Undecidable Problems in Unreliable Computations

Richard Mayr*

Department of Computer Science, University of Edinburgh,
JCMB, Edinburgh EH9 3JZ, UK. e-mail: mayrri@dcs.ed.ac.uk

Abstract. Lossy counter machines are defined as Minsky n-counter machines where the values in the counters can spontaneously decrease at any time. While termination is decidable for lossy counter machines, structural termination (termination for every input) is undecidable. This undecidability result has far reaching consequences. Lossy counter machines can be used as a general tool to prove the undecidability of many problems, for example (1) The verification of systems that model communication through unreliable channels (e.g. model checking lossy fifo-channel systems and lossy vector addition systems). (2) Several problems for reset Petri nets, like structural termination, boundedness and structural boundedness. (3) Parameterized problems like fairness of broadcast communication protocols.

1 Introduction

Lossy counter machines (LCM) are defined just like Minsky counter machines [19], but with the addition that the values in the counters can spontaneously decrease at any time. This is called 'lossiness', since a part of the counter is lost. (In a different framework this corresponds to lost messages in unreliable communication channels.) There are many different kinds of lossiness, i.e. different ways in which the counters can decrease. For example, one can define that either a counter can only spontaneously decrease by 1, or it can only become zero, or it can change to any smaller value. All these different ways are described by different *lossiness relations* (see Section 2).

The addition of lossiness to counter machines weakens their computational power. Some types of lossy counter machines (with certain lossiness relations) are not Turing-powerful, since reachability and termination are decidable for them. Since lossy counter machines are *weaker* than normal counter machines, any *undecidability* result for lossy counter machines is particularly interesting.

The main result of this paper is that *structural termination* (termination for every input) is undecidable for every type of lossy counter machine (i.e. for every lossiness relation).

This result can be applied to prove the undecidability of many problems. To prove the undecidability of a problem X, it suffices to choose a suitable lossiness relation L and reduce the structural termination problem for lossy counter machines with lossiness relation L to the problem X. The important and nice point here is that problem X does *not* need to simulate a counter machine perfectly. Instead, it suffices if X can simulate a counter machine imperfectly, by simulating only a lossy counter machine. Furthermore, one can choose the right type of imperfection (lossiness) by choosing the lossiness relation L.

Thus lossy counter machines can be used as a general tool to prove the undecidability of problems. Firstly, they can be used to prove new undecidability results, and secondly they can be used to give more elegant, simpler and much shorter proofs of existing results (see Section 5).

* Work supported by DAAD Post-Doc grant D/98/28804.

2 Definitions

Definition 1. *A n-counter machine [19] M is described by a finite set of states Q, an initial state $q_0 \in Q$, a final state $accept \in Q$, n counters $c_1, \ldots, c_n$ and a finite set of instructions of the form $(q : \ c_i := c_i + 1; \mathsf{goto}\ q')$ or $(q : \ \mathsf{If}\ c_i = 0\ \mathsf{then}\ \mathsf{goto}\ q'\ \mathsf{else}\ c_i := c_i - 1; \mathsf{goto}\ q'')$ where $i \in \{1, \ldots, n\}$ and $q, q', q'' \in Q$. A* configuration *of M is described by a tuple $(q, m_1, \ldots, m_n)$ where $q \in Q$ and $m_i \in \mathbb{N}$ is the content of the counter c_i ($1 \leq i \leq n$). The size of a configuration is defined by $size((q, m_1, \ldots, m_n)) := \sum_{i=1}^{n} m_i$. The possible computation steps are defined by*

1. $(q, m_1, \ldots, m_n) \to (q', m_1, \ldots, m_i + 1, \ldots, m_n)$ *if there is an instruction* $(q : \ c_i := c_i + 1; \mathsf{goto}\ q')$.
2. $(q, m_1, \ldots, m_n) \to (q', m_1, \ldots, m_n)$ *if there is an instruction* $(q : \ \mathsf{If}\ c_i = 0\ \mathsf{then}\ \mathsf{goto}\ q' \mathsf{else}\ c_i := c_i - 1; \mathsf{goto}\ q'')$ *and* $m_i = 0$.
3. $(q, m_1, \ldots, m_n) \to (q'', m_1, \ldots, m_i - 1, \ldots, m_n)$ *if there is an instruction* $(q : \ \mathsf{If}\ c_i = 0\ \mathsf{then}\ \mathsf{goto}\ q' \mathsf{else}\ c_i := c_i - 1; \mathsf{goto}\ q'')$ *and* $m_i > 0$.

A *run* of a counter machine is a (possibly infinite) sequence of configurations $s_0, s_1, \ldots$ with $s_0 \to s_1 \to s_2 \to s_3 \to \ldots$. *Lossiness relations* describe spontaneous changes in the configurations of lossy counter machines.

Definition 2. *Let $\xrightarrow{s}$ (for 'sum') be a relation on configurations of n-counter machines*

$$(q, m_1, \ldots, m_n) \xrightarrow{s} (q', m'_1, \ldots, m'_n) \ :\Leftrightarrow \ (q, m_1, \ldots, m_n) = (q', m'_1, \ldots, m'_n) \vee \left(q = q' \wedge \sum_{i=1}^{n} m_i > \sum_{i=1}^{n} m'_i \right)$$

This relation means that either nothing is changed or the sum of all counters strictly decreases. Let id be the identity relation. A relation $\xrightarrow{l}$ is a lossiness relation *iff $id \subseteq \xrightarrow{l} \subseteq \xrightarrow{s}$. A* lossy counter machine *(LCM) is given by a counter machine M and a lossiness relation $\xrightarrow{l}$. Let $\to$ be the normal transition relation of M. The lossy transition relation $\Longrightarrow$ of the LCM is defined by $s_1 \Longrightarrow s_2 \ :\Leftrightarrow \ \exists s'_1, s'_2.\ s_1 \xrightarrow{l} s'_1 \to s'_2 \xrightarrow{l} s_2$. An* arbitrary lossy counter machine *is a lossy counter machine with an arbitrary (unspecified) lossiness relation. The following relations are lossiness relations:*

Perfect *The relation id is a lossiness relation. Thus arbitrary lossy counter machines subsume normal counter machines.*

Classic Lossiness *The* classic lossiness relation *$\xrightarrow{cl}$ is defined by $(q, m_1, \ldots, m_n) \xrightarrow{cl} (q', m'_1, \ldots, m'_n) \ :\Leftrightarrow \ q = q' \wedge \forall i.\, m_i \geq m'_i$. Here the contents of the counters can become any smaller value. A relation $\xrightarrow{l}$ is called a* subclassic lossiness relation *iff $id \subseteq \xrightarrow{l} \subseteq \xrightarrow{cl}$.*

Bounded Lossiness *A counter can lose at most $x \in \mathbb{N}$ before and after every computation step. Here the lossiness relation $\xrightarrow{l(x)}$ is defined by $(q, m_1, \ldots, m_n) \xrightarrow{l(x)} (q', m'_1, \ldots, m'_n) \ :\Leftrightarrow \ q = q' \wedge \forall i.\, m_i \geq m'_i \geq max\{0, m_i - x\}$. Note that $\xrightarrow{l(x)}$ is a subclassic lossiness relation.*

Reset Lossiness *If a counter is tested for zero, then it can suddenly become zero. The lossiness relation $\xrightarrow{rl}$ is defined as follows: $(q, m_1, \ldots, m_n) \xrightarrow{rl} (q', m'_1, \ldots, m'_n)$*

iff $q = q'$ *and for all* i *either* $m_i = m'_i$ *or* $m'_i = 0$ *and there is an instruction* $(q : \mathsf{If}\ c_i = 0\ \mathsf{then\ goto}\ q'\ \mathsf{else}\ c_i := c_i - 1; \mathsf{goto}\ q'')$. *Note that* $\xrightarrow{rl}$ *is subclassic.*

The definition of these lossiness relations carries over to other models like Petri nets [21], where places are considered instead of counters.

Definition 3. *For any arbitrary lossy* n*-counter machine and any configuration* s *let* $runs(s)$ *be the set of runs that start at configuration* s*. (There can be more than one run if the counter machine is nondeterministic or lossy.) Let* $runs^\omega(s)$ *be the set of infinite runs that start at configuration* s*. A run* $r = \{(q^i, m_1^i, \ldots, m_n^i)\}_{i=0}^\infty \in runs^\omega(s)$ *is* space-bounded *iff* $\exists c \in \mathbb{N}. \forall i.\ \sum_{j=1}^n m_j^i \leq c$*. Let* $runs_b^\omega(s)$ *be the space-bounded infinite runs that start at* s*. An (arbitrary lossy)* n*-counter machine* M *is*

zero-initializing *iff in the initial state* q_0 *it first sets all counters to 0.*
space-bounded *iff the space used by* M *is bounded by a constant* c*.* $\exists c \in \mathbb{N}. \forall r \in runs((q_0, 0, \ldots, 0)). \forall s \in r.\ size(s) \leq c$
input-bounded *iff in every run from any configuration the size of every reached configuration is bounded by the input.* $\forall s. \forall r \in runs(s). \forall s' \in r.\ size(s') \leq size(s)$
strongly-cyclic *iff every infinite run from any configuration visits the initial state* q_0 *infinitely often.* $\forall q \in Q, m_1, \ldots, m_n \in \mathbb{N}. \forall r \in runs^\omega((q, m_1, \ldots, m_n)).$ $\exists m'_1, \ldots, m'_n \in \mathbb{N}.\ (q_0, m'_1, \ldots, m'_n) \in r$*.*
bounded-strongly-cyclic *iff every space-bounded infinite run from any configuration visits the initial state* q_0 *infinitely often.* $\forall q \in Q, m_1, \ldots, m_n \in \mathbb{N}.$ $\forall r \in runs_b^\omega((q, m_1, \ldots, m_n)). \exists m'_1, \ldots, m'_n \in \mathbb{N}.\ (q_0, m'_1, \ldots, m'_n) \in r$

If M *is input-bounded then it is also space-bounded. If* M *is strongly-cyclic then it is also bounded-strongly-cyclic. If* M *is input-bounded and bounded-strongly-cyclic then it is also strongly-cyclic.*

3 Decidable Properties

Since arbitrary LCM subsume normal counter machines, nothing is decidable for them. However, some problems are decidable for classic LCM (with the classic lossiness relation). They are not Turing-powerful. The following results in this section are special cases of positive decidability results in [4, 5, 2].

Lemma 4. *Let* M *be a classic LCM and* s *a configuration of* M*. The set* $pre^*(s) := \{s' \mid s' \Longrightarrow^* s\}$ *of predecessors of* s *is effectively constructible.*

Theorem 5. *Reachability is decidable for classic LCM.*

Lemma 6. *Let* M *be a classic LCM with initial configuration* s_0*. It is decidable if there is an infinite run that starts at* s_0*, i.e. if* $runs^\omega(s_0) \neq \emptyset$*.*

Theorem 7. *Termination is decidable for classic LCM.*

It has been shown in [4] that even model checking classic LCM with the temporal logics EF and EG (natural fragments of computation tree logic (CTL) [7, 10]) is decidable.

4 The Undecidability Result

We show that structural termination is undecidable for LCM for every lossiness relation. We start with the problem CM, which was shown to be undecidable by Minsky [19].

CM

Instance: A 2-counter machine M with initial state q_0.
Question: Does M accept $(q_0, 0, 0)$?

BSC-ZI-CM$_b^\omega$

Instance: A bounded-strongly-cyclic, zero-initializing 3-counter machine M with initial state q_0.
Question: Does M have an infinite space-bounded run from $(q_0, 0, 0, 0)$, i.e. $runs_b^\omega((q_0, 0, 0, 0)) \neq \emptyset$?

Lemma 8. *BSC-ZI-CM$_b^\omega$ is undecidable.*

Proof. We reduce CM to BSC-ZI-CM$_b^\omega$. Let M be a 2-counter machine with initial state q_0. We construct a 3-counter machine M' as follows: First M' sets all three counters to 0. Then it does the same as M, except that after every instruction it increases the third counter c_3 by 1. Every instruction of M of the form $(q : c_i := c_i + 1;\ \mathsf{goto}\ q')$ with $(1 \leq i \leq 2)$ is replaced by $q : c_i := c_i + 1;\ \mathsf{goto}\ q_2$ and $q_2 : c_3 := c_3 + 1;\ \mathsf{goto}\ q'$, where q_2 is a new state. Every instruction of the form $(q : \mathsf{If}\ c_i = 0\ \mathsf{then}\ \mathsf{goto}\ q'\ \mathsf{else}\ c_i := c_i - 1; \mathsf{goto}\ q'')$ with $(1 \leq i \leq 2)$ is replaced by three instructions: $q : \mathsf{If}\ c_i = 0\ \mathsf{then}\ \mathsf{goto}\ q_2\ \mathsf{else}\ c_i := c_i - 1;\ \mathsf{goto}\ q_3,\ q_2 : c_3 := c_3 + 1;\ \mathsf{goto}\ q',\ q_3 : c_3 := c_3 + 1;\ \mathsf{goto}\ q''$ where q_2, q_3 are new states.

Finally, we replace the accepting state $accept$ of M by the initial state q_0' of M', i.e. we replace every instruction ($\mathsf{goto}\ accept$) by ($\mathsf{goto}\ q_0'$). M' is zero-initializing by definition. M' is bounded-strongly-cyclic, because c_3 is increased after every instruction and only set to zero at the initial state q_0'.

$\Rightarrow$ If M is a positive instance of CM then it has an accepting run from $(q_0, 0, 0)$. This run has finite length and is therefore space-bounded. Then M' has an infinite space-bounded cyclic run that starts at $(q_0', 0, 0, 0)$. Thus M' is a positive instance of BSC-ZI-CM$_b^\omega$.

$\Leftarrow$ If M' is a positive instance of BSC-ZI-CM$_b^\omega$ then there exists an infinite space-bounded run that starts at the configuration $(q_0', 0, 0, 0)$. By the construction of M' this run contains an accepting run of M from the configuration $(q_0, 0, 0)$. Thus M is a positive instance of CM. □

$\exists n$LCM$^\omega$

Instance: A strongly-cyclic, input-bounded 4-counter LCM M with initial state q_0.
Question: Does there exist an $n \in \mathbb{N}$ s.t. $runs^\omega((q_0, 0, 0, 0, n)) \neq \emptyset$?

Theorem 9. *$\exists n$LCM$^\omega$ is undecidable for every lossiness relation.*

Proof. We reduce BSC-ZI-CM$_b^\omega$ to $\exists n$LCM$^\omega$ with any lossiness relation $\xrightarrow{l}$. For any bounded-strongly-cyclic, zero-initializing 3-counter machine M we construct a strongly-cyclic, input-bounded lossy 4-counter machine M' with initial state q_0' and lossiness

relation $\xrightarrow{l}$ as follows: The 4-th counter c_4 holds the 'capacity'. In every operation it is changed in a way s.t. the sum of all counters never increases. (More exactly, the sum of all counters can increase by 1, but only if it was decreased by 1 in the previous step.) Every instruction of M of the form $(q:\ c_i := c_i + 1;\ \mathsf{goto}\ q')$ with $(1 \leq i \leq 3)$ is replaced by two instructions: $q:\ \mathsf{If}\ c_4 = 0\ \mathsf{then}\ \mathsf{goto}\ fail\ \mathsf{else}\ c_4 := c_4 - 1;\ \mathsf{goto}\ q_2,\ q_2:\ c_i := c_i + 1;\ \mathsf{goto}\ q'$, where $fail$ is a special final state and q_2 is a new state. Every instruction of the form $(q:\ \mathsf{If}\ c_i = 0\ \mathsf{then}\ \mathsf{goto}\ q'\ \mathsf{else}\ c_i := c_i - 1; \mathsf{goto}\ q'')$ with $(1 \leq i \leq 3)$ is replaced by two instructions: $q:\ \mathsf{If}\ c_i = 0\ \mathsf{then}\ \mathsf{goto}\ q'\ \mathsf{else}\ c_i := c_i - 1;\ \mathsf{goto}\ q_2,\ q_2 : c_4 := c_4 + 1;\ \mathsf{goto}\ q''$, where q_2 is a new state.

M' is bounded-strongly-cyclic, because M is bounded-strongly-cyclic. M' is input-bounded, because every run from a configuration $(q, m_1, \ldots, m_4)$ is space-bounded by $m_1 + m_2 + m_3 + m_4$. Thus M' is also strongly-cyclic.

$\Rightarrow$ If M is a positive instance of BSC-ZI-CM$_b^\omega$ then there exists a $n \in \mathbb{N}$ and an infinite run of M that starts at $(q_0, 0, 0, 0)$, visits q_0 infinitely often and always satisfies $c_1 + c_2 + c_3 \leq n$. Since $id \subseteq \xrightarrow{l}$, there is also an infinite run of M' that starts at $(q_0, 0, 0, 0, n)$, visits q_0 infinitely often and always satisfies $c_1 + c_2 + c_3 + c_4 \leq n$. Thus M' is a positive instance of $\exists n$LCM$^\omega$.

$\Leftarrow$ If M' is a positive instance of $\exists n$LCM$^\omega$ then there exists an $n \in \mathbb{N}$ s.t. there is an infinite run that starts at the configuration $(q_0', 0, 0, 0, n)$. This run is space-bounded, because it always satisfies $c_1 + c_2 + c_3 + c_4 \leq n$. By the construction of M', the sum of all counters can only increase by 1 if it was decreased by 1 in the previous step. By the definition of lossiness (see Def. 2) we get the following: If lossiness occurs (when the contents of the counters spontaneously change) then this strictly and permanently decreases the sum of all counters. It follows that lossiness can only occur at most n times in this infinite run and the sum of all counters is bounded by n. Thus there is an infinite suffix of this run of M' where lossiness does not occur. Thus there exist $q' \in Q$, $m_1', \ldots, m_4' \in \mathbb{N}$ s.t. an infinite suffix of this run of M' without lossiness starts at $(q', m_1', \ldots, m_4')$. It follows that there is an infinite space-bounded run of M that starts at $(q', m_1', \ldots, m_3')$. Since M is bounded-strongly-cyclic, this run must eventually visit q_0. Thus there exist $m_1'', \ldots, m_3'' \in \mathbb{N}$ s.t. an infinite space-bounded run of M starts at $(q_0, m_1'', \ldots, m_3'')$. Since M is zero-initializing, there is an infinite space-bounded run of M that starts at $(q_0, 0, 0, 0)$. Thus M is a positive instance of BSC-ZI-CM$_b^\omega$. □

Note that this undecidability result even holds under the additional condition that the LCMs are strongly-cyclic and input-bounded. It follows immediately that model checking LCM with the temporal logics CTL (computation-tree logic [7, 10]) and LTL (linear-time temporal logic [22]) is undecidable, since the question of $\exists n$LCM$^\omega$ can be expressed in these logics. There are two variants of the structural termination problem:

STRUCTTERM-LCM, VARIANT 1

Instance: A strongly-cyclic, input-bounded 4-counter LCM M with initial state q_0.
Question: Does M terminate for all inputs from q_0 ?
Formally: $\forall n_1, \ldots, n_4 \in \mathbb{N}.\ runs^\omega((q_0, n_1, n_2, n_3, n_4)) = \emptyset$?

STRUCTTERM-LCM, VARIANT 2

Instance: A strongly-cyclic, input-bounded 4-counter LCM M with initial state q_0.

Question: Does M terminate for all inputs from every control state q ?
Formally: $\forall n_1, \ldots, n_4 \in \mathbb{N}.\, \forall q \in Q.\, runs^\omega((q, n_1, n_2, n_3, n_4)) = \emptyset$?

Theorem 10. *Structural termination is undecidable for lossy counter machines. Both variants of STRUCTTERM-LCM are undecidable for every lossiness relation.*
Proof. The proof of Theorem 9 carries over, because the LCM is strongly-cyclic and the 3-CM in BSC-ZI-CM$_b^\omega$ is zero-initializing. □

SPACE-BOUNDEDNESS FOR LCM

Instance: A strongly-cyclic 4-counter LCM M with initial configuration $(q_0, 0, 0, 0, 0)$.
Question: Is M space-bounded ?

Theorem 11. *Space-boundedness for LCM is undecidable for all lossiness relations.*

Theorem 12. *Structural space-boundedness for LCM is undecidable for every lossiness relation.*
Proof. The proof is similar to Theorem 10. An extra counter is used to count the length of the run. It is unbounded iff the run is infinite. All other counters are bounded. □

5 Applications

5.1 Lossy Fifo-Channel Systems

Fifo-channel systems are systems of finitely many finite-state processes that communicate with each other by sending messages via unbounded fifo-channels (queues, buffers). In lossy fifo-channel systems these channels are lossy, i.e. they can spontaneously lose (arbitrarily many) messages. This can be used to model communication via unreliable channels. While normal fifo-channel systems are Turing-powerful, some safety-properties are decidable for lossy fifo-channel systems [2, 5, 1]. However, liveness properties are undecidable even for lossy fifo-channel systems. In [3] Abdulla and Jonsson showed the undecidability of the *recurrent-state problem* for lossy fifo-channel systems. This problem is if certain states of the system can be visited infinitely often. The undecidable core of the problem is essentially if there exists an initial configuration of a lossy fifo-channel system s.t. it has an infinite run. The undecidability proof in [3] was done by a long and complex reduction from a variant of Post's correspondence problem (2-permutation PCP [23], which is (wrongly) called cyclic PCP in [3]).
Lossy counter machines can be used to give a much simpler proof of this result. The lossiness of lossy fifo-channel systems is classic lossiness, i.e. the contents of a fifo-channel can change to any substring at any time. A lossy fifo-channel system can simulate a classic LCM (with some additional deadlocks) in the following way: Every lossy fifo-channel contains a string in X^* (for some symbol X) and is used as a classic lossy counter. The only problem is the test for zero. We test the emptiness of a channel by adding a special symbol Y and removing it in the very next step. If it can be done then the channel is empty (or has become empty by lossiness). If this cannot be done, then the channel was not empty or the symbol Y was lost. In this case we get a deadlock. These additional deadlocks do not affect the existence of infinite runs, and thus the results of Section 4 carry over. Thus the problem $\exists n\text{LCM}^\omega$ (for the classic lossiness relation) can be reduced to the problem above for lossy fifo-channel systems and the undecidability follows immediately from Theorem 9.

5.2 Model Checking Lossy Basic Parallel Processes

Petri nets [21] (also described as 'vector addition systems' in a different framework) are a widely known formalism used to model concurrent systems. They can also be seen as counter machines without the ability to test for zero, and are not Turing-powerful, since the reachability problem is decidable for them [17]. Basic Parallel Processes correspond to communication-free nets, the (very weak) subclass of labeled Petri nets where every transition has exactly one place in its preset. They have been studied intensively in the framework of model checking and semantic equivalences (e.g. [12, 18, 6, 15, 20]).

An instance of the *model checking problem* is given by a system S (e.g. a counter machine, Petri net, pushdown automaton,...) and a temporal logic formula φ. The question is if the system S has the properties described by φ, denoted $S \models \varphi$.

The branching-time temporal logics EF, EG and EG$_\omega$ are defined as extensions of Hennessy-Milner Logic [13, 14, 10] by the operators EF, EG and EG_ω, respectively. $s \models EF\varphi$ iff there exists an s' s.t. $s \xrightarrow{*} s'$ and $s' \models \varphi$. $s_0 \models EG_\omega\varphi$ iff there exists an infinite run $s_0 \to s_1 \to s_2 \to \ldots$ s.t. $\forall i.\, s_i \models \varphi$. EG is similar, except that it also includes finite runs that end in a deadlock. Alternatively, EF and EG can be seen as fragments of computation-tree logic (CTL [7, 10]), since $EF\varphi = true\,\mathcal{U}\,\varphi$ and $EG\varphi = \varphi\, w\mathcal{U}\, false$.

Model checking Petri nets with the logic EF is undecidable, but model checking Basic Parallel Processes with EF is $PSPACE$-complete [18]. Model checking Basic Parallel Processes with EG is undecidable [12]. It is different for lossy systems: By induction on the nesting-depth of the operators EF, EG and EG_ω, and constructions similar to the ones in Lemma 4 and Lemma 6, it can be shown that model checking classic LCM with the logics EF, EG and EG$_\omega$ is decidable. Thus it is also decidable for classical lossy Petri nets and classical lossy Basic Parallel Processes (see [4]).

However, model checking lossy Basic Parallel Processes with nested EF and EG/EG_ω operators is still undecidable for every subclassic lossiness relation. This is quite surprising, since lossy Basic Parallel Processes are an extremely weak model of infinite-state concurrent systems and the temporal logic used is very weak as well.

Theorem 13. *Model checking lossy Basic Parallel Processes (with any subclassic lossiness relation) with formulae of the form $EFEG_\omega\Phi$, where Φ is a Hennessy-Milner Logic formula, is undecidable.*

Proof. Esparza and Kiehn showed in [12] that for every counter machine M (with all counters initially 0) a Basic Parallel Processes P and a Hennessy-Milner Logic formula φ can be constructed s.t. M does not halt iff $P \models EG_\omega\varphi$. The construction carries over to subclassic LCM and subclassic lossy Basic Parallel Processes. The control-states of the counter machine are modeled by special places of the Basic Parallel Processes. In every infinite run that satisfies φ exactly one of these places is marked at any time.

We reduce $\exists n$LCM$^\omega$ to the model checking problem. Let M be a subclassic LCM. Let P be the corresponding Basic Parallel Processes as in [12] and let φ be the corresponding Hennessy-Milner Logic formula as in [12]. We use the same subclassic lossiness relation on M and on P. P stores the contents of the 4-th counter in a place Y. Thus $P\|Y^n$ corresponds to the configuration of M with n in the 4-th counter (and 0 in the others). We define a new initial state X and transitions $X \xrightarrow{a} X\|Y$ and $X \xrightarrow{b} P$, where

a and b do not occur in P. Let $\Phi := \varphi \wedge \neg\langle b\rangle true$. Then M is a positive instance of $\exists n\text{LCM}^\omega$ iff $X \models EFEG_\omega\Phi$. The result follows from Theorem 9. □

For Petri nets and Basic Parallel Processes, the meaning of Hennessy-Milner Logic formulae can be expressed by boolean combinations of constraints of the form $p \geq k$ (at least k tokens on place p). Thus the results also hold if boolean combinations of such constraints are used instead of Hennessy-Milner Logic formulae. Another consequence of Theorem 13 is that model checking lossy Petri nets with CTL is undecidable.

5.3 Reset/Transfer Petri Nets

Reset Petri nets are an extension of Petri nets by the addition of reset-arcs. A reset-arc between a transition and a place has the effect that, when the transition fires, all tokens are removed from this place, i.e. it is reset to zero. Transfer nets and transfer arcs are defined similarly, except that all tokens on this place are moved to some different place. It was shown in [8] that termination is decidable for 'Reset Post G-nets', a more general extension of Petri nets that subsumes reset nets and transfer nets. (For normal Petri nets termination is $EXPSPACE$-complete [24]). While boundedness is trivially decidable for transfer nets, the same question for reset nets was open for some time (and even a wrong decidability proof was published). Finally, it was shown in [8, 9] that boundedness (and structural boundedness) is undecidable for reset Petri nets. The proof in [8] was done by a complex reduction from Hilbert's 10th problem (a simpler proof was later given in [9]).
Here we generalize these results by using lossy counter machines. This also gives a unified framework and considerably simplifies the proofs.

Lemma 14. *Reset Petri nets can simulate lossy counter machines with reset-lossiness.*

Theorem 15. *Structural termination, boundedness and structural boundedness are undecidable for lossy reset Petri nets with every subclassic lossiness relation.*

Proof. It follows from Lemma 14 that a lossy reset Petri net with subclassic lossiness relation $\xrightarrow{l}$ can simulate a lossy counter machine with lossiness relation $\xrightarrow{l} \cup \xrightarrow{rl}$. The results follow from Theorem 10, Theorem 11 and Theorem 12. □

The undecidability result on structural termination carries over to transfer nets (instead of a reset the tokens are moved to a special 'dead' place), but the others don't. Note that for normal Petri nets structural termination and structural boundedness can be decided in polynomial time (just check if there is a positive linear combination of effects of transitions). Theorem 13 and Theorem 15 also hold for arbitrary lossiness relations, but this requires an additional argument. The main point is that Petri nets (unlike counter machines) can increase a place/counter and decrease another in the same step.

5.4 Parameterized Problems

We consider verification problems for systems whose definition includes a parameter $n \in \mathbb{N}$. Intuitively, n can be seen as the size of the system. Examples are

- Systems of n indistinguishable communicating finite-state processes.
- Systems of communicating pushdown automata with n-bounded stack.
- Systems of (a fixed number of) processes who communicate through (lossy) buffers or queues of size n.

Let $P(n)$ be such a system with parameter n. For every fixed n, $P(n)$ is a system with finitely many states and thus (almost) every verification problem is decidable for it. So the problem $P(n) \models \Phi$ is decidable for any temporal logic formula Φ from any reasonable temporal logic, e.g. modal μ-calculus [16] or monadic second-order theory. The *parameterized verification problem* is if a property holds independently of the parameter n, i.e. for any size. Formally, the question is if for given P and Φ we have $\forall n \in \mathbb{N}.\, P(n) \models \Phi$ (or $\neg\exists n \in \mathbb{N}.\, P(n) \models \neg\Phi$). Many of these parameterized problems are undecidable by the following meta-theorem.

Theorem 16. *A parameterized verification problem is undecidable if it satisfies the following conditions:*

1. *It can encode an n-space-bounded lossy counter machine (for some lossiness relation) in such a way that $P(n)$ corresponds to the initial configuration with n in one counter and 0 in the others.*
2. *It can check for the existence of an infinite run.*

Proof. By a reduction of $\exists n$LCM$^\omega$ and Theorem 9. The important point is that in the problem $\exists n$LCM$^\omega$ one can require that the LCM is input-bounded. □

The technique of Theorem 16 is used in [11] to show the undecidability of the model checking problem for linear-time temporal logic (LTL) and broadcast communication protocols. These are systems of n indistinguishable communicating finite-state processes where a 'broadcast' by one process can affect all other $n-1$ processes. Such a broadcast can be used to set a simulated counter to zero. However, there is no test for zero. One reduces $\exists n$LCM$^\omega$ with lossiness relation $\xrightarrow{rl}$ to the model checking problem. In the same way, similar results can be proved for parameterized problems about systems with bounded buffers, stacks, etc.

6 Conclusion

Lossy counter machines can be used as a general tool to show the undecidability of many problems. It provides a unified way of reasoning about many quite different classes of systems. For example the recurrent-state problem for lossy fifo-channel systems, the boundedness problem for reset Petri nets and the fairness problem for broadcast communication protocols were previously thought to be completely unrelated. Yet lossy counter machines show that the principles behind their undecidability are the same. Moreover, the undecidability proofs for lossy counter machines are very short and much simpler than previous proofs of weaker results [3, 8].

Lossy counter machines have also been used in this paper to show that even for very weak temporal logics and extremely weak models of infinite-state concurrent systems, the model checking problem is undecidable (see Subsection 5.2). We expect that many more problems can be shown to be undecidable with the help of lossy counter machines, especially in the area of parameterized problems (see Subsection 5.4).

Acknowledgments: Thanks to Javier Esparza and Petr Jančar for fruitful discussions.

References

[1] P. Abdulla, A. Bouajjani, and B. Jonsson. On-the-fly Analysis of Systems with Unbounded, Lossy Fifo Channels. In *10th Intern. Conf. on Computer Aided Verification (CAV'98)*. LNCS 1427, 1998.

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

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

[4] A. Bouajjani and R. Mayr. Model checking lossy vector addition systems. In *Proc. of STACS'99*, volume 1563 of *LNCS*. Springer Verlag, 1999.

[5] G. Cécé, A. Finkel, and S.P. Iyer. Unreliable Channels Are Easier to Verify Than Perfect Channels. *Information and Computation*, 124(1):20–31, 1996.

[6] S. Christensen, Y. Hirshfeld, and F. Moller. Bisimulation equivalence is decidable for Basic Parallel Processes. In E. Best, editor, *Proceedings of CONCUR 93*, volume 715 of *LNCS*. Springer Verlag, 1993.

[7] E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. volume 131 of *LNCS*, pages 52–71, 1981.

[8] 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.

[9] C. Dufourd, P. Jančar, and Ph. Schnoebelen. Boundedness of Reset P/T Nets. In *Proc. of ICALP'99*, volume 1644 of *LNCS*. Springer Verlag, 1999.

[10] E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science : Volume B, FORMAL MODELS AND SEMANTICS*. Elsevier, 1994.

[11] J. Esparza, A. Finkel, and R. Mayr. On the verification of broadcast protocols. In *Proc. of LICS'99*. IEEE, 1999.

[12] 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*. Springer Verlag, 1995.

[13] M. Hennessy and R. Milner. On observing nondeterminism and concurrency. volume 85 of *LNCS*, pages 295–309, 1980.

[14] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of Association of Computer Machinery*, 32:137–162, 1985.

[15] Y. Hirshfeld, M. Jerrum, and F. Moller. A polynomial-time algorithm for deciding bisimulation equivalence of normed Basic Parallel Processes. *Journal of Mathematical Structures in Computer Science*, 6:251–259, 1996.

[16] D. Kozen. Results on the propositional μ-calculus. *TCS*, 27:333–354, 1983.

[17] E. Mayr. An algorithm for the general Petri net reachability problem. *SIAM Journal of Computing*, 13:441–460, 1984.

[18] R. Mayr. Weak bisimulation and model checking for Basic Parallel Processes. In *Foundations of Software Technology and Theoretical Computer Science (FST&TCS'96)*, volume 1180 of *LNCS*. Springer Verlag, 1996.

[19] M.L. Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, 1967.

[20] F. Moller. Infinite results. In Ugo Montanari and Vladimiro Sassone, editors, *Proceedings of CONCUR'96*, volume 1119 of *LNCS*. Springer Verlag, 1996.

[21] J.L. Peterson. *Petri net theory and the modeling of systems*. Prentice-Hall, 1981.

[22] A. Pnueli. The temporal logic of programs. In *FOCS'77*. IEEE, 1977.

[23] K. Ruohonen. On some variants of Post's correspondence problem. *Acta Informatica*, 19:357–367, 1983.

[24] H. Yen. A unified approach for deciding the existence of certain Petri net paths. *Information and Computation*, 96(1):119–137, 1992.