre}(t)| = n$, one nondeterministically selects n identities, consumes some of their tokens as indicated by $\mathsf{Pre}(t)$, and produces new tokens with the identities specified by $\mathsf{Post}(t)$. However, some of these n identities might not be present in s, and we should introduce null vectors wherever necessary: $s' \in (\mathbb{N}^k)^*$ is a $\mathbf{0}$-*extension* of $s \in (\mathbb{N}^k \setminus \mathbf{0})^*$ (or s is the $\mathbf{0}$-*contraction* of s') $\overset{\text{def}}{\Leftrightarrow}$ s is obtained by removing all $\mathbf{0}$'s from s'. Once an extension s' is built, one selects in it a subword of n vectors $x_1, \ldots, x_n$ s.t. every vector contains enough tokens, i.e. with $x_i \geqslant \mathsf{Pre}(t)(i)$. If the condition is fulfilled, the corresponding tokens are consumed and $\mathsf{Post}(t)(i)$ is added to the resulting vector, yielding a new sequence s''. This s'' may contain null vectors, e.g. when all tokens with some identity have been consumed. Hence, the marking one really reaches is the $\mathbf{0}$-contraction of s''. Note that any way of firing t requires at most n insertions.

Definition 9 *(Transition system of a PDN).* Let $\mathcal{N}$ be a k-PDN. Then the labelled transition system $\mathcal{S}(\mathcal{N}) = \langle X, \Sigma, \to \rangle$ is defined by:

- $X = (\mathbb{N}^k \setminus \mathbf{0})^*$.
- Let $s, s' \in X$ and $t \in T$ with $n = |\mathsf{Pre}(t)|$. Then $s \xrightarrow{\lambda(t)} s'$ iff:
  1. there exists $u_0 x_1 u_1 \cdots u_{n-1} x_n u_n$ a $\mathbf{0}$-extension of s with $u_i \in (\mathbb{N}^k)^*$ and $x_i \in \mathbb{N}^k$ for all i;
  2. for $i \in \{1, \ldots, n\}$, $x_i \geqslant \mathsf{Pre}(t)(i)$;
  3. and defining $y_i = x_i - \mathsf{Pre}(t)(i) + \mathsf{Post}(t)(i)$, s' is the $\mathbf{0}$-contraction of $u_0 y_1 u_1 \cdots u_{n-1} y_n u_n$.

We rely on the standard graphical depiction of high level nets and use (pictures of) Petri nets where arcs connected to a transition t are labelled with bags of variables that must be instantiated by ordered identities. The number of these variables is exactly $|\mathsf{Pre}(t)|$ and the ordering of the corresponding identities is carried by the transition. For concision and readability, it is convenient to allow orderings of the variables that are not total: this stands for all possible linearizations. For instance, we can simulate a transition t in which two unrelated variables x and y appear, by having a non-deterministic choice between three transitions t_1, t_2 and t_3, the first one assuming $x < y$, the second one assuming $y < x$ and the last one with y substituted by x. Analogously, a transition with variables x and y so that $x \leqslant y$, can be simulated by two transitions one assuming $x < y$ and the other one with y substituted by x.

Using these graphical conventions, Fig. 1 depicts a *PDN* with a single transition t given by:

$$\mathsf{Pre}(t) = (1,0,0)(0,0,0)(0,1,0) \quad \text{and} \quad \mathsf{Post}(t) = (0,0,0)(0,0,1)(0,0,0)$$

(with places of P ordered by their index).

5.1.2. ν-Petri Nets

ν-Petri Nets can be seen as a restriction of *PDN*s where the domain of identities $\mathbb{D}$ still infinite is now unordered. In this restricted framework, we need variables to establish a correspondence between the identities of the tokens in the different places. We introduce a countable set *Var* of variables including a subset of special variables $\Upsilon \subset Var$ with $|\Upsilon| = |Var \setminus \Upsilon| = \infty$. The role of Υ is to select identities that are not present in the current marking.

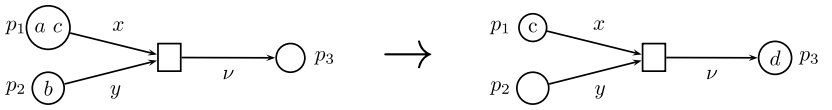

Fig. 2. Firing in a ν-Petri net (with $\sigma(x) = a$, $\sigma(y) = b$, $\sigma(\nu) = d$).

Definition 10 *(ν-Petri net).* A *ν-Petri net* is a tuple $\mathcal{N} = \langle P, T, \mathsf{Pre}, \mathsf{Post}, \Sigma, \lambda \rangle$, where:

- P is a finite set of places,
- T is a finite set of transitions with $P \cap T = \emptyset$,
- for every $p \in P$ and every $t \in T$, $\mathsf{Pre}(p,t) \in (Var \setminus \Upsilon)^{\oplus}$ and $\mathsf{Post}(p,t) \in Var^{\oplus}$,
- Σ is a finite alphabet, and
- $\lambda : T \to \Sigma_{\epsilon}$ is the labelling function.

$Var(t)$ denotes the set of variables x for which there is a place p such that $(\mathsf{Pre}(p,t) + \mathsf{Post}(p,t))(x)$ is not empty. A marking is a mapping $M : P \to \mathbb{D}^{\oplus}$.

Definition 11 *(Transition system of a ν-Petri net).* Let $\mathcal{N}$ be a ν-Petri net. Then, the labelled transition system $\mathcal{S}(\mathcal{N}) = \langle X, \Sigma, \to \rangle$ is defined by:

- $X = (\mathbb{D}^{\oplus})^P$.
- Let $s, s' \in X$ and $t \in T$. Then $s \xrightarrow{\lambda(t)} s'$ iff there exists an injection $\sigma : Var(t) \to \mathbb{D}$ such that for every $p \in P$:
  1. for every $x \in Var(t) \setminus \Upsilon$, $s(p)(\sigma(x)) \geqslant \mathsf{Pre}(p,t)(x)$ and $s'(p)(\sigma(x)) = s(p)(\sigma(x)) - \mathsf{Pre}(p,t)(x) + \mathsf{Post}(p,t)(x)$,
  2. for every $\nu \in \Upsilon \cap Var(t)$, $s(p)(\sigma(\nu)) = 0$ and $s'(p)(\sigma(\nu)) = \mathsf{Post}(p,t)(\nu)$,
  3. for every $d \in \mathbb{D} \setminus \sigma(Var(t))$, $s'(p)(d) = s(p)(d)$.

The graphical representation of a ν-Petri net is similar to that of a Petri net with expressions in $(Var)^{\oplus}$ defining the incidence matrices labelling the arcs of the net. Fig. 2 illustrates the firing of a transition in such nets. Observe that $\sigma(\nu)$ cannot belong to $\{a, b, c\}$.

In ν-Petri Nets, markings can be identified up to renaming of identities. Thus, markings of a *ν-PN* with k places can be represented as elements in $(\mathbb{N}^k)^{\oplus}$, each tuple representing the occurrences in each place of one identity [25]. For instance, if $P = \{p_1, p_2\}$ and marking s is such that $s(p_1) = \{|a, a, b|\}$ and $s(p_2) = \{|b|\}$, then its *abstract* representation is $\{|(2,0), (1,1)|\}$.

5.1.3. Classes of nets

Given a net $\mathcal{N}$ with identities and an initial marking, a place p of $\mathcal{N}$ is *bounded* if there exists some positive integer b such that for every reachable marking and identity, the number of tokens in p carrying this identity is at most b. Therefore, a bounded place may contain arbitrarily many identities, provided each of them appears an *a priori* bounded number of times. If a *PDN* (resp. a ν-Petri net) has k unbounded places and m places bounded by some b, then we can use as state space $(Q \times \mathbb{N}^k)^*$ (resp. $(Q \times \mathbb{N}^k)^{\oplus}$) with $Q = \{0, \ldots, b\}^m$.

We denote the class of initialized *PDN* with k unbounded places by PDN_k and their state space by $\mathbf{X}_k^* = \{(Q \times \mathbb{N}^k)^* \mid Q \text{ finite}\}$. We denote the class of initialized *ν-PN* with k unbounded places by *ν-PN*$_k$ and their state space by $\mathbf{X}_k^{\oplus} = \{(Q \times \mathbb{N}^k)^{\oplus} \mid Q \text{ finite}\}$. Moreover, we take $\mathbf{X}^* = \{(\mathbb{N}^k)^* \mid k > 0\}$ and $\mathbf{X}^{\oplus} = \{(\mathbb{N}^k)^{\oplus} \mid k > 0\}$.

5.2. Self-witnesses and consequences

Proposition 10. *For every $k \geqslant 0$, ν-PN$_k$ and PDN$_k$ are self-witnessing.*

Proof. We start with *ν-PN*$_k$. Let $(Q \times \mathbb{N}^k)^{\oplus} \in \mathbf{X}_k^{\oplus}$. We consider an alphabet $\Sigma = \{a_q \mid q \in Q\} \cup \{a_1, \ldots, a_k\}$ and we define $\gamma : \Sigma^* \to (Q \times \mathbb{N}^k)^{\oplus}$ by:

$$\gamma\left(a_{q_1} a_1^{n_1^1} \ldots a_k^{n_1^k} \ldots a_{q_l} a_1^{n_l^1} \ldots a_k^{n_l^k}\right) = \left\{\left|\left(q_1, n_1^1, \ldots, n_1^k\right), \ldots, \left(q_l, n_l^1, \ldots, n_l^k\right)\right|\right\}$$

Let us build N in *ν-PN*$_k$ such that $L(N) \cap dom(\gamma)\overline{dom(\gamma)} = L_{\gamma}$. Assume $Q = \{q_1, \ldots, q_r\}$. Fig. 3 shows the case with $k = 1$ and $r = 2$.

The only unbounded places of N are $p_1, \ldots, p_k$ (hence $N \in$ *ν-PN*$_k$). We consider $q_1, \ldots, q_r$ as places, a place st that stores all the names that have been used (once each name, hence bounded), and places $c_0, c_1, \ldots, c_k$ containing one name in mutual exclusion. When the name is in c_0 it is non-deterministically copied in some q (action labelled by a_q), and moved to c_1. For every, $1 \leqslant i \leqslant k$, when the name is in c_i it can be copied arbitrarily often to p_i (action labelled by a_i). At any

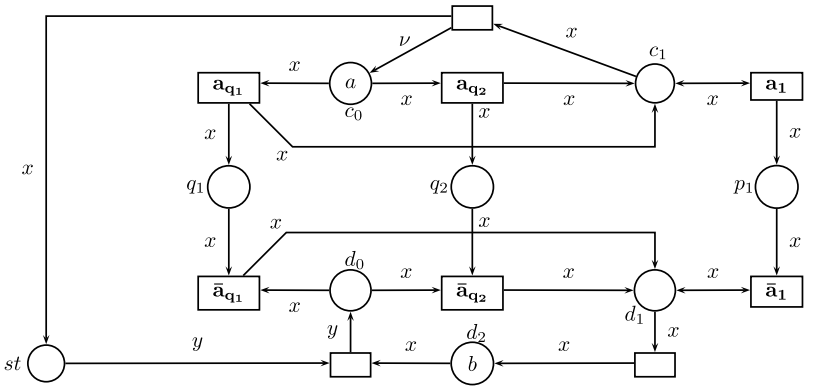

Fig. 3. Net in ν-PN_1 recognizing a witness of $(Q \times \mathbb{N})^{\oplus}$ with $|Q| = 2$.

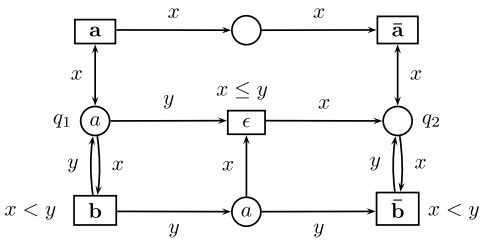

Fig. 4. PDN recognizing a witness of $\mathbb{N}^*$.

time, this name can be transferred to c_{i+1} when $i < k$ or to st for $i = k$ (action labelled by ϵ). In the last case a fresh name is put in c_0 (thanks to $\nu \in \Upsilon$).

The second phase is analogous, with control places $d_0, d_1, \ldots, d_{k+1}$, marked in mutual exclusion with names taken from st. At any point, the name in d_{k+1} can be removed, and one name moved from st to d_0 (action labelled by ϵ). That name must appear in some q. Thus, for each q we have a transition that removes the name from d_0 and q and puts it in d_1 (action labelled by $\bar{a}_q$). For each $1 \leqslant i \leqslant k$, the name in d_i can be removed zero or more times from p_i (action labelled by $\overline{a_i}$). At any point, the name is transferred from d_i to d_{i+1} (actions labelled by ϵ).

The initial and final marking is that with a name in c_0 and another name in d_{k+1} (and empty elsewhere). It holds that $L(N) \cap dom(\gamma)\overline{dom(\gamma)} = L_\gamma$, so we conclude.

The case of PDN_k is analogous to that of ν-PN_k. Let $(Q \times \mathbb{N}^k)^* \in \mathbf{X}_k^*$. We define $\Sigma = \{a_q \mid q \in Q\} \cup \{a_1, \ldots, a_k\}$ and $\gamma : \Sigma^* \to (Q \times \mathbb{N}^k)^*$ by:

$$\gamma\left(a_{q_1} a_1^{n_1^1} \ldots a_k^{n_1^k} \ldots a_{q_l} a_1^{n_l^1} \ldots a_k^{n_l^k}\right) = \left(q_1, n_1^1, \ldots, n_1^k\right) \cdots \left(q_l, n_l^1, \ldots, n_l^k\right)$$

The net N in PDN_k that we build is similar to the ν-PN we built in the case of ν-PN_k, except for two differences: On the one hand, whenever a fresh name was put in c_0, now we put a *greater* name (that is, we replace ν by a variable y such that $x < y$). On the other hand, whenever we took from st another name, now we take a greater name (that is, we assume $x < y$). Finally, the initial and final marking is that with one name in c_0 and a smaller name in d_{k+1}. Again, it holds that $L(N) \cap dom(\gamma)\overline{dom(\gamma)} = L_\gamma$, and we conclude. □

Fig. 4 shows a *PDN* recognizing a witness of N^*. Notice that since ν-PN_k and PDN_k are self-witnessing for every $k \geqslant 0$, so are ν-*PN* and *PDN*.

Proposition 11. $\mathbf{X}_1^* \not\sqsubseteq_{refl} \mathbf{X}^{\oplus}$, $\mathbf{X}_{k+1}^{\oplus} \not\sqsubseteq_{refl} \mathbf{X}_k^{\oplus}$ *and* $\mathbf{X}_{k+1}^* \not\sqsubseteq_{refl} \mathbf{X}_k^*$ *for all* k.

Proof. $\mathbf{X}_1^* \not\sqsubseteq_{refl} \mathbf{X}^{\oplus}$ holds because $ot(\mathbb{N}^*) = \omega^{\omega^\omega} \not\leqslant \omega^{\omega^k} = ot((\mathbb{N}^k)^{\oplus})$, so that $\mathbb{N}^* \not\sqsubseteq_{refl} (\mathbb{N}^k)^{\oplus}$ for all k. The others are obtained similarly, considering that $ot((Q \times \mathbb{N}^k)^{\oplus}) = \omega^{\omega^k * |Q|}$ and $ot((Q \times \mathbb{N}^k)^*) = \omega^{\omega^{\omega^k * |Q|}}$. □

Corollary 3. ν-*PN* $\prec$ *PDN. Moreover,* $PDN_1 \not\preccurlyeq \nu$-*PN.*

Proof. ν-*PN* $\preccurlyeq$ *PDN* is from [11]. $PDN_1 \not\preccurlyeq \nu$-*PN* is a consequence of Proposition 4, considering that both classes are self-witnessing, and that $\mathbf{X}_1^* \not\sqsubseteq_{refl} \mathbf{X}^{\oplus}$. □

We can even be more precise in the hierarchy of Petri Nets extensions.

Proposition 12. *For any $k \geqslant 0$, ν-$PN_k \prec \nu$-PN_{k+1} and $PDN_k \prec PDN_{k+1}$.*

Proof. Clearly ν-$PN_k \preccurlyeq \nu$-PN_{k+1} and $PDN_k \preccurlyeq PDN_{k+1}$ for any $k \geqslant 0$. For the converses, again we can apply Proposition 4, considering that all the classes considered are self-witnessing and that $\mathbf{X}^{\oplus}_{k+1} \not\sqsubseteq_{refl} \mathbf{X}^{\oplus}_{k}$ and $\mathbf{X}^{*}_{k+1} \not\sqsubseteq_{refl} \mathbf{X}^{*}_{k}$ hold. □

Finally, we can strengthen the result $AWN \prec \nu$-PN proved in [11] in a very straightforward way.

Proposition 13. *ν-$PN_1 \not\preccurlyeq AWN$.*

Proof. ν-PN_1 is self-witnessing, and $\mathbf{X}^{\oplus}_{1} \not\sqsubseteq_{refl} \{\mathbb{N}^k \mid k > 0\}$ because $\mathbb{N}^{\oplus} \not\sqsubseteq_{refl} \mathbb{N}^k$ for all k (indeed, $ot(\mathbb{N}^{\oplus}) = \omega^{\omega} \not\leqslant \omega^k = ot(\mathbb{N}^k)$). By Proposition 4 we conclude. □

Again, the previous result is tight. Indeed, a ν-PN with no unbounded places can be simulated by a Petri net, so that ν-$PN_0 \simeq PN$.

6. Timed Petri nets

A timed Petri net [12] is a Petri net whose tokens have an age that evolves synchronously with time elapsing. The transition system of such a net has two kinds of transitions. Either time elapses (with no restriction) and all the token ages are updated accordingly; or a net transition is fired, consuming and producing tokens as in ordinary nets. However the ages of tokens to be consumed can be required to belong to time intervals while the ages of tokens to be produced can be selected non-deterministically in time intervals. Consequently, the arcs of a timed Petri net are labelled by multisets of intervals. In this section, $\mathcal{I}$ is the set of intervals with bounds in $\mathbb{N} \cup \{\infty\}$.

Definition 12 *(Timed Petri nets).* A *timed Petri net* (*TdPN*) $\mathcal{N}$ is a tuple $(P, T, \mathsf{Pre}, \mathsf{Post}, \lambda)$ where:

- P is a finite set of places,
- T is a finite set of transitions with $P \cap T = \emptyset$,
- Pre, the backward incidence mapping, is a mapping from T to $(\mathcal{I}^{\oplus})^P$,
- Post, the forward incidence mapping, is a mapping from T to $(\mathcal{I}^{\oplus})^P$,
- $\lambda : T \to \Sigma_{\varepsilon}$ is a labelling function.

Since $(\mathcal{I}^{\oplus})^P$ is isomorphic to $(P \times \mathcal{I})^{\oplus}$, $\mathsf{Pre}(t)$ and $\mathsf{Post}(t)$ may also be considered as multisets. Given a place p and a transition t, if the multiset $\mathsf{Pre}(t)(p)$ (resp. $\mathsf{Post}(t)(p)$) is non-null then it defines a *pre-arc* (resp. *post-arc*) of t connected to p.

A *configuration* μ of a *TdPN* is an item of $(\mathbb{R}^{\oplus}_{\geqslant 0})^P$ (or equivalently $(P \times \mathbb{R}_{\geqslant 0})^{\oplus}$). Intuitively, a configuration is a marking extended with age information for the tokens. We will write (p, τ) for a token which is in place p and whose age is τ. A configuration is then a finite sum of such pairs. A token (p, τ) then *belongs* to the configuration μ whenever $(p, \tau) \leqslant \mu$ (in terms of multisets). For a configuration μ and $d \in \mathbb{R}_{\geqslant 0}$ we write $\mu + d$ to denote the configuration obtained from μ by increasing the age of all tokens by d. Given a configuration $\mu \in (P \times \mathbb{R}_{\geqslant 0})^{\oplus}$ and a multiset $f \in (P \times \mathcal{I})^{\oplus}$, we say that μ *satisfies* f, and write $\mu \models f$, if and only if there exists a multiset $x \in (P \times \mathbb{R}_{\geqslant 0} \times \mathcal{I})^{\oplus}$ verifying the following conditions:

$$\begin{cases} \pi_{1,2}(x) = \mu \\ \pi_{1,3}(x) = f \\ \forall (p, \tau, I) \in sup(x), \quad \tau \in I \end{cases}$$

Here $\pi_{i,j}$ is the mapping that, given a multiset of tuples x, outputs the multiset of pairs corresponding to the projection of the tuples over their ith and jth components. The intuition underlying the satisfaction relation is that a multiset of aged tokens exactly corresponds to a multiset of timed requirements specified by intervals.

We now describe the semantics of a *TdPN* as a transition system. As discussed above, this system consists of timed and discrete transitions. The timed transitions are silent transitions (i.e. labelled by ε).

Definition 13 *(Transition system of a TdPN).* Let $\mathcal{N}$ be a *TdPN*. The labelled transition system $\mathcal{S}(\mathcal{N}) = \langle X, \Sigma, \to \rangle$ is defined by:

- $X = (P \times \mathbb{R}_{\geqslant 0})^{\oplus}$.
- The transitions are defined as follows:
  1. For each $d \in \mathbb{R}_{\geqslant 0}$, there is a delay transition $\mu \xrightarrow{\varepsilon} \mu + d$.

2. Given a transition $t \in T$ and two configurations $\mu, \mu' \in (P \times \mathbb{R}_{\geqslant 0})^{\oplus}$, we write $\mu \xrightarrow{\lambda(t)} \mu'$, if and only if there exist two multisets ${}^\bullet\mu, \mu^\bullet \in (P \times \mathbb{R}_{\geqslant 0})^{\oplus}$ such that:

$$\begin{cases} {}^\bullet\mu \models \mathsf{Pre}(t) \\ \mu^\bullet \models \mathsf{Post}(t) \\ {}^\bullet\mu \leqslant \mu \\ \mu' = \mu - {}^\bullet\mu + \mu^\bullet \end{cases}$$

The intuition of the previous definition is as follows: ${}^\bullet\mu$ is the set[3] of tokens which is removed from the configuration μ when firing transition t, whereas $\mu^\bullet$ is the set of tokens that are created by the transition firing. Moreover, the ages of all these tokens need to satisfy the constraints specified by the various arcs (conditions written using the $\models$ operator defined above). Finally, the new configuration is given by μ' computed as $\mu' = \mu - {}^\bullet\mu + \mu^\bullet$.

A *path* in the *TdPN* $\mathcal{N}$ is a sequence $\mu_0 \xrightarrow{\varepsilon} \mu_0 + d_1 \xrightarrow{\lambda(t_1)} \mu_1 \xrightarrow{\varepsilon} \mu_1 + d_2 \xrightarrow{\lambda(t_2)} \mu_2 \dots$ in the above transition system, which alternates between delay and discrete transitions. A *timed transition sequence* is a finite timed word over the alphabet T, the set of transitions of $\mathcal{N}$. A *firing sequence* is a timed transition sequence $(t_1, \tau_1)(t_2, \tau_2)\dots$ such that $\mu_0 \xrightarrow{\varepsilon} \mu_0 + \tau_1 \xrightarrow{\lambda(t_1)} \mu_1 \xrightarrow{\varepsilon} \mu_1 + (\tau_2 - \tau_1) \xrightarrow{\lambda(t_2)} \mu_2 \dots$ is a path. If $(p, \tau) \leqslant \mu$ is a token of a configuration μ, it is a *dead token* whenever for every interval I labelling a pre-arc of p, τ is strictly greater than the upper bound of I. This means that this token cannot be used anymore by a pre-arc to fire a transition. The untimed word which is read along a path $\mu_0 \xrightarrow{\varepsilon} \mu_0 + d_1 \xrightarrow{\lambda(t_1)} \mu_1 \xrightarrow{\varepsilon} \mu_1 + d_2 \xrightarrow{\lambda(t_2)} \mu_2 \dots$ is the projection over Σ of the timed word, i.e., $\lambda(t_1)\lambda(t_2)\dots$.

Definition 14 *(Untimed language of a TdPN).* Let $\mathcal{N}$ be a *TdPN* and μ_0, μ_f be two configurations with *integer* ages. Then $L(\mathcal{N}, \mu_0, \mu_f)$ is the *untimed coverability language* associated with $\mathcal{S}(\mathcal{N})$, μ_0 the initial configuration and μ_f the configuration to be covered.

We have required that the ages of tokens in the two configurations are integer. Indeed, in order to represent these configurations, we could simply require that the ages are rational. However with a standard change of scale time, rationals can be transformed to integers (both in the configurations and in the net) without modifying the language.

Now the key observation is that w.r.t. the untimed language of a *TdPN* (as in timed automata), it is sufficient to look at an abstraction of the configurations, called regions. By max we denote the maximal integer appearing in the bounds of intervals of the net and in the age of tokens in μ_0 and μ_f. In the following, we denote by $\mathbf{0}$ the empty multiset $\{|\ |\}$.

Definition 15 *(Regions of TdPNs).* A *region* $\mathcal{R}$ for a *TdPN* $\mathcal{N}$ is a sequence $a_0 a_1 \dots a_n a_\infty$ where $n \in \mathbb{N}$ and:

- $a_0 \in (P \times F_{\max})^{\oplus}$ with $F_{\max} = \{0, 1, \dots, \max\}$;
- for all $0 < i \leqslant n$, $a_i \in (P \times F_{\max-1})^{\oplus}$ with $F_{\max-1} = \{0, 1, \dots, \max-1\}$ and $a_i \neq \mathbf{0}$;
- $a_\infty \in (P \times F_\infty)^{\oplus}$ with $F_\infty = \{\infty\}$.

We informally explain the semantics of a region. Given the multiset of tokens defining a configuration, we obtain its associated region as follows. We put in a_∞ all the tokens whose ages are strictly greater than max and forget their ages. We then put in a_0 the tokens with integral ages and add the information about their ages. Finally, we order the remaining tokens depending on the fractional part of their ages in $a_1, \dots, a_n$, forget their fractional part, and only store the integral part of their ages. Hence n is the number of different positive fractional values for ages of the remaining tokens. For instance, consider the multiset of tokens $(p, 1) + (p, 2.8) + (q, 0.8) + (q, 5.1) + (r, 1.5)$. Then, if the maximal constant is 4, its region encoding will be $a_0 a_1 a_2 a_\infty$ where $a_0 = (p, 1)$ (because there is a single token with integral age), $a_\infty = (q, \infty)$ (because the age of token $(q, 5.1)$ is 5.1, hence above the maximal constant), $a_1 = (r, 1)$ (among all fractional parts, 0.5 is the smallest one), and $a_2 = (p, 2) + (q, 0)$ (all tokens with fractional part 0.8).

Furthermore we can define an (infinite but countable) transition system over regions that generate the untimed words of the net. Rather than giving a formal cumbersome definition, we informally present it:

- We associate silent transitions with time elapsing. Since we can split the time elapsing, we consider two kinds of such transitions.
  1. Given a region $a_0 a_1 \dots a_n a_\infty$, when $a_0 \neq \mathbf{0}$ we first partition $a_0 = b_0 + c_0$ where b_0 (resp. c_0) is the multiset of tokens with age strictly less than max (resp. equal to max). The new region is now the word $\mathbf{0} b_0 a_1 \dots a_n (a_\infty + c_0')$ where c_0' is the bag of tokens of c_0 with max substituted by ∞. This transition corresponds to a small time elapsing that does not let the ages of tokens of a_n to reach or overcome an integral value.

[3] This is a language misuse, the right term should be "multiset", as there can be several tokens with the same age.

2. Given a region $\mathbf{0}a_1 \dots a_n a_\infty$ when $n > 0$, the new region is now the word $b_0 a_1 \dots a_{n-1} a_\infty$ where b_0 is the bag of tokens of a_n with their integral component increased by one. This transition corresponds to the time elapsing that lets the ages of tokens of a_n reach an integral value.

- The information associated with the age of tokens in a region is sufficient to know whether they belong to an interval labelling a pre-arc. So given a region $a_0 a_1 \dots a_n a_\infty$, in order to fire t:
  1. We must constitute a word $b_0 b_1 \dots b_n b_\infty$ with $b_i \leqslant a_i$ for every $i \in \{0, \dots, n\} \cup \infty$ such that for every place p there is a bijective mapping from the intervals of the multiset $\mathsf{Pre}(t)(p)$ to the tokens labelled by p in $b_0, b_1, \dots, b_n, b_\infty$. The first step of the firing consists then in deleting these tokens, leading to an intermediate region $c_0 c_1 \dots c_n c_\infty = (a_0 - b_0)(a_1 - b_1) \cdots (a_n - b_n)(a_\infty - b_\infty)$ where the c_i's for $1 \leqslant i \leqslant n$ such that $c_i = \mathbf{0}$ are then deleted.
  2. Then for every place p and every interval of the multiset $\mathsf{Post}(t)(p)$, we choose a token whose fractional part may be either null, either a non-null existing one or a new non-null one, in this last case increasing n and choosing any position in the fractional order. The choice must lead to an age belonging to the interval. These new tokens "added" to $c_0 c_1 \dots c_n c_\infty$ lead to the region reached by this firing of t (as there are non-deterministic choices, several but finitely many firings of t are possible).

Given $\mathcal{R}_0$ (resp. $\mathcal{R}_f$) the abstraction of μ_0 (resp. μ_f), it is routine to check that the corresponding coverability language is exactly $L(\mathcal{N}, \mu_0, \mu_f)$. Furthermore, the state space of this abstract transition system is a wpo and this system is a WSTS. Since the abstract transition rule is effective, the family of untimed languages of *TdPN* fulfils the same standard decidability properties as the ones already presented. We refer to [12,26] for more information.

Looking more carefully at this state space it appears to be isomorphic to the one of a Petri Data Net. This suggests that these models could be equally expressive w.r.t. their coverability languages and this is what we prove in the next theorems.

Theorem 5. *Let $\mathcal{N}$ be a Petri Data Net and m_0, m_f be two markings of $\mathcal{N}$. There exists $\mathcal{N}'$ a TdPN and two configurations μ_0, μ_f such that $L(\mathcal{N}', \mu_0, \mu_f) = L(\mathcal{N}, m_0, m_f)$.*

Proof. Let us first describe the principles of the simulation. Places of $\mathcal{N}'$ will contain two kinds of tokens: the tokens of age belonging to $[0, 1]$ will be *relevant* while the older tokens will be irrelevant. We define the relevant part of a marking of $\mathcal{N}'$ as the marking where the irrelevant tokens have been deleted.

The simulation of a transition firing will last 1 time unit (t.u.), so the markings of $\mathcal{N}'$ at instants $0, 1, 2, \dots$ are the basis of the simulation.

Our simulation is lossy in the following sense. If there is a firing sequence $m_0 \xrightarrow{\sigma} m$ in $\mathcal{N}$, then there is at least one *perfect simulation* $\mu_0 \xrightarrow{\sigma} \mu$ in $\mathcal{N}'$ with the same associated word. Furthermore all firing sequences of $\mathcal{N}'$ will be perfect or lossy simulations. A *lossy simulation* is a sequence that leads to markings at integer instants whose relevant parts are covered by the relevant part of markings reached by a perfect simulation with the same associated word.

For technical reasons, a place p of $\mathcal{N}$ will be simulated by two places p_0, p_1 of $\mathcal{N}'$. Let $m_0 \xrightarrow{\sigma} m$ be a firing sequence in $\mathcal{N}$, with n current identities $x^1 < \cdots < x^n$ in m and denote m by the word $(\sum_{p \in P} \lambda_p^1 \cdot p) \cdots (\sum_{p \in P} \lambda_p^n \cdot p)$.

Let $\mu_0 \xrightarrow{\sigma} \mu$ be some perfect simulation of σ in $\mathcal{N}'$. There will be exactly n fractional parts of ages of relevant tokens in μ. Assume that the length of the firing sequence σ is even (resp. odd). Let us denote $a_0 a_1 \dots a_n a_\infty$ be the region associated with μ. Then the word a_i fulfils $a_i = \sum_{p \in P} \lambda_p^i \cdot (p_0, 0)$ (resp. $a_i = \sum_{p \in P} \lambda_p^i \cdot (p_1, 0)$). a_0 will contain tokens of control places (to be detailed later) and a_∞ will be equal to $\mathbf{0}$.

Let us describe the control places:

- $time_0, time_1$ are the places that schedule the operations. At an even (resp. odd) instant, place $time_0$ (resp. $time_1$) has a token with age 0. Then after one t.u., a transition tt_0 (resp. tt_1) ending the simulation process is fired getting this token and producing a token with age 0 in $time_1$ (resp. $time_0$).
- Place $idle_0$ (resp. $idle_1$) has a token only present at even (resp. odd) instants. The consumption of this token by transition t_0 (resp. t_1) starts the simulation process of transition t. When a simulation is started, a token (whose time is irrelevant) is produced in place $trans_0$ (resp. $trans_1$). This token enables to transfer relevant tokens that *will not be used* in the transition firing. When such a token (say in place q_0) has age 1 it is consumed by $tr.q_0$ and a token is produced in q_1. At the end of the simulation the token has the same age as the original one at the beginning of the simulation. Observe that some tokens may be *forgotten* (case of a lossy simulation). These forgotten tokens cannot be used in the sequel since their ages become greater than 1.
- Let us recall that all variables occurring in a transition t of $\mathcal{N}$ are totally ordered. Thus the transition simulation consumes and produces the tokens required by variables, beginning by the greatest variable. Let us illustrate this simulation in the example of Fig. 5. The token "with identity Z" in place q which must be consumed will be the first one to reach age 1, so it is deleted by transition $simZ.t_0$. Then transition $simY.t_0$ produces the token "with identity Y" in place s_1. Finally, transition $simX.t_0$ consumes the token "with identity X" in place p_0 and simultaneously produces the token in place r_1. Observe that these transitions must let time elapse due to the interval constraints. This avoids to use the same identity for X, Y and Z.

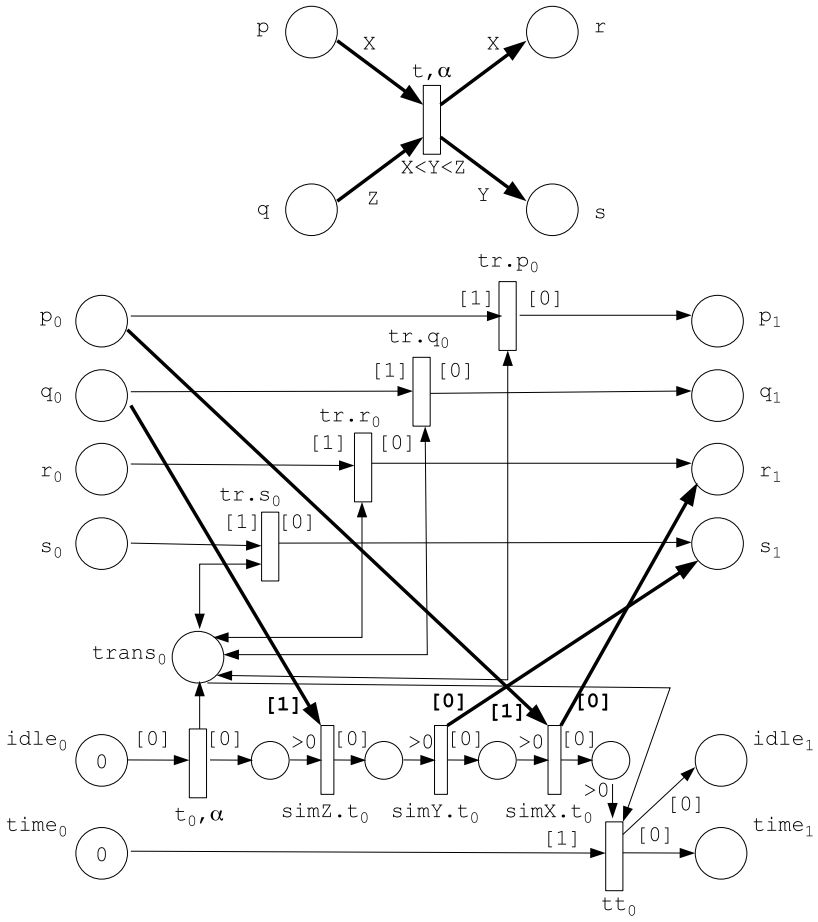

Fig. 5. Simulation of a transition at even instants.

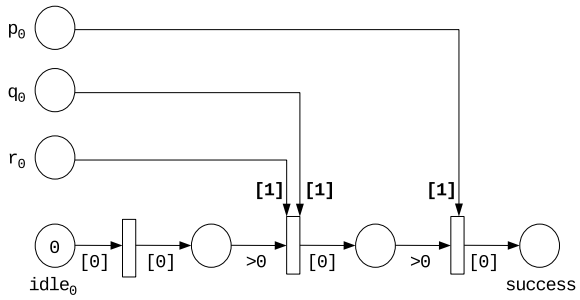

Fig. 6. Test of covering $p(q+r)$ at even instants.

Let us now explain by an example (see Fig. 6) how to check coverability of marking $p(q+r)$ in $\mathcal{N}$. At even (resp. odd) instants one consumes the token in $idle_0$ (resp. $idle_1$) and proceeds to test the coverability. First one lets time elapse until we obtain a token in q and r with age 1. Then after some time elapsing we must obtain a token in p with age 1 and we conclude positively by covering marking $(success, 0)$. The generalization is straightforward.

The specification of the initial marking of $\mathcal{N}'$ is immediate and left to the reader. □

In order to prove the reverse implication, we recall that for any *TdPN* $\mathcal{N}$ there is a *TdPN* $\mathcal{N}'$ with the same language and such that the only interval occurring in post-arcs is $[0, 0]$ (Theorem 4 of [27]).

Theorem 6. *Let $\mathcal{N}$ be a TdPN and two configurations μ_0, μ_f (with integer ages) of $\mathcal{N}$. Then there exists $\mathcal{N}'$ a Petri Data Net and m_0, m_f two markings of $\mathcal{N}'$ such that $L(\mathcal{N}', m_0, m_f) = L(\mathcal{N}, \mu_0, \mu_f)$.*

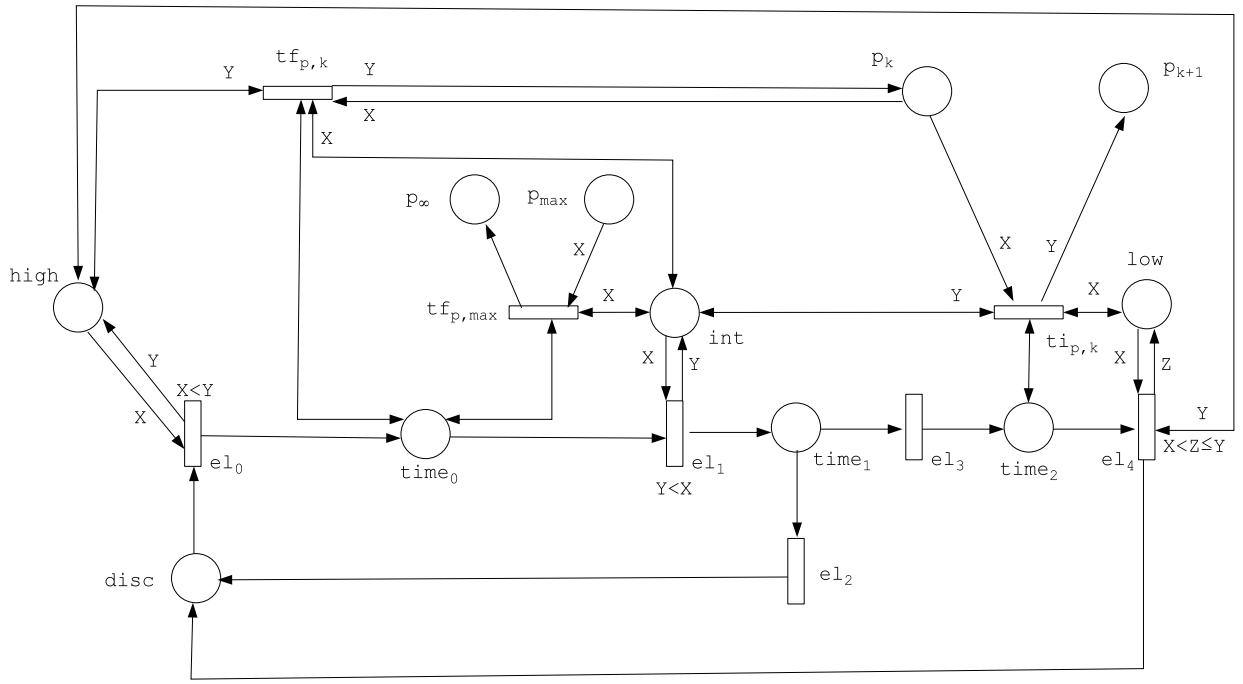

Fig. 7. Simulation of time elapsing.

Proof. As in the previous proof, our simulation is a lossy simulation allowing to "lose" tokens of $\mathcal{N}$ in the simulating net $\mathcal{N}'$, as it does not change the coverability language. We first describe the principle of the simulation.

After some initialization stage, places *low*, *high* and *int* always contain a single token. In the sequel of the proof we denote the identity contained in such a place by the name of the place. Every non-null fractional part of the current configuration of $\mathcal{N}$ is represented by an identity x such that: $low \leqslant x \leqslant high$. The order of such identities is the reverse order of the fractional part: for two identities of fractional parts $x < y$ the fractional part of x is greater than the fractional part of y. For every simulation, *low* is just a lower bound of the identity with the highest fractional part but if there is at least a token in $\mathcal{N}$ whose age has a non-null fractional part then there is always a simulation for which *low* is equal to this identity. Furthermore, identity *int* corresponds to tokens whose ages (less or equal than max) have a null fractional part.

During the simulation, *int* only decreases while *low* only increases, and as in the initialization step we ensure that $int < low$, this inequation will always be fulfilled. At any instant of the simulation, the identities that label tokens are between *int* and *high* and only tokens which have identities between *low* and *high* or equal to *int* are still relevant for the simulation.

Let max be the maximal constant occurring in $\mathcal{N}$, μ_0 and μ_f. For every place p of $\mathcal{N}$, $\mathcal{N}'$ has the following places: $p_0, p_1, \ldots, p_{\max}, p_\infty$. Place p_k contains the tokens of $\mathcal{N}$ in p with age less or equal than max and integral part equal to k. Place p_∞ contains the tokens of $\mathcal{N}$ in p with age greater than max; this place contains black tokens as the fractional part of the age is irrelevant for such tokens in $\mathcal{N}$.

During the simulation, place *disc* is either empty or contains a black token that allows the simulation of discrete transitions of $\mathcal{N}$. Let us first describe the simulation of time elapsing as illustrated in Fig. 7. Transition el_0 begins to perform the simulation of a small elapse of time whose only effect (see above the definition of the transition system over regions) is that there is no more tokens (with age less or equal than max) with integral ages. It increases *high* in order to assign this value to the tokens with integral ages. While $time_0$ is marked, transition $tf_{p,k}$ with $k <$ max "updates" tokens with integer age in p_k changing their identity to *high*. Transition $tf_{p,\max}$ transfers tokens with age max from p_k to p_∞. As said before, some tokens can be forgotten but they will not perturb the simulation since at the end of the transfer *int* is decreased (transition el_1). Then either we stop the time elapsing simulation (transition el_2) or proceed (transition el_3) to let an additional amount of time that corresponds to letting the tokens with greatest fractional part reach their next integral value by changing their identity from *low* to *int* and moving tokens from p_k to p_{k+1}. When *low* is different from the identity of tokens with greatest fractional part, no transfer occurs. At the end of a simulation, a new value is chosen for *low* greater than the former value and less or equal than *high* (transition el_4). When this choice corresponds to the identity of the new greatest fractional part the simulation is exact. Otherwise, the tokens whose fractional parts have associated identities less than *low* are "lost".

The simulation of a transition of $\mathcal{N}$ is straightforward. In order to simplify its presentation, we can assume w.l.o.g. that pre-arcs are labelled by multisets of intervals $[0],]0, 1[, [1], \ldots, [\max],]\max, \infty[$. This can be easily obtained by duplicating transitions (see for instance [27]). As said before, post-arcs are labelled by a multiset over interval [0]. Rather than defining

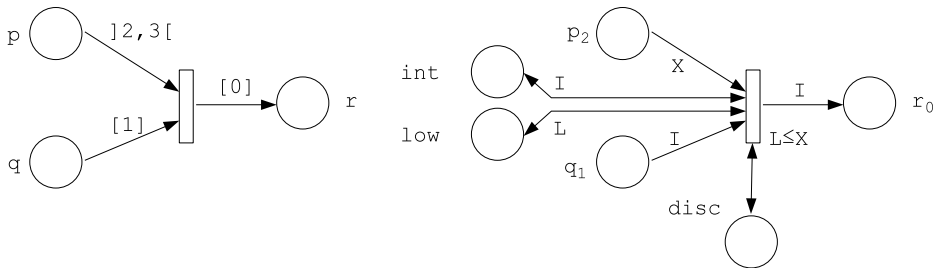

Fig. 8. Simulation of a transition.

Table 1
Summary of results.

Quantitative results. (All results are new)
For every $k \in \mathbb{N}$ $VASS_k \prec VASS_{k+1} \not\preccurlyeq AWN_k$
For every $k, p \in \mathbb{N}$ $LCS(k, p) \prec LCS(k+1, p) \prec LCS(1, p+1)$
For every $k \in \mathbb{N}$ ν-$PN_k \prec \nu$-PN_{k+1} and $PDN_k \prec PDN_{k+1}$

Qualitative results. (New results are ν-$PN \prec DN$ and $PDN \simeq TdPN$)
$VASS \prec \mathcal{M} \prec DN \simeq PDN \simeq TdPN$
where $\mathcal{M}$ is either ν-PN or LCS

it formally we illustrate the translation on Fig. 8. For instance, since the arc from p to t is labelled by $]2, 3[$, we are looking for a token in place p_2 with identity between *low* and *high*. The other cases are similar. Observe that since post-arcs are labelled by 0, there is no new fractional part. This avoids to handle the undesirable case where a new fractional part would be the greatest one, as it would require to decrease *low*, which is forbidden by our simulation.

Checking the coverability condition is performed by stopping the simulation and then consuming tokens in places p_k with identity *int* corresponding to the (integral-age valued) tokens of μ_f. □

7. Conclusion and perspectives

To show a strict hierarchy of WSTS classes, we have proposed a generic method based on two principles: the ability of WSTS to recognize some specific witness languages linked to their state space, and the use of order theory to show the absence of order reflections from one wpo to another. This allowed us to unify some existing results, while also solving open problems. We summarize the current picture on expressiveness of WSTS in Table 1 w.r.t. number of resources and type of resources. On the other hand, showing equivalence between WSTS classes is a problem deeply linked to the semantics of the models, and hence that remains to be solved on a case-by-case basis.

An interesting case that remains open is the relative expressiveness of *LCS* and ν-*PN*. Their state space are quite distinct but their order type are the same for some values of their parameters. We conjecture that there is no reflection from one to the other, but such a proof would require more than order type analysis.

All the models that we have studied in this paper use a state space whose order type is bounded by ϵ_0. However, the theory that we have developed can equally address state spaces with a greater state space. For instance, it is known that the Kruskal ordering has an order type greater than ϵ_0 [8], even for unlabelled binary trees. Thus, it is tempting to look at WSTS based on trees [28–30]. We believe some interesting problems might lie in this direction.

Appendix A. Complements to the proof of Proposition 1

We first introduce a few additional notations that we need for the proof of this section.[4]

Let A be a well-ordered set. $X \subseteq A$ is a directed subset of A if $\forall x, y \in X$, $\exists z \in X$, $x \leqslant z \wedge y \leqslant z$. A downward closed directed subset of A is called an irreducible ideal[5] of A. We denote by $Idl(A)$ the set of irreducible ideals of A.

Proposition 14. *Let A be a well-ordered set. Then any downward closed subset of A is a finite union of irreducible ideals.*

Note that $\nu_I : A \to Idl(A)$ given by $\nu_I(x) = {\downarrow}x$ is an order-embedding. Because of this, we will identify x with ${\downarrow}x$.

[4] Although the notations vary slightly from "Forward Analysis for WSTS: Part I: Completions" by A. Finkel and J. Goubault-Larrecq (STACS'09), the beginning of this section is a straight rewriting of results from this paper.

[5] Some authors have been using the term "ideal" as a shortcut for either a downward closed subset, or for a directed one. To avoid any confusion, we will only speak of irreducible ideals and of downward closed subsets.

Proposition 15. *Let $I \in Idl(\mathbb{N}^k)$. I can be written as $(x_1, \ldots, x_k)$ with $x_i \in \mathbb{N} \cup \{\omega\}$, and:*

$$(y_1, \ldots, y_k) \in (x_1, \ldots, x_k) \iff (\forall i,\ x_i \in \mathbb{N} \Longrightarrow y_i \leqslant x_i)$$

For example, $(\omega, 4)$ denotes the subset of $\mathbb{N}^2$ whose elements are those with 4 or lower as their second coordinate. This can be seen as an extension of the classic ordinal representation, where $\omega = \mathbb{N}$.

Proposition 16. *Let $I \in Idl(A^{\oplus})$. I can be written as $\{I_1^{\omega}, \ldots, I_p^{\omega}, J_1, \ldots, J_q\}$ where $I_1, \ldots, I_p, J_1, \ldots, J_q$ are irreducible ideals of A, and with*

$$x \in [\![\{I_1^{\omega}, \ldots, I_p^{\omega}, J_1, \ldots, J_q\}]\!] \iff \begin{cases} x = x_1 \cup \cdots \cup x_p \cup y_1 \cup \cdots \cup y_q \\ \forall 1 \leqslant k \leqslant p, \quad a \in x_k \Longrightarrow a \in I_k \\ \forall 1 \leqslant k \leqslant q, \quad y_k = \emptyset \vee (y_k = \{a\} \wedge a \in J_k) \end{cases}$$

For example $\{1^{\omega}, 3\}$ describes the subset of $\mathbb{N}^{\oplus}$ whose elements are those that contain any number of 0 or 1, and at most one element equal to 2 or 3. Note that an irreducible ideal has more than one possible representation. We have for example $\{2^{\omega}, 1\} = \{2^{\omega}\}$.

Proposition 17. $\mathbb{N}^3 \not\sqsubseteq \mathbb{N}^{\oplus}$.

Proof. Assume φ is an order-embedding from $\mathbb{N}^3$ to $\mathbb{N}^{\oplus}$.

We consider the following sets:

- $A_x = \{(n, 0, 0) \mid n \in \mathbb{N}\}$,
- $A_y = \{(0, n, 0) \mid n \in \mathbb{N}\}$,
- $A_z = \{(0, 0, n) \mid n \in \mathbb{N}\}$.

For any $\alpha \in \{x, y, z\}$, $\varphi(A_\alpha)$ is an infinite chain of $\mathbb{N}^{\oplus}$ with least upper bound an element of $Idl(\mathbb{N}^{\oplus})$. If this element is the entire set, for any element x of $\mathbb{N}^3$, we can find an element x' of A_α such that $\varphi(x) \leqslant \varphi(x')$, contradicting the order embedding.

Thus, let $\{\omega^{k_\alpha}, {k'_\alpha}^{\omega}\} \cup B_\alpha$ be this element.

We remark that for any three pairs of integers, we can choose one of these pairs that is less or equal than the least upper bound of the two others.

This means, that we can find α, β and γ, such that:

$$(k_\alpha, k'_\alpha) \leqslant (\max\{k_\beta, k_\gamma\}, \max\{k'_\beta, k'_\gamma\})$$

Without loss of generality, we will assume $\alpha = x$, $\beta = y$ and $\gamma = z$. Then, we define $A_{y,z}[a] = \{(a, n, n) \mid n \in \mathbb{N}\}$.

In the same way as before, we have the image of $A_{y,z}[a]$ an infinite chain of $\mathbb{N}^{\oplus}$, with least upper bound $\{\omega^{k_{y,z}[a]}, (k'_{y,z}[a])^{\omega}\} \cup B_{y,z}[a]$. Because φ is an order embedding, for any $a \in \mathbb{N}$, this least upper bound is greater than both $\{\omega^{k_y}, {k'_y}^{\omega}\} \cup B_y$ and $\{\omega^{k_z}, {k'_z}^{\omega}\} \cup B_z$, implying that:

$$\forall a \in \mathbb{N}, \quad k_x \leqslant k_{y,z}[a] \quad \text{and} \quad k'_x \leqslant k'_{y,z}[a]$$

As we have $\varphi(n, 0, 0) \to \omega^{k_x}.{k'_x}^{\omega}.B_x$, we can find an a_0 such that $\varphi(a_0, 0, 0) = \{p_1, \ldots, p_{k_x}, q_1, \ldots, q_r\} \cup B_x$ with:

- $r \in \mathbb{N}$,
- $\forall 1 \leqslant i \leqslant k_x$, $p_i \geqslant \max(k'_x, M)$, where M is the greatest value in B_x,
- $\forall 1 \leqslant i \leqslant r$, $q_i \leqslant k'_x$.

We define $P = \{p_1, \ldots, p_{k_x}\}$ and $Q = \{q_1, \ldots, q_r\}$. We have:

$$P \cup Q \cup B_x \leqslant \{\omega^{k_{y,z}[a_0]}, k'_{y,z}[a_0]^{\omega}\} \cup B_{y,z}[a_0]$$

Elements of P are greater than all elements in Q and B_0, thus:

$$Q \cup B_x \leqslant \{\omega^{k_{y,z}[a_0]-k_x}, k'_{y,z}[a_0]^{\omega}\} \cup B_{y,z}[a_0]$$

Because $k'_x \leqslant k'_{y,z}[a_0]$, we have:

$$\{{k'_x}^{\omega}\} \cup B_x \leqslant \{\omega^{k_{y,z}[a_0]-k_x}, k'_{y,z}[a_0]^{\omega}\} \cup B_{y,z}[a_0]$$
$$\Longrightarrow \quad \{\omega^{k_x}, {k'_x}^{\omega}\} \cup B_x \leqslant \{\omega^{k_{y,z}[a_0]}, k'_{y,z}[a_0]^{\omega}\} \cup B_{y,z}[a_0]$$

and because that means that each image of an element of A_x can be compared to an element of $A_{y,z}[a_0]$, we get a contradiction that concludes the demonstration. □

References

[1] A. Finkel, A generalization of the procedure of Karp and Miller to well structured transition systems, in: T. Ottmann (Ed.), ICALP, in: Lecture Notes in Computer Science, vol. 267, Springer, 1987, pp. 499–508.
[2] A. Finkel, P. McKenzie, C. Picaronny, A well-structured framework for analysing Petri net extensions, Inform. and Comput. 195 (2004) 1–29.
[3] G. Geeraerts, J.-F. Raskin, L.V. Begin, Well-structured languages, Acta Inform. 44 (2007) 249–288.
[4] R. Lazic, T. Newcomb, J. Ouaknine, A.W. Roscoe, J. Worrell, Nets with tokens which carry data, Fund. Inform. 88 (2008) 251–274.
[5] R. Alur, C. Courcoubetis, T.A. Henzinger, The observational power of clocks, in: B. Jonsson, J. Parrow (Eds.), CONCUR, in: Lecture Notes in Computer Science, vol. 836, Springer, 1994, pp. 162–177.
[6] D. Figueira, S. Figueira, S. Schmitz, P. Schnoebelen, Ackermannian and primitive-recursive bounds with Dickson's lemma, in: LICS, IEEE Computer Society, 2011, pp. 269–278.
[7] D.H.J. de Jongh, R. Parikh, Well partial orderings and hierarchies, Indag. Math. 80 (1977) 195–207.
[8] D. Schmidt, Well-partial orderings and their maximal order types, Habilitationsscrift, 1979.
[9] A. Weiermann, A computation of the maximal order type of the term ordering on finite multisets, in: K. Ambos-Spies, B. Löwe, W. Merkle (Eds.), CiE, in: Lecture Notes in Computer Science, vol. 5635, Springer, 2009, pp. 488–498.
[10] F. Rosa-Velardo, D. de Frutos-Escrig, Decidability and complexity of Petri nets with unordered data, Theoret. Comput. Sci. 412 (2011) 4439–4451.
[11] F. Rosa-Velardo, G. Delzanno, Language-based comparison of Petri nets with black tokens, pure names and ordered data, in: A.H. Dediu, H. Fernau, C. Martín-Vide (Eds.), LATA, in: Lecture Notes in Computer Science, vol. 6031, Springer, 2010, pp. 524–535.
[12] P.A. Abdulla, A. Nylén, Timed Petri nets and BQOs, in: J.M. Colom, M. Koutny (Eds.), ICATPN, in: Lecture Notes in Computer Science, vol. 2075, Springer, 2001, pp. 53–70.
[13] P.A. Abdulla, G. Delzanno, L.V. Begin, Comparing the expressive power of well-structured transition systems, in: J. Duparc, T.A. Henzinger (Eds.), CSL, in: Lecture Notes in Computer Science, vol. 4646, Springer, 2007, pp. 99–114.
[14] P.A. Abdulla, G. Delzanno, L.V. Begin, A language-based comparison of extensions of Petri nets with and without whole-place operations, in: A.H. Dediu, A.-M. Ionescu, C. Martín-Vide (Eds.), LATA, in: Lecture Notes in Computer Science, vol. 5457, Springer, 2009, pp. 71–82.
[15] P.A. Abdulla, G. Delzanno, L.V. Begin, A classification of the expressive power of well-structured transition systems, Inform. and Comput. 209 (2011) 248–279.
[16] S. Schmitz, P. Schnoebelen, Multiply-recursive upper bounds with Higman's lemma, in: L. Aceto, M. Henzinger, J. Sgall (Eds.), ICALP (2), in: Lecture Notes in Computer Science, vol. 6756, Springer, 2011, pp. 441–452.
[17] R. Bonnet, A. Finkel, S. Haddad, F. Rosa-Velardo, Ordinal theory for expressiveness of well structured transition systems, in: M. Hofmann (Ed.), FOSSACS, in: Lecture Notes in Computer Science, vol. 6604, Springer, 2011, pp. 153–167.
[18] R. Bonnet, A. Finkel, S. Haddad, F. Rosa-Velardo, Comparing Petri data nets and timed Petri nets, Research Report LSV-10-23, Laboratoire Spécification et Vérification, ENS Cachan, France, 2010, 16 pp.
[19] D. Brand, P. Zafiropulo, On communicating finite-state machines, J. ACM 30 (1983) 323–342.
[20] G. Memmi, A. Finkel, An introduction to fifo nets-monogeneous nets: A subclass of fifo nets, Theoret. Comput. Sci. 35 (1985) 191–214.
[21] B. Vauquelin, P. Franchi-Zannettacci, Automates a file, Theoret. Comput. Sci. 11 (1980) 221–225.
[22] P.A. Abdulla, B. Jonsson, Verifying programs with unreliable channels, Inform. and Comput. 127 (1996) 91–101.
[23] A. Finkel, Detection and avoidance of deadlocks for protocols, Research Report 463, University Paris 11-Orsay, France, 1989, 18 pp.
[24] A. Finkel, Decidability of the termination problem for completely specified protocols, Distributed Computing 7 (1994) 129–135.
[25] F. Rosa-Velardo, M. Martos-Salgado, D. de Frutos-Escrig, Accelerations for the coverability set of Petri nets with names, Fund. Inform. 113 (2011) 313–341.
[26] P.A. Abdulla, P. Mahata, R. Mayr, Dense-timed Petri nets: Checking zenoness, token liveness and boundedness, Log. Methods Comput. Sci. 3 (2007).
[27] P. Bouyer, S. Haddad, P.-A. Reynier, Timed Petri nets and timed automata: On the discriminating power of zeno sequences, Inform. and Comput. 206 (2008) 73–107.
[28] O. Kouchnarenko, P. Schnoebelen, A formal framework for the analysis of recursive-parallel programs, in: V.E. Malyshkin (Ed.), PaCT, in: Lecture Notes in Computer Science, vol. 1277, Springer, 1997, pp. 45–59.
[29] S. Joshi, B. König, Applying the graph minor theorem to the verification of graph transformation systems, in: A. Gupta, S. Malik (Eds.), CAV, in: Lecture Notes in Computer Science, vol. 5123, Springer, 2008, pp. 214–226.
[30] R. Meyer, On boundedness in depth in the pi-calculus, in: G. Ausiello, J. Karhumäki, G. Mauri, C.-H.L. Ong (Eds.), IFIP TCS, in: IFIP, vol. 273, Springer, 2008, pp. 477–489.