Under consideration for publication in Formal Aspects of Computing

On Termination and Invariance for Faulty Channel Machines

Patricia Bouyer[⋆], Nicolas Markey[⋆], Joël Ouaknine[‡], Philippe Schnoebelen[⋆], and James Worrell[‡]
[⋆]LSV, ENS Cachan, CNRS, France
[‡]Department of Computer Science, Oxford University, UK

Abstract. A *channel machine* consists of a finite controller together with several fifo channels; the controller can read messages from the head of a channel and write messages to the tail of a channel. In this paper we focus on channel machines with *insertion errors*, i.e., machines in whose channels messages can spontaneously appear. We consider the *invariance* problem: does a given insertion channel machine have an infinite computation all of whose configurations satisfy a given predicate? We show that this problem is primitive-recursive if the predicate is closed under message losses. We also give a non-elementary lower bound for the invariance problem under this restriction. Finally, using the previous result, we show that the satisfiability problem for the safety fragment of Metric Temporal Logic is non-elementary.

Keywords: Channel Machines; Computational Complexity; Metric Temporal Logic; Primitive Recursive; Well-Structured Systems

1. Introduction

Many recent developments in the area of automated verification, both theoretical and practical, concern infinite-state systems. Although such systems are not, in general, amenable to fully algorithmic analysis, a number of important classes of models with decidable verification problems have been identified. Several of these classes, such as Petri nets, process algebras, rewrite systems, faulty channel machines, timed automata, and many more, are instances of *well-structured transition systems*, for which various problems are decidable—see [FS01] for a comprehensive survey.

Well-structured transition systems are predicated on the existence of 'compatible well-quasi orders', which guarantee, for example, that certain fixed-point computations terminate. Unfortunately, these properties are often non-constructive in nature, so that although convergence is guaranteed, the number of steps to convergence is not necessarily known. Thus, while decidability of a given type of system can be established by recourse to the general theory of well-structured systems, establishing complexity bounds typically requires a more specialised analysis of the model at hand. A classic example is the coverability problem for Petri

Correspondence and offprint requests to: James Worrell, Department of Computer Science, Oxford University, UK e-mail: jbw@cs.ox.ac.uk

nets, which can be shown to be decidable using generic techniques [FS01] but for which optimal complexity bounds require more specific techniques [Rac78].

In this paper, we are interested in a particular kind of well-structured transition system, known as faulty channel machines. A channel machine (also known as a queue automaton) consists of a finite-state controller equipped with several unbounded fifo channels (queues, buffers). Transitions of the machine can write messages (letters) to the tail of a channel and read messages from the head of a channel. Channel machines can be used, for example, to model distributed protocols that communicate asynchronously.

Channel machines are easily seen to be Turing powerful [BZ83], and all non-trivial verification problems concerning them are therefore undecidable. In [AJ93, Fin94, CFPI96, AJ96], Abdulla and Jonsson, and Finkel *et al.* independently introduced *lossy channel machines* as channel machines operating over an unreliable medium; more precisely, they made the assumption that messages held in channels could at any point vanish nondeterministically. Not only was this a compelling modelling assumption, enabling the representation of fault-tolerant protocols, for example, but it also endowed the underlying transition systems of lossy channel machines with a well-structure, thanks to Higman's lemma [Hig52]. As a result, several non-trivial problems, such as control-state reachability, are decidable for lossy channel machines.

Abdulla and Jonsson admitted in [AJ93] that they were unable to determine the complexity of the various problems they had shown to be decidable. Such questions remained open for almost a decade, despite considerable research interest in the subject. Finally, Schnoebelen showed in [Sch02] that virtually all non-trivial decidable problems concerning lossy channel machines have non-primitive recursive complexity. This result, in turn, settled the complexity of a host of other problems, usually via reduction from reachability for lossy channel machines. A more precise analysis has been given more recently [CS08, SS11], in which the complexity of reachability and termination for lossy channel systems is placed at level $\mathcal{F}_{\omega^\omega}$ in the fast-growing hierarchy of recursive functions.

Other models of unreliable media in the context of channel machines have also been studied in the literature. In [CFPI96], for example, the effects of various combinations of insertion, duplication, and lossiness errors are systematically examined. Although insertion errors are well-motivated (as former users of modems over telephone lines can attest!), they were surprisingly found in [CFPI96] to be theoretically uninteresting: in the presence of insertion errors channels become redundant since read- and write-transitions are continuously enabled (the former because of potential insertion errors, the latter by assumption, as channels are unbounded). Consequently, most verification problems trivially reduce to questions on finite automata.

Recently, however, slightly more powerful models of channel machines with insertion errors have appeared as key tools in the study of Metric Temporal Logic (MTL), an extension of linear temporal logic for reasoning about real-time systems [HMP92, Koy90]. In [OW05, OW06a], the authors showed that MTL formulas can capture the computations of insertion channel machines *equipped with primitive operations for testing channel emptiness*. Emptiness testing provides some measure of control over insertion errors, and this new class of faulty channel machines has non-primitive recursive reachability problem and undecidable recurrent control-state reachability problem. Consequently, MTL satisfiability and model checking were established to be non-primitive recursive over finite words [OW05], and undecidable over infinite words [OW06a].

Independently of Metric Temporal Logic, the notion of emptiness testing is very natural. Counter machines, for instance, are usually assumed to incorporate primitive zero-testing operations on counters, and likewise pushdown automata are able to detect empty stacks. Variants of Petri nets have also explored emptiness testing for places, usually resulting in a great leap in computational power.

Our main focus in this paper is on termination and, more generally, invariance for insertion channel machines. One problem we consider is the *termination* problem for insertion channel machines with emptiness tests (ICMETs): given such a machine, are all of its computations finite? We show that termination is non-elementary, yet primitive recursive. This result is quite surprising, as the closely related problems of reachability and recurrent reachability are respectively non-primitive recursive and undecidable. Moreover, the mere *decidability* of termination for insertion channel machines follows from the theory of well-structured transition systems, in a manner similar to that for lossy channel machines. In the latter case, however, termination is non-primitive recursive, as shown in [Sch02].

Emptiness tests are one of a number of possible tests that can be added to insertion channel machines. A more general notion is *absence tests*, which allow the machine to check that a certain symbol is not on the channel [BMO$^+$08]. The key feature of such tests that preserves decidability is their compatibility with the sub-word order on channel configurations. A framework that subsumes a wide variety of such compatible tests is to restrict attention to configurations that satisfy a given predicate L, where L *downward-closed*, i.e., closed with respect to the sub-word order on channel configurations. Thus we are led to consider the *invariance problem* for insertion channel machines: given an insertion channel machine and downward-closed predicate L, is there an infinite computation all of whose configurations satisfy L? The invariance problem for ICMs generalises the termination problem for ICMETs since emptiness tests can be characterised in terms of downward-closed predicates. Our main result is that the invariance problem is primitive-recursive.

On the practical side, one of the main motivations for studying termination of insertion channel machines arises from the safety fragment of Metric Temporal Logic. Safety MTL was shown to be decidable in [OW06b], although no non-trivial bounds on the complexity could be established at the time. In this paper we show that the satisfiability problem for Safety MTL is non-elementary by reduction from the termination problem for ICMETs. We note that in a similar vein, an EXPSPACE lower bound for the complexity of satisfiability of an extension of Linear Temporal Logic on data words was given in [Laz11] via a reduction from the termination problem for counter machines with incrementation errors.

This paper extends the conference paper [BMO$^+$08] by generalising from the termination problem for ICMET to the invariance problem for ICM and by including a proof that satisfiability for Safety MTL is non-elementary.

2. Decision Problems for Faulty Channel Machines

In this section, we briefly review some key decision problems for lossy and insertion channel machines (the latter equipped with emptiness testing). Apart from the results on termination and invariance for insertion channel machines, which are presented in the following sections, all results that appear here are either known or follow easily from known facts. Note that, for lossy channel machines, allowing tests on channel contents does not make verification harder [BBS06].

The *reachability* problem asks whether a given control state of a channel machine is reachable. This problem was shown to be non-primitive recursive for lossy channel machines in [Sch02] and in [CS08] it was shown not even to be multiply recursive. Decidability of this problem can be established using Higman's Lemma [FS01]. Based on an analysis of the length of "bad sequences" in Higman's lemma the problem was shown in [SS11] to lie in level $\mathcal{F}_{\omega^\omega}$ in the fast-growing hierarchy of recursive functions. Reachability for insertion channel machines is logspace equivalent to reachability for lossy channel machines via a simple dualisation construction [OW05], so all the above complexity results carry over.

The *termination* problem asks whether all computations of a channel machine are finite, starting from the initial control state and empty channel contents. For lossy channel systems termination and reachability have the same complexity [Sch02, CS08]. However we prove in Section 5 that termination of insertion channel machines is primitive recursive. In Section 4 we give a non-elementary lower bound for this problem.

The *invariance* problem asks whether a given channel machine has an infinite computation all of whose configurations satisfy a given predicate L. (We postpone details about effective representations of infinite sets of configurations until later.) The invariance problem for lossy channel systems is decidable [AČJT00] provided that the predicate L is *upwards closed* with respect to the sub-word order on channel contents. This last problem generalises (non)termination and is thus non-primitive recursive. Our main result is that the invariance problem for insertion channel systems is primitive recursive for downward closed predicates. More generally, we show primitive recursiveness of the associated function problem: compute the set $EG(L)$ of all configurations from which there is an infinite computation, all of whose configurations satisfy a given downward closed predicate L. This is not computable for lossy channel systems even for trivial predicates. In particular, taking L be the set of all configurations, the problem specialises to *structural (non)termination*: compute the set of states from which there is an infinite computation. Structural termination was shown to be undecidable for lossy channel machines in [May03].

Given a channel machine $\mathcal{S}$ and two distinguished control states p and q of $\mathcal{S}$, a *response* property is an assertion that every p state is always eventually followed by a q state in any infinite computation of $\mathcal{S}$. Note that a counterexample to a response property is a computation that eventually visits p and forever avoids q afterwards. The undecidability of response properties for lossy channel machines follows easily from that of structural termination, as the reader may wish to verify.

In the case of insertion channel machines, response properties are decidable, albeit at non-primitive recursive cost (by reduction from reachability). For decidability one first shows using the theory of well-structured transition systems that the set of all reachable configurations, the set of p-configurations, and the set of configurations that have infinite q-avoiding computations are all effectively computable. It then suffices to check whether their mutual intersection is empty.

	Lossy Channel Machines	**Insertion Channel Machines**
Reachability	non-primitive recursive	non-primitive recursive
Termination	non-primitive recursive	non-elementary / primitive recursive
Invariance (D)	non-primitive recursive	non-elementary / primitive recursive
Invariance (F)	undecidable	non-elementary / primitive recursive
Response	undecidable	non-primitive recursive
Recurrence	undecidable	undecidable
CTL / LTL	undecidable	undecidable

Fig. 1. Complexity of decision problems for faulty channel machines.

The *recurrence* problem asks, given a channel machine and a distinguished control state, whether the machine has a computation that visits the distinguished state infinitely often. It is undecidable for lossy channel machines by reduction from response, and was shown to be undecidable for insertion channel machines in [OW06a].

Finally, *CTL and LTL model checking* for both lossy and insertion channel machines are undecidable, which can be established along the same lines as the undecidability of recurrence.

These results are summarised in Figure 1. Here, when we refer to the invariance problem, we consider upward-closed predicates for lossy channel systems and downward closed predicates for insertion channel systems (the predicates typically used in practice are the system's control states, both upward-closed and downward-closed). The decision and function versions of the invariance problem are indicated with (D) and (F) respectively.

3. Definitions

A *channel machine* is a tuple $\mathcal{S} = (Q, \Sigma, C, \Delta)$, where Q is a finite set of control states, Σ is a finite channel alphabet, C is a finite set of channel names, and $\Delta \subseteq Q \times Op \times Q$ is the transition relation, where $Op = \{c!a, c?a, c{=}\varepsilon : c \in C, a \in \Sigma\}$ is the set of transition operations. Intuitively, label $c!a$ denotes the writing of message a to the tail of channel c, label $c?a$ denotes the reading of message a from the head of channel c, and label $c{=}\varepsilon$ tests channel c for emptiness.

We first define an *error-free* operational semantics for channel machines. Given $\mathcal{S}$ as above, a *configuration* of $\mathcal{S}$ is a pair (q, U), where $q \in Q$ is the control state and $U \in (\Sigma^*)^C$ gives the contents of each channel. Let us write *Conf* for the set of possible configurations of $\mathcal{S}$. By a slight abuse of notation we sometimes denote the contents of channel c in a configuration γ by $\gamma(c)$.

The rules in Δ induce an *Op*-labelled transition relation on *Conf*, as follows:

(1) $(q, c!a, q') \in \Delta$ yields a transition $(q, U) \xrightarrow{c!a} (q', U')$, where $U'(c) = U(c)\cdot a$ and $U'(d) = U(d)$ for $d \neq c$. *In other words, the channel machine moves from control state q to control state q', writing message a to the tail of channel c and leaving all other channels unchanged.*

(2) $(q, c?a, q') \in \Delta$ yields a transition $(q, U) \xrightarrow{c?a} (q', U')$, where $U(c) = a\cdot U'(c)$ and $U'(d) = U(d)$ for $d \neq c$. *In other words, the channel machine reads message a from the head of channel c while moving from control state q to control state q', leaving all other channels unchanged.*

(3) $(q, c{=}\varepsilon, q') \in \Delta$ yields a transition $(q, U) \xrightarrow{c=\varepsilon} (q', U)$, provided $U(c)$ is the empty word. *In other words, the transition is only enabled if channel c is empty; all channel contents remain the same.*

If the only transitions allowed are those listed above, then we call $\mathcal{S}$ an *error-free* channel machine. This machine model is easily seen to be Turing powerful [BZ83]. As discussed earlier, however, we are interested in channel machines with (potential) *insertion errors*; intuitively, such errors are modelled by postulating that channels may at any time acquire additional messages interspersed throughout their current contents.

For our purposes, it is convenient to adopt a *lazy* model of insertion errors, given next. Slightly different models, such as those of [CFPI96, OW06a], have also appeared in the literature. As the reader may easily check, all these models are equivalent insofar as reachability and termination properties are concerned.

The lazy operational semantics for channel machines with insertion errors simply augments the transition relation on *Conf* with the following rule:

(4) $(q, c?a, q') \in \Delta$ yields a transition $(q, U) \xrightarrow{c?a} (q', U)$. *In other words, insertion errors occur 'just in time', immediately prior to a read operation; all channel contents remain unchanged.*

The channel machines defined above are called *insertion channel machines with emptiness testing*, or *ICMET*s. A machine that does not feature any transitions following rule (3) above is simply an *insertion channel machine*, or *ICM*.

A *run* of an insertion channel machine is a finite or infinite sequence of transitions of the form $\gamma_0 \xrightarrow{op_0} \gamma_1 \xrightarrow{op_1} \ldots$ that is consistent with the lazy operational semantics.

3.1. Termination and Invariance

One of our main interests in this paper is the following problem:

Termination Problem for ICMETs. Given an ICMET $\mathcal{S}$ and a configuration γ of $\mathcal{S}$, are all runs of $\mathcal{S}$ starting from γ finite?

Note that the same question for ICMs (without emptiness tests) trivially reduces to termination for the underlying finite automaton. By contrast, we will give a non-elementary lower bound for the termination problem for ICMETs.

When considering upper bounds we work with a more general problem—the invariance problem for ICMs. To introduce this we need some preliminary notions.

Let $u = u_1u_2 \ldots u_m$ and $v = v_1v_2 \ldots v_n$ be words in Σ^*. We say that u is a *subword* of v, written $u \preccurlyeq v$, if there is an order embedding $f : \{1, \ldots, m\} \to \{1, \ldots, n\}$ such that $u_i = v_{f(i)}$. Intuitively this just says that u can be obtained from v by erasing some letters. For example, *higman* is a subword of *highmountain.* The subword ordering naturally extends to the set *Conf* of configurations of a channel machine $\mathcal{S}$ by stipulating that $(q, U) \preccurlyeq (q', U')$ iff $q = q'$ and $U(c) \preccurlyeq U'(c)$ for each channel c.

A set of configurations $L \subseteq \mathit{Conf}$ is said to be *downward closed* or a *lower set* if $\gamma \in L$ and $\gamma' \preccurlyeq \gamma$ imply that $\gamma' \in L$. Given a lower set $L \subseteq \mathit{Conf}$, let M be the set of minimal elements of $\mathit{Conf} \setminus L$. It follows from Higman's Lemma [Hig52] that M is finite and that every element of $\mathit{Conf} \setminus L$ is above some member of M. We call M the *cobasis* of L. The existence of M implies in particular that L is regular. We henceforth assume that L is represented by a non-deterministic automaton.

Given a channel machine $\mathcal{S} = (Q, \Sigma, C, \Delta)$, and a lower set $L \subseteq \mathit{Conf}$, define $EG(L)$ to be the set of configurations γ_0 from which there is an infinite computation $\gamma_0 \to \gamma_1 \to \cdots \to \gamma_n \to \cdots$ with $\gamma_i \in L$ for all $i \in \mathbb{N}$. Here we have adopted the notation of computation tree logic: $EG(L)$ is read 'there exists a path along which L holds globally'. We are interested in the following decision problem:

Invariance Problem for ICMs. Given a channel machine $\mathcal{S}$, a lower set $L \subseteq \mathit{Conf}$, and a configuration γ of $\mathcal{S}$, is $\gamma \in EG(L)$?

Example 1. If, for some location q_0 and channel c, we have $L = \{(q, U) : q = q_0 \text{ implies } U(c) = \varepsilon\}$, then for any computation that is globally in L channel c must be empty upon entering location q_0. Thus we can encode the termination problem for ICMETs as a special case of the invariance problem for ICMs.

More generally than emptiness tests $c = \varepsilon$ one can consider *avoidance tests* $w \not\preccurlyeq c$, which check that a given word w is not a subword of the contents channel c. Using the fact that every downward closed set of configurations has a finite cobasis, one can give polynomial-time translations in both directions between the termination problem for ICMs with avoidance tests and the invariance problem for ICMs.

4. Termination is Non-Elementary

In this section, we show that the termination problem for ICMETs is non-elementary. More precisely, we show that the termination problem for ICMETs of size n in the worst case requires time at least $2 \Uparrow \Omega(\log n)$. (The expression $2 \Uparrow m$, known as tetration, denotes an exponential tower of 2s of height m.)

Our proof proceeds by reduction from the termination problem for two-counter machines in which the counters are tetrationally bounded; the result then follows from standard facts in complexity theory (see, e.g., [HU79]).

```
Procedure $\textsc{Inc}(u_{k+1})$
  $\textsc{Reset}(u_k)$
  repeat
    $c?x\,;\, d!(1-x)$       /* Increment counter $u_{k+1}$ while transferring $c$ to $d$ */
    $\textsc{Inc}(u_k)$
  until $\textsc{IsZero}(u_k)$ or $x = 0$
  while not $\textsc{IsZero}(u_k)$ do
    $c?x\,;\, d!x$            /* Transfer remainder of $c$ to $d$ */
    $\textsc{Inc}(u_k)$
  endwhile
  test($c{=}\varepsilon$)            /* Check that there were no insertion errors on $c$, otherwise halt */
  repeat
    $d?x\,;\, c!x$            /* Transfer $d$ back to $c$ */
    $\textsc{Inc}(u_k)$
  until $\textsc{IsZero}(u_k)$
  test($d{=}\varepsilon$)            /* Check that there were no insertion errors on $d$, otherwise halt */
  return
```

Fig. 2. Procedure to increment counter u_{k+1}. Initially, this procedure assumes that counter u_{k+1} is encoded in binary on channel c, with least significant bit at the head of the channel; moreover, c is assumed to comprise exactly $2{\Uparrow}k$ bits (using padding 0s if need be). In addition, channel d is assumed to be initially empty. Upon exiting, channel c will contain the incremented value of counter u_{k+1} (modulo $2{\Uparrow}(k+1)$) in binary, again using $2{\Uparrow}k$ bits, and channel d will be empty. We regularly check that no insertion errors have occurred on channels c or d by making sure that they contain precisely the right number of bits. This is achieved using counter u_k (which can count up to $2{\Uparrow}k$ and is assumed to work correctly) together with emptiness tests on c and d. If an insertion error does occur during execution, the procedure will either halt, or the next procedure to handle channels c and d (i.e., any command related to counter u_{k+1}) will halt.

Without insertion errors, it is clear that a channel machine can directly simulate a two-counter machine simply by storing the values of the counters on one of its channels. To simulate a counter machine in the presence of insertion errors, however, we require periodic integrity checks to ensure that the representation of the counter values has not been corrupted. Below we give a simulation that follows the 'yardstick' construction of Meyer and Stockmeyer [SM73, LNO+08]: roughly speaking, we use an m-bounded counter to check the integrity of a 2^m-bounded counter.

Theorem 2. The termination problem for ICMETs is non-elementary.

Proof. Let us say that a counter is m-bounded if it can take values in $\{0, 1, \ldots, m-1\}$. We assume that such a counter u comes equipped with procedures $\textsc{Inc}(u)$, $\textsc{Dec}(u)$, $\textsc{Reset}(u)$, and $\textsc{IsZero}(u)$, where $\textsc{Inc}$ and $\textsc{Dec}$ operate modulo m, and increment, resp. decrement, the counter. We show how to simulate a deterministic counter machine $\mathcal{M}$ of size n equipped with two $2{\Uparrow}n$-bounded counters by an ICMET $\mathcal{S}$ of size $2^{O(n)}$. We use this simulation to reduce the termination problem for $\mathcal{M}$ to the termination problem for $\mathcal{S}$.

By induction, assume that we have constructed an ICMET $\mathcal{S}_k$ that can simulate the operations of a $2{\Uparrow}k$-bounded counter u_k. We assume that $\mathcal{S}_k$ correctly implements the operations $\textsc{Inc}(u_k)$, $\textsc{Dec}(u_k)$, $\textsc{Reset}(u_k)$, and $\textsc{IsZero}(u_k)$ (in particular, we assume that the simulation of these operations by $\mathcal{S}_k$ is guaranteed to terminate). We describe an ICMET $\mathcal{S}_{k+1}$ that implements a $2{\Uparrow}(k+1)$-bounded counter u_{k+1}. $\mathcal{S}_{k+1}$ incorporates $\mathcal{S}_k$, and thus can use the above-mentioned operations on the counter u_k as subroutines. In addition, $\mathcal{S}_{k+1}$ has two extra channels c and d on which the value of counter u_{k+1} is stored in binary. We give a high-level description.

We say that a configuration of $\mathcal{S}_{k+1}$ is *clean* if channel c has size $2{\Uparrow}k$ and channel d is empty. We ensure that all procedures on counter u_{k+1} operate correctly when they are invoked in clean configurations of $\mathcal{S}_{k+1}$, and that they also yield clean configurations upon completion. In fact, we only give details for the procedure $\textsc{Inc}(u_{k+1})$—see Figure 2; the others should be clear from this example.

Since the counter u_k is assumed to work correctly, the above procedure is guaranteed to terminate, having produced the correct result, in the absence of any insertion errors on channels c or d. On the other hand, insertion errors on either of these channels will be detected by one of the two emptiness tests, either immediately or in the next procedure to act on them.

The initialisation of the induction is handled using an ICMET $\mathcal{S}_1$ with no channel (in other words, a finite automaton) of size 2, which can simulate a 2-bounded counter (i.e., a single bit). The finite control of the counter machine, likewise, is duplicated using a further channel-less ICMET.

Using a product construction, it is straightforward to conflate these various ICMETs into a single one, $\mathcal{S}$, of size exponential in n (more precisely: of size $2^{O(n)}$). As the reader can easily check, $\mathcal{M}$ has an infinite computation iff $\mathcal{S}$ has an infinite run. The result follows immediately. $\square$

5. Invariance is Primitive Recursive

The central result of our paper is the following:

Theorem 3. The invariance problem for ICMs is primitive recursive. More precisely, when restricting to the class of ICMs that have at most k channels, the invariance problem is in $(k+1)$-EXPSPACE.

To add some colour to the development below, we consider a slight specialisation of the invariance problem. Given a channel alphabet Σ we assume a distinguished symbol $\lhd \notin \Sigma$, akin to the end-of-tape marker in Turing machines. We suppose that the lower set L in the statement of the invariance problem specifies, amongst other things, that at most one instance of $\lhd$ be on each channel at any time.

For each channel c we postulate a special operation $c?!\lhd$ that consists of consecutive read and write operations of the special symbol $\lhd$ on channel c. We assume that this is the only type of operation mentioning $\lhd$ (i.e., there are no operations that just read or just write $\lhd$). Given a transition $\gamma \xrightarrow{c?!\lhd} \gamma'$ such that $\gamma, \gamma' \in L$, it must be that $\lhd$ is on the head of $\gamma(c)$, is successfully read by the operation and is placed on the tail of $\gamma'(c)$, that is, the special operation $c?!\lhd$ is not subject to insertion errors as long as we remain in L. We call a transition $\gamma \xrightarrow{c?!\lhd} \gamma'$ a *cycling transition.* It is straightforward that, as far as termination and invariance are concerned, the above conventions are harmless.

Note that given a segment of a computation that begins and ends with cycling transitions on channel c,

$$\gamma_0 \xrightarrow{c?!\lhd} \gamma_1 \longrightarrow \gamma_2 \longrightarrow \cdots \longrightarrow \gamma_{n-1} \xrightarrow{c?!\lhd} \gamma_n \,,$$

it must hold that the length of channel c in the initial configuration $|\gamma_0(c)|$ is at most the length n of the computation.

Observe also that in an infinite computation of an ICM for each channel c, either there are infinitely many cycling transitions or the machine eventually stops consuming letters from channel c.

5.1. Rank Functions and Equivalence Relations

Throughout this section let $\mathcal{S} = (Q, \Sigma, C, \Delta)$ be an ICM and $L \subseteq \mathit{Conf}$ a lower set of configurations with cobasis M. Motivated by Example 1, define a *test* to be a pair (c, w) consisting of a channel $c \in C$ and word $w \in \Sigma^*$. Intuitively a configuration (q, U) fails the test if $w \preccurlyeq U(c)$. Define *Test* be the set $\{(c, w) : \exists \gamma \in M, \gamma(c) = w\}$. Next we introduce rank functions to measure how close a configuration is to failing a test.

Definition 4. Let (c, w) be a test with $w = w_1 \dots w_m \in \Sigma^*$. Given a configuration γ of $\mathcal{S}$ with $\gamma(c) = x \lhd y$, where $x, y \in \Sigma^*$, let $w_1 w_2 \dots w_l$ be the longest prefix of w that is a subword of x, and $w_{l+1} w_{l+2} \dots w_{l+p}$ the longest prefix of $w_{l+1} w_{l+2} \dots w_m$ that is a subword of y. Then define $\mathit{rank}_{c,w}(\gamma) = w_1 \dots w_l \lhd w_{l+1} \dots w_{l+p}$. Note that $w_1 \dots w_{l+p}$ is the longest prefix of w that embeds in $\gamma(c)$.

Example 5. Suppose $\Sigma = \{x, y, z\}$. If $w = xyz$ and $\gamma(c) = yx \lhd zxy$, then $\mathit{rank}_{c,w}(\gamma) = x \lhd y$.

The following proposition, whose proof is immediate, characterises the different ways in which $\mathit{rank}_{c,w}$ can change across a transition of $\mathcal{S}$.

Proposition 6. Suppose that $\gamma \xrightarrow{op} \gamma'$ is a transition of $\mathcal{S}$ such that $\mathit{rank}_{c,w}(\gamma) = u \lhd v$ and $\mathit{rank}_{c,w}(\gamma') = u' \lhd v'$. If $\mathit{rank}_{c,w}(\gamma) \neq \mathit{rank}_{c,w}(\gamma')$ then there are three possibilities:

(i) $op = c!\sigma$ for some $\sigma \in \Sigma$, $u' = u$, and $v' = v\sigma$

(ii) $op = c?\sigma$ and $|u'| < |u|$

(iii) $op = c?!\lhd$, $u = \varepsilon$, $v' = \varepsilon$, and $u' = v$.

Lemma 7 (Cycling Lemma). Let $\rho = \gamma_0 \longrightarrow \gamma_1 \longrightarrow \cdots \longrightarrow \gamma_m$ be a computation of $\mathcal{S}$ and (c, w) a test such that there are at least $(|w| + 1)^2$ transitions $\gamma_i \longrightarrow \gamma_{i+1}$ with $\mathit{rank}_{c,w}(\gamma_i) \neq \mathit{rank}_{c,w}(\gamma_{i+1})$. Then ρ contains a cycling transition on channel c.

Proof. Consider the subsequence of transitions in ρ that change $\mathit{rank}_{c,w}$, Of every $|w| + 1$ consecutive such transitions at least one must be a read operation since the length of $\mathit{rank}_{c,w}(\gamma)$ is bounded by $|w| + 1$. Thus ρ contains at least $|w| + 1$ read operations that change $\mathit{rank}_{c,w}$. Considering all such read operations we observe that Case (ii) in Proposition 6 cannot apply to all these operations (due to the condition $|u'| < |u|$ therein) and that Case (iii) must apply at least once, i.e., there exists a cycling transition. □

The proof of our main result, Theorem 9, involves an induction on the number of channels. To set this up we introduce the following indexed family of equivalence relations. Given a set of channels $D \subseteq C$, define the equivalence $\equiv_D$ on Conf by $(q, U) \equiv_D (q', U')$ iff $q = q'$ and $U(c) = U'(c)$ for all $c \in D$. The coarsest such equivalence is $\equiv_\emptyset$, which represents equality of control locations, and the finest such equivalence is $\equiv_C$, which denotes complete equality of configurations.

Lemma 8 (Pumping Lemma). Suppose that $\rho = \gamma_0 \longrightarrow \gamma_1 \longrightarrow \cdots \longrightarrow \gamma_n$ is a computation of $\mathcal{S}$ such that $\gamma_i \in L$ for $0 \leq i \leq n$ and $\gamma_0 \equiv_D \gamma_n$ for some set of channels $D \subseteq C$. If $\mathit{rank}_{c,w}$ is constant along ρ for all $(c, w) \in \mathit{Test}$ such that $c \notin D$ then $\gamma_0 \in \mathit{EG}(L)$.

Proof. Note that $\gamma_0 \equiv_D \gamma_n$ implies that configurations γ_0 and γ_n have the same underlying control state. Thus we can extend ρ to an infinite computation ρ':

$$\gamma_0 \longrightarrow \gamma_1 \longrightarrow \cdots \longrightarrow \gamma_n \longrightarrow \gamma_{n+1} \longrightarrow \cdots \longrightarrow \gamma_{2n} \longrightarrow \gamma_{2n+1} \longrightarrow \cdots$$

by repeatedly firing the finite sequence of transitions that occur along ρ. Moreover, since $\gamma_0 \equiv_D \gamma_n$, it holds that $\gamma_i \equiv_D \gamma_j$ for all i, j such that $i \equiv j \pmod n$. It remains to show that all configurations in the infinite computation ρ' lie in the lower set L.

Pick a configuration γ in the cobasis M and a configuration γ_i appearing in ρ'. We must show that $\gamma \not\preccurlyeq \gamma_i$. Writing $j = i \bmod n$, we have $\gamma_j \equiv_D \gamma_i$, i.e., γ_j and γ_i have the same control state and agree on channels $c \in D$.

Since all configurations in ρ lie in L it holds that $\gamma \not\preccurlyeq \gamma_j$. We now consider three cases. The first case is that γ and γ_j have different control states. Then γ and γ_i also have different control states and $\gamma \not\preccurlyeq \gamma_i$. The second case is that there exists a channel $c \in D$ with $\gamma(c) \not\preccurlyeq \gamma_j(c)$. But $\gamma_i(c) = \gamma_j(c)$ and thus $\gamma(c) \not\preccurlyeq \gamma_i(c)$. The remaining case is that there exists a channel $c \notin D$ with $\gamma(c) \not\preccurlyeq \gamma_j(c)$. Writing $\gamma(c) = w = w_1 \dots w_m$, there is by assumption a unique value assumed by $\mathit{rank}_{c,w}$ along ρ. This value has the form $w_1 \dots w_l \lhd w_{l+1} \dots w_k$, where $w_1 w_2 \dots w_k$ is the longest prefix of w that is a subword of $\gamma_j(c)$. In particular, $k < m$ and w_{k+1} is not written on channel c along ρ, nor, therefore, along ρ'. We conclude that $w \not\preccurlyeq \gamma_i(c)$. This completes the case analysis, and we conclude that $\gamma \not\preccurlyeq \gamma_i$. □

5.2. The Main Argument

Once again, let $\mathcal{S} = (Q, \Sigma, C, \Delta)$ by an ICM and $L \subseteq \mathit{Conf}$ a lower set with cobasis M.

Given a computation $\rho = \gamma_0 \longrightarrow \gamma_1 \longrightarrow \cdots \longrightarrow \gamma_m$ of $\mathcal{S}$, we say that a set $S \subseteq \mathit{Conf}$ is *α-frequent* in ρ, where $0 \leq \alpha \leq 1$, if ρ visits S at least αm times. We also say that $\equiv_D$ has *index β on S* if it partitions S into β equivalence classes.

Theorem 9. The invariance problem for ICMs is primitive-recursive.

Proof. Let $\mathcal{S} = (Q, \Sigma, C, \Delta)$ be a channel machine and $L \subseteq \mathit{Conf}$ a lower set with cobasis M. Denote by n the joint size of $\mathcal{S}$ and M (under some reasonable encoding). Clearly we can assume that

$$\sum_{(c,w) \in \mathit{Test}} (|w| + 1) \leq n\,. \tag{1}$$

Suppose that $\gamma_0 \notin \mathit{EG}(L)$. We show that there exist α and β, with β/α primitive-recursive in n, such that for any run ρ in L originating in γ_0, there exists $S \subseteq \mathit{Conf}$ that is α-frequent in ρ and has cardinality β.

It follows that any such run has length at most β/α; otherwise it would contain two identical configurations, contradicting the fact that $\gamma_0 \notin \mathit{EG}(L)$. But then the invariance problem can be decided in primitive-recursive time just by exploring all runs of length β/α.

We establish the existence of α and β with the desired properties by induction on the set of channels C. The induction hypothesis is that for any $D \subseteq C$ there exist α and β, with β/α in $n^{n^{\cdot^{n^{O(1)}}}}$ (where the tower of n's has height $|D|+1$), such that for any run ρ in L originating in γ_0, there exists $S \subseteq \mathit{Conf}$ that is α-frequent in ρ and such that $\equiv_D$ has index at most β on S.

For the base case, $D = \emptyset$, define $\alpha = 1$ and $\beta = |Q|$. Then given a computation $\rho = \gamma_0 \longrightarrow \gamma_1 \longrightarrow \cdots \longrightarrow \gamma_m$, $S = \{\gamma_i \colon 0 \leq i \leq m\}$ is α-frequent in ρ, and $\equiv_\emptyset$ has index at most β on S.

The inductive step is given in Lemma 10, below. Roughly speaking, the idea of the proof is as follows. If for all tests $(c,w) \in \mathit{Test}$ with $c \notin D$ there is a sufficiently long computation over which $\mathit{rank}_{c,w}$ remains constant, then applying the Pumping Lemma, Lemma 8, we conclude that $\gamma_0 \in \mathit{EG}(L)$, a contradiction. On the other hand, if $\mathit{rank}_{c,w}$ changes with sufficient frequency along a computation, then the Cycling Lemma, Lemma 7, gives an upper bound on the frequency of cycling transitions on channel c, and therefore a bound on the size of channel c.

□

Lemma 10. Suppose that $\rho = \gamma_0 \longrightarrow \gamma_1 \longrightarrow \cdots \longrightarrow \gamma_m$ is a computation of $\mathcal{S}$, where $\gamma_i \in L$ for $0 \leq i \leq m$, but $\gamma_0 \notin \mathit{EG}(L)$. Let $S \subseteq \mathit{Conf}$ be a set of configurations that is α-frequent in ρ, and let D be a proper subset of the set of channels C such that $\equiv_D$ has index β on S. Then there exists $S' \subseteq S$ and $D' \supsetneq D$ such that S' is α'-frequent in ρ, $\equiv_{D'}$ has index β' on S', and

$$(\beta'/\alpha') \leq n^{k(\beta/\alpha)} \tag{2}$$

for some k polynomial in n.

Proof. The run ρ visits S at least αm times, and out of every β such visits two configurations are equivalent under $\equiv_D$. Hence we can identify in ρ a subsequence $\gamma_0 \xrightarrow{*} \gamma_{i_1} \xrightarrow{+} \gamma_{i'_1} \xrightarrow{*} \gamma_{i_2} \xrightarrow{+} \gamma_{i'_2} \xrightarrow{*} \cdots \xrightarrow{*} \gamma_{i_k} \xrightarrow{+} \gamma_{i'_k} \xrightarrow{*} \gamma_m$, where $\gamma_{i_j}\gamma_{i'_j} \in S$ are *matching pairs* of configurations with $\gamma_{i_j} \equiv_D \gamma_{i'_j}$, and $k \geq \alpha m/\beta$.

Next we partition ρ into segments, called *blocks*, where each block contains $2n^2$ consecutive matching pairs. Denoting the number of blocks by p, we have

$$p \geq \frac{\alpha m}{2n^2\beta}\,. \tag{3}$$

The p blocks have various lengths, but we are interested in the shortest ones. In particular, the $p/2$ shortest blocks must all be shorter than $\ell \overset{\text{def}}{=} 2m/p$, otherwise the $p/2$ longer blocks would not fit in ρ. Now (3) entails that

$$\ell \leq \frac{4n^2\beta}{\alpha}\,. \tag{4}$$

Since $\gamma_0 \notin \mathit{EG}(L)$, Lemma 8 (the Pumping Lemma) tells us that for each matching pair $\gamma_{i_j}\gamma_{i'_j}$ there exists a test (c,w), $c \notin D$, such that $\mathit{rank}_{c,w}$ is not constant along the segment $\gamma_{i_j} \xrightarrow{+} \gamma_{i'_j}$ in ρ. But each block contains $2n^2$ matching pairs, so by (1) for each block there is a test (c,w) such that $\mathit{rank}_{c,w}$ changes at least $2(|w|+1)^2$ times within that block. Then, by Lemma 7, each block must contain two cycling transitions of some channel $c \notin D$.

Considering again the $p/2$ shortest blocks, there is a channel $d \in C \setminus D$ such that $\frac{p}{2|C|}$ of these blocks contains two cycling transitions of channel d. For any configuration γ in such a block, we have $|\gamma(d)| \leq \ell$ since the block has length at most ℓ.

Now define $S' = \{\gamma \in S : |\gamma(d)| \leq \ell\}$ and $D' = D \cup \{d\}$. Then S' is α'-frequent in ρ for $\alpha' = \frac{p}{2|C|m}$ and $\equiv_{D'}$ has index $\beta' \leq \beta|\Sigma|^\ell$ on S'. Employing the inequalities (3) and (4), we conclude that

$$(\alpha')^{-1} \leq 4n^3\beta\alpha^{-1} \quad \text{and} \quad \beta' \leq \beta n^{4n^2\beta\alpha^{-1}}\,. \tag{5}$$

Finally, multiplying the two inequalities in (5) yields the desired inequality (2) for a suitable choice of k. □

Remark 11. From the proof of Theorem 9 we see that the invariance problem for an ICM $\mathcal{S}$ with at most k channels is in $(k+1)$-EXPSPACE. Indeed the induction hypothesis shows that such a machine has an infinite computation all of whose configurations are in the given lower set L if and only if it has a finite computation of length at least $n^{n^{\cdot^{n^{O(1)}}}}$, where n denotes the size of $\mathcal{S}$ and the cobasis of L, and the tower of n's has height $k+1$.

Corollary 12. Given an ICM and a lower set of configurations L, the problem of computing $EG(L)$ is primitive recursive. In particular, the structural termination problem—are all computations of the machine finite, starting from the initial control state but regardless of the initial channel contents?—is primitive recursive for ICMETs.

Proof. Given a channel system $\mathcal{S}$ with set of configurations *Conf* we define the *one-step predecessor* operation by

$$\mathit{Pred}(S) = \{\gamma \in \mathit{Conf} : \exists \gamma' \in S.\, \gamma \longrightarrow \gamma\}\,,$$

for $S \subseteq \mathit{Conf}$. Two key properties of *Pred* are (i) $\mathit{Pred}(S)$ is downward closed whenever S is downward closed; (ii) given an automaton representing a downward closed set S, one can compute (in polynomial time) an automaton representing $\mathit{Pred}(S)$. Property (i) follows immediately from operational semantics of ICM (in particular, the property that read-transitions are always enabled). Property (ii) can be broken down into two steps. If we represent lower sets as languages of finite nondeterministic automata, then computing unions and intersections of lower sets is a straightforward polynomial-time operation. Thus it suffices to show the effectiveness of *Pred* for systems with only a single transition. This is straightforward and we omit the details.

Now the proof of Theorem 9 showed how to compute from a given channel machine $\mathcal{S}$ and lower set L a "threshold" θ such that a configuration γ satisfies $EG(L)$ if and only if it has a computation of length θ, all of whose configurations satisfy L. Thus, defining, $F : \mathit{Conf} \to \mathit{Conf}$ by $F(S) = L \cap \mathit{Pred}(L)$, we have that $EG(L) = F^{\theta}(\mathit{Conf})$. Since θ is primitive-recursive and F can be computed in exponential time, it follows that $EG(L)$ is primitive-recursive. $\square$

6. Metric Temporal Logic

Metric Temporal Logic (MTL) is a real-time extension of linear temporal logic [HMP92, Koy90] that is a formalism in real-time verification. *Prima facie* there is no connection between MTL and channel systems, however in this section we prove that the satisfiability problem for the safety fragment of MTL is non-elementary by reduction from the termination problem for ICMET.

A *time sequence* $\tau = \tau_1\tau_2\tau_3\ldots$ is an infinite sequence of time values $\tau_i \in \mathbb{R}_{\geq 0}$ that is strictly increasing and unbounded. A *timed word* over finite alphabet Σ is a pair $\rho = (\sigma, \tau)$, where $\sigma = \sigma_1\sigma_2\ldots$ is an infinite word over Σ and τ is a time sequence.

Definition 13. Given an alphabet Σ of atomic events, the formulas of MTL are built up from Σ by monotone Boolean connectives and time-constrained versions of the **next** operator $\bigcirc$, **until** operator $\mathcal{U}$ and the **always** operator $\Box$ as follows:

$$\varphi ::= \mathbf{true} \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid a \mid \bigcirc_I \varphi \mid \varphi_1\, \mathcal{U}_I\, \varphi_2 \mid \Box_I \varphi\,,$$

where $a \in \Sigma$, and $I \subseteq \mathbb{R}_{\geq 0}$ is an interval with endpoints in $\mathbb{N} \cup \{\infty\}$. We admit a derived **eventually** operator $\Diamond_I \varphi := \mathbf{true}\, \mathcal{U}_I\, \varphi$.

Definition 14. Given a timed word $\rho = (\sigma, \tau)$ and an MTL formula φ, the satisfaction relation $(\rho, i) \models \varphi$ (read ρ satisfies φ at position i) is defined by the following clauses (we omit the Boolean operators):

- $(\rho, i) \models a$ iff $\sigma_i = a$
- $(\rho, i) \models \bigcirc_I \varphi$ iff $\tau_{i+1} - \tau_i \in I$ and $(\rho, i+1) \models \varphi$
- $(\rho, i) \models \varphi_1\, \mathcal{U}_I\, \varphi_2$ iff there exists $j \geqslant i$ such that $(\rho, j) \models \varphi_2$, $\tau_j - \tau_i \in I$, and $(\rho, k) \models \varphi_1$ for all k such that $i \leqslant k < j$.
- $(\rho, i) \models \Box_I \varphi$ iff all $j \geqslant i$ such that $\tau_j - \tau_i \in I$, $(\rho, j) \models \varphi$

We say that ρ satisfies φ, denoted $\rho \models \varphi$, if $(\rho, 1) \models \varphi$.

Safety MTL is the fragment of MTL obtained by requiring that the interval I in each until operator $\mathcal{U}_I$ have finite length, i.e., all that eventualities be bounded. This restriction is quite natural from the point of view of system verification: it is not good enough to merely know that something will happen; one usually wants to know that it will happen reasonably soon. We are interested in the following problem.

Safety MTL Satisfiability. Is a given Safety MTL formula φ satisfied by some timed word?

The satisfiability problem for MTL over infinite timed words was shown to be undecidable in [OW06a] by reduction from the recurrence problem for ICMET. Decidability can be recovered either by a semantic restriction to finite words [OW05] or by a syntactic restriction to Safety MTL. (Observe that the recurrence property $\Box\Diamond a$, "a happens infinitely often", cannot be expressed in Safety MTL.) Decidability of Safety MTL was established in [OW06b] using Higman's lemma, however no upper or lower complexity bounds were given therein. The complexity of MTL over finite words is equivalent to that of reachability for ICMET, and therefore non-primitive recursive [OW05].

Below we give a reduction of the termination problem for ICMET to the satisfiability problem for Safety MTL, showing that the latter is non-elementary. Although this reduction uses ideas that are familiar from [OW05, OW06b, LW08], we provide details since neither the "backwards" encoding of lossy channel machines in [LW08] nor the encoding of ICMET in [OW05, OW06b] directly apply to the problem at hand.

Let $\mathcal{S} = (Q, C, \Sigma, \Delta)$ be an ICM. We define a timed language $L_{\mathcal{S}}$ over the alphabet $(C \times \Sigma) \cup \Delta$ whose words represent non-terminating computations of $\mathcal{S}$. To aid readability we denote a pair $(c, a) \in C \times \Sigma$ by c_a. A timed word (σ, τ) is in $L_{\mathcal{S}}$ iff

- The sequence of events is $\sigma = w_1\delta_1 w_2\delta_2 \ldots$, where $w_i \in (C \times \Sigma)^*$ and $\delta_i \in \Delta$. We require that $\delta_1\delta_2 \ldots$ be a legitimate path through the underlying control automaton. The idea is that w_i represents the channel contents of the i-th configuration and δ_i the ith transition.
- For all $i \geq 1$ the time of δ_i is i (transitions are encoded at integer time points)
- For all $i \geq 1$, if op_i the operation performed by δ_i, then:
  - For each event c_a at time t in the interval $(i-1, i)$, except possibly the first one, there is a matching event c_a at time $t+1$. The exception applies in case the first event is c_a and $op_i = c?a$. Thus we preserve the channel contents by copying events from one time unit to the next, except possibly the first event in case it is read from the channel. Note that there is nothing in this convention to prevent insertion errors on the channel.
  - If $op_i = c!a$ then the last event in the interval $(i, i+1)$ is c_a. If this event happens at time t then there is no event at time $t-1$ (corresponding to writing letter a to channel c).
  - If op_i is $c = \varepsilon$ then there are no events of the form c_a, $a \in \Sigma$, in the interval $(i-1, i)$.

A slightly subtle point is that the above encoding of ICMET computations corresponds to a more permissive model of insertion errors than the lazy model adopted in Section 3. Here we allow in addition that extra symbols may appear on the channel. However computations involving such "eager" insertion errors can clearly be simulated under the lazy model, so termination is unaffected.

From the description above, it is clear that $L_{\mathcal{S}}$ is non-empty if and only if $\mathcal{S}$ has a non-terminating computation. Next we show that there is a Safety MTL formula $\varphi_{\mathcal{S}}$ such that $L_{\mathcal{S}}$ is the set of timed words satisfying $\varphi_{\mathcal{S}}$. $\varphi_{\mathcal{S}}$ is defined as the conjunction of several formula, each one expressing a requirement in the definition of $L_{\mathcal{S}}$. We do give not formulas corresponding to the first two bullet points above, but focus on the third bullet point, which is the most interesting case.

- For each transition $\delta \in \Delta$, each channel $c \in C$ and letter $a \in \Sigma$, if the operation of δ is not $c?a$ then we include a formula

  $$\Box(c_a \wedge \Diamond_{(0,1)}\delta \rightarrow \Diamond_{\{1\}}c_a)\,.$$

  This formula ensures that every c_a-event at most one time unit before δ is matched by a corresponding c_a-event one time unit later. This corresponds to the fact that δ does not consume any letters from the channel.

Otherwise, if the operation of δ is $c?a$, then we instead include a formula

$$\Box(\bigcirc c_a \wedge \Diamond_{(0,1)}\delta \to \bigcirc\Diamond_{\{1\}}c_a)\,.$$

This formula ensures that every c_a-event at most one time unit before a given δ-event, except possibly the first such, is matched by a corresponding c_a-event one time unit later. This corresponds to the fact that after executing δ each letter remains on the channel except possibly for the letter at the head of the channel.

- For each transition $\delta \in \Delta$ whose operation is $c!a$ we include a formula

$$\Box(\bigcirc\delta \to \Diamond_{\{1\}} \bigcirc c_a)\,.$$

This formula ensures that the event immediately preceding δ is followed in one time unit by an event such that the very next event is c_a, i.e., a new letter a is written to the end of the channel.

- For each transition $\delta \in \Delta$ and each letter $a \in \Sigma$, if the operation of δ is $c = \varepsilon$ then we include a formula

$$\Box\neg(c_a \wedge \Diamond_{(0,1)}\delta)\,.$$

This formula says that c_a cannot occur less than one time unit before δ, i.e., the channel is empty immediately prior to a successful zero test.

With $\varphi_{\mathcal{S}}$ defined as above, it holds that $\varphi_{\mathcal{S}}$ is satisfiable if and only if $\mathcal{S}$ has a non-terminating computation. From Theorem 2 we deduce that

Theorem 15. The termination problem for ICMET is non-elementary.

7. Conclusion

The main result of this paper is that the invariance problem for insertion channel machines and downward closed predicates has non-elementary, yet primitive-recursive complexity. As a corollary we have shown that the termination problem for ICMETs is primitive-recursive. This result is in sharp contrast with the termination problem for lossy channel machines, which has non-primitive recursive complexity. Another difference between the two models is that while the function version of the invariance problem of ICMs is primitive-recursive the analogous problem is undecidable for lossy channel systems.

Another interesting difference with lossy channel machines can be highlighted by quoting a slogan from [Sch02]: "*Lossy systems with k channels can be [polynomially] encoded into lossy systems with one channel.*" We can deduce from Theorems 2 and 3 that any such encoding, in the case of insertion channels machines, would require non-elementary resources to compute, if it were to preserve termination properties.

Our original motivation for studying ICMs was in connection with Metric Temporal Logic. While Metric Temporal Logic is undecidable over infinite time words [OW06a], satisfiability for the safety fragment (in which all eventualities are time-bounded) is decidable [OW06b]. However the decidability proof in [OW06b] used general results about well-structured systems and yielded no complexity bounds. In this paper we have given a non-elementary lower bound for the satisfiability problem for Safety Metric Temporal Logic by reduction from the termination problem for ICMET. We leave open whether the former problem is primitive recursive.

References

[AČJT00] Parosh Aziz Abdulla, Karlis Čerans, Bengt Jonsson, and Yih-Kuen Tsay. Algorithmic analysis of programs with well quasi-ordered domains. *Information and Computation*, 160(1/2):109–127, 2000.

[AJ93] Parosh Aziz Abdulla and Bengt Jonsson. Verifying programs with unreliable channels. In *Proc. LICS '93*, pages 160–170. IEEE Computer Society Press, 1993.

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

[BBS06] Christel Baier, Nathalie Bertrand, and Philippe Schnoebelen. On computing fixpoints in well-structured regular model checking, with applications to lossy channel systems. In *Proc. LPAR 2006*, volume 4246 of *Lecture Notes in Artificial Intelligence*, pages 347–361. Springer, 2006.

[BMO+08] Patricia Bouyer, Nicolas Markey, Joël Ouaknine, Philippe Schnoebelen, and James Worrell. On termination for faulty channel machines. In *Proc. STACS 2008*, volume 1 of *LIPIcs*, pages 121–132. Schloß Dagstuhl - Leibniz-Zentrum für Informatik, 2008.

[BZ83] Daniel Brand and Pitro Zafiropulo. On communicating finite-state machines. *Journal of the ACM*, 30(2):323–342, 1983.

[CFPI96] 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.

[CS08] Pierre Chambart and Philippe Schnoebelen. The ordinal recursive complexity of lossy channel systems. In *Proc. LICS 2008*, pages 205–216. IEEE Computer Society, 2008.

[Fin94] Alain Finkel. Decidability of the termination problem for completely specificied protocols. *Distributed Computing*, 7(3):129–135, 1994.

[FS01] Alain Finkel and Philippe Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1–2):63–92, 2001.

[Hig52] Graham Higman. Ordering by divisibility in abstract algebras. *Proceedings of the London Mathematical Society*, 2(7):326–336, 1952.

[HMP92] Thomas A. Henzinger, Zohar Manna, and Amir Pnueli. What good are digital clocks? In *Proc. 19th International Colloquium on Automata, Languages and Programming (ICALP'92)*, volume 623 of *Lecture Notes in Computer Science*, pages 545–558. Springer, 1992.

[HU79] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.

[Koy90] Ron Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.

[Laz11] Ranko Lazić. Safety alternating automata on data words. *ACM Trans. Comput. Log.*, 12(2):10, 2011.

[LNO$^+$08] Ranko Lazić, Tom Newcomb, Joël Ouaknine, A. W. Roscoe, and James Worrell. Nets with tokens which carry data. *Fundam. Inform.*, 88(3):251–274, 2008.

[LW08] Slawomir Lasota and Igor Walukiewicz. Alternating timed automata. *ACM Trans. Comput. Log.*, 9(2), 2008.

[May03] Richard Mayr. Undecidable problems in unreliable computations. *Theoretical Computer Science*, 297(1):35–65, 2003.

[OW05] Joël Ouaknine and James Worrell. On the decidability of Metric Temporal Logic. In *Proc. LICS 2005*, pages 188–197. IEEE Computer Society Press, 2005.

[OW06a] Joël Ouaknine and James Worrell. On metric temporal logic and faulty Turing machines. In *Proc. FoSSaCS 2006*, volume 3921 of *Lecture Notes in Computer Science*, pages 217–230. Springer, 2006.

[OW06b] Joël Ouaknine and James Worrell. Safety metric temporal logic is fully decidable. In *Proc. TACAS 2006*, volume 3920 of *Lecture Notes in Computer Science*, pages 411–425. Springer, 2006.

[Rac78] Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6(2):223–231, 1978.

[Sch02] Philippe Schnoebelen. Verifying lossy channel systems has nonprimitive recursive complexity. *Information Processing Letters*, 83(5):251–261, 2002.

[SM73] Larry J. Stockmeyer and Albert R. Meyer. Word problems requiring exponential time: Preliminary report. In *Proc. STOC '73*, pages 1–9. ACM, 1973.

[SS11] Sylvain Schmitz and Philippe Schnoebelen. Multiply-recursive upper bounds with Higman's lemma. In *Proc. ICALP 2011*, volume 6756 of *Lecture Notes in Computer Science*, pages 441–452. Springer, 2011.