Zeno, Hercules and the Hydra: Downward Rational Termination Is Ackermannian

Ranko Lazić[1], Joël Ouaknine[2], and James Worrell[2]

[1] Department of Computer Science, University of Warwick, UK
[2] Department of Computer Science, University of Oxford, UK

Abstract. Metric temporal logic (MTL) is one of the most prominent specification formalisms for real-time systems. Over infinite timed words, full MTL is undecidable, but satisfiability for its safety fragment was proved decidable several years ago [18]. The problem is also known to be equivalent to a fair termination problem for a class of channel machines with insertion errors. However, the complexity has remained elusive, except for a non-elementary lower bound. Via another equivalent problem, namely termination for a class of rational relations, we show that satisfiability for safety MTL is not primitive recursive, yet is Ackermannian, i.e., among the simplest non-primitive recursive problems. This is surprising since decidability was originally established using Higman's Lemma, suggesting a much higher non-multiply recursive complexity.

1 Introduction

Metric temporal logic (MTL) is one of the most popular approaches for extending temporal logic to the real-time setting. MTL extends linear temporal logic by constraining the temporal operators with intervals of real numbers. For example, the formula $\Diamond_{[3,4]}\varphi$ means that φ will hold within 3 to 4 time units in the future. There are two main semantic paradigms for MTL: continuous (state-based) and pointwise (event-based)—cf. [3, 12]. In the former, an execution of a system is modelled by a flow which maps each point in time to the state propositions that are true at that moment. In the latter, one records only a countable sequence of events, corresponding to instantaneous changes in the state of the system. In this paper we interpret MTL over the pointwise semantics[3] and assume that time is *dense* (arbitrarily many events can happen in a single time unit) but *non-Zeno* (only finitely many events can occur in a single time unit).

Over the past few years, the theory of *well-structured transition systems* has been used to obtain decidability results for MTL. Well-structured transition systems are a general class of infinite-state systems for which certain verification problems, such as reachability and termination, are decidable; see [9] for a comprehensive survey. In [19] satisfiability and model checking for MTL were shown

[3] Note that it follows from the thesis work of Henzinger [11] that safety MTL satisfiability is undecidable over the continuous semantics.

to be decidable by reduction to the reachability problem for a class of well-structured transition systems. Likewise, for a syntactically defined fragment of MTL that expresses safety properties, called *safety MTL*, model checking and satisfiability were shown decidable over infinite timed words by reduction to the termination problem on well-structured transition systems [18].

Extracting well-structured systems from MTL formulas relies on Higman's Lemma, which states that over a finite alphabet the subword order is a well-quasi order. Analysis of termination arguments that use Higman's Lemma has been applied to bound the complexity of reachability in lossy channel systems and insertion (or gainy) channel systems: two classes of well-structured systems that arise naturally in the modelling of communication over faulty media. For the reachability and termination problem in lossy channel systems, an upper bound in level $\mathfrak{F}_{\omega^\omega}$ of the fast-growing hierarchy was obtained in [7]. (Recall that $\mathfrak{F}_{<\omega}$ comprises the primitive recursive functions, Ackermann's function lies in $\mathfrak{F}_\omega$, while $\mathfrak{F}_{\omega^\omega}$ contains the first non-multiply recursive function.) The same paper also shows that neither problem lies in a lower level of the hierarchy and observes that both lower and upper bounds carry over to MTL satisfiability over *finite* words and to reachability in insertion channel systems, among many other problems.[4] An upper bound in $\mathfrak{F}_{\omega^\omega}$ for safety MTL satisfiability has also been sketched in [21] using related techniques.

Meanwhile, complexity lower bounds for safety MTL have been obtained utilising a correspondence with the termination problem for insertion channel systems. In [4] it is shown that termination for insertion channel machines with emptiness tests is primitive recursive, though non-elementary.[5] This result is used to give a non-elementary lower bound in $\mathfrak{F}_3$ for the satisfiability problem for safety MTL. An improved lower bound in $\mathfrak{F}_4$ is given in [13], again via insertion channel machines, but still leaving a considerable gap with the above-mentioned $\mathfrak{F}_{\omega^\omega}$ upper bound. This gap was highlighted recently in [14].

The key to determining the precise complexity of satisfiability for safety MTL is to study a refined version of the termination problem for channel machines—namely the *fair termination problem.* Roughly speaking, an infinite computation of an insertion channel machine is *fair* if every message that is written to the channel is eventually consumed—and not continuously preempted by insertion errors. (In the translation between channel machines and MTL, fairness corresponds in a precise sense to the *non-Zenoness* assumption.) We obtain lower and upper complexity bounds for this problem that are *Ackermannian*, i.e., that lie in level $\mathfrak{F}_\omega$ of the fast-growing hierarchy. These bounds also apply to safety MTL satisfiability, finally closing the above-mentioned complexity gap.

Unlike [4], we consider channel machines with a single channel. In [4], without the hypothesis of fairness, the termination problem was shown to be non-

[4] Incidentally, the *model-checking* problem over infinite timed words for safety MTL against timed automata can also be shown to have complexity precisely in $\mathfrak{F}_{\omega^\omega}$, following arguments presented in [19] together with the results of [7].

[5] In the presence of insertion errors, read-transitions can always be taken, so the channel is redundant unless there is an extra hypothesis, such as emptiness tests.

elementary in the number of channels. On the other hand, fair termination is already undecidable if there are two channels. But with a single channel fair termination is non-primitive recursive in the size of the channel alphabet. In common with [4] we find that termination for insertion channels has a lower complexity than termination for lossy channel systems or reachability for either type of system, neither of which is multiply recursive.

Our technical development is carried out in a slightly more abstract framework than insertion channel systems. We study the termination problem for well-structured transition systems whose states are words over a given alphabet, and whose transition relation is a rational relation that is (downwards) compatible with the subword order. (This is similar to the basic framework of regular model checking [2], but with the additonal hypothesis of monotonicity.)

To obtain an Ackermannian upper bound, we associate a *Hydra battle* with each finite computation of such a system. For our purposes, a Hydra battle is a sequence of 'flat' regular expressions that express assertions about states in the computation. Each regular expression can be seen as arising from its precedessor by a process of truncation (by the sword of Hercules) and regeneration. Our Hydra correspond to the classical tree Hydra of Kirby and Paris [15] via a natural correspondence between flat regular expressions and trees of height 2.

The basic pattern for proving our lower bound result is a standard one, namely to reduce from the halting problem for Ackermannianly bounded Turing machines by simulating their computations. However, in contrast to the common approach in the literature, in which a large function and its inverse are computed weakly before and after the simulation respectively (cf. e.g. [7, 22, 14]), we bootstrap a counter that can count accurately to an Ackermannian bound even in the presence of insertion errors. The bootstrapping involves extending Stockmeyer's yardstick construction, which reaches beyond the elementary functions, to surpass all primitive recursive ones.

2 Preliminaries

2.1 Fast Growing Hierarchy

We define an initial segment of the fast growing hierarchy [16] of computable functions by following the presentation of Figueira et al. [8].

For each $k \in \mathbb{N}$, class $\mathfrak{F}_k$ is the closure under substitution and limited recursion of constant, sum and projection functions, and F_n functions for $n \leq k$. The latter are defined so that F_0 is the successor function, and each F_{n+1} is computed by iterating F_n:

$$F_0(x) = x + 1 \qquad\qquad F_{n+1}(x) = F_n^{x+1}(x)$$

The following are a few simple observations:

- $\mathfrak{F}_0 = \mathfrak{F}_1$ contains all linear functions, like $\lambda x.x + 3$ or $\lambda x.2x$;
- $\mathfrak{F}_2$ contains all elementary functions, like $\lambda x.2^{2^x}$;

– $\mathfrak{F}_3$ contains all tetration functions, like $\lambda x.\,\underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{x}$.

The hierarchy is strict for $k \geq 1$, i.e., $\mathfrak{F}_k \subsetneq \mathfrak{F}_{k+1}$, because $F_{k+1} \notin \mathfrak{F}_k$. Also, for each $k \geq 1$ and $f \in \mathfrak{F}_k$, there exists $p \geq 1$ such that F_k^p majorises f, i.e., $f(x_1, \ldots, x_n) < F_k^p(\max(x_1, \ldots, x_n))$ for all $x_1, \ldots, x_n$ [16, Theorem 2.10].

The union $\bigcup_k \mathfrak{F}_k$ is the class of all primitive recursive functions, while F_ω defined by $F_\omega(x) = F_x(x)$ is an Ackermann-like non-primitive recursive function; we call Ackermannian such functions that lie in $\mathfrak{F}_\omega \setminus \bigcup_k \mathfrak{F}_k$.

We remark that, following this pattern for successor and limit ordinals, the hierarchy can be continued up to level ω^ω. The union $\bigcup_{\alpha<\omega^\omega} \mathfrak{F}_\alpha$ is the class of all multiply recursive functions, and the non-multiply recursive functions in $\mathfrak{F}_{\omega^\omega}$ have been called 'hyper-Ackermannian'.

2.2 Finite Transducers

We work with normalised transducers with ϵ-transitions, whose input and output alphabets are the same. They are tuples of the form $\langle Q, \Sigma, \delta, I, F\rangle$, where Q is a finite set of states, Σ is a finite alphabet, $\delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times (\Sigma \cup \{\epsilon\}) \times Q$ is a transition relation, and $I, F \subseteq Q$ are sets of initial and final states respectively. We write transitions as $q \xrightarrow{a|a'} q'$, which can be thought of as reading a from the input word (if $a \in \Sigma$) and writing a' to the output word (if $a' \in \Sigma$).

For a transducer $\mathcal{T}$ as above, we say that τ is a *transduction* iff it is a path $q_0 \xrightarrow{a_1|a_1'} q_1 \cdots \xrightarrow{a_n|a_n'} q_n$ where q_0 is initial and q_n is final, and we write $\mathrm{In}(\tau)$ and $\mathrm{Out}(\tau)$ for the words $a_1 \ldots a_n$ and $a_1' \ldots a_n'$ respectively. The relation of $\mathcal{T}$ is then $\mathrm{R}(\mathcal{T}) = \{\langle \mathrm{In}(\tau), \mathrm{Out}(\tau)\rangle \;:\; \tau \text{ is a transduction of } \mathcal{T}\}$. The transducers recognise exactly rational relations between Σ^* and Σ^* (cf. e.g. [20, Chapter IV]).

A *computation* of a transducer $\mathcal{T}$ from a word w_1 is a finite or infinite sequence of words $w_1, w_2, \ldots$ such that $w_1 \;\mathrm{R}(\mathcal{T})\; w_2 \;\mathrm{R}(\mathcal{T}) \cdots$.

If q and q' are states of a transducer $\mathcal{T}$, we write $\mathcal{T}(q, q')$ for the transducer obtained from $\mathcal{T}$ by making q the only initial state and q' the only final state.

2.3 Composing Transducers

We write $⨾$ for relational composition, as well as for its counterpart in terms of transducers. Recalling a standard definition of the latter operation, given two transducers $\mathcal{T}_1 = \langle Q_1, \Sigma, \delta_1, I_1, F_1\rangle$ and $\mathcal{T}_2 = \langle Q_2, \Sigma, \delta_2, I_2, F_2\rangle$, the transition relation of their composition $\mathcal{T}_1 ⨾ \mathcal{T}_2 = \langle Q_1 \times Q_2, \Sigma, \delta, I_1 \times I_2, F_1 \times F_2\rangle$ is defined so that every output of $\mathcal{T}_1$ must be consumed by an input of $\mathcal{T}_2$:

$$\langle q_1, q_2\rangle \xrightarrow{a|a'} \langle q_1', q_2'\rangle \text{ iff } \begin{cases} q_1 \xrightarrow{a|\epsilon} q_1' \text{ and } a' = \epsilon \text{ and } q_2 = q_2', \text{ or} \\ q_1 \xrightarrow{a|a''} q_1' \text{ and } q_2 \xrightarrow{a''|a'} q_2' \text{ for some } a'' \in \Sigma, \text{ or} \\ q_1 = q_1' \text{ and } a = \epsilon \text{ and } q_2 \xrightarrow{\epsilon|a'} q_2'. \end{cases}$$

We then have $\mathrm{R}(\mathcal{T}_1 ⨾ \mathcal{T}_2) = \mathrm{R}(\mathcal{T}_1) ⨾ \mathrm{R}(\mathcal{T}_2)$.

2.4 Downwards Monotone Transducers

Given an alphabet Σ, we write $\sqsubseteq$ for the subword ordering on Σ^*, i.e., $w \sqsubseteq w'$ iff w' can be obtained from w by a number of insertions of letters. The *downward closure* of a subset L of Σ^*, i.e., $\{w \mid \exists w'.\, w \sqsubseteq w' \wedge w' \in L\}$, is denoted by $\downarrow L$.

We say that a relation R on Σ^* is *downwards monotone* iff, whenever $w_1\ R\ w_2$, every replacement of w_1 by a subword w_1' can be matched on the right-hand side of R, i.e., $\forall w_1, w_2, w_1'.\ w_1\ R\ w_2 \wedge w_1' \sqsubseteq w_1 \Rightarrow \exists w_2'.\ w_1'\ R\ w_2' \wedge w_2' \sqsubseteq w_2$. Note that this is the same notion as downward compatibility of R with respect to $\sqsubseteq$ in the theory of well-structured transition systems [9].

A transducer $\mathcal{T}$ is *downwards monotone* iff its relation $\mathrm{R}(\mathcal{T})$ has the property.

Proposition 1. *Composing transducers preserves downward monotonicity.*

We leave open the decidability of whether a given transducer is downward monotone. We note however that this problem is at least as hard as the regular Post embedding problem ($\mathrm{PEP}^{\mathrm{reg}}$) [6], and therefore not multiply recursive [7].

2.5 Downward Rational Termination

The principal problem we study is whether a given downwards monotone transducer terminates from a given word:

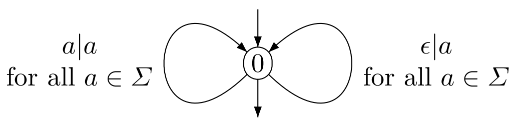
> Given a downwards monotone transducer $\mathcal{T}$ and a word w_1 over its alphabet, is every computation of $\mathcal{T}$ from w_1 finite?

We remark that the standard rational termination problem, i.e., without the assumption of downward monotonicity, is undecidable. Indeed, it is straightforward to compute a transducer that recognises the one-step relation between configurations of a given Turing machine.

Another closely related problem is gainy rational termination (also called increasing rational termination [14]):

> Given a transducer $\mathcal{T}$ and a word w_1 over its alphabet, is every computation of $\mathcal{T}_{\sqsubseteq}$ from w_1 finite?

Here $\mathcal{T}_{\sqsubseteq} = \sqsubseteq \fatsemi \mathcal{T} \fatsemi \sqsubseteq$, where $\sqsubseteq$ on the right-hand side denotes a transducer whose relation is the subword ordering over the alphabet Σ of $\mathcal{T}$:



Thus, $\mathcal{T}_{\sqsubseteq}$ can be thought of as a ‘faulty’ version of $\mathcal{T}$ that may gain arbitrary letters in both input and output words, i.e., suffers from ‘insertion errors’.

By observing that $\mathcal{T}_{\sqsubseteq}$ is downwards monotone for every transducer $\mathcal{T}$, gainy rational termination reduces to downward rational termination. Conversely, for a downwards monotone transducer $\mathcal{T}$, it is easy to see that $\mathcal{T}$ has an infinite computation from w_1 iff the same is true of $\mathcal{T}_{\sqsubseteq}$.

3 Upper Bound

We obtain an Ackermannian upper bound for downward rational termination by proving that, given an instance $\mathcal{T}, w_1$ of the problem, there is an Ackermannianly large positive integer $N(\mathcal{T}, w_1)$ such that if $\mathcal{T}$ terminates from w_1 then all its computations from w_1 have lengths bounded by $N(\mathcal{T}, w_1)$.

At the heart of the proof, there is an analysis of computations of $\mathcal{T}$ from w_1 in terms of how frequently they contain words that belong to certain regular languages. A trivial case is when the regular language consists of all words over the alphabet of $\mathcal{T}$, for which the frequency is 1. More interestingly, our central lemma (Lemma 6) shows that, assuming that the frequency of the language of a regular expression E in a computation of length N is u^{-1} and that N is sufficiently large in terms of u, either some segment of the computation can be pumped to produce an infinite computation, or E can be refined to some E' whose frequency is some smaller u'^{-1}. The notion of refinement of the regular expressions is such that only finitely many successive refinements are ever possible, and so if $\mathcal{T}$ terminates from w_1 then repeated applications of the lemma must stop because N is not sufficiently large. Moreover, the refinements of the regular expressions and the decreases in their frequences observe certain bounds (that depend on $\mathcal{T}$, but not on w_1 or N), which together with the preceding reasoning enables us to obtain a global bound on the lengths of all the computations (provided that $\mathcal{T}$ terminates from w_1).

Before the central lemma, we have two lemmas that are about pumpability of computation segments, and its connection with the regular expressions and their refinements. Leading to the main result, we have another two lemmas, which are concerned with bounding the sequences of regular expressions and frequences that can arise from repeated applications of the central lemma, and consequences of those bounds for the lengths of computations. However, we first introduce the class of regular expressions used and the notion of refinement, as well as a useful class of auxiliary transducers.

3.1 Flat Regular Expressions (FRE)

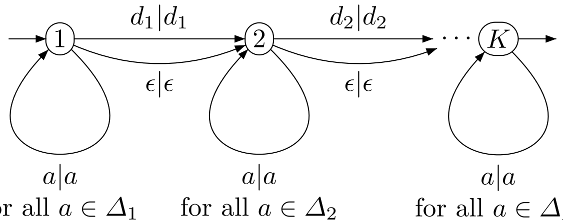
A prominent role in the sequel is played by the following subclass of the simple regular expressions of Abdulla et al. [1]: we say that a regular expression over an alphabet Σ is *flat* iff it is of the form $\Delta_1^* d_1 \Delta_2^* d_2 \cdots \Delta_K^*$ with $K \geq 1$, $\Delta_1, \ldots, \Delta_K \subseteq \Sigma$ and $d_1, \ldots, d_{K-1} \in \Sigma \cup \{\epsilon\}$.

For such a regular expression E, let: the *length* of E be K; the *height* of E be $\max_{i=1}^{K} |\Delta_i|$. If $l \in \mathbb{N}^+$, let us say that E is *l-refined* by E' iff E' can be obtained from E by replacing some Δ_i^* with an FRE $E^\dagger$ over Δ_i such that:

- the length of $E^\dagger$ is at most l, and
- the height of $E^\dagger$ is strictly less than $|\Delta_i|$, i.e., each set in $E^\dagger$ is strictly in Δ_i.

In that case, E' is also an FRE over Σ, of length at most $K + l - 1$. When each set in $E^\dagger$ has size $|\Delta_i| - 1$, we call the refinement *maximal*.

For E still as above, let $\mathcal{I}_E$ denote an identity transducer on the downward closure of the language of E as follows:



Indeed, $\mathrm{R}(\mathcal{I}_E) = \{\langle w, w\rangle \ : \ w \in {\downarrow}\mathrm{L}(E)\}$, so $\mathcal{I}_E$ is downwards monotone.

3.2 Pumpable Transductions

Since finite sequences of consecutive transductions can be seen as single transductions of composite transducers, it suffices to consider pumpability of transductions instead of considering it for computation segments. The notion we define applies to transductions between words in the language of an FRE $E = \Delta_1^* d_1 \cdots \Delta_K^*$, and essentially requires that, for all i, while reading the portion of the input word in Δ_i^*, the transduction visits a part of the transducer that is able to consume any word in Δ_i^*. The composition with the identity transducer is a technical tool to ensure that traversing different paths in the state-transition graph still produces words that conform to E.

Definition 2. *If $\mathcal{T}$ is a downwards monotone transducer and $E = \Delta_1^* d_1 \cdots \Delta_K^*$ is a FRE over an alphabet Σ, and τ is a transduction of composite transducer $\mathcal{T} \fatsemi \mathcal{I}_E$ such that* $\mathrm{In}(\tau) \in \mathrm{L}(E)$*, let us say that τ is* pumpable *iff it can be factored as $s_1 \xrightarrow{\tau_1} s_1' \xrightarrow{d_1|e_1} s_2 \xrightarrow{\tau_2} s_2' \xrightarrow{d_2|e_2} \cdots s_K \xrightarrow{\tau_K} s_K'$, where, for each $i \in \{1, \ldots, K\}$, $\Delta_i^* \subseteq {\downarrow}\mathrm{dom}(\mathrm{R}((\mathcal{T} \fatsemi \mathcal{I}_E)(s_i, s_i')))$.*

Lemma 3. *If $\mathcal{T}$ is downwards monotone, and $\mathcal{T} \fatsemi \mathcal{I}_E$ has a transduction τ such that* $\mathrm{In}(\tau) \in \mathrm{L}(E)$ *and which is pumpable, then $\mathcal{T}$ has an infinite computation from any word in* ${\downarrow}\mathrm{L}(E)$*.*

The following is a 'pumping lemma': roughly, if a transduction from E to E is such that its input word is not in the language of any 'short' refinement of E, then it is pumpable. Here 'short' amounts to a bound which is the product of the length of E and the size of the transducer's state space.

Lemma 4. *Suppose that: $\mathcal{T}$ is a downwards monotone transducer with set of states Q and alphabet Σ; E is a FRE over Σ, of length K; τ is a transduction of $\mathcal{T} \fatsemi \mathcal{I}_E$ such that* $\mathrm{In}(\tau) \in \mathrm{L}(E)$*. Then either τ is pumpable, or* $\mathrm{In}(\tau) \in \mathrm{L}(E')$ *for some $K|Q|$-refinement E' of E.*

3.3 Sword of Hercules

Our central lemma, assuming that γ is a computation of length N from w_1 of a downwards monotone $\mathcal{T}$ which terminates from w_1, can be applied repeatedly to γ to yield some sequence $\langle E_0, u_0 \rangle, \langle E_1, u_1 \rangle, \ldots$ of pairs of FREs and positive integers, as long as N is sufficiently large. For each h, there are at least $\lfloor N/u_h \rfloor$ occurrences in γ of words from the language of E_h. Moreover, each E_{h+1} refines E_h, and the length of E_{h+1} as well as u_{h+1} are bounded by elementary functions of: the number of states of $\mathcal{T}$, the length and height of E_h, and u_h. Recalling the notion of refinement, each application of the lemma can be thought of as a strike of Hercules on the FRE E_h, after which the latter has a Hydra-like response: although some component of the form Δ_h^* is removed from E_h, it is replaced in E_{h+1} by some FRE $E_h^\dagger$. The height of $E_h^\dagger$, however, must be strictly smaller than the size of Δ_h, but the bound on its length grows with every strike.

Definition 5. *For $\alpha \in (0, 1]$, let us say that a regular expression E is α-*frequent *in a sequence of words $w_1, \ldots, w_N$ iff there exists $J \in \{1, \ldots, N\}$ of size $\lfloor N\alpha \rfloor$ such that $w_j \in \mathrm{L}(E)$ for all $j \in J$.*

Lemma 6. *Suppose that $\gamma = w_1, \ldots, w_N$ is a computation of a downwards monotone transducer $\mathcal{T}$ with set of states Q and alphabet Σ, and that $\mathcal{T}$ terminates from w_1. If an FRE E over Σ and $u \in \mathbb{N}^+$ are such that $N \geq 16u^2$ and E is u^{-1}-frequent in γ, then there exists a $K|Q|^{4u}$-refinement E' of E which is ${u'}^{-1}$-frequent in γ, where $K = \mathrm{len}(E)$, $H = \mathrm{hgt}(E)$, and $u' = 16u^2K(H+1)^{2K|Q|^{4u}}$.*

3.4 Slaying the Hydra

The next two lemmas show that every sequence of pairs of FREs and positive integers that can arise from repeated applications of Lemma 6 is finite, i.e., Hercules always defeats the Hydra eventually, and that if $N \geq 16u^2$ for every u in such a sequence and $\mathcal{T}$ terminates from w_1, then $\mathcal{T}$ cannot have a computation from w_1 of length N. Moreover, from the single-step bounds in Lemma 6, we establish a bound for each pair in terms of $|Q|$, $|\Sigma|$ and the distance from the initial pair $\langle \Sigma^*, 1 \rangle$, where Q and Σ are the state space and the alphabet of $\mathcal{T}$.

We first define a directed graph which contains every sequence that Lemma 6 can yield. To show that every path that starts from $\langle \Sigma^*, 1 \rangle$ is finite, we also introduce a measure on FREs E over Σ in terms of $|\Sigma|$-tuples of natural numbers. The latter records, for each $s \in \{1, \ldots, |\Sigma|\}$, how many sets of size s occur in E.

We say that a sequence $y_0, y_1, \ldots$ of tuples in some $\mathbb{N}^k$ is *bad* iff there do not exist $i < j$ such that $y_i \leq y_j$, where $\leq$ is the pointwise ordering. We recall that, by Dickson's Lemma, $\leq$ is a well-quasi ordering on $\mathbb{N}^k$, i.e., there is no infinite bad sequence. Hence, the finiteness of every path from $\langle \Sigma^*, 1 \rangle$ follows once we show that every corresponding sequence of measures in $\mathbb{N}^{|\Sigma|}$ is bad.

Definition 7. *Given a set of states Q and an alphabet Σ, let $\Upsilon_{Q,\Sigma}$ be the graph:*

- *the vertices are pairs $\langle E, u \rangle$ where E is an FRE over Σ and $u \in \mathbb{N}^+$;*

– *there is an edge from* $\langle E, u\rangle$ *to* $\langle E', u'\rangle$ *iff* E' *is a* $K|Q|^{4u}$*-refinement of* E *and* $u' = 16u^2K(H+1)^{2K|Q|^{4u}}$*, where* $K = \mathrm{len}(E)$ *and* $H = \mathrm{hgt}(E)$*.*

Definition 8. *For* $E = \Delta_1^* d_1 \cdots \Delta_K^*$ *an FRE over* Σ *and* $s \in \{0, \ldots, |\Sigma|\}$*, let* $Y_s(E) = |\{i \,:\, i \in \{1, \ldots, K\} \text{ and } |\Delta_i| = s\}|$*.*

Lemma 9. *Suppose that* Q *is a set of states,* Σ *is an alphabet, and* $\langle E_0, u_0\rangle \to \langle E_1, u_1\rangle \to \ldots$ *is a path from* $\langle \Sigma^*, 1\rangle$ *in* $\Upsilon_{Q,\Sigma}$*. Then* $\langle Y_1(E_0), \ldots, Y_{|\Sigma|}(E_0)\rangle$*,* $\langle Y_1(E_1), \ldots, Y_{|\Sigma|}(E_1)\rangle$*, ... is a bad sequence, and letting* $f(u) = 16u^{3+2u^{1+4u}}$*, we have* $\sum_{s=0}^{|\Sigma|} Y_s(E_h), u_h < f^{h+\max(|Q|,|\Sigma|)}(2)$ *for all* h*.*

Lemma 10. *Suppose that:* $\mathcal{T}$ *is a downwards monotone transducer with set of states* Q *and alphabet* Σ*;* $\mathcal{T}$ *terminates from* w_1*;* $N \geq 16u^2$ *for all vertices* $\langle E, u\rangle$ *that are reachable from* $\langle \Sigma^*, 1\rangle$ *in* $\Upsilon_{Q,\Sigma}$*. Then* $\mathcal{T}$ *does not have a computation from* w_1 *of length* N*.*

3.5 Main Result

Given the preceding lemmas, it remains to do two things. The first is to show that, in every graph $\Upsilon_{Q,\Sigma}$, the positive integers in all vertices that are reachable from $\langle \Sigma^*, 1\rangle$ are bounded by an Ackermannian function of $|Q|$ and $|\Sigma|$. Although the vertices and edges of $\Upsilon_{Q,\Sigma}$ can be encoded using the classical Hydra trees of Kirby and Paris [15], we do not require the full generality of the latter, but are able to obtain an Ackermannian bound using Lemma 9 and recent results of Figueira et al. [8] on lengths of bad sequences of tuples of natural numbers.

Writing $N(|Q|, |\Sigma|)$ for the obtained bound, it then remains to argue that a computation of $\mathcal{T}$ from w_1 can be non-deterministically guessed and checked in Ackermannian time or space, but that can be done by a straightforward non-deterministic algorithm that explores the state-transition graph of the iterated transducer $\mathcal{T}^{N(|Q|,|\Sigma|)-1}$ on the fly.

Theorem 11. *Termination for a downwards monotone transducer* $\mathcal{T}$ *with set of states* Q *and alphabet* Σ*, from a word* w_1 *over* Σ*, is decidable by an algorithm whose complexity is bounded by an Ackermannian function. For fixed* $|\Sigma|$*, the bound is in* $\mathfrak{F}_{|\Sigma|+2}$*.*

Proof. From Lemma 9, for every path $\langle E_0, u_0\rangle \to \langle E_1, u_1\rangle \to \ldots$ from vertex $\langle \Sigma^*, 1\rangle$ in $\Upsilon_{Q,\Sigma}$, the sequence $\langle Y_1(E_0), \ldots, Y_{|\Sigma|}(E_0)\rangle$, $\langle Y_1(E_1), \ldots, Y_{|\Sigma|}(E_1)\rangle$, ... in $\mathbb{N}^{|\Sigma|}$ is bad, and for all h, $\max(Y_1(E_h), \ldots, Y_{|\Sigma|}(E_h)) < f^{h+\max(|Q|,|\Sigma|)}(2)$, i.e., in the terminology of Figueira et al. [8], the sequence is $\max(|Q|, |\Sigma|)$-controlled by the function $g(h) = f^h(2)$. Since f is in class $\mathfrak{F}_2$ of the fast growing hierarchy, we have that g belongs to $\mathfrak{F}_3$. Also, g is monotone and satisfies $g(h) \geq \max(1, h)$ for all h, and we can assume that $|\Sigma| \geq 1$. Hence, [8, Proposition 5.2] applies and gives us a function $M_s(t)$ such that M_s is in $\mathfrak{F}_{s+2}$ for each $s \geq 1$, and the length of $\langle Y_1(E_0), \ldots, Y_{|\Sigma|}(E_0)\rangle$, $\langle Y_1(E_1), \ldots, Y_{|\Sigma|}(E_1)\rangle$, ... is at most $M_{|\Sigma|}(\max(|Q|, |\Sigma|))$.

Since the distance of each $\langle E, u \rangle$ reachable from $\langle \Sigma^*, 1 \rangle$ in $\Upsilon_{Q,\Sigma}$ is at most $M_{|\Sigma|}(\max(|Q|, |\Sigma|)) - 1$, we have by Lemma 9 that $N(|Q|, |\Sigma|) \geq 16u^2$, where

$$N(k, s) = 16(g(M_s(\max(k, s)) - 1 + \max(k, s)))^2.$$

Therefore, by Lemma 10, $\mathcal{T}$ terminates from w_1 iff it does not have a computation from w_1 of length $N(|Q|, |\Sigma|)$.

We conclude that termination of $\mathcal{T}$ from w_1 is decidable by guessing and checking an $N(|Q|, |\Sigma|)$-long computation of $\mathcal{T}$ from w_1, which is equivalent to guessing and checking a transduction of the iterated transducer $\mathcal{T}^{N(|Q|,|\Sigma|)-1}$ from w_1. It follows that space $O(N(|Q|, |\Sigma|) \times (\log |Q| + \log |\Sigma|) + \log |w_1|)$ is sufficient for a non-deterministic algorithm.

Recalling that $M_{|\Sigma|}$ is in $\mathfrak{F}_{|\Sigma|+2}$ and that g is in $\mathfrak{F}_3 \subseteq \mathfrak{F}_{|\Sigma|+2}$, we have that $N(|Q|, |\Sigma|)$ as a function of $|Q|$ is also in $\mathfrak{F}_{|\Sigma|+2}$. Therefore, as a function of the combined size of $\mathcal{T}$ and w_1, the non-deterministic space bound is in $\mathfrak{F}_{|\Sigma|+2}$ when $|\Sigma|$ is fixed, and in $\mathfrak{F}_\omega$ in general. Since the classes involved of the fast growing hierarchy are closed under squaring and exponentiation, the same coarse classifications apply to consequent deterministic space and time bounds. □

4 Lower Bound

We use the following variant of the fast growing functions F_k, which give rise to the *Ackermann hierarchy* (cf. e.g. [10]):

$$A_1(x) = 2x \qquad\qquad A_{k+1}(x) = A_k^x(1), \text{ for } k \geq 1.$$

For example, A_2 is exactly exponentiation of 2, and A_3 is exactly tetration of 2. One can check that, for all $k, p \geq 1$, there exists $x_{k,p} \geq 0$ such that, for all $x \geq x_{k,p}$, we have $A_k(x) > F_{k-1}^p(x)$; hence $A_k \notin \mathfrak{F}_{k-1}$ if $k \geq 2$ by [16, Theorem 2.10]. Conversely, $A_k(x) \leq F_k(x)$ for all $k \geq 1$ and $x \geq 0$, so $A_k \in \mathfrak{F}_k$.

To obtain our lower bound result, we provide a construction of 'dependent counter programs' D_1, D_2, ... such that each D_{k+1} is computable from D_k in logarithmic space. For every k, D_k consists of routines for basic counter operations (initialisation, increment, decrement, zero testing, maximum testing), and is dependent in the sense that it may operate on an as yet unspecified counter by calling the latter's operations as subroutines. Moreover, D_k is closely related to the A_k function above: provided C is a counter program that reliably implements a counter bounded by N (in the sense that transducers that correspond to its routines compute correctly, even if insertion errors are possible), then $D_k[C]$ reliably implements a counter bounded by $A_k(N)$. Given a Turing machine of size K, we then use $D_K[C]$ with C reliable up to K to build a transducer that reliably simulates $A_K(K)$ steps of the machine (in the presence of insertion errors), and diverges iff the machine halts.

Theorem 12. *Given a deterministic Turing machine $\mathcal{M}$ of size K, we have that a transducer $\mathcal{T}(\mathcal{M})$ and a word w_1, over an alphabet of linear size, are computable in elementary time, such that $\mathcal{M}$ halts within time $A_K(K)$ iff $\mathcal{T}(\mathcal{M})_{\sqsubseteq}$ does not terminate from w_1.*

5 Safety MTL Satisfiability

We now show that the satisfiability problem for the safety fragment of MTL is inter-reducible with the termination problem for gainy transducers (equivalently, for downwards monotone transducers, cf. Sect. 2.5), thus improving the best known upper and lower bounds for the former. This reduction relies on results in the literature concerning *insertion channel machines (ICMs)*—a model that is very closely related to gainy transducers.

The formulas of MTL are built over a set of atomic events Σ using monotone Boolean connectives and time-constrained versions of the *next* operator $\bigcirc$, *until* operator $\mathcal{U}$, and the *dual until* operator $\widetilde{\mathcal{U}}$:

$$\varphi ::= \top \mid \bot \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid a \mid \bigcirc_I \varphi \mid \varphi_1 \, \mathcal{U}_I \, \varphi_2 \mid \varphi_1 \, \widetilde{\mathcal{U}}_I \, \varphi_2 ,$$

where $a \in \Sigma$ and $I \subseteq \mathbb{R}_{\geq 0}$ is an interval with endpoints in $\mathbb{N} \cup \{\infty\}$.

A *timed word* over alphabet Σ is a pair $\rho = \langle \sigma, \tau \rangle$, where σ is an infinite word over Σ and τ is an infinite sequence of non-negative reals that is strictly increasing and unbounded (i.e., *non-Zeno*). The *satisfiability problem* for MTL asks whether a given formula is satisfied by some timed word. This problem was shown undecidable in [17], motivating the introduction of the sub-logic safety MTL in [18]. Safety MTL is the fragment of MTL obtained by requiring that the interval I in each until operator $\mathcal{U}_I$ have finite length. Thus safety MTL allows bounded eventualities, such as $\Diamond_{(0,1)}\varphi$, but not unbounded eventualites, such as $\Diamond_{(0,\infty)}\varphi$. The satisfiability problem for safety MTL was shown to be decidable in [18] by an argument involving Higman's Lemma. It was later observed that this argument yields an upper bound in level $\mathfrak{F}_{\omega^\omega}$ of the fast-growing hierarchy [21]. A non-elementary lower bound (in $\mathfrak{F}_3$) is given in [4] and an improved lower bound in $\mathfrak{F}_4$ is given in [13].

Theorems 11 and 12 yield upper and lower bounds for safety MTL satisfiability that are both in $\mathfrak{F}_\omega$ through four reductions:

where an ICM is a finite-state automaton acting on an unbounded channel that is subject to insertion errors, and their *fair termination problem* ask whether there is no infinite computation which is *fair*, i.e., in which every message written to the channel is eventually read.

The reductions (i) and (ii) are almost immediate and require only logarithmic space. Details for the reduction (iii), which can also be done in logarithmic space, can be found in [4]. The most complex reduction is (iv): it is doubly exponential, and its details are available in [13, Proposition 5.27], which builds on a translation from MTL to channel machines in [5].

References

1. Abdulla, P.A., Collomb-Annichini, A., Bouajjani, A., Jonsson, B.: Using forward reachability analysis for verification of lossy channel systems. Formal Meth. Sys. Des. 25(1), 39–65 (2004)
2. Abdulla, P.A., Jonsson, B., Nilsson, M., Saksena, M.: A survey of regular model checking. In: CONCUR. LNCS, vol. 3170, pp. 35–48. Springer (2004)
3. Alur, R., Henzinger, T.A.: Back to the future: Towards a theory of timed regular languages. In: FOCS. pp. 177–186. IEEE Comput. Soc. (1992)
4. Bouyer, P., Markey, N., Ouaknine, J., Schnoebelen, P., Worrell, J.: On termination and invariance for faulty channel machines. Formal Asp. Comput. 24(4-6), 595–607 (2012)
5. Bouyer, P., Markey, N., Ouaknine, J., Worrell, J.: The cost of punctuality. In: LICS. pp. 109–120. IEEE Comput. Soc. (2007)
6. Chambart, P., Schnoebelen, P.: Post embedding problem is not primitive recursive, with applications to channel systems. In: FSTTCS. LNCS, vol. 4855, pp. 265–276. Springer (2007)
7. Chambart, P., Schnoebelen, P.: The ordinal recursive complexity of lossy channel systems. In: LICS. pp. 205–216. IEEE Comput. Soc. (2008)
8. Figueira, D., Figueira, S., Schmitz, S., Schnoebelen, P.: Ackermannian and primitive-recursive bounds with Dickson's Lemma. In: LICS. pp. 269–278. IEEE Comput. Soc. (2011), ext. ver. in CoRR, vol. abs/1007.2989
9. Finkel, A., Schnoebelen, P.: Well-structured transition systems everywhere! Theor. Comput. Sci. 256(1-2), 63–92 (2001)
10. Friedman, H.: Long finite sequences. J. Combin. Theor., Ser. A 95(1), 102–144 (2001)
11. Henzinger, T.A.: The Temporal Specification and Verification of Real-Time Systems. Ph.D. thesis, Stanford University (1991), tech. rep. STAN-CS-91-1380
12. Henzinger, T.A.: It's about time: Real-time logics reviewed. In: CONCUR. LNCS, vol. 1466, pp. 439–454. Springer (1998)
13. Jenkins, M.: Synthesis and Alternating Automata over Real Time. D.Phil. thesis, Oxford University (2012)
14. Karandikar, P., Schmitz, S.: The parametric ordinal-recursive complexity of Post embedding problems. In: FoSSaCS. LNCS, vol. 7794, pp. 273–288. Springer (2013)
15. Kirby, L., Paris, J.: Accessible independence results for Peano arithmetic. Bull. London Math. Soc. 14(4), 285–293 (1982)
16. Löb, M., Wainer, S.: Hierarchies of number-theoretic functions. I. Arch. Math. Log. 13(1), 39–51 (1970)
17. Ouaknine, J., Worrell, J.: On metric temporal logic and faulty Turing machines. In: FoSSaCS. LNCS, vol. 3921, pp. 217–230. Springer (2006)
18. Ouaknine, J., Worrell, J.: Safety metric temporal logic is fully decidable. In: TACAS. LNCS, vol. 3920, pp. 411–425. Springer (2006)
19. Ouaknine, J., Worrell, J.: On the decidability of metric temporal logic over finite words. Log. Meth. Comput. Sci. 3(1) (2007)
20. Sakarovitch, J.: Elements of Automata Theory. Camb. Univ. Press (2009)
21. Schmitz, S.: Scientific report - ESF Short Visit to Oxford University (2012)
22. Schnoebelen, P.: Revisiting Ackermann-hardness for lossy counter machines and reset Petri nets. In: MFCS. LNCS, vol. 6281, pp. 616–628. Springer (2010)