Theory of Well Structured Transition Systems and Extended Vector Addition Systems

Rémi Bonnet

October 25, 2012

Contents

Introduction

From a luxury available only to a select few, computers and embedded systems have turned into an omnipresent technology. This evolution has made us highly dependant on the correct behaviour of these systems: without even considering the possibility of a failure in a critical system like a nuclear power plant or an airplane, any hiccup in a major search engine or in the stock market protocols would have disastrous consequences. Therefore, it has become more and more important to check that systems will not fail, even under unusual circonstances.

An important step in any verification of a real-world system is to define a model of it, on which one will be able to run algorithms that check if the required properties are fulfilled. If this step is generally feasible when the system can only be in a finite number of states, those that can have an infinite number of states give rise to important challenges. Indeed, one will often face the barrier of undecidability: there is some models (the canonical one being Turing machines) on which there is no algorithm that can check if a non-trivial property is fulfilled. Moreover, even if some algorithms do exist, their running time might make them unusable in practice.

Well Structured Transition Systems

Well Structured Transition Systems (WSTS) are a general class of infinite-state systems that enjoy monotonicity properties [28]. These properties allow many problems to be decidable [33, 6] like control state reachability (is a specific line of the program reachable?), termination (is there an infinite execution of the program?), and with additionnal assumptions, boundedness (is the set of reachable states finite?).

The interest of this class is threefold:

- Some important classes of infinite-state systems are directly WSTS. Petri Nets (that we will describe later) are for example a widely used model of concurrency, that is a WSTS. Many results on Petri Nets are thus obtained directly from the framework of WSTS.

- Some systems may not have the required monotonicity, but can be over-approximated in order to be a WSTS. If this means new behaviors of the system are introduced, any negative answer for control state reachability and positive answer for termination or boundedness will still be valid for the original system. This is an approach used for example in [1], [3] or [2].

- Finite-state automata are a particular kind of WSTS, and algorithms designed for WSTS can be applied to finite automatas. This can yield surprisingly effective new algorithms. See for example [22].

There are two main classes of algorithms on WSTS, both relying heavily on the monotony:

- *backward algorithms*, that start from a final state, then compute progressively an over-approximation of the predecessor states. These algorithms generally allow to decide safety properties like control state reachability. The properties of WSTS allow for a simple representation of the approximation of predecessor states.

- *forward algorithms*, that start from an initial state, then compute progressively an over-approximation of the reachable states. These algorithms generally allow to decide liveness properties like termination or properties related to the full set of reachable states like boudedness [6, 33]. However, the representation of the approximation of reachable states is not immediate and that makes deciding properties more precise than boundedness difficult. Recently, a procedure to decide such properties has been proposed by Finkel and Goubault-Larrecq [29, 30], but its termination was not guaranteed.

Petri Nets

A simple class of infinite-state systems are counter systems, in which states are given by a set of counter values (integers, generally required to be non-negative) while the transitions of the system perform various operations on these counters. Of course, the set of allowed operations is of great importance for the modeling power and decidability of the class. For example, as soon as two counters can be tested for emptyness, incremented and decremented, one gets the class of Minsky machines [51], that are Turing-powerful systems, which make all non-trivial properties undecidable.

If only increments and decrements are allowed, one gets the class of Vector Addition Systems. This class is equivalent to Petri Nets [52], which are a widely used model for concurrent programs, where each counter correponds to a place containing undistingishable tokens such that the value of the counter correponds to the number of tokens inside the place. Reachability for Petri Nets is decidable [49, 42, 47] as is termination and boundedness by the WSTS framework [33, 6]. Among the various algorithms designed to check properties on Petri Nets, let us mention the Karp-Miller tree [41], which explores the reachable states in a forward manner, performing acceleration when possible in order to compute an over-approximation of the reachability set, and is an instance (and a precursor) of the forward algorithm for WSTS.

Many alterations of Petri Nets have been defined, aiming either to restrict their power in order to obtain more efficient algorithms, or to extend it in order to find up to what point the decidability can be preserved:

- A first category of extensions is one where new kind of transitions are allowed. For example, on top of increments and decrements (token creation and suppression in Petri Net terms), one can allow transitions to apply affine functions on the counters (transfering/copying/emptying whole places in Petri Net terms). Depending on the coefficients allowed in the matrix associated to the function, this gives the class of Self-Modifying Nets [61] (relative integers), Reset Nets [23] (diagonal matrix with only 0 and 1), Affine Nets [31] (non-negative integers) or Post-Self-Modifying Nets / strongly increasing Affine Nets [61, 31] (non-negative integers + positive integers on the diagonal). Another possibility is to allow the transitions to check whether a counter value is zero (whether a place is empty in Petri Net terms) [55] as long as some restriction on these transition is used to avoid being Turing-powerful.

- A second category of extensions is obtained by attaching data to individual tokens, for example time values ([12]), but also abstract values ([56, 45]). For such extensions, a place can no longer be considered as a counter, but given a suitable representation of the configurations, one can hope to be able to extend the algorithms used for Petri Nets without extensions.

In this work, we are mostly interested in finding the frontier of decidability: we know that Petri Nets enjoy very good decidability properties, but how much can we extend these in order to maintain these decidability properties? And on a side note, are these extensions meaningful? (i.e. do they allow to model systems that could not be modelled without them).

Outline

- In chapter 1, we will recall some facts about Well Structured Transition Systems and Petri Nets. After introducing the usual verification problems for these systems, we will provide a summary of the decidability results currently available.

- In chapter 2, we will consider the problem of computing a finite representation of the cover (an over-approximation of the reachability set), that would allow to solve the open problems presented at end of chapter 2. We will present an extension of the works of Finkel and Goubault-Larrecq on complete WSTS ([29, 30]) by introducing *acceleration strategies*, that allow to reach the maximal elements of this cover. The most important result of this chapter is quite surprising: for complete WSTS, if cover is recursively enumerable, then it is recursive and admits a computable finite representation. We illustrate this method on strictly monotonic WSTS.

- Chapter 3 will focus on the question of expressiveness of WSTS, specifically of Petri Net extensions. As many extensions of Petri Nets have been designed, it is a natural question to ask whether these extensions allow to express more behaviours than the basic model. We provide (with some conditions) a very simple result: if an extension of a Petri Net changes the underlying state space, it is highly likely that this increases expressiveness.

- After these general results, we will turn on a specific VAS extension, which are VAS with two resets. In chapter 4, we will show that one can use the witnesses of unboudedness described by Dufourd *et al.* in [24] in order to enumerate the elements of the cover, which by our previous results, entails the computability of a finite representation of this cover.

- We end this work by another VAS extension, which are VAS with hierarchical zero-tests, described in chapter 5. We show in section 5.2 that the proof of reachability for VAS of Leroux [47] can be adapted to this model, yielding a new proof of reachability as an alternative to the original proof of Reinhardt [54, 55]. Our new reachability proof relies heavily on a well order on runs of VAS with hierarchical zero-tests, that we recycle in section 5.3 in order to enumerate the maximal elements of the cover, again obtaining computability of the cover. Then, in section 5.4, we use the two previous results to derive decidability of LTL model-checking and of regularity.

Chapter 1

Preliminaries

We denote by $\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{Q}$ the usual sets of natural integers, relative integers and rationals. We write $\leq$ the canonical ordering on these sets. We also define $\mathbb{Q}_{\geq 0} = \{x \in \mathbb{Q} \mid x \geq 0\}$, $\mathbb{Q}_{>0} = \{x \in \mathbb{Q} \mid x > 0\}$, $\mathbb{N}_{>0} = \{x \in \mathbb{N} \mid x > 0\}$ and $\mathbb{N}_\omega$ by $\mathbb{N} \cup \{\omega\}$. The addition and substraction on $\mathbb{N}_\omega$ is extended by $\omega \pm x = \omega$ for all $x \in \mathbb{N}$ and $\omega + \omega = \omega$ (ω will never be substracted).

Given a set X, we write $\mathcal{P}(X)$ (resp. $\mathcal{P}_{fin}(X)$) its set of subsets (resp. finite subsets). We write $X \subseteq Y$ (resp. $X \subseteq_{fin} Y$) if X is a subset (resp. finite subset) of Y. The cardinal of a finite set is written $card(X)$. Finally, a singleton set $\{x\}$ is written x when it is clear from the context that we are speaking of a set. Addition and multiplication are extended on sets by, given X and Y two subsets of $\mathbb{Q}$, $X+Y = \{x+y \mid x \in X \wedge y \in Y\}$ and $X*Y = \{x*y \mid x \in X\}$. With the singleton notation, we also get, given $k \in \mathbb{Q}$, $k * X = \{k * x \mid x \in X\}$. Given $k \in \mathbb{N}$, we define $k \star X$ by: $0 \star X = \{0\}$ and $(k+1) \star X = (k \star X) + X$. Finally, we have $\mathbb{N} \star X = \bigcup_{k \in \mathbb{N}} k \star X$. Note that $k * X$ and $k \star X$ will be in general different sets.

Given two sets X and Y, we write $f : X \to Y$ when f is a *partial function* from X to Y. The *domain* of f is written $dom(f)$ and a function $f : X \to Y$ is *total* if $dom(f) = X$. We also call total functions *mappings*. We will mostly consider partial functions, so we will call simply function any partial function, and we will precise when we want to speak specifically of total functions. An *injection* is a function $f : X \to Y$ if $f(x) = f(x')$ implies $x = x'$. If $X' \subseteq X$, we write $f(X') = \{f(x) \mid x \in X'\}$. Similarly, for $Y' \subseteq Y$, $f^{-1}(Y') = \{x \in X \mid f(x) \in Y'\}$. The set of partial (resp. total) functions from X to Y is written $X \to Y$ (resp. Y^X). Given $f : X \to Z$ and $f' : Z \to Y$, the composition of f and f' is written $f' \circ f$ or ff' (be careful to the change of order) and defined by $y = ff'(x) = (f' \circ f)(x)$ if there exists $z \in Z$ such that $z = f(x)$ and $y = f'(z)$.

A *relation* on X is a subset $R \subseteq X \times X$. Like for functions, given $X' \subseteq X$, we write $R(X) = \{y \mid \exists x \in X'. (x,y) \in R\}$. The composition of relations is also written $R' \circ R$ or RR' and defined by $(x,y) \in R' \circ R = RR'$ if there exists z such that $(x,z) \in R$ and $(z,y) \in R'$. A relation is identified to a function if for every $x \in X$, $R(x)$ is either empty or a singleton. In this case, all notions defined here for relations and functions coincides by treating $R(x) = \{y\}$ as y. Given a relation R, we define its inverse R^{-1} by $(y,x) \in R^{-1} \iff (x,y) \in R$ and its reflexive transitive closure R^* by $R^* = \bigcup_{k \in \mathbb{N}} R^k$.

We consider *vectors* of length d as a special kind of total functions with domain $\{0, \ldots, d-1\}$ and we shorten $X^{\{0,\ldots,d-1\}}$ as X^d. We also write any vector of X^d as $x = (x(0), x(1), \ldots, x(d-1))$ with $x(i) \in X$. Finally, we define the subvectors $x(k \ldots \ell) = (x(k), \ldots, x(\ell))$. We also call a vector of any length a *finite sequence* and we call total functions from $\mathbb{N}$ to X *infinite sequences*. The set of finite sequences is written $X^{<\omega}$ while $X^{\mathbb{N}}$, the set of infinite sequences, is also written X^ω. The set of all sequences (finite and infinite) is thus written $X^{\leq\omega}$. When we are speaking of sequences, we will use the notation $u = (u_i)_{0 \leq i < \ell}$ where $\ell \in \mathbb{N}_\omega$ is the length of the sequence, or simply $u = (u_i)$ when we don't care about the length. Given $(u_i)_{0 \leq i < \ell}$ a sequence, a subsequence is a sequence $(v_i)_{1 \leq i < \ell'}$ such that there exists a strictly increasing mapping φ from $\{i \in \mathbb{N} \mid i < \ell'\}$ to $\{i \in \mathbb{N} \mid i < \ell\}$ with $v_i = u_{\varphi(i)}$. We also use the vector notation $u(i)$ to denote the i-th element of the sequence and $u(k \ldots)$ to denote the subsequence $(u(k+n))_{n \in \mathbb{N}}$.

We also call finite sequences *words*, and when speaking about words, we use the alternate notation $u = x_0 \ldots x_{\ell-1}$ and write the set of all words X^*. The length of a word u is written $|u|$. The concatenation of two words u and v is simply written uv and the empty word is denoted ε with $\varepsilon a = a\varepsilon = a$. In this formalism, if v would be a subsequence of u, we say that v is a subword of u. Given a set of words X^*, a *language* on X is a subset of X^*. The concatenation of two languages is given by $LL' = \{uv \mid u \in L \land v \in L'\}$. We will use especially often the singleton shortcut defined earlier when defining concatenated languages, for example $uL = \{uv \mid v \in L\}$.

Given a set X, we denote by $X^\oplus$ the set of finite multisets of X, that is the set of total functions $m : X \to \mathbb{N}$ with a finite support $sup(m) = \{x \in X \mid m(x) > 0\}$. We use the set-like notation $\{\!|\ \ldots\ |\!\}$ for multisets when convenient, with $\{\!|\ x^n\ |\!\}$ describing the multiset containing x n times. We use $\cup$ and $-$ for multiset operations where:

$$\begin{array}{rcl}(m \cup m')(x) & = & m(x) + m'(x) \\ (m - m')(x) & = & max(m(x) - m'(x), 0)\end{array}$$

1.1 Orderings

A *partial ordering* $\preceq$ on X is a reflexive, transitive and anti-symmetric relation on X. It is a *total ordering* if for any $x, y \in X$, we either have $x \preceq y$ or $y \preceq x$. Total orderings are also called *linear orderings*. If $\preceq$ is a partial (resp. total) ordering on X, we write that that $(X, \preceq)$ is a partially (resp. totally) ordered set. We write $x \prec y$ if $x \preceq y$ and $y \not\preceq x$. An *antichain* of $(X, \preceq)$ is a subset $Y \subseteq X$ such that elements of Y are pairwise incomparable (i.e. for all $y, y' \in Y$, $y \neq y' \implies y \not\preceq y'$). As we will mostly use partial orderings in this work, we will call partial orderings simply orderings and specify when we require the order to be total. If there is no ambiguity possible on the order used (see section 1.1.2 for the default orderings on the sets will we use), we simply write that X is a partially (resp. totally) ordered set. Also, when considering a generic ordered set, we shorten $(X, \leq)$ as X.

Given $(X, \preceq)$ an ordered set, the *upward closure* of a set $E \subseteq X$ is $\uparrow E = \{y \in X \mid \exists x \in E,\ x \preceq y\}$ and conversely the downward closure of E is $\downarrow E = \{y \in X \mid \exists x \in E,\ y \preceq x\}$. E is *upward-closed* (resp. *downward-closed*) if $E = \uparrow E$ (resp. $E = \downarrow E$). A downward-closed (resp. upward-closed) set E has a basis B if $E = \downarrow B$ (resp. $E = \uparrow B$). E has a *finite basis* if B can be chosen finite. An *upper bound* $x \in X$ of $E \subseteq X$ is such that $y \preceq x$ for every $y \in E$. The *least upper bound* of a set E, if it exists, is written $lub(E)$. We write $Max\ E$ (resp. $Min\ E$) the set of maximal elements (resp. minimal elements) of E. We define the notion of increasing for sequences, functions and relations by:

- A sequence $(x_n)_{n \in \mathbb{N}}$ is *increasing* (respectively *strictly increasing*, *decreasing*, *strictly decreasing*) if for every $n \in \mathbb{N}$, we have $x_n \preceq x_{n+1}$ (respectively $x_n \prec x_{n+1}$, $x_n \succeq x_{n+1}$, $x_n \succ x_{n+1}$).
- If $(Y, \preceq_Y)$ is an ordered set, a function $f : X \to Y$ is *increasing* (resp. *strictly increasing*) if $dom(f)$ is upward closed and $x \leq x'$ implies $f(x) \preceq_Y f(x')$ (resp. $x \prec x'$ implies $f(x) \prec_Y f(x')$) for all $x, x' \in dom(f)$.

- A relation $R \subseteq X \times X$ is *increasing* (resp. *strictly increasing*) if for all $(x, y) \in R$, and $x \preceq x'$ (resp. $x \prec x'$), there exists $(x', y') \in R$ such that $y \preceq y'$ (resp. $y \prec y'$).

Finally, a total function $f : X \to Y$ is an *order embedding* if $x \preceq x' \iff f(x) \preceq_Y f(x')$.

1.1.1 Well orderings

An ordering $\preceq$ on X is *well founded* if there is no infinite strictly decreasing sequence. It is *well* if there is also no infinite antichain. There are equivalent formulations for this definition, as given by the following proposition:

Proposition 1.1. *[44] Given an ordered set $(X, \preceq)$, the following propositions are equivalent:*

- *$\preceq$ is well founded and there is no infinite antichain in $(X, \preceq)$.*
- *For any infinite sequence (x_i) in X^ω, one can find $i \prec j$ such that $x_i \preceq x_j$.*
- *For any infinite sequence (x_i) in X^ω, one can extract an infinite increasing subsequence.*
- *Any upward closed subset of X admits a finite basis.*

The fourth characterization of well-ordered sets will be of particular interest in the sequel. Indeed, it means that any upward closed set can be finitely represented by its set of minimal elements.

1.1.2 Default Orderings

Unless otherwise stated, we equip these sets with the following orderings:

- $\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{Q}$ are ordered by the canonical ordering.
- The order on $\mathbb{N}_\omega$ is the extension of the order on $\mathbb{N}$ by considering ω as strictly larger than any integer.
- The order on $X \times Y$, given orders on X and Y is the product ordering given by:

$$(x, y) \leq (x', y') \iff x \leq x' \wedge y \leq y'$$

- The order on X^d, given an order on X, is the pointwise ordering given by:

$$x \leq y \iff \forall i \in \{0, \dots, d-1\}.\ x(i) \leq y(i)$$

- If X is ordered, the order on $X^\oplus$, is the multiset embedding ordering given by:

$$\{\mid x_0, \dots, x_{p-1} \mid\} \leq^{emb} \{\mid y_0, \dots, y_{q-1} \mid\}$$
$$\iff$$
$$\exists \text{ a total injection } \varphi : \{0, \dots, p-1\} \to \{0, \dots, q-1\}.$$
$$\forall i \in \{0, \dots, p-1\},\ x_i \leq y_{\varphi(i)}$$

  If $m \leq^{emb} m'$, we say that m is embedded in m'.

- If X is ordered, the order on X^*, is the word embedding ordering given by:

$$u \leq^{emb} v$$
$$\iff$$
$$\exists \text{ a strictly increasing mapping } \varphi : \{0, \ldots, |u|-1\} \to \{0, \ldots, |v|-1\}.$$
$$\forall i \in \{0, \ldots, p-1\}.\ u_i \leq v_{\varphi(i)}$$

  If $u \leq^{emb} v$, we say u is embedded in v. Note that this is different from u being a subword of v, unless X is ordered by equality.

Most of these orders are well-orders, as given by the following propositions:

Proposition 1.2. $\mathbb{N}$ *is well-ordered.*

Proposition 1.3. *(Dickson's lemma)*
If X and Y are well-ordered, then $X \times Y$, ordered by the product ordering and X^d, ordered by the pointwise ordering are well-ordered.

Proposition 1.4. *(Higman's lemma)*
If X is well-ordered by $\leq$, then $X^{\oplus}$ and X^* ordered by $\leq^{emb}$ are well-ordered.

Finally, given an ordered set X, we will use in some rare cases the lexicographic ordering $\leq_{lex}$ on X^d that is defined by:

$$(x_0, \ldots, x_{d-1}) \leq_{lex} (y_0, \ldots, y_{d-1}) \iff \exists k \in \{0, \ldots, d-1\}. \begin{cases} x_k < y_k \\ \forall 0 \leq i \leq k-1.\ x_i \leq y_i \end{cases}$$

Proposition 1.5. *If X is well-ordered by $\leq$, then X^d is well-ordered by $\leq_{lex}$.*

1.2 Transition Systems

We will consider many kinds of transition sytems in this work. The simplest kind we will study is the following:

Definition 1.1. *A* Transition System *(shortly: TS) is a tuple $\mathcal{S} = \langle X, \longrightarrow \rangle$ where:*

- *X is a set of states.*
- *$\longrightarrow \subseteq X \times X$ is the transition relation.*

The reflexive transitive closure of $\longrightarrow$ is written $\longrightarrow\!\!\!\!\rightarrow$. The sets of *immediate successors*, *immediate predecessors* and the *reachability set* are defined by:

$$\begin{array}{rcl} Pre_{\mathcal{S}}(y) & = & \longrightarrow(X) = \{x \in X \mid x \longrightarrow y\} \\ Post_{\mathcal{S}}(x) & = & \overset{-1}{\longrightarrow}(X) = \{y \in X \mid x \longrightarrow y\} \\ Reach_{\mathcal{S}}(x) & = & \longrightarrow\!\!\!\!\rightarrow(X) = \{y \in X \mid x \longrightarrow\!\!\!\!\rightarrow y\} \end{array}$$

A transition system is *finite-branching* if for every $x \in X$, $Post_{\mathcal{S}}(x)$ is finite. It is *infinite-branching* otherwise. Most transition systems in this work will be finite-branching.

Moreover, if X is ordered, we define the *cover* by:

$$Cover_{\mathcal{S}}(x) = \downarrow Reach_{\mathcal{S}}(x)$$

Sometimes, studying the reachability set (or one of its variation like the cover) won't be enough to answer our questions. For example, one might be interested in whether some specific transition is used, the usual example being "whenever a 'request' transition is used, an 'answer' transition must be used sometime after". This will be done by labelling the transition with a finite set of actions. We allow there a transition to be labelled by more than one action (or by none). This might make some problems harder to answer (for example the regularity of the trace language, see section 5.4.2), so we will sometimes require that each transition is labelled by a single action.

Moreover, when we work on purely state problems (and hence labels are meaningless), we will allow ourselves to use an infinite set of actions in order to provide to the reader informations about what transitions are used (this will be the case mainly in section 5.3). However, the sections that deal with results relying on labels (chapter 3 and section 5.4.2) will only use finite set of actions.

Definition 1.2. *A* Labelled Transition System *(shortly: LTS) is a tuple* $\mathcal{S} = \langle X, A, \longrightarrow \rangle$ *where:*

- X *is a set of states.*
- A *is a set of labels.*
- $\longrightarrow \subseteq X \times A^* \times X$ *is the transition relation.*

We write $x \xrightarrow{u} y$ if $(x, u, y) \in \longrightarrow$. We define $\xrightarrow{\cdot}\!\!\!\!\rightarrow$ as a kind of transitive reflexive closure of $\xrightarrow{\cdot}$, i.e. by the smallest relation satisfying:

- $x \xrightarrow{\varepsilon}\!\!\!\!\rightarrow x$ for any $x \in X$.
- $x \xrightarrow{u}\!\!\!\!\rightarrow y$ if $x \xrightarrow{u} y$.
- $x \xrightarrow{uv}\!\!\!\!\rightarrow z$ if there exists $y \in X$ such that $x \xrightarrow{u}\!\!\!\!\rightarrow y \xrightarrow{v}\!\!\!\!\rightarrow z$.

Note that in general $x \xrightarrow{a}\!\!\!\!\rightarrow y$ doesn't imply $x \xrightarrow{a} y$ as $x \xrightarrow{a}\!\!\!\!\rightarrow y$ might have been obtained by $x \xrightarrow{a} x' \xrightarrow{\varepsilon} y$.

For $\mathcal{S} = \langle X, A, \longrightarrow \rangle$ a LTS and given L a language on A, we define $\xrightarrow{L} = \bigcup_{u \in L} \xrightarrow{u}$ and $\xrightarrow{L}\!\!\!\!\rightarrow = \bigcup_{u \in L} \xrightarrow{u}\!\!\!\!\rightarrow$. We write $\longrightarrow = \xrightarrow{A^*}$ and $\longrightarrow\!\!\!\!\rightarrow = \xrightarrow{A^*}\!\!\!\!\rightarrow$.

A (finite or infinite) run ρ of a LTS $\mathcal{S}$ is a (finite or infinite) sequence $x_0.a_1.x_1.a_2 \ldots a_n.x_n \ldots$ alternating states and actions such that $\forall i.\ x_{i-1} \xrightarrow{a_i} x_i$. We write $\rho = x_0 \xrightarrow{a_1} x_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n}$

$x_n \cdots$. Given such a run ρ, we define $src(\rho) = x_0$, $acts(\rho) = a_1 \ldots a_n \ldots$ and if ρ is finite, $tgt(\rho) = x_n$ where x_n is the last element of the sequence.

A labelled transition system equipped with an initial state and a final state recognizes a *reachability language*, that is the set of words that allow to go from the initial state to the final state. If X is ordered, we also define the *coverability language*, that is the set of words that allow to go from the initial state to a state greater than the final state:

$$\begin{aligned} L_r(\mathcal{S}, x, y) &= \{u \in A^* \mid x \xrightarrow{u}\!\!\!\rightarrow y\} \\ L_c(\mathcal{S}, x, y) &= \{u \in A^* \mid \exists y' \geq y.\ x \xrightarrow{u}\!\!\!\rightarrow y'\} \end{aligned}$$

We also define the *finite trace language* and the *infinite trace language* that contain respectively all finite transition sequences and all infinite transition sequences that can be obtained from an initial state:

$$\begin{aligned} L_t(\mathcal{S}, x) &= \{u \in A^* \mid \exists y.\ x \xrightarrow{u}\!\!\!\rightarrow y\} \\ L_t^\omega(\mathcal{S}, x_0) &= \{u \in A^\omega \mid \exists (x_k)_{k>1}.\ \forall k \in \mathbb{N}.\ x_k \xrightarrow{u(k)}\!\!\!\rightarrow x_{k+1}\} \end{aligned}$$

A LTS $\mathcal{S} = \langle X, A, \longrightarrow \rangle$ can be relabelled by a function $\gamma : A \to B^*$. This gives raise to the LTS $\mathcal{S}_\gamma = \langle X, B, \longrightarrow_\gamma \rangle$ where $\longrightarrow_\gamma = \{(x, \gamma(u), y) \mid (x, u, y) \in \longrightarrow\}$ where γ is extended by morphism on A^*.

1.2.1 Functional Transition Systems

It will be sometimes be more practical to split the transition relation into a (finite) number of functions. We define an alternate formalism for this case (a similar idea can be found in [30]).

Definition 1.3. *A* functional Transition System *(shortly: f-TS) is a tuple* $\mathcal{S} = \langle X, F \rangle$ *where:*

- X *is a set of states*
- F *is a set of partial functions from* X *to* X*.*

A functional Transition System $\mathcal{S} = \langle X, F \rangle$ induces a Transition System $\mathcal{S}' = \langle X, \longrightarrow \rangle$ by:

$$(x, y) \in \longrightarrow \iff \exists f \in F,\ f(x) = y$$

Moreover, one can label a functional Transition System by a set of labels A and a function $\gamma : F \to A^*$. This gives birth to a f-LTS $\langle X, F, A, \gamma \rangle$ and a LTS $\langle X, A, \longrightarrow \rangle$ where $\longrightarrow$ is defined by:

$$x \xrightarrow{u} y \iff \exists f \in F. \begin{cases} f(x) = y \\ \gamma(f) = u \end{cases}$$

We assimilate a functional (Labelled) Transition System to its associated TS (LTS) and we port all notations defined for TS (LTS) to f-TS (f-LTS).

1.2.2 Well Structured Transition Systems

The most important class of transition systems that we will consider are Well Structured Transition Systems (shortly: WSTS), on which many properties are known to be decidable. There is different monotony properties that the transition relation can fulfill. We define here the two most important ones, as in [33].

Definition 1.4. *[28], [6], [33]*

A TS $\langle X, \longrightarrow \rangle$ is a Well Structured Transition System *with* strong monotonicity *(resp.* weak monotonicity*) if:*

- *X is a well-ordered set.*
- *$\longrightarrow$ is an increasing relation (resp. $\twoheadrightarrow$ is an increasing relation)*

A LTS $\langle X, A, \longrightarrow \rangle$ is a Well Structured Transition System *with* strong monotonicity *(resp.* weak monotonicity*) if:*

- *X is a well-ordered set.*
- *For any $u \in A^*$, $\xrightarrow{u}$ is an increasing relation (resp. $\overset{u}{\twoheadrightarrow}$ is an increasing relation).*

One can check that any WSTS with strong monotony is also a WSTS with weak monotony. Unless otherwise stated, our WSTS are using the strong monotonicity condition.

A sufficient condition for a f-LTS or f-TS to be a WSTS is that all its functions are increasing (this corresponds to the definition of WSTS in [30]) even if this not a necessary condition.

Example 1.1. *Let $\mathcal{S} = \langle X, A, F, \gamma \rangle$ be a f-LTS defined by:*

- *$X = \{x_1, x_3, x'_1, x'_2, x'_3\}$ with $x'_1 \geq x_1$ and $x'_3 \geq x_3$*
- *$A = \{a, b\}$*
- *$F = \{f, g, h\}$ with:*

$$\begin{array}{lll} dom(f) = \{x_1\} & f(x_1) = x_3 & \gamma(f) = ab \\ dom(g) = \{x'_1\} & f(x'_1) = x'_2 & \gamma(g) = a \\ dom(h) = \{x'_2\} & f(x'_2) = x'_3 & \gamma(h) = b \end{array}$$

$$\begin{array}{ccccc} x'_1 & \xrightarrow{\quad a \quad} & x'_2 & \xrightarrow{\quad b \quad} & x'_3 \\ \leq & & & & \leq \\ x_1 & & \xrightarrow{\qquad\quad ab \qquad\quad} & & x_3 \end{array}$$

f is not increasing, because its domain is not upward closed. However, $\mathcal{S}$ is a WSTS with weak monotony. It doesn't have strong monotony, because $x'_1 \overset{ab}{\not\rightarrow} x'_3$.

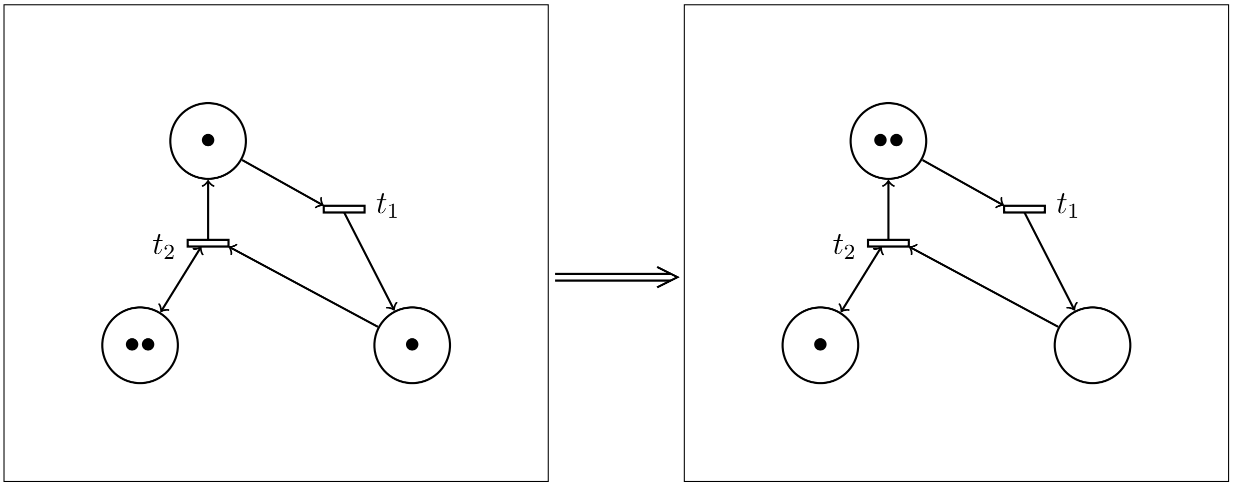


Figure 1.1: Firing a Petri Net transition (here t_2)

1.3 Petri Nets

One of the most studied kind of Well Structured Transition Systems is the class of Petri Nets [52]. Intuitively, a Petri Net is made of some finite number of *places* that can contain indistinguishable tokens and some finite number of *transitions* that are linked to places by oriented arcs. A transitions can be *fired* (see figure 1.1) by consuming one token from each place where there is an arc from the place to the transition, and producing one token in each place where there is an arc from the transition to the place.

A usual formal definition is:

Definition 1.5. *A Petri Net is a tuple* $\langle P, T, F, H \rangle$ *where:*

- P *is a finite set of* places
- T *is a finite set of* transitions
- $F \subseteq (P \times T) \to \mathbb{N}$ *is the multiset of arcs from places to transitions.*
- $H \subseteq (T \times P) \to \mathbb{N}$ *is the multiset of arcs from transitions to places.*

A Petri Net induces a LTS $lts(\mathcal{N}) = \langle \mathbb{N}^P, T, \longrightarrow \rangle$ where $\longrightarrow$ is defined by:

$$x \xrightarrow{t} y \text{ iff } \forall p \in P. \begin{cases} x(p) \geq F(p,t) \\ y(p) = x(p) + H(t,p) - F(p,t) \end{cases}$$

There is two main ways of extending Petri Nets:

- By adding new types of arcs, that will perform other operations than adding or removing a token. Some possible arcs are *reset* (that remove all tokens from one place), *transfer* (that transfer all tokens from one place to another) and *inhibitory* (that prevent the firing of the transition if a place is non-empty). Petri Nets with resets and transfers are still WSTS, but inhibitory arcs break the monotony.

- By adding data to tokens, which may be abstract data [45] or some time information [12]. Such an extension will change the state space of the associated transition system to include the additionnal data. We are considering here only extensions that are still WSTS. A definition of some of these extensions is available in section 3.4.

We will study some of these extensions in more detail in the remainder of this work. Of particular interest will be the reset arcs, as their presence or absence will make a major difference in decidability.

1.4 Vector Addition Systems

We present now an alternate formulation for Petri Nets that we will use in the remainder of this work:

Definition 1.6. *A Vector Addition System (shortly: VAS) of dimension d is a tuple $\langle A, \delta \rangle$ where:*

- *A is the finite set of actions.*
- *$\delta : A \to \mathbb{Z}^d$ provides the effect of an action on the counters.*

To a Vector Addition System, one associates a f-LTS $\langle \mathbb{N}^d, \overline{A}, A, \gamma \rangle$ where:

- the functions $\overline{a}$ are defined by:

$$\begin{array}{rcl} dom(\overline{a}) & = & \{x \in \mathbb{N}^d \mid x + \delta(a) \geq 0\} \\ \overline{a}(x) & = & x + \delta(a) \end{array}$$

- γ is defined by:

$$\gamma(\overline{a}) = a$$

δ is extended by morphism on A^*. If $x \in dom(\overline{u})$, we say that u is fireable from x. An important property of Vector Addition Systems is that the effect of a sequence of transitions is independent from the starting state (that only determines whether the sequence of transitions is fireable) : if $x \xrightarrow{u} y$, then $y = x + \delta(u)$.

1.4.1 Vector Addition Systems with States

There is many ways of enriching Vector Addition Systems to make them more expressive or more practical. One that doesn't add expressive power but that will be quite practical is to add control states:

Definition 1.7. *A Vector Addition System with States (shortly: VASS) of dimension d is a tuple $\langle Q, A, \delta, tr \rangle$ where:*

- *Q is the finite set of control states.*

- *A is the finite set of actions.*
- *$\delta : A \to \mathbb{Z}^d$ provides the effect of an action on the counters.*
- *$tr : A \to Q \times Q$ provides the effect of an action on the control state.*

To a Vector Addition System with States, one associates a f-LTS $\langle Q \times \mathbb{N}^d, \overline{A}, A, \gamma\rangle$ where:

- $\overline{a}$ is defined by, if $tr(a) = (q, q')$:

$$\begin{array}{rcl} dom(\overline{a}) & = & \{q\} \times \{x \in \mathbb{N}^d \mid x + \delta(a) \geq 0\} \\ \overline{a}(q, x) & = & (q', x + \delta(a)) \end{array}$$

- γ is defined by:

$$\gamma(\overline{a}) = a$$

One can simulate a VASS by a VAS, for example by this way:

Proposition 1.6. *Let $\mathcal{V}_1 = \langle\{q_1, \ldots, q_k\}, A, \delta, tr\rangle$ be a VASS of dimension d and $\mathcal{V}_2 = \langle A, \delta'\rangle$ be a VAS of dimension $d + k$ where δ' is defined by:*

$$\begin{array}{rcll} \delta'(a)(i) & = & \delta(a)(i) & \text{if } 0 \leq i < d \\ \delta'(a)(d+i) & = & -1 & \text{if } 0 \leq i < k \text{ and } tr(a) = (q_i, q) \text{ with } q \neq q_i \\ \delta'(a)(d+i) & = & 1 & \text{if } 0 \leq i < k \text{ and } tr(a) = (q, q_i) \text{ with } q \neq q_i \\ \delta'(a)(d+i) & = & 0 & \text{if } 0 \leq i < k \text{ otherwise} \end{array}$$

Then if we define $\varphi : Q \times \mathbb{N}^d \to \mathbb{N}^{d+k}$ by $\varphi(q_i, x) = xe_{i,k}$ (the concatenation of x and the i-th unitary vector of length k), given $\mathcal{S}_1 = \langle X, A, \longrightarrow_1\rangle$ and $\mathcal{S}_2 = \langle X, A, \longrightarrow_2\rangle$ the transition systems associated to $\mathcal{V}_1$ and $\mathcal{V}_2$, we have:

$$x \xrightarrow{a}_1 y \iff \varphi(x) \xrightarrow{a}_2 \varphi(y)$$

It is also possible to simulate a VASS by a VAS by only increasing the dimension by 3. See for example [40].

Thus adding states doesn't add any expressive power. However, adding transfer, resets or other operations does, and we will see in section 1.8 that such operations have a significant impact on decidability.

1.5 Lossy Transition Systems

One of the possible ways to get a well-structured transition system is to approximate a transition system by allowing states to "lose value":

Definition 1.8. *A labelled transition system $\mathcal{S} = \langle X, A, \longrightarrow\rangle$ where X is ordered is* lossy *if for every $x, y \in X$ such that $y \leq x$, we have $x \xrightarrow{\varepsilon} y$.*

A labelled transition system $\mathcal{S}^- = \langle X, A, \longrightarrow_-\rangle$ is a lossy closure *of $\mathcal{S} = \langle X, A, \longrightarrow\rangle$ if $\mathcal{S}^-$ is lossy and $x \xrightarrow{u}_- y$ implies either $x \xrightarrow{u} y$ or $u = \varepsilon$ with $y \leq x$.*

The lossy closure of a labeled transition system is a WSTS with the same cover:

Proposition 1.7. *Let $\mathcal{S} = \langle X, A, \longrightarrow \rangle$ be a labeled transition system. Then, $\mathcal{S}^-$ is a well structured transition system such that for any $x \in X$, $Cover_{\mathcal{S}^-}(x) = Cover_{\mathcal{S}}(x)$.*

Other properties of the transition system like the reachability set or the recognized languages are not preserved by this approximation. However, in the case the original transition system was already a WSTS, the lossy closure preserves the coverability languages:

Proposition 1.8. *Let $\mathcal{S} = \langle X, A, \longrightarrow \rangle$ be a well-structured transition system. Then, for any $x, y \in X$, we have $L_c(\mathcal{S}^-, x, y) = L_c(\mathcal{S}, x, y)$.*

Note that if a VAS isn't lossy, its lossy closure (obtained by adding transitions that can decrease any counter) is still a VAS.

1.6 Usual problems for VAS and WSTS

We define here formally the problems that are most often considered for VAS and VAS extensions (more generally for WSTS). We will describe in this section some of these problems, and tell whether they are decidable for VAS. The next section will recall results for the general case of WSTS.

1.6.1 Reachability

The central problem for the verification of transition systems is reachability, which simply asks whether a state (say: an error state) is reachable from the initial state.

Decision Problem:	REACHABILITY
Input:	a TS $\mathcal{S} = \langle X, \longrightarrow \rangle$
	$x, y \in X$
Question:	is $y \in Reach_{\mathcal{S}}(x)$?

REACHABILITY for VAS is decidable. It is notable however that the decidability of this problem doesn't rely at all on the fact that VAS are WSTS but on proofs specific to the semantics of VAS. While the first proof was provided by Mayr and Kosaraju in the early 80's ([49], [42]), one can mention the recent proof by Leroux ([46], [47]) which is significantly easier to apprehend than the original one. The complexity of reachability is unknown (EXPSPACE lower bound by [19] but no upper bound).

1.6.2 Coverability

A problem related to reachability is coverability, a generally easier question:

Decision Problem:	COVERABILITY
Input:	a TS $\mathcal{S} = \langle X, \longrightarrow \rangle$ with X ordered
	$x, y \in X$
Question:	is $y \in Cover_{\mathcal{S}}(x)$?

It is straightforward to reduce COVERABILITY to REACHABILITY for VAS by adding transitions that decrease the counters (this reduction is generally available for most classes of systems). It is also possible to show the decidability of COVERABILITY directly through WSTS theory (see theorem 1 on page 26). Finally, a third separate proof by Rackoff [53] provides the result by bounding the length of possible covering sequences, providing an EXPSPACE algorithm. By combining this upper bound with the EXPSPACE lower bound of [19], this makes COVERABILITY EXPSPACE-complete.

It is notable that control state reachability is a special case of COVERABILITY, which makes it of particular interest for verification purposes, as it corresponds to checking whether a particular line of a program can be executed:

Decision Problem:	CONTROL STATE REACHABILITY
Input:	a TS $\mathcal{S} = \langle Q \times X, \longrightarrow \rangle$
	$q, q' \in Q$ and $x \in X$
Question:	$\exists x' \in X.\ (q, x') \in Reach_{\mathcal{S}}(q, x)$?

1.6.3 Boundedness and Termination

Reachability and coverability are safety problems, i.e. that require only to look at finite transition sequences. We can also look at liveness problems. One of these problems is termination, that is closely related to boundedness:

Decision Problem:	TERMINATION
Input:	a TS $\mathcal{S} = \langle X, \longrightarrow \rangle$
	$x_0 \in X$
Question:	is there $(x_n) \in X^\omega$ such that:
	$\forall n \in \mathbb{N}.\ x_n \longrightarrow x_{n+1}$

Decision Problem:	BOUNDEDNESS
Input:	a TS $\mathcal{S} = \langle X, \longrightarrow \rangle$
	$x \in X$
Question:	is $Reach_{\mathcal{S}}(x)$ finite?

Of course, a system can be unbounded only if it doesn't terminate. Moreover, assuming one can "count" the number of steps (this is the case for VAS), termination reduces to boundedness by adding a counter that goes unbounded on any infinite trace. This means that TERMINATION reduces to BOUNDEDNESS. However, the opposite is not true as one can find WSTS for which termination is decidable but boundedness is not, for example VAS extended with resets [23].

These problems are decidable for WSTS that enjoy some additionnal properties, see theorem 1 on page 26. In the case of VAS, one can either apply the general result on WSTS, or use the specific proof of Rackoff [53] to get that these problems are EXPSPACE-complete (needing [19] for the lower bounds)

1.6.4 Place-Boundedness

A generalization of boundedness for counter systems is place-boundedness, where we ask if a specific place is unbounded:

Decision Problem:	PLACE-BOUNDEDNESS
Input:	a TS $\mathcal{S} = \langle \mathbb{N}^d, \longrightarrow \rangle$
	$x \in \mathbb{N}^d$
	$i \in \{0, \ldots, d-1\}$
Question:	is $\{y(i) \mid y \in \mathit{Reach}_{\mathcal{S}}(x)\}$ finite?

It is clear that BOUNDEDNESS reduces to PLACE-BOUNDEDNESS. However, the inverse is false: for VAS with transfers, PLACE-BOUNDEDNESS is undecidable while BOUNDEDNESS is decidable [23]. This is due to the fact that VAS with transfers is a class that is not stable by projecting away counters (the closure by projection is VAS with resets). For Vector Addition Systems, PLACE-BOUNDEDNESS has been shown decidable (and EXPSPACE) by a generalization of the Rackoff proof ([64, 21, 13]).

In chapter 2, we will look at a problem called CLOVERABILITY, that can be seen as a generalization of PLACE-BOUNDEDNESS to transition systems using state spaces different from $\mathbb{N}^d$.

1.6.5 Repeated Control State Reachability

A generalization of termination is repeated control state reachability: does there exists a run that visits infinitely often a control state. This problem is generally equivalent to repeated coverability, that can be defined even in the absence of control states in the model.

Decision Problem:	REPEATED COVERABILITY
Input:	a TS $\mathcal{S} = \langle X, \longrightarrow \rangle$
	$x_0 \in X$, $y \in X$
Question:	is there $(x_n)_n \in X^\omega$ such that:
	$\forall n \in \mathbb{N}.\ x_n \longrightarrow x_{n+1}$
	$\{n \in \mathbb{N} \mid x_n \geq y\}$ infinite

Decision Problem:	REPEATED CONTROL STATE REACHABILITY
Input:	a TS $\mathcal{S} = \langle Q \times X, \longrightarrow \rangle$
	$(q_0, x_0) \in Q \times X$
	$q \in Q$
Question:	is there $(q_n, x_n)_n \in (Q \times X)^\omega$ such that:
	$\forall n \in \mathbb{N},\ (q_n, x_n) \longrightarrow (q_{n+1}, x_{n+1})$
	$\{n \in \mathbb{N} \mid q_n = q\}$ infinite

A common mistake[1] is to think that REPEATED CONTROL STATE REACHABILITY can be reduced to PLACE-BOUNDEDNESS by adding a counter that is increased whenever the place

[1]... which the author made himself more than once

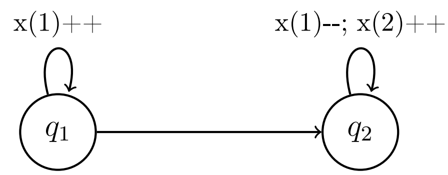


Figure 1.2: x_2 is unbounded, but q_2 can't be visited infinitely often

is visited. However, this reduction is not sound: the counter will be unbounded if for any $n \in \mathbb{N}$, one can find a run that visits the state n times, but this doesn't mean there exists a run that visits the state infinitely often. Figure 1.2 pictures such a Vector Addition System.

Proofs of decidability of REPEATED CONTROL STATE REACHABILITY usually rely on the detection of an increasing loop $(q, x) \longrightarrow\!\!\!\!\!\rightarrow (q, x')$ with $x' \geq x$. This problem was shown to be originally decidable by Esparza [25, 26], and EXPSPACE later by Habermehl [39]. One can also note that this problem can be expressed in the logic of [13] (shown to have EXPSPACE model checking).

1.6.6 LTL Model Checking

Linear-time logic is a widely used logic in order to express safety and liveness properties that is strongly related to REPEATED CONTROL STATE REACHABILITY.

Definition 1.9. *Given a set A, the set of LTL formulae is given by the following grammar, where a ranges over A :*

$$\varphi \quad ::= \quad \mathit{true} \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathcal{X}\varphi \mid \varphi_1 \mathcal{U} \varphi_2$$

Formulas are interpreted on infinite sequences over the alphabet A. We denote that $w = (a_n)_{n \in \mathbb{N}}$ satisfies a formula φ by $w \models \varphi$. This relation is defined inductively on the structure of φ by:

$$\begin{array}{lcllcl}
w & \models & \mathit{true} & & \\
w & \models & a & \iff & a_0 = a \\
w & \models & \neg\varphi & \iff & w \not\models \varphi \\
w & \models & \varphi_1 \wedge \varphi_2 & \iff & w \models \varphi_1 \text{ and } w \models \varphi_2 \\
w & \models & \mathcal{X}\varphi & \iff & w(1\ldots) \models \varphi \\
w & \models & \varphi_1 \mathcal{U} \varphi_2 & \iff & \exists i \in \mathbb{N}.\ w(i\ldots) \models \varphi_2 \wedge \forall j \in \{0,\ldots,i\}.\ w(j\ldots) \models \varphi_1
\end{array}$$

This allows us to define the following problem:

Decision Problem:	LTL MODEL CHECKING
Input:	a LTS $\mathcal{S} = \langle X, A, \longrightarrow \rangle$
	$x_0 \in X$
	φ a LTL formula on A
Question:	is there an infinite run σ of $\mathcal{S}$ such that:
	$src(\sigma) = x_0$
	$acts(\sigma) \models \varphi$

LTL formulas can be represented by Büchi automatas:

Definition 1.10. *A* Buchi automaton *is a tuple* $\langle Q, \longrightarrow, F\rangle$ *where* $\langle Q, \longrightarrow\rangle$ *is a finite automaton and* $F \subseteq Q$.

An infinite run $(q_0, x_0) \xrightarrow{a_1} (q_1, x_1) \cdots \xrightarrow{a_k} (q_k, x_k) \cdots$ of a Buchi Automata is accepted iff $\{i \in \mathbb{N} \mid q_i \in F\}$ is infinite. Given a LTL formula φ, one can build a Buchi automaton $\mathcal{B}_\varphi$ such that the set of infinite words satisfying φ is exactly the infinite words accepted by $\mathcal{B}_\varphi$. We refer to the abundant literature on this subject for the construction (Proposition 4.1 of [25], but also [36] and [34]).

Given a labelled transition system $\mathcal{S}$ and a formula φ, one can build $\mathcal{S} \times \mathcal{B}_\varphi$. Then, a state in F is covered infinitely often in $\mathcal{S} \times \mathcal{B}_\varphi$ iff φ is satisfied by $\mathcal{S}$. This the idea behind the following well known result, that we won't describe more here:

Proposition 1.9. *Let* $\mathbf{S}$ *be a class stable by product with a finite automata. Then,* LTL MODEL CHECKING *on* $\mathbf{S}$ *reduces to* REPEATED CONTROL STATE REACHABILITY *on* $\mathbf{S}$

Thus, LTL is EXPSPACE for Vector Addition Systems [25, 26, 39]. A related logic, called CTL, is however undecidable, as is LTL if predicates on states are added. We mention again [64], [21] and [13] as works that define some other decidable logics on Vector Addition Systems.

1.7 Decidability of WSTS problems

In order to be able to decide in general the previously defined problems for WSTS, we need to require that the transition functions have some effectiveness properties.

The most basic requirement is to require every function $f \in F$ to be computable (in the functional setting), or $\longrightarrow$ to be decidable (in the usual setting). This is the classic definition of *effective* for finite-branching WSTS, so we will say in that case that $\mathcal{S}$ is effective.

However, this requirement won't be enough to have any decidability result if we look at infinite-branching WSTS. Indeed, if we define the reasonable problem of testing membership in the $\downarrow Post_{\mathcal{S}}(x)$ set, we have:

Decision Problem:	POST MEMBERSHIP
Input:	a WSTS $\langle X, \longrightarrow\rangle$
	$x, y \in X$
Question:	is $y \in \downarrow Post_{\mathcal{S}}(x)$?

Proposition 1.10. *There exists a class of effective WSTS such that* POST-MEMBERSHIP *is not decidable.*

Proof. We will encode Turing machines into effective WSTS. Let M be a Turing machine with an accepting state. A finite run of M is accepted if it ends in the accepting state and rejected otherwise. It is well-known that one can not decide if a Turing machine has an accepting run.

To every Turing machine M, we associate a f-TS $\mathcal{S}_M = \langle X, F \rangle$ defined as follows:

$$\begin{array}{rcl} X & = & \{\textsc{init}, \textsc{accept}, \textsc{reject}\} \\ F & = & \{\delta_\rho \mid \rho \text{ is a finite run of } M\} \\ \delta_\rho(\textsc{init}) & = & \textsc{accept} \text{ if } \rho \text{ is accepting, } \textsc{reject} \text{ otherwise} \end{array}$$

This is a WSTS if X is ordered by the equality. It is effective, because given a run, one can look at the final state to see if it is accepting or not. However, one can not check if $\textsc{accept} \in Post_{\mathcal{S}_M}(\textsc{init})$. □

Therefore, a stronger requirement for our infinite-branching WSTS would be to require that POST-MEMBERSHIP is decidable. Indeed, we will see in the next sections that some of our decidability results will require this one. However, in the general settings, this is still not enough to get the decidability of the usual problems on WSTS:

Proposition 1.11. *There is a class of WSTS with decidable* POST-MEMBERSHIP *such that* TERMINATION *and* COVERABILITY *are undecidable.*

Proof. We reduce again this problem to acceptance for Turing machine.

To every Turing machine M, we associate a f-TS $\mathcal{S}_M = \langle X, F \rangle$ defined as follows:

$$\begin{array}{rcl} X & = & \{\textsc{init}\} \cup (\{\textsc{ready}, \textsc{accept}, \textsc{reject}\} \times \mathbb{N}) \\ F & = & \{f_{\text{ALLOC}(n)} \mid n \in \mathbb{N}\} \cup \{f_{\text{RUN}(n)} \mid n \in \mathbb{N}\} \cup \{f_{\text{LOOP}}\} \\ f_{\text{ALLOC}(n)}(\textsc{init}) & = & (\textsc{ready}, n) \\ f_{\text{RUN}(n)}(\textsc{ready}, n) & = & \begin{cases} (\textsc{accept}, n) & \text{if there exists an accepting run of length } n \\ (\textsc{reject}, n) & \text{otherwise} \end{cases} \\ f_{\text{LOOP}}(\textsc{accept}, n) & = & (\textsc{accept}, n) \end{array}$$

Because one can decide whether there exists an accepting run of a Turing machine of bounded length, POST-MEMBERSHIP is decidable. However, whether $(\textsc{accept}, 0) \in Cover_{\mathcal{S}_M}(\textsc{init})$ is undecidable, and there is an infinite run of $\mathcal{S}_M$ if and only if $(\textsc{accept}, 0) \in Cover_{\mathcal{S}_M}(\textsc{init})$. □

If one wants to get the decidability of COVERABILITY and TERMINATION for infinite-branching WSTS, one needs even stronger properties. Let us recall the PRED BASIS property [33] and define the symetrical POST BASIS property.

Decision Problem:	PRED BASIS for a class of WSTS **S**
Input:	$\mathcal{S} = \langle X, \longrightarrow \rangle$
	$x, y \in X$
Output:	$Min\ Pre_{\mathcal{S}}(\uparrow x)$

Computation Problem:	POST BASIS for a class of WSTS **S**
Input:	$\mathcal{S} = \langle X, \longrightarrow \rangle \in \mathbf{S}$
	$x \in X$
Output:	$Max\ Post_{\mathcal{S}}(\downarrow x) = Max\ Post_{\mathcal{S}}(x)$

Note that POST BASIS is only meaningful when $Max\ Post_{\mathcal{S}}(x)$ is finite (for example in the case of a finite-branching WSTS, for which it is equivalent to effectiveness). In the sequel, when we say that POST BASIS is computable, we imply it also exists. Let us recall the main results about WSTS:

Theorem 1. *([28, 33, 6])*

- COVERABILITY *is decidable for WSTS with decidable* PRED BASIS*.*
- TERMINATION *is decidable for WSTS with decidable* POST BASIS*.*

A system with decidable POST BASIS can be considered as a finite branching WSTS for most problems. Indeed, because of the monotony of the system, we can ignore all transitions from x that don't go towards a maximum element of $Max\ Post_{\mathcal{S}}(x)$. This gives us the following definition (originally found in [8]):

Definition 1.11. *A WSTS $\mathcal{S}$ is* essentially finite-branching *if for any $x \in States(\mathcal{S})$, $Max\ Post_{\mathcal{S}}(x)$ is finite.*

1.8 Summary of results for extensions of VAS

The decidability status of the various extensions of VAS have been extensively studied, and only a few results remain. A lot of results can be derived from the general publications on WSTS [6, 33]. There is also works aimed at some specific extensions. Among the most important of these results, one can note:

- The works of Dufourd *et al.* on reset arcs [23, 24] that showed that most problems turn to be undecidable with 3 resets, while boundedness is still decidable with 2 resets.
- A summary by Mayr [50] on results on counter machines, that include a proof of undecidability of repeated coverability with 2 resets or transfers.
- The works of Reinhardt [55] that showed that reachability was decidable with hierarchical zero-tests.

Moreover, one can note the following reductions:

- One reset arc or transfer arc can be simulated by one zero-test for any problem.
- Abstract data can simulate any number of reset arcs, for most problems (with the notable exception of boudedness) [56, 45].
- Reset arcs and transfer arcs can simulate zero-tests for reachability.

This allows to draw the following picture of the known decidability results:

	VAS	transfers			resets			abstract data	zero-tests	
		1	2	≥ 3	1	2	≥ 3		1	hier.
reachability	yes	yes	no	no	yes	no	no	no	yes	yes
coverability	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
termination	yes	yes	yes	yes	yes	yes	yes	yes	yes	?
boundedness	yes	yes	yes	yes	yes	yes	no	yes	yes	?
place-boundedness	yes	?	?	no	?	?	no	no	?	?
repeated coverability	yes	?	no	no	?	no	no	no	?	?

We aim in the next chapters to complete this view.

Part I

Theory of Well Structured Transition Systems

Chapter 2

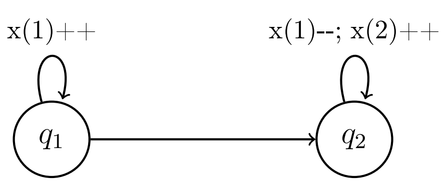
Forward Analysis

This chapter is joint work with Alain Finkel, unpublished.

The well-known Karp and Miller algorithm [41] constructs the coverability tree of a Vector Addition System by using regular accelerations. The information provided by such an algorithm is greater than the one provided by the backward algorithm of WSTS [33]: it doesn't only solve the COVERABILITY problem, but provides a finite representation of the cover. This finite representation can be used to answer other problems, for example BOUNDEDNESS or its refinement PLACE-BOUNDEDNESS.

Thus, we are interested in computing a finite representation of the cover. Typically, this representation is the set of maximal elements of $Lim\ Cover_{\mathcal{S}}(x)$ (for a suitable notion of limits, to be defined in next section), written $Clover_{\mathcal{S}}(x)$ [30] when it is finite. This gives the following problem:

Computation problem:	CLOVER SET
Input:	$\mathcal{S} = \langle X, F \rangle$
	$x \in X$
Output:	$Clover_{\mathcal{S}}(x)$



$$Clover(q_1, 0, 0) = \{(q_1, \omega, 0), (q_2, \omega, \omega)\}$$
$$Cover(q_1, 0, 0) = {\downarrow}Clover(q_1, 0, 0) \cap Q \times \mathbb{N}^2$$

Figure 2.1: A VASS and the finite representation of its cover

As an example, let us consider the VASS of figure 2.1. Its cover can be adequately represented by using $\mathbb{N}_\omega$, the *completion* of $\mathbb{N}$.

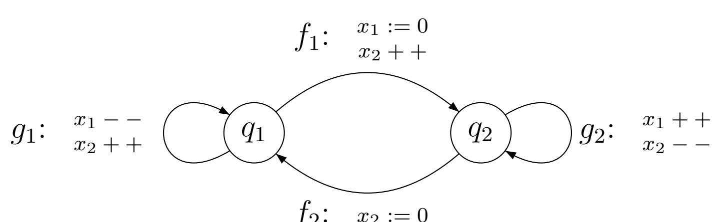
When one wants to extend the Karp-Miller tree to WSTS with a larger state space, one is faced with three main difficulties:

- Defining the notion of completion and limits for sets greater than $\mathbb{N}$.
- Extending the transition system (in a sensible way) to work on the completion instead of the normal state space.
- Making sure that the procedure terminates.

Unfortunately, there is no hope of having a general algorithm that provides such a representation for any WSTS. Indeed, BOUNDEDNESS is undecidable for some WSTS, for example VAS with resets [23]. Even if we require the WSTS to have strict monotonicity, which makes BOUNDEDNESS decidable [33], we can still have PLACE-BOUNDEDNESS undecidable, for example for ν-Petri Nets [56]. Despite this, Finkel and Goubault-Larrecq described a possible generalization in two steps:

- In [29], they established a theory of finite representations of closed sets (a restriction of downward-closed sets), so that such sets can be seen as the elements below a finite number of limit elements. Moreover, for an algebra that includes most of the state spaces used in WSTS, they provided a precise description of these limits.
- In [30], they proposed a systematic way to turn a WSTS in a *complete* WSTS, that has the same cover, and in which a conceptual Karp and Miller procedure can be run, computing the closure of the cover. However, this procedure was not guaranteed to terminate.

In [30], as in the original Karp-Miller algorithm, only regular accelerations were considered: when one detects a state x and a finite sequence of increasing functions $g \in F^*$ such that $x < g(x)$, one computes the limit $l = lub\ \{g^n(x) \mid n \in \mathbb{N}\}$. However, there might exist limits that can not be reached by such patterns. For example, the following VASS with (two) resets, first shown in [24] and also used in [**?**] is unbounded, but no regular acceleration would be able to show it:



$$(q_1, 0, 0) \xrightarrow{\prod_{k=1}^{n} f_1 g_2^k f_2 g_1^k} (q_1, 0, n)$$

From the initial state $(q_1, 0, 0)$, the infinite non-regular word $\prod_{k=1}^{\infty} f_1 g_2^k f_2 g_1^k$ is the unique possibility to obtain an unbounded counter x_2. Because regular expressions are not the only languages that can be enumerated, one can look, for models where these expressions are insufficient, at other possibility of accelerations. This will be the topic of this chapter.

After having recalled more formally the main results from [29] and [30], we will show a surprising result: with reasonable hypothesis, in order to be able to compute the maximal elements of the clover, it is sufficient to be able to enumerate its elements. We will then introduce a notion of acceleration strategy, and apply our results on a few models: strictly monotonic complete WSTS (section 2.6), VAS with 2 resets (chapter 4) and VAS with hierarchical zero-tests (section 5.3).

2.1 A bit of Order Theory

Directed subsets. A *directed subset* of X is a non-empty subset D such that every pair of elements of D has an upper bound in D. Chains, i.e. totally ordered subsets, are examples of directed subsets. A *directed complete partial ordering* (shortly: *dcpo*) is an ordering in which every directed subset has a least upper bound. The *way below* relation $\ll$ on a dcpo is defined by $x \ll y$ iff, for every directed subset D such that $y \leq lub(D)$, there is $z \in D$ such that $x \leq z$. We define $\twoheaddownarrow E = \{y \in X \mid \exists x \in E,\ y \ll x\}$. X is *continuous* iff for every $x \in X$, $\twoheaddownarrow x$ is a directed subset, and has x as least upper bound. If $\leq$ is a well order on X and turns X into a continuous dcpo, we say that X is a *continuous directed complete well-ordering* (shortly: *cdcwo*). Most of the sets in this section will be cdcwo.

Open and Closed sets. Given a dcpo X, and $E \subseteq X$, we define $Lim\ E = \{lub(D) \mid D$ directed subset of $E\}$. Note that $E \subseteq Lim\ E$. $Lim\ E$ can be thought of E plus all limits from elements of E. A subset D of a dcpo X is (Scott)-*closed* iff D is downward-closed and $Lim\ D \subseteq D$. An *open* subset is the complement of a closed subset. Particular cases of closed subsets that we will use are $\downarrow B$ (in any dcpo, for any finite set B) and $Lim\ D$ (in cdcwo only, for any downward closed subset D – see [29], proposition 3.5). Finally, we have the important property that, in a cdcwo X, any closed subset Y has a finite basis, i.e. a finite set $B \subseteq X$ such that $Y = \downarrow B$ (see [29], proposition 3.3).

Completions In [29] and [30], Finkel and Goubault-Larrecq showed that the usual state spaces of WSTS can be completed in a cdcwo in different ways, but that these are all equivalent. One of these construction is the *ideal completion* that associates to any ordered set X the set $Idl(X)$ made of the directed downward closed sets of X, ordered by inclusion. $Idl(X)$ is always a continuous dcpo ([9], Proposition 2.2.22), but might not be well-ordered ([30], Lemma 1). However, if we restrict ourselves to sets built from integers and finite sets by cartesian product, disjoint union, multiset, words, and trees, then this ideal completion will yield a cdcwo ([29], Theorem 5.3). X is embedded in $Idl(X)$ by $\eta : x \to \downarrow x$, so one can see this construction as "adding limit elements".

Example 2.1. $Idl(\mathbb{N}) = \{\downarrow x \mid x \in \mathbb{N}\} \cup \mathbb{N}$.

In this case, we have $\mathbb{N}$ *isomorphic to* $\{\downarrow x \mid x \in \mathbb{N}\}$ *through* η*, and* $Idl(\mathbb{N})$ *contains an extra element greater than all others, usually written* ω*.*

We give here another example of this completion, that we will need in a later proof:

Proposition 2.1. *([29], Theorem 5.3)*

Let $I \in Idl(X^{\oplus})$. I can be written as $\{| I_1^\omega, \cdots, I_p^\omega, J_1, \cdots, J_q |\}$ with $I_1, ..., I_p, J_1, ..., J_q \in Idl(X)$, and where:

$$M \in \{| I_1^\omega, \cdots, I_p^\omega, J_1, \cdots, J_q |\}$$
$$\Longleftrightarrow$$
$$\begin{cases} M = M_1 \cup ... \cup M_p \cup M'_1 \cup ... \cup M'_q \\ \forall 1 \leq k \leq p,\ x \in M_k \implies x \in I_k \\ \forall 1 \leq k \leq q,\ M_k = \emptyset \vee (M_k = \{| x |\} \wedge x \in J_k) \end{cases}$$

Example 2.2. $\{| 2^\omega, 5, \omega |\} \subseteq \mathbb{N}^{\oplus}$ *contains all multisets that have at most:*

- *One element of any value.*
- *One element of value 5 or lower.*
- *Any number of elements of value 2 or lower.*

2.2 Effectivity of orderings

We will require our cdcwo $(X, \leq)$ to be *effective*, i.e. to satisfy the following properties:

- X is recursively enumerable.
- Decidability: Given $x, y \in X$, one can decide whether $x \leq y$.
- Effective complement of open sets: Given a finite subset B such that $U = \uparrow B$ is open, one can compute a finite subset B' such that $\downarrow B' = X \backslash U$

(1) and (2) are natural requirement. If (3) is less standard, most ordered set will satisfy this property. Actually, all data types presented in [29] (sets built from finite sets and integers by disjoint union, cartesian products, words and multisets) even have *effective complement of upward closed sets* ([38], Definition 5 and Section 4), i.e. the complement of any upward closed set (not necessarily open) is computable [1].

From the effectivity of the complement of open sets, we get the effectivity of the complement of closed sets.

Proposition 2.2. *Let $(X, \leq)$ be an effective cdcwo. Then, for every finite subset $B \subseteq X$ such that $\downarrow B$ is closed, one can compute a finite subset $B' \subseteq X$ such that $\uparrow B'$ is the complement of $\downarrow B$ (i.e., $\uparrow B' = X \backslash \downarrow B$).*

Proof. The complement of $\downarrow B$ is an open set U. Now, if we guess a basis B' of U (i.e. a finite subset $B' \subseteq X$ such that $\uparrow B' = U$), one can get a basis B'' of the complement of $\uparrow B$ (with property (3) of effective ordered sets) and we may check whether $\downarrow B''$ is equal to $\downarrow B$ (it suffices to check that for any $b'' \in B''$, there exists $b \in B$ such that $b'' \leq b$, and that for

[1]Note however that the complement of an upward closed set may not have a finite basis, hence the question of representing it is non-trivial. We will not need such representations here, as we will manipulate only closed sets

any $b \in B$, there exists $b'' \in B''$ such that $b \leq b''$) and this is possible since $\leq$ is decidable. Hence, since X is recursively enumerable, one can enumerate all possible finite basis $B' \subseteq X$ of U, until a correct one is found. □

This allows us to have this important proposition, that will be our main tool to turn computation problems into decision problems:

Proposition 2.3. *If Y is an open or closed subset of an effective cdcwo X, a finite basis of Y can be computed iff membership in Y is decidable.*

Proof. If a basis is computable, membership is decidable by decidability of the order, so let us look at the other direction.

We consider a closed subset Y of a dcpo X such that membership in Y is decidable. As the finite subsets of Y are recursively enumerable, we only need to have a decision procedure to check whether, given a finite subset B of Y, if $Y = \downarrow B$. We note that $Y = \downarrow B$ if and only if:

(1) $\downarrow B \subseteq Y$.

(2) $(Y \setminus \downarrow B) \cap Y = \emptyset$

(1) corresponds to checking that all elements of B are in Y (because Y is downward closed), and this is decidable by hypothesis. Moreover, (2) is equivalent to checking that all minimal elements of $(Y \setminus \downarrow B)$ are not in Y, which is another finite set of instances of membership in Y. Note that one can compute the minimal elements of $Y \setminus \downarrow B$ by our assumptions on the effectivity of the orderings (proposition 2.2).

The case of an open subset is symetrical. □

For example, this proposition allows to confuse the notions of computable POST BASIS and effective POST MEMBERSHIP that were described in section 1.7 (we recall that our definition of a computable POST BASIS includes the fact that this basis exists and thus that the system is essentially finite branching).

Proposition 2.4. *Let* **S** *be a class of complete WSTS on an effective cdcwo.* POST BASIS *is computable if and only if* **S** *is essentially finite branching and* POST MEMBERSHIP *is decidable.*

Proof. This is a corollary of proposition 2.3. □

In the remainder of this chapter, we assume that all our cdcwo are effective.

2.3 Complete WSTS

Given a WSTS $\langle X, F \rangle$, if X is a cdcwo, we can guarantee that $Clover_{\mathcal{S}}(x)$ exists, as $Lim\ Cover_{\mathcal{S}}(x)$ is a closed set, and hence has a finite basis.

2.3.1 Continuous Transition Functions

However, simply requiring that our WSTS has a cdcwo as state space is not enough. Indeed, as we are trying to build an algorithm exploring $Lim\ Cover_{\mathcal{S}}(x)$ in a forward manner, we should hope that $Post_{\mathcal{S}}(Lim\ Cover_{\mathcal{S}}(x)) \subseteq Lim\ Cover_{\mathcal{S}}(x)$. However, this won't be the case in general:

Proposition 2.5. *There exists a WSTS $\mathcal{S}$ and $x \in States(\mathcal{S})$ such that: $Post_{\mathcal{S}}(Lim\ Cover_{\mathcal{S}}(x)) \not\subseteq Lim\ Cover_{\mathcal{S}}(x)$*

Proof. We define $\mathcal{S} = \langle \mathbb{N} \cup \{\omega, \omega_2\}, \{f\}\rangle$ by:

$$\begin{aligned} f(x) &= x+1 \qquad \text{for } x \in \mathbb{N} \\ f(\omega) &= \omega_2 \\ f(\omega_2) &= \omega_2 \end{aligned}$$

and where $\leq$ is defined by: $x < x+1 < \omega < \omega_2$ for $x \in N$.

We have $Cover_{\mathcal{S}}(0) = \mathbb{N}$, and hence that $Lim\ Cover(\mathcal{S}) = \mathbb{N} \cup \{\omega\}$. This means that $Post_{\mathcal{S}}(Lim\ Cover_{\mathcal{S}}(0)) \not\subseteq Lim\ Cover_{\mathcal{S}}(0)$. □

To avoid this problematic case, [30] required that the functions commute with the limits. We recall that a monotonic function $f : X \to Y$ is *continuous* if $dom(f)$ is an open subset of X and for any directed subset $U \subseteq X$, $lub(f(U)) = f(lub(U))$. Note that the composition of continuous functions is continuous.

Definition 2.1. *[30] A WSTS $\mathcal{S} = \langle X, F\rangle$ is* complete *if (1) X is a cdcwo, and, (2) every function f in F is continuous.*

In such complete WSTS $\mathcal{S}$, we will consider the different ways to access the elements of $Lim\ Cover_{\mathcal{S}}(x)$. Of course, it will happen that we want to consider the composition of such constructions, which will be made possible by the following proposition:

Proposition 2.6. *Let $\mathcal{S} = \langle X, F\rangle$ be a complete WSTS and $x, y, z \in X$ such that $y \in Lim\ Cover_{\mathcal{S}}(x)$ and $z \in Lim\ Cover_{\mathcal{S}}(y)$. Then, $z \in Lim\ Cover_{\mathcal{S}}(x)$.*

Proof. First, we have by induction on k that $Post^{k+1}_{\mathcal{S}}(Lim\ U) \subseteq Post^{k}_{\mathcal{S}}(Lim\ Post_{\mathcal{S}}(U)) \subseteq Lim\ Cover_{\mathcal{S}}(Cover_{\mathcal{S}}(U)) \subseteq Lim\ Post^{*}_{\mathcal{S}}(U)$. Thus, we get $Cover_{\mathcal{S}}(Lim\ U) \subseteq Lim\ Cover_{\mathcal{S}}(U)$. By taking $U = Cover(x)$, this leads to $Lim\ Cover(Lim\ Cover(U)) \subseteq Lim\ Cover(U)$ which is another formulation of the proposition. □

2.3.2 Completions of Vector Addition Systems

The completion of a Vector Addition System is straightforward: one just has to allow ω's in the states.

Definition 2.2. *Let $\mathcal{V} = \langle A, \delta\rangle$. The complete transition system associated to $\mathcal{V}$ is $\langle \mathbb{N}^d_\omega, \overline{A}\rangle$ where:*

$$\begin{aligned} dom(\overline{a}) &= \{x \in \mathbb{N}^d \mid x + \delta(a) \geq 0\} \\ \overline{a}(x) &= x + \delta(a) \end{aligned}$$

There is no specific difficulty in allowing ω's in the states. Indeed, if a state has an ω in some component, then all its successors will also have an ω in the same component. Moreover, this component can't prevent transitions to be fired, so everything happens as if we have projected away this component. Thus, all properties that are known decidable on the normal transition system of a Vector Addition System are also true for its completion.

Let us note that some care must be taken when looking at the completions of Vector Addition Systems extensions. If the completion of a Vector Addition System with resets is still a Vector Addition System with resets (we will show a close result in section 4.3), and the same is true for Vector Addition Systems with hierarchical zero-tests (for the same reasons as in basic Vector Addition Systems), this is not the case for Vector Addition Systems with transfers, that are strictly monotonic transition systems whose completions are not strictly monotonic: If a transfers the first component in the second, we have:

$$\begin{array}{ccc} (1,\omega) & \xrightarrow{a} & (0,\omega) \\ < & & = \\ (0,\omega) & \xrightarrow{a} & (0,\omega) \end{array}$$

Indeed, one can notice (we won't give a formal proof) that the completions of Vector Addition Systems with transfers actually behave in the same way as Vector Addition Systems with resets.

2.4 Computation of the Clover

We first note that we can relate CLOVER SET to a decision problem:

Decision problem:	CLOVERABILITY
Input:	a WSTS $\langle X, F\rangle$
	$x, y \in X$
Question:	is $y \in \downarrow Clover_{\mathcal{S}}(x)$?

Indeed, by taking $Y = Lim\ Cover_{\mathcal{S}}(x_0)$ in proposition 2.3, this means that CLOVER SET is computable iff CLOVERABILITY is decidable. However, in our case we may strengthen this result by replacing the decidability of cloverability by the *semi-decidability of cloverability*. We first show an easy lemma.

Lemma 2.7. *Let $\mathcal{S} = \langle X, F\rangle$ be a complete WSTS, and V a closed subset of $Lim\ Cover_{\mathcal{S}}(x)$ with $x \in V$. We have $V \subsetneq Lim\ Cover_{\mathcal{S}}(x)$ if and only there exists $y \in Max\ V$, $z \in Min\ (X\backslash V)$ such that $z \in \downarrow Post_{\mathcal{S}}(y)$.*

Proof. $\Rightarrow$ Let's assume that for any $y \in Max\ V$, we have $\downarrow Post_{\mathcal{S}}(y) \subseteq V$. Then, we have $\downarrow Post_{\mathcal{S}}(V) \subseteq V$, which leads by induction (V is an invariant of the transition relation) to $\downarrow Reach_{\mathcal{S}}(V) \subseteq V$. As $x \in V$, this contradicts $V \subsetneq Lim\ Cover_{\mathcal{S}}(x)$. Hence, there exists $z \notin V$, $z \in \downarrow Post_{\mathcal{S}}(x)$. Because this last set is downward closed, we can assume z to be minimal among the elements not in V, so $z \in Min\ (X\backslash V)$ and we have our result.

$\Leftarrow$ Let's assume that $V = Lub\ Cover_{\mathcal{S}}(x)$. Then, if we take $y \in Max\ V$, we have that $\downarrow Post_{\mathcal{S}}(y) \subseteq \downarrow Post_{\mathcal{S}}(Lim\ Cover_{\mathcal{S}}(x))$ which leads by proposition 2.6 to $\downarrow Post_{\mathcal{S}}(y) \subseteq Lim\ Cover_{\mathcal{S}}(x) = V$. Of course, this means that there cannot exist $z \notin V$, $z \in \downarrow Post_{\mathcal{S}}(y)$.

□

Theorem 2. *Let* **S** *be a class of complete WSTS with decidable* POST MEMBERSHIP. *If* CLOVERABILITY *is semi-decidable in* **S**, CLOVER SET *is computable in* **S**.

Proof. Since CLOVERABILITY is semi-decidable and the state space X is recursively enumerable, there is an algorithm enumerating an infinite sequence $x_0 = x$, x_1, x_2, ..., x_i, ... of all elements of $Lim\ Cover_{\mathcal{S}}(x)$. This yields an increasing (for $\subseteq$) sequence $V_i = \bigcup_{0 \leq j \leq i} \downarrow x_j$ of underapproximations of $Lim\ Cover_{\mathcal{S}}(x)$. This sequence will eventually stabilize to $Lim\ Cover_{\mathcal{S}}(x)$ since all the maximal elements of $Lim\ Cover_{\mathcal{S}}(x)$ will eventually be found.

Now, we need to be able to detect when we have reached the index i such that $V_i = Lim\ Cover_{\mathcal{S}}(x)$. To do this, we note that we have $V_i \subsetneq Lim\ Cover_{\mathcal{S}}(x)$ if and only if there exist $y \in Max\ V_i$ and $z \in Min\ (X \backslash V_i)$ such that $z \in \downarrow Post_{\mathcal{S}}(y)$ (by lemma 2.7). As the V_i are closed subsets of $Lim\ Cover_{\mathcal{S}}(x)$, our problem reduces to deciding whether $z \in \downarrow Post_{\mathcal{S}}(y)$ for a finite number of $y, z \in X$, which are instances of POST MEMBERSHIP. Thus, we can check when we have reached $V_i = Lim\ Cover_{\mathcal{S}}(x)$, and this makes our algorithm terminates. □

Note that the hypothesis of this theorem aren't restrictive at all. First, [29] (Section 6) showed that from most WSTS, one can build a complete WSTS with the same *Cover*. Secondly, in the most often encountered case of finite-branching WSTS, having computable transitions functions is enough to have decidable POST MEMBERSHIP.

This means the only difficuly part in order to show that a WSTS has a computable CLOVER SET is to show that CLOVERABILITY is semi-decidable. We will propose in the next section some ideas to show this in the general case while specific cases will be considered in chapters 4 and 5.

2.5 Acceleration Strategies

Theorem 2 tells us that in order to compute CLOVER SET, it is enough to find a way to enumerate the maximal elements of $Lub\ Cover_{\mathcal{S}}(x)$. A way to do this is to find a collection of potential witnesses, and enumerate these candidates, for example regular expressions built on the alphabet F. However, regular expressions are not the only possibility, and we define a more general notion of acceleration strategy, intended to be used for finite-branching WSTS, by:

Definition 2.3. *An* acceleration strategy *for a complete WSTS $\mathcal{S} = \langle X, F \rangle$ is a recursively enumerable set* STRAT *of computable and monotonic relations*[2] $h \subseteq X \times X$ *such that, for all $x \in X$, we have:*

[2] We use the function notation for these relations, as defined in chapter 1. When it is clear that h is a function from X to X, we will treat $h(x)$ as an element of X and not as a singleton subset of X.

- $h(x) \subseteq Lim\ Cover_{\mathcal{S}}(x)$
- $h(x)$ *is finite.*

For STRAT an acceleration strategy for $\mathcal{S}$, we define $Acc_{\mathcal{S}}^{\text{STRAT}}(x) = \{y \in h(x) \mid h \in (\text{STRAT} \cup F)^*\}$. This set defines a transitive relation: if we have $y \in Acc_{\mathcal{S}}^{\text{STRAT}}(x)$ and $z \in Acc_{\mathcal{S}}^{\text{STRAT}}(y)$, then we would also have $z \in Acc_{\mathcal{S}}^{\text{STRAT}}(x)$. Moreover, because we have that $Lim\ Cover_{\mathcal{S}}(Lim\ Cover_{\mathcal{S}}(x)) = Lim\ Cover_{\mathcal{S}}(x)$ (by proposition 2.6), we get that $Acc_{\mathcal{S}}^{\text{STRAT}}(x) \subseteq Lim\ Cover_{\mathcal{S}}(x)$. Hence, this set corresponds to the element of the $Lim\ Cover_{\mathcal{S}}(x)$ which are reachable thanks to the acceleration strategy.

As a first example of an acceleration strategy, we look at the traditional accelerations used in a Karp and Miller tree. They consist, from a state $x \in \mathbb{N}_\omega^d$, in iterating a finite sequence $g \in F^*$ of transitions. In this case, one adds to the Karp-Miller tree a new node labeled by $lub\ \{g^n(x) \mid n \in \mathbb{N}\}$. This is possible because $(g^n(x))_n$ is an increasing sequence in a cdcwo, hence it has a lub. We generalize this construction to any complete WSTS: given $g \in F^*$, we define $g^\infty : X \to X$ by $dom(g^\infty) = dom(g)$ and $g^\infty(x) = lub\ \{g^n(x) \mid n \in \mathbb{N}\}$ if $x < g(x)$, and $g(x)$ otherwise. The function g^∞ is well-defined as $(g^n(x))_{n\in\mathbb{N}}$ is an increasing sequence. This function g^∞ is monotonic if g is monotonic, and for a set F of monotonic functions, we define $\text{ITER}(F) = \{g^\infty \mid g \in F^*\}$, $\text{ITER}^k(F) = \text{ITER}(\text{ITER}^{k-1}(F) \cup F)$, and $\text{ITER}^\infty(F) = \bigcup_{k\in\mathbb{N}} \text{ITER}^k(F)$.

If the functions g^∞ are computable for VAS, nothing guarantees that they will be computable for any other WSTS. For this reason, [30] defines ∞-*effective* WSTS that are WSTS, where for any $x \in X$ and $g \in F^*$ with $g(x) > x$, $lub\ \{g^n(x) \mid n \in \mathbb{N}\}$ is computable. Hence, $\text{ITER}(F)$ is an *acceleration strategy* for any ∞-effective complete WSTS.

A first question one can ask is whether $\text{ITER}^\infty(F)$ is more powerful than $\text{ITER}(F)$ (i.e. when running a Karp-Miller tree, is it useful to consider any ancestor on a branch, or only ancestors that are not separated from the current node by an acceleration). The answer is that in the particular case of VAS, these strategies are equivalent, but this correspondance is not verified in general:

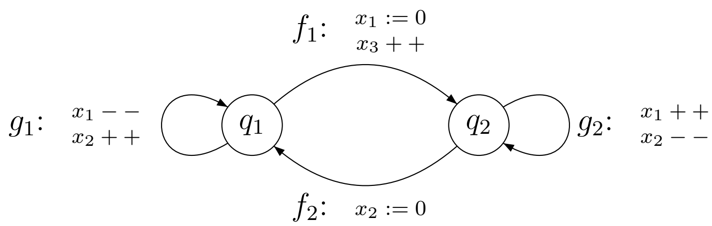
Proposition 2.8. *For $\mathcal{S} = \langle \mathbb{N}_\omega^d, F\rangle$ the labeled transition system of a VAS, and $x \in \mathbb{N}_\omega^d$, we have $Acc_{\mathcal{S}}^{\text{ITER}(F)}(x) = Acc_{\mathcal{S}}^{\text{ITER}^\infty(F)}(x)$.*

Proof. We have $x \in \mathbb{N}_\omega^d$ and $h \in (\text{ITER}^\infty(F) \cup F)^*$, and we need to show that there exists $h' \in (\text{ITER}(F) \cup F)^*$ such that $h(x) = h'(x)$.

We have $h = f_0 g_1^\infty f_1 g_2^\infty \dots g_n^\infty f_n$ with $f_i \in F^*$ and $g_i \in \text{ITER}^\infty(F)$. We define $x_k = f_0 g_1^\infty f_1 \dots g_k^\infty f_k(x)$. We can consider that $g_{k+1}(x_k) \geq x_k$, otherwise we would have $x_{k+1} = g_{k+1} f_{k+1}(x_k)$ and hence a simpler decomposition. Thus, the only effect of g_k is to add ω to some components of x_k. Because $g_k(x_k) \in Lub\ Cover_{\mathcal{S}}(x_k)$, we consider a sequence $g'_{k,\ell} \in F^*$ such that $lub\ (g'_{k,\ell}(x_k))_{\ell\in\mathbb{N}} = g_k(x_k)$. But, because $g'_{k,\ell}(x_k)$ has the same component equal to ω than x_k, and that $g_k(x_k) \geq x_k$, there exists p_k such that we get $g'_{k,p_k}(x_k) \geq x_k$, and for each $i \in \{1, \dots, d\}$, $x_k(i) < g_k(x_k)(i) \iff x_k(i) < g'_{k,p_k}(x_k)$. Then, we have ${g'}_{k,p_k}^{\infty}(x_k) = g_k^\infty(x_k)$, and this concludes our proof, by taking $h' = f_0 {g'}_{1,p_1}^{\infty} f_1 \dots {g'}_{n,p_n}^{\infty} f_n$ with $f_k, g'_{k,p_k} \in F^*$. ☐

Proposition 2.9. *There exists a complete WSTS $\mathcal{S} = \langle X, F\rangle$ and $x \in States(\mathcal{S})$ such that $Acc_{\mathcal{S}}^{\text{ITER}(F)}(x) \subsetneq Acc_{\mathcal{S}}^{\text{ITER}^\infty(F)}(x)$.*

Proof. We consider the following system (a VASS with two resets):



Then, we have $(g_1^\infty f_1 g_2^\infty f_2)^\infty(\omega, 0, 0) = (\omega, 0, \omega)$, but there exists no $h \in \textsc{iter}(F)$ such that $h(\omega, 0, 0) = (\omega, 0, \omega)$ (there must be a loop using the two resets, but this loop can't preserve the ω in the first two component, because its transitions are only normal transitions).

Another more abstract system that will even require to go up to $\textsc{iter}^k(F)$ is the WSTS $\langle \mathbb{N}_\omega^d, F, \leq_{lex} \rangle$ where $\leq_{lex}$ is the lexicographic ordering on vectors, $F = \{f_1, \ldots, f_d\}$ and f_i with $dom(f_i) = \mathbb{N}_\omega^d$ is defined by:

$$\begin{array}{lrcll} & f_i(x)(i+1) & = & x(i+1)+1 & \\ \text{if } x(i) \geq x(i+1): & f_i(x)(i) & = & 0 & \\ & f_i(x)(j) & = & x(i) & \text{for } j \notin \{i, i+1\} \\ & & & & \\ \text{otherwise:} & f_i(x) & = & x & \end{array}$$

The condition $(\mathcal{C}) : x(i) \geq x(i+1)$ looks unusual in the definition of a WSTS, but it doesn't prevent monotony here. Indeed, we consider $x \leq_{lex} y$. Four cases might occur depending on whether x and y fulfill the condition $(\mathcal{C})$. The two where x and y have the same status is immediate, so we look at the two others.

- x satisfies $\mathcal{C}$ and y doesn't. Then, it means that we have $y(i+1) \geq_{lex} x(i+1) + 1$, as increasing this component is the only way to deactivate $\mathcal{C}$. But, this means that $y \geq_{lex} f_i(x)$.

- x doesn't satisfy $\mathcal{C}$ and y does. Then, we have that $f_i(y) \geq_{lex} y \geq x$ and $f_i(x) = x$.

Now, one can check by induction on k that functions of $\textsc{iter}^k(F)$ can only add up to k ω. ☐

However, iterating a single sequence and computing the *exact lub* is not the only way to compute the clover.

- We may consider other languages than the regular language $L = g^*$ of the iteration of a sequence g. For instance, we may consider non-regular languages L, for example the set of finite prefixes of the infinite word $aba^2b^2a^3b^3...a^nb^n...$ and we may use functions of the form $h_L(x) = lub\ \{g(x) \mid g \in L\}$ if the set $\{g(x) \mid g \in L\}$ is directed or even relations $h_L(x) = Max\ Lim\ \{g(x) \mid g \in L\}$.

- We may also under-approximate the lub when it is non-computable. For example, the *strongly increasing* ω*-well-structured nets* were defined in [31], and the set $\textsc{under-iter}(F) = \{h_g \mid g \in F^*\}$ with $dom(h_g) = \{x \in \mathbb{N}_\omega^d \mid x < g(x)\}$ where h_g is defined by $h_g(x)(i) = \omega$ if $x(i) < g(x)(i)$ and otherwise $h_g(x)(i) = g(x)(i)$, was shown to be an acceleration strategy, and sufficient to compute the clover, while the functions of $\textsc{iter}(F)$ could be non-computable.

Those acceleration strategies allow us to define procedure 1, which is a Karp-Miller tree parameterized by an acceleration strategy. This algorithm explores the reachability set, stopping branches on states that are lower than a state already present in the tree [3] (lines 4 to 6). We iterate on all the nodes (lines 7 to 11), and try the various acceleration strategies on these in a *fair* way: if line 9 is executed infinitely often for the same node n, then any $h \in \textsc{strat}$ is chosen infinitely often.

Procedure 1 A parameterized Karp-Miller tree

Inputs: $\mathcal{S} = \langle X, F \rangle$, a complete WSTS
$\textsc{strat}$, an acceleration strategy on $\mathcal{S}$
$x_0 \in X$, the initial state

1: $\mathcal{T} \leftarrow$ a tree with a single root n_0, labeled by x_0. $N \leftarrow \{n_0\}$
2: **while** $N \neq \emptyset$ **do**
3: Remove a node n of label x from N
4: **for all** $x' \in Post_{\mathcal{S}}(x)$ **do**
5: **if** $\neg\exists n_a$ a node of $\mathcal{T}$ of label x_a with $x_a \geq x'$ **then**
6: Add a new child n', of label x', to n in $\mathcal{T}$ and add n' to N
7: **for all** n, node of $\mathcal{T}$ **do**
8: $x \leftarrow label(n)$
9: Pick fairly $h \in \textsc{strat}$ and $y \in h(x)$
10: **if** $\neg\exists n_a$ a node of $\mathcal{T}$ of label x_a with $x_a \geq y$ **then**
11: $label(n) \leftarrow y$. Add n to N
12: **return** $Max\ \{label(n) \mid n \in \mathcal{T}\}$

This procedure cannot be guaranteed to terminate when applied to complete WSTS since it would allow to decide the boundedness problem for Reset Petri nets which is undecidable. Moreover, for the same reason (developed in [30] for a similar clover procedure), one cannot decide whether this procedure terminates, even when the strategy parameter is fixed to be $\textsc{iter}(F)$.

The following lemma describes the invariant fulfilled by the branches of the tree during the execution of the procedure, and comes directly from the transitivity of *Acc*:

Lemma 2.10. *At any point of the procedure, let n and n' be two nodes such that n' is a descendant of n. Moreover, let x be the initial label of n (before any update), and x' the current label of n'. Then, $x' \in Acc_{\mathcal{S}}^{\textsc{strat}}(x)$.*

[3] Unlike traditionnal Karp-Miller, we are not comparing only to the ancestor nodes. This doesn't change correctness, but may improve termination in some case. See [30] for an example.

This lemma allows to prove the correctness of the procedure, in a similar way as for the usual Karp-Miller tree:

Proposition 2.11. *If procedure 1 terminates with output B, then $B = Clover_{\mathcal{S}}(x_0)$.*

Proof. We get directly $\downarrow B \subseteq Lim\ Cover_{\mathcal{S}}(x_0)$ from lemma 2.10, so let us show that $Lim\ Cover_{\mathcal{S}}(x_0) \subseteq \downarrow B$. We take $y \in Cover_{\mathcal{S}}(x_0)$ and we reason by induction on the length n of the number of functions $f_1, f_2, \ldots, f_n \in F$ such that $y \leq f_1 f_2 \ldots f_n(x_0)$. The induction hypothesis implies that there exists $x_0' \in \downarrow B$ such that $y \in Post_{\mathcal{S}}(x_0')$. Note that whenever a node is updated or created, it is added to N, which means that because $x_0' \in \downarrow B$, the loop of lines 4 to 6 is executed for a node n of label x with $x \geq x_0'$. Thus there exists $y' \geq y$, with $y' \in Post_{\mathcal{S}}(x)$, and we either have that there exists a node of $\mathcal{T}$ whose label is greater than y', or a node of label y' is added. As the labels of nodes can only increase, this concludes this part of the demonstration. ☐

If the procedure is not guaranteed to terminate, one can relate its termination to the closure properties of $Acc_{\mathcal{S}}^{\text{STRAT}}(x)$:

Lemma 2.12. *Let $\mathcal{S}$ be a complete WSTS and* STRAT *an acceleration strategy for $\mathcal{S}$. If, for any strictly increasing sequence $(y_n)_{n\in\mathbb{N}}$ with $y_{n+1} \in Acc_{\mathcal{S}}^{\text{STRAT}}(y_n)$, we have $lub\ \{y_n \mid n \in \mathbb{N}\} \in Acc_{\mathcal{S}}^{\text{STRAT}}(y_0)$, then procedure 1 terminates on input $\mathcal{S}$,* STRAT *and any $x_0 \in States(\mathcal{S})$.*

Proof. Let us assume the procedure does not terminate. First, we prove that if x is the label of a node at some point of the algorithm, for any $y \in Acc_{\mathcal{S}}^{\text{STRAT}}(x)$, eventually a label $y' \geq y$ will be present in the tree (and because labels only increase, the existence of such a label stays true from that point). We do this by induction on the length of $h = f_1 \ldots f_n$ such that $y \in h(x)$. By induction hypothesis, at some point a label greater than $f_1 \ldots f_{n-1}(x)$ appears. The node with this label is added to N, so if $f_n \in F$, the loop of lines 4 to 6 will be executed, which means that a label greater than y will be added if it is not already present. Similarly, if $f_n \in$ STRAT, the loop of lines 7 to 11 is executed infinitely often, which means by fairness that f_n will eventually be picked.

Now, we consider the following two cases:

- A node is updated infinitely often. This means that during the execution of the procedure, the node has been successively updated to values $(x_n)_{n\in\mathbb{N}}$ with $x_{n+1} = f_n(x_n)$, $f_n \in$ STRAT. Let $y = lub\{x_n \mid n \in \mathbb{N}\}$. Since $x_{n+1} \in Acc_{\mathcal{S}}^{\text{STRAT}}(x_n)$, one deduces from the hypothesis of the lemma that there exists $y \in \downarrow Acc_{\mathcal{S}}^{\text{STRAT}}(x_0)$, which means that a label $\geq y$ would eventually be present in the tree, and which contradicts the fact that the node continues to be update to values $x_n \leq y$.

- No node is updated infinitely often. This means that each node is added to N only a finite number of times, and hence go through the loop of lines 4 to 6 only a finite number of times. Hence, the generated tree is finitely-branching. By Konig's lemma, this means that if the procedure doesn't terminate, there is an infinite branch being created. In this branch, by the well-ordering on X, we can find a strictly increasing subsequence of labels $(x_i)_{i\in\mathbb{N}}$. But, by lemma 2.10, we have $x_{n+1} \in Acc_{\mathcal{S}}^{\text{STRAT}}(x_n)$ and by the initial remark, this would violate the fact that $y = lub\ \{x_n \mid n \in \mathbb{N}\}$ would eventually be present in the tree.

□

Note that if $\downarrow Acc_{\mathcal{S}}^{\text{STRAT}}(x)$ is closed for any x, the condition of this lemma is verified. Moreover, if the condition of the lemma is verified, the algorithm terminates, which means that $\downarrow Acc_{\mathcal{S}}^{\text{STRAT}}(x) = Lub\ Cover_{\mathcal{S}}(x)$ (because of lemma 2.10). Hence, $\downarrow Acc^{\text{STRAT}}(x)$ closed for any x is another possible sufficient condition for the termination of the algorithm.

2.6 Application on Strictly Monotonic Complete WSTS

In [30], the authors defined the notion of *clover-flattable*, that is equivalent to the termination of a conceptual Karp-Miller like procedure they introduced, and that is an instance of our procedure with $\text{STRAT} = \text{ITER}(F)$. However, they showed that determining if a system is clover-flattable is undecidable. This prompts us to search for a simpler criteria, that can be checked directly by looking at the state space and the functions.

It is known that strictly monotonic WSTS enjoy additionnal decidability properties: BOUNDEDNESS is guaranteed to be decidable. When the WSTS is complete, ∞-effective and in a "small" state space, we get an additionnal result:

Theorem 3. *For ∞-effective strictly monotonic complete WSTS $\mathcal{S} = \langle X, F\rangle$ such that $X = \mathbb{N}_\omega^d$, the procedure 1 terminates on inputs ($\mathcal{S}$, $\text{ITER}(F)$, x).*

Proof. We consider $\mathcal{S} = \langle \mathbb{N}_\omega^d, F, \leq\rangle$ and we want to show $\downarrow Acc_{\mathcal{S}}^{\text{ITER}(F)}(x) = Lub\ Cover_{\mathcal{S}}(x)$. We take $x \in \mathbb{N}_\omega^d$ and $\ell \in Max\ Lub\ Cover_{\mathcal{S}}(x)$. We want to show that there exists $h \in (\text{ITER}(F) \cup F)^*$ such that $h(x) \geq \ell$. We will assume that $\ell \notin Cover_{\mathcal{S}}(x)$, or the result is immediate.

Without loss of generality, we'll only consider runs $\gamma : x \xrightarrow{f_1} x_1 \xrightarrow{f_2} x_2 ... \xrightarrow{f_n} x_n$ such that for all $i \neq j$, $x_i \neq x_j$.

For such runs, we define $\alpha(\gamma) = (x, x_1)(x_1, x_2)...(x_{n-1}, x_n) \in (\mathbb{N}_\omega^d \times \mathbb{N}_\omega^d)^*$, that we order by $\sqsubseteq^{emb}$ with $\sqsubseteq$ defined by:

$$(x, y) \sqsubseteq (x', y') \iff x < x' \quad \text{or,} \begin{cases} x = x' \\ y = y' \end{cases}$$

Let's show that $\sqsubseteq$ is a well-ordering on the pairs that might appear in a run. Indeed, assume that we have an infinite sequence of $\mathbb{N}_\omega^d \times \mathbb{N}_\omega^d$. By well-ordering on $\mathbb{N}_\omega^d$, we can extract an infinite subsequence such that the first component is increasing. From that point, either we extract an infinite subsequence such that the first component is strictly increasing (which leads to an increasing subsequence for $\sqsubseteq$), or an infinite subsequence such that the first component is stationnary. But, the number of possible x_{k+1} for a given x_k, $k \in \{0, \ldots, n-1\}$, is finite, hence we get an increasing subsequence for $\sqsubseteq$. This leads to $\sqsubseteq^{emb}$ well-ordering on runs.

Now, back at our initial problem, we know that there exists runs $(\gamma_i)_{i\in\mathbb{N}}$ with $src(\gamma_i) = x$, $tgt(\gamma_i)$ strictly increasing (we can find it strictly increasing because $\ell \notin Cover_{\mathcal{S}}(x)$) and $lub\{tgt(\gamma_i)\} = \ell$. Hence, we can extract an infinite increasing subsequence of such runs for $\sqsubseteq^{emb}$. Let $x \longrightarrow x_1 \longrightarrow x_2 \ldots \longrightarrow x_k$ be the smallest common prefix of these runs in this sequence. This means there exists two runs γ_i and γ_j, $i < j$ such that $\gamma_i = x \longrightarrow \ldots x_k \longrightarrow x_{k+1}\gamma'_i$ and $\gamma_j = x \longrightarrow \ldots x_k \longrightarrow x'_{k+1}\gamma'_j$ with $x_{k+1} \neq x'_{k+1}$. By well-ordering, because $(x_k, x_{k+1}) \not\sqsubseteq (x_k, x'_{k+1})$, we have $(x_k, x_{k+1})\alpha(\gamma'_i) \sqsubseteq^{emb} \alpha(\gamma'_j)$. Hence, $\gamma'_j = x'_{k+1} \longrightarrow \ldots \longrightarrow x'_{k+r} \longrightarrow x'_{k+r+1} \ldots$ with $x_k \leq x'_{k+r}$. This means we have $x_k \xrightarrow{f} x'_{k+r}$ with $x_k < x'_{k+r}$. (because, by our earlier condition on the runs, $x_k \neq x'_{k+r}$). We define $\ell' = f^\infty(x_k)$. Because, $x_k < \ell'$ and we had runs sourcing from x_k and whose set of targets had ℓ has least upper bound, we still have $\ell \in \mathit{Lub}\ \mathit{Cover}_\mathcal{S}(\ell')$, and because ℓ was a maximal element of the cover, this is still true, leading to $\ell \in \mathit{Max}\ \mathit{Lub}\ \mathit{Cover}_\mathcal{S}(\ell')$. So we have $x \xrightarrow{f_1 \ldots f_k} x_k \xrightarrow{f^\infty} \ell'$. If $\ell \in \mathit{Cover}_\mathcal{S}(\ell')$, we have shown our result. If not, we can restart this construction, starting from ℓ' instead of x. Let's show that this procedure will end in at most d steps.

Assume that the previous procedure had built sequences $y_1, \ldots y_{d+1}$ ℓ_1, ℓ_{d+1}, with $y_i < \ell_i$ such that there exists $g_i, h_i \in F^*$ $(1 \leq i \leq d+1)$, with:

$$x \xrightarrow{g_1} y_1 \xrightarrow{h_1^\infty} \ell_1 \xrightarrow{g_2} y_2 \xrightarrow{h_2^\infty} \ell_2 \cdots \xrightarrow{g_{d+1}} y_{d+1} \xrightarrow{h_{d+1}^\infty} \ell_{d+1}$$

By continuity, if $y_{d+1} \in \mathit{dom}(h_{d+1}^\infty)$, there exists $z_{d+1} \in \mathbb{N}^d$, $z_{d+1} < y_{d+1}$ such that $z_{d+1} \in \mathit{dom}(h_{d+1}^\infty)$. We define $m_{d+1} = h_{d+1}(z_{d+1})$. Again by continuity, because any sequence converging toward y_{d+1} will eventually be greater than z_{d+1}, this means that there exists $m_d \in \mathbb{N}^d$, $m_d < \ell_d$ such that $m_d \xrightarrow{g_{d+1}} z_d$. Continuing the same reasoning, there exists $z_d \in \mathbb{N}^d$, $z_d < y_d$ such that $m_d \leq g_d^\infty(z_d)$. Because $m_d \in \mathbb{N}^d$, there exists k such that $m_d \leq g_d^k(z_d)$. Without loss of generality, we will assume that $z_d \xrightarrow{g_d^k} m_d$ (for example by increasing z_{d+1} and m_{d+1} in a suitable way). Iterating this construction, this builds the following run (for $1 \leq i \leq d+1$, $r_i \in \mathbb{N}$):

$$x \xrightarrow{g_1} z_1 \xrightarrow{h_1^{r_1}} m_1 \xrightarrow{g_2} \cdots \xrightarrow{g_{d+1}} z_{d+1} \xrightarrow{h_{d+1}^{r_{d+1}}} m_{d+1}$$

Now, let us consider replacing $h_1^{r_1}$ by h_1^∞ in the previous run. By strict monotony, each time we add an iteration of the loop, we increase strictly m_{d+1}. As we can do this an unbounded number of times, this would add at least one ω to m_{d+1}. Then, we can iterate the second loop, adding at least another ω, and as we can do this $d+1$ times, we get a contradiction.

Hence the procedure of the first part terminates in at most d steps, and we get a run $x \xrightarrow{g_1 h_1^\infty g_2 h_2^\infty \cdots g_n h_n^\infty g_{n+1}} \ell$. □

Petri nets and Post-Self-Modifying nets [61] are ∞-effective strictly monotonic complete WSTS with a state space equal to $\mathbb{N}_\omega^d$. The clover is non-computable for strictly monotonic non-complete WSTS (example: Transfer Nets, whose completions are identical to the completions of Reset Nets) and for non-strictly monotonic complete WSTS, (see Theorem 5.14 in [31]) both with $\mathbb{N}_\omega^d$ as state space. A result similar to Theorem 3 can be found in [28] (Theorem 4.18) but the completeness hypothesis was missing and the effectivity hypothesis were not sufficiently explicited. [31] considers also *strongly* increasing complete WSTS and uses the strategy UNDER-ITER(F) described above to show the computability of the cover for these systems. However, strongly increasing is a stronger requirement than just strictly increasing, so theorem 3 is the first result on the computability of the clover for ∞-effective strictly monotonic complete WSTS

Chapter 3

Expressiveness

This chapter is joint work with Alain Finkel, Serge Haddad and Fernando Rosa Vellardo, originally published in [16].

As we have seen in the previous chapters, many classes of WSTS have been defined. A lot of these classes are (syntaxic) extensions of other ones, for example Petri Nets can be extended by adding new types of arcs. A natural question that arise is whether these extensions are useful, i.e. if the new class can exhibit behaviours that the basic class could not.

In order to answer this question, we need to precise what we mean by *behaviours*, as various definitions have been used in related work [4, 5, 58]. The four usual ones are the languages L_r (reachability), L_c (coverability), L_t (finite traces) and L_t^ω (infinite traces) that we defined earlier. Thus, we have four ways to compare expressiveness of WSTS classes. Let us explain that two of these are unsuitable for our study. Indeed, reachability is generally undecidable in Petri Nets extensions [23], and such extensions will be able to recognize any recursively enumerable language if we consider reachability languages. Similarly, repeated coverability is undecidable for these extensions, which makes membership in L_t^ω undecidable. For these reasons, it is sensible to compare WSTS based on their coverability languages or trace languages. These two languages are strongly related:

- The trace language is the finite union of the coverability languages where the final states are the minimum elements of the set.
- Assuming one can define transition sequences u that test whether the current state is greater than x_f (which is the case for most WSTS classes), we have $L_c(\mathcal{S}, x_0, x_f) = L_t(\mathcal{S}, x_0) \cap A^*u$.

Because of this, classes of WSTS will often recognize the same trace languages and coverability languages (if they are stable by finite union and product by a finite automata). For technical reasons, we will focus in this chapter on coverability languages, which are also those that are studied in [35, 4, 5, 58]. This gives us the following definition:

Definition 3.1. *Let $\mathbf{S}_1$ and $\mathbf{S}_2$ be two classes of WSTS. We write $\mathbf{S}_1 \preceq \mathbf{S}_2$ whenever for every language $L_c(\mathcal{S}_1, x_1, x_1')$ 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 x_2, x_2' two states of $\mathcal{S}_2$ such that $L_c(\mathcal{S}_2, x_2, x_2') = L_c(\mathcal{S}_1, x_1, x_1')$.*

Thus, we will try to compare the coverability languages that can be obtained by various models. It is important to see that the expressive power of a 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 interwined in the proofs, which meant one specific proof was required for each model. The aim of this chapter is to separate them in order to have a formal and generalizable method.

We will look more specifically at the state spaces. Such a study is related to the relevance of resources: does adding additional resources (counters, channels, tapes, clocks, stacks, etc...) actually yield an increase in expressiveness. For example, if we look at Timed Automata, clocks are a strict resource: Timed Automata with k clocks are less expressive that Timed Automata with $k+1$ clocks [10]. 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 [27]. Thus, we aim to prove some results that will provide us with tools to establish strict relations between classes of WSTS that have distinct state spaces.

To be as general as possible, we consider in this chapter WSTS with weak monotonicity. These WSTS will be labelled by an alphabet A, with any word of A^* being allowed to be a single transition label.

3.1 A bit of ordinal theory

We will use in this chapter set theoretical ordinals. Let us recall a few properties of these objects. The class of ordinals is totally ordered by inclusion, and each ordinal α is equal to the set of ordinals $\{\beta \mid \beta < \alpha\}$ below it. Every totally well ordered set X is isomorphic to a unique ordinal $ot(X)$, called the *order type* of X.

In the context of ordinals, we identity 0 with $\emptyset$, n with $\{0, ..., n-1\}$ and ω with $\mathbb{N}$, ordered by the usual order. Moreover, given α and α' ordinals, we define $\alpha + \alpha'$ as the order type of $(\{0\} \times \alpha) \cup (\{1\} \times \alpha')$ ordered by $\leq_{lex}$. In the same way, $\alpha * \alpha'$ is defined as the order type of $\alpha' \times \alpha$ ordered by $\leq_{lex}$. Note that these operations are not commutative: we have $1 + \omega = \omega \neq \omega + 1$. This definition of $+$ and $*$ coincides 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 the generalized lexicographic ordering $\leq_{lex}$ defined by:

$$f <_{lex} g \iff \exists x \in \beta. \left\{ \begin{array}{l} f(x) < g(x) \text{ and,} \\ \forall y < x.\ f(y) = g(y) \end{array} \right.$$

As expected, we have $\alpha^0 = 1$ and $\alpha^1 = \alpha$ for any ordinal α. We define the ordinal ϵ_0, (also called the first fixed point of the exponentiation) by $\epsilon_0 = lub\ \{\alpha_k \mid \alpha_0 = 0 \wedge \forall k \in \mathbb{N}.\ \alpha_k = \omega^{\alpha_{k-1}}\}$. In this chapter, we will only need ordinals less than ϵ_0, that is, those that can be bounded by a finite tower $\omega^{\omega^{\cdot^{\cdot^{\cdot^{\omega}}}}}$. These can be represented by the hierarchy of ordinals in Cantor Normal Form (CNF) that is recursively given by the following rules:

- $C_0 = \{0\}$.
- $C_{n+1} = \{\omega^{\alpha_1} + \cdots + \omega^{\alpha_p} \mid p \in \mathbb{N}, \ \alpha_1, \ldots, \alpha_p \in C_n \text{ and } \alpha_1 \geq \cdots \geq \alpha_p \}$ ordered by:

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

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

3.2 A method for comparing WSTS

3.2.1 A new tool: order reflections

We recall that an order embedding is a mapping $\varphi : X \to Y$ such that for all $x, x' \in X$, $x \leq x' \iff \varphi(x) \leq \varphi(x')$. We define here a weaker version of order embeddings:

Definition 3.2. *Let X and Y be two ordered sets. A mapping $\varphi : X \to Y$ is an* order reflection *(shortly: reflection) if for all $x, x' \in X$:*

$$\varphi(x) \leq \varphi(x') \implies x \leq x'$$

We will write $X \sqsubseteq Y$ if there is an order 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. $X \sqsubset Y \iff X \sqsubseteq Y \wedge Y \not\sqsubseteq X$). It should be noted that every reflection is injective, as $\varphi(x) = \varphi(x') \implies x = x'$ and that any injective mapping to a set equipped with the identity is a reflection. Moreover, the composition of two reflections is a reflection (making $\sqsubseteq_{refl}$ a transitive relation).

If φ is an embedding from X to Y then X is isomorphic to $\varphi(X)$ and hence can be identified to it. The existence of an embedding from a set to another is a stronger requirement than the existence of a reflection. In particular, it can be the case that a set cannot be embedded in another, even if reflections exist, as implied by the following result:

Proposition 3.1. *The following properties hold:*

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

Proof. We first show that $\mathbb{N}^k \sqsubseteq_{refl} \mathbb{N}^\oplus$ for any $k \in \mathbb{N}$.

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

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

Now we define φ by :

$$\varphi(x_1, ...x_k) = \{| \ (2N_k + x_1), (2N_k + x_2), \cdots, (2N_k + x_k) \ |\} \cup ac(o_k(x_1, ...x_k))$$

We claim this is an order reflection.

Indeed, let us take $X = (x_1, ..., x_k)$ and $X' = (x'_1, ..., x'_k)$ and assume that we have $\varphi(X) \leq^{emb} \varphi(X')$. Then, there is a bijective mapping σ:

$$\begin{array}{l} \sigma : \ \ \varphi(X) \to \varphi(X') \\ \qquad\qquad \text{with :} \\ \qquad \varphi(X) = \{| \ 2N_k + x_1, ..., 2N_k + x_k, 2N_k - (o_k(X)+1), o_k(X) \ |\} \\ \qquad \varphi(X') = \{| \ 2N_k + x'_1, ..., 2N_k + x'_k, 2N_k - (o_k(X')+1), o_k(X') \ |\} \\ \qquad \forall x \in \varphi(X).\ x \leq \sigma(x) \end{array}$$

The cardinality of $\varphi(X)$ and $\varphi(X')$ are the same, and the elements of the form $2N_k + x_i$ can only be mapped to elements also of the form $2N_k + x'_j$, so we have

$$\begin{array}{rcl} \sigma(2N_k - (o_k(X)+1)) & = & 2N_k - (o_k(X')+1) \\ \sigma(o_k(X)) & = & o_k(X') \end{array}$$

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 $x_1 \leq x_2 \leq ... \leq x_k$. Let us assume that there exists i such that $x_j \leq x'_j$ for all $j > i$ and $x_i > x'_i$. Because $\sigma(x_i) \neq x'_i$, this means we have $\sigma(x_i) = x'_j$ for some $j \neq i$.

Two cases may occur:

- $\underline{j > i}$: Then by cardinality, we have an element x_p in $\{x_{i+1}, ..., x_k\}$ that is mapped to an element $x'_{p'}$ with $p' \leq i$. Thus, we have $x_i \leq x_p \leq x'_{p'} \leq x'_i$, contradicting our hypothesis that $x'_i < x_i$.

- $\underline{j < i}$: Then, we have $x_i \leq x'_j \leq x'_i$, contradicting again our hypothesis.

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

In order to show that for all k, we have $\mathbb{N}^k \not\sqsubseteq \mathbb{N}^{\oplus}$, it suffices to show that $\mathbb{N}^3 \not\sqsubseteq \mathbb{N}^{\oplus}$. We now show the absence of order embedding from $\mathbb{N}^3$ to $\mathbb{N}^{\oplus}$. To do that, 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 $\mathbb{N}^\oplus$ so $\downarrow\varphi(A_\alpha)$ is a directed downward closed subset. Because φ is an order embedding, and $\downarrow A_\alpha \neq \mathbb{N}^3$, we have $\downarrow\varphi(A_\alpha) \neq \mathbb{N}^\oplus$.

By using the form of the elements in the completion of $\mathbb{N}^\oplus$ (proposition 2.1 on page 31), we get that $lub(\varphi(A_\alpha)) = \{|\ \omega^{k_\alpha}, {k'_\alpha}^\omega\ |\} + B_\alpha$ for some $k_\alpha \in \mathbb{N}$, $k'_\alpha \in \mathbb{N}$ and $B_\alpha \in \mathbb{N}^\oplus$.

We remark that for any three pairs of integers, we can choose one of these pairs that is less or equal than the lub of the two others. This means, that we can find α, β and γ, such that:

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

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

In the same way as before, $\varphi(A_{y,z}[a])$ is an infinite chain of $\mathbb{N}^\oplus$. Let $\varphi(A_{y,z}[a]) = \{|\ \omega^{k_{y,z}[a]}, (k'_{y,z}[a])^\omega\ |\} + B_{y,z}[a]$. Because φ is an order embedding, for any $a \in \mathbb{N}$, this limit is greater or equal than both $\{|\ \omega^{k_y}, {k'_y}^\omega\ |\} + B_y$ and $\{|\ \omega^{k_z}, {k'_z}^\omega\ |\} \cup B_z$, implying that for all $a \in \mathbb{N}$:

$$\begin{array}{rcccl} k_x & \leq & max(k_y, k_z) & \leq & k_{y,z}[a] \\ k'_x & \leq & max(k'_y, k'_z) & \leq & k'_{y,z}[a] \end{array}$$

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

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

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

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

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

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

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

$$\begin{array}{rl} & \{{k'_x}^\omega\} \cup B_x \leq \{\omega^{k_{y,z}[a_0]-k_x}, k'_{y,z}[a_0]^\omega\} \cup B_{y,z}[a_0] \\ \Rightarrow & \{\omega^{k_x}, {k'_x}^\omega\} \cup B_x \leq \{\omega^{k_{y,z}[a_0]}, k'_{y,z}[a_0]^\omega\} \cup B_{y,z}[a_0] \end{array}$$

This means that for each $M \in A_x$, we can find $M' \in A_{y,z}[a_0]$ such that $\varphi(M) \leq \varphi(M')$. But this would mean if φ was an order reflection that $A_x \in \downarrow A_{y,z}[a_0]$. As this is not the case, we got our contradition.

To conclude the demonstration, it only remains to show that $\mathbb{N}^2 \sqsubseteq \mathbb{N}^\oplus$. This is done by noticing that the following function is an embedding:

$$\varphi : \begin{array}{rcl} \mathbb{N}^2 & \to & \mathbb{N}^\oplus \\ (a,b) & \to & \{\mid a+2, 1^b \mid\} \end{array}$$

□

3.2.2 Expressiveness of WSTS and order reflections

We will now show that reflections are appropriate for the comparison of WSTS. In particular, the existence of a reflection implies the relation between the corresponding classes of WSTS. We write $WSTS_X$ the class of WSTS with state space X, and we get:

Theorem 4. *Let X and Y be two well-ordered sets. We have:*

$$X \sqsubseteq_{refl} Y \implies WSTS_X \preceq WSTS_Y$$

Proof. This is shown by taking a WSTS of state space X, looking at its lossy closure through the order reflection, and realizing this is another WSTS which recognizes the same language.

Formally, let $L = L_c(\mathcal{S}, x_0, x_f)$ for some WSTS $\mathcal{S} = \langle X, A, \longrightarrow \rangle$. Because coverability languages are preserved by lossy closure (proposition 1.8), 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 labelled transition system $\mathcal{S}_\varphi = \langle \varphi(X), A, \longrightarrow_\varphi \rangle$ where $\longrightarrow_\varphi$ is defined by:

$$\varphi(x) \xrightarrow{u}_\varphi \varphi(y) \iff x \xrightarrow{u} y$$

We show that $\mathcal{S}_\varphi \in WSTS_Y$. Indeed, if we take $\varphi(x_1)$, $\varphi(x_1')$ and $\varphi(x_2)$ such that $\varphi(x_1) \xrightarrow{u}\!\!\!\!\rightarrow_\varphi \varphi(x_1')$ and $\varphi(x_2) \geq \varphi(x_1)$, then we have by definition of $\mathcal{S}_\varphi$, and because φ is a reflection, that $x_1 \xrightarrow{u}\!\!\!\!\rightarrow x_1'$ and $x_2 \geq x_1$, which means, by well-structure of $\mathcal{S}$, that there exists $x_2' \geq x_1'$ such that $x_2 \xrightarrow{u}\!\!\!\!\rightarrow x_2'$. By the lossiness property of $\mathcal{S}$, we have $x_2' \xrightarrow{\varepsilon}\!\!\!\!\rightarrow x_1'$, and thus $\varphi(x_2') \xrightarrow{\varepsilon}\!\!\!\!\rightarrow_\varphi \varphi(x_1')$, which leads to $\varphi(x_2) \xrightarrow{u}\!\!\!\!\rightarrow_\varphi \varphi(x_1')$. We have shown that $\mathcal{S}_\varphi$ is a WSTS.

Moreover, we clearly have $L_c(\mathcal{S}, x, y) = L_c(\mathcal{S}_\varphi, \varphi(x), \varphi(y))$, which concludes our proof. □

We would like to obtain the converse of the previous result: $X \not\sqsubseteq_{refl} Y \implies WSTS_X \not\preceq WSTS_Y$. First, we only present this result for "simple" state spaces. The case of more complex state spaces will be handled in later sections.

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

Definition 3.3. *Let X be a well-ordered set and A a finite alphabet. A surjective partial function from A^* to X is called a A-representation of X. Given a A-representation η of X, we define $L_\eta = \{u\overline{v} \mid u, v \in dom(\eta)$ and $\eta(v) \leq \eta(u)\}$. A language $L \in (A \cup \overline{A})^*$ is a η-witness (shortly: witness) of X if $L \cap dom(\eta)\overline{dom(\eta)} = L_\eta$.*

In particular, L_η is a witness of X for any A-representation η of X. 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 $\eta(u)$), followed by some v that represents $\eta(v) \leq \eta(u)$. Witness languages are useful in proving strict relations between classes of WSTS:

Theorem 5. *Let L be a witness of X. If $X \not\sqsubseteq_{refl} Y$ then there are no $y, y' \in Y$ and no $\mathcal{S} \in WSTS_Y$ such that $L = L_c(\mathcal{S}, y, y')$.*

Proof. Assume by contradiction that L is a covering language of a WSTS $\mathcal{S}$ whose state space is Y with y and y' as initial and final states, respectively. For each $x \in X$, let us take $u_x \in A^*$ such that $\eta(u_x) = x$. The word $u_x\overline{u_x}$ is recognized by $\mathcal{S}$, hence we can find y_x and y'_x such that

$$y_0 \xrightarrow{u_x}\!\!\!\!\rightarrow y_x \xrightarrow{\overline{u_x}}\!\!\!\!\rightarrow y'_x \geq y_f$$

We define $\varphi(x) = y_x$. Let us see that φ is an order reflection from X to Y, thus reaching a contradiction. Assume that $\varphi(x) \leq \varphi(x')$. Since $\mathcal{S}$ is a WSTS any sequence fireable from $\varphi(x)$ is also fireable from $\varphi(x')$ and the state reached by this subsequence is greater or equal than the one reached from $\varphi(x)$. Hence, the state reached after $u_{x'}\overline{u_x}$ is bigger than the one reached after $u_x\overline{u_x}$, which means that $u_{x'}\overline{u_x} \in L \cap dom(\eta)\overline{dom(\eta)}$, implying $x \leq x'$, so that φ is an order reflection. ∎

The simple state spaces we mentioned before, will be the ones produced by the following grammar (Q and A finite sets ordered by equality):

$$\begin{array}{rcl} \Gamma & ::= & Q \\ & | & \mathbb{N} \\ & | & A^* \\ & | & \Gamma \times \Gamma \end{array}$$

As $\mathbb{N}$ is isomorphic to A^* when A is a singleton, any set produced by Γ is isomorphic to a set $Q \times A_1^* \times \cdots \times A_k^*$ where Q and each A_i are finite sets.

Proposition 3.2. *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 $X = Q \times A_1^* \times \cdots \times A_k^*$, ordered by its canonic order (which is the cartesian product of equality on Q and word embedding ordering on A_i^* for all i). Without loss of generality, we will assume that the A_i's are disjoint. We also define $A = \bigcup_{1 \leq i \leq k} A_i$ and we choose arbitrarily a $q_0 \in Q$. Finally, we define $B = \{b_q \mid q \in Q\}$, also assumed disjoint from A.

We define a functionnal labelled WSTS $\mathcal{S} = \langle X, \Sigma, F, \gamma \rangle$ by:

- $\Sigma = A \cup B \cup \overline{A} \cup \overline{B}$
- $F = \{f_\sigma \mid \sigma \in \Sigma\}$

- For $\sigma \in \Sigma$, $\gamma(f_\sigma) = \sigma$.
- For $a \in A$:

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

- For $\overline{a} \in \overline{A}$:

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

- For $b_p \in B$:

$$f_{b_p}(q, u_1, ..., u_k) = (q', u'_1, ..., u'_k) \iff \begin{cases} q = q_0 \\ q' = p \\ u'_i = u_i \end{cases}$$

- For $\overline{b_p} \in \overline{B}$:

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

We define $\eta(x) = (q, u_1, ..., u_k)$ iff $x \in b_q || u_1 || \cdots || u_k$, 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 $u_i, v_i \in A^*$). η is a $(A \cup B)$-representation of X.

We consider $\mathcal{S}^-$ the lossy closure of $\mathcal{S}$ and we define $L = L_c\left(\mathcal{S}^-, (q_0, \varepsilon, ..., \varepsilon), (q_0, \varepsilon, ..., \varepsilon)\right)$ and we have:

$$L \cap dom(\eta)\overline{dom(\eta)} = \{u\overline{v} \mid u, v \in dom(\eta) \text{ and } \eta(v) \leq \eta(u)\}$$

This conclude the demonstration. □

When a WSTS can recognize a witness of its own state space the following holds:

Proposition 3.3. *Let X be a well-ordered set produced by Γ and Y any well-ordered set. Then,*

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

Proof. The direction from left to right is given by Theorem 4. Hence, we have to prove that $X \not\sqsubseteq_{refl} Y \Rightarrow WSTS_X \not\preceq WSTS_Y$. To do that, we take a witness L of X recognized by a WSTS of state space X (proposition 3.2). By theorem 5, this language can not be recognized by a WSTS of state space Y, hence the result. □

3.2.3 Self-witnessing WSTS classes

The reason we were able to build our equivalence between the existence of a reflection from X to Y and $WSTS_X \preceq WSTS_Y$ for any well-ordered set X produced by Γ was proposition 3.2. However, we conjecture that for any state space X that embeds $\mathbb{N}^{\oplus}$, there is no WSTS of state space X that can recognize a witness of X. This prompts us to define a new notion:

Definition 3.4. *Let $\mathbf{X}$ be a class of well-ordered sets 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.*

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 well-ordered sets by $\mathbf{X} \sqsubseteq_{refl} \mathbf{X}'$ if for any $X \in \mathbf{X}$, there exists $X' \in \mathbf{X}'$ such that $X \sqsubseteq_{refl} X'$.

Proposition 3.4. *Let $(\mathbf{X}, \mathbf{S})$ be a self-witnessing WSTS class and $\mathbf{S}'$ a WSTS class using state spaces inside $\mathbf{X}'$. Then:*

$$\mathbf{S} \preceq \mathbf{S}' \implies \mathbf{X} \sqsubseteq_{refl} \mathbf{X}'$$

Moreover, if $\mathbf{S}' = WSTS_{\mathbf{X}'}$:

$$\mathbf{S} \preceq \mathbf{S}' \iff \mathbf{X} \sqsubseteq_{refl} \mathbf{X}'$$

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 $\mathbf{S} \preceq \mathbf{S}'$, there is $\mathcal{S}' \in \mathbf{S}'$ recognizing L. $\mathcal{S}'$ has state space $X' \in \mathbf{X}'$, and by theorem 5, $X \sqsubseteq_{refl} X'$.

For the second implication, for any $X \in \mathbf{X}$, there exists $X' \in \mathbf{X}'$ such that $X \sqsubseteq_{refl} X'$. From theorem 4, we deduce $WSTS_X \preceq WSTS_{X'}$. Hence, $WSTS_{\mathbf{X}} \preceq WSTS_{\mathbf{X}'}$. ☐

We will see in sections 3.3 and 3.4 that many usual classes of WSTS, even those outside the algebra Γ, are self-witnessing.

3.2.4 How to prove the non-existence of reflections?

Because of propositions 3.3 and 3.4, the non existence of reflections will be a powerful tool to prove strict relations between WSTS. We provide here a simple way from order theory. Let us recall that a *linearization* of a partial order $\leq_X$ on X is a linear order $\leq'_X$ on X such that $x \leq_X y \implies x \leq'_X y$. A linearization of a well order is a well total order, hence isomorphic to an ordinal. We extend the definition of order types to non-total well orders:

Definition 3.5. *Let $(X, \leq_X)$ be a well ordered set. The* maximal order type *(shortly: order type) of $(X, \leq_X)$ is:*

$$ot(X, \leq_X) = lub\ \{ot(X, \leq'_X) \mid \leq'_X \text{ linearization of } \leq_X\}$$

The existence of the *lub* comes from ordinal theory. De Jongh and Parikh [20] even show that this *lub* is actually attained. Let $Down(X)$ be the set of downward closed subsets of X. Then, another possible characterization of the maximal order type is the following:

Proposition 3.5.

$$ot(X) + 1 = lub\ \{\alpha \mid \exists f : \alpha \to Down(X).\ f\ \textit{strictly increasing}\}$$

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

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

$$\begin{array}{rcll} f(\beta) & = & \{x \in X \mid x <' \varphi(\beta)\} & \text{for } \beta < ot(X) \\ f(ot(X)) & = & X & \end{array}$$

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 then prove that $ot(X) + 1 \geq lub\ \{\alpha \mid \exists f : \alpha \to Down(X).\ f$ strictly increasing$\}$

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

$$x \leq_f y \text{ iff } \forall \beta < \alpha.\ y \in f(\beta) \implies x \in f(\beta)$$

$\leq_f$ is clearly reflexive and transitive. Let $\leq_{tie}$ be a linearization of $\leq_X$. We define the order $\leq'_f$ by:

$$x \leq'_f y \iff \begin{cases} x \leq_f y \wedge y \not\leq_f x \quad \text{or,} \\ x \leq_f y \wedge y \leq_f x \wedge x \leq_{tie} y \end{cases}$$

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

Let us prove that $\leq'_f$ is a linear order. Pick any x and y. If they are equivalent w.r.t. $\leq_f$, we get the result by linearity of $\leq_{tie}$. 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. Thus $x \in f(\beta')$. Since β' is arbitrary, this shows that $x \leq'_f y$.

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

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

$$\varphi(\beta) = \min_{\leq'_f}\ \{x \in X' \mid x \notin f(\beta)\}$$

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 increasing.

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 $\varphi(\beta) \leq'_f y$. 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 $(X', \leq'_f)$ which means $\alpha \leq ot(X') = ot(X)+1$. $\square$

We also show a simple lemma, that states order reflections preserve strict inclusion of downward closed sets:

Lemma 3.6. *Let X and Y be two well-ordered sets and φ a reflection from X to Y. Let $D \subsetneq X$ with $D = \downarrow D$. Then $\downarrow\varphi(D) \subsetneq Y$*

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

This leads us to the proposition that we use to separate many classes of WSTS:

Proposition 3.7. *[63] Let X and Y be two well-ordered sets. We have:*

$$X \sqsubseteq_{refl} Y \implies ot(X) \leq ot(Y)$$

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 3.6, g is strictly increasing. By the characterization of order type in proposition 3.5, we have $ot(X) \leq ot(Y)$. □

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 3.1) that an ordinal α below ε_0 is uniquely determined by $Cantor(\alpha)$, hence the validity of the following definition.

Definition 3.6. *(Hessenberg 1906, [20]) The natural addition, denoted $\oplus$, and the natural multiplication, denoted $\otimes$, are defined by:*

$$\begin{array}{rcl} Cantor(\alpha \oplus \alpha') & = & Cantor(\alpha) \cup Cantor(\alpha') \\ Cantor(\alpha \otimes \alpha') & = & \{\mid \beta \oplus \beta' \mid \beta \in Cantor(\alpha), \beta' \in Cantor(\alpha') \mid\} \end{array}$$

(Note that $Cantor(\alpha)$ is a multiset and that the previous union is to be understood as multiset union)

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 [20], and Schmidt [59] have shown how to compose order types with the disjoint union, the cartesian product, and the Higman ordering. A more recent and difficult result, by Weiermann [63], provides us with the order type of multisets. These results are summed up here:

Proposition 3.8. *([20], [59], [63])*

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

- $ot(X^{\oplus}) = \omega^{ot(X)}$ *for* $ot(X) < \epsilon_0$

Formulas exist even for $ot(X) \geq \epsilon_0$. We refer the interested reader to [20] and [63] for the complete formulas. With these general results we can obtain many strict relations between well-ordered sets.

Corollary 3.9. *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)	$(\mathbb{N}^k)^{\oplus} \sqsubset_{refl} (\mathbb{N}^{k+1})^{\oplus}$	(5)	$\mathbb{N}^k \sqsubset_{refl} \Sigma^*$ *(for* $\lvert\Sigma\rvert > 1$*)*
(3)	$(\mathbb{N}^k)^* \sqsubset_{refl} (\mathbb{N}^{k+1})^*$		

Proof. The non-strict relations in (1), (2) and (3) are clear, and for (4) this is proposition 3.1. For (5), $\varphi(n_1, \ldots, n_k) = a^{n_1}b \ldots ba^{n_k}$ is a reflection. Strictness follows from proposition 3.7 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 $ot(\Sigma^*) = \omega^{\omega^{|\Sigma|-1}}$. □

3.3 Vector Addition Systems and Lossy Channel Systems

The state spaces described by Γ and used in proposition 3.3 are exactly those of Petri Nets and Lossy Channel Systems. We will look more closely at these systems to see the implication of this proposition regarding their expressiveness.

3.3.1 Vector Addition Systems

We consider first *Vector Addition Systems with States* (definition 1.7). We recall that given a VASS of dimension k $\langle Q, A, \delta, tr\rangle$ and a relabelling $\gamma : A \to \Sigma^*$, we get a labelled WSTS $\langle Q \times \mathbb{N}^d, \overline{A}, \Sigma, \overline{\gamma}\rangle$ where:

- Functions of $\overline{A}$ are given by, if $tr(a) = (q, q')$:

$$\begin{aligned} dom(\overline{a}) &= \{q\} \times \{x \in \mathbb{N}^d \mid x + \delta(a) \geq 0\} \\ \overline{a}(q, x) &= (q', x + \delta(a)) \end{aligned}$$

- $\overline{\gamma}$ is given by:

$$\overline{\gamma}(\overline{a}) = \gamma(a)$$

Let us denote by $VASS_d$ the class of the transition systems obtained from VASS of dimension d. Notice that the state space of any VASS with dimension d is in $\mathbf{X}_d = \{Q \times \mathbb{N}^d \mid Q$ finite $\}$. Then we have the following:

Theorem 6. *For any* $d > 0$, $VASS_d \not\preceq WSTS_{\mathbf{X}_{d-1}}$.

Proof. We remark that the WSTS defined in the proof of proposition 3.2 is the transition system of a VASS when $X = Q \times \mathbb{N}^d$. This means that $VASS_d$ is self-witnessing, and therefore so is $WSTS_{\mathbf{X}_d}$. Since $\mathbb{N}^d \not\sqsubseteq_{refl} Q \times \mathbb{N}^{d-1}$ for all finite Q (indeed, $ot(\mathbb{N}^d) = \omega^d > \omega^{d-1} * |Q| = ot(Q \times \mathbb{N}^{d-1})$), we have $\mathbf{X}_d \not\sqsubseteq_{refl} \mathbf{X}_{d-1}$ and by proposition 3.4 we conclude. □

3.3.2 Lossy Channel Systems

Given M an alphabet we define $Op(M) = \{read(a) \mid a \in M\} \cup \{write(a) \mid a \in M\} \cup \{nop\}$. For $a \in M, u \in M^*$, this defines functions $\overline{op}$ from M^* to M^* by $\overline{read(a)}(au) = u$, $\overline{write(a)}(u) = ua$ and $\overline{nop}(u) = u$.

Definition 3.7. *A* Channel System *(shortly: CS) with k channels is a tuple (Q, M, A, δ, tr) where:*

- *Q is a finite (and non-empty) set of control states,*
- *M is a finite set of messages,*
- *A is a finite set of transitions,*
- *$\delta : A \to Op(M)^k$ is a mapping providing the effect of a transition on channels,*
- *$tr : A \to Q \times Q$ is a mapping providing the effect of a transition on the control state.*

To a Channel System and a labelling $\gamma : A \to \Sigma^*$, we associate a functional LTS $\langle Q \times M^*, \Sigma, \overline{A}, \overline{\gamma}\rangle$ where:

- The functions $\overline{a}$ are defined, if $tr(a) = (q, q')$ and $\delta(a) = (op_1, \ldots, op_k)$, by:

$$\begin{aligned} dom(\overline{a}) &= \{q\} \times dom(op_1) \times \ldots \times dom(op_k) \\ \overline{a}(q, u_1, \ldots, u_k) &= (q', \overline{op_1}(u_1), \ldots, \overline{op_k}(u_k)) \end{aligned}$$

- $\overline{\gamma}$ is given by:

$$\overline{\gamma}(\overline{a}) = \gamma(a)$$

Taking the lossy closure of this transition system gives a WSTS, called a *Lossy Channel System* (shortly: LCS).

We define $LCS(k, p)$ as the class of Lossy Channel Systems with k channels and p messages. A classic result is that one can encode many channels in one, as long as an additional character (a separator) becomes available for the channel alphabet.

Proposition 3.10. *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 $L_c(\mathcal{S}, x_0, x_f) = L_c(\mathcal{S}', x'_0, x'_f)$.*

Proof. We keep a notion of “active channel" through the control states. We also consider channel numbering to be modulo k, i.e. that channel $k + 1$ is actually channel 1. Let M be the set of messages of $\mathcal{S}$ and $\#$ be a channel symbol with $\# \notin M$. 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 $\mathcal{S}$, $1 \leq i \leq k$ is the current active channel and u_j is the content of the simulated j-th channel. Reading a character in the i-th channel requires it to be the active channel. Writing a character in the i-th requires the the $i + 1$-th channel to be active.

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 precise this result by adding strict inclusions:

Theorem 7. $LCS(k,p) \prec LCS(k+1,p) \prec LCS(1,p+1)$

Proof. $LCS(k,p) \preceq LCS(k+1,p)$ clearly holds. The proof that $LCS(k+1,p) \preceq LCS(1,p+1)$ is based on the well-known fact that one can simulate the $k+1$ channels by inserting a new symbol k times as delimiters. We provide here a quick proof of this statement:

For the strictness, we remark again that the WSTS introduced in the proof of proposition 3.2 is actually a LCS, that is, given a state space $X = Q \times (\Sigma_p^*)^k$, 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, $ot(Q \times (\Sigma_p^*)^k) = \omega^{\omega^{p-1}*k} * |Q|$. This implies that $(\Sigma_p^*)^{k+1} \not\sqsubseteq_{refl} Q \times (\Sigma_p^*)^k$ and $\Sigma_{p+1}^* \not\sqsubseteq_{refl} Q \times (\Sigma_p^*)^k$ for all Q. To conclude we only need to apply proposition 3.4. □

Moreover, in [5] (Theorem 1) the authors prove that $VASS \prec LCS$. We can easily get back this result:

Proposition 3.11.

$$LCS(1,2) \not\preceq VASS$$

Proof. As in the previous result, we remark that $LCS(1,2)$ and $VASS$ are self-witnessing. Thus, we only need to apply proposition 3.4, considering that for any $d > 0$, $M_2^* \not\sqsubseteq_{refl} \mathbb{N}^d$ (corollary 3.9). □

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

3.4 Petri Net extensions with data

Many extensions of Petri Nets with data have been defined in the literature to gain expressive power for better modeling capabilities. Data Nets [45] 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. It is known since [4, **?**] that LCS are strictly less expressive than Petri Data Nets ([4] compares LCS and a model called constrained multiset rewriting system and [**?**] shows that Petri Data Nets are equivalent to these rewriting systems).

A similar model, in which tokens can only be compared with equality, is that of ν-Petri Nets [57]. The relative expressive power of Data Nets and ν-Petri Nets has been an open problem since [58]. In this section we prove that ν-Petri Nets are strictly less expressive than Data Nets. To do so, we will work with the subclass of Data Nets without whole place operations, called *Petri Data Nets*, since Abdulla *et al.* showed that Petri Data Nets were as expressive as Data Nets [5].

3.4.1 Definition of ν-Petri Nets and Petri Data Nets

We use here Petri Net formalism to explain informally their semantics, as the intuitions behind the definitions are easier this way.

Petri Data Nets A *Petri Data Net* (shortly: PDN) is a Petri net where each token carries an *identity* from a linearly ordered and dense domain $\mathbb{V}$. If P is the set of places where the tokens can be, a marking m of a *PDN* can be seen, as a multiset of pairs in $\mathbb{V} \times P$, or as a mapping from $\mathbb{V}$ to $P^{\oplus}$. However, two key features of Petri Data Nets will guide our choice for another representation of states:

1. A marking m has only finitely many tokens. Thus, denoting $v_1 < \cdots < v_m$ the identity of tokens present in m and gathering all tokens carrying the same identity v_i, one obtains a (non-null) place vector x_i in $\mathbb{N}^{card(P)}$. Therefore, m can be written $(v_1, x_1) \cdots (v_m, x_m)$ where x_i are vectors of dimension $card(P)$ different from 0.
2. The concrete identities v_i are irrelevant, and only their relative *order* is useful with respect to the semantics of the net. Thus, m can be safely abstracted as the sequence $x_1 \cdots x_m$ in $(\mathbb{N}^d \setminus 0)^*$ where d is the number of places.

Every transition a of a PDN specifies a sequence of n ordered potential identities and for any such identity specifies the tokens $F(a)$ to be consumed and $H(a)$ to be produced. Thus, $F(a)$ and $H(a)$ are two sequences of n (possibly null) place vectors.

Definition 3.8 (Petri Data Nets)**.** *A* Petri Data Net *of dimension* d *is a tuple* $\mathcal{N} = \langle A, F, H\rangle$ *where:*

- *A is a finite set of transitions,*
- *$F : A \to (\mathbb{N}^d)^*$ is a mapping denoting how many tokens are consumed.*
- *$H : A \to (\mathbb{N}^d)^*$ is a mapping denoting how many tokens are produced.*

From a marking $m \in (\mathbb{N}^k \setminus 0)^*$. In order to fire a transition a with $|F(a)| = n$, one nondeterministically selects n identities, consumes some of their tokens as indicated by $F(t)$, and produces new tokens with the identities specified by $H(t)$. However, some of these n identities might not be present in s, and we should introduce null vectors wherever necessary: $m' \in (\mathbb{N}^k)^*$ is a 0-*extension* of $m \in (\mathbb{N}^k \setminus 0)^*$ (equivalently: m is a 0-*contraction* of m') if m can be obtained from m' by erasing all null vectors.

Once such an 0-extension m' is built, one selects in it a subword of n vectors $x_1, \ldots, x_n$ such that every of these vectors contains enough tokens for the transition to be fired, i.e. for all $i \in \{1, \ldots, n\}$, $x_i \geq F(t)(i)$. In this case, for each $i \in \{1, \ldots, n\}$, $F(t)(i)$ is substracted

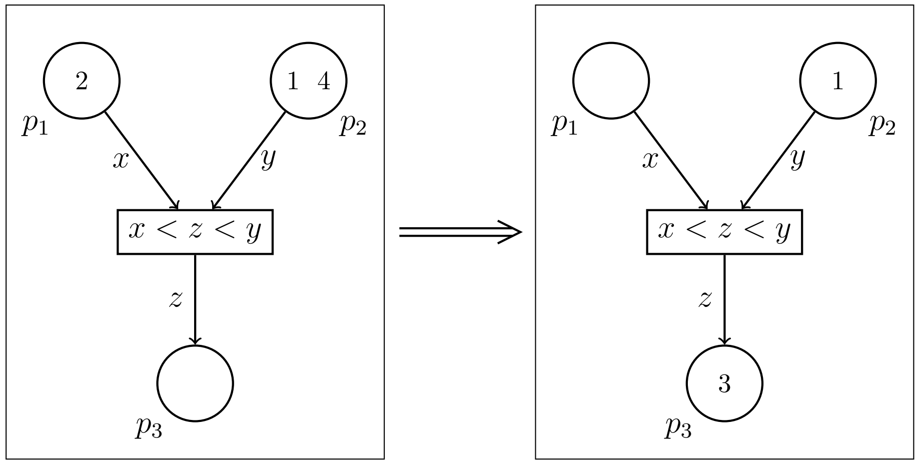


Figure 3.1: Firing of a Petri data net transition

and $H(t)(i)$ is added, yielding a new marking m''. This m'' may contain null vectors when all tokens with some identity have been consumed, so we take the 0-contraction of m'', giving our final marking m'''. Formally, the semantics of a PDN (in VAS-like notations) are given by:

Definition 3.9 (Transition system of a PDN)**.** *Let $\mathcal{N}$ be a PDN of dimension d. Its associated transition system $\mathcal{S}_{\mathcal{N}} = \langle X, A, \longrightarrow \rangle$ is defined by:*

- $X = (\mathbb{N}^d \setminus 0)^*$
- $m \xrightarrow{a} m'$ *if there exists* $u_0 x_1 u_1 \cdots u_{n-1} x_n u_n$ *a* 0*-extension of* m *with* $u_i \in (\mathbb{N}^k)^*$ *and* $x_i \in \mathbb{N}^k$ *such that:*
  1. $\forall i \in \{1, \ldots, n\}$, $x_i \geq F(t)(i)$,
  2. m' *is the* 0*-contraction of* $u_0 y_1 u_1 \cdots u_{n-1} y_n u_n$ *where* $y_i = x_i - F(t)(i) + H(t)(i)$.

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 by variables (whose number if $|F(t)|$) that must be instantiated in a way that respects a constraint labelling the transition. For concision and readability, it is convenient to allow orderings of the variables that are not total. Such a transition would then represent several transitions, each of these corresponding to a possible linearization of the constraint.

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 $x \leq y$, can be simulated by two transitions one assuming $x < y$ and the other one with y substituted by x.

Using these graphical conventions, figure 3.1 depicts a PDN with a single transition a given by:

$$\begin{array}{rcl} F(a) & = & (1,0,0)(0,0,0)(0,1,0) \\ H(a) & = & (0,0,0)(0,0,1)(0,0,0) \end{array}$$

ν-Petri Nets ν-Petri Nets can be seen as a restriction of Petri Data Nets where the domain of identities $\mathbb{V}$ still infinite is now unordered. If we would like to define this model exactly as for Data Nets, it is sensible to add a construction that ensures that a newly created token has a value distinct from any already present one (this was not necessary in Petri Data Nets, as one could maintain a token storing the largest value and then any token created above this value would be distinct from any existing value). Because of this, we introduce a countable set Var of variables including a subset of special variables $\Upsilon \subset Var$ with $card(\Upsilon) = card(Var \setminus \Upsilon) = \omega$. The role of Υ is to select values that are not present in the current marking.

Definition 3.10 (ν-Petri Net)**.** *A* ν-Petri Net *(shortly: ν-PN) of dimension d is a tuple $\mathcal{N} = \langle A, F, H \rangle$, where:*

- *A is a finite set of transitions,*
- *$F : A \to Var \setminus \Upsilon \to \mathbb{N}^d$ is a mapping denoting how many tokens are consumed such that for all a, $dom(F(a))$ is finite.*
- *$H : A \to Var \to \mathbb{N}^d$ is a mapping denoting how many tokens are produced such that for all a, $dom(H(a))$ is finite.*

To represent the markings of a ν-Petri Net, we can use the same reasoning as for Petri Data Nets, as only the equalities/inequalities between identities matter. This means that a marking of a ν-Petri Net will be an element of $(\mathbb{N}^d \setminus 0)^{\oplus}$. Now, to fire a transition a from a marking m, we will first take a 0-extension m' of m (adding as many 0 into m as we want), then each variable of $dom(F(a)) \cup dom(H(a))$ is mapped to an element of m', with variables of Υ being mapped to distinct 0 elements. For each $\alpha \in dom(F(a))$, $F(a)(\alpha)$ is subtracted to the elements to which α is mapped, then for each $\alpha \in dom(H(a))$, $H(a)(\alpha)$ is added to the element to which α is mapped. The 0-contraction of the resulting marking is the marking obtained after the transition. Formally, we have:

Definition 3.11 (Transition System of a ν-Petri Net)**.** *Let $\mathcal{N}$ be a ν-Petri net. Its associated transition system $\langle X, A, \longrightarrow \rangle$ is defined by:*

- *$X = (\mathbb{N}^d \setminus 0)^{\oplus}$*
- *$m \xrightarrow{a} m'$ if we have:*

$$\begin{array}{rll} m_0 \cup \{\mid x_1, \ldots, x_p \mid\} \cup \{\mid 0^q \mid\} & \textit{is a 0-extension of} & m \\ m_0 \cup \{\mid x'_1, \ldots, x'_p \mid\} \cup \{\mid y'_1, \ldots, y'_q \mid\} & \textit{is a 0-extension of} & m' \end{array}$$

such that there exists a mapping φ from $(dom(F(a)) \cup dom(H(a))) \setminus \Upsilon$ to $\{1, \ldots, p\}$ and a mapping ψ from $dom(H(a)) \cap \Upsilon$ to $\{1, \ldots, q\}$ with:

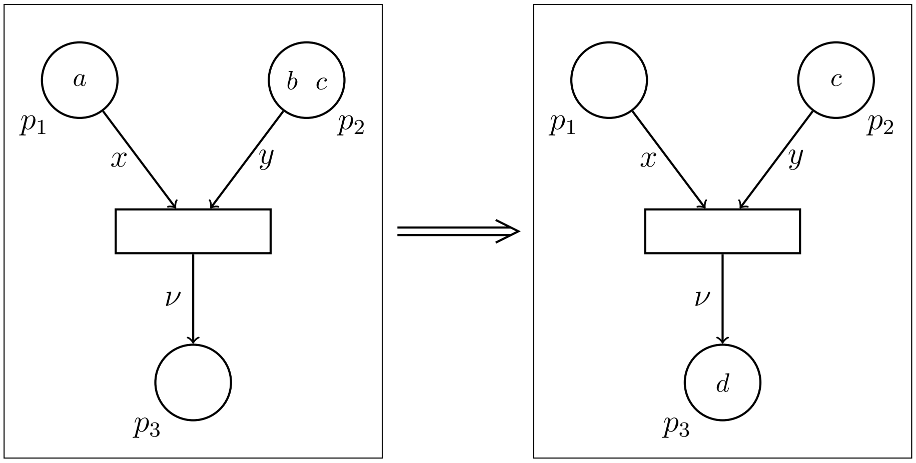


Figure 3.2: Firing of a ν-Petri Net transition

$$
\begin{aligned}
x_i &\geq \sum_{\varphi(\alpha)=i} F(a)(\alpha) \\
x'_i &= x_i - \sum_{\varphi(\alpha)=i} F(a)(\alpha) + \sum_{\varphi(\alpha)=i} H(a)(\alpha) \\
y'_i &= \sum_{\varphi(\alpha)=i} H(a)(\alpha)
\end{aligned}
$$

The graphical representation of a ν-Petri net is similar to that of a Petri Data Net except that if each arc is labelled by a variable, there is no other constraint than the implicit constraint due to arcs being labelled by the same variable. In such a graphical representation, we use $\nu, \nu_1, \ldots, \nu_k$ for variables of Υ.

Figure 3.2 illustrates the firing of a transition in such nets. Observe that the token created by the transition cannot belong to $\{a, b, c\}$.

Classes of Nets Given $\mathcal{N}$ a ν-Petri Net or Petri Data Net with an initial marking, a place i 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 i 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 Petri Data Net (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 the transition system, possibly relabelled, of Petri Data Nets with k unbounded places by PDN_k and their state space by $\mathbf{X}^*_k = \{(Q \times \mathbb{N}^k)^* \mid Q \text{ finite}\}$. Similarly, we denote the class of the transition system, possibly relabelled, of ν-PN with k unbounded places by ν-PN_k and their state space by $\mathbf{X}^{\oplus}_k = \{(Q \times \mathbb{N}^k)^{\oplus} \mid Q \text{ finite}\}$. Moreover, we take $\mathbf{X}^* = \{(\mathbb{N}^k)^* \mid k > 0\}$ and $\mathbf{X}^{\oplus} = \{(\mathbb{N}^k)^{\oplus} \mid k > 0\}$.

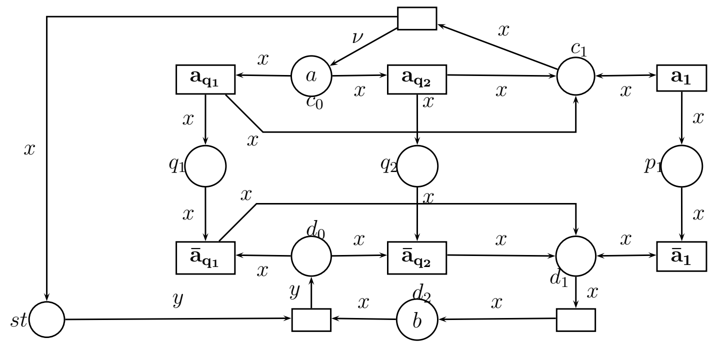


Figure 3.3: ν-Petri Net recognizing a witness of $(Q \times \mathbb{N})^{\oplus}$ with $|Q| = 2$

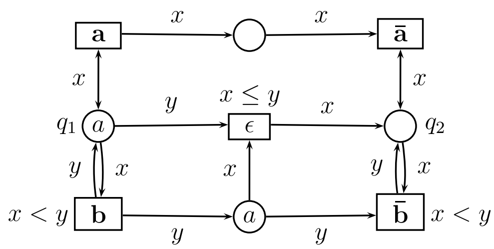


Figure 3.4: Petri Data Net recognizing a witness of $\mathbb{N}^*$

3.4.2 Self-Witnesses and Consequences

Proposition 3.12. *For every $d \geq 0$, ν-PN_d and PDN_d are self-witnessing.*

Proof. We start with ν-PN_k. Let $(Q \times \mathbb{N}^d)^{\oplus} \in \mathbf{X}_d^{\oplus}$. We consider an alphabet $\Sigma = \{\sigma_q \mid q \in Q\} \cup \{\sigma_0, \ldots, \sigma_{d-1}\}$ and we define $\eta : \Sigma^* \to (Q \times \mathbb{N}^k)^{\oplus}$ by:

$$\eta(a_{q_0} a_0^{n_{0,0}} \ldots a_{d-1}^{n_{0,d-1}} \ldots a_{q_k} a_0^{n^{k,0}} \ldots a_{d-1}^{n_{k,d-1}})$$
$$=$$
$$\{\mid (q_0, n_{0,0}, \ldots, n_{0,d-1}), \ldots, (q_k, n_{k,0}, \ldots, n_{k,d-1}) \mid\}$$

Let us build $ts(\mathcal{N}) \in \nu$-PN_d and $x, y \in (Q \times \mathbb{N}^d)^{\oplus}$ such that $L_c(ts(\mathcal{N}), x, y) \cap dom(\eta)\overline{dom(\eta)} = L_\eta$. Assume $Q = \{q_0, \ldots, q_r\}$. Figure 3.3 shows the case with $d = 1$ and $r = 2$.

We take d unbounded places $p_1, \ldots, p_d$ (hence $\mathcal{N} \in \nu$-PN_d). Moreover, we take $q_1, \ldots, q_r$ as bounded places, a bounded place st that stores all the identities that have been used (once each identity, hence bounded), and bounded places $c_0, c_1, \ldots, c_d$ containing one identity in mutual exclusion. When the identity is in c_0 it is non-deterministically copied in some q (action labelled by a_q), and moved to c_1. For every $1 \leq i \leq d$, when the identity is in c_i it can be copied arbitrarily often to p_i (action labelled by a_i). At any time, this identity can be transferred to c_{i+1} when $i < d$ or to st for $i = d$ (action labelled by ε). In the last case a fresh identity is put in c_0 (thanks to $\nu \in \Upsilon$).

The second phase is analogous, with bounded places $c'_0, c'_1, ..., c'_{k+1}$, marked in mutual exclusion with identities taken from st. At any point, the identity in c'_{k+1} can be removed, and one identity moved from st to c'_0 (action labelled by ϵ). That identity must appear in some q. Thus, for each q we have a transition that removes the identity from c'_0 and q and puts it in c'_1 (action labelled by $\bar{a}_q$). For each $1 \le i \le d$, the identity in c'_i can be removed zero or more times from p_i (action labelled by $\bar{a}_i$). At any point, the identity is transferred from c'_i to c'_{i+1} (actions labelled by ε).

The initial and final marking x is with an identity in c_0 and another identity in d_{k+1} (and empty elsewhere). One can check that $L_c(ts(\mathcal{N}), x, x) \cap dom(\eta)\overline{dom(\eta)} = L_\eta$, so we conclude.

The case of PDN_d is analogous to that of $\nu\text{-}PN_d$. Let $(Q \times \mathbb{N}^d)^* \in \mathbf{X}^*_d$. We define $\Sigma = \{\sigma_q \mid q \in Q\} \cup \{\sigma_0, \ldots, \sigma_{d-1}\}$ and $\eta : \Sigma^* \to (Q \times \mathbb{N}^d)^*$ by:

$$\eta(a_{q_0} a_0^{n_{0,0}} \ldots a_{d-1}^{n_{0,d-1}} \ldots a_{q_k} a_0^{n^{k,0}} \ldots a_{d-1}^{n_{k,d-1}}) = (q_0, n_{0,0}, \ldots, n_{0,d-1}), \ldots, (q_k, n_{k,0}, \ldots, n_{k,d-1})$$

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

Figure 3.4 shows a *PDN*recognizing a witness of $\mathbb{N}^*$. Notice that since $\nu\text{-}PN_k$ and PDN_k are self-witnessing for every $k \ge 0$, so are ν-*PN* and *PDN*.

Proposition 3.13. *$\mathbf{X}^*_1 \not\sqsubseteq_{refl} \mathbf{X}^\oplus$, $\mathbf{X}^\oplus_{k+1} \not\sqsubseteq_{refl} \mathbf{X}^\oplus_k$ 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 $ot(\mathbb{N}^*) = \omega^{\omega^\omega} \not\le \omega^{\omega^k} = ot((\mathbb{N}^k)^\oplus)$, 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 3.14. *ν-PN $\prec$ PDN. Moreover, $PDN_1 \not\preceq \nu$-PN.*

Proof. ν-*PN* $\preceq$ *PDN* is from [58]. $PDN_1 \not\preceq \nu$-*PN* is a consequence of proposition 3.4, 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 3.15. *For any $k \ge 0$, $\nu\text{-}PN_k \prec \nu\text{-}PN_{k+1}$ and $PDN_k \prec PDN_{k+1}$.*

Proof. Clearly $\nu\text{-}PN_k \preceq \nu\text{-}PN_{k+1}$ and $PDN_k \preceq PDN_{k+1}$ for any $k \ge 0$. For the converses, again we can apply proposition 3.4, 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 *VASS* $\prec \nu$-*PN* proved in [58] in a very straightforward way.

Proposition 3.16. *$\nu\text{-}PN_1 \not\preceq$ VASS*

Proof. Both *VASS* and ν-PN_1 are self-witnessing, and $\mathbf{X}_1^{\oplus} \not\sqsubseteq_{refl} \{\mathbb{N}^k \mid k > 0\}$ because $\mathbb{N}^{\oplus} \not\sqsubseteq_{refl} \mathbb{N}^k$ for all k (indeed, by proposition 3.8 $ot(\mathbb{N}^{\oplus}) = \omega^{\omega} \not\leq \omega^k = ot(\mathbb{N}^k)$). By proposition 3.4 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-$PN_0 \simeq VASS$.

3.5 Summary of results

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 below 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.

Quantitative results. (All results are new.)

For every $k \in \mathbb{N}$, $VASS_k \prec VASS_{k+1} \not\preceq VASS_k$

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 result is ν-$PN \prec DN$)

$VASS \prec LCS \prec DN \simeq PDN$

$VASS \prec \nu\text{-}PN \prec DN \simeq PDN$

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.

As all the models that we have studied in this paper use a state space whose order type is bounded by ϵ_0, it is tempting to look at WSTS that would use a greater state space. It is known that the Kruskal ordering has an order type greater than ϵ_0 [59], even for unlabelled binary trees. However, studies of WSTS based on trees have been quite scarce [43]. We believe some interesting problems might lie in this direction.

Part II

Extended Vector Addition Systems

Chapter 4

Vector Addition Systems with 2 resets

This chapter is unpublished material.

Vector Addition Systems with resets (equivalently Reset Nets) are a natural extension of Vector Addition Systems where one allows operations to set counters to zero. This can be used as a direct modeling tool, or seen as an underapproximation of a zero-test. Moreover, one can show that Affine Nets [31], one of the largest Petri Net extension obtained without changing the state space can be simulated by Reset Nets in a sensible way.

From a verification point of view, Vector Addition Systems with resets seem to be on the frontier of decidability for most problems considered. Indeed, the decidability and undecidability of these problems is precisely known since the works of Dufourd *et al.* in the late nineties [23, 24]. Mayr [50] later published an overview of the decidability status of problems in lossy counter machines, making some results a bit more precise. Here is a summary of previous results, with the addition of some results that will be later shown in chapter 5:

	no reset	1 reset	2 resets	three resets
REACHABILITY	decidable [49]	decidable (red. from [55])	undecidable [23]	undecidable [23]
COVERABILITY	decidable (WSTS)	decidable (WSTS)	decidable (WSTS)	decidable (WSTS)
BOUNDEDNESS	decidable (strict WSTS)	decidable [24]	decidable [24]	undecidable [23]
PLACE-BOUNDEDNESS	decidable [41]	decidable (red. from 5.3)	?	undecidable [23]
REP. COVERABILITY	decidable [25]	decidable (red. from 5.4.1)	undecidable [50]	undecidable [23]

One can see that the number of resets is of great importance when looking at decidability problems. Actually, more than the number of resets, the limiting factor is the number *of counters* that can be reset, as many reset transitions can be collapsed into one if they all reset the same counter.

In this chapter, we fill this gap by showing the decidability of PLACE-BOUNDEDNESS for Vector Addition Systems with 2 resets.

4.1 Definition

We introduce formally Vector Addition Systems with resets:

Definition 4.1. *A Vector Addition System with resets (shortly : VAS$_{rr}$) of dimension d is a tuple $\langle A, \delta, \mathcal{R}\rangle$ where:*

- *A is a finite set of transition labels,*
- *δ is a mapping from A to $\mathbb{Z}^d$ and,*
- *$\mathcal{R}$ is a mapping from A to $\{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$.*

A VAS with resets is a VAS with k resets if there exists a subset $R \subseteq \{0, \ldots, d-1\}$ such that for all $a \in A$, $\mathcal{R}(a) \subseteq R$.

$\delta(a)$ is the vector added to a state by the transition a, while $\mathcal{R}(a)$ defines which counters are reset by a. The semantics of this system are defined formally by the partial functions $\overline{a} : \mathbb{N}_\omega^d \mapsto \mathbb{N}_\omega^d$ with:

$$\begin{array}{rcl} dom(\overline{a}) & = & \{x \in \mathbb{N}_\omega^d \mid x + \delta(a) \geq 0\} \\ \overline{a}(x)(i) & = & \begin{cases} x(i) + \delta(a)(i) & \text{if } i \notin \mathcal{R}(a) \\ 0 & \text{otherwise} \end{cases} \end{array}$$

We associate to $\mathcal{V}$ the complete WSTS $ts(\mathcal{V}) = \langle \mathbb{N}_\omega^d, \overline{A}\rangle$. We write $\textsc{vasrr} = \{ts(\mathcal{V}) \mid \mathcal{V}$ is a VAS with 2 resets$\}$. Properties of the normal transition system (defined on $\mathbb{N}^d$) can be lifted to its complete transition system, as explained by the more general proposition 4.3.

We can define the extension with states, as we have defined VASS from VAS in definition 1.7 on page 1.7. We won't give explicitly this definition, which should be straightforward.

4.2 Regular loops

Given $\mathcal{S} = \langle \mathbb{N}_\omega^d, \overline{A}\rangle \in \textsc{vasrr}$, we define $\textsc{djs}(\mathcal{S})$, a set of functions from $\mathbb{N}$ to A^* by $\textsc{djs}(\mathcal{S}) = \{n \to \prod_{0 \leq k \leq n} u_0 v_1^k u_1 \ldots v_p^k u_p \mid u_0, v_1, \ldots, u_p \in A^*\}$.

Definition 4.2. *$\varphi \in \textsc{djs}$ is a* regular loop *on $x \in \mathbb{N}_\omega^d$, if for all $k \in \mathbb{N}$, $\overline{\varphi(k+1)}(x) > \overline{\varphi(k)}(x)$.*

This definition of regular loops is similar to the definition of "regular path schemes" in [24]. Formally, we have:

Proposition 4.1. *Let (w, f, x_0) be a regular path scheme (as defined in [24]). There exists a regular loop φ (as defined in definition 4.2) such that:*

$$\downarrow\{\overline{w(0)w(1)\ldots w(k)}(x_0) \mid k \in \mathbb{N}\} = \downarrow\{\overline{\varphi(0)\varphi(1)\ldots\varphi(k)}(x_0) \mid k \in \mathbb{N}\}$$

Proof. To show that, let us quote the original definition with minor alterations to be consistent with our notations of Vector Addition Systems:

We use p_1, p_2 to denote elements of $\{0,1\}$. We define $p^\sharp = 3-p$. Markings are presented as (unordered) tuples $(r, p_1 \leftarrow x, p_2 \leftarrow y)$ where r ranges over submarkings on nonresetable places, while x, y are values of the resetable places. The (finite!) range of r is denoted by fcs; it can be viewed as a finite control states *set. (...)*
Given a pair $(r, p \leftarrow b_1)$, $(r', p' \leftarrow b_2)$, where $b_1, b_2 \in \mathbb{N}$ (and $p = p'$ is allowed), by a path scheme, *of* order n, *we mean a triple (w, f, x_0), where $w : \mathbb{N} \to A^+$, $f : \mathbb{N} \to \mathbb{N}$, $x_0 \in \mathbb{N}$, such that $\forall x \geq x_0 : (r, p \leftarrow b_1, p^\sharp \leftarrow x) \xrightarrow{w(x)}_n (r', p' \leftarrow b_2, p'^\sharp \leftarrow f(x))$ ($\longrightarrow_n$ is the transition relation restrained to the runs where at most n reset transitions are used). The path scheme has the maximum property if for every $x \geq x_0$, there is no $y > f(x)$ s.t. $(r, p \leftarrow b_1, p^\sharp \leftarrow x) \xrightarrow{*}_n (r', p' \leftarrow b_2, p'^\sharp \leftarrow y)$.*
A function $g : \mathbb{N} \to \mathbb{N}$ is (i,k)-regular, where $0 \leq i < d$, $i, k \in \mathbb{N}$ iff there are rational constants ρ_1, ρ_2 such that: for every $x \in \mathbb{N}$, $x \bmod k = i$ implies $g(x) = \rho_1 x + \rho_2$ ($\in \mathbb{N}$; note that it imposes $\rho_1 k \in \mathbb{N}$). We refer to ρ_1 (ρ_2) as the first *(*second*) coefficient of g.*
A function $f : \mathbb{N} \to \mathbb{N}$ is k-regular, zhere $k \in \mathbb{N}$, $k > 0$ iff there are functions $f_0, f_1, \ldots, f_{k-1}$ such that f_i is (i,k)-regular ($i = 0, 1, \ldots, k-1$), all f_i's have the same first coefficient ρ and $f(x) = f_i(x)$ for $x \bmod k = i$; ρ is then the coefficient of f. We call f regular if it is k-regular for some k.
A path scheme (w, f, x_0) is regular if f is k-regular for some k, and for each $i \in \{0, 1, 2, \ldots, k-1\}$, we have $m \in \mathbb{N}$, $u_1, v_1, u_2, v_2, \ldots, u_m, v_m, u_{m+1} \in A^*$ and (i,k)-regular functions $g_1, g_2, \ldots, g_m$ such that: for every $x \geq x_0$, $x \bmod k = i$ implies $w(x) = u_1 v_1^{g_1(x)} u_2 v_2^{g_2(x)} \ldots u_m v_m^{g_m(x)} u_{m+1}$.
A regular witness (for the net N) is a reachable marking $(r, p_1 \leftarrow x_0, p_2 \leftarrow 0)$ together with a regular path scheme (w, f, x_0) (of order n for some $n \in \mathbb{N}$) which is related to the pair $(r, p_2 \leftarrow 0)$, $(r, p_2 \leftarrow 0)$ and has the property: $\forall x \geq x_0 : x < f(x)$.

Let us now explain how this corresponds to our regular loop definition. We take a regular path scheme (w, f, x_0). There is $k \in \mathbb{N}$ such that for each $i \in \{0, 1, 2, \ldots, k-1\}$, we have for $n \bmod k = i$, $w(n) = u_{i,1} v_{i,1}^{g_{i,1}(n)} u_2 \ldots v_{i,p_i}^{g_{i,p_i}(n)} u_{i,p_i+1}$ with $g_{i,1}, \ldots, g_{i,p_i}$ (i,k)-regular functions. First, we note that a (i,k)-function restrained on $\{n \mid n \bmod k = i\}$ is an affine function (with rationals coefficients), so we have that:

$$w(kn)w(kn+1)w(k(n+1)-1) = \prod_{0 \leq i \leq k-1} u_{i,1} {v_{i,1}}^{a_{i,1}n + b_{i,1}} u_{i,2} \ldots {v_{i,p_i}}^{a_{i,p_i}n + b_{i,p_i}} u_{i,p_i+1}$$

Thus, we have a new regular path scheme (w', f', x_0) where $f'(n) = f(kn)$ and $w'(n) = \prod_{0 \leq i \leq k-1} u_{i,1} {v_{i,1}}^{a_{i,1}n + b_{i,1}} u_{i,2} \ldots {v_{i,p_i}}^{a_{i,p_i}n + b_{i,p_i}} u_{i,p_i+1}$. Now, notice that v^{an+b} can be rewritten as $u_1 v_1^n u_2 \ldots v_a^n u_{a+1}$ where $u_i = \varepsilon$ for $i \leq a$, $u_{a+1} = v^b$ and $v_i = v$. Thus, a regular path scheme (w, f, x_0) (according to the definition of [24]) induces a regular loop on x_0 (according to the definition of this chapter). ☐

Thus, we will be able to use the following proposition (updated to use our definition)

Proposition 4.2. *([24], Section 4 / Proposition 1) Let $\mathcal{S} \in$ VASRR, $x_0 \in \mathbb{N}_\omega^d$ and assume that there exists an infinite strictly increasing sequence $(x_n)_{n \in \mathbb{N}}$ with $x_{n+1} \in Reach_{\mathcal{S}}(x_n)$. Then, one of the following statements is true:*

(1) There exists $u \in A^*$ and $y \in Reach_{\mathcal{S}}(x_0)$ such that u is a simple loop on y.

(2) There exists $\varphi \in \textsc{djs}(\mathcal{S})$ and i $y \in Reach_{\mathcal{S}}(x_0)$ such that φ is a regular loop on y.

4.3 Generalized Vector Addition Systems with 2 resets

In this chapter, we will need a more robust class of transition system, that will contain any function f^ω obtained by iterating a sequence f of a VAS_{rr} (see proposition 4.4). For this, we allow transitions to set a counter to any value (including ω), and we separate the precondition of the transitions from the decrementations of the counters:

Definition 4.3. *A generalized Vector Addition System with States and 2 resets of dimension d is a tuple $\mathcal{V} = \langle Q, A, \rho, \delta, \mu, tr \rangle$ where:*

- *Q is a finite set of control states.*
- *A is a finite set of actions.*
- *$\rho : Q \times A \to \mathbb{N}^d$ provides the prerequisites to fire a transition.*
- *$\delta : Q \times A \to \mathbb{Z}^d$ provides the effect of the transition.*
- *$\mu : Q \times A \to (\mathbb{N}_\omega \cup \bot)^2 \times \bot^{d-2}$ indicates which counters are set to a precise value, and,*
- *$tr : Q \times A \to Q$ provides the next control state.*

The semantics of this system is given by $ts(\mathcal{V}) = \langle \mathbb{N}_\omega^d \times Q, \overline{Q \times A} \rangle$ where:

$$\begin{array}{rcl} dom(\overline{(q,a)}) & = & \uparrow\rho(q,a) \times \{q\} \\ \overline{(q,a)}(x)(i) & = & (y, tr(q,a)) \quad \text{where } y \text{ is defined by:} \\ y(i) & = & \begin{cases} x(i) + \delta(q,a)(i) & \text{if } \mu(q,a) = \bot \\ \mu(q,a)(i) & \text{if } \mu(q,a) \neq \bot \end{cases} \end{array}$$

We now show that these generalized VAS_{rr} can be faithfully simulated by normal VAS_{rr} (without even needing to use ω-values). We are using states in the simulating system for convenience.

Proposition 4.3. *Let $\mathcal{V}$ be a generalized Vector Addition System with States and two Resets and $x \in Q \times \mathbb{N}_\omega^d$. One can build $\mathcal{V}'$ a Vector Addition System with States and two Resets, $x' \in Q' \times \mathbb{N}^d$ and φ a continuous function from $Q' \times \mathbb{N}_\omega^d$ to $Q \times \mathbb{N}_\omega^d$ such that:*

$$\begin{array}{rcl} Reach_{ts(\mathcal{V})}(x) & = & \varphi(Reach_{ts(\mathcal{V}')}(x')) \\ y' \xrightarrow{a}_{ts(\mathcal{V}')} z' & \Longrightarrow & \varphi(y') \xrightarrow{a}_{ts\mathcal{V}} \varphi(z') \\ y \xrightarrow{a}_{ts(\mathcal{V})} z & \Longrightarrow & \forall y' \in \varphi^{-1}(y).\ \exists z' \in \varphi^{-1}(z).\ y' \xrightarrow{a}_{ts(\mathcal{V}')} z' \end{array}$$

Proof. Let $\mathcal{V} = \langle Q, A, \rho, \delta, \mu, tr\rangle$ of dimension d. The idea is to encode in the control states what components are equal to ω. Thus, these components can be ignored for transitions prerequisites. Transitions that set a component a specific value will update this control state.

Let $\Omega = \{o : \{0, \dots, d-1\} \to \{0,1\}\}$. We build $\mathcal{V}' = \langle Q \times \Omega, A, \rho', \delta', \mu', tr'\rangle$ by:

$$\begin{array}{rcl} \rho'(q,o,a)(i) & = & \left\{ \begin{array}{ll} \rho(q,a)(i) & o(i)=0 \\ 0 & o(i)=1 \end{array} \right. \\ \delta'(q,o,a)(i) & = & \delta(q,a) \\ \mu'(q,o,a)(i) & = & \mu(q,a)(i) \\ tr'(q,o,a)(i) & = & (tr(q,a), o') \\ & & \text{with: } o'(i) = \left\{ \begin{array}{ll} o(i) & \text{if } \mu(i) = \bot \\ 0 & \text{if } \mu(i) \in \mathbb{N} \\ 1 & \text{if } \mu(i) = \omega \end{array} \right. \end{array}$$

We define $\varphi : Q \times \Omega \times \mathbb{N}_\omega^d \to Q \times \mathbb{N}_\omega^d$ by:

$$\begin{array}{rcl} \varphi(q,o,s') & = & (q,s) \\ & & \text{with: } y(i) = \left\{ \begin{array}{ll} s'(i) & \text{if } o(i)=0 \\ \omega & \text{if } o(i)=1 \end{array} \right. \end{array}$$

Finally, we also define $x' = (q, o, s') \in Q \times \Omega \times \mathbb{N}^d$ from $x = (q, s)$ by:

$$\begin{array}{rcl} o(i) & = & \left\{ \begin{array}{ll} 1 & \text{if } s(i) = \omega \\ 0 & \text{if } s(i) < \omega \end{array} \right. \\ s'(i) & = & \left\{ \begin{array}{ll} s(i) & \text{if } s(i) < \omega \\ 0 & \text{otherwise} \end{array} \right. \end{array}$$

With these definitions, it is straightforward to show that the conditions of the proposition are fulfilled. □

4.4 Accelerations in VASRR

We describe now the construction of an acceleration strategy partly taken from the works of Dufourd, Jančar and Schnoebelen [24] that will allow us to compute the cover for VAS with two resets.

Definition 4.4. *For $\mathcal{S} \in$ VASRR, we define $\overline{\text{DJS}}(\mathcal{S}) = \{\overline{\varphi} \mid \varphi \in \text{DJS}(\mathcal{S})\}$ by $dom(\overline{\varphi}) = \bigcap_{n\in\mathbb{N}} dom(\overline{\varphi(n)})\}$ and for $x \in \mathbb{N}_\omega^d$, $\overline{\varphi}(x) = lub\ \{\overline{\varphi(n)}(x) \mid n \in \mathbb{N}\}$ if φ is a regular loop on x, and $\overline{\varphi}(x) = x$ otherwise.*

We note that u is a sequence of transitions of a reset net, the value of $\overline{u^n}(x)(i)$ given $x(i)$ is given by $x(i) + n * \delta(u)(x)$ if u doesn't reset the i-th component, and $\overline{u}(x)(i)$ otherwise. More generally, the functions $\overline{\varphi}$ for $\varphi \in \text{DJS}(\mathcal{S})$ are computable.

Proposition 4.4. *For $\mathcal{S} \in$ VASRR, $\text{ITER}^\infty(\overline{\text{DJS}}(\mathcal{S}) \cup \overline{A})$ is an acceleration strategy.*

Proof. One can show by induction on $k \in \mathbb{N}$, that $H_k = \textsc{iter}^k(\overline{\textsc{djs}}(\mathcal{S}) \cup F)$ is a set of functions such that for any $h \in H_k$, one can find $\rho(h) \in \mathbb{N}^d$, $\delta(h) \in \mathbb{N}^d_\omega$ and $\mu(h) \in (\mathbb{N} \cup \{\bot\})^2 \times \{\bot\}^{d-2}$ such that $dom(h) = \uparrow\rho(h)$, $h(x)(i) = x(i) + \delta(h)(i)$ if $\mu(h) = \bot$ and $\mu(h)(i)$ if $\mu(h) \neq \bot$. From that, these functions are monotonic and computable. As $\textsc{iter}^\infty(\overline{\textsc{djs}}(\mathcal{S}) \cup F)$ is r.e. by construction, we get our result. □

Given a state x, we say that $u \in A^*$ is a *simple loop* on x if $x \in dom(\overline{u})$ and there exists i, such that $\mu(\overline{u})(i) = \bot$ (i.e. no resets are encountered on place i), $\delta(\overline{u})(i) > 0$ and $x(i) < \omega$. [24] used simple loops as a first kind of witness for unboundedness. Moreover, they showed that if there was no simple loop in an unbounded system, then there must exist a regular loop that witnesses unboundedness (and adds an ω in the two components that are reset). By applying all possible simple loops and regular loops, we can get the following result

Lemma 4.5. *Let $\mathcal{S} \in \textsc{vasrr}$, $(\alpha_0, r_0) \in \mathbb{N}^2_\omega \times \mathbb{N}^{d-2}_\omega$, and assume that $\{r \in \mathbb{N}^{d-2}_\omega \mid \exists \alpha \in \mathbb{N}^2_\omega, (\alpha, r) \in Reach_{\mathcal{S}}(\alpha_0, r_0)\}$ is finite. Then, Lub $Cover_{\mathcal{S}}(\alpha_0, r_0) = Acc^{\text{STRAT}}_{\mathcal{S}}(\alpha_0, r_0)$ for $\textsc{strat} = \textsc{iter}^\infty(\overline{\textsc{djs}}(\mathcal{S}) \cup \overline{A})$*

Proof. In this proof, to avoid heavy notations, we use the following shortcuts when there is no risk of confusion: $0 = \{0\}$, $1 = \{1\}$ and $01 = \{0, 1\}$.

Let $\mathcal{V} = \langle A, \delta, \mathcal{R}\rangle$ such that $\mathcal{S} = LTS(\mathcal{V})$. Let $R = \{r \in \mathbb{N}^{d-2}_\omega \mid \exists \alpha \in \mathbb{N}^2_\omega, (\alpha, r) \in Reach_{\mathcal{S}}(\alpha_0, r_0)\}$. We define $\leq_\omega$ a new ordering on $\mathbb{N}_\omega$ where ω is incomparable with elements of $\mathbb{N}$ and $\leq_\omega$ is the normal ordering on $\mathbb{N}$. This ordering is extended pointwise on $\mathbb{N}^d_\omega$. For $h \in H$ and $x \in X$, we say that h is a *real acceleration* if $h(x)$ contains strictly more ω's than x. For each $r \in R$, the set $U_r = \{\alpha \in \mathbb{N}^2_\omega \mid$ There exists a real acceleration in H on $(\alpha, r)\}$ is upward-closed for $\leq_\omega$. Hence, its complement D_r is a finite union of $\mathbb{N}^2 \times Q_{r,\emptyset}$, $\mathbb{N} \times Q_{r,0}$, $Q_{r,1} \times \mathbb{N}$ and $Q_{r,01}$ where $Q_{r,\emptyset}$ is a finite subset of $\mathbb{N}^0_\omega$ (so either the empty set, or singleton containing the empty vector), $Q_{r,0}$ and $Q_{r,1}$ are finite subsets of $\mathbb{N}_\omega$ and $Q_{r,01}$ is a finite subset of $\mathbb{N}^2_\omega$.

With these definitions, we will build a generalized VAS with states, that will simulate the original one, and whose reachability set will be shown finite. As the transitions of our newly defined will correspond either to normal transitions or to simple/regular loops, this will give us the result.

We define the finite set D representing the subset of the states of $\mathcal{S}$ that we will simulate, and Q the control structure of our new VASS by:

$$\begin{array}{rcl} D &=& \{(\alpha, r) \mid r \in R \wedge \alpha \in D_r\} \\ Q &=& \bigcup_{r \in R}\{(r, \emptyset)\} \times Q_{r,\emptyset} \cup \\ && \bigcup_{r \in R}\{(r, 0)\} \times Q_{r,0} \cup \\ && \bigcup_{r \in R}\{(r, 1)\} \times Q_{r,1} \cup \\ && \bigcup_{r \in R}\{(r, 01)\} \times Q_{r,01} \end{array}$$

As our state space only simulates D, we need a way to turn any state outside this set into D. This is done by an acceleration function *acc*. Let $(\alpha, r) \notin D$. We define $acc(\alpha, r) = (\alpha', r)$ where $\alpha' \in \mathbb{N}^2_\omega$ is obtained from α by replacing some components by ω such that $\alpha' \in D_r$ and that $(\alpha', r) \in \downarrow Acc^H_{\mathcal{S}}(\alpha, r)$. At least one such α' can be defined, as we can apply real accelerations on (α, x) until no more can be performed (at most two accelerations are required) because each real acceleration adds an ω to the state. However, more than one possible value exists, so we arbitrarily resolve this ambiguousity by taking one.

Now, we can define our translation function φ from D (the simulated state space) to $\mathbb{N}_\omega^2 \times Q$ (the state space of our new VASS).

$$\begin{array}{rcll}
\varphi(\alpha, r) & = & (\alpha(0), \alpha(1), (r, \emptyset)) & \text{if } \alpha \in \mathbb{N}^2 \times Q_{r,0} \\
\varphi(\alpha, r) & = & (\alpha(0), \omega, (r, 0, \alpha(1))) & \text{if } \alpha \in \mathbb{N} \times Q_{r,1} \\
\varphi(\alpha, r) & = & (\omega, \alpha(1), (r, 1, \alpha(0))) & \text{if } \alpha \in Q_{r,1} \times \mathbb{N} \\
\varphi(\alpha, r) & = & (\omega, \omega, (r, 01, \alpha(0), \alpha(1))) & \text{if } \alpha \in Q_{r,01}
\end{array}$$

This function is an injection. We write $D' = \varphi(D)$ and φ^{-1} its inverse (defined on D'). Note that D' is upward-closed.

The final part of the simulation is to translate the operations. We define $F' = \{\overline{(a, q)} \mid a \in A \wedge q \in Q$ with:

$$\begin{array}{rcl}
dom(\overline{(a, q)}) & = & (\mathbb{N}^2 \times \{q\}) \cap D' \\
\overline{(a, q)}(x, q) & = & \varphi(acc(\varphi^{-1}(x, q) + \delta(a)))
\end{array}$$

Note that the translation of φ either preserves the counters of $\mathcal{S}$, or turns them into control states (setting the counter to ω). Similary, φ^{-1} may turn back a control state into a counter, which means that the counter would be set to a precise value. Moreover, *acc* may only fix some of them into ω, hence the effect of each of these functions on the counters is either to add a fixed value to them, or to set them to a fixed value. This is a generalized reset transition as defined above. For this reason, we have that $\mathcal{S}' = \langle \mathbb{N}^2 \times Q, F', \leq \rangle$ is the transition system associated to a generalized VASS with 2 resets. We will show that this is a faithful simulation:

(1) If $\varphi(y) \in Reach_{\mathcal{S}'}(\varphi(x))$, then $y \in \downarrow Acc_{\mathcal{S}}^{H}(x)$ (we recall $H = \text{ITER}^{\infty}(\overline{\text{DJS}}(\mathcal{S}) \cup F)$)

(2) If $y \in Reach_{\mathcal{S}}(x)$, then, there exists $y' \geq y$ such that $\varphi(y') \in Reach_{\mathcal{S}'}(\varphi(x))$.

We show both of these by induction on the length of the transition sequence. For (1), we first note that if $x \in D'$, we have $Reach_{\mathcal{S}'}(x) \subseteq D'$, so we can consider $\varphi(z)$ such that $\varphi(z) \in Reach_{\mathcal{S}'}(\varphi(x))$ and $y \in Post_{\mathcal{S}}(\varphi(z))$. Then, by induction hypothesis, $z \in \downarrow Acc_{\mathcal{S}}^{H}(x)$. But, by definition of $\overline{(a, q)}$, we have $y = acc(z + \delta(a))$, which means that $y \in \downarrow Acc_{\mathcal{S}}^{H}(z)$. By transitivity of *Acc*, we get $y \in \downarrow Acc_{\mathcal{S}}^{H}(x)$. For (2), we consider z such that $z \in Reach_{\mathcal{S}}(x)$ and $y \in Post_{\mathcal{S}}(z)$. Then, by induction hypothesis, there exists $z' \geq z$ such that $\varphi(z') \in Reach_{\mathcal{S}'}(\varphi(x))$. We consider $a \in A$ such that $y = z + \delta(a)$. Then, we have $\overline{a, q}(\varphi(z')) = \varphi(acc(z' + \delta(a)))$. We define $y' = acc(z' + \delta(a)) \geq y$, and we have $\varphi(y') \in Reach_{\mathcal{S}'}(\varphi(x))$.

This means that for any $x \in \mathbb{N}_\omega^d$, $Cover_{\mathcal{S}}(x) \subseteq \downarrow\varphi^{-1}(Reach'_{\mathcal{S}}(\varphi(x))) \subseteq Acc_{\mathcal{S}}^{H}(x)$.

We claim $\mathcal{S}'$ has a finite set of reachable states. Indeed, if it is not, it means there is either a simple loop or a regular loop on some reachable state (α, q). We look at these two cases:

- If it is a simple loop, then the iteration of a word of H^* is an element of H^*, hence $(\alpha, q) \in U_r$, which is a contradiction.

- If it is a regular loop, then one can remark that the values of the first two counters encountered after each iteration of the regular loop are different from ω. Thus, each of these iteration can be made to correspond to a run without acceleration (by continuity). This means that we get an acceleration of $\textsc{djs}(\mathcal{S})$, which is again a contradiction.

Because $\mathcal{S}'$ has a finite set of reachable states, we have $\downarrow\varphi^{-1}(Reach'_{\mathcal{S}}(\varphi(\alpha, r_0)))$ is a closed set, which means because $Cover_{\mathcal{S}}(x) \subseteq \downarrow\varphi^{-1}(Reach'_{\mathcal{S}}(\varphi(x))) \subseteq Acc^H_{\mathcal{S}}(x)$ that we have $Lub\ Cover_{\mathcal{S}}(x) \subseteq \downarrow\varphi^{-1}(Reach'_{\mathcal{S}}(\varphi(x))) \subseteq Acc^H_{\mathcal{S}}(x)$, and because we also have $Acc^H_{\mathcal{S}}(x) \subseteq Lub\ Cover_{\mathcal{S}}(x)$, we end with $Acc_{\mathcal{S}}(\alpha_0, r_0) = Lub\ Cover_{\mathcal{S}}(\alpha_0, r_0)$, which concludes the demonstration. □

4.5 Computing the Cover

With the help of $\textsc{iter}^\infty$, we can generalize this result into:

Theorem 8. *The procedure 1 terminates for any $\mathcal{S} \in \textsc{vasrr}$, any $x \in States(\mathcal{S})$ and for* $\textsc{strat} = \textsc{iter}^\infty(\overline{\textsc{djs}}(\mathcal{S}) \cup \overline{A})$

We first prove a technical lemma:

Lemma 4.6. *We consider $\mathcal{S} \in \textsc{vasrr}$ and an increasing sequence $(x_n)_{n\in\mathbb{N}}$ with $x_{n+1} \in Acc^H_{\mathcal{S}}(x_n)$. We define:*

$$\begin{array}{rcl} Y &=& \{y \in \mathbb{N}^d_\omega \mid \exists j \ \in \mathbb{N}, y \in Acc^H_{\mathcal{S}}(x_j) \wedge x_{j+1} \in Acc^H_{\mathcal{S}}(y)\} \\ proj(Y) &=& \{r \in \mathbb{N}^{d-2}_\omega \mid \exists \alpha \in \mathbb{N}^2,\ (\alpha, r) \in Y\} \end{array}$$

Then, if $proj(Y)$ is finite, one can build a generalized VASS with 2 resets $\mathcal{V}' \in \textsc{vasrr}$ with $LTS(\mathcal{V}') = \langle \mathbb{N}^{d'}_\omega, F', \leq\rangle$, an injective continuous function $\varphi : Y \to \mathbb{N}^{d'}_\omega$ and a function $\psi : H' \to H$ (with $H' = \textsc{iter}^\infty(\overline{\textsc{djs}}(\mathcal{S}') \cup F')$) such that:

- *$\{r \in \mathbb{N}^{d'-2}_\omega \mid \exists \alpha \in \mathbb{N}^2, (\alpha, r) \in Reach_{\mathcal{S}'}(\varphi(x_0))\}$ is finite.*
- *$\forall h' \in H',\ \forall x \in Y,\ h'(\varphi(x)) = \varphi(\psi(h')(x))$.*

Proof. First, we remark that if $w = u_1 u_2 \ldots u_n \ldots$ is a simple or regular loop on x (we have $u_i = u_j$ for simple loops) then for any prefix v of w, there exists $v' \in A^*$ such that $w = vv'w'$ and w' is a simple or regular loop on $\overline{vv'}(x)$. Hence, this means that all states encountered along accelerations also belong to Y.

We define a generalized VASS with 2 resets $\mathcal{V}' = \langle Q, A, \rho', \delta', \mu', tr'\rangle$:

$$\begin{array}{rcl} Q &=& proj(Y) \\ A &=& \{a' \mid a \in A\} \\ tr'(q,a) &=& q + \delta(a)(2 \ldots d-1) \\ && \text{for } i \in \{1,2\}: \\ \rho'(\overline{(q,a)})(i) &=& max(0, -\delta(a)(i)) \\ \delta'(\overline{(q,a)})(i) &=& \delta(a)(i) \\ \mu'(\overline{(q,a)})(i) &=& \left\{ \begin{array}{ll} 0 & \text{if } i \in \mathcal{R}(a) \\ \bot & \text{if } i \notin \mathcal{R}(a) \end{array}\right. \end{array}$$

Then, one can define the following translations:

$$\begin{aligned} \varphi(x) &= (x(0), x(1), x(2\ldots d-1)) \\ \psi(\overline{(q_1,a_1)\ldots(q_n,a_n)}) &= \overline{a_1\ldots a_n} \end{aligned}$$

As the functions of H' are defined by composition and union of $\overline{(q_1,a_1)\ldots(q_n,a_n)}$, we extend ψ by morphism on H'. It is technically straightforward to show that (1) and (2) are fulfilled by this simulation. $\square$

We now prove theorem 8.

Let $H = \text{ITER}^\infty(\overline{\text{DJS}}(\mathcal{S}) \cup \overline{A})$. Thanks to lemma 2.12, we only have to show that for any increasing sequence $(x_n)_{n\in\mathbb{N}}$ with $x_{n+1} \in Acc^H_{\mathcal{S}}(x_n)$, we have $lub\{x_n \mid n \in \mathbb{N}\} \in Acc^H_{\mathcal{S}}(x_0)$. By eventually extracting subsequences, we assume that for each $i \in \{1,\ldots,d\}$, we either have $\forall n \in \mathbb{N}, x_n(i) \neq \omega$ or $\forall n \in \mathbb{N},\ x_n(i) = \omega$. We perform the proof by induction on the number of components that are not equal to ω.

We consider $Y = \{y \in \mathbb{N}^d_\omega \mid \exists j\ \in \mathbb{N}, y \in Acc^H_{\mathcal{S}}(x_j) \wedge x_{j+1} \in Acc^H_{\mathcal{S}}(y)\}$ and $proj(Y) = \{r \in \mathbb{N}^{d-2}_\omega \mid \exists \alpha \in \mathbb{N}^2,\ (\alpha, r) \in Y\}$.

Two cases may occur:

- $proj(Y)$ is infinite. Then, since $(\mathbb{N}^d_\omega, \leq)$ is a wpo, there exists $i \in \{3,\ldots,d\}$, and $y_1, y_2 \in Y$ such that $y_1 \leq y_2$ and $y_1(i) < y_2(i)$. Let $k_1 \in \mathbb{N}$ and $h_1, h'_1 \in H^*$ be such that $y_1 = h_1(x_{k_1})$ and $x_{k_1+1} = h'_1(y_1)$. Similarly, we take $k_2 \in \mathbb{N}$ and $h_2, h'_2 \in H^*$ be such that $y_2 = h_2(x_{k_2})$ and $x_{k_2+1} = h'_2(y_2)$. We define $k'_2 = max(k_1+1, k_2)$. Let $y'_2 = h_2(x_{k'_2})$. Because $k'_2 \geq k_2$, we have $x_{k'_2} \geq x_{k_2}$ and by the monotony of the functions of H, we get $y'_2 \geq y_2$. As we have $y'_2 \in Acc^H_{\mathcal{S}}(y_1)$, let $h \in H^*$ such that $y'_2 = h(y_1)$. Then, we have $h^\infty \in H$, and $h^\infty(y'_2) \geq y_2$, with $h^\infty(y'_2)$ having more ω's than y_2. But because, we have $x_{k_2+1} \in Acc^H_{\mathcal{S}}(y_2)$, we consider the sequence $(w_i)_{i\in\mathbb{N}}$, $w_i \in H^*$ such that $x_{k_2+i+1} = w_i(x_{k_2+i})$. Then, we get the sequence z_i by $z_0 = h'_2(h^\infty(y'_2))$ and $z_{i+1} = w_i(z_i)$. This is a sequence that contains strictly more ω's than the original one, so we get that $lub\{x_i \mid i \in \mathbb{N}\} \leq lub\{z_i \mid i \in \mathbb{N}\} \in Acc^H_{\mathcal{S}}(z_0)$. By transitivity of Acc, as $z_0 \in Acc^H_{\mathcal{S}}(x_0)$, we have $lub\{x_i \mid i \in \mathbb{N}\} \in Acc^H_{\mathcal{S}}(x_0)$.

- $proj(Y)$ is finite. Intuitively, this means that if we project our system on the non-resetable components, the path we are considering is visiting only a finite number of states. Formally, one can build $\mathcal{S}' = \langle \mathbb{N}^{d'}_\omega, F', \leq\rangle \in$ VASRR of dimension d', an injective continuous function $\varphi : Y \to \mathbb{N}^{d'}_\omega$ and a function $\psi : H' \to H$ (with $H' = \text{ITER}^\infty(\overline{\text{DJS}}(\mathcal{S}') \cup F')$) such that (1) $\{r \in \mathbb{N}^{d'-2}_\omega \mid \exists \alpha \in \mathbb{N}^2, (\alpha, r) \in Reach_{\mathcal{S}'}(\varphi(x_0))\}$ is finite. (2) $\forall h' \in H',\ \forall x \in Y,\ h'(\varphi(x)) = \varphi(\psi(h')(x))$. This is lemma 4.6 which corresponds to defining a system that simulates only the runs of $\mathcal{S}$ that stay inside Y. (2) implies that $\forall x \in Y,\ Acc^{H'}_{\mathcal{S}'}(\varphi(x)) \subseteq \varphi(Acc^H_{\mathcal{S}}(x))$. Now, because of (1) we can apply lemma 4.5 and get that $\varphi(lub\ \{h(x_i)\}) = lub\ \{\varphi(h(x_i))\} \in Acc^{H'}_{\mathcal{S}'}(\varphi(x_0))$. This leads that $\varphi(lub\ \{h(x_i)\}) \in \varphi(Acc^H_{\mathcal{S}}(x_0))$, and by injectivity of φ, that $lub\ \{h(x_i)\} \in Acc^H_{\mathcal{S}}(x_0)$.

Corollary 4.7. *For Vector Addition Systems with two resets,* CLOVER SET *is computable.*

As a consequence, place-boundedness (an instance of the more general CLOVERABILITY) is decidable.

4.6 Summary of results on VAS with resets

With the additional result (underlined) presented in this chapter, we can present an extensive overview of the decidability results on VAS with resets:

	no reset	one reset	two resets	three resets
REACHABILITY	decidable [49]	decidable (red. from [55])	undecidable [23]	undecidable [23]
COVERABILITY	decidable (WSTS)	decidable (WSTS)	decidable (WSTS)	decidable (WSTS)
BOUNDEDNESS	decidable (strict WSTS)	decidable [24]	decidable [24]	undecidable [23]
PLACE-BOUNDEDNESS	decidable [41]	decidable (red. from 5.3)	<u>decidable</u>	undecidable [23]
REP. COVERABILITY	decidable [25]	decidable (red. from 5.4.1)	undecidable [50]	undecidable [23]

Despite this array being completely filled, some open questions remain, mostly regarding complexity. One interesting one is inspired by recent works by Praveen *et al.* [18], that extends the Rackoff proof to strongly increasing Affine Nets (that would correspond in our setup to allow operations that add the content of one counter to another). In order to further extend this work to VAS with 2 resets, one would have to bound the maximum length of possible regular loops to consider. The proof proposed by Dufourd *et al.* already uses a length-based reasoning, so this might lead to an interesting upper bound. We leave such an analysis for further work.

Chapter 5

Vector Addition Systems with hierarchical zero-tests

The results of this chapter are based on joint work with Alain Finkel, Jerome Leroux and Marc Zeitoun, originally published in [17], [14]. The results of section 5.2 were originally published in [15]. These publications were based on the Vector Addition Systems with one zero-test model. The extension to hierarchical zero-tests is original.

A Vector Addition System is a restricted version of a Counter Machine where the only operations allowed by transitions are incrementations or decrementations of counters. Implicitly, the decrementations allow to contraint the firing of a transition by testing whether a given counter is greater than a constant value. It is tempting to allow the complementary operation, that would test whether a given counter is less than a constant value.

However, one can see that allowing such an operation is equivalent to allow operations that test whether a given counter is equal to zero (or more generally to any constant value). Unfortunately, it is known that counter machines that allow incrementation, decrementation, and testing counters for zero are Turing-complete as soon as two counters are available [51].

Thus, if we want to allow such an operation, we must restrict it to apply on a single counter if we want meaningful decidability results. Actually, we will look at a slightly more general model, introduced by Reinhardt [54], where zero-tests are allowed on multiple counters as long as the counters are ordered in a way that a counter can be tested for zero only if all counters of lesser index are also tested for zero. We introduce this model formally in the following way:

Definition 5.1. *A Vector Addition System with hierarchical zero-tests (shortly: VAS_{0^*}) of dimension d is a tuple $\langle A_0, A_1, \ldots, A_d, \delta\rangle$ where:*

- *A_i's are finite set of actions (performing i zero-tests).*
- *$\delta : \bigcup_{0\leq i\leq d} A_i \to \mathbb{Z}^d$ provides the effect of the actions on the counters.*

We define $A_{\leq p} = \bigcup_{0\leq i\leq p} A_i$ and $A = A_{\leq d}$. If we force all A_i's for $i \geq 2$ to be empty, we get the natural class of *Vector Addition Systems with one zero-tests.*

One can associate to a VAS_{0^*} a transition system $\langle \mathbb{N}^d, \overline{A} \rangle$ where, for $a \in A_i$, the function $\overline{a}$ is defined by:

$$\begin{aligned} x \in dom(a) \quad &\iff \begin{cases} \forall j \leq i.\ x(i) = 0 \\ x + \delta(a) \geq 0 \end{cases} \\ \overline{a}(x) \quad &= x + \delta(a) \end{aligned}$$

Note that this transition system is not a WSTS unless $A = A_0$. Indeed, if we take for example $a \in A_1$ with $\delta(a) = 0$, a is fireable from $(0, 0, \ldots, 0)$ but not from $(1, 0, \ldots, 0)$. For this reason, specific methods are required to verify such systems. We describe in the next section the state of the art, and propose in sections 5.2 and 5.3 new techniques that allow to respectively show the decidability of REACHABILITY and CLOVERABILITY. Section 5.4 will apply the previous results to derive some related results.

5.1 Related Work

The study of VAS with hierarchical zero-tests (or of the subclass of VAS with one zero-test) began recently, but already a fair number of results are known for these models. Reinhardt ([54, 55]) has shown that the reachability problem is decidable for VAS with hierarchical zero-tests. For the subclass of VAS with one zero-tests, Abdulla and Mayr have shown that the coverability problem is decidable in [7] by using the backward procedure of WSTS (see section 5.3.1 for a summary of Abdulla technique and a comparison with the technique presented in section 5.3). Finally, boundedness, termination and reversal-boundedness (whether the counters can alternate infinitely often between the increasing and the decreasing modes) were shown to be decidable by using a forward procedure, a finite but non-complete Karp and Miller tree (Finkel and Sangnier [32]).

The decidability results on VAS with hierarchical zero-tests have already been used to show the decidability of problems on other models, for example the Priced Timed Petri Nets of Abdulla [7]. Moreover, Atig and Ganty have shown that this class of VAS were equivalent to a subclass of Pushdown Counter Automatas were the stack is restricted to *index-bounded* behaviour, i.e. such that the associated context-free grammar never uses more than a bounded number of variables at the same time. It could be argued that a reasonable number of programs will in fact follow such a stack discipline.

5.2 Reachability in VAS with hierarchical zero-tests

If reachability was already shown to be decidable by Reinhardt [54, 55], a few reasons prompt us to provide an alternate proof:

- The proof of Reinhardt is involved, and is difficult to apprehend.
- Leroux recently provided an extremely short proof (compared to the previous versions) of the reachability for VAS. By providing an extension of this proof to VAS with

hierarchical zero-tests, we believe that people with a good knowledge of Leroux proof will have an easier time understand this proof.

Let us try to summarize the proof structure of [47], that we will mimic. The main idea is that if a transition relation has some properties, one can find a witness of non-reachability. As witness of reachability always exist (it is simply the sequence of transitions used to go from the initial state to the final state), by enumerating all possible witness of reachability and non-reachability, we are guaranteed to terminate at some point. These required properties are given by the notion of almost semilinear set, which itself relies on the notions of asymptotically definable periodic sets and Lambert sets, that generalizes linear and semilinear sets. After having given in section 5.2.1 the definitions of asymptotically definable periodic, Lambert and almost semilinear sets, we will recall in section 5.2.2 some tools from [47], and especially the result that if a relation is almost semilinear, one can find a witness of non-reachability which is a Presburger forward invariant.

Now, to prove that our reachability relation is almost semilinear, we have to show that each finite run can be associated a production relation, such that (1) the set of runs, ordered by inclusion of their production relations is well-ordered and (2) these productions relations are asymptotically definable. With a few additionnal assumptions, this means the reachability relation can be written as a finite sum and union of productions relations (the relations associated to the minimal elements of the previously defined well-order) and can be shown to be almost semilinear. We will introduce our version of these production relations in section 5.2.3 and prove that they are well-ordered. Then, section 5.2.5 will show that these production relations are asymptotically definable and we will conclude in section 5.2.6.

5.2.1 Definable conic sets, Lambert and almost semilinear sets

A set $C \subseteq \mathbb{Q}^d$ is conic if it is periodic and $\mathbb{Q}_{\geq 0}C = C$. A conic set is finitely generated if there exists a finite set $\{c_1, \dots, c_n\} \subseteq \mathbb{Q}$ such that $C = \mathbb{Q}_{\geq 0}c_1 + \dots + \mathbb{Q}_{\geq 0}c_n$.

Definition 5.2. *([47], Definitions 3.1 and 4.1)*
A conic set is said to be definable *if it can be defined in $FO(\mathbb{Q}, +, \leq, 0, 1)$.*
A periodic set $P \subseteq \mathbb{N}^d$ is said to be asymptotically definable *if $\mathbb{Q}_{\geq 0}P$ is definable.*

Definition 5.3. *([48], Definition 4.6)*
A set $L \subseteq \mathbb{N}^d$ is Lambert if it is a finite union of sets $b_i + P_i$ where $b_i \in \mathbb{N}^d$ and $P_i \subseteq \mathbb{N}^d$ is an asymptotically definable periodic set.

The stability of Lambert sets will be of importance in the sequel. We have the following properties[1]:

Proposition 5.1. *Given $L \subseteq \mathbb{N}^{d_1}, L' \subseteq \mathbb{N}^{d_2}$ Lambert sets and $k \in \mathbb{N}$:*

1. *For $d_1 = d_2$, $L \cup L'$ is Lambert.*
2. *$L \times L'$ is Lambert.*

[1] we recall that we have $0 \star X = \{0\}$, $(k+1) \star X = (k \star X) + X$ and $\mathbb{N} \star X = \bigcup_{k \in \mathbb{N}} k \star X$

3. For $d_1' < d_1$, $\{x \in \mathbb{N}^{d_1'} \mid \exists y \in \mathbb{N}^{d_1-d_1'},\ (x,y) \in L\}$ is Lambert.

4. For $d_1 = d_2$, $L + L'$ is Lambert.

5. $k \star L$ is Lambert.

6. $\mathbb{N} \star L$ is an asymptotically definable periodic set (more generally Lambert).

7. If δ is a linear function, then $\delta(L)$ is Lambert.

Proof. We have $L = \bigcup_{1\le i\le p} b_i + P_i$ and $L' = \bigcup_{1\le i\le q} b_i' + P_i'$ with $b_i \in \mathbb{N}^{d_1}$, $b_i' \in \mathbb{N}^{d_2}$ and $P_i \subseteq \mathbb{N}^d$, $P_i' \subseteq \mathbb{N}^d$ asymptotically definable periodic sets.

(1) is by definition of a Lambert set.

For (2), we have:

$$\begin{aligned} L \times L' &= \bigcup_{1\le i\le p} \bigcup_{1\le j\le q} (b_i + P_i) \times (b_i' + P_i') \\ &= \bigcup_{1\le i\le p} \bigcup_{1\le j\le q} (b_i, b_j') + P_i \times P_j'. \end{aligned}$$

Because P_i and P_i' are asymptotically definable, $P_i \times P_i'$ is aymptotically definable, which makes $L \times L'$ Lambert.

To show (3), we first show the property for asymptotically definable periodic sets. Let's take P an asymptotically definable periodic set and $P' = \{x \in \mathbb{Q}^{d_1'} \mid \exists y \in \mathbb{Q}^{d_1-d_1'},\ (x,y) \in P\}$. Then if $x \in P'$ and $x' \in P'$, we have $y, y' \in \mathbb{Q}^{d_1-d_1'}$ such that $(x,y) \in P$ and $(x', y') \in P$, which gives $(x + x', y + y') \in P$ and $x + x' \in P'$. Moreover, we have:

$$\begin{aligned} \mathbb{Q}_{\ge0}P' &= \{x \in \mathbb{Q}^{d_1'} \mid \exists y \in \mathbb{Q}^{d_1-d_1'},\ \exists k \in \mathbb{Q}_{\ge0},\ (kx, y) \in P\} \\ &= \{x \in \mathbb{Q}^{d_1'} \mid \exists y \in \mathbb{Q}^{d_1-d_1'},\ (x, y) \in \mathbb{Q}_{\ge0}P\} \end{aligned}$$

which means that from a definition of $\mathbb{Q}_{\ge0}P$ in $FO(\mathbb{Q}, +, \le, 0, 1)$, we easily get the definition of $\mathbb{Q}_{\ge0}P'$. And if $b_i = (c_i, c_i')$ with $c_i \in \mathbb{Q}^{d_1'}$, we have:

$$\{x \in \mathbb{Q}^{d_1'} \mid \exists y \in \mathbb{Q}^{d_1-d_1'},\ (x,y) \in L\} = \bigcup_i c_i + \{x \in \mathbb{Q}^{d_1'} \mid \exists y \in \mathbb{Q}^{d_1-d_1'},\ (x,y) \in P_i\}$$

which gives us the result.

To show (4), we note that $L + L' = \bigcup_{1\le i\le p} \bigcup_{1\le j\le q} (b_i + b_j') + (P_i + P_j')$. Because the sum of periodic sets is periodic, $L + L'$ is periodic. Moreover, we get easily get the definition of $\mathbb{Q}_{\ge0}(P + P') = \mathbb{Q}_{\ge0}P + \mathbb{Q}_{\ge0}P'$ from the definition of $\mathbb{Q}_{\ge0}P$ and $\mathbb{Q}_{\ge0}P'$ in $FO(\mathbb{Q}, +, \le, 0, 1)$. Hence, $L + L'$ is Lambert.

(5) is a direct consequence of (4).

To show (6), we notice that $\mathbb{N} \star L$ is periodic, and we have $\mathbb{Q}_{\ge0}(\mathbb{N} \star L) = \sum_i \mathbb{Q}_{\ge0}b_i + \sum_i \mathbb{Q}_{\ge0}P_i$. As $\mathbb{Q}_{\ge0}P_i$ is definable in $FO(\mathbb{Q}, +, \le, 0, 1)$, so is $\mathbb{Q}_{\ge0}(\mathbb{N} \star L)$. This makes $\mathbb{N} \star L$ asymptotically definable.

Let's finally show (7). We have $\delta(L) = \bigcup_{1 \leq i \leq p} \delta(b_i) + \delta(P_i)$. As δ is linear, we have $\delta(P_i)$ periodic and $\mathbb{Q}_{\geq 0}\delta(P_i) = \delta(\mathbb{Q}_{\geq 0}P_i)$, which makes $\mathbb{Q}_{\geq 0}\delta(P_i)$ easily definable from the definition of $\mathbb{Q}_{\geq 0}P_i$ in $FO(\mathbb{Q}, +, \leq, 0, 1)$. □

Definition 5.4. *([47], Definition 4.6)*
A set $X \subseteq \mathbb{N}^d$ is almost semilinear *if for all Presburger sets S, $S \cap X$ is Lambert.*

5.2.2 Important results from Leroux

We recall in this section a few important results from [47].

For a set $X \subseteq \mathbb{Q}^d$, the closure of X, written $cl(X)$ is defined by:

$$cl(X) = \{l \mid \forall \tau > 0,\ \exists x \in X,\ max_i(x - l)(i) < \tau \wedge max_i(l - x)(i) < \tau\}$$

We have this useful characterization of asymptotically definable periodic sets, that we will use to show that our production relation are asymptotically definable:

Theorem 9. *([47], Theorem 3.8)*
A periodic set $P \subseteq \mathbb{N}^d$ is asymptotically definable if and only if the conic set $cl((\mathbb{Q}_{\geq 0}P) \cap V)$ is finitely generated for every vector space $V \subseteq \mathbb{Q}^d$

The second theorem needed is the one motivating almost semilinear sets. An almost semilinear relation admits witnesses of non-reachability:

Theorem 10. *([47], Theorem 6.1)*
Let R be a reflexive relation over $\mathbb{N}^d$ such that R^* is almost semilinear. Let $X, Y \subseteq \mathbb{N}^d$ be two Presburger sets such that $R^* \cap (X \times Y)$ is empty. There exists a partition of $\mathbb{N}^d$ into a Presburger R-forward invariant that contains X and a Presburger R-backward invariant that contains Y.

5.2.3 Production relations

For all the remaining sections, we will fix a VAS_{0^*} $\mathcal{V} = \langle A_0, \ldots, A_d, \delta \rangle$ of dimension d. We also consider $\mathcal{V}_p = \langle A_0, \ldots, A_p, \emptyset, \ldots, \emptyset, \delta_{|A_{\leq p}} \rangle$ the restriction of $\mathcal{V}$ to its transitions testing at most the p first counters for zero. We have $\longrightarrow$ (or $\xrightarrow{A^*}$) the transition relation of $\mathcal{V}$ and $\xrightarrow{A^*_{\leq p}}$ the transition relation of $\mathcal{V}_p$.

In the next sections, we will prove by induction on p the following result:

Theorem 11. *Let $\mathcal{V}$ be a VAS with hierarchical zero-tests and let $p \geq 0$. $\xrightarrow{A^*_{\leq p}}\!\!\!\!\rightarrow$ is an almost semilinear relation.*

Note that the base case (for $p = 0$) corresponds to the original proof of Leroux. However, it can also be seen as an instance of the generalized proof below, where the case $p = -1$ corresponds to the trivial of the case of the empty transition system (i.e. $\longrightarrow\!\!\!\!\rightarrow$ is the identity).

We will use as induction hypothesis a direct consequence of this result:

Corollary 5.2. *Let $\mathcal{V}$ be a VAS with hierarchical zero-tests and let $p \geq 0$. Then, the following set is almost semilinear:*

$$\{(x, v, y) \mid \exists u \in A^*_{\leq p}.\ |u| = v \wedge x \xrightarrow{u}\!\!\!\!\rightarrow v\}$$

Proof. The idea is to add counters that will compute that Parikh image of the path. Formally, if we consider a VAS with hierarchical zero-tests $\mathcal{V} = \langle A_0, A_1, \ldots, A_p, \delta\rangle$ of dimension d with $A = \bigcup A_i = \{a_1, a_2, \ldots, a_k\}$, we define $\mathcal{V}_2 = \langle A_0, A_1 \ldots, A_p, \delta'\rangle$ of dimension $d + k$ by:

$$\begin{array}{rcll} \delta'(a_i)(j) & = & \delta(a_i)(j) & \text{if } 0 \leq j < d \\ \delta'(a_i)(j) & = & 1 & \text{if } d \leq j < d+k \text{ and } j = d+i \\ \delta'(a_i)(j) & = & 0 & \text{if } d \leq j < d+k \text{ and } j \neq d+i \end{array}$$

Then, we have $(x, v) \xrightarrow{A^*_{\leq p}}\!\!\!\!\rightarrow_2 (x', v')$ iff there exists $u \in A^*_{\leq p}$ such that $x \xrightarrow{u} y$ and $v + |u| = v'$. By theorem 11, the reachability relation of $\mathcal{V}_2$ is almost semilinear, and by intersection with a semilinear, so is $\xrightarrow{A^*_{\leq p}}\!\!\!\!\rightarrow_2 \cap \mathbb{N}^d \times 0^k \times \mathbb{N}^d \times \mathbb{N}^k$, which is (up to rearranging components) the set $\{(x, v, y) \mid \exists u \in A^*_{\leq p}.\ |u| = v \wedge x \xrightarrow{u}\!\!\!\!\rightarrow v\}$. □

We recall from preliminaries that a run ρ of $\mathcal{V}$ is a sequence $m_0.a_1.m_1.a_2 \ldots a_n.m_n$ alternating markings $m_i \in \mathbb{N}^d$ and actions $a_i \in A$ such that for all $1 \leq i \leq n$, $m_{i-1} \xrightarrow{a_i} m_i$. m_0 is called the *source* of ρ, written $src(\rho)$. m_n is called the *target* of ρ, written $tgt(\rho)$. $a_1 a_2 \ldots a_n$ is called the *actions* of ρ, written $acts(\rho)$. A run ρ of $\mathcal{V}$ is also a run of $\mathcal{V}_p$ if all transitions appearing in ρ belong to $A_{\leq p}$. A single marking m is said to be a run of $\mathcal{V}_{-1}$.

We recall the definitions of the productions relations for a VAS of [47], adapted to our case by restricting the relation to runs of $\mathcal{V}_0$.

- For a marking $m \in \mathbb{N}^d$, $\xrightarrow[\mathcal{V}_0,[m]]{} \subseteq \mathbb{N}^d \times \mathbb{N}^d$ is defined by:

$$x \xrightarrow[\mathcal{V}_0,[m]]{} y \iff \exists u \in A^*_0,\ m + x \xrightarrow{u}\!\!\!\!\rightarrow m + y$$

- For a run $\rho = m_0.a_1.m_1 \ldots a_n.m_n$ of $\mathcal{V}_0$, $\xrightarrow[\mathcal{V}_0,\rho]{}$ is defined by:

$$\xrightarrow[\mathcal{V}_0,\rho]{} = \xrightarrow[\mathcal{V}_0,[m_0]]{} \circ \xrightarrow[\mathcal{V}_0,[m_1]]{} \circ \cdots \xrightarrow[\mathcal{V}_0,[m_n]]{}$$

Now, we also define the production relations $\xrightarrow[\mathcal{V}_p,[m]]{}$ for $p \geq 1$ by:

$$x \xrightarrow[\mathcal{V}_p,[m]]{} y \iff \begin{cases} \exists u \in A^*_{\leq p},\ m + x \xrightarrow{u}\!\!\!\!\rightarrow m + y \\ \forall 1 \leq i \leq p.\ x(i) = y(i) = 0 \end{cases}$$

To extend the definition of a production relation to a run ρ of $\mathcal{V}_p$, we consider the decomposition of $\rho = \mu_0.a_1.\mu_1 \ldots a_k.\mu_k$ such that forall $1 \leq i \leq k$, μ_i is a run of $\mathcal{V}_{p-1}$ and $a_i \in A_p$. In that case, we define the production relation of ρ by:

$$\xrightarrow[\mathcal{V}_p,\rho]{} = \xrightarrow[\mathcal{V}_{p-1},\mu_0]{} \circ \xrightarrow[\mathcal{V}_p,[tgt(\mu_0)]]{} \circ \xrightarrow[\mathcal{V}_{p-1},\mu_1]{} \circ \cdots \circ \xrightarrow[\mathcal{V}_p,[tgt(\mu_{k-1})]]{} \circ \xrightarrow[\mathcal{V}_{p-1},\mu_k]{}$$

Note that if we define $\xrightarrow[\mathcal{V}_{-1},m]{}$ to be the identity (m being a trivial run), this definition instantiated to $p = 0$ coincides with the earlier definition.

Proposition 5.3. *Let $0 \leq p \leq d$. For $m \in\in 0^p \times \mathbb{N}^{d-p}$ and μ a run of $\mathcal{V}_p$, $\xrightarrow[\mathcal{V}_{p,[m]}]{}$ and $\xrightarrow[\mathcal{V}_{p,\mu}]{}$ are periodic.*

Proof. The result is easy for $\xrightarrow[\mathcal{V}_{p,[m]}]{}$. We conclude by the fact the composition of periodic relations is periodic. ☐

Proposition 5.4. *For a run ρ of $\mathcal{V}_p$, we have:*

$$(src(\rho), tgt(\rho)) + \xrightarrow[\mathcal{V}_{p,\rho}]{} \subseteq \xrightarrow{A^*_{\leq p}}\!\!\!\!\rightarrow$$

Proof. We show this result on the length on ρ. Without loss of generality, we assume that p is the minimum value for which ρ is a run of $\mathcal{V}_p$ (-1 if the run is a single marking).

We have to consider two cases:

- $\rho = m$ is immediate by definition of $\xrightarrow[\mathcal{V}_{p,\rho}]{} = \xrightarrow[\mathcal{V}_{p,[m]}]{}$ given $src(\rho) = tgt(\rho) = m$.
- $\rho = \rho_1 \xrightarrow{a} \rho_2$ with $a \in A_p$, ρ_1 a run of $\mathcal{V}_{p-1}$ and ρ_2 a run of $\mathcal{V}_p$. Let $(x, z) \in \xrightarrow[\mathcal{V}_{p,\rho}]{}$. Then, as $\xrightarrow[\mathcal{V}_{p,\rho}]{} = \xrightarrow[\mathcal{V}_{p-1,\rho_1}]{} \circ \xrightarrow[\mathcal{V}_{p,[tgt(\rho_1)]}]{} \circ \xrightarrow[\mathcal{V}_{p,\rho_2}]{}$, there exists $y_1, y_2 \in \mathbb{N}^d$ such that $x \xrightarrow[\mathcal{V}_{p-1,\rho_1}]{} y_1 \xrightarrow[\mathcal{V}_{p,[tgt(\rho_1)]}]{} y_2 \xrightarrow[\mathcal{V}_{p,\rho_2}]{} z$. By induction hypothesis, there exists u_1 and u_3 in $A^*_{\leq p}$ such that:

$$\begin{array}{rcl} src(\rho) + x & \xrightarrow{u_1}\!\!\!\!\rightarrow & tgt(\rho_1) + y_1 \\ src(\rho_2) + y_2 & \xrightarrow{u_3}\!\!\!\!\rightarrow & tgt(\rho) + z \end{array}$$

  Also, by definition of $\xrightarrow[\mathcal{V}_{p,[tgt(\rho_1)]}]{}$, we have:

$$tgt(\rho_1) + y_1 \xrightarrow{u_2}\!\!\!\!\rightarrow tgt(\rho_1) + y_2$$

  Moreover, by definition of $\xrightarrow[\mathcal{V}_{p,[tgt(\rho_1)]}]{}$, we have, for all $i \in \{0, \ldots, p-1\}$, $y_2(i) = 0$. And because ρ is a run, this means that $tgt(\rho_1) \xrightarrow{a} src(\rho_2)$, which leads to $tgt(\rho_1) + y_2 \xrightarrow{a} src(\rho_2) + y_2$. By putting all the parts together, we have:

$$src(\rho) + x \xrightarrow{u_1}\!\!\!\!\rightarrow tgt(\rho_1) + y_1 \xrightarrow{u_2}\!\!\!\!\rightarrow tgt(\rho_1) + y_2 \xrightarrow{a} src(\rho_2) + y_2 \xrightarrow{u_2}\!\!\!\!\rightarrow tgt(\rho) + z$$

  and this gives us that $(src(\rho), tgt(\rho)) + (x, z) \in \xrightarrow{A^*_{\leq p}}\!\!\!\!\rightarrow$.

☐

5.2.4 Well-orderings of production relations

For two runs μ, μ' of $\mathcal{V}_p$, let us define $\preceq_p$ by:

$$\mu \preceq_p \mu' \iff (src(\mu'), tgt(\mu')) + \overrightarrow{\mathcal{V}_{p,\mu'}} \subseteq (src(\mu), tgt(\mu)) + \overrightarrow{\mathcal{V}_{p,\mu}}$$

Our aim is to show that $\preceq_p$ is a well-order. To do that, we define the order $\trianglelefteq_p$ on runs of $\mathcal{V}_p$ in the following way:

- For $\rho = m_0 \xrightarrow{a_1} m_1 \ldots \xrightarrow{a_k} m_k$ and $\mu = m'_0 \xrightarrow{b_1} m'_1 \ldots \xrightarrow{b_\ell} m'_\ell$ runs of $\mathcal{V}_0$ (this requires $a_i, b_i \in A_0$), we get a definition similar as in [47] (the condition $a_k = b_\ell$ has been added to be consistent with the general definition for $\mathcal{V}_p$):

$$m_0 \xrightarrow{a_1} m_1 \ldots \xrightarrow{a_k} m_k \trianglelefteq_0 m'_0 \xrightarrow{b_1} m'_1 \ldots \xrightarrow{b_\ell} m'_\ell \iff \begin{cases} m_0 \leq m'_0 \\ a_k = b_\ell \\ m_k \leq m'_\ell \\ \prod_{1\leq i\leq k}(a_i, m_i) \leq^{emb} \prod_{1\leq i\leq \ell}(b_i, m'_i) \end{cases}$$

  with $(a, m) \leq (b, m') \iff a = b \wedge m \leq m'$

- For $\rho = \rho_0 \xrightarrow{a_1} \rho_1 \ldots \xrightarrow{a_k} \rho_k$ and $\mu = \mu_0 \xrightarrow{b_1} \mu_1 \ldots \xrightarrow{b_\ell} \mu_\ell$ runs of $\mathcal{V}_p$ (with ρ_i, μ_i runs of $\mathcal{V}_{p-1}$ and $a_i, b_i \in A_p$), we have:

$$\rho_0 \xrightarrow{a_1} \rho_1 \ldots \xrightarrow{a_k} \rho_k \trianglelefteq_p \mu_0 \xrightarrow{b_1} \mu_1 \ldots \xrightarrow{b_\ell} \mu_\ell \iff \begin{cases} \rho_0 \trianglelefteq_{p-1} \mu_0 \\ a_k = b_l \\ \rho_k \trianglelefteq_{p-1} \mu_l \\ \prod_{1\leq i\leq k}(a_i, \rho_i) \leq^{emb} \prod_{1\leq i\leq \ell}(b_i, \mu_i) \end{cases}$$

  with $(a, \rho) \leq (b, \mu) \iff a = b \wedge \rho \trianglelefteq_{p-1} \mu$.

Note that by considering m a run of $\mathcal{V}_{-1}$ with $\trianglelefteq_{-1}=\leq$, the definition of $\trianglelefteq_0$ concides with the definition of $\trianglelefteq_p$ for $p = 0$.

Recursive applications of Higman's lemma gives us the following result:

Proposition 5.5. *For any $p \geq 0$, the order $\trianglelefteq_p$ is well.*

We also have the following property, shown by a straightforward induction on p.

Lemma 5.6. *For ρ, μ runs of $\mathcal{V}_p$, we have:*

$$\rho \trianglelefteq_p \mu \implies \begin{cases} src(\rho) \leq src(\mu) \\ tgt(\rho) \leq tgt(\mu) \end{cases}$$

Now, we need to prove the following:

Proposition 5.7. *For ρ, μ runs of $\mathcal{V}_p$, we have:*

$$\rho \trianglelefteq_p \mu \implies \rho \preceq_p \mu$$

Proof. We show this by induction on p. Let ρ and μ be two runs of $\mathcal{V}_p$ with $\rho \trianglelefteq_p \mu$. By definition, this means that we have $\rho = \rho_0 \xrightarrow{a_1} \rho_1 \dots \xrightarrow{a_k} \rho_k$ and $\mu = \mu_0 \xrightarrow{b_1} \mu_1 \dots \xrightarrow{b_\ell} \mu_\ell$ such that:

- $\rho_0 \trianglelefteq_{p-1} \mu_0$.
- $\rho_k \trianglelefteq_{p-1} \mu_\ell$.
- There exists a strictly increasing mapping φ from $\{1, \dots, k\}$ to $\{1, \dots, l\}$ such that $a_i = b_{\varphi(i)}$ and $\rho_i \trianglelefteq_{p-1} \mu_{\varphi(i)}$. Because we required $a_k = b_\ell$ and $\rho_k \trianglelefteq_{p-1} \mu_\ell$, we can assume that $\varphi(k) = \ell$.

We extend φ by taking $\varphi(0) = 0$. Let us consider the parts of the run μ that doesn't appear in ρ, i.e. for each $i \in \{0, \dots, k-1\}$, we define μ'_i, a run of $\mathcal{V}_p$ by:

$$\begin{aligned} \mu'_i = tgt(\mu_{\varphi(i)}) &\xrightarrow{b_{\varphi(i)+1}} \mu_{\varphi(i)+1} \\ &\dots \\ &\xrightarrow{b_{\varphi(i+1)-1}} \mu_{\varphi(i+1)-1} \end{aligned}$$

Note that μ'_i is non-empty because φ is strictly increasing. It can however be reduced to a single marking $tgt(\mu_{\varphi(i)}) = tgt(\mu_{\varphi(i+1)-1})$ if $\varphi(i+1) = \varphi(i) + 1$. This is the case when no transitions have been suppressed at this position.

We have $\rho_{i+1} \trianglelefteq_{p-1} \mu_{\varphi(i+1)}$ which by lemma 5.6 means that $src(\rho_{i+1}) \leq src(\mu_{\varphi(i+1)})$. As we also have $tgt(rho_i) \xrightarrow{a_{i+1}} src(\rho_{i+1})$ and $tgt(\mu_{\varphi(i+1)-1}) \xrightarrow{b_{\varphi(i+1)}} src(\mu_{\varphi(i+1)})$ with $a_{i+1} = b_{\varphi(i+1)}$, we have $tgt(\rho_i) \leq tgt(\mu_{\varphi(i+1)-1}) = tgt(\mu'_i)$. Morever, because $\rho_i \trianglelefteq_{p-1} \mu_{\varphi(i)}$, we have by lemma 5.6 $tgt(\rho_i) \leq tgt(\mu_{\varphi(i)}) = src(\mu'_i)$.

Now, let us consider $(r_i, s_i) \in \xrightarrow[\mathcal{V}_{p,\mu'_i}]{}$. By proposition 5.4, we get:

$$(src(\mu'_i) + r_i, tgt(\mu'_i) + s_i) \in \xrightarrow{A^*_{\leq p}}\!\!\!\!\!\rightarrow$$

But, because we have:

- $src(\mu'_i) = tgt(\mu_{\varphi(i)}) \geq tgt(\rho_i)$ (by hypothesis $\rho \trianglelefteq_p \mu$)
- $tgt(\mu'_i) = tgt(\mu_{\varphi(i+1)} - 1) \geq tgt(\rho_i)$ (by hypothesis $\rho \trianglelefteq_p \mu$)
- For all $j \in \{1, \dots, p\}$, $src(\mu'_i)(j) = tgt(\mu'_i)(j) = 0$ (because $b_{\varphi(i)+1}$ and $b_{\varphi(i+1)}$ are fired from these states, and these transition belong to A_p)

we get:

$$(tgt(\mu_{\varphi(i)}) - tgt(\rho_i) + r, tgt(\mu_{\varphi(i+1)-1}) - tgt(\rho_i) + s) \in \xrightarrow[\mathcal{V}_{p,[tgt(\rho_i)]}]{}$$

Now, we can consider a pair $(x, y) \in \xrightarrow[\mathcal{V}_{p,\mu}]{}$. We have:

$$
\begin{array}{rlll}
\xrightarrow[\nu_{p,\mu}]{} = \xrightarrow[\nu_{p-1,\mu_0}]{} & \circ \xrightarrow[\nu_{p,[tgt(\mu_0)]}]{} & \circ \xrightarrow[\nu_{p-1,\mu_1}]{} \\
& \circ \xrightarrow[\nu_{p,[tgt(\mu_1)]}]{} & \circ \xrightarrow[\nu_{p-1,\mu_2}]{} \\
& \circ \quad \dots & \circ \dots \\
& \circ \xrightarrow[\nu_{p,[tgt(\mu_{\ell-1}]}]{} & \circ \xrightarrow[\nu_{p-1,\mu_\ell}]{}
\end{array}
$$

But, by taking into account our previously defined runs μ'_i, we can redecompose our run in the following way (with the fact that production relations are transitive, which means that $\xrightarrow[\nu_{p-1,\mu_i}]{} \circ \xrightarrow[\nu_{p-1,[tgt(\mu_i)]}]{} = \xrightarrow[\nu_{p-1,\mu_i}]{}$):

$$
\begin{array}{rllll}
\xrightarrow[\nu_{p,\mu}]{} = & \xrightarrow[\nu_{p-1,\mu_0}]{} & \circ & \xrightarrow[\nu_{p,\mu'_0}]{} & \circ \\
& \xrightarrow[\nu_{p-1,\mu_{\varphi(1)}}]{} & \circ & \xrightarrow[\nu_{p,\mu'_1}]{} & \circ \\
& \dots & & & \\
& \xrightarrow[\nu_{p-1,\mu_{\varphi(k-1)}}]{} & \circ & \xrightarrow[\nu_{p,\mu'_{k-1}}]{} & \circ \\
& \xrightarrow[\nu_{p-1,\mu_{\varphi(k)}}]{} & & &
\end{array}
$$

This means there exists (r_i, s_i) such that:

$$
\begin{array}{lllll}
x & \xrightarrow[\nu_{p-1,\mu_0}]{} & r_0 & \xrightarrow[\nu_{p,\mu'_0}]{} & s_0 \\
& \xrightarrow[\nu_{p-1,\mu_{\varphi(1)}}]{} & r_1 & \xrightarrow[\nu_{p,\mu'_1}]{} & s_1 \\
& \dots & & & \\
& \xrightarrow[\nu_{p-1,\mu_{\varphi(k-1)}}]{} & r_{k-1} & \xrightarrow[\nu_{p,\mu'_{k-1}}]{} & s_{k-1} \\
& \xrightarrow[\nu_{p-1,\mu_{\varphi(k)}}]{} & y & &
\end{array}
$$

We have already shown that for all $i \in \{0, \dots, k-1\}$, $(r_i, s_i) \in \xrightarrow[\nu_{p,\mu'_i}]{}$ implies that $(tgt(\mu_{\varphi(i)}) - tgt(\rho_i) + r_i, tgt(\mu_{\varphi(i+1)} - 1) - tgt(\rho_i) + s_i) \in \xrightarrow[\nu_{p,[tgt(\rho_i)]}]{}$

By induction hypothesis, we also have that for all $i \in \{0, \dots, k\}$, $(s_{i-1}, r_i) \in \xrightarrow[\nu_{p-1,\mu_{\varphi(i)}}]{}$ implies that $(src(\mu_{\varphi(i)}) - src(\rho_i) + s_{i-1}, tgt(\mu_{\varphi(i)}) - tgt(\rho_i) + r_i) \in \xrightarrow[\nu_{p-1,\rho_i}]{}$.

Finally, because we have $tgt(\mu_{\varphi(i+1)} - 1) \xrightarrow{b_{\varphi(i)}} src(\mu_{\varphi(i+1)})$ and $tgt(\rho_i) \xrightarrow{a_i} src(\rho_{i+1})$ with $a_i = b_{\varphi(i)}$, it means we have $tgt(\mu_{\varphi(i+1)-1}) - tgt(\rho_i) = src(\mu_{\varphi(i+1)}) - src(\rho_{i+1})$.

Combining these three parts, we get:

$$
\begin{array}{rcl}
x + src(\mu_{\varphi(0)}) - src(\rho_0) & \xrightarrow[\nu_{p-1,\rho_0}]{} & r_0 + tgt(\mu_{\varphi(0)}) - tgt(\rho_0) \\
& \xrightarrow[\nu_{p,[tgt(\rho_0)]}]{} & s_0 + tgt(\mu_{\varphi(1)-1}) - tgt(\rho_0) = \\
s_0 + src(\mu_{\varphi(1)}) - src(\rho_1) & \xrightarrow[\nu_{p-1,\rho_1}]{} & r_1 + tgt(\mu_{\varphi(1)}) - tgt(\rho_1) \\
& \xrightarrow[\nu_{p,[tgt(\rho_1)]}]{} & s_1 + tgt(\mu_{\varphi(2)-1}) - tgt(\rho_1) = \\
& \dots & \\
s_{k-2} + src(\mu_{\varphi(k-1)}) - src(\rho_{k-1}) & \xrightarrow[\nu_{p-1,\rho_{k-1}}]{} & r_{k-1} + tgt(\mu_{\varphi(k-1)}) - tgt(\rho_{k-1}) \\
& \xrightarrow[\nu_{p,[tgt(\rho_{k-1})]}]{} & s_{k-1} + tgt(\mu_{\varphi(k)-1}) - tgt(\rho_{k-1}) = \\
s_{k-1} + src(\mu_{\varphi(k)}) - src(\rho_k) & \xrightarrow[\nu_{p-1,\rho_k}]{} & y + tgt(\mu_{\varphi(k)}) - tgt(\rho_k)
\end{array}
$$

We have shown $(x + src(\mu) - src(\rho), y + tgt(\mu) - tgt(\rho)) \in \xrightarrow[\mathcal{V}_{p,\rho}]{}$ which was what we wanted to demonstrate. ☐

Theorem 12. *$\preceq$ is a well-order on runs of $\mathcal{V}_p$.*

Proof. Let us consider an infinite sequence of runs of $\mathcal{V}_p$. Because $\trianglelefteq_p$ is a well-order on these runs, it means that we can extract an infinite increasing subsequence. By proposition 5.7, this is also an infinite increasing subsequence for $\preceq_p$. ☐

5.2.5 The production relations are asymptotically definable

The relations $\xrightarrow[\mathcal{V}_{p,\mu}]{}$ are finite compositions of relations $\xrightarrow[\mathcal{V}_{q,[m]}]{}$ for $q \leq p$ and $m \in 0^q \times \mathbb{N}^{d-q}$. To show that these relations are asymptotically definable, we first recall two results from [47]:

Lemma 5.8. *([47], Lemma 8.2)*
If R and R' are two asymptotically definable periodic relations, then $R \circ R'$ is an asymptotically definable periodic relation.

Theorem 13. *([47], Theorem 8.1)*
For $m \in \mathbb{N}^d$, $\xrightarrow[\mathcal{V}_{0,[m]}]{}$ is asymptotically definable.

Hence our aim is to generalize this last theorem to $\xrightarrow[\mathcal{V}_{p,[m]}]{}$ for any p and $m \in 0^p \times \mathbb{N}^{d-p}$

To do that, we will use theorem 9 on 80 that says that $\xrightarrow[\mathcal{V}_{p,[m]}]{}$ is asymptotically definable if and only if the following conic space is finitely generated for every vector space $V \subseteq \mathbb{Q}^d \times \mathbb{Q}^d$:

$$cl((\mathbb{Q}_{\geq 0} \xrightarrow[\mathcal{V}_{p,[m]}]{}) \cap V) = cl(\mathbb{Q}_{\geq 0}(\xrightarrow[\mathcal{V}_{p,[m]}]{} \cap V))$$

We take such a vector space V. We define $X = 0^p \times \mathbb{N}^{d-p}$ and $Y = (X \times X) \cap V$.

We will re-use the idea of Leroux' intraproductions but by restricting them to X. Let $Q_{m,V} = \{y \in X \mid \exists (x, z) \in (m, m) + Y,\ x \xrightarrow{A^*_{\leq p}} y \xrightarrow{A^*_{\leq p}} z\}$ and $I_{m,V} \subseteq \{0, \ldots, d-1\}$ by $i \in I_{m,V} \iff \{q(i) \mid q \in Q_{m,V}\}$ is infinite. Note that for $i \in \{0, \ldots, p-1\}$, $i \notin I_{m,V}$ as for all $q \in Q_{m,V}$, $q(i) = 0$.

An *intraproduction* for (m, Y) is a triple (r, x, s) such that $x \in X$ and $(r, s) \in Y$ with:

$$r \xrightarrow[\mathcal{V}_{p,[m]}]{} x \xrightarrow[\mathcal{V}_{p,[m]}]{} s$$

An intraproduction is *total* if $x(i) > 0$ for every $i \in I_{m,V}$. The following lemma can be proved exactly as Lemma 8.3 of [47]:

Lemma 5.9. *There exists a total intraproduction for (m, V_0).*

Proof. (This proof is a straightforward adaptation from [47])

Since finite sums of intraproductions are intraproductions, it is sufficient to prove that for every $i \in I_{m,V}$, there exists an intraproduction (r, x, s) for (m, Y) such that $x(i) > 0$. We fix $i \in I_{m,V}$.

We first prove that there exists $q, q' \in Q_{m,V}$ such that $q \leq q'$ and $q(i) < q'(i)$. Since $i \in I_{m,V}$, there exists an infinite sequence (q_n) of markings $q_n \in Q_{m,V}$ such that $(q_n(i))$ is strictly increasing. Since $\mathbb{N}^d$ is well-ordered, we can find $k < \ell$ such that $q_k \leq q_\ell$. As we also have $q_k(i) < q_\ell(i)$, we have our property.

So we consider these $q, q' \in Q_{m,V}$ with $q \leq q'$ and $q(i) < q'(i)$. As $q \in Q_{m,V}$, then there exists $(r, s) \in Y$ such that:

$$m + r \xrightarrow{A^*_{\leq p}} q \xrightarrow{A^*_{\leq p}} m + s$$

Symmetrically, as $q' \in Q_{m,V}$, there exists $(r', s') \in Y$ such that:

$$m + r' \xrightarrow{A^*_{\leq p}} q' \xrightarrow{A^*_{\leq p}} m + s'$$

As r, r', s and s' are in X, it means that adding one of these vectors will not prevent firing transition of $A_{\leq p}$. This means that from the previous transitions sequences, we can deduce:

- $(m + r') + r \xrightarrow{A^*_{\leq p}} q' + r$ from $m + r' \xrightarrow{A^*_{\leq p}} q'$.
- $q + ((q' - q) + r) \xrightarrow{A^*_{\leq p}} (m + s) + ((q' - q) + r)$ from $q \xrightarrow{A^*_{\leq p}} m + s$.
- $(m + r) + ((q' - q) + s) \xrightarrow{A^*_{\leq p}} q + ((q' - q) + s)$ from $m + r \xrightarrow{A^*_{\leq p}} q$.
- $q' + s \xrightarrow{A^*_{\leq p}} (m + s') + s$ from $q' \xrightarrow{A^*_{\leq p}} m + s'$.

By combining these parts, we get:

$$m + r + r' \xrightarrow{A^*_{\leq p}} m + s + r + (q' - q) \xrightarrow{A^*_{\leq p}} m + s + s'$$

which means that $(r + r', s + r + (q' - q), s + s')$ is the intraproduction we were looking for, with $(s + r + (q' - q))(i) > 0$. □

Given a finite set $I \subseteq \{0, \ldots, d-1\}$ and a marking $m \in \mathbb{N}^d$, we denote by m^I the vector of $\mathbb{N}^d_\omega$ defined by $m^I(i) = \omega$ if $i \in I$ and $m^I(i) = m(i)$ otherwise. We also define the order $\leq_\omega$ by $x \leq_\omega y$ if for all i, $y(i) = \omega$ or $x(i) = y(i)$ (equivalently, there exists $I \subseteq \{0, \ldots, d-1\}$ such that $x^I = y$). For a relation $\longrightarrow$ on $\mathbb{N}^d$ and $(x, y) \in \mathbb{N}^d_\omega$, we define $x \longrightarrow y$ if there exists $x', y' \in \mathbb{N}^d$ with $x' \longrightarrow y'$, $x' \leq_\omega x$ and $y' \leq_\omega y$.

Let $Q = \{q^{I_{m,V}} \mid q \in Q_{m,V}\}$ and $\mathcal{G}$ the complete directed graph with nodes Q such that the edge from q to q' is labeled by (q, q'). For $w \in (Q \times Q)^*$, we define $TProd(w) \subseteq \mathbb{N}^{A_{\leq p}}$ by:

$$\begin{array}{rcl} TProd(\varepsilon) & = & \{0^{A_{\leq p}}\} \\ TProd((q,q')) & = & \left\{ |u| \mid \exists (x,x') \in X \times X,\ x \leq_\omega q,\ x' \leq_\omega q',\ x \xrightarrow{A_p A^*_{<p}} x' \right\} \\ TProd(uv) & = & TProd(u) + TProd(v) \end{array}$$

We define the periodic relation $R_{m,V}$ on Y by $r\ R_{m,V}\ s$ if:

1. $r(i) = s(i) = 0$ for every $i \notin I_{m,V}$
2. there exists a cycle labeled by w in $\mathcal{G}$ on the state $m^{I_{m,V}}$ and $v \in TProd(w)$ such that $r + \delta(v) = s$.

Lemma 5.10. *For $(q,q') \in (Q \times Q)$, $TProd((q,q'))$ is Lambert.*

Proof. We first define the following Presburger sets (for $a \in A_p$):

$$\begin{array}{rcl} P(a) & = & \{(x', u, y) \in X \times \mathbb{N}^A \times X \mid \exists x \leq_\infty q,\ x \xrightarrow{a} x' \wedge y \leq_\infty q'\} \\ P' & = & \{(x, u, y) \in X \times \mathbb{N}^A \times X \mid x \leq_\infty q \wedge y \leq_\infty q'\} \end{array}$$

We also define $R = \{(x', v, y) \in \mathbb{N}^d \times \mathbb{N}^A \times \mathbb{N}^d \mid \exists u \in A^*,\ x' \xrightarrow{u} y \wedge |u| = v\}$. This is an almost semilinear set (corollary 5.2), which means that $R \cap P(a)$ and $R \cap P'$ are Lambert sets.

Now, we note that we have the following:

$$\begin{array}{rcl} TProd((q,q')) & = & \bigcup_{a \in A_p} |a| + \{u \mid \exists (x,y) \in \mathbb{N}^d \times \mathbb{N}^d,\ (x,u,y) \in R \cap P(a)\} \cup \\ & & \{u \mid \exists (x,y) \in \mathbb{N}^d \times \mathbb{N}^d,\ (x,u,y) \in R \cap P'\} \end{array}$$

By projection (proposition 5.1), we have that $\{u \mid \exists (x,y) \in \mathbb{N}^d \times \mathbb{N}^d,\ (x,u,y) \in R \cap P\}$ are Lambert sets for $P = P(a)$ or $P = P'$. Then, as the union of Lambert sets is Lambert, we have shown that $TProd((q,q'))$ is Lambert. □

Lemma 5.11. *The periodic relation $R_{m,V}$ is asymptotically definable.*

Proof. Let $P \subseteq \mathbb{N}^{Q \times Q}$ be the Parikh image of the language L made of words labeling cycles in $\mathcal{G}$ on the state $m^{I_{m,V}}$. L is a language recognized by a finite automaton, hence P is a Presburger set.

Now, let's show that $R'_{m,V} = \{TProd(w) \mid w \in L\}$ is a Lambert set. We have:

$$R'_{m,V} \quad = \quad \left\{ \sum_{a \in Q \times Q} v(a) \star TProd(a) \mid v \in P \right\}$$

P is Presburger, hence there exists $(d_i)_{1\leq i\leq p}$, $(e_{i,j})_{1\leq i\leq p, 1\leq j\leq n_i}$ with $d_i, e_{i,j} \in \mathbb{N}^{Q\times Q}$ and $P = \bigcup_i d_i + \Sigma_j \mathbb{N} e_{i,j}$. This gives:

$$\begin{aligned}
R'_{m,V} &= \bigcup_{1\leq i\leq p} \bigcup_{v\in\mathbb{N}^p} \sum_{1\leq j\leq n_i} \sum_{a\in Q\times Q} (d_i + v(j) * e_{i,j})(a) \star TProd(a) \\
&= \bigcup_{1\leq i\leq p} \sum_{a\in Q\times Q} d_i(a) \star TProd(a) + \bigcup_{1\leq i\leq p} \sum_{1\leq j\leq n_i} \bigcup_{k\in\mathbb{N}} \sum_{a\in Q\times Q} (k * e_{i,j})(a) \star TProd(a) \\
&= \bigcup_{1\leq i\leq p} \sum_{a\in Q\times Q} d_i(a) \star TProd(a) + \bigcup_{1\leq i\leq p} \sum_{1\leq j\leq n_i} \mathbb{N} \star \left(\sum_{a\in Q\times Q} e_{i,j}(a) \star TProd(a) \right)
\end{aligned}$$

For all $a \in Q\times Q$, we have seen that $TProd(a)$ is Lambert. So because Lambert sets are stable by addition, union and $\mathbb{N}\star$, (proposition 5.1), $R'_{m,V}$ is Lambert.

We define $V_{I_{m,V}} = \{x \in \mathbb{N}^d \mid \forall i \notin I_{m,V},\ x(i) = 0\}$ and $R''_{m,V} = \{(r, r + \delta(x)) \mid r \in V_{I_{m,V}} \wedge x \in R'_{m,V}\} = \{(r,r) \mid r \in V_{I_{m,V}}\} + \{0\}^d \times \delta(R'_{m,V})$. By proposition 5.1, we have $R''_{m,V}$ built from $R'_{m,V}$ by the image through a linear function and the sum with a Presburger set, which means $R''_{m,V}$ is Lambert. But, $R''_{m,V}$ is periodic, which means $R''_{m,V} = \mathbb{N} \star R''_{m,V}$ is asymptotically definable. Finally, as proposition 9, gives us that asymptotically definable sets are stable by intersection with vector spaces, $R_{m,V} = R''_{m,V} \cap V$ is asymptotically definable. $\square$

Now, we will show that our graph $\mathcal{G}$ is an accurate representation of the reachability relation:

Lemma 5.12. *Let w be the label of a path in $\mathcal{G}$ from $m_1^{I_{m,V}}$ to $m_2^{I_{m,V}}$ and $v \in TProd(w)$. Then, there exists u in $A^*_{\leq p}$ with $|u| = v$ and $(x,y) \in X \times X$, $x \leq_\omega m_1^{I_{m,V}}$ and $y \leq_\omega m_2^{I_{m,V}}$ such that $x \xrightarrow{u} y$.*

Proof. We show this by induction on the length of w. Let $w = w_0(q,q')$ where w_0 is a path from $m_1^{I_{m,V}}$ to $m_3^{I_{m,V}}$ and (q,q') is an edge from $m_3^{I_{m,V}}$ to $m_2^{I_{m,V}}$ and $v \in TProd(w_0(q,q'))$. This means there exists $v_1 \in TProd(w_0)$, $v_2 \in TProd(q,q')$ such that $v = v_1 + v_2$. By induction hypothesis, there exists $u_1 \in X \times X$, $x'_0 \leq_\infty m_1^{I_{m,V}}$ and $y'_0 \leq_\infty m_3^{I_{m,V}}$ such that $x'_0 \xrightarrow{u_1} y'_0$ and $|u_1| = v_1$.

By definition of $TProd((q,q'))$, as $v_2 \in TProd((q,q'))$, there exists $x'_1 \leq m_3^{I_{m,V}}$, $y'_1 \leq_\infty m_2^{I_{m,V}}$ and $u_2 \in A_p A^*_{<p} \cup A^*_{<p}$ such that $x'_1 \xrightarrow{u_2} y'_1$ and $|u_2| = v_2$. Let $z = max(y'_0, x'_1)$. We have $z(1) = y'_0(1) = x'_1(1) = m_3(1) = 0$, which gives us:

$$x'_0 + (z - y'_0) \xrightarrow{u_1} z \xrightarrow{u_2} y'_1 + (z - x'_1)$$

As $z^{I_{m,V}} = {y'_0}^{I_{m,V}} = {x'_1}^{I_{m,V}} = m_3^{I_{m,V}}$, we have $(z - y'_0) \leq_\infty 0^{I_{m,V}}$ and $(z - x'_1) \leq_\infty 0^{I_{m,V}}$, which allows us to define $x = x'_0 + (z - y'_0) \leq_\infty m_1^{I_{m,V}}$ and $y = y'_1 + (z - x'_1) \leq_\infty m_2^{I_{m,V}}$. $u = u_1 u_2$ completes the result. $\square$

We now show a lemma for the other direction:

Lemma 5.13. *Let $(m_1, m_2) \in Q_{m,V} \times Q_{m,V}$ with $u \in A^*_{\leq p}$ such that $m_1 \xrightarrow{u} m_2$. There exists $w \in (Q \times Q)^*$ label of a path from $m_1^{I_{m,V}}$ to $m_2^{I_{m,V}}$ such that $|u| \in TProd(w)$.*

Proof. Let $u = u_0 a_1 u_1 \dots a_n u_n$ with $u_i \in A^*_{<p}$ and $a_i \in A_p$. We define $(x_i)_{0 \leq i \leq n}$, $x_i \in X$ (x_i's for $i < n$ are in X because $a_i \in A_p$ transitions follow these states, and $x_n = m_2$ is in X by hypothesis) by:

$$m \xrightarrow{u_0} x_0 \xrightarrow{a_1 u_1} x_1 \xrightarrow{a_2 u_2} x_2 \cdots \xrightarrow{a_n u_n} x_n = m_2$$

We have for all i, $x_i \in X$, which leads that $|u_1| \in TProd((m_1^{I_{m,V}}, x_0^{I_{m,V}}))$ and for all $i \in \{0, \dots, n-1\}$, $|a_i u_i| \in TProd((x_i^{I_{m,V}}, x_{i+1}^{I_{m,V}}))$. Hence, we can get $|u| \in TProd(w)$ by defining $w = (m_1^{I_{m,V}}, x_0^{I_{m,V}})(x_0^{I_{m,V}}, x_1^{I_{m,V}}) \dots (x_{n-1}^{I_{m,V}}, m_2^{I_{m,V}})$. ☐

Thanks to lemmas 5.12 and 5.13, we can now prove the following lemma in the same way as lemma 8.5 of [47]:

Lemma 5.14. $cl(\mathbb{Q}_{\geq 0} R_{m,V}) = cl(\mathbb{Q}_{\geq 0}(\xrightarrow{\nu_{p,[m]}} \cap V))$

Proof. (This proof is a straightforward adaptation from [47])

Let us first prove the inclusion $\supseteq$. Let $(r, s) \in Y$ be such that $r \xrightarrow[\nu_{p,[m]}]{} s$. In this case, there exists a word $u \in A^*_{\leq p}$ such that $m + r \xrightarrow{w} m + s$. Observe that $m + n * r$ and $m + n * s$ are in $Q_{m,V}$ for every $n \in \mathbb{N}$. Hence, $r(i) > 0$ or $s(i) > 0$ implies $i \in I_{m,V}$ and we deduce that $(m + r)^{I_{m,V}} = (m + s)^{I_{m,V}} = m^{I_{m,V}}$. By lemma 5.13, because $m + r \xrightarrow{u} m + s$, there exists w label of a cycle on $m^{I_{m,V}}$ and such that $|u| \in TProd(w)$. As $r + \delta(|u|) = s$, we have proved that $(r, s) \in R_{m,V}$.

Now, let us prove the inclusion $\subseteq$. Let $(r, s) \in R_{m,V}$. In this case, $(r, s) \in Y$ satisfies $r(i) = s(i) = 0$ for every $i \notin I_{m,V}$ and there exists a word $w = a_1 \dots a_k$ with $a_i \in Q \times Q$, $v \in TProd(w)$ such that $r + \delta(v) = s$. By lemma 5.12, there exists $u \in A^*_{\leq p}$ with $|u| = v$, $r' \leq_\infty 0^{I_{m,V}}$, and $s' \leq_\infty 0^{I_{m,V}}$ such that $m + r' \xrightarrow{u} m + s'$. We consider a total intraproduction (r'', x, s'') for (m, Y). Because $r' \leq_\infty 0^{I_{m,V}}$, there exists $p \in \mathbb{N}$ such that $r' \leq p * x$. Because $r'(1) = x(1) = 0$, from $m + r' \xrightarrow{u} m + s'$, we get $m + p * x \xrightarrow{u} m + p * x + \delta(u)$. And as we also have $r(1) = 0$, we get:

$$m + p * x + r \xrightarrow{w'} m + p * x + r + \delta(w') = m + p * x + s$$

This means $(r, s) \in \xrightarrow{\nu_{p,[m']}}$ where $m' = m + p * x$. Since a production relation is periodic, we get for all $n \in \mathbb{N}$, $(n*r, n*s) \in \xrightarrow{\nu_{p,[m']}}$. As $(p*r'', p*x, p*s'')$ is an intraproduction for (m, Y), we have $m + p * r'' \xrightarrow{*} m' \xrightarrow{*} m + s''$. We deduce the relation $(m + p * r'') + n * r \xrightarrow{*} m' + n * r$ from $(m + p * r'') \xrightarrow{*} m'$ and the relation $m' + n * s \xrightarrow{*} (m + p * s'') + n * s$ from $m' \xrightarrow{*} (m + p * s'')$. We deduce that the following relation holds for every $n \in \mathbb{N}$:

$$m + p * r'' + n * r \xrightarrow{*} m + p * s'' + n * s$$

And as we have $(r'', s'') \in Y$ and $(r, s) \in Y$, we have $p * (r', s') + \mathbb{N} * (r, s) \subseteq \xrightarrow{\nu_{p,[m]}} \cap Y$. Thus $(r, s) \in cl(\mathbb{Q}_{\geq 0}(\xrightarrow{\nu_{p,[m]}} \cap Y))$. From the inclusion $R_{m,V} \subseteq cl(\mathbb{Q}_{\geq 0}(\xrightarrow{\nu_{p,[m]}} \cap Y))$ we get the inclusion $cl(\mathbb{Q}_{\geq 0} R_{m,V}) \subseteq cl(\mathbb{Q}_{\geq 0}(\xrightarrow{\nu_{p,[m]}} \cap Y))$. ☐

Finally, as $\overrightarrow{\nu_{p,\mu}}$ is a finite composition of elements of the form $\overrightarrow{\nu_{k,[m]}}$ for $j \leq p$, we have proven the following result:

Theorem 14. *If μ is a run of $\mathcal{V}_p$, then $\overrightarrow{\nu_{p,\mu}}$ is asymptotically definable.*

5.2.6 Decidability of reachability

We have now all the results necessary to prove theorem 11, by adapting the proof of theorem 9.1 of [47].

This problem is equivalent to prove that $\longrightarrow \cap((m,n)+D)$ is a Lambert relation for every $(m,n) \in \mathbb{N}^d \times \mathbb{N}^d$ and for every finitely generated periodic relation $D \subseteq \mathbb{N}^d \times \mathbb{N}^d$. We introduce the order $\leq^D$ over D defined by $x \leq_P x'$ if $x' \in x + D$. Because D is finitely generated, there exists $a_1, \ldots, a_q \in D$ such that $D = \mathbb{N}a_1 + \mathbb{N}a_2 + \ldots \mathbb{N}a_q$. Hence, if we define the surjective function f from $\mathbb{N}^q$ to D defined by $f(x) = \Sigma_i x(i)a_i$, we have $x \leq x' \implies f(x) \leq^P f(x')$, and because $\leq$ is a well-order on $\mathbb{N}^q$, $\leq^D$ is a well-order on D. We introduce the set $\Omega_{m,D,n}$ of runs μ such that $(src(\mu), tgt(\mu)) \in (m,n) + D$. Thanks to theorem 12, this set is well-ordered by the relation $\preceq_p^D$ defined by $\mu \preceq_p \mu'$ and $(src(\mu), tgt(\mu)) - (m,n) \leq^D (src(\mu'), tgt(\mu')) - (m,n)$. We deduce that $B = min_{\preceq_p^D}(\Omega_{m,D,n})$ is finite.

We now show the following equality:

$$\longrightarrow \cap((m,n)+P) = \bigcup_{\mu \in B} (src(\mu), tgt(\mu)) + (\overrightarrow{\nu_{p,\mu}} \cap P)$$

Let us first prove $\supseteq$. Let $\mu \in \Omega_{m,D,n}$. Proposition 5.4 shows that $(src(\mu), tgt(\mu)) + \overrightarrow{\nu_{p,\mu}} \in \xrightarrow{*}$. Since $(src(\mu), tgt(\mu)) \in (m,n) + D$ and D is periodic we deduce the inclusion $\subseteq$.

Now, let us prove $\subseteq$. Let (x', y') in the intersection $\xrightarrow{*} \cap((m,n)+D)$. There exists a run $\mu' \in \Omega_{m,D,n}$ such that $x' = src(\mu')$ and $y' = tgt(\mu')$. There exists $\mu \in min_{\preceq_p^D}(\Omega_{m,P,n})$ such that $\mu \preceq_p^D \mu'$. We deduce that $(x', y') \in (src(\mu), tgt(\mu)) + (\overrightarrow{\nu_{p,\mu}} \cap D)$ and we have proved the inclusion $\subseteq$.

Theorem 14 shows that $\overrightarrow{\nu_{p,\mu}}$ is an asymptotically definable relation. As P is a finitely generated relation, it is also asymptotically definable. Asymptotically definable relations are stable by finite intersections ([48], Lemma 4.5) and we deduce that $\overrightarrow{\nu_{p,\mu}} \cap D$ is asymptotically definable. This induces that $\xrightarrow{*} \cap((m,n)+P)$ is a Lambert relation for every $(m,n) \in \mathbb{N}^d \times \mathbb{N}^d$ and for every finitely generated periodic relation $D \subseteq \mathbb{N}^d \times \mathbb{N}^d$. Therefore, $\xrightarrow{A^*_{\leq p}}\!\!\!\!\rightarrow$ is almost semilinear.

Because $\left(\xrightarrow{A_p A^*_{<p} \cup A^*_{<p}}\right)^* = \xrightarrow{A^*_{\leq p}}$, we can now apply theorem 10 and get:

Proposition 5.15. *If X and Y are two Presburger sets such that $\xrightarrow{A^*_{\leq p}} \cap (X \times Y) = \emptyset$, then there exists a Presburger $\xrightarrow{A_p A^*_{<p} \cup A^*_{<p}}$-forward invariant X' with $X' \cap Y = \emptyset$.*

Now that we have shown the existence of such an invariant, we only need to show that we are able to test whether a given set is an invariant:

Proposition 5.16. *Whether a Presburger set X is a $\xrightarrow{A_pA^*_{<p}\cup A^*_{<p}}$-forward invariant is decidable.*

Proof. X is a forward invariant for $\xrightarrow{A_pA^*_{<p}\cup A^*_{<p}}$ if and only if $\xrightarrow{A_p}(X) \subseteq X$ and $\xrightarrow{A^*_{<p}}(X) \subseteq X$. Because $\xrightarrow{A_p}(X)$ is a Presburger set, the first condition is decidable as the inclusion of Presburger sets, and the second reduces to deciding whether $\xrightarrow{A^*_{<p}} \cap (X \times \mathbb{N}^d \backslash X)$ is empty, which is an instance of the reachability problem in a VAS with $p-1$ hierarchical zero-tests, which is decidable by induction hypothesis. ☐

This allows us to conclude:

Theorem 15. *Reachability in VAS_{0^*} is decidable.*

Proof. By the propositions 5.15 and 5.16, reachability is co-semidecidable by enumerating Presburger forward invariants, and semidecidability is clear. ☐

5.3 Cloverability in VAS with hierachical zero-tests

If COVERABILITY can be easily obtained from REACHABILITY, there is no known reduction from CLOVERABILITY to REACHABILITY. We show here how to use the well-orders defined above to show the decidability of this problem.

In order to do that, we show that we can see a VAS with hierarchical zero-tests as some kind of WSTS:

Definition 5.5. *Given $\mathcal{V} = \langle A_0, ..., A_p, \delta\rangle$ a VAS_{0^*} of dimension d, we define $wsts_k(\mathcal{V}) = \langle 0^k \times \mathbb{N}_\omega^{d-k}, \overline{A_k A^*_{<k}}\rangle \cup \overline{A^*_{<k}}$ where $\overline{a_1 \dots a_k} = \overline{a_1}.\overline{a_2}\dots\overline{a_k}$.*

Proposition 5.17. *If $\mathcal{V}$ is a $VAS_{0^*}p$, for $k \le p$, $wsts_k(\mathcal{V})$ is a complete WSTS with decidable* POST MEMBERSHIP.

Proof. First, let's show that $wsts_k\mathcal{V}$ is a WSTS. Let's take $x, y, x' \in 0^k \times \mathbb{N}_\omega^{d-k}$ and $u \in A^*_{\le k}$ such that $x \xrightarrow{u}\!\!\!\!\rightarrow_{lts(\mathcal{V})} x'$ and $y \ge x$. We have $x' - x \in 0^k \times \mathbb{N}^{d-k}$. Because we are only increasing the counters that can not be tested for zero, we get $x' \xrightarrow{u} y + x' - x$ and $wsts_k(\mathcal{V})$ is a WSTS.

Let's now show completeness. As [29] already explained that $0^p \times \mathbb{N}_\omega^{d-p}$ is a cdcwo, we only have to show continuity, i.e. that for any downward closed set $Y \subseteq 0^d \times \mathbb{N}_\omega^{d-p}$, we have $Lim \downarrow Post_\mathcal{S}(Y) = Lim \downarrow Post_\mathcal{S}(Lim\ Y)$. As $Y \subseteq Lim\ Y$, the direction $\subseteq$ is immediate, so let's look at the other direction and take $x \in Post_\mathcal{S}(Lim\ Y)$. We have a sequence $y_i \in Y$, $y \in Lim\ Y$ and $u \in A^*$ such that $lim\ y_i = y$ and $y \xrightarrow{u}\!\!\!\!\rightarrow x$. As $lim\ y_i = y$, there exists $i \in \mathbb{N}$ such that for all $j \ge i$, u is fireable from y_j. Let x_j be such that $y_j \xrightarrow{u} x_j$. This leads to $x \in Lim \downarrow Post_\mathcal{S}(Y)$, and as we have shown have $Post_\mathcal{S}(Lim\ Y) \subseteq Lim \downarrow Post_\mathcal{S}(Y)$, we obtain by closure $Lim \downarrow Post_\mathcal{S}(Lim\ Y) \subseteq Lim \downarrow Post_\mathcal{S}(Y)$.

Now, we consider the problem of POST MEMBERSHIP and reduce it to reachability in VAS_{0^*} with $p-1$ hierarchical zero-tests. We have $x, y \in 0^k \times \mathbb{N}_\omega^{d-k}$ and we want to know if there exists $z \in Post_{ts(\mathcal{V})}(x, A_k \cup \{\varepsilon\})$, $u \in A^*_{\leq k-1}$ and $y' \in 0^k \times \mathbb{N}_\omega^{d-k}$ such that $y \leq y'$ and $z \xrightarrow{u} y'$. To check that, we define $\mathcal{V}'$ similar to $\mathcal{V}$ but with additionnal transition into A_0 that can decrease each component of index in $\{k+1, \ldots, d\}$. Then, for z fixed, the previous problem reduces to reachability of y in $\mathcal{V}'$. Reachability in such a VAS_{0^*} is decidable, by theorem 15 (or [55]) □

We write WSTS-VASZH(K) the class of WSTS obtained by this construction from any VAS_{0^*} with k hierarchical zero-tests.

This transformation of the original LTS of a VAS_{0^*} into a WSTS doesn't lose any information required to compute the cover of the original LTS. Indeed, we have the following:

Proposition 5.18. *Let $\mathcal{V}$ be a VAS_{0^*} with p hierarchical zero-tests and $x \in 0^p \times \mathbb{N}_\omega^{d-p}$. We define recursively:*

$$\begin{aligned} C_{p+1} &= \{x\} \\ C_k &= Cover_{wsts_k(\mathcal{S})}(C_{k+1}) \end{aligned}$$

Then, we have $C_0 = Cover_{ts(\mathcal{S})}(x)$.

Proof. Let us first prove $C_0 \subseteq Cover_{ts(\mathcal{V})}(x)$. We take $y_0 \in C_0$. This means there exists for $i \in \{1, \ldots, p\}$, $y_i \in X$, such that (considering $x = y_{p+1}$), $y_i \in Cover_{wsts_i(\mathcal{V})}(y_{i+1})$. Because of monotony, we can build by induction for $i \in \{0, \ldots, p+1\}$ the sequence $x_{p+1} = y_{p+1}$, and $x_i \in Reach_{wsts_i(\mathcal{V})}(x_{i+1})$ with $x_i \geq y_i$. Because $Reach_{wsts_i(\mathcal{V})}(x_{i+1}) \subseteq Reach_{ts(\mathcal{V})}(x_{i+1})$, this gives the existence of $x_0 \geq y_0$ such that $x_0 \in Reach_{ts(\mathcal{V})}(x)$.

To show the other direction, we consider $y \in Cover_{ts(\mathcal{V})}(x)$ and a run γ such that $src(\gamma) = x$ and $tgt(\gamma) \geq y$. We decompose this run by $\gamma = \gamma_p \gamma_{p-1} \ldots \gamma_0$ where γ_i fulfills:

- $tgt(\gamma_i) \in 0^i \times \mathbb{N}_\omega^{d-i}$
- For $i < p$, γ_i doesn't visit any state in $0^{i+1} \times \mathbb{N}_\omega^{d-i-1}$.

Note that the second item means that the transitions inside γ_i live inside $A_{\leq i}$. Hence, $tgt(\gamma_i) \in C_0$ and this concludes our proof. □

This means that in order to show that CLOVER SET is computable for VAS_{0^*}, we only need to show that this is the case for WSTS-VASZH(K). Moreover, to prove that, thanks to proposition 3 and theorem 2, we will use an acceleration strategy that will be sufficient to solve this problem.

5.3.1 Duality of the backward and forward algorithm

The next sections may be a bit abstract, as we are using the powerful theorem 2 to obtain directly the computability of CLOVER SET. We present here an other view of the proof (restricted to VAS with one sero-test like presented in [17]), as a dual of the algorithm deciding coverability in VAS with one zero-test presented by Abdulla and Mayr in [7]. Let us describe in few words the idea behind this algorithm (each step is obtained by using the previous step has an oracle):

Coverability [7]	Cloverability [17]
Decide REACHABILITY for $wsts_0(\mathcal{V})$	Decide REACHABILITY for $wsts_0(\mathcal{V})$
Decide PRED MEMBERSHIP for $wsts_1(\mathcal{V})$	Decide POST MEMBERSHIP for $wsts_1(\mathcal{V})$
Compute PRED BASIS for $wsts_1(\mathcal{V})$ (using directly membership and proposition 2.3 of page 33)	Compute POST BASIS for $wsts_1(\mathcal{V})$ (by completing the *Post* set, and taking maximal elements)
Run the backward procedure of WSTS on $wsts_1(\mathcal{V})$	Run the Karp-Miller procedure of WSTS on $wsts_1(\mathcal{V})$

Indeed, as explained in more depth in chapter 2, one can look at the Karp-Miller algorithm as the dual of the backward algorithm of WSTS by replacing upward closed set by downward closed sets. However, the properties of downward closed sets are less desirable than those of upward closed sets: in this particular case, one can not obtain directly POST BASIS from POST MEMBERSHIP, because POST BASIS is only defined by taking the completion of the *Post*-set, which means that one must be able to test membership of the limit elements, and not of only the "normal" elements. The next sections aim to show that one can test such a membership by using a recursively enumerable set of possible witnesses.

5.3.2 Productive sequences

We reuse here the well-order $\trianglelefteq_p$ defined in section 5.2.4 in order to define accelerations. The idea is that if we have $\rho \trianglelefteq_p \rho'$, then ρ' is obtained from ρ by inserting some additionnal transitions. We show that iterating these additionnal transitions is a "good" acceleration.

We write $run(x_0, u)$ the run $x_0 \xrightarrow{a_0} x_1 \xrightarrow{a_1} \cdots \xrightarrow{a_n} x_n$ when $u = a_0 \dots a_n$. Formally, if we have two runs $run(x, u)$ and $run(y, v)$ of a VAS $\mathcal{V}_p$ with p hierarchical zero-tests with $run(x, u) \trianglelefteq_p run(y, v)$, we want to define $run(x, u)\ \nabla_p\ run(y, v) \in ((A^*_{\leq p})^{|u|+1}$ that will express the loops that can be inserted between the letters of u ($|u| - 1$ positions). We will build this operator by recursion on p:

- For $\rho = x_0 \xrightarrow{a_1} x_1 \xrightarrow{a_2} \cdots \xrightarrow{a_m} x_m$ and $\mu = y_0 \xrightarrow{b_1} y_1 \xrightarrow{b_2} \cdots \xrightarrow{b_n} y_n$ runs with 0 hierarchical zero-tests with $\rho \trianglelefteq_0 \mu$, we consider $\varphi : \{1, \dots, m\} \to \{1, \dots, n\}$ the strictly increasing mapping induced by the word embedding order, that we extend by $\varphi(0) = 0$. Because we have $x_m \leq y_n$ and $a_m = b_n$, we also require that $\varphi(m) = n$. Then, we define:

$$
\begin{aligned}
\rho\ \nabla_0\ \mu = \quad & (b_{\varphi(0)+1} b_{\varphi(0)+2} \dots b_{\varphi(1)-1}, \\
& b_{\varphi(1)+1} b_{\varphi(1)+2} \dots b_{\varphi(2)-1}, \\
& \dots \\
& b_{\varphi(m-1)+1} b_{\varphi(m-1)+2} \dots b_{\varphi(m)-1})
\end{aligned}
$$

- For $\rho = \rho_0 \xrightarrow{a_1} \rho_1 \xrightarrow{a_2} \cdots \xrightarrow{a_m} \rho_m$ and $\mu = \mu_0 \xrightarrow{b_1} \mu_1 \xrightarrow{b_2} \cdots \xrightarrow{b_n} \mu_n$ runs with p hierarchical zero-tests such that ρ_i, μ_i are runs with $p - 1$ hierarchical zero-tests and

$a_i, b_i \in A_p$ with $\rho \trianglelefteq_p \mu$, we consider $\varphi : \{1, \ldots, m\} \to \{1, \ldots, n\}$ the strictly increasing mapping induced by the word embedding order, that we extend by $\varphi(0) = 0$. Because we have $\rho_m \trianglelefteq_{p-1} \mu_n$ and $a_m = b_n$, we also require that $\varphi(m) = n$. We recall that $acts(\rho)$ denotes the sequence of actions appearing in ρ. Then, we define:

$$\begin{aligned}
\rho \nabla_p \mu \quad = \quad & (\rho_0 \ \nabla_{p-1} \ \mu_{\varphi(0)}) \\
& (b_{\varphi(0)+1} acts(\mu_{\varphi(0)+1}) \ldots b_{\varphi(1)-1} acts(\mu_{\varphi(1)-1}) \\
& (\rho_1 \ \nabla_{p-1} \ \mu_{\varphi(1)}) \\
& (b_{\varphi(1)+1} acts(\mu_{\varphi(1)+1}) \ldots b_{\varphi(2)-1} acts(\mu_{\varphi(2)-1}) \\
& \quad \ldots \\
& (b_{\varphi(m-1)+1} acts(\mu_{\varphi(m-1)+1}) \ldots b_{\varphi(m)-1} acts(\mu_{\varphi(m)-1}) \\
& (\rho_m \ \nabla_{p-1} \ \mu_{\varphi(m)})
\end{aligned}$$

Let $\mathcal{V}_p$ be a VAS with p hierarchical zero-tests with $x \in \mathbb{N}_\omega^d$ and $u \in A_{\leq p}^*$. If there exists ρ run of $\mathcal{V}_p$ such that $run(x, u) \trianglelefteq_p \rho$ and $src(\rho) = x$, we say that $\Delta = run(x, u) \nabla_p \rho$ is productive for $\mathcal{V}_p$, x and u. In this case, for $k \in \mathbb{N}$ and assuming $u = a_0 a_1 \ldots a_n$ and $\Delta = (v_1, \ldots, v_n)$, we define $acc(u, \Delta, k) \in A_{\leq p}^*$ by:

$$acc(u, \Delta, n) = a_0 v_1^k a_1 v_2^k \ldots v_n^k a_n$$

Note that if Δ was obtained from the $run(x, u) \trianglelefteq run(x, u')$ inequality with u and u' fireable from x, we have $acc(u, \Delta, 1) = u'$ and hence is fireable. Actually, in this case, we will show in proposition 5.20 below that all $acc(u, \Delta, k)$ are fireable from x. But first, we state a very simple lemma (which can be seen as an instance of proposition 5.3) that shows the condition necessary to iterate a sequence:

Lemma 5.19. *Let $\mathcal{V}$ be a VAS with p hierarchical zero-tests, $x \in \mathbb{N}_\omega^{d-p}$ and ρ be a run of $\mathcal{V}$ with:*

- $src(\rho) - x \in 0^p \times \mathbb{N}_\omega^{d-p}$
- $tgt(\rho) - x \in 0^p \times \mathbb{N}_\omega^{d-p}$

Then, we have:

$$x + p * (src(\rho) - x) \xrightarrow{acts(\rho)^p} y + p * (tgt(\rho) - y)$$

Proposition 5.20. *Let $\mathcal{V}$ be a VAS with p hierarchical zero-tests, $x, x' \in \mathbb{N}_\omega^d$, $u, u' \in A_{\leq p}^*$ be two transitions sequences fireable respectively from x and x' such that $run(x, u) \trianglelefteq_p run(x', u')$. We define $\Delta = run(x, u) \nabla_p run(x', u')$*

If $y \in \mathbb{N}_\omega^d$ and $y' \in \mathbb{N}_\omega^d$ are such that $x \xrightarrow{u} y$ and $x' \xrightarrow{u'} y'$, then for any $n \in \mathbb{N}$, we have:

$$x + k * (x' - x) \xrightarrow{acc(u,\Delta,k)} y + k * (y' - y)$$

Proof. This will be shown by induction on p. For $p = -1$, we consider the empty transition system $\mathcal{V}_{-1}$ that has only runs reduced to a single state, and for which the result is immediate.

We take $\mathcal{V}_p$ a VAS with p hierarchical zero-tests, $x \in \mathbb{N}_\omega^d$ and $u \in A_{\leq p}^*$ fireable from x. We consider a productive sequence $\Delta = (v_1, \ldots, v_n)$. This means there exists $u' \in A_{\leq p}^*$ fireable from x with $run(x, u) \trianglelefteq_p run(x, u')$. This means we have:

$$\begin{array}{rcl} run(x,u) & = & \rho_0 a_1 \rho_1 a_2 \ldots a_m \rho_m \\ run(x,u') & = & \mu_0 b_1 \mu_1 b_2 \ldots b_n \mu_n \end{array}$$

with $a_i, b_i \in A_p$ and ρ_i, μ_i runs of $\mathcal{V}_{p-1}$ the VAS with $p-1$ hierarchical zero-tests obtained from $\mathcal{V}_p$ by removing the A_p transitions. We have also φ, a strictly increasing mapping from $\{0, \ldots, m\}$ to $\{0, \ldots, n\}$ with $\varphi(0) = 0$, $\varphi(m) = n$, $\forall 0 \leq i \leq m.\ \rho_i \trianglelefteq_{p-1} \mu_{\varphi(i)}$ and $\forall 1 \leq i \leq m.\ a_i = b_{\varphi(i)}$.

Because we have $\rho_i \trianglelefteq_{p-1} \mu_{\varphi(i)}$, we define $\Delta_i = \rho_i \nabla_{p-1} \mu_{\varphi(i)}$. By induction hypothesis, we have:

$$\forall k \in \mathbb{N}.\ src(\rho_i) + k * (src(\mu_{\varphi(i)}) - src(\rho_i)) \xrightarrow{acc(acts(\rho_i),\Delta_i,k)} tgt(\rho_i) + k * (tgt(\mu_{\varphi(i)}) - tgt(\rho_i))$$

Moreover, for any $i \in \{0, \ldots, m-1\}$, we have:

- $tgt(\rho_i) \leq tgt(\mu_{\varphi(i)})$ (by lemma 5.6)
- $tgt(\rho_i) \leq tgt(\mu_{\varphi(i+1)-1})$ from $src(\rho_{i+1}) \leq src(\mu_{\varphi(i+1)})$ (again by lemma 5.6) and $a_{i+1} = b_{\varphi(i+1)}$
- $tgt(\rho_i), tgt(\mu_{\varphi(i+1)-1} \in 0^p \times \mathbb{N}_\omega^d$ because $a_{i+1}, b_{\varphi(i+1)} \in A_p$ are fired after these states.
- $tgt(\rho_i), tgt(\mu_{\varphi(i)}) \in 0^p \times \mathbb{N}_\omega^d$ because $a_{i+1}, b_{\varphi(i)+1} \in A_p$ are fired after these states.

This means the hypothesis of lemma 5.19 apply to μ'_i defined by:

$$\mu'_i = tgt(\mu_{\varphi(i)}) b_{\varphi(i)+1} \mu_{\varphi(i)+1} b_{\varphi(i)+2} \mu_{\varphi(i)+2} \ldots b_{\varphi(i+1)-1} \mu_{\varphi(i+1)-1}$$

By applying this lemma, we get:

$$\forall k \in \mathbb{N}.\ tgt(\rho_i) + k * (tgt(\mu_{\varphi(i)}) - tgt(\rho_i)) \xrightarrow{acts(\mu'_i)^n} tgt(\rho_i) + k * (tgt(\mu_{\varphi(i+1)-1}) - tgt(\rho_i))$$

Finally, we note that because $a_i = b_{\varphi(i)}$, we have $tgt(\mu_{\varphi(i+1)-1}) - tgt(\rho_i) = src(\mu_{\varphi(i+1)}) - src(\rho_{i+1})$ which gives:

$$\forall k \in \mathbb{N}.\ tgt(\rho_i) + k * (tgt(\mu_{\varphi(i+1)-1}) - tgt(\rho_i)) \xrightarrow{a_1} src(\rho_{i+1}) + k * (src(\mu_{\varphi(i+1)}) - src(\rho_{i+1}))$$

But as we have $\Delta = \Delta_0 acts(\mu'_0) \Delta_1 \ldots acts(\mu'_{m-1}) \Delta_m$, we can put the pieces together and get:

$$
\begin{array}{rcl}
src(\rho_0) + k * (src(\mu_{\varphi(0)}) - src(\rho_0)) & \xrightarrow{acc(acts(\rho_0),\Delta_0,k)}\!\!\!\!\twoheadrightarrow & tgt(\rho_0) + k * (tgt(\mu_{\varphi(0)}) - tgt(\rho_0)) \\
& \xrightarrow{acts(\mu_0')^n}\!\!\!\!\twoheadrightarrow & tgt(\rho_0) + k * (tgt(\mu_{\varphi(1)-1}) - tgt(\rho_0)) \\
& \xrightarrow{a_1} & src(\rho_1) + k * (src(\mu_{\varphi(1)}) - src(\rho_1)) \\
& \xrightarrow{acc(acts(\rho_1),\Delta_1,k)}\!\!\!\!\twoheadrightarrow & tgt(\rho_1) + k * (tgt(\mu_{\varphi(1)}) - tgt(\rho_1)) \\
& \xrightarrow{acts(\mu_1')^k}\!\!\!\!\twoheadrightarrow & tgt(\rho_1) + k * (tgt(\mu_{\varphi(2)-1}) - tgt(\rho_1)) \\
& \xrightarrow{a_2} & src(\rho_2) + k * (src(\mu_{\varphi(2)}) - src(\rho_2)) \\
& \dots & \\
& \xrightarrow{a_m} & src(\rho_m) + k * (src(\mu_{\varphi(m)}) - src(\rho_m)) \\
& \xrightarrow{acc(acts(\rho_m),\Delta_m,k)}\!\!\!\!\twoheadrightarrow & tgt(\rho_m) + k * (tgt(\mu_{\varphi(m)}) - tgt(\rho_m))
\end{array}
$$

Finally, we note that by definition of acc, we have:

$$
\begin{array}{rlll}
acc(acts(\rho), \Delta, k) = acc(acts(\rho_0), \Delta_0, k) & acts(\mu_0')^k & a_1 & acc(acts(\rho_1), \Delta_1, k) \\
& acts(\mu_1')^k & a_2 & acc(acts(\rho_2), \Delta_2, k) \\
& \dots & & \dots \\
& acts(\mu_m')^k & a_m & acc(acts(\rho_m), \Delta_m, k)
\end{array}
$$

This gives us that:

$$
src(\rho_0) + k * (src(\mu_{\varphi(0)}) - src(\rho_0)) \xrightarrow{acc(acts(\rho),\Delta,k)}\!\!\!\!\twoheadrightarrow tgt(\rho_m) + k * (tgt(\mu_{\varphi(m)}) - tgt(\rho_m))
$$

Because $src(\rho_0) = src(\rho)$, $src(\mu_{\varphi(0)}) = src(\mu)$, $tgt(\rho_m) = tgt(\rho)$ and $tgt(\mu_{\varphi(m)}) = tgt(\mu)$, that concludes the demonstration. □

δ is extended to $\Delta \in (A^*_{\leq q})^{<\omega}$ by $\delta(v_0, v_1, \dots, v_n) = \sum \delta(v_i)$. This allows us to define the following set of functions (not an acceleration strategy because it doesn't fulfill $h(x) \geq x$):

Definition 5.6. *Given $\mathcal{S} \in$ VASZH(P), for $q \leq p$, we define:*

$$
\text{PROD(Q)} = \{f_{u,\Delta} \mid u \in A^*_{\leq q}, \Delta \in ((A_{\leq q})^*)^{<\omega}\}
$$

by:

$$
\begin{array}{rcl}
dom(f_{u,\Delta}) & = & \{x \in \mathbb{N}^d_\omega \mid \Delta \textit{ is productive for } \mathcal{V}, x, u\} \\
f_{u,\Delta}(x) & = & x + \omega * \delta(\Delta)
\end{array}
$$

A straightforward corollary of proposition 5.20 is that all these functions stay inside *Lim Cover*:

Proposition 5.21. *Let $\mathcal{S} \in$ WSTS-VASZH(P) and $x \in 0^p \times \mathbb{N}^{d-p}_\omega$. For every $f \in$ PROD(P), $f(x) \in Lim\ Cover_{\mathcal{S}}(x)$.*

Moreover, this set of functions allow to reach any limit element of the cover of our system:

Proposition 5.22. *Let* $\mathcal{S} \in$ WSTS-VASZH(P) *and* $x \in 0^p \times \mathbb{N}_\omega^{d-p}$. *For every* $y \in Lim\ Cover_{\mathcal{S}}(x)$, *there exists* $f \in$ PROD(P) *such that* $f(x) = y$.

Proof. Let $y \in Lim\ Cover_{\mathcal{S}}(x)$. There exists a sequence $(\rho_k)_{k\in\mathbb{N}}$ with $\forall k \in \mathbb{N}.\ src(\rho_k) = x$ and $y \leq lub\ \{tgt(\rho_k) \mid k \in \mathbb{N}\}$. We assume that x doesn't contain any ω's, as this would just mean some components are ignored in the system (as explained in section 2.3.2). This means that for all k, $tgt(\rho_k) \in \mathbb{N}^d$, and by extracting a subsequence of (ρ_k) we can require that for any $i \in \{0, \ldots, d-1\}$:

- $y(i) = \omega \implies tgt(\rho_k)(i)$ is strictly increasing.
- $y(i) < \omega \implies \forall k \in \mathbb{N}.\ tgt(\rho_k)(i) = y(i)$.

Because $\trianglelefteq_p$ is a well-order, there exists $k < k'$ such that $\rho_k \trianglelefteq_p \rho_{k'}$. Let $\Delta = \rho_k \nabla_p \rho_{k'}$. By definition, Δ is productive for x, so we can pick $f_{acts(\rho_k),\Delta}$ and we have $x \in dom(f_{acts(\rho_k),\Delta})$. Moreover, $\delta(\Delta) = tgt(\rho_{k'}) - tgt(\rho_k)$, so we have $f_{acts(\rho_k),\Delta}(x) = x + \omega * \delta(\Delta) = y$. ☐

5.3.3 Decidability of the cover

By theorem 2, it is enough to show that CLOVERABILITY is semi-decidable and POST MEMBERSHIP is decidable. Propositions 5.21 and 5.22 show that CLOVERABILITY is semi-decidable by enumerating the functions inside PROD(P). As we have also shown in proposition 5.17 that POST MEMBERSHIP is decidable, we get:

Theorem 16. CLOVER SET *is computable for* VAS_{0^*}

5.4 Other problems

We show here the decidability of a few other problems on VAS_{0^*}, which are mainly consequences of the two previous sections.

5.4.1 Repeated Control State Reachability and LTL

We present here a proof of the decidability of REPEATED CONTROL STATE REACHABILITY (equivalent to REPEATED COVERABILITY as described in section 1.6.5). This requires the addition of control states to Vector Addition Systems with hierarchical zero-tests. This is done similarly as for Vector Addition Systems (definition 1.7 on page 18). All results that were shown for Vector Addition Systems with hierarchical zero-tests stay true with the addition of control states.

In the basic Vector Addition System with States, REPEATED CONTROL STATE REACHABILITY was shown equivalent to the presence of a specific increasing loop (see [25]). We define here a similar notion. Let $\leq_p$ be the order defined by $x \leq_p y$ if $x \leq y$ and for $i \in \{0, \ldots, p-1\}$, $x(i) = y(i)$. Then, we define:

Definition 5.7. *We consider a Vector Addition System with States and hierarchical zero-tests. Let $(q, x) \in Q \times \mathbb{N}^d$. A word $u \in A^*$ is a $\leq_p$-increasing loop on (q, x) if:*

- $u \in A^+_{\leq p}$.
- $(q, x) \in dom(\overline{u})$ *and* $\overline{u}(q, x) \geq_p (q, x)$.

With this definition, we proceed in three steps:

- We first show that q can be repeatedly covered from (q_0, x_0) if there exists a $\leq_p$-increasing loop (for some $p \leq d$) on $(q, x) \in Cover_{\mathcal{S}}(x)$.
- Then, we show that given $y \in \mathbb{N}^d_\omega$, one can decide whether there is a $\leq_p$-increasing loop on (q, x) with $x \in (\downarrow y) \cap \mathbb{N}^d$.
- We conclude by noticing that given $q \in Q$, by proposition 16, we can compute a finite set $B \subseteq \mathbb{N}^d_\omega$ such that $\{x \in \mathbb{N}^d \mid (q, x) \in Cover_{\mathcal{S}}(q_0, x_0)\} = (\downarrow B) \cap \mathbb{N}^d$

Lemma 5.23. *Let $\mathcal{S} \in$ VASZH(D), $x_0 \in \mathbb{N}^d$ and $q_0, q \in Q$. If q can be repeatedly covered from (q_0, x_0), then there exists $(q, x) \in Cover_{\mathcal{S}}(q_0, x_0)$, $p \leq d$ and a $\leq_p$-increasing loop on (q, x).*

Proof. Let $(q_0, x_0) \xrightarrow{a_1} (q_1, x_1) \cdots \xrightarrow{a_k} (q_k, x_k) \cdots$ be an infinite run covering q infinitely often. Let p be the highest number such that $\{k \in \mathbb{N} \mid a_k \in A_p\}$ is infinite. By extracting a suffix, we can consider that all $a_k \in A_{\leq p}$. Then, because $\mathbb{N}^d$ is well-ordered, one can find a strictly increasing mapping $\varphi : \mathbb{N} \to \mathbb{N}$ such that for all $k \in \mathbb{N}$:

- $(q_{\varphi(k)}, x_{\varphi(k)}) \leq (q_{\varphi(k+1)}, x_{\varphi(k+1)})$
- $a_{\varphi(k)+1} \in A_p$

Because $a_{\varphi(k)+1} \in A_p$, this means that $x_{\varphi(k)} \in 0^p \times \mathbb{N}^{d-p}$. As q is covered infinitely often, we can find a run from $x_{\varphi(k)}$ to $x_{\varphi(k+k')}$ such that q is covered in an intermediate state. Let $u \in A^*_{leqp}$ and $v \in A^*_{\leq p}$ such that $(q_{\varphi(k)}, x_{\varphi(k)}) \xrightarrow{u} (q, x) \xrightarrow{v} (q_{\varphi(k)}, x_{\varphi(k+k')})$. Then, because $x_{\varphi(k)} \leq_p x_{\varphi(k+k')}$, by monotony, there exists $x' \geq x$ such that $(q, x) \xrightarrow{v} (q_{\varphi(k)}, x_{\varphi(k+k')}) \xrightarrow{u} (q, x')$. We have shown the existence of our loop. ☐

We now define precisely our problem of existence of an increasing loop:

Decision Problem:	INITIALIZED INCREASING LOOP
Input:	$\mathcal{S} \in$ VASZH(D)
	$p \leq d$
	$y \in \mathbb{N}^d_\omega$
	$q \in Q$
Question:	does there exists $x \in (\downarrow y) \cap \mathbb{N}^d$,
	$u \in A^+_{\leq p}$ such that $(q, x) \leq_p \overline{u}(q, x)$?

and we will show that it is decidable by reduction to reachability:

Lemma 5.24. INITIALIZED INCREASING LOOP *is decidable.*

Proof. (In this proof, ω^m describes the vector with m ω's. Also, we recall that e_i denotes the vector whose all components are equal to 0 except the i-th one which is equal to 1.)

Let us take $x \in 0^p \times \mathbb{N}_\omega^{d-p}$. Without loss of generality (by reordering counters), we have $x = (0^p, \omega^m, b)$, $b \in \mathbb{N}^n$, with $d = p + m + n$.

We will build a VAS_{0^*} $\mathcal{V}'$ inducing a transition system $\mathcal{S}'$ that will mimic $\mathcal{S}$ in the following sense (x represents a state $\mathcal{S}$, and x' the associated state in $\mathcal{S}'$):

- The counters that can be tested for zero are preserved.
- A counter $x(i)$ for $p \leq i \leq p + m - 1$ (the ones for which $x(i) = \omega$) is replaced by two counters $x'(i)$ and $x'(i+m)$, such that $x'(i) - x'(i+m) \leq x(i)$. This simulates a lossy counter that can go arbitrarily below its initial value.
- A counter $x(i)$ for $p+m \leq i \leq p+m+n-1$ (the ones for which $x(i) < \omega$) is replaced by one counter $x'(i+m)$ such that $x'(i+m) \leq x(i)$. This simulates a lossy counter.

Note that states of $\mathcal{S}$ will be represented as (q, t, v, z) with $q \in Q$, $t \in \mathbb{N}^p$, $v \in \mathbb{N}^m$ and $z \in \mathbb{N}^n$ while states of $\mathcal{S}'$ will be represented as (q, t, v, w, z) with $q \in Q$, $t \in \mathbb{N}^p$, $(v, w) \in \mathbb{N}^m \times \mathbb{N}^m$ and $z \in \mathbb{N}^n$.

Formally, we define $\mathcal{V}' = \langle A'_0, A_1, \ldots, A_p, \delta', tr\rangle$ of dimension $d' = p + 2m + n$ by:

$$
\begin{aligned}
A'_0 &= A_0 \cup \\
&\quad \{leak_{q,i} \mid p+1 \leq i \leq p+m \land q \in Q\} \cup \\
&\quad \{syncadd_{q,i} \mid p+1 \leq i \leq p+m \land q \in Q\} \cup \\
&\quad \{syncsub_{q,i} \mid p+1 \leq i \leq p+m \land q \in Q\} \cup \\
&\quad \{leak_{q,i} \mid p+2m+1 \leq i \leq p+2m+n \land q \in Q\} \\
\delta'(a) &= (x, v, 0, w) \qquad \text{for } a \in A_{\leq p} \text{ and } \delta(a) = (x, v, w) \\
\delta'(leak_i) &= -e_i \\
\delta'(syncadd_i) &= -e_i - e_{i+m} \\
\delta'(syncsub_i) &= +e_i + e_{i+m} \\
tr'(a) &= tr(a) \qquad \text{for } a \in A_{\leq p} \\
tr'(a_{q,i}) &= (q, q) \qquad \text{for } a \in \{leak, syncadd, syncsub\}
\end{aligned}
$$

The transition of $A_{\leq p}$ are translated by simply affecting the counters that are supposed to simulate $\mathcal{S}$ ones, while transitions that perform more zero-tests than p are discarded. The $leak_i$ transitions makes some counters lossy, and finally the $syncadd_i$ and $syncsub_i$ transitions imply that only the relative value of the counters i and $i+m$ matters (for $p \leq i \leq p+m-1$). This simulates a counter living in $\mathbb{Z}$.

We will show that the existence of $x, x' \in 0^p \times \mathbb{N}^{d-p}$ and $u \in A_{\leq p}^+$ such that $x \leq (0^p, \omega^m, b)$, $x \leq_p x'$ and $x \xrightarrow{u}\!\!\!\!\rightarrow_{\mathcal{S}} x'$ is equivalent to the reachability in $\mathcal{S}'$ of $(q, 0^p, 0^m, 0^m, b)$ from itself using at least one transition in $A_{\leq p}$. Note that the class of Vector Addition System with States and hierarchical zero-tests is stable by product with a finite automata, so this question of reachability with respect to the regular expression $A'^* A_{leqp} A'^*$ reduces to basic reachability.

- $\Rightarrow$ Let us assume the existence of $x, x' \in 0^p \times \mathbb{N}^{d-p}$ and $u \in A^+$ such that $x \leq y$, $(q,x) \xrightarrow{u} (q,x')$ and $x \leq x'$.

  Let $x = (0^p, \alpha_1, \beta_1)$, $x' = (0^p, \alpha_2, \beta_2)$ with $\alpha_1 \leq \alpha_2$, $\beta_1 \leq \beta_2$ and $\beta_1 \leq b$.

  Because $(q, 0^p, \alpha_1, \beta_1) \xrightarrow{u}_{\mathcal{S}} (q, 0^p, \alpha_2, \beta_2)$, we have $(q, 0^p, \alpha_1, \alpha_1, \beta_1) \xrightarrow{u}_{\mathcal{S}'} (q, 0^p, \alpha_2, \alpha_1, \beta_2)$. Moreover, because $\beta_1 \leq b$, we also have that $(q, 0^p, \alpha_1, \alpha_1, b) \xrightarrow{leak^* syncadd^*}_{\mathcal{S}'} (q, 0^p, \alpha_2, \alpha_2, \beta_2 + b - \beta_1)$.

  Then, we have :

$$
\begin{aligned}
(q, 0^p, 0^m, 0^m, b) &\xrightarrow{syncadd^*}_{\mathcal{S}'} (q, 0^p, \alpha_1, \alpha_1, b) \\
&\xrightarrow{u}_{\mathcal{S}'} (q, 0^p, \alpha_1, \alpha_2, b + \beta_2 - \beta_1) \\
&\xrightarrow{leak^*}_{\mathcal{S}'} (q, 0^p, \alpha_1, \alpha_1, b) \\
&\xrightarrow{syncsub^*}_{\mathcal{S}'} (q, 0^p, 0^m, 0^m, b)
\end{aligned}
$$

- $\Leftarrow$ Assume that we have $(q, 0^p, 0^m, 0^m, b) \xrightarrow{u}_{\mathcal{S}'} (q, 0^p, 0^m, 0^m, b)$. We will show there exist $x, x' \in 0^p \times \mathbb{N}^{d-p}$ with $x \leq y$, $x \leq x'$ and such that $x \xrightarrow{u'}_{\mathcal{S}} x'$ with u' is obtained from u by deleting all letters that are not in $A_{\leq p}$.

  Let $(q_i, t_i, v_i, w_i, z_i)_{0 \leq i \leq k}$ be a sequence such that:

  - For $i \in \{0, \dots, k-1\}$, $(q_i, t_i, v_i, w_i, z_i) \longrightarrow_{\mathcal{S}'} (q_{i+1}, t_{i+1}, v_{i+1}, w_{i+1}, z_{i+1})$.
  - $(q_0, t_0, v_0, w_0, z_0) = (q_k, t_k, v_k, w_k, z_k) = (q, 0^p, 0^m, 0^m, b)$.

  Let $\alpha \in \mathbb{N}^m$ be the vector defined by, for $i \in \{1, \dots, m\}$, $\alpha(i) = max\ \{w_j(i) \mid 0 \leq j \leq k\}$. We define γ from $\mathbb{N}^{d+m}$ to $\mathbb{N}^d$ by $\gamma(t, v, w, z) = (t, v - w + \alpha, z)$. Then, an easy induction on the length of the transition sequence gives that:

$$(q_1, s_1) \xrightarrow{u}_{\mathcal{S}'} (q_2, s_2) \implies \exists s_2' \in \mathbb{N}^d, s_2' \geq_p \gamma(s_2) \wedge (q_1, \gamma(s_1)) \xrightarrow{u'}_{\mathcal{S}} (q_2, s_2')$$

  This gives the result.

□

And now, we can just conclude:

Theorem 17. REPEATED CONTROL STATE REACHABILITY *is decidable for Vector Addition Systems with States and hierarchical zero-tests.*

Proof. We consider a Vector Addition System with States and hierarchical zero-tests with $x_0 \in \mathbb{N}^d$ and $q_0, q \in Q$. By lemma 5.23, q is repeatedly coverable from (q_0, x_0) if and only if there exists $p \leq d$ such that there exists $(q, y) \in Max\ Lim\ Cover_{\mathcal{S}}(x)$, and $x \leq y$ such that there is a $\leq_p$-increasing loop on (q, x).

The set of $y \in \mathbb{N}_\omega^d$ such that $(q, y) \in Max\ Lim\ Cover_{\mathcal{S}}(x)$ is finite and computable (theorem 16). Thus, REPEATED CONTROL STATE REACHABILITY reduces to a finite number of instances of INITIALIZED INCREASING LOOP which are decidable by lemma 5.24. □

Proposition 1.9 on page 24 allows us to also get:

Theorem 18. LTL MODEL CHECKING *is decidable for Vector Addition Systems with hierarchical zero-tests.*

5.4.2 Regularity

To finish our overview on problems for VAS_{0^*}, we study the following problem:

Decision Problem:	REGULARITY
Input:	a LTS $\mathcal{S} = \langle X, A, \longrightarrow \rangle \in \mathbf{S}$
	$x \in X$
Question:	is $L_t(\mathcal{S}, x)$ regular?

This problem is known to be decidable for simply-labelled Vector Addition Systems, but not if we allow labelling functions [37, 62]. We will show that regularity is still decidable for simply-labelled VAS_{0^*} by a simple reduction to regularity for Vector Addition Systems. Indeed, given a VAS_{0^*} $\mathcal{V}$, we will prove that either:

- The zero-tests are really used (for a notion of usage to be defined), and the language of traces can't be regular, or,
- The zero-tests aren't really used, and we can define a simply-labelled VAS $\mathcal{V}'$ with the same language of traces.

Let $\mathcal{V} = \langle A_0, A_1, \ldots, A_d, \delta \rangle$ and $x \in \mathbb{N}^d$. For $p \leq d$, we define:

$$\Omega_p = \{\alpha \in \mathbb{N}^p \mid \exists \beta \in \mathbb{N}^{d-p}.\ \exists y \in \mathbb{N}^d.\ x \longrightarrow\!\!\!\!\rightarrow (\alpha, \beta) \longrightarrow\!\!\!\!\rightarrow A^* A_p\}$$

If Ω_p is bounded, it means that the zero-tests inside A_p aren't really used: because it is only used after bounded behaviour of the counters it is supposed to test, one can encode these counters in control states. Thus, we have the following lemma:

Lemma 5.25. *Let Ω_p be finite for every p. Then one can build a VAS $\mathcal{V}' = \langle A, \delta' \rangle$ of dimension d and $x' \in Q \times \mathbb{N}^d$ such that $L_t(ts(\mathcal{V}'), x')$ is regular if and only if $L_t(ts(\mathcal{V}), x)$ is regular.*

Proof. Let $\Omega = \bigcup_{1 \leq p \leq d} \Omega_p$. We define a VASS $\mathcal{V}' = \langle Q, A', \delta, tr \rangle$ (note that tr is a function from $A' = Q \times A$ to $Q \times Q$) by:

$$\begin{aligned} Q &= \Omega \cup \{vasmode\} \\ A' &= A \times Q \\ dom(tr) &= \{(q, a) \mid a \in A_p, q \in \Omega \text{ s.t. } \forall i \in \{0, \ldots, p-1\}.\ q(i) = 0\} \cup \\ &\quad\ \{vasmode\} \times A_0 \\ tr(q, a) &= \begin{cases} (q, q + a) & \text{if } q \in \Omega \text{ and } q + \delta(a) \in \Omega \\ (q, vasmode) & \text{otherwise} \end{cases} \end{aligned}$$

We have two categories of control states: as long as we are in the "predecessors" of the zero-tests Ω, we keep this information to be able to know when to fire a zero-test. When we get outside this set, we know that we can only fire normal transitions, so we don't need this information any more and we get into the control state *vasmode*. This makes $\mathcal{V}'$ simulate faithfully $\mathcal{V}$:

$$\begin{aligned} L_t(ts(\mathcal{V}), x) &= \varphi(L_t(\mathcal{V}), (x, x)) \\ \psi(L_t(ts(\mathcal{V})) &= L_t(\mathcal{V}, (x, x)) \end{aligned}$$

where:

- φ is the morphism $(Q \times A)^* \to A^*$ defined by:

$$\varphi(q, a) = a$$

- ψ is the transducer $A^* \to (Q \times A)^*$ using states Q defined by:

$$q \xrightarrow{a?\ (q,a)!}_{\psi} \begin{cases} (q, q + a) & \text{if } q \in \Omega \text{ and } q + \delta(a) \in \Omega \\ (q, vasmode) & \text{otherwise} \end{cases}$$

Because regular languages are stable by morphisms and regular transduction, we get our result. □

Lemma 5.26. *Let Ω be infinite. Then, $L_t(\mathcal{S}, x)$ is not regular.*

Proof. Let p such that Ω_p is infinite. Ω_p is infinite, so because our transition is finite-branching, by Konig's lemma, there exists an infinite run ρ $x_0 \xrightarrow{u_1} x_1 \xrightarrow{u_2} \cdots \xrightarrow{u_k} x_k \cdots$ such that:

- For all $k \in \mathbb{N}$, there exists $v_k \in A^*A_p$ and $y_k \in \mathbb{N}^d$ such that $x_k \xrightarrow{v_k} y_k$.
- For all $k \in \mathbb{N}$, $x_k(0, \ldots, p-1) < x_{k+1}(0, \ldots, p-1)$ (which implies $\delta(u_k)(0, \ldots, p-1) > 0$)

Let us assume that $L_t(\mathcal{S}, x)$ is regular. $\{u_1 u_2 \ldots u_{k-1} u_k v_k \mid k \in \mathbb{N}\} \subseteq L_t(\mathcal{S}, x)$, so by the pumping lemma, there exists $k, \ell \in \mathbb{N}$ with $k \leq \ell$ such that for all $n \in \mathbb{N}$:

$$u_1 \ldots u_{k-1}(u_k \ldots u_\ell)^n v_\ell \in L_t(\mathcal{S}, x)$$

But $\delta(u_k \ldots u_\ell)(0, \ldots, p-1) > 0$, which contradicts the fact that $v_\ell \in A^*A_p$. □

Theorem 19. REGULARITY *is decidable for VAS_{0^*}.*

Proof. One can decide whether Ω is bounded by reduction to CLOVER SET. Then, lemmas 5.25 and 5.26 allow to conclude. □

5.5 A summary of results on VAS with hierarchical zero-tests

We can now sum up the various results known on VAS with hierarchical zero-tests. Results marked by [*] are those presented of this chapter, or that can be directly derived from them.

	VAS	VAS + 1 zero-test	VAS + hierarchical
COVERABILITY	yes [41],[6],[33],...	yes [55],[7],[*]	yes [55],[*]
CLOVERABILITY	yes [41],[6],[33],...	yes [*]	yes [*]
REACHABILITY	yes [49],[42],[47]	yes [55],[*]	yes [55],[*]
BOUNDEDNESS	yes [41],[6],[33],...	yes [32],[*]	yes [*]
TERMINATION	yes [41],[6],[33],...	yes [32],[*]	yes [*]
LTL (ON ACTIONS)	yes [25],[39]	yes [*]	yes [*]
LTL (ON STATES)	no [25]	no [25]	no [25]
REGULARITY (UNLABELED)	yes [62]	yes [*]	yes [*]
REGULARITY (WITH LABELING)	no [62]	no [62]	no [62]

We believe this provide a comprehensive survey of problems that are decidable on VAS with hierarchical zero-tests. It is interesting to note that despite looking (at least syntaxically) much closer to Minsky machines than VAS, VAS with hierarchical zero-tests enjoy exactly the same decidability properties. However, some open problems remain:

- The complexity of these problem on VAS with hierarchical zero-tests are totally unknown, apart from the lower bound of EXPSPACE derived from coverability for VAS [19]. It can be shown that they are all as hard as reachability on VAS, and given the complexity of this problem is an important problem that is still open (and difficult), it may take time before getting any meaningful result. Extending the Rackoff proof seems difficult, given that this proof relies heavily on the fact that the value of a counter doesn't matter once it gets big enough. A possibility would be to bound the running time of the algorithms derived from the proofs presented here by the works of Schnoebelen *et al.* [27, 60], but that would yield horrendous upper bound, way above the primitive recursive hierarchy.

- Given the large similarity (from a decidability point of view) between VAS and VAS with hierarchical zero-tests, it would be interesting to find a significant result that would either show their difference, or that they have an underlying structure that make them basically the same. A study of their reachability languages show that $L = \{a^n b^n \mid n \in \mathbb{N}\}$ can be recognized by both, but L^* needs one zero-test. However, this can hardly be called a major difference.

- A larger model that includes VAS with hierarchical zero-test would be VAS with an added stack (by [11]). Trying to extend the proofs presented in this chapter to VAS with a stack is tempting, especially given that there seems to be an order on runs similar to the one used in this chapter, but this leads to some complications that are not obvious to resolve.

Conclusion

We have shown various decidability results on extensions of Vector Addition Systems and we can now complete the partial view presented in section 1.8:

	VAS	transfers			resets			abstract	zero-tests	
		1	2	≥ 3	1	2	≥ 3	data	1	hier.
reachability	yes	yes	no	no	yes	no	no	no	yes	yes
coverability	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
termination	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes*
boundedness	yes	yes	yes	yes	yes	yes	no	yes	yes	yes*
place-boundedness	yes	yes*	yes*	no	yes*	yes*	no	no	yes*	yes*
repeated coverability	yes	yes*	no	no	yes*	no	no	no	yes*	yes*

These results were based on two main ideas:

- We can extend the works of Finkel and Goubbault [29, 30] to be able to compute the cover by using interesting "patterns" of the system (like the one generated by the well-order of section 5.2, or the one presented by Dufourd *et al.* in [24])
- The proof of reachability of Leroux [47] uses a well-order that is similar to the one that can found in VAS with hierarchical zero-tests.

On top of these decidability results, we presented a quick analysis of expressiveness questions for extensions of Vector Addition Systems. We argued that the following rule of thumb can be followed: "If the state spaces differ, the expressiveness probably does". We didn't try to compare systems with the same state space, as we believed it would be hard to derive a general idea. However, it should be reasonably easy to prove (for example) that VAS with resets are more expressive than VAS. We left that as an exercice to the reader.

Some interesting questions remain, that we will study in further work:

- The gap between the only known lower bound (EXPSPACE by [19]) and the ones that could be obtained by the works of Schnoebelen *et al.* [27, 60] is huge. Finding better results would be extremely interesting.
- It is notable that chapter 5 uses a good deal of WSTS theory, despite VAS with hierarchical zero-tests not being monotonic. It would be interesting to see if the theory of WSTS can actually be extended to a wider class of systems, some of them displaying non-monotonic behaviour. A prime candidate for this would be VAS with a stack, as a generalization of hierarchical zero-tests [11].

Acknowledgements

Bibliography

[1] P. Abdulla, A. Bouajjani, J. Cederberg, F. Haziza, and A. Rezine. Monotonic abstraction for programs with dynamic memory heaps. In A. Gupta and S. Malik, editors, *Computer Aided Verification*, volume 5123 of *Lecture Notes in Computer Science*, pages 341–354. Springer Berlin / Heidelberg, 2008.

[2] P. Abdulla, J. Cederberg, and T. Vojnar. Monotonic abstraction for programs with multiply pointed structures. In *Reachability Problems*, 2011.

[3] P. Abdulla, G. Delzanno, and A. Rezine. Approximated parameterized verification of infinite-state processes with global conditions. *Formal Methods in System Design*, 34:126–156, 2009.

[4] P. Abdulla, G. Delzanno, and L. Van Begin. Comparing the expressive power of well-structured transition systems. In J. Duparc and T. Henzinger, editors, *Computer Science Logic*, volume 4646 of *Lecture Notes in Computer Science*, pages 99–114. Springer Berlin / Heidelberg, 2007.

[5] P. Abdulla, G. Delzanno, and L. Van Begin. A language-based comparison of extensions of petri nets with and without whole-place operations. In A. Dediu, A. Ionescu, and C. Martín-Vide, editors, *Language and Automata Theory and Applications*, volume 5457 of *Lecture Notes in Computer Science*, pages 71–82. Springer Berlin / Heidelberg, 2009.

[6] P. A. Abdulla, K. Cerans, B. Jonsson, and Y.-K. Tsay. General decidability theorems for infinite-state systems. In *11th IEEE Symp. on Logic in Computer Science, LICS'96*, pages 313–321, New Brunswick, New Jersey, 1996. IEEE Computer Society Press.

[7] P. A. Abdulla and R. Mayr. Minimal cost reachability/coverability in priced timed Petri nets. In L. de Alfaro, editor, *3rd Int. Conf. on Foundations of Software Science and Computation Structures, FoSSaCS'09*, volume 5504 of *Lect. Notes Comp. Sci.*, pages 348–363, York, UK, 2009. Springer.

[8] P. A. Abdulla, K. Čerans, B. Jonsson, and Y.-K. Tsay. Algorithmic analysis of programs with well quasi-ordered domains. *Inform. Comput.*, 2000.

[9] S. Abramsky and A. Jung. Domain theory. 1994.

[10] R. Alur, C. Courcoubetis, and T. Henzinger. The observational power of clocks. In *CONCUR*, 1994.

[11] M. F. Atig and P. Ganty. Approximating petri net reachability along context-free traces. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2011, December 12-14, 2011, Mumbai, India*, volume 13 of *LIPIcs*, pages 152–163. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.

[12] B. Berthomieu, M. Boyer, and M. Diaz. Time petri nets. In *Petri Nets*, pages 123–161. Wiley Online Library.

[13] M. Blockelet and S. Schmitz. Model checking coverability graphs of vector addition systems. In *Proceedings of MFCS'11*, volume 6907, pages 108–119, 2011.

[14] R. Bonnet. Decidability of LTL model checking for vector addition systems with one zero-test. In G. Delzanno and I. Potapov, editors, *5th Workshop on Reachability Problems, RP'11*, volume 6945 of *Lect. Notes Comp. Sci.*, pages 85–95, Genova, Italy, 2011. Springer.

[15] R. Bonnet. The reachability problem for vector addition systems with one zero-test. In F. Murlak and P. Sankowski, editors, *36th Mathematical Foundations of Computer Science, MFCS'11*, volume 6907 of *Lect. Notes Comp. Sci.*, pages 145–157, Warsaw, Poland, 2011. Springer.

[16] R. Bonnet, A. Finkel, S. Haddad, and F. Rosa-Velardo. Ordinal theory for expressiveness of wsts. 2011.

[17] R. Bonnet, A. Finkel, J. Leroux, and M. Zeitoun. Place-boundedness for vector addition systems with one zero-test. In K. Lodaya and M. Mahajan, editors, *30th IARCS Conf. on Foundations of Software Technology and Theoretical Computer Science, FSTTCS'10*, volume 8 of *Leibniz International Proceedings in Informatics*, pages 192–203, Chennai, India, 2010. Leibniz-Zentrum für Informatik.

[18] R. Bonnet, A. Finkel, and M. Praveen. Extending the rackoff technique to affine nets. In *FSTTCS'12*, 2012. (to appear).

[19] E. Cardoza, R. Lipton, and A. R. Meyer. Exponential space complete problems for petri nets and commutative semigroups. In *Proceedings of the eighth annual ACM symposium on Theory of computing*, STOC '76, pages 50–54, New York, NY, USA, 1976. ACM.

[20] D. H. J. de Jongh and R. Parikh. Well partial orderings and hierarchies. 80:195–207, 1977.

[21] S. Demri. On selective unboundedness of VASS. In *Proceedings of INFINITY'10*, pages 1–15, 2010.

[22] L. Doyen and J.-F. Raskin. Antichain algorithms for finite automata. In J. Esparza and R. Majumdar, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 6015 of *Lecture Notes in Computer Science*, pages 2–22. Springer Berlin / Heidelberg, 2010.

[23] C. Dufourd, A. Finkel, and P. Schnoebelen. Reset nets between decidability and undecidability. In K. Larsen, S. Skyum, and G. Winskel, editors, *Automata, Languages and Programming*, volume 1443 of *Lecture Notes in Computer Science*, pages 103–115. Springer Berlin / Heidelberg, 1998. 10.1007/BFb0055044.

[24] C. Dufourd, P. Jančar, and Ph. Schnoebelen. Boundedness of reset P/T nets. In J. Wiedermann, P. van Emde Boas, and M. Nielsen, editors, *26th Int. Colloquium on Automata, Languages and Programming, ICALP'99*, volume 1644 of *Lect. Notes Comp. Sci.*, pages 301–310, Prague, Czech Republic, 1999. Springer.

[25] J. Esparza. On the decidability of model checking for several μ-calculi and Petri Nets. In S. Tison, editor, *Trees in Algebra and Programming, CAAP'94*, volume 787 of *Lect. Notes Comp. Sci.*, pages 115–129. Springer, 1994.

[26] J. Esparza. Decidability and complexity of Petri Net problems: An introduction. In W. Reisig and G. Rozenberg, editors, *Lectures on Petri Nets I: Basic Models*, volume 1491 of *Lect. Notes Comp. Sci.*, pages 374–428. Springer, 1998.

[27] D. Figueira, S. Figueira, S. Schmitz, and P. Schnoebelen. Ackermann and primitive-recursive bounds with dickson's lemma. Technical Report arXiv:1007.2989, Jul 2010.

[28] A. Finkel. Reduction and covering of infinite reachability trees. *Information and Computation*, 89(2):144–179, 1990.

[29] A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, part I: Completions. In S. Albers and J.-Y. Marion, editors, *26th Symp. on Theoretical Aspects of Computer Science, STACS'09*, pages 433–444. Springer, 2009.

[30] A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, Part II: Complete WSTS. In S. Albers, A. Marchetti-Spaccamela, Y. Matias, S. E. Nikoletseas, and W. Thomas, editors, *36th Int. Colloquium on Automata, Languages and Programming, ICALP'09*, volume 5556 of *Lect. Notes Comp. Sci.*, pages 188–199. Springer, 2009.

[31] A. Finkel, P. McKenzie, and C. Picaronny. A well-structured framework for analysing Petri Net extensions. *Inf. Comput.*, 195(1-2):1–29, 2004.

[32] A. Finkel and A. Sangnier. Mixing coverability and reachability to analyze VASS with one zero-test. In D. Peleg and A. Muscholl, editors, *SOFSEM'10*, volume 5901 of *LNCS*, pages 394–406. Springer, 2010.

[33] A. Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theoret. Comput. Sci.*, 256(1–2):63–92, 2001.

[34] P. Gastin and D. Oddoux. Fast LTL to Büchi automata translation. In G. Berry, H. Comon, and A. Finkel, editors, *13th Int. Conf. on Computer Aided Verification, CAV'01*, volume 2102 of *Lect. Notes Comp. Sci.*, pages 53–65. Springer, 2001.

[35] G. Geeraerts, J.-F. Raskin, and L. Van Begin. Well-structured languages. *Acta Informatica*, 44:249–288, 2007. 10.1007/s00236-007-0050-3.

[36] R. Gerth, D. Peled, M. Y. Vardi, and P. Wolper. Simple on-the-fly automatic verification of linear temporal logic. In *Proceedings of the Fifteenth IFIP WG6.1 International Symposium on Protocol Specification, Testing and Verification XV*, pages 3–18, London, UK, UK, 1996. Chapman & Hall, Ltd.

[37] A. Ginzburg and M. Yoeli. Vector addition systems and regular languages. *Journal of Computer and System Sciences*, 20(3):277 – 284, 1980.

[38] J. Goubault-Larrecq. On a generalizationof a result by Valk and Jantzen. Technical report, LSV, ENS de Cachan, 2009.

[39] P. Habermehl. On the complexity of the linear-time μ-calculus for Petri nets. In P. Azéma and G. Balbo, editors, *Application and Theory of Petri Nets 1997*, volume 1248 of *Lect. Notes Comp. Sci.*, pages 102–116. Springer, 1997.

[40] J. E. Hopcroft and J.-J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoret. Comput. Sci.*, 8:135–159, 1979.

[41] R. M. Karp and R. E. Miller. Parallel program schemata. *J. Comput. System Sci.*, 2:147–195, 1969.

[42] S. R. Kosaraju. Decidability of reachability in vector addition systems (preliminary version). In *14th ACM Symp. on Theory of Computing, STOC'82*, pages 267–281, New York, NY, USA, 1982. ACM.

[43] O. Kouchnarenko and S. Ph. A formal framework for the analysis of recursive-parallel programs. In *PaCT'97*, 1997.

[44] J. B. Kruskal. The theory of well-quasi-ordering: A frequently discovered concept. *Journal of Combinatorial Theory, Series A*, 13(3):297 – 305, 1972.

[45] R. Lazic, T. Newcomb, J. Ouaknine, A. W. Roscoe, and J. Worrell. Nets with tokens which carry data. In *Proceedings of the 28th international conference on Applications and theory of Petri nets and other models of concurrency*, ICATPN'07, pages 301–320, Berlin, Heidelberg, 2007. Springer-Verlag.

[46] J. Leroux. The general vector addition system reachability problem by Presburger inductive invariants. In *24th IEEE Symp. on Logic in Computer Science, LICS'09*, pages 4–13. IEEE Computer Society Press, 2009.

[47] J. Leroux. Vector addition system reachability problem: A short self-contained proof. In A. H. Dediu, S. Inenaga, and C. Martín-Vide, editors, *Language and Automata Theory and Applications - 5th International Conference, LATA 2011, Tarragona, Spain, May 26-31, 2011. Proceedings*, volume 6638 of *Lecture Notes in Computer Science*, pages 41–64. Springer, 2011.

[48] J. Leroux. Vector addition system reachability problem: a short self-contained proof. In *38th ACM Symp. on Principles of Programming Languages, POPL'11*, pages 307–316. ACM, 2011.

[49] E. W. Mayr. An algorithm for the general Petri net reachability problem. In *13th ACM Symp. on Theory of Computing, STOC'81*, pages 238–246, New York, NY, USA, 1981. ACM.

[50] R. Mayr. Undecidable problems in unreliable computations. *Theoret. Comput. Sci.*, 297(1-3):337–354, 2003.

[51] M. L. Minsky. Recursive unsolvability of post's problem of "tag" and other topics in theory of turing machines. *The Annals of Mathematics*, 1961.

[52] C. A. Petri. Fundamentals of a theory of asynchronous information flow. In *IFIP Congress*, 1962.

[53] Ch. Rackoff. The covering and boundedness problems for vector addition systems. *Theoret. Comput. Sci.*, 6(2):223–231, 1978.

[54] K. Reinhardt. Reachability in Petri Nets with inhibitor arcs. `http://www-ti.informatik.uni-tuebingen.de/~reinhard/abstracts.html`, 1995. (unpublished).

[55] K. Reinhardt. Reachability in Petri Nets with inhibitor arcs. *Electr. Notes Theor. Comput. Sci.*, 223:239–264, 2008.

[56] F. Rosa-Velardo and D. de Frutos-Escrig. Name creation vs. replication in petri net systems. *Fundam. Inf.*, 88(3):329–356, Aug. 2008.

[57] F. Rosa-Velardo and D. de Frutos-Escrig. Decidability and complexity of petri nets with unordered data. volume 412 of *Theoretical Computer Science*, pages 4439–4451. 2011.

[58] F. Rosa-Velardo and G. Delzanno. Language-based comparison of petri nets with black tokens, pure names and ordered data. In A.-H. Dediu, H. Fernau, and C. Martín-Vide, editors, *Language and Automata Theory and Applications*, volume 6031 of *Lecture Notes in Computer Science*, pages 524–535. Springer Berlin / Heidelberg, 2010.

[59] D. Schmidt. Well-partial orderings and their maximal order types. 1979.

[60] S. Schmitz and Ph. Schnoebelen. Algorithmic aspects of WQO theory. 2012.

[61] R. Valk. Self-modifying nets, a natural extension of Petri nets. In *Proceedings of ICALP'78*, pages 464–476, 1978.

[62] R. Valk and G. Vidal-Naquet. Petri nets and regular languages. *J. Comput. System Sci.*, 23(3):299–325, 1981.

[63] A. Weiermann. A computation of the maximal order type of the term ordering on finite multisets. In K. Ambos-Spies, B. Löwe, and W. Merkle, editors, *Mathematical Theory and Computational Practice*, volume 5635 of *Lecture Notes in Computer Science*, pages 488–498. Springer Berlin / Heidelberg, 2009.

[64] H.-C. Yen. A unified approach for deciding the existence of certain petri net paths. *Information and Computation*, 96(1):119 – 137, 1992.