The Well Structured Problem for Presburger Counter Machines

Alain Finkel
LSV, ENS Paris-Saclay, CNRS, Université Paris-Saclay, IUF, France
alain.finkel@ens-paris-saclay.fr

Ekanshdeep Gupta
Chennai Mathematical Institute, Chennai, India
ekanshdeep@cmi.ac.in

Abstract
We introduce the well structured problem as the question of whether a model (here a counter machine) is well structured (here for the usual ordering on integers). We show that it is undecidable for most of the (Presburger-defined) counter machines except for Affine VASS of dimension one. However, the strong well structured problem is decidable for all Presburger counter machines. While Affine VASS of dimension one are not, in general, well structured, we give an algorithm that computes the set of predecessors of a configuration; as a consequence this allows to decide the well structured problem for 1-Affine VASS.

2012 ACM Subject Classification Theory of computation → Logic and verification; Theory of computation → Verification by model checking

Keywords and phrases Well structured transition systems, infinite state systems, Presburger counter machines, reachability, coverability

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2019.41


Funding The work reported was carried out in the framework of ReLaX, UMI2000 (ENS Paris-Saclay, CNRS, Univ. Bordeaux, CMI, IMSc). This work was also supported by the grant ANR-17-CE40-0028 of the French National Research Agency ANR (project BRAVAS).

1 Introduction

Context. Well Structured Transition Systems (WSTS) [9, 8] are a well-known model to solve termination, boundedness, control-state reachability and coverability problems. It is well known that Petri nets and Vector Addition Systems with States (VASS) are WSTS and that Minsky machines are not WSTS. But the characterization of counter machines which are well structured (resp. with strong monotony) is surprisingly unknown. Moreover, given a counter machine, can we decide whether it is well structured (resp. with strong monotony)? These questions are relevant since a positive answer could allow to verify particular instances of undecidable models like Minsky machines and counter machines. In this paper, we consider Presburger counter machines (PCM) where each transition between two control-states is labelled by a Presburger formula which describes how each counter is modified by the firing of the transition. The PCM model includes Petri nets, Minsky machines and most of the counter machine models studied in the literature, for example counter machines where transitions between control-states are given by affine functions having Presburger domains [3, 11].

Affine VASS. In an Affine VASS (AVASS), transitions between control-states are labelled by affine functions whose matrices have elements in $\mathbb{Z}$ (and not in $\mathbb{N}$ as usual). AVASS extends VASS (where transitions are translations) and positive affine VASS (introduced as
The Well Structured Problem for Presburger Counter Machines

self-modified nets in [21] and studied as affine well structured nets in [13]. [4] extends the Rackoff technique to AVASS where all matrices are larger than the identity matrix: for this subclass, coverability and boundedness are shown in EXPSPACE. The variation of VASS which may go below 0, called Z-VASS, is studied in [15] and for their extension, Z-Affine VASS, reachability is shown NP-complete for VASS with resets, PSPACE-complete for VASS with transfers and undecidable in general [2, 1]; let us remark that all Z-Affine VASS have positive matrices.

Moreover AVASS allow the simulation of the zero-test so they are at least as expressive as Minsky machines. But for dimension one, AVASS are more expressive than Minsky machines: in fact, Post* is computable as a Presburger formula for 1-counter Minsky machines but this is not the case for 1-AVASS which can generate the set of all the powers of 2 (this set is not the solution of any Presburger formula).

The computation of the set \( \text{Pre}^* \) of all predecessors of a configuration is effective for 2-VASS (extended with one zero-test and resets) [12] as a Presburger formula and for pushdown automata [5] as a regular language. But the computation of \( \text{Pre}^* \) fails for 3-VASS and for Pushdown VAS since \( \text{Pre}^* \) is neither semilinear nor regular [19].

Our contributions. We introduce two new problems related to well structured systems and Presburger counter machines. The so-called well structured problem: (1) given a PCM, is it a WSTS? and the strong well structured problem: (2) given a PCM, is it a WSTS with strong monotony?

We prove that the well structured problem is undecidable for PCM even if restricted to dimension one (1-PCM) with just Presburger functions (i.e. piecewise affine functions); undecidability is also verified for Affine VASS in dimension two (2-Affine VASS). The undecidability proofs use the fact that Minsky machines can be simulated by both 1-PCM and 2-Affine VASS. However, we prove the decidability of the well structured problem for 1-Affine VASS (which subsumes 1-Minsky machines). Since the strong monotony can be expressed as a Presburger formula, the strong well structured problem is decidable for all PCMs. These results are summarised below:

<table>
<thead>
<tr>
<th>PCM</th>
<th>Well Structured Problem</th>
<th>Strong Well Structured Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-PCM (functional)</td>
<td>U</td>
<td>D</td>
</tr>
<tr>
<td>Functional 1-PCM</td>
<td>U [Theorem 14]</td>
<td>D</td>
</tr>
<tr>
<td>2-AVASS</td>
<td>U</td>
<td>D</td>
</tr>
<tr>
<td>2-Minsky machines</td>
<td>U [Theorem 15]</td>
<td>D</td>
</tr>
<tr>
<td>1-AVASS</td>
<td>D [Theorem 26]</td>
<td>D</td>
</tr>
</tbody>
</table>

We give an algorithm that computes \( \text{Pre}^* \) of a 1-AVASS and this extends a similar known result for 1-Minsky machines and 1-VASS (and for pushdown automata [5]). The computation of \( \text{Pre}^* \) allows us to give a simple proof that reachability and coverability are decidable for 1-AVASS (in fact reachability is known to be PSPACE-complete for polynomial one-register machines [10] which contains 1-AVASS). Moreover, the computation of \( \text{Pre}^* \) allows to decide the well structured problem for 1-AVASS. These results are summarised below:

<table>
<thead>
<tr>
<th>1-PCM (functional)</th>
<th>Reachability</th>
<th>Coverability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-AVASS</td>
<td>D [Corollary 24]</td>
<td>D</td>
</tr>
<tr>
<td>( d )-totally positive AVASS</td>
<td>D [Theorem 29]</td>
<td>D</td>
</tr>
<tr>
<td>( d )-positive AVASS ( (d \geq 2) )</td>
<td>U [Theorem 28]</td>
<td>D [WSTS]</td>
</tr>
<tr>
<td>2-AVASS</td>
<td>U</td>
<td>U [Corollary 18]</td>
</tr>
</tbody>
</table>
Outline. We introduce in Section 2 two models, well structured transition systems (WSTS) and Presburger counter machines (PCM); we show that the property for an ordering to be well is undecidable. Section 3 analyses the decidability of the well structured problems for many classes of PCM and Affine VASS. Section 4 studies the decidability of reachability and coverability for the classes studied in Section 3.

Due to space constraints, some proofs are deferred to an extended version of this paper freely available online under the same title.

2 Counter machines and WSTS

A relation $\leq$ on a set $E$ is a quasi ordering if it is reflexive and transitive; it is an ordering if moreover $\leq$ is antisymmetric. A quasi ordering $\leq$ on $E$ is a well quasi ordering (wqo) if for all infinite sequences of elements of $E$, $(e_i)_{i \in \mathbb{N}}$, there exists two indices $i < j$ such that $e_i \leq e_j$.

For an ordered set $(E, \leq)$ and a subset $X \subseteq E$, the upward closure of $X$ denoted by $\uparrow X$ is defined as follows: $\uparrow X = \{x \mid \exists y \in X \text{ such that } y \leq x\}$. $X$ is said to be upward closed if $X = \uparrow X$.

2.1 Arithmetic counter machines

A $d$-dim arithmetic counter machine (short, $d$-arithmetic counter machine or an arithmetic counter machine) is a tuple $M = (Q, \Phi, \rightarrow)$ where $Q$ is a finite set of control-states, $\Phi$ is a set of logical formulae with $2d$ free variables $x_1, ..., x_d, x'_1, ..., x'_d$ and $\rightarrow \subseteq Q \times \Phi \times Q$ is the transition relation between control-states. We can also without loss of generality assume that $\rightarrow$ covers $\Phi$, i.e. $\Phi$ does not have unnecessary formulae. A configuration of $M$ refers to an element of $Q \times \mathbb{N}^d$. The operational semantics of a $d$-arithmetic counter machine $M$ is a transition system $S_M = (Q \times \mathbb{N}^d, \rightarrow)$ where $\rightarrow \subseteq (Q \times \mathbb{N}^d) \times (Q \times \mathbb{N}^d)$ is the transition relation between configurations. For a transition $(q, \phi, q')$ in $M$, we have a transition $(q; x_1, ..., x_d) \rightarrow (q'; x'_1, ..., x'_d)$ in $S_M$ iff $\phi(x_1, ..., x_d, x'_1, ..., x'_d)$ holds. Note that we are slightly abusing notation by using the same $\rightarrow$ for both $M$ and $S_M$. We may omit $\Phi$ from the definition of a counter machine if it is clear from context.

A $d$-dim arithmetic counter machine $M$ with initial configuration $c_0$ is defined by the tuple $M = (Q, \Phi, \rightarrow, c_0)$ where $(Q, \Phi, \rightarrow)$ is a $d$-arithmetic counter machine and $c_0 \in Q \times \mathbb{N}^d$ is the initial configuration. An arithmetic counter machine is effective if the transition relation is decidable (there is a decidable procedure to determine if there is a transition $x \rightarrow y$ between any two configurations $x, y$) and this is the case when it is given by an algorithm, a recursive relation, or decidable first order formulae (for instance Presburger formulae). An arithmetic counter machine is said to be functional if each formula in $\Phi$ that labels a transition in $M$ defines a partial function.

Most usual counter machines can be expressed with Presburger formulae. It is well known that Presburger arithmetic with congruence relations without quantifiers is equivalent in expressive power to standard Presburger arithmetic [14].

Definition 1. A Presburger counter machine (PCM) is an arithmetic counter machine $M = (Q, \Phi, \rightarrow)$ such that $\Phi$ is a set of Presburger formulae with congruence relations without quantifiers.

Proposition 2 ([6]). The property for a $d$-dim PCM to be functional is decidable in NP.
41:4 The Well Structured Problem for Presburger Counter Machines

Minsky machines with $d$ counters are $d$-PCM $M = (Q, \Phi, \to)$ where $\Phi$ consists of either translations with upward closed guards, or formulae of the form $\bigwedge_{i=1}^{d}(x_i = x_i') \land x_k = 0$ for varying $k$ (zero-tests). Vector Addition Systems with States (VASS) are Minsky machines without zero-tests. An Affine VASS with $d$ counters ($d$-AVASS) is a $d$-PCM where each transition is labelled by a formula equivalent to an affine function of the form $f(x) = Ax + b$ where $A \in M_d(\mathbb{Z})$ is a $d \times d$ matrix over $\mathbb{Z}$ and $b \in \mathbb{Z}^d$. The domain of such a function would be the (Presburger) set of all $x \in \mathbb{N}^d$ such that $Ax + b \in \mathbb{N}^d$. For convenience, we will denote $d$-AVASS transitions by a pair $(A, b) \in M_d(\mathbb{Z}) \times \mathbb{Z}^d$. Note that AVASS is an extension of VASS where transitions are not labelled by vectors but by affine functions $(A_i, b_i)$. Let us define positive and totally-positive AVASS. A positive AVASS $S$ is an AVASS such that every matrix $A_i$ of $S$ is positive. This model has been studied for instance in [13]. A totally-positive AVASS $S$ is a positive AVASS such that every vector $b_i$ of $S$ is positive. For totally positive AVASS, an instance of the boundedness problem has been shown decidable in [13]. Note that we say something is positive if it is greater than or equal to 0, not strictly greater than 0.

Example 3. The machine $M_1$ in Figure 1 is a 1-AVASS but it is not a 1-VASS because there is a negative transition from $q_1$ to $q_1$.

Proposition 4 ([6]). Checking whether a given PCM is a VASS, AVASS, positive AVASS or a totally positive AVASS is decidable.

2.2 Well structured transition systems

A transition system is a tuple $S = (X, \to)$ where $X$ is a (potentially infinite) set of configurations and $\to \subseteq X \times X$ is the transition relation between configurations. We denote by $\xrightarrow{\to}$ the reflexive and transitive closure of $\to$. For a subset $S \subseteq X$, we denote by $\text{Pre}(S) := \{ t \mid t \to s \text{ for some } s \in S \}$, and $\text{Pre}^*(S) := \{ t \mid t \xrightarrow{\to} s \text{ for some } s \in S \}$. Similarly for $\text{Post}(S)$ and $\text{Post}^*(S)$.

An ordered transition system $S = (X, \to, \leq)$ is a transition system $(X, \to)$ with a quasi-ordering $\leq$ on $X$. Given two configurations $x, y \in X$, $x$ is said to cover $y$ if there exists a configuration $y' \geq y$ such that $x \xrightarrow{\to} y'$. An ordered transition system $S = (X, \to, \leq)$ is monotone, if for all configurations $s, t, s' \in X$ such that $s \to t, s' \geq s$ implies that $s'$ covers $t$. $S$ is strongly monotone if for all configurations $s, t, s' \in X$ such that $s \to t, s' \geq s$ implies that there exists $t' \geq t$ such that $s' \to t'$.

Definition 5 ([8]). A well structured transition system (WSTS) is an ordered transition system $S = (X, \to, \leq)$ such that $(X, \leq)$ is a wqo and $S$ is monotone.

The coverability problem is to determine, given two configurations $s$ and $t$, whether there exists a configuration $t'$ such that $s \xrightarrow{\to} t' \geq t$ (s covers t). This problem is one often studied alongside well-structuredness.
Let us consider the usual wqo $\leq$ on $Q \times \mathbb{N}^d$ associated with a $d$-counter machine $M = (Q, \rightarrow): (q_1; x_1, x_2, \ldots, x_d) \leq (q_2; y_1, \ldots, y_d) \iff (q_1 = q_2) \land (\bigwedge_{i=1}^d x_i \leq y_i)$.

We say that an arithmetic counter machine $M = (Q, \Phi, \rightarrow)$ is well structured (or is a WSTS) iff its associated transition system $S_M$ is a WSTS under the usual ordering. Since the usual ordering on $(Q \times \mathbb{N}^d, \leq)$ is a wqo, let us remark that the associated ordered transition system $S_M = (Q \times \mathbb{N}^d, \rightarrow, \leq)$ is a WSTS iff $S_M$ is monotone.

Given a counter machine $M = (Q, \rightarrow)$, the control-state reachability problem is trivially well structured since there is a reduction from the Halting Problem as follows: 

\[ M \text{ is a WSTS} \iff \text{Post}^*(q_1, 0) = \{q_1, 9\} \]

Hence we can deduce that $(q_1, 10)$ cannot cover $(q_1, 19)$. Hence $M$ is not well structured. By Section 4, we show that deciding whether arbitrary (non-effective) transition systems are well-structured is undecidable. 

We introduce two new problems related to WSTS and Presburger counter machines.

- The well structured problem: given a PCM, is it a WSTS?
- The strong well structured problem: given a PCM, is it a WSTS with strong monotony?

Example 6. The machine $M_1$ (Figure 1) is not strongly monotone since we have: 

\[ (q_1, 0) \xrightarrow{x=19-x} (q_1, 19). \]

However, we see that $\text{Post}^*(q_1, 10) = \{(q_1, 9), (q_1, 10)\}$. Therefore we can deduce that $(q_1, 10)$ cannot cover $(q_1, 19)$. Hence $M_1$ is not well structured. We give, in Section 4, an algorithm for deciding whether a 1-AVASS is well structured.

It is shown in [8] that almost every transition system can be turned into a WSTS for the termination ordering which is not, in general, decidable. So the problem is not only to decide whether a system is a WSTS in general; we have to choose a decidable ordering. We show that deciding whether arbitrary (non-effective) transition systems are well-structured for the usual (decidable) ordering on natural numbers is undecidable.

Proposition 7. The well structured problem for 1-arithmetic counter machines is undecidable.

We now show that restricting to effective transition systems does not allow us to decide the property of being a WSTS.

Corollary 8. The well structured problem (for the usual ordering on $\mathbb{N}$) for effective transition systems whose set of configurations is included in $\mathbb{N}$ is undecidable.

Proof. There exists a reduction from the Halting Problem as follows:

Given a Turing machine $M$, we define a transition system $S_M = (\mathbb{N}, \rightarrow_M)$ as follows: 

If $(m = 0) \lor (M \text{ does not halt in } m \text{ steps})$, then, for all $n$, there is a transition $m \rightarrow_M n$. Hence this transition relation $\rightarrow_M$ is decidable. Now, if $M$ does not halt, then there is a transition $m \rightarrow_M n$ for all $m, n \in \mathbb{N}$. This satisfies monotony, hence in this case, $S_M$ is a WSTS. However, if $M$ halts in exactly $m$ steps, then there is no transition from $m + 1$ but there is, in any case, a transition from $0$ to $n$ for all $n$. Hence in this case, $S_M$ is not a WSTS. Therefore, $S_M$ is a WSTS iff $T$ does not halt.

2.3 Testing whether an ordering is well

In the previous results, the usual well ordering on natural numbers is not necessarily the unique decidable ordering when considering the well structured problem for counter machines. Let $\leq$ be a decidable quasi ordering relation on $\mathbb{N}^d$. If we are interested in whether a counter machine with this ordering is WSTS, it raises the natural question of whether we can decide if $\leq$ is a wqo. Unfortunately, but unsurprisingly, we first show that this property is undecidable in dimension one ($d = 1$).
Proposition 9. The property for a decidable ordering on \( \mathbb{N} \) to be a well ordering is undecidable.

Let us study the case of Presburger-definable orderings in \( \mathbb{N} \). Among many equivalent characterizations of wqo, we know that a quasi ordering is well iff it satisfies well-foundedness and the finite anti-chain property. Both of these properties can be expressed using monadic second order variables. But, it is shown in [17] that Presburger Arithmetic with a single monadic variable becomes undecidable. Hence, this cannot directly be used to check if a Presburger-definable ordering is a wqo. However, we still have the following result:

Proposition 10. The property for a Presburger relation on \( \mathbb{N} \) to be a well quasi ordering is decidable.

3 The well structured problem for PCM

In the sequel, whenever we talk about PCM being WSTS, we will consider the usual ordering on \( Q \times \mathbb{N}^d \) defined in subsection 2.2. We introduce a general technique to prove undecidability of checking whether a counter machine of some class is a WSTS. Let \( S_0 \) be the class of machines we are interested in. We will show reduction from reachability in Minsky machines.

Lemma 11. Suppose we have a procedure which takes a 2 counter Minsky machine with initial state \( M = (Q, \rightarrow_M, q_0) \) and a control-state \( q_1 \) as input and generates a machine \( N \) of class \( S_0 \) which satisfies the following two requirements:

1. All control-states in \( M \) are reachable implies \( N \) is a WSTS.
2. \( N \) is a WSTS implies \( q_1 \) is reachable in \( M \) from \((q_0;0,0)\).

Then, the well structured problem for \( S_0 \) is undecidable.

Proof. Suppose that the well structured problem for \( S_0 \) is decidable. We will use the above procedure to get an algorithm for Minsky machine reachability. Fix \((M, q_1)\), where \( M = (Q, \rightarrow_M, q_0) \). We want to check if \( q_1 \) is reachable from \((q_0;0,0)\).

Let \( |Q| = n \). Consider all \( 2^{n-2} \) subsets \( Q' \subseteq Q \) satisfying that \( \{q_0, q_1\} \subseteq Q' \). For each such \( Q' \), let \( \rightarrow_{Q'} \) denote the restriction of \( \rightarrow_M \) to the set \( Q' \times Q' \). Hence, we can associate a Minsky machine \( M' = (Q', \rightarrow_{Q'}, q_0) \) to each such subset \( Q' \). We call \( M' \) a sub-machine of \( M \) corresponding to \( Q' \).

Now, for each sub-machine \( M' \), we consider the machine \( N' \) of class \( S_0 \), generated by the given procedure from \((M', q_1)\). If there exists \( M' \) such that \( N' \) is a WSTS, then we have that \( q_1 \) is reachable in \( M' \) (by condition (2)), hence in \( M \).

On the other hand, if \( q_1 \) was reachable in \( M \), then let \( Q_{reach} \subseteq Q \) be the set of all control-states of \( M \) which are reachable from \((q_0;0,0)\). Let its corresponding sub-machine be \( M' \). Since all control-states of \( M' \) are reachable (by choice of \( Q_{reach} \)), therefore the corresponding \( N' \) will be a WSTS (by condition (1)).

Hence, \( q_1 \) is reachable in \( M \) from \((q_0;0,0)\) iff there exists a subset \( Q' \subseteq Q \) satisfying that \( \{q_0, q_1\} \subseteq Q' \) such that the corresponding sub-machine \( M' \) is a WSTS. Since there are only \( 2^{n-2} \) such subsets, we can check all of them to decide whether \( q_1 \) is reachable in \( M \).

Hence, we have given an algorithm to check reachability in Minsky machine. Therefore, the well structured problem for \( S_0 \) is undecidable.

We will use Lemma 11 to prove that the well structured problem for functional 1-dim PCMs is undecidable. To apply Lemma 11, we need to give an algorithm which takes a Minsky machine \( M = (Q, \rightarrow_M, q_0) \) and a control-state \( q_1 \), and generates a functional 1-dim PCM \( N_1 \) satisfying conditions (1) and (2).
Construction of a functional 1-dim PCM $N_1$

Let $(M, q_0)$ be given. The procedure to generate a 1-dim PCM $N_1$ is as follows:

Let $v_p(n)$ denote the largest power of $p$ dividing $n$. For $M = (Q, \rightarrow_M, q_0)$, we define the 1-PCM $N_1 = (Q, \rightarrow_N, (q_0, 1))$ with the same set $Q$ of control-states. We will represent the values of the two counters $(m, n)$ by the one-counter values $2^m3^n c$ for any $c$ such that $v_2(c) = v_3(c) = 0$. Conversely, a configuration $(q, n)$ of $N_1$ will correspond to $(q; v_2(n), v_3(n))$ of $M$. Note that, we are allowing multiplication by constants $c$ in $N_1$ as long as $v_2(n)$ and $v_3(n)$ remain unchanged.

Increment/decrement of counters corresponds to multiplication/division by 2 and 3 which is Presburger expressible. Similarly, zero-test corresponds to checking divisibility by 2 and 3 which is again Presburger-expressible. So first, for each transition in $\rightarrow_M$, we add the corresponding transition to $\rightarrow_N$.

Now, to get the suitable properties of conditions (1) and (2), we will add two more types of transitions to $\rightarrow_N$. For each control-state $q$, we add a transition $(q, x_1 = 6x_1 + 1, q_0)$ to $\rightarrow_N$. We shall call it a “reset-transition” because $v_2(6x_1 + 1) = v_3(6x_1 + 1) = 0$, so this transition corresponds to a counter-reset in $M$ from anywhere regardless of our present configuration. Note that such a transition would not change the reachability set in $M$. This “reset-transition” is crucial in forcing well-structuredness in $N$. Also, we add a transition $(q_0, x_1 = 0 \land x_1 = 0, q_1)$ to $\rightarrow_N$ to ensure condition (2). Since the configuration $(q_0, 0)$ cannot be reached from the initial configuration $(q_0, 1)$ during any run of $N_1$, this will also not affect the reachability set of $N_1$. Note that, all of our transitions are functional, hence $N_1$ is a functional 1-dim PCM.

Now, we show that the construction of $N_1$ satisfies conditions (1) and (2).

\begin{lemma}
The functional 1-dim PCM $N_1$ satisfies condition (1).
\end{lemma}

\begin{proof}
Suppose that all control-states of $M$ are reachable from $(q_0; 0, 0)$. Then we claim that $N_1$ will be a WSTS. Suppose there is a transition $(q, n) \rightarrow_N (q', m)$ and $(q, n')$ is a configuration with $(q, n') \geq (q, n)$. Hence we want to show existence of some path $(q, n') \rightarrow_N (q', m') \geq (q', m)$.

Case 1: The transition $(q, n) \rightarrow_N (q', m)$ is a “reset-transition”. Hence $q' = q_0$ and $m = 6n + 1$.

In this case, note that since $n' \geq n$, the transition $(q, n') \rightarrow_N (q_0, 6n' + 1) \geq (q_0, m)$ satisfies the requirement.

Case 2: The transition $(q, n) \rightarrow_N (q', m)$ is not a “reset-transition”. In this case, $m \leq 3n$ because the above transition corresponds, in $M$ to an increment/decrement in $c_1$ or $c_2$ or a zero-test. In each case, we can check that $m \leq 3n$. Let there be a path $(q_0; 0, 0) \rightarrow_M (q'; n_1, n_2)$ in $M$ for some $n_1, n_2$. Such a path exists because all control-states in $M$ are reachable. Hence, we take the “reset-transition” $(q, n') \rightarrow_N (q_0, 6n' + 1)$ and follow the corresponding path $(q_0, 6n' + 1) \rightarrow_N (q', 2^{n_1}3^{n_2} (6n' + 1)) \geq (q', 3n) \geq (q', m)$.

Hence we have again shown monotony to prove that $N_1$ is a WSTS.

Hence we have shown that if all control-states of $M$ are reachable, then $N_1$ is monotone.
\end{proof}

\begin{lemma}
The functional 1-dim PCM $N_1$ satisfies condition (2).
\end{lemma}

\begin{proof}
Since there is a transition $(q_0, 0) \rightarrow_N (q_1, 0)$, we deduce that if $N_1$ is a WSTS, then $(q_0, 1) \rightarrow_N (q_1, n)$ for some $n$ by monotony because $(q_0, 0) \leq (q_0, 1)$. Also note that since $N_1$ simulates $M$, hence reachability of $q_1$ in $N_1$ implies that $q_1$ is reachable from $(q_0; 0, 0)$ in $M$.
\end{proof}
Since we have provided a construction of functional 1-dim PCM $N_1$ satisfying conditions (1) and (2), from Lemma 11 we have that:

▶ **Theorem 14.** The well structured problem for functional 1-dim PCMs is undecidable.

Similarly, we can use Lemma 11 to show this result for 2 counter Minsky machines. This construction can be found in the extended version of this paper.

▶ **Theorem 15.** The well structured problem for 2-dim Minsky machines is undecidable.

Now, we make the observation that we can perform zero-tests using affine functions. The basic idea is that a transition $x' = -x$ is only satisfied by a counter whose value is 0. Increments/decrements can already be implemented in 2-AVASS since translations are affine functions. A zero test on the first counter can be done by having a transition labelled by $\left( \begin{array}{cc} -1 & 0 \\ 0 & 1 \end{array} \right)$, and similarly for second counter. Since we can implement both increment/decrements and zero-tests with 2-AVASS, we can simulate 2-counter Minsky machines with 2-AVASS. Note that we can extend this result to $d$-AVASS simulating $d$-counter Minsky machines.

As a direct consequence of this and Theorem 15, we have that:

▶ **Corollary 16.** The well structured problem for 2-AVASS is undecidable.

However, if we consider strong monotony instead of monotony, the above undecidability results can be turned into a decidability result. Strong monotony can be expressed in Presburger arithmetic as follows:

$$\bigwedge_{\phi \in \Phi} (\forall x_1 \ldots \forall x_d \forall x'_1 \ldots \forall x'_d \forall y_1 \ldots \forall y_d ((\bigwedge_{i=1}^{d} x_i \leq y_i) \land \phi(x_1, ..., x_d, x'_1, ..., x'_d)) \Rightarrow (\exists y'_1 \ldots \exists y'_d (\bigwedge_{i=1}^{d} x'_i \leq y'_i) \land \phi(y_1, ..., y_d, y'_1, ..., y'_d)))$$

Since Presburger arithmetic is decidable, the strong well structured problem for $d$-PCM is decidable.

▶ **Remark 17.** The validity of the formula of strong monotony can also be decided for extended PCM defined in decidable extensions of Presburger Arithmetic.

### 4 Decidability results for 1-AVASS

Now, let us look at some reachability and coverability results for the various models of AVASS. First, we can simulate 2-counter Minsky machines with 2-AVASS. Since coverability and reachability are undecidable for 2-counter Minsky machines, we directly have the following result:

▶ **Corollary 18.** Control-state reachability, hence coverability is undecidable for 2-AVASS.

Similarly, we showed in Construction of functional 1-PCM $N_1$ that we can also simulate 2-counter Minsky machines with functional 1-PCM. Hence, we also have the following:

▶ **Corollary 19.** Control-state reachability, hence coverability is undecidable for functional 1-PCM.
Now, let us examine the case of 1-AVASS. For 1-AVASS, reachability and consequently coverability is decidable from work done in [10]. We show that checking whether it is a WSTS is also decidable. Moreover, we give a simpler proof of decidability of reachability and coverability.

Given $M = (Q, \rightarrow)$ a 1-AVASS and a final configuration $(q_f, n_f)$ that we want to check reachability for, we present Algorithm 1 which computes $Pre^*(q_f, n_f)$ as a Presburger formula.

A transition $(q, x' = ax + b, q')$ is positive if $a \geq 0$. Let a cycle/path in $M$ be called positive if all transitions are positive. A cycle $(q_1, \ldots, q_k)$ is called a simple cycle if $q_1, \ldots, q_k$ are all pairwise distinct.

Let us denote by $Pre_q$ the set $Pre^*(q_f, n_f) \cap (\{q\} \times \mathbb{N})$. For a transition $t = (q, x' = ax + b, q')$ and a given subset of $X \subseteq \mathbb{N}$, let $Pre^t(X)$ denote $\{n : an + b \in X\}$. For a simple cycle $c$ rooted at $q$ with an effective guard and transition, extend the above notation $Pre^c$ for $i$ repetitions of the cycle. Then, let $Pre^{c^i}(X) := \bigcup_{i \in \mathbb{N}} Pre^c(X)$. We will conveniently replace $X$ by a formula which denotes a subset of $\mathbb{N}$.

\begin{algorithm}
1: \textbf{procedure} \textsc{computePre}^*
2: \hspace{1em} for all $q \in Q$ do
3: \hspace{2em} $\phi_q \equiv \bot$
4: \hspace{2em} $\phi_{q_f} \equiv (n = n_f)$
5: \hspace{1em} for all $q \in Q$ do
6: \hspace{2em} \hspace{1em} for all simple cycles $c$ rooted at $q$ do
7: \hspace{2em} \hspace{2em} $c$.transition = \textsc{simplifyTransition}(c)
8: \hspace{2em} \hspace{2em} $c$.guard = \textsc{computeGuard}(c)
9: \hspace{2em} notFinished = True
10: \hspace{2em} while notFinished = True do
11: \hspace{2em} \hspace{1em} notFinished = False
12: \hspace{2em} \hspace{1em} for all $q \in Q$ do
13: \hspace{2em} \hspace{2em} $\phi' = \phi_q$
14: \hspace{2em} \hspace{2em} for all transitions $t = (q, x' = ax + b, q') \in \rightarrow$ do
15: \hspace{2em} \hspace{2em} \hspace{1em} \textsc{exploreTransition}(t)
16: \hspace{2em} \hspace{2em} for all simple cycles $c$ containing $q$ do
17: \hspace{2em} \hspace{2em} \hspace{1em} \textsc{exploreCycle}(c)
18: \hspace{2em} \hspace{2em} if $\phi' \neq \phi_q$ then \hspace{1em} \hspace{1em} \hspace{1em} \text{\textcolor{red}{Check equality as Presburger formulae}}
19: \hspace{2em} \hspace{2em} notFinished = True
\end{algorithm}

The algorithm will keep a variable $\phi_q$ for each control-state $q \in Q$ which will store a Presburger formula (with one free variable $n$) denoting the currently discovered subset of $Pre_q$. Let this be denoted by $[\phi_q]$, i.e., $[\phi_q] := \{n : \phi_q(n)\}$. For uniformity, we can assume that $\phi_q$ is a disjunction of formulae of form $range \land mod$ where $range \equiv (r \leq n \leq s)$ ($s$ possibly $\infty$) and $mod \equiv (n = d_q \land d)$. We initially simplify each simple cycle into a meta-transition which is the composition of all individual transitions in the cycle. We will also compute the guard of a cycle. Since each positive transition has an upward closed guard and each negative transition has a downward closed guard, the guard of a cycle will be of the form $r \leq n \leq s$ for some $r, s \in \mathbb{N}$ ($s$ possibly $\infty$). Hence, we will only consider a cycle in terms of its guard and its meta-transition.
The Well Structured Problem for Presburger Counter Machines

We use two main procedures in COMPUTEPre*:  
1. EXPLORETRANSITION: Given a transition $t = (q, x' = ax + b, q')$, it computes $Pre^t(\phi_q)$ and appends it to $\phi_q$.  
2. EXPLORECYCLE: Given a simple cycle $c$ rooted at $q$, it computes $Pre^c(\phi_q)$ and appends it to $\phi_q$.  

Lemma 20. For any transition $t$, and any simple cycle $c$, given $\phi_q$, $Pre^t(\phi_q)$ and $Pre^c(\phi_q)$ are both Presburger expressible and effectively computable.

With this lemma the algorithm is well-defined. Now let us prove the termination and the correctness of the algorithm.

Proposition 21. Algorithm COMPUTEPre* terminates.

Proof. For each $q$, we will show that $Pre_q$ can be obtained in finitely many iterations of the algorithm. Let $q \in Q$ be arbitrary.

Case 1: $Pre_q$ is finite:  
Each value will be discovered in finitely many iterations, hence $Pre_q$ will be obtained in finitely many iterations.

Case 2: $Pre_q$ is infinite:  
Since we are talking about reaching $(q_f, n_f)$, we note that the only transitions which can decrease arbitrarily large values are transitions of the form $x' = b$ or $x' = x - a, a > 0$. Hence, since $Pre_q$ has arbitrarily large values, and each run has to reach $n_f$ (i.e. has to be decreased), we can see that there must either be a transition $x' = b$, or a positive cycle with meta-transition $x' = x - a$, reachable from $q$ through a positive path.

Case 2.i: There is a transition $x' = b$:  
In this case, there exists $N$ such that for all $n \geq N$, the same path suffices. In this case, once the aforementioned path is discovered, $\{ n : n \geq N \}$ becomes a subset of $[\phi_q] \subseteq Pre_q$, which leaves finitely many values in $Pre_q \setminus [\phi_q]$, which can again be discovered by finitely many additional runs.

Case 2.ii: There are positive cycles with meta-transition $x' = x - a$:  
The idea is that we will cover $Pre_q$ when we compute $Pre^c$ for such a cycle $c$. This is because for such a cycle, all that matters is the value of the counter modulo $a$. Since there are only finitely many distinct values modulo $a$, these will again be discovered in finitely many runs. Hence, each cycle will be discovered in finitely many runs. Therefore since there are finitely many simple cycles, the corresponding values of $Pre_q$ will also be discovered in finitely many runs.

Hence, for all $q$, in finitely many runs we will get $Pre_q = [\phi_q]$. At such a point, the algorithm has to stop, hence termination is guaranteed.

Theorem 22 (Correctness). Given a 1-AVASS $M = (Q, \rightarrow)$ and a configuration $(q, n)$, the algorithm COMPUTEPre* computes $Pre^*(q, n)$ as a Presburger formula.

Proof. We will show that Algorithm 1 upon termination will always have $[\phi_q] = Pre_q$.

That $[\phi_q] \subseteq Pre_q$ should be clear. Suppose the algorithm terminates with $[\phi_q] \subseteq Pre_q$ for some $q \in Q$. For some value $n \in Pre_q \setminus [\phi_q]$, consider a path which covers $(q_2, n_2)$, say the path is $(q, n) \rightarrow (p_1, n_1) \rightarrow \ldots \rightarrow (p_m, n_m) \rightarrow (q_2, n')$. In such a path, consider the largest $i$, such that $n_i \notin [\phi_{p_i}]$. Now, in the last iteration of the algorithm, since $n_{i+1} \notin [\phi_{p_{i+1}}]$ (by choice of $i$), hence, we will explore the edge to include $n_i \in [\phi_{p_i}]$. Hence, the algorithm would not have terminated. Contradiction. Hence, when the algorithm terminates, $[\phi_q] = Pre_q$.  

As a matter of fact, reachability is undecidable for VASS with two resets in dimension 2. This can be used to determine whether a given -AVASS is a WSTS or not.

We observe that the set \( \{q_1\} \times \mathbb{N} \) is a subset of \( \operatorname{Pre}^*(\top(q_2, b)) \). Since we can compute \( \operatorname{Pre}^*(q, n) \), we can also compute \( \operatorname{Pre}^*(\top(q, n)) \) by the same technique as in Corollary 24.

This can be used to determine whether a given 1-AVASS is a WSTS as follows.

**Theorem 26.** The well-structured problem is decidable for 1-AVASS.

**Proof.** First we show that \( M \) is a WSTS, iif for all negative transitions \((q_1, (x' = ax + b), q_2)\), the set \( \{q_1\} \times \mathbb{N} \) is a subset of \( \operatorname{Pre}^*(\top(q_2, b)) \).

If the transition is positive, i.e. \( a \geq 0 \), then we directly have the transition \((q_1, n') \rightarrow (q_2, an + b) \geq (q_2, an + b) \). If the transition is negative, then we have that \((q_2, an + b) \leq (q_2, b)\). Since \((q_1, n) \in \operatorname{Pre}^*(\top(q_2, b))\) (by hypothesis, since it is a negative transition), hence we have that \((q_1, n') \rightarrow (q_2, b) \geq (q_2, b) \geq (q_2, an + b)\). Hence, \( M \) is monotone. Therefore, \( M \) is a WSTS iif for all negative transitions \((q_1, (x' = ax + b), q_2)\), the set \( \{q_1\} \times \mathbb{N} \) is a subset of \( \operatorname{Pre}^*(\top(q_2, b)) \).

Now, since \( \operatorname{Pre}^*(\top(q, n)) \) is computable, we can check that for each negative transition \((q_1, (x' = ax + b), q_2)\), the set \( \{q_1\} \times \mathbb{N} \) is a subset of \( \operatorname{Pre}^*(\top(q_2, n)) \) to determine whether \( M \) is a WSTS or not.

**Example 27.** Let us consider machine \( M_1 \) in Figure 1 and its negative transition \((q_1, (x' = 19 - x), q_1)\). We observe that the set \( \operatorname{Pre}^*(\top(q_1, 19)) = \{q_1, q_2\} \times \{n : n \geq 19\} \) does not contain \( \{q_1\} \times \mathbb{N} \), hence machine \( M_1 \) is not a WSTS. However, in this example (Figure 1), if we replace the transition \((q_1, (x' = x - 3), q_2)\) by \((q_1, (x' = x + 1), q_2)\), we will get a new machine \( M_2 \) which is still not a 1-AVASS, but it is a WSTS.

Let us focus our attention to positive AVASS now. We know that for positive 1-AVASS reachability is decidable from Corollary 24. We show that reachability is undecidable for positive 2-AVASS by reduction from Post’s Correspondence Problem (PCP) [16].

Our result completes the view about decidability of reachability for VASS extensions in small dimensions.

As a matter of fact, reachability is undecidable for VASS with two resets in dimension 3 (to adapt the proof in [7]), hence for positive 3-AVASS but it is decidable for VASS with two resets in dimension 2 [12]. If we replace resets by affine functions, reachability becomes undecidable in dimension two.
41:12 The Well Structured Problem for Presburger Counter Machines

\[
\begin{pmatrix}
2^{\mu_1} & 0 & 0 \\
0 & 2^{\mu_2} & 0 \\
0 & 0 & 2^{\mu_3}
\end{pmatrix}
\begin{pmatrix}
(a_1)_{12} \\
(b_1)_{12} \\
(b_2)_{12}
\end{pmatrix}
\begin{pmatrix}
2^{\nu_1} & 0 & 0 \\
0 & 2^{\nu_2} & 0 \\
0 & 0 & 2^{\nu_3}
\end{pmatrix}
\begin{pmatrix}
(a_3)_{12} \\
(b_3)_{12} \\
(b_4)_{12}
\end{pmatrix}
\]

\[q_0 \xrightarrow{\delta} q_1 \xrightarrow{(I, \delta)} q_2 \xrightarrow{\cdots} \]

**Figure 2** Construction for undecidability of reachability for positive 2-AVASS by reduction from PCP.

Reichert gives in [20] a reduction from the Post correspondence problem to reachability in a subclass of 2-AVASS and we may remark that his proof is still valid for positive 2-AVASS. Blondin, Haase and Mazowiecki made some similar observations [1] for subclasses of 3−Z-AVASS, with positive matrices. Our proof is essentially the same as [20].

**Theorem 28.** Reachability is undecidable for positive 2-AVASS.

**Proof.** Suppose we are given an instance of PCP, i.e. we are given \(a_1, ..., a_k, b_1, ..., b_k \in \{0, 1\}^*\) for some \(k \in \mathbb{N}\). We want to check if there exists some sequence of numbers \(n_1, ..., n_\ell \in \{1, ..., k\}\) such that \(a_{n_1}a_{n_2}...a_{n_\ell}\) (concatenated as strings).

We will construct the positive 2-AVASS as demonstrated in Figure 2, where \(|a_i|\) refers to the length of the string, and \((a_i)_{12}\) refers to the number encoded by the string \(a_i\) if read in binary (most significant digit to the left). The idea is that we use the two counters to store the value of \((a_{n_1}...a_{n_\ell})_{12}\) and \((b_{n_1}...b_{n_\ell})_{12}\) for any \(n_1, ..., n_\ell\). But we first increment each counter to keep track of leading zeroes. Now, the configuration \((q_2; 0, 0)\) is reachable from \((q_0; 0, 0)\) in the positive 2-AVASS described in Figure 2 iff the given PCP has an affirmative answer. Hence, checking reachability in positive \(d\)-AVASS is undecidable for \(d \geq 2\).

Also, we note that positive-AVASS are well-structured with strong monotony. Hence coverability is decidable [13]. If we look at totally-positive AVASS, we can see that coverability is already decidable by the same argument. However, reachability is also decidable.

**Theorem 29.** Reachability is decidable in totally-positive AVASS for any dimension.

**Proof.** Let \(M = (Q, \rightarrow)\) be a totally-positive \(d\)-AVASS. Given \((q_0; n_1, ..., n_d)\), suppose we want to check reachability of \((q_f; m_1, ..., m_d)\). Let \(N = \max\{m_1, ..., m_d\}\). Let \(f_N : \mathbb{N} \to \{1, ..., N, \omega\}\) be the function which is identity on \(\{1, ..., N\}\) and maps \(\{N+1, ..., \}\) to \(\omega\). Extend this function to the set \(\mathbb{N}^d\) component-wise. Since \(M\) is totally-positive, we can restrict our search space from \(Q \times \mathbb{N}^d\) to \(Q \times \{0, ..., N, \omega\}^d\) by applying \(f_N\) to each configuration and using the following arithmetic rules: \(0, \omega = 0, \) and for all \(k \geq 1, k, \omega = \omega \) and \(\omega + k = \omega\).

We claim that if \((q_f; m_1, ..., m_d)\) is reachable, then it is reachable in this restricted search-space. This follows from the fact that given any element \((n_1, ..., n_d)\) of \(\mathbb{N}^d\), and a totally positive transition \(t = (A, b)\), we will have that \(t(f_N(n_1, ..., n_d)) = f_N(t(n_1, ..., n_d))\) \((t\) acts on \(f_N(n_1, ..., n_d)\) to give an element in \(\{0, ..., N, \omega\}^d\). This is because a totally positive transition cannot decrease a value other than by multiplying it by 0, hence any value greater than \(N\) will continue to be greater than \(N\). Also note that, by choice of \(N\), \(f_N(m_1, ..., m_d) = (m_1, ..., m_d)\).

Once we have this, we can make an induction on the length of the path to see that if \((q_f; m_{k1}, ..., m_{kd})\) is reachable, it is reachable in the restricted search-space \(Q \times \{0, ..., N\}^d\).

Since \(Q \times \{0, ..., N, \omega\}^d\) is finite, this shows decidability of reachability.
5 Conclusion and perspective

We introduced two variants of the well structured problem for PCM and we solve it for many classes of PCMs. Moreover, we answer the decidability questions for reachability and coverability for classes of PCMs and AVASSs (we summarise the results of Section 4 in Figure 3).

Many open problems can be attacked like the complexity of reachability for 1-AVASS (reachability is NP for 1-VASS and PSPACE for polynomial VASS), the size of Pre$^*$ of a 1-AVASS (and its relation with the theory of flattable VASS [18]), and the decidability of the property for a Presburger relation on $\mathbb{N}^d$ to be a well-quasi ordering for $d \geq 2$.

We also open the way to study the decidability of the well structured problems (for various orderings) for many other models like pushdown counter machines, FIFO automata, Petri nets extensions. For instance, we wish to solve the well structured problems for FIFO automata. We know that lossy FIFO automata are well structured (for the subword ordering) but what is the class of perfect FIFO automata which is well structured (for the prefix ordering)?

References


Rémi Bonnet, Alain Finkel, and M. Praveen. Extending the Rackoff technique to Affine nets.

Ahmed Bouajjani, Javier Esparza, Alain Finkel, Oded Maler, Peter Rossmanith, Bernard

Stéphane Demri, Alain Finkel, Valentin Goranko, and Govert van Drimmelen. Model-checking

Catherine Dufourd, Alain Finkel, and Philippe Schnoebelen. Reset Nets between Decidability
of the 25th International Colloquium on Automata, Languages and Programming (ICALP’98)*,
volume 1443 of *Lecture Notes in Computer Science*, pages 103–115, Aalborg, Denmark, July

Alain Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theoretical


Alain Finkel, Stefan Goller, and Christoph Haase. Reachability in Register Machines with
Polynomial Updates. In Krishnendu Chatterjee and Jiri Sgall, editors, *Proceedings of the
38th International Symposium on Mathematical Foundations of Computer Science (MFCS’13)*,
volume 8087 of *Lecture Notes in Computer Science*, pages 409–420, Klosterneuburg, Austria,

Alain Finkel and Jérôme Leroux. How to Compose Presburger-Accelerations: Applications to

Alain Finkel, Jérôme Leroux, and Grégoire Sutre. Reachability for Two-Counter Machines
with One Test and One Reset. In Sumit Ganguly and Paritosh Pandya, editors, *Proceedings of the

FSTTCS.2018.31.

Alain Finkel, Pierre McKenzie, and Claudine Picaronny. A Well-Structured Framework for


Christoph Haase and Simon Halfon. Integer Vector Addition Systems with States. In Joël
Ouaknine, Igor Potapov, and James Worrell, editors, *Reachability Problems - 8th International

Vesa Halava. Another proof of undecidability for the correspondence decision problem - Had I

Matthias Horbach, Marco Voigt, and Christoph Weidenbach. The Universal Fragment of Presburger Arithmetic with Unary Uninterpreted Predicates is Undecidable. *CoRR*, abs/1703.01212,

