Model Checking Lossy Vector Addition Systems

Ahmed Bouajjani[1] and Richard Mayr[2]

[1] Verimag, Centre Equation, 2 avenue de Vignate, 38610 Gières, France.
Ahmed.Bouajjani@imag.fr
[2] Institut für Informatik, TU-München, Arcisstr. 21, D-80290 München, Germany.
mayrri@informatik.tu-muenchen.de

Abstract. Lossy VASS (vector addition systems with states) are defined as a subclass of VASS in analogy to lossy FIFO-channel systems. They can be used to model concurrent systems with unreliable communication. We analyze the decidability of model checking problems for lossy systems and several branching-time and linear-time temporal logics. We present an almost complete picture of the decidability of model checking for normal VASS, lossy VASS and lossy VASS with test for zero.

1 Introduction

VASS's (vector addition systems with states) can model communicating systems through unbounded unordered buffers, and hence they can be seen as abstractions of fifo-channels systems, when the ordering between messages in the channels is not relevant but only their number. Communicating systems are often analyzed under the assumption that they communicate through unreliable channels. Hence, we consider *lossy* models of communicating systems, i.e. models where messages can be lost. Recent works are about lossy unbounded fifo-channels systems [AJ93, AJ96, CFI96]. The reachability problem is decidable for these models, which implies the decidability of the verification problem for safety properties. However, liveness properties cannot be checked for lossy fifo-channel systems, unless for very special ones like single eventualities. In particular, it is impossible to model check lossy channel systems under fairness conditions. Here we study verification problems for VASS and VASS with inhibitor arcs (counter machines) under the assumption of lossiness, i.e. the contents of a place/counter can spontaneously get lower at any time.

Using the approach introduced in [CFI96, ACJT96], it can be shown very easily that the set $pre^*(S)$ of predecessors of any set of configurations S is effectively constructible for lossy VASS even with inhibitor arcs, and that this set can be represented by simple linear constraints (SC for short), where integer variables can be compared only with constants. Moreover, for lossy VASS, the set $post^*(S)$ of successors is SC definable and effectively constructible, but interestingly, for lossy VASS with inhibitor arcs these sets are not constructible although they are SC definable.

Local model checking, or simply *model checking*, consists in deciding whether a given configuration of a system satisfies a given formula of a temporal logic, and

C. Meinel and S. Tison (Eds.): STACS'99, LNCS 1563, pp. 323–333, 1999.

global model checking consists in constructing the set of all configurations that satisfy a given formula. We address these problems for a variety of linear-time and branching-time properties. We express these properties in a temporal logic, called AL (Automata Logic), which is based on automata on finite and infinite sequences to specify path properties (in the spirit of ETL), and the use of path quantifiers to express branching-time properties (like in ECTL* [Tho89]). The basic state predicates in this logic are SC constraints.

Our main positive result is that for lossy VASS, the global model checking is decidable for the logic $\exists$AL with only upward closed constraints, and dually for $\forall$AL with downward closed constraints ($\forall$AL and $\exists$AL are the universal and existential positive fragments of AL. They subsume respectively the corresponding well-known fragments $\forall$CTL* and $\exists$CTL* [GL94] of the logic CTL*). When only infinite paths are considered our decidability result also holds for normal VASS. A corollary is that linear-time properties on finite and infinite paths (on infinite paths only) are decidable for lossy VASS (normal VASS). We can even construct the set of all the configurations satisfying these properties. This generalizes the result in [Esp97] where only model checking is considered. Notice also that $\forall$AL is strictly more expressive than all linear-time temporal logics.

These decidability results break down if we relax any of the restrictions: model checking becomes undecidable if we consider $\forall$AL or $\exists$AL formulae with both downward and upward closed constraints, or if we consider lossy VASS with inhibitor arcs. Also, even if we use only propositional constraints in the logic (i.e., only constraints on control locations) the use of negation must be restricted: model checking is undecidable for CTL and lossy VASS. However, it is decidable for the fragments EF and EG of CTL even for lossy VASS with inhibitor arcs, but surprisingly, global model checking is undecidable for EG and lossy VASS (while it is decidable for EF and lossy VASS with inhibitor arcs). As a side effect we obtain that normal VASS (Petri nets) and lossy VASS with inhibitor arcs (lossy counter machines) are incomparable.

The missing proofs can be found in the full version of the paper.

2 Vector Addition Systems with States

Definition 1. *A n-dim VASS $\mathcal{S}$ is a tuple $(\Sigma, \mathcal{X}, Q, \delta)$ where Σ is a set of action labels, $\mathcal{X}$ is a set of variables such that $|\mathcal{X}| = n$, Q is a finite set of control states, δ is a finite set of transitions of the form (q_1, a, Δ, q_2) where $a \in \Sigma$, $\Delta \in \mathbb{Z}^n$.*

A *configuration* of $\mathcal{S}$ is a pair $\langle q, \boldsymbol{u}\rangle$ where $q \in Q$ and $\boldsymbol{u} \in \mathbb{N}^n$. Let $\mathcal{C}(\mathcal{S})$ be the set of configurations of $\mathcal{S}$. Given a configuration $s = \langle q, \boldsymbol{u}\rangle$, we let $State(s) = q$ and $Val(s) = \boldsymbol{u}$.

We define a *transition relation* $\longrightarrow$ on configurations as follows: $\langle q_1, \boldsymbol{u}_1\rangle \xrightarrow{a} \langle q_2, \boldsymbol{u}_2\rangle$ iff $\exists \tau = (q_1, a, \Delta, q_2) \in \delta$, $\boldsymbol{u}_2 = \boldsymbol{u}_1 + \Delta$. Let $post_\tau(\langle q_1, \boldsymbol{u}_1\rangle)$ (resp. $pre_\tau(\langle q_2, \boldsymbol{u}_2\rangle)$) denote the configuration $\langle q_2, \boldsymbol{u}_2\rangle$ (resp. $\langle q_1, \boldsymbol{u}_1\rangle$), i.e., the immediate successor (resp. predecessor) of $\langle q_1, \boldsymbol{u}_1\rangle$ (resp. $\langle q_2, \boldsymbol{u}_2\rangle$) by the transition τ. Then, we let *post* (resp. *pre*) denote the union of the $post_\tau$'s (resp. pre_τ's)

for all the transitions $\tau \in \delta$. In other words, $post(\langle q, \boldsymbol{u}\rangle) = \{\langle q', \boldsymbol{u}'\rangle \ : \ \exists a \in \Sigma.\ \langle q, \boldsymbol{u}\rangle \xrightarrow{a} \langle q', \boldsymbol{u}'\rangle\}$, and $pre(\langle q, \boldsymbol{u}\rangle) = \{\langle q', \boldsymbol{u}'\rangle \ : \ \exists a \in \Sigma.\ \langle q', \boldsymbol{u}'\rangle \xrightarrow{a} \langle q, \boldsymbol{u}\rangle\}$. Let $post^*$ and pre^* be the reflexive-transitive closures of $post$ and pre.
Given a configuration s, a *run* of the system $\mathcal{S}$ starting from s is a finite or infinite sequence $s_0 a_0 s_1 a_1 \dots s_n$ such that $s = s_0$ and, for every $i \geq 0$, $s_i \xrightarrow{a_i} s_{i+1}$. We denote by $Run_f(s, \mathcal{S})$ (resp. $Run_\omega(s, \mathcal{S})$) the set of finite (resp. infinite) runs of $\mathcal{S}$ starting from s.
A *lossy VASS* is defined as a VASS with a *weak transition relation* $\Longrightarrow$ on configurations. We define the relation $\Longrightarrow$ as follows: $\langle q_1, \boldsymbol{u}_1\rangle \stackrel{a}{\Longrightarrow} \langle q_2, \boldsymbol{u}_2\rangle$ iff $\exists \boldsymbol{u}'_1, \boldsymbol{u}'_2 \in \mathbb{N}^n$, $\boldsymbol{u}_1 \geq \boldsymbol{u}'_1$, $\langle q_1, \boldsymbol{u}'_1\rangle \xrightarrow{a} \langle q_2, \boldsymbol{u}'_2\rangle$, and $\boldsymbol{u}'_2 \geq \boldsymbol{u}_2$.
The weak transition relation induces corresponding notions of runs, successor and predecessor functions defined by considering the weak transition relation $\Longrightarrow$ instead of $\longrightarrow$.

Definition 2. *We order vectors of natural numbers by* $(u_1, \dots, u_n) \leq (v_1, \dots, v_n)$ *iff* $\forall i \in \{1, \dots, n\}.\ u_i \leq v_i$.
Given a set $S \subseteq \mathbb{N}^n$*, we denote by* $min(S)$ *the set of minimal elements of* S *w.r.t. the relation* $\leq$.
Let $S \subseteq \mathbb{N}^n$*. Then,* S *is* upward *(resp.* downward*)* closed *iff* $\forall \boldsymbol{u} \in \mathbb{N}^n.\ \boldsymbol{u} \in S \Rightarrow (\forall \boldsymbol{v} \in \mathbb{N}^n.\ \boldsymbol{v} \geq \boldsymbol{u}$ *(resp.* $\boldsymbol{v} \leq \boldsymbol{u}) \Rightarrow \boldsymbol{v} \in S)$*. Given a set* $S \subseteq \mathbb{N}^n$*, we denote by* $S{\uparrow}$ *(resp.* $S{\downarrow}$*) the upward (resp. downward) closure of* S*, i.e., the smallest upward (resp. downward) closed set which contains* S.

Lemma 3. *Every set* $S \subseteq \mathbb{N}^n$ *has a finite number of minimal elements. A set is upward closed if and only if* $S = min(S){\uparrow}$*. The union and the intersection of two upward (resp. downward) closed sets is an upward (resp. downward) closed set. The complement of an upward closed set is downward closed and vice-versa.*

Definition 4 (Simple constraints, upward/downward closed constraints). *Let* $\mathcal{X} = \{x_1, \dots, x_n\}$ *be a set of variables ranging over* $\mathbb{N}$.

1. *A* simple constraint *over* $\mathcal{X}$*, SC for short, is any boolean combination of constraints of the form* $x \geq c$ *where* $x \in \mathcal{X}$ *and* $c \in \mathbb{N} \cup \{\infty\}$.
2. *An* upward closed *(resp.* downward closed*) constraint over* $\mathcal{X}$*, UC (resp. DC) for short, is any positive boolean combination of constraints of the form* $x \geq c$ *(resp.* $x < c$*) where* $x \in \mathcal{X}$ *and* $c \in \mathbb{N} \cup \{\infty\}$.

Constraints are interpreted in the standard way as a subset of $\mathbb{N}^n$ ($\leq$ is the usual ordering and $<$ is the strict inequality). Given a simple constraint ξ, we let $[\![\xi]\!]$ denote the set of vectors in $\mathbb{N}^n$ satisfying ξ. Notice that the constraints $x < 0$ and $x \geq \infty$ correspond to $\emptyset$ and that $x \geq 0$ and $x < \infty$ correspond to $\mathbb{N}$.

Definition 5. *A set* S *is SC (resp. UC, DC) definable if there exists an SC (resp. UC, DC)* ξ *such that* $S = [\![\xi]\!]$.

Definition 6 (Normal forms).
1. *A* canonical product *is a constraint of the form* $\boldsymbol{\ell} \leq \boldsymbol{x} \leq \boldsymbol{u}$,
2. *A* canonical upward closed product *is a constraint of the form* $\boldsymbol{\ell} \leq \boldsymbol{x}$,
3. *A* canonical downward closed product *is a constraint of the form* $\boldsymbol{x} \leq \boldsymbol{u}$,

where $\boldsymbol{\ell} \in \mathbb{N}^n$ *and* $\boldsymbol{u} \in (\mathbb{N} \cup \{\infty\})^n$.
A SC (resp. UC, DC) in normal form *is either* $\emptyset$, *or a finite disjunction of canonical (resp. canonical upward closed, canonical downward closed) products.*

Lemma 7. *Every SC (resp. UC, DC) is equivalent to a SC (UC, DC) in normal form.*

Proposition 8. *SC definable sets are closed under boolean operations, and UC definable sets as well as DC definable sets are closed under union and intersection. The complement of a UC definable set is a DC definable set and vice-versa. A subset of* $\mathbb{N}^n$ *is UC definable (resp. DC definable) if and only if it is an upward (resp. downward) closed set. A set is SC definable if and only if it is a boolean combination of upward closed sets.*

Let $\mathcal{S} = (\Sigma, \mathcal{X}, Q, \delta)$ be a n-dim VASS with $Q = \{q_1, \ldots, q_m\}$. Then, every set of configurations of $\mathcal{S}$ is defined as a union $C = \{q_1\} \times S_1 \cup \cdots \cup \{q_m\} \times S_m$ where the S_i's are sets of n-dim vectors of natural numbers. The set of configurations C is SC (resp. UC, DC) definable if all the S_i's are SC (resp. UC, DC) definable. We represent SC definable sets by simple constraints in normal form coupled with control states. From now on, we consider a canonical product to be a pair of the form $\langle q, \boldsymbol{\ell} \leq \boldsymbol{x} \leq \boldsymbol{u} \rangle$ where $q \in Q$. A simple constraint is either $\emptyset$ or a finite disjunction of canonical products. We use $\mathrm{SC}(Q, \mathcal{X})$ (resp. $\mathrm{UC}(Q, \mathcal{X})$, $\mathrm{DC}(Q, \mathcal{X})$) to denote the set of simple constraints (resp. upward closed, downward closed constraints). We omit the parameters Q and $\mathcal{X}$ when they are known from the context.

3 Computing Successors and Predecessors

Lemma 9. *The class SC is effectively closed under the operations post and pre for any lossy VASS's.*

Proof. These operations are distributive w.r.t. union. Hence, it suffices to consider separately each transition $\tau = (q, a, \Delta, q')$ and perform them on canonical products:
1. $post_\tau(\langle q, \boldsymbol{\ell} \leq \boldsymbol{x} \leq \boldsymbol{u} \rangle) = \langle q', \boldsymbol{x} \leq \boldsymbol{u} + \Delta \rangle$.
2. $pre_\tau(\langle q', \boldsymbol{\ell} \leq \boldsymbol{x} \leq \boldsymbol{u} \rangle) = \langle q, (\boldsymbol{\ell} - \Delta) \sqcap \boldsymbol{0} \leq \boldsymbol{x} \rangle$,

where $\forall \boldsymbol{u}, \boldsymbol{v} \in \mathbb{N}^n$, $\boldsymbol{u} \sqcap \boldsymbol{v}$ is the vector such that $\forall i \in \{1, \ldots, n\}$. $(\boldsymbol{u} \sqcap \boldsymbol{v})_i = max(u_i, v_i)$. □

Notice that for lossy VASS's, the *pre* image of any set of configurations is upward closed and its *post* image is downward closed. This also holds for pre^* and $post^*$.

Theorem 10. *For every n-dim lossy VASS $\mathcal{S}$, and every n-dim SC set S, the set $pre^*(S)$ is UC definable and effectively constructible.*

Proof. Since the set $pre^*(S)$ is upward closed, by Proposition 8 we deduce that it is UC definable. The construction of this set is similar to the one given in [CFI96, ACJT96] for lossy channel systems. □

Theorem 11. *For every n-dim lossy VASS $\mathcal{S}$, and every n-dim SC set S, the set $post^*(S)$ is DC definable and effectively constructible.*

Proof. Since $post^*(S)$ is downward closed, by Proposition 8 we deduce that $post^*(S)$ is DC definable. This set can be constructed using the Karp-Miller algorithm for the construction of the coverability graph [KM69]. □

4 Automata and Automata Logic

We use finite automata to express properties of computations. These automata are labeled on states and edges as well. State labels are associated with predicates on the configurations of a given system and edge labels are associated with the actions of the system.

Definition 12. *Let Λ and Σ be two finite alphabets. A labeled transition graph over (Λ, Σ) is a tuple $\mathcal{G} = (Q, q_{init}, \Pi, \delta)$ where Q is a finite set of states, q_{init} is the initial state, $\Pi : Q \to \Lambda$ is a state labeling function, $\delta \subseteq Q \times \Sigma \times Q$ is a finite set of labeled transitions. We write $q \xrightarrow{a} q'$ when $(q, a, q') \in \delta$.*
Given a state q, a run of $\mathcal{G}$ starting from q is a finite or infinite sequence $q_0 a_0 q_1 a_1 q_2 \ldots$ such that $q_0 = q$ and $\forall i \geq 0.\ q_i \xrightarrow{a_i} q_{i+1}$.

Definition 13 (Automata on finite sequences). *A finite-state automaton over (Λ, Σ) on finite sequences is a tuple $\mathcal{A}_f = (Q, q_{init}, \Pi, \delta, F)$ where $(Q, q_{init}, \Pi, \delta)$ is a labeled transition graph over (Λ, Σ), and $F \subseteq Q$ is a set of final states. A finite sequence $\lambda_0 a_0 \lambda_1 a_1 \ldots \lambda_n \in \Lambda(\Sigma\Lambda)^*$ is* accepted *by $\mathcal{A}_f$ if there is a run $q_0 a_0 q_1 a_1 \ldots q_n$ of $\mathcal{A}_f$ starting from q_{init} such that $\forall i \in \{0, \ldots, n\}$. $\Pi(q_i) = \lambda_i$, and $q_n \in F$. Let $L(\mathcal{A}_f)$ be the set of sequences in $\Lambda(\Sigma\Lambda)^*$ accepted by $\mathcal{A}_f$.*

Definition 14 (Büchi ω-automata). *A finite-state Büchi automaton over (Λ, Σ) is a tuple $\mathcal{A}_\omega = (Q, q_{init}, \Pi, \delta, F)$ where $(Q, q_{init}, \Pi, \delta)$ is a labeled transition graph over (Λ, Σ), and $F \subseteq Q$ is a set of repeating states. An infinite sequence $\lambda_0 a_0 \lambda_1 a_1 \ldots \lambda_n \in (\Lambda\Sigma)^\omega$ is* accepted *by $\mathcal{A}_\omega$ if there is a run $q_0 a_0 q_1 a_1 \ldots$ of $\mathcal{A}_\omega$ starting from q_{init} such that $\forall i \geq 0.\ \Pi(q_i) = \lambda_i$, and $\overset{\infty}{\exists}\, i \geq 0.\ q_i \in F$. We denote by $L(\mathcal{A}_\omega)$ the set of sequences in $(\Lambda\Sigma)^\omega$ accepted by $\mathcal{A}_\omega$.*

Definition 15 (Closed ω-automata). *A closed ω-automaton is a Büchi automaton $\mathcal{A}_{\omega c} = (Q, q_{init}, \Pi, \delta, F)$ such that $F = Q$.*

Remark. [Tho90] Büchi automata define ω-regular sets of infinite sequences. They are closed under boolean operations. Closed ω-automata define closed ω-regular sets in the Cantor topology (the class F in the Borel hierarchy). They correspond to the class of ω-regular safety properties. Closed ω-automata are closed under intersection and union, but not under complementation.

We introduce an automata-based branching-time temporal logic called AL (Automata Logic). This logic is defined in the spirit of the extended temporal logic ETL and is an extension of ECTL* [Tho89]. The logic AL is more expressive than CTL and CTL*, and allows to express all ∞-regular linear-time properties on finite and infinite computations.

Definition 16 (Automata Logic). *Given a set of control states Q and a set of variables $\mathcal{X}$, we let $\mathcal{F}$ denote a subset of $SC(Q,\mathcal{X})$, and we let π range over elements of $\mathcal{F}$. Then, the set of AL($\mathcal{F}$) formulae is defined by the following grammar:*

$$\varphi ::= \pi \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists\mathcal{A}_f(\varphi_1,\ldots,\varphi_m) \mid \forall\mathcal{A}_f(\varphi_1,\ldots,\varphi_m) \mid \exists\mathcal{A}_\omega(\varphi_1,\ldots,\varphi_m) \mid \forall\mathcal{A}_\omega(\varphi_1,\ldots,\varphi_m)$$

where $\mathcal{A}_f$ (resp. $\mathcal{A}_\omega$) is a finite-state automaton on finite (resp. infinite) sequences over $(\Lambda = \{\lambda_1,\ldots,\lambda_m\}, \Sigma)$. We consider standard abbreviations like $\Rightarrow$.

Definition 17. *We use $\star$ to denote f or ω. Let $\mathcal{S} = (\Sigma, \mathcal{X}, Q, \delta)$ be a n-dim (lossy) VASS, We define a satisfaction relation between configurations of $\mathcal{S}$ and AL($\mathcal{F}$) as follows:*

$$\begin{aligned}
s \models (q,\xi) &\text{ iff } State(s) = q \text{ and } Val(s) \in [\![\xi]\!] \\
s \models \neg\varphi &\text{ iff } s \not\models \varphi \\
s \models \varphi_1 \vee \varphi_2 &\text{ iff } s \models \varphi_1 \text{ or } s \models \varphi_2 \\
s \models \varphi_1 \wedge \varphi_2 &\text{ iff } s \models \varphi_1 \text{ and } s \models \varphi_2 \\
s \models \exists\mathcal{A}_\star(\varphi_1,\ldots,\varphi_m) &\text{ iff } \exists\rho = s_0 a_0 \ldots \in Run_\star(s,\mathcal{S}).\ \exists\sigma = \lambda_{i_0} a_0 \ldots \in L(\mathcal{A}_\star). \\
&\qquad |\sigma| = |\rho| \text{ and } \forall j.\ 0 \le j < |\rho|.\ s_j \models \varphi_{i_j} \\
s \models \forall\mathcal{A}_\star(\varphi_1,\ldots,\varphi_m) &\text{ iff } \forall\rho = s_0 a_0 \ldots \in Run_\star(s,\mathcal{S}).\ \exists\sigma = \lambda_{i_0} a_0 \ldots \in L(\mathcal{A}_\star) \\
&\qquad |\sigma| = |\rho| \text{ and } \forall j.\ 0 \le j < |\rho|.\ s_j \models \varphi_{i_j}
\end{aligned}$$

For every formula φ, let $[\![\varphi]\!]_{\mathcal{S}} := \{s \in \mathcal{S} \mid s \models \varphi\}$.

Definition 18 (Fragments of AL). *$\exists$AL($\mathcal{F}$) is the fragment of AL that uses only constraints from $\mathcal{F}$, conjunction, disjunction and existential path quantification. $\forall$AL($\mathcal{F}$) is the fragment of AL that uses only constraints from $\mathcal{F}$, conjunction, disjunction and universal path quantification. Let X be (some fragment of) the logic AL. Then X_f (resp. X_ω, $X_{\omega c}$) denote the fragment of X where only automata on finite sequences (resp. Büchi, closed ω-automata) are used.*

AL is a weaker logic than the modal μ-calculus, but many widely known temporal logics are fragments of AL. Every propositional linear-time property, in particular LTL properties, can be expressed in AL. CTL* is a fragment of AL since every path formula in CTL* corresponds to an LTL formula. Thus, CTL is also a fragment of AL. Clearly, $\forall$AL and $\exists$AL subsume the positive universal and existential fragments of CTL* denoted $\forall$CTL* and $\exists$CTL* (notice that LTL is a fragment of $\forall$CTL*).

We consider two fragments of CTL called EF and EG. The logic EF uses SC predicates, boolean operators, the one-step next operator and the operator EF which is defined by $[\![EF\varphi]\!] = pre^*([\![\varphi]\!])$, The logic EG is defined like EF, except that the operator EF is replaced by the operator EG, which is defined as follows: $s \models EG\varphi$ iff there exists a complete run that starts at s and always satisfies φ. By a complete run we mean either an infinite run or a finite run ending in a deadlock. We use the subscripts f or ω to denote the fragments of these logics obtained by interpreting their formulae on either finite or infinite paths only. Then, it can be seen that $\mathrm{EF} = \mathrm{EF}_f \subseteq \mathrm{CTL}_f \subseteq \mathrm{CTL}^*_f \subseteq \mathrm{AL}_f$. It can also be seen that EG_ω is a fragment of $\mathrm{AL}_{\omega c}$ but EG is not (due to the finite paths).

5 Model Checking

Definition 19 (Model checking and global model checking problems).

1. *The* model checking problem *is if $s \in [\![\varphi]\!]_{\mathcal{S}}$ for configuration s and formula φ.*
2. *The* global model checking problem *is whether for any formula φ the set $[\![\varphi]\!]_{\mathcal{S}}$ is effectively constructible.*

Lemma 20. *Let $\mathcal{S}$ be a lossy VASS. Then for every formula φ of the form $\exists \mathcal{A}_f(\pi_1, \ldots, \pi_m)$ where all the π_i are SC, the set $[\![\varphi]\!]_{\mathcal{S}}$ is SC definable and effectively constructible.*

Proof. By a generalized pre* construction (see Theorem 10). □

Theorem 21. *The global model checking problem for lossy VASS and the logic AL_f is decidable.*

Proof. By induction on the nesting-depth and Lemma 20. □

The following results even hold for non-lossy VASS. The aim is to show decidability of the global model checking problem for VASS and the logic $\exists AL_\omega(UC)$. We define a generalized notion of configurations of VASS which includes the symbol ω. This symbol denotes arbitrarily high numbers of tokens on a place. It is used as an abbreviation in the following way: $\langle q, (\omega, \omega, \ldots, \omega, x_{k+1}, \ldots, x_n)\rangle \models \varphi :\iff \exists n_1, \ldots, n_k \in \mathbb{N}.\ \langle q, (n_1, n_2, \ldots, n_k, x_{k+1}, \ldots, x_n)\rangle \models \varphi$. (Of course the ω can occur at any position, e.g. $\langle q, (x_1, x_2, \omega, x_4, \omega, x_6)\rangle$.)

Lemma 22. *Let $\mathcal{S}$ be a VASS and φ a formula of the form $\exists \mathcal{A}_\omega(\pi_1, \ldots, \pi_m)$ where all the π_i are in UC. Let s be a generalized configuration of $\mathcal{S}$ (i.e. it can contain ω). It is decidable if $s \models \varphi$.*

Proof. (Sketch) First construct the Karp-Miller coverability graph [KM69]. Then check for the existence of cycles in this graph that have an overall positive effect of the fired transitions. These cycles may contain the same node several times. This check is done with the help of Parikh's Theorem. The property holds iff such a cycle with overall positive effect exists, because it can be repeated infinitely often. □

Lemma 23. *Let $\mathcal{S}$ be a VASS and φ a formula of the form $\exists \mathcal{A}_\omega(\pi_1, \ldots, \pi_m)$ where all the π_i are in UC. The set $[\![\varphi]\!]_{\mathcal{S}}$ is UC definable and effectively constructible.*

Proof. $[\![\varphi]\!]_{\mathcal{S}}$ is upward closed, because all π_i are upward closed. Thus, it is characterized by the finite set of its minimal elements (see Lemma 3). To find the minimal elements, we use a construction that was described by Valk and Jantzen in [VJ85]. The important point here is that we can use Lemma 22 to check the existence of configurations that satisfy φ. For example, if $\langle q, (\omega, x_2, x_3)\rangle \models \varphi$ then we can check if $\langle q, (n_1, x_2, x_3)\rangle \models \varphi$ for $n_1 = 0$, $n_1 = 1$, $n_1 = 2$, ... until we find the minimal n_1 s.t. $\langle q, (n_1, x_2, x_3)\rangle \models \varphi$. □

Theorem 24. *The global model checking problem is decidable for VASS and the logic $\exists AL_\omega(UC)$.*

Proof. By induction on the nesting-depth of the formula and Lemma 23. □

Theorem 25. *The global model checking problem is decidable for lossy VASS and the logic $\exists AL(UC)$.*

Proof. By induction on the nesting depth and Theorems 21 and 24. □

Theorem 26. *The model checking problem for lossy VASS and $AL_{\omega c}$ is decidable.*

Proof. By induction on the nesting-depth of the formula and an analysis of all computations which is finite by Dickson's Lemma. □

Theorem 27. *Model checking lossy VASS with the logic EG is decidable.*

Theorems 26 and 27 say that the model checking problem is decidable for a lossy VASS and an EG-formula/ $AL_{\omega c}$-formula φ. However, in both cases the set $[\![\varphi]\!]_{\mathcal{S}}$ is not effectively constructible (although it is SC definable). If it were constructible then Lemma 20 could be used to decide model checking lossy VASS with formulae of the form $EFEG_\omega \pi$, where π is a constraint in SC. However, this problem has very recently been shown to be undecidable.

Proposition 28. *Model checking lossy VASS with formulae of the form $EFEG_\omega\pi$, where π is a constraint in SC is undecidable.*
Proof. This is a corollary of a more general undecidability result for lossy BPP (Basic Parallel Processes), which follows (not immediately) from the result on lossy counter machines in Proposition 30 (see [May98]). □

Remark. This undecidability result also implies undecidability of model checking lossy VASS with the logic $\exists AL_\omega$. One can encode properties of the form $EFEG_\omega\pi$ in $\exists AL_\omega$ in the following way: Let $\mathcal{A}_\omega$ be an automaton with states q, q', and transitions $q \to q$, $q \to q'$ and $q' \to q'$ which are labeled with any action. The predicate $true$ is assigned to q and the predicate π is assigned to q'. q is the initial state and q' is the only repeating state. Let $\mathcal{A}'_\omega$ be an automaton with only one state q which is the initial state and repeating and a transition $q \to q$ with any action. The predicate π is assigned to q. Then for any lossy VASS s we have $s \models EFEG_\omega\pi \iff s \models \mathcal{A}_\omega(true, \pi) \vee \mathcal{A}'_\omega(\pi)$.

Lossy VASS can be extended with inhibitor arcs. This means introducing transitions that can only fire if some defined places are empty (i.e. they can test for zero). Thus lossy VASS with inhibitor arcs are equivalent to lossy counter machines. Normal VASS with inhibitor arcs are Turing-powerful, but lossy VASS with inhibitor arcs are not.

Theorem 29. *For lossy VASS with inhibitor arcs*
1. *the global model checking problem is decidable for the logic AL_f.*
2. *model checking is decidable for the logics $AL_{\omega c}$ and EG.*

Inhibitor arcs can never keep a transition from firing, because one can just loose the tokens on the places that inhibit it. However, after such a transition has fired, the number of tokens on the inhibiting places is fixed and known exactly. Such a guarantee is impossible to achieve in lossy VASS without inhibitor arcs. Thus not all results for lossy VASS carry over to lossy VASS with inhibitor arcs.

Proposition 30. *Let $\mathcal{S}$ be a lossy VASS with inhibitor arcs. It is undecidable if there exists an initial configuration s s.t. there is an infinite run of $(s, \mathcal{S})$.*
Proof. This is a corollary of a more general undecidability result for lossy counter machines in [May98]. The main idea is that one can enforce that lossiness occurs only finitely often in the infinite run. □

Theorem 31. *Model checking lossy VASS with inhibitor arcs with the logic LTL is undecidable.*
Proof. We reduce the problem of Proposition 30 to the model checking problem. We construct a lossy VASS with inhibitor arcs $\mathcal{S}'$ that does the following: First it guesses an arbitrary configuration s of $\mathcal{S}$ doing only the atomic action a. Then it simulates $\mathcal{S}$ on s doing only the atomic action b. Let $\mathcal{A}_\omega$ be a Büchi-automaton with initial state q and repeating state q' and transitions $q \xrightarrow{a} q$, $q \xrightarrow{b} q'$ and $q' \xrightarrow{b} q'$. Let s' be the initial state of $\mathcal{S}'$. We have reduced the question of Proposition 30 to the question if $(s', \mathcal{S}') \models \exists\mathcal{A}_\omega(true, true)$. This question can be expressed in LTL. □

It follows immediately that model checking lossy VASS with inhibitor arcs with $AL_\omega(UC)$ is undecidable. It is interesting to compare this result with Proposition 28. For undecidability it suffices to have either inhibitor arcs in the system or downward closed constraints in the logic. One can be encoded in the other and vice versa. The set $post^*(s)$ is DC definable since it is downward closed. However, it is not constructible for lossy VASS with inhibitor arcs (unlike for lossy VASS, see Theorem 11).

Theorem 32. *$post^*(s)$ is not constructible for lossy VASS with inhibitor arcs.*

Proof. Boundedness is undecidable for reset Petri nets [DFS98]. This result carries over to lossy reset Petri nets. Lossy VASS with inhibitor arcs can simulate lossy reset Petri nets. It follows that boundedness is undecidable for lossy VASS with inhibitor arcs and thus $post^*(s)$ is not constructible. □

6 Conclusion

We have established results for normal VASS and lossy VASS with inhibitor arcs (lossy counter machines). Interestingly, it turns out that these two models are incomparable. Moreover, all the positive/negative results we obtained for lossy VASS with inhibitor arcs are the same as for lossy fifo-channel systems. Note that lossy fifo-channel systems can simulate lossy VASS with inhibitor arcs, but only with some additional deadlocks.
The following table summarizes the results on the decidability of model checking for VASS, lossy VASS with test for zero, lossy VASS and lossy fifo-channel systems. By '++' we denote the fact that for any formula φ the set $[\![\varphi]\!]$ is SC definable and effectively constructible (global model checking), while '+' means that only model checking is decidable. We denote by — that model checking is undecidable. The symbol '?' denotes an open problem.

Logic	**VASS**	**Lossy VASS+0**	**Lossy VASS**	**Lossy FIFO**
AL_f/EF	— [Esp97]	++	++ [AJ93]	++ [AJ93]
$\exists AL_\omega(UC)/LTL$	++/+[Esp97]	—	++	— [AJ96]
$\exists AL(UC)$	?	—	++	— [AJ96]
$AL_{\omega c}$/EG	— [EK95]	+	+ [AJ93]	+ [AJ93]
$\exists AL_\omega$/CTL	— [EK95]	—	—	— [AJ96]

The results in this table are new, except where references are given. For normal VASS and LTL, decidability of the model checking problem was known [Esp97], but the construction of the set $[\![\varphi]\!]$ is new. The results in [AJ93] are just about EF and EG formulae without nesting, not for the full logics AL_f and $AL_{\omega c}$.

Acknowledgment: We thank Peter Habermehl for interesting discussions.

References

[ACJT96] P. Abdulla, K. Cerans, B. Jonsson, and Y-K. Tsay. General Decidability Theorems for Infinite-state Systems. In *LICS'96*. IEEE, 1996.

[AJ93] P. Abdulla and B. Jonsson. Verifying Programs with Unreliable Channels. In *LICS'93*. IEEE, 1993.

[AJ96] P. Abdulla and B. Jonsson. Undecidable verification problems for programs with unreliable channels. *Information and Computation*, 130(1):71–90, 1996.

[CFI96] Gérard Cécé, Alain Finkel, and S. Purushothaman Iyer. Unreliable Channels Are Easier to Verify Than Perfect Channels. *Information and Computation*, 124(1):20–31, 1996.

[DFS98] C. Dufourd, A. Finkel, and Ph. Schnoebelen. Reset nets between decidability and undecidability. In *Proc. of ICALP'98*, volume 1443 of *LNCS*. Springer Verlag, 1998.

[EK95] J. Esparza and A. Kiehn. On the model checking problem for branching time logics and Basic Parallel Processes. In *CAV'95*, volume 939 of *LNCS*, pages 353–366. Springer Verlag, 1995.

[Esp97] J. Esparza. Decidability of model checking for infinite-state concurrent systems. *Acta Informatica*, 34:85–107, 1997.

[GL94] O. Grumberg and D. Long. Model Checking and Modular Verification. *ACM Transactions on Programming Languages and Systems*, 16, 1994.

[KM69] R. Karp and R. Miller. Parallel program schemata. *JCSS*, 3, 1969.

[May98] R. Mayr. Lossy counter machines. Technical Report TUM-I9827, TU-München, October 1998. wwwbrauer.informatik.tu-muenchen.de/∼mayrri.

[Tho89] W. Thomas. Computation Tree Logic and Regular ω-Languages. LNCS 354, 1989.

[Tho90] W. Thomas. Automata on Infinite Objects. In *Handbook of Theo. Comp. Sci.* Elsevier Sci. Pub., 1990.

[VJ85] R. Valk and M. Jantzen. The Residue of Vector Sets with Applications to Decidability Problems in Petri Nets. *Acta Informatica*, 21, 1985.