Undecidable Problems in Unreliable Computations

Richard Mayr [a]

[a] *LIAFA - Université Denis Diderot - Case 7014 - 2, place Jussieu, F-75251 Paris Cedex 05. France. E-mail:* `mayr@liafa.jussieu.fr`

Abstract

Lossy counter machines are defined as Minsky 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.

Key words: Counter machines, lossy counter machines, decidability

1 Introduction

Lossy counter machines (LCM) are defined just like Minsky counter machines [25], 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).

Historically, the notion of 'lossiness' was first defined to model communication through unreliable channels. The main example are lossy fifo-channel systems, which are systems of finite-state processes that communicate through lossy fifo-channels (buffers) of unbounded length. These lossy fifo-channels are unreliable, because they can spontaneously lose messages. Since normal (non-lossy) fifo-channel systems are Turing-powerful, automatic analysis of them is restricted to special cases [4]. Lossy fifo-channel systems are not Turing-powerful, since reachability and some safety-properties are decidable for them [2,6,1]. However, some liveness-properties like the so-called 'recurrent-state problem' are undecidable even for lossy fifo-channel systems [3]. The result of this paper, the undecidability of structural termination for lossy counter machines, is much more general and subsumes this result (see Section 5).

The rest of the paper is structured as follows. In Section 2 we define lossiness relations and lossy counter machines. In Section 3 we show some decidable properties of lossy counter machines, and in Section 4 we prove the main undecidability result. Section 5 gives several examples how this result can be applied. In the last two sections we discuss possible generalizations and draw some conclusions.

2 Definitions

Definition 1 *An n-counter machine [25] 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 $c_1, \dots, c_n$ and a finite set of instructions of the form* $(q : \ c_i := c_i + 1; \mathsf{goto}\ q')$ *or* $(q : \ \mathsf{If}\ c_i = 0\ \mathsf{then\ goto}\ q'\ \mathsf{else}\ c_i := c_i - 1; \mathsf{goto}\ q'')$ *where* $i \in \{1, \dots, n\}$ *and* $q, q', q'' \in Q$.

A configuration *of M is described by a tuple* $(q, m_1, \dots, m_n)$ *where* $q \in Q$ *and* $m_i \in \mathbb{N}$ *is the content of the counter* c_i $(1 \le i \le n)$*. 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 as follows:*

(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 $(q : \; \mathsf{If}\ c_i = 0\ \mathsf{then\ goto}\ q'\ \mathsf{else}\ c_i := c_i - 1; \mathsf{goto}\ q'')$ *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 $(q : \; \mathsf{If}\ c_i = 0\ \mathsf{then\ goto}\ q'\ \mathsf{else}\ c_i := c_i - 1; \mathsf{goto}\ q'')$ *and* $m_i > 0$.

A counter machine is deterministic *iff for every control-state* $q \in Q$ *there is at most one instruction* $(q : \; \ldots\ldots)$ *at this control-state. A* run *of a counter machine is a (possibly infinite) sequence of configurations* $s_0, s_1, \ldots$ *with* $s_0 \to s_1 \to s_2 \to s_3 \to \ldots$.

Now we define *lossiness relations*, which 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 which is defined as follows.*

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

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 lossy counter machine 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 for every $x \in \mathbb{N}$.

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 for all i either
(1) $m_i = m_i$, or
(2) $m_i' = 0$ and there is an instruction
$(q :$ If $c_i = 0$ then goto q' else $c_i := c_i - 1;$ goto $q'')$.
Note that $\xrightarrow{rl}$ is a subclassic lossiness relation.

The definition of these lossiness relations carries over to other models like Petri nets [27], where places are considered instead of counters and the control-states q are ignored.

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. For a run r and a configuration s we write $s \in r$ to indicate that s is one of the configurations that occur in r. 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 size of 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, no interesting properties are 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 [5,6,2].

Lemma 4 (Dickson's Lemma [10])
Given an infinite sequence of vectors $\vec{x}_1, \vec{x}_2, \vec{x}_3, \ldots$ in $\mathbb{N}^k$ there are $i < j$ s.t. $\vec{x}_i \leq \vec{x}_j$ ($\leq$ taken componentwise).

Lemma 5 *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.*

PROOF. *Since M is a classic LCM, the set $pre^*(s)$ is upward closed and can thus be characterized by its finitely many minimal elements. These minimal elements can be effectively constructed because of Dickson's Lemma [10] (see also [5]).* □

Theorem 6 *Reachability is decidable for classic LCM.*

PROOF. *Given two configurations s and s', the question if $s \Longrightarrow^* s'$ is equivalent to $s \in pre^*(s')$. This is decidable by Lemma 5.* □

Lemma 7 *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 $runs^\omega(s_0) \neq \emptyset$.*

PROOF. *We analyze all runs by breadth-first search. If there is no infinite run then all runs will eventually terminate. Thus, the algorithm will terminate and give the correct answer 'no'. If there is an infinite run, then, by Dickson's Lemma [10], we will eventually reach a configuration s s.t. there is a previous configuration s' in the same run with $s \geq s'$. In this case there is an infinite cyclic run from s' to s', because M is classical lossy. Thus, in this case the algorithm also terminates and gives the correct answer 'yes'.* □

In [2] Abdulla and Jonsson proved a more general result that subsumes Lemma 7. They showed that the existence of an infinite run from a given initial configuration is decidable even for lossy FIFO-channel systems.
Theorem 8 *Termination is decidable for classic LCM.*

PROOF. *A classic LCM M with initial configuration s_0 is terminating iff $runs^\omega(s_0) = \emptyset$. This is decidable by Lemma 7.* $\square$

It has been shown in [5] that even model checking classic LCM with the temporal logics EF and EG (natural fragments of computation tree-logic (CTL) [9,14]) is decidable.
Another interesting observation is that Petri nets and classical lossy counter machines are incomparable. For Petri nets, model checking with the temporal logic EF is undecidable, but model checking with LTL is decidable [15]. For classical lossy counter machines it is just vice-versa. For classical lossy counter machines model checking with EF is decidable [5], but model checking with LTL is undecidable [5] (see also Theorem 11).

4 The Undecidability Result

We show that structural termination (i.e., termination for every input) is undecidable for LCM for every lossiness relation. We start with the problem CM, which was shown to be undecidable by Minsky [25].
<u>CM</u>
Instance: A deterministic 2-counter machine M with initial state q_0.
Question: Does M accept $(q_0, 0, 0)$?
We reduce the problem CM to the following problem.
<u>BSC-ZI-CM$_b^\omega$</u>
Instance: A deterministic 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 9 *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 $(1 \le i \le 2)$ 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*

$$(q: \ \mathsf{If}\ c_i = 0\ \mathsf{then\ goto}\ q'\ \mathsf{else}\ c_i := c_i - 1; \mathsf{goto}\ q'')$$

with $(1 \leq i \leq 2)$ *is replaced by*

$$q: \ \mathsf{If}\ c_i = 0\ \mathsf{then\ goto}\ q_2\ \mathsf{else}\ c_i := c_i - 1;\ \mathsf{goto}\ q_3$$

$$q_2 : c_3 := c_3 + 1;\ \mathsf{goto}\ q'$$

$$q_3 : c_3 := c_3 + 1;\ \mathsf{goto}\ q''$$

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* (goto *accept*) *by* (goto q_0')*.* M' *is deterministic, because* M *is deterministic.* 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 exactly one accepting run from* $(q_0, 0, 0)$ *(we assume without restriction that* M *has exactly one accepting state and that this state has no outgoing transitions). 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.* □

Now we consider the central problem for lossy counter machines.

$\exists n$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 10 *$\exists nLCM^\omega$ is undecidable for every lossiness relation.*

PROOF. *We reduce BSC-ZI-CM*$_b^\omega$ *to* $\exists nLCM^\omega$ *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*

$$q: \ \mathsf{If}\ c_4 = 0\ \mathsf{then\ goto}\ \mathit{fail}\ \mathsf{else}\ c_4 := c_4 - 1;\ \mathsf{goto}\ q_2$$

$$q_2 : c_i := c_i + 1;\ \mathsf{goto}\ q'$$

where 'fail' is a special final state and q_2 is a new state. Every instruction of the form ($q:$ If $c_i = 0$ then goto q' else $c_i := c_i - 1$; goto q'') *with* $(1 \le i \le 3)$ *is replaced by*

$$q: \text{ If } c_i = 0 \text{ then goto } q' \text{ else } c_i := c_i - 1; \text{ goto } q_2$$

$$q_2 : c_4 := c_4 + 1; \text{ goto } q''$$

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 $(q, m_1, \dots, m_4)$ 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 $c_1 + c_2 + c_3 \le n$. 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 $c_1 + c_2 + c_3 + c_4 \le n$. Thus M' is a positive instance of $\exists nLCM^\omega$.*
- $\Leftarrow$ *If M' is a positive instance of $\exists nLCM^\omega$ 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 $c_1 + c_2 + c_3 + c_4 \le n$. 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 $q' \in Q$, $m_1', \dots, m_4' \in \mathbb{N}$ s.t. an infinite suffix of this run of M' without lossiness starts at $(q', m_1', \dots, m_4')$. It follows that there is an infinite space-bounded run of M that starts at $(q', m_1', \dots, m_3')$. Since M is bounded-strongly-cyclic, this run must eventually visit q_0. Thus there exist $m_1'', \dots, m_3'' \in \mathbb{N}$ s.t. an infinite space-bounded run of M starts at $(q_0, m_1'', \dots, m_3'')$. 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.
This result can be used to show that model checking LCM with the temporal logics CTL (computation-tree logic [9,14]) and LTL (linear-time temporal logic [28]) is undecidable, since the question of $\exists n$LCM$^\omega$ can be encoded in these logics.
Theorem 11 *Model checking LCM with the temporal logics CTL and LTL is undecidable for every lossiness relation.*

PROOF. *Let M be a lossy 4-counter LCM with lossiness relation $\xrightarrow{l}$ and initial state q_0. We construct the LCM M' as follows: Let q_0' be the new initial state of M'. M' has the same instructions as M plus the following ones:*

$$q_0' : c_4 := c_4 + 1; \ \mathsf{goto}\ q_0'$$

$$q_0' : c_4 := c_4 + 1; \ \mathsf{goto}\ q_0$$

We label these two new instructions with action 'a' and all others with action 'b'. Then we have that M is a positive instance of $\exists nLCM^\omega$ iff M' satisfies the LTL formula $(q_0', 0, 0, 0, 0) \models \langle a \rangle true\, U\, (\langle b \rangle true\, wU\, false)$ or the CTL formula $(q_0', 0, 0, 0, 0) \models E[\langle a \rangle true\, U\, E[(\langle b \rangle true\, wU\, false)]]$ □

The following two variants of the structural termination problem are equivalent. An LCM is a positive instance of variant 1 iff it is a positive instance of variant 2, because of the imposed condition that the LCM is strongly-cyclic. The only reason why we define both variants is to point out this fact.

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 12 *Structural termination is undecidable for lossy counter machines. Both variants of STRUCTTERM-LCM are undecidable for every lossiness relation.*

PROOF. *The proof of Theorem 10 carries over, because the LCM is strongly-cyclic and the 3-CM in BSC-ZI-CM_b^ω is zero-initializing.* □

SPACE-BOUNDEDNESS FOR LCM

Instance: A strongly-cyclic 4-counter LCM M with the initial configuration $(q_0, 0, 0, 0, 0)$.

Question: Is M space-bounded ?

Theorem 13 *Space-boundedness for LCM is undecidable for all lossiness relations.*

PROOF. *We reduce BSC-ZI-CM$_b^\omega$ to the space-boundedness problem for LCM. Let M be the 3-CM from BSC-ZI-CM$_b^\omega$. We take the LCM M' from the proof of Theorem 10 and modify it as follows (obtaining a new LCM M''): At the final state 'fail' we do not stop. Instead we add c_1, c_2 and c_3 to c_4, set c_1, c_2 and c_3 to 0 and increase c_4 by 1 and go to the initial state q_0'' of M''. Formally, this is defined by*

$$\begin{aligned}
&\mathit{fail}: \text{If } c_1 = 0 \text{ then goto } f_2 \text{ else } c_1 := c_1 - 1;\ \text{goto } d_1\\
&d_1: \quad c_4 := c_4 + 1;\ \text{goto } \mathit{fail}\\
&f_2: \quad \text{If } c_2 = 0 \text{ then goto } f_3 \text{ else } c_2 := c_2 - 1;\ \text{goto } d_2\\
&d_2: \quad c_4 := c_4 + 1;\ \text{goto } f_2\\
&f_3: \quad \text{If } c_3 = 0 \text{ then goto } f_4 \text{ else } c_3 := c_3 - 1;\ \text{goto } d_3\\
&d_3: \quad c_4 := c_4 + 1;\ \text{goto } f_3\\
&f_4: \quad c_4 := c_4 + 1;\ \text{goto } q_0''
\end{aligned}$$

The initial configuration of M'' is $(q_0'', 0, 0, 0, 0)$. Now we show that M is a positive instance of BSC-ZI-CM$_b^\omega$ iff M'' is bounded.

$\Rightarrow$ *If M is a positive instance of BSC-ZI-CM$_b^\omega$ then it uses only a finite amount k of space, i.e., we have always $c_1 + c_2 + c_3 < k$ in both M and M''. If the value in c_4 becomes larger than k then there are two cases.*

(1) If M'' does not lose then it will enter an infinite space-bounded cyclic computation which never visits the state 'fail' again. Thus these runs of M'' are bounded.

(2) In order to visit the state 'fail' again M'' must lose at least once. This is at most compensated in the state 'fail' (the sum of the counters is increased by 1), but not more than that. Thus these runs of M'' are bounded as well.

Thus all computations of M'' from $(q_0'', 0, 0, 0, 0)$ are space-bounded.

$\Leftarrow$ *If M is a negative instance of BSC-ZI-CM$_b^\omega$ then the computation of M'' from $(q_0'', 0, 0, 0, 0)$ without losses will visit the state 'fail' infinitely often and the sum of all counters will become arbitrarily high. (The run without losses is one possible run, since by Def. 2 $id \subseteq \xrightarrow{l}$.) Thus M'' is not space-bounded.* $\square$

Remark 14 *It follows directly from Theorem 13 that the set of reachable configurations of a LCM cannot be effectively constructed. (If one could construct this set then one could decide boundedness). In particular, this non-constructibility result also holds for classical LCM. The set of reachable configurations of a classical LCM is always semilinear, since it is downward closed. Thus, the set of reachable configurations of a classical LCM is semilinear, but not effectively semilinear.*

It has already been stated in [6] that the regular expression that describes the set of reachable configurations of a lossy fifo-channel system cannot be effectively constructed, although it always exists. (The proof in [6] contains a slight error.) This result is subsumed by the more general Theorem 13 and Remark 14.

STRUCTURAL SPACE-BOUNDEDNESS FOR LCM

Instance: A strongly-cyclic 5-counter LCM. M.
Question: Is M space-bounded for every initial configuration $(q, n_1, n_2, n_3, n_4, n_5)$?

Theorem 15 *Structural space-boundedness for LCM is undecidable for every lossiness relation.*

PROOF. *The proof is similar to Theorem 12. An extra counter c_5 is used to count the length of the run. It is unbounded iff the run is infinite. All other counters are bounded.* □

5 Applications

Lossy counter machines can be used to prove the undecidability of many problems.

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 [2,6,1]. 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 reduction from a variant of Post's correspondence problem, namely 2-permutation PCP. The undecidability of 2-permutation PCP has been shown by Ruohonen in [29]. There is some confusion of the names of the problems in the literature. Abdulla and Jonsson [3] use Ruohonen's result on the undecidability of 2-permutation PCP and cite [29], but they refer to the problem '2-permutation PCP' as 'cyclic PCP'. However, the real cyclic PCP is a different problem, which is also defined and shown to be undecidable by Ruohonen in [29].

Lossy counter machines can be used to give a much simpler proof of the undecidability results for lossy FIFO-channel systems. 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 length of the string encodes the value in the counter. The only problem is the test for zero. We test the emptiness of a fifo-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 $\exists n\mathrm{LCM}^\omega$ (for the classic lossiness relation) can be reduced to the problem above for lossy fifo-channel systems and the undecidability follows immediately from Theorem 10.

5.2 Model Checking Lossy Basic Parallel Processes

Petri nets [27] (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 [22]. Basic Parallel Processes [7] 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., [17,23,24,8,20,26]).

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_ω are defined as extensions of Hennessy-Milner Logic [18,19,14] 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 $s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \ldots$ 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 [9,14]), since $EF\varphi = E[\mathit{true}\,\mathcal{U}\,\varphi]$ and $EG\varphi = E[\varphi\, w\mathcal{U}\, \mathit{false}]$.

Model checking Petri nets with the logic EF is undecidable [15], but model checking Basic Parallel Processes with EF is *PSPACE*-complete [23]. Model checking Basic Parallel Processes with EG is undecidable [17]. 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 5 and Lemma 7, it can be shown that model checking classic LCM with the logics EF, EG and

EG_ω is decidable. Thus it is also decidable for classical lossy Petri nets and classical lossy Basic Parallel Processes (see [5]).
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. (Note in particular that lossy Basic Parallel Processes are *normed*, i.e., from every reachable state there is a terminating computation.)
Theorem 16 *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 [17] 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 $\exists nLCM^\omega$ to the model checking problem. Let M be a subclassic LCM. Let P be the corresponding Basic Parallel Processes as in [17] and let φ be the corresponding Hennessy-Milner Logic formula as in [17]. 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 $\exists nLCM^\omega$ iff $X \models EFEG_\omega\Phi$. The result follows from Theorem 10. $\square$

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 16 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 [12] 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 [30]). 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 [12] that boundedness (and structural boundedness) is undecidable for reset Petri nets. The proof in [12] was done by a complex reduction from Hilbert's 10th problem (a simpler proof was later given in [11,13]).

Here we generalize these results by using lossy counter machines. This also gives a unified framework and considerably simplifies the proofs.

Lemma 17 *Reset Petri nets can simulate the infinite runs of lossy counter machines with reset-lossiness.*

PROOF. *For every n-counter LCM M (with the reset lossiness relation $\xrightarrow{rl}$) we construct a reset Petri net N in the following way: Let there be places $c_1, \ldots, c_n$ that hold the contents of the counters and a place q for every state $q \in Q$ of the finite control of M. Every marking of this net N where exactly one of the places q contains exactly one token corresponds to a configuration of the counter machine M and vice versa. For every instruction of M of the form $(q: \ c_i := c_i + 1;\ \mathsf{goto}\ q')$ with $(1 \leq i \leq n)$ there is a transition that takes one token from q, puts one token on c_i, puts one token on q' and resets all places except $q', c_1, \ldots, c_n$. The firing of this transition exactly simulates the computation step of M. For every instruction of M of the form $(q: \ \mathsf{If}\ c_i = 0\ \mathsf{then\ goto}\ q'\ \mathsf{else}\ c_i := c_i - 1; \mathsf{goto}\ q'')$ with $(1 \leq i \leq n)$ there are two transitions: The first transition takes a token from q, puts a token on q' and resets c_i and all places except $q', c_1, \ldots, c_n$. Instead of being tested for zero the place/counter c_i is reset to zero. If the place/counter c_i actually was zero before the transition fired, then this was a faithful simulation of the computation step of the counter machine. If the place/counter c_i was not zero before, then it was still a faithful simulation of a computation step of the* reset-lossy *counter machine, because c_i could suddenly have become zero (empty) by lossiness (see the Def. 2 of reset lossiness $\xrightarrow{rl}$). The second transition takes one token from q and one from c_i, puts one token on q'' and resets all places except $q'', c_1, \ldots, c_n$. This transition can only fire if c_i is not zero (empty) and faithfully simulates the computation step of the counter machine.*

The only problem with this simulation is that it is possible that in N all tokens on the places q are lost. This causes a deadlock in N. The same thing cannot happen in M, because the finite-control cannot be lost. Thus, N simulates M with some extra deadlocks. However, we still have that

- *For every infinite run of M there is an infinite run of N that faithfully simulates it.*
- *For every infinite run of N there is an infinite run of M that faithfully simulates it.*

Thus, the reset net N faithfully simulates all infinite runs of M. □

Theorem 18 *Structural termination, boundedness and structural boundedness are undecidable for lossy reset Petri nets with every subclassic lossiness relation.*

PROOF. *It follows from Lemma 17 that a lossy reset Petri net with subclassic lossiness relation $\xrightarrow{l}$ can simulate the infinite runs of a lossy counter machine with lossiness relation $\xrightarrow{l} \cup \xrightarrow{rl}$. The results follow from Theorem 12, Theorem 13 and Theorem 15.* $\square$

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. For example, boundedness is decidable for transfer nets [12]. 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 16 and Theorem 18 also hold for arbitrary lossiness relations instead of just subclassic ones, but this requires an additional argument. When a Petri net (weakly) simulates a lossy counter machine (e.g., like in Lemma 17) then special places are used to encode the finite-control. If the lossiness relation on the Petri net is not subclassic then the simulated control-state could change by lossiness. This is a problem for lossy counter machines, because (by using the 'capacity' in c_4) one wants to make sure that lossiness cannot occur infinitely often. But now it can happen again as follows:

$$q: \; c_1 := c_1 + 1; \; \textsf{goto } q'$$

By lossiness the control-state could change from q' back to q while the counter c_1 is decreased by 1. The result is an infinite loop at q where c_1 stays at the same value.

On can get around this problem by using the special features of Petri nets. Petri nets (unlike counter machines) can increase a place/counter and decrease another in the same step. So, instead of decreasing the capacity and increasing a counter in the next step (like in Theorem 10) we can do both in one step with one transition. This solves the problem, because now the sum of all places never increases, not even temporarily as in lossy counter machines. Then the proofs of Theorem 16 and Theorem 18 carry over to all lossiness relations.

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 [21] 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 19 *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 $\exists nLCM^\omega$ and Theorem 10. The important point here is that in the problem $\exists nLCM^\omega$ one can require that the LCM is input-bounded.* □

The technique of Theorem 19 is used in [16] to show the undecidability of the fairness problem for broadcast communication protocols. These are systems of n indistinguishable communicating finite-state processes. The rules for communication are as follows:

(1) Two processes can communicate directly by handshake.

(2) One process can broadcast a message, which is received (immediately) by all other $n-1$ processes.

Every message sent or received by a process can change its internal state, which in turn defines what actions it can perform and how it reacts to messages. The rules for communication are defined independently from the number n of processes in the system. If one considers processes with k internal states then any configuration of the broadcast protocol with n processes can be described by a tuple $(m_1, m_2, \ldots, m_k)$ where m_i is the number of processes in state i and $\sum_{j=1}^{k} m_j = n$. Every such m_i can be seen as the content of a counter which is bounded by n. A broadcast can cause all processes in a certain state to change to another state. This can be used to reset such a simulated space-bounded counter to zero. Note however, that no test for zero is possible. The problem if such a broadcast protocol terminates (i.e., for every number n of processes the system terminates) is undecidable, because it satisfies the conditions of Theorem 19 (the lossiness relation used here is reset-lossiness). Thus all fairness properties, like those expressible in the temporal logics CTL [9,14]) and LTL (linear-time temporal logic [28]), are undecidable as well.

In the same way, similar results can be proved for parameterized problems about systems with bounded buffers, stacks, etc.

6 Extensions

The proofs of the main undecidability results in Theorem 10 and Theorem 12 work only for LCM with at least 4 counters. The question arises, if fewer counters suffice, like the two counters used in normal counter machines. However, the methods used to reduce the number of counters in normal counter machines do not carry over to LCM. They use codings which are not robust under lossiness. Also these codings require a lot of computation and some types of LCM are not exactly Turing-powerful. The decidability of structural termination for LCM with 3 or less counters probably depends on the particular lossiness relation.

The computational power of lossy counter machines also depends very much on the particular lossiness relation. However, a few general observations can be made. The utmost one can expect from a LCM is the following:

- There is at least one computation that gives the correct result, since $id \subseteq \xrightarrow{l}$.
- There may be other computations that give results that are smaller than the correct result (by the definition of lossiness).

For some operations, e.g., addition and multiplication, this optimal behavior can be achieved. However, for other operations like subtraction it is impossible, since the obtained result may even be larger than the correct one. In fact, many versions of LCM cannot even compare two numbers. Thus, it should be stressed that we do not advocate LCM as a model of computation, but rather as a means of proving undecidability.

Another question is if the undecidability results can be extended to more general lossiness relations than $\xrightarrow{s}$ (see Def. 2). (Even $\xrightarrow{s}$ can hardly be called lossiness any more, since it allows some counters to increase while others decrease.) One idea is to introduce functions $f : \mathbb{N}^n \mapsto \mathbb{N}$ s.t. if $s \xrightarrow{l} s'$ then either $s = s'$ or $f(s') < f(s)$. (In the case of $\xrightarrow{s}$ the function f is the sum.) Again this depends very much on the lossiness relation relation $\xrightarrow{l}$. In the proof of Theorem 10 a balance must be kept in the 4th counter, to ensure that the LCM is input-bounded and lossiness can occur only finitely often in the infinite run. This balance must be updated (computed) on the lossy counter machine, which is not always Turing-powerful. In the simple case of the 'sum' function this is trivial, but for more general functions f it is a problem.

7 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 [3,12].

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 and to an anonymous referee for detailed comments.

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 P. Habermehl. Symbolic reachability analysis of FIFO-channel systems with nonregular sets of configurations. In *Proc. of ICALP'97*, volume 1256 of *LNCS*, 1997.

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

[6] 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.

[7] S. Christensen. *Decidability and Decomposition in Process Algebras*. PhD thesis, Edinburgh University, 1993.

[8] 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.

[9] 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.

[10] L.E. Dickson. Finiteness of the odd perfect and primitive abundant numbers with distinct factors. *American Journal of Mathematics*, 35:413–422, 1913.

[11] C. Dufourd. *Réseaux de Petri avec Reset/Transfert: Décidabilité et indécidabilité.* PhD thesis, ENS de Cachan, 1998.

[12] 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.

[13] 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.

[14] 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.

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

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

[17] 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.

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

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

[20] 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.

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

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

[23] 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.

[24] R. Mayr. On the complexity of bisimulation problems for Basic Parallel Processes. In *Proc. of ICALP 2000*, volume 1853 of *LNCS*. Springer Verlag, 2000.

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

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

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

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

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

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