



HAL
open science

The Parametric Complexity of Lossy Counter Machines

Sylvain Schmitz

► **To cite this version:**

Sylvain Schmitz. The Parametric Complexity of Lossy Counter Machines. ICALP 2019, Jul 2019, Patras, Greece. pp.129:1–129:15, 10.4230/LIPIcs.ICALP.2019.129 . hal-02020728v2

HAL Id: hal-02020728

<https://hal.science/hal-02020728v2>

Submitted on 21 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License

The Parametric Complexity of Lossy Counter Machines

Sylvain Schmitz 

LSV, ENS Paris Saclay & CNRS, Université Paris-Saclay, France

IUF, France

sylvain.schmitz@lsv.fr

Abstract

The reachability problem in lossy counter machines is the best-known ACKERMANN-complete problem and has been used to establish most of the ACKERMANN-hardness statements in the literature. This hides however a complexity gap when the number of counters is fixed. We close this gap and prove F_d -completeness for machines with d counters, which provides the first known uncontrived problems complete for the fast-growing complexity classes at levels $3 < d < \omega$. We develop for this an approach through antichain factorisations of bad sequences and analysing the length of controlled antichains.

2012 ACM Subject Classification Theory of computation \rightarrow Computational complexity and cryptography

Keywords and phrases Counter machine, well-structured system, well-quasi-order, antichain, fast-growing complexity

Funding ANR-14-CE28-0005 PRODAQ, ANR-17-CE40-0028 BRAVAS

Acknowledgements I thank Philippe Schnoebelen for his comments on a preliminary draft.

1 Introduction

Mayr and Meyer exhibited in 1981 ‘the first uncontrived decidable problems which are not primitive-recursive,’ namely the finite containment and equality problems in Petri nets [32]. McAloon [34], Clote [8], and Howell and Yen [24] subsequently proved Ackermannian upper bounds for these two problems, essentially matching Mayr and Meyer’s lower bound.

Such an astronomical complexity could have been an isolated phenomenon with only a few examples related to the original problems, but uncontrived problems with a similar complexity actually occur in logic (e.g., relevance logic [48], data logics [11, 16], interval temporal logic [36], linear logic [29], metric temporal logic [28]), verification (e.g., counter machines [44, 23], fragments of the π -calculus [4], broadcast protocols [43], rewriting systems [21], register automata [11, 18]), and games (e.g., partial observation energy games [37], bisimulation games on pushdown automata [25]), and even higher complexities also occur naturally [30, 6, 38, 20, 19, 10]; see [39, Sec. 6] for an overview.

This abundance of results is largely thanks to a framework [42, 43, 40] that comprises:

- The definition of an ordinal-indexed hierarchy $(F_\alpha)_\alpha$ of *fast-growing complexity classes*, along with assorted notions of reductions and completeness suitable to work with such high complexities [39]. The previous decision problems are complete for $\text{ACKERMANN} = F_\omega$ under primitive-recursive reductions; F_ω is the lowest non primitive-recursive class in the hierarchy, where $\text{TOWER} = F_3$ corresponds to problems solvable in time bounded by a tower of exponentials and where each F_k for a finite k is primitive-recursive.
- The identification of *master* decision problems, which allow to establish completeness more easily than from first principles. For instance, reachability in lossy counter machines [48, 44, 46] plays a similar role for ACKERMANN as, e.g., 3SAT for NP or QBF for PSPACE, and has been used to derive most of the known ACKERMANN-hardness results [4, 11, 16, 36, 28, 23, 21, 18, 25].



- Lower bound techniques for establishing the complexity of such master problems: this typically relies on implementing *weak computers* for *Hardy functions* and their inverses in the formalism at hand, allowing to build a large but bounded working space on which a Turing or a Minsky machine can then be simulated [48, 44, 46, 6, 38, 20, 19].
- Upper bound techniques relying on combinatorial statements, called *length function theorems*, on the length of *controlled bad sequences* over well-quasi-orders, which are used to prove the termination of the decision procedures [34, 8, 50, 7, 15, 41, 3, 38].

From an algorithmic perspective, these results are negative and one could qualify such problems as merely ‘not undecidable.’ What we gain however are insights into the computational power of the models, allowing to compare them and to identify the main sources of complexity—e.g., in lossy counter machines, the key parameter is the number of counters. Furthermore, from a modelling perspective, a formalism with a tremendous computational power that nevertheless falls short of Turing completeness can be quite satisfactory.

Contributions. In this paper, we revisit the proof of the best-known result in this area, namely the ACKERMANN-completeness of reachability in lossy counter machines (LCMs). Those are simply multi-counter Minsky machines with a *lossy semantics* that allows the counters to decrease in an uncontrollable manner during executions; see Section 2.

The gap in the current state of knowledge appears when one fixes the key complexity parameter, i.e., the number d of counters. Indeed, the best known lower bound for LCM reachability is F_d -hardness when $d \geq 3$ [40, Thm. 4.9], but the best known upper bound is F_{d+1} [15, 42, 3]. This complexity gap reveals a serious shortcoming of the framework advertised earlier in this introduction, and also impacts the complexity of many problems shown hard through a reduction from LCM reachability.

Our first main contribution in Proposition 10 is an F_d upper bound, which together with the lower bound from [40, Thm. 4.9] entails the following completeness result.

► **Theorem 1.** *LCM Reachability is F_ω -complete, and F_d -complete if the number $d \geq 3$ of counters is fixed.*

Note that this provides an uncontrived decision problem for every class F_k with $3 \leq k \leq \omega$, whereas no natural F_k -complete problems were previously known for the intermediate primitive-recursive levels strictly between TOWER and ACKERMANN, i.e., for $3 < k < \omega$.

As we recall in Section 3, reachability in lossy counter machines can be solved using the generic *backward coverability algorithm* for *well-structured systems* [1, 17]. As usual, we derive our complexity upper bound by bounding the length of the bad sequences that underlie the termination argument for this algorithm. The main obstacle here is that the length function theorems in [15, 42, 3]—i.e., the bounds on the length of controlled bad sequences over \mathbb{N}^d —are essentially optimal and only yield an F_{d+1} complexity upper bound.

We circumvent this using a new approach in Section 4. We restrict our attention to *strongly controlled* bad sequences rather than the more general *amortised controlled* ones (see Section 4.1), which in turn allows us to work on the *antichain factorisations* of bad sequences (see Section 4.2). This entails that, in order to bound the length of strongly controlled bad sequences, it suffices to bound the length of strongly controlled antichains. This is tackled in Section 5, where we prove a *width function theorem* on the length of controlled antichains over \mathbb{N}^d ; to the best of our knowledge, this is the first statement of this kind specific to antichains rather than bad sequences. We wrap up with the proof of Proposition 10 in Section 6.

The developments of Sections 4 and 5 form our second main contribution. They are of wide interest beyond lossy counter machines, as they can be applied whenever the termination of an algorithm relies on \mathbb{N}^d having finite (controlled) bad sequences or antichains.

2 Lossy Counter Machines

Syntax. A *lossy counter machine* (LCM) [33] is syntactically identical to a Minsky machine $M = (Q, \mathcal{C}, \delta)$, where the transitions in $\delta \subseteq Q \times \mathcal{C} \times \{=0?, ++, --\} \times Q$ operate on a finite set Q of control locations and a finite set \mathcal{C} of counters through *zero-tests* $c=0?$, *increments* $c++$ and *decrements* $c--$.

Operational Semantics. The semantics of an LCM differ from the usual, ‘reliable’ semantics of a counter machine in that the counter values can decrease in an uncontrolled manner at any point of the execution. Formally, a *configuration* $q(\mathbf{v})$ associates a control location q in Q with a counter valuation \mathbf{v} in $\mathbb{N}^{\mathcal{C}}$, i.e. counter values can never go negative. The set of configurations $Q \times \mathbb{N}^{\mathcal{C}}$ is ordered by the product ordering: $q(\mathbf{v}) \leq q'(\mathbf{v}')$ if $q = q'$ and $\mathbf{v}(c) \leq \mathbf{v}'(c)$ for all $c \in \mathcal{C}$.

A transition of the form $(q, c, \text{op}, q') \in \delta$ defines a set of *reliable computation steps* $q(\mathbf{v}) \rightarrow q'(\mathbf{v}')$, where $\mathbf{v}(c') = \mathbf{v}'(c')$ for all $c \neq c'$ in \mathcal{C} and

- if $\text{op} = =0?$, then $\mathbf{v}(c) = \mathbf{v}'(c) = 0$,
- if $\text{op} = ++$, then $\mathbf{v}(c) + 1 = \mathbf{v}'(c)$, and
- if $\text{op} = --$, then $\mathbf{v}(c) = \mathbf{v}'(c) + 1$.

A *lossy computation step* is then defined by allowing counter values to decrease arbitrarily between reliable steps: $q(\mathbf{v}) \rightarrow_{\ell} q'(\mathbf{v}')$ if there exist $\mathbf{w} \leq \mathbf{v}$ and $\mathbf{w}' \geq \mathbf{v}'$ such that $q(\mathbf{w}) \rightarrow q'(\mathbf{w}')$. We write as usual \rightarrow_{ℓ}^* for the transitive reflexive closure of \rightarrow_{ℓ} .

Reachability. The decision problem we tackle in this paper is the following.

► **Problem** (LCM Reachability).

instance A lossy counter machine (Q, \mathcal{C}, δ) and two configurations $q_0(\mathbf{v}_0)$ and $q_f(\mathbf{v}_f)$.

question Is $q_f(\mathbf{v}_f)$ reachable from $q_0(\mathbf{v}_0)$, i.e., does $q_0(\mathbf{v}_0) \rightarrow_{\ell}^* q_f(\mathbf{v}_f)$?

Note that, due to the lossy semantics, this is equivalent to the *coverability* problem, which asks instead whether there exists $\mathbf{v} \geq \mathbf{v}_f$ such that $q_0(\mathbf{v}_0) \rightarrow_{\ell}^* q_f(\mathbf{v})$. Indeed, such a \mathbf{v} exists if and only if $q_0(\mathbf{v}_0) \geq q_f(\mathbf{v}_f)$ or $q_0(\mathbf{v}_0) \rightarrow_{\ell}^* q_f(\mathbf{v}_f)$.

While many problems are undecidable in LCMs [33, 45], these systems are in fact *well-structured* in the sense of [1, 17], which means that their coverability problem is decidable, as further discussed in Section 3. The ACKERMANN-hardness of reachability was first shown by Schnoebelen [44] in 2002,¹ while an ACKERMANN upper bound follows from the length function theorems for Dickson’s Lemma [34, 8, 15, 42, 3]. Note that LCM reachability is equivalent to reachability in counter machines with *incrementing errors* [11] and to coverability in *reset counter machines* [46, Sec. 6], and this also holds if we fix the number of counters.

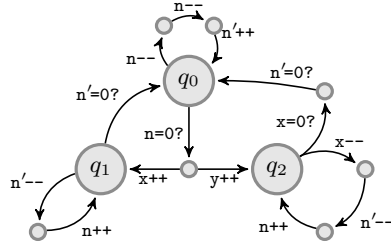
► **Example 2** (Weak \log^*). Figure 1a shows the pseudo-code for a program computing $\log^* n$ in y , i.e., the inverse of a tower of exponentials of height y . Indeed, when started with $x = y = 0$, each time the program visits the location q_2 , it has performed $n := \log n$ by

¹ Urquhart [48] showed independently in 1999 and using a similar approach the same result for the closely related model of alternating expansive vector addition systems.

```

q0: while n mod 2 = 0 do
    n := n/2;
    if x = n then
q2:   y := y+1; x := 0;
    else
q1:   x := x+1;
    done

```



(a) An integer program.

(b) A counter machine implementing Figure 1a.

■ **Figure 1** The counter machine for Example 2.

looping over q_1 . Figure 1b displays a counter machine with counters $\mathcal{C} \stackrel{\text{def}}{=} \{x, y, n, n'\}$, where n' is an auxiliary counter used to perform division by two. Note that the deterministic choice **if** $x = n$ is replaced by a non-deterministic choice between going to q_1 or q_2 , but q_2 checks that this choice was correct by decrementing x and n' in lockstep and checking that they are both equal to zero before returning to q_0 . If started in a configuration $q_0(0, 0, n, 0)$ and using reliable semantics, this machine reaches $q_0(0, y, 0, 0)$ exactly for $y = \log^* n$. With lossy semantics, it might also reach smaller values of y .

3 Well Structured Systems

Well-structured transition systems (WSTS) [1, 17] form a family of computational models where the (usually infinite) set of configurations is equipped with a well-quasi-ordering (see Section 3.1) that is ‘compatible’ with the computation steps (see Section 3.2). The existence of this well-quasi-ordering allows for the decidability of some important behavioural properties like termination (from a given initial configuration) or coverability, see Section 3.3.

3.1 Well-Quasi-Orders

A *quasi-order* (qo) is a pair (X, \leq) where $\leq \subseteq X \times X$ is transitive and reflexive; we write $x < y$ for the associated strict ordering, when $x \leq y$ and $y \not\leq x$, $x \perp y$ for incomparable elements, when $x \not\leq y$ and $y \not\leq x$, and $x \equiv y$ for equivalent elements, when $x \leq y$ and $y \leq x$. The *upward-closure* $\uparrow Y$ of some $Y \subseteq X$ is defined as $\uparrow Y \stackrel{\text{def}}{=} \{x \in X \mid \exists y \in Y. x \geq y\}$; we write $\uparrow x$ instead of $\uparrow\{x\}$ for singletons and say that a set $U \subseteq X$ is *upwards-closed* when $U = \uparrow U$. We call a finite or infinite sequence x_0, x_1, x_2, \dots over X *bad* if for all indices $i < j$, $x_i \not\leq x_j$; if $x_i \perp x_j$ for all $i < j$, then x_0, x_1, x_2, \dots is an *antichain*.

A *well-quasi-order* (wqo) [22, 27] is a qo (X, \leq) where bad sequences are finite. Equivalently, (X, \leq) is a wqo if and only if it is both *well-founded*, i.e., there does not exist any infinite decreasing sequences $x_0 > x_1 > x_2 > \dots$ of elements in X , and has the *finite antichain condition*, i.e., there are no infinite antichains. Still equivalently, (X, \leq) is a wqo if and only if it has the *ascending chain condition*: any increasing sequence $U_0 \subseteq U_1 \subseteq U_2 \subseteq \dots$ of upwards-closed subsets of X eventually stabilises, i.e., $\bigcup_{i \in \mathbb{N}} U_i = U_k = U_{k+1} = U_{k+2} = \dots$ for some k . Still equivalently, (X, \leq) is a wqo if and only if it has the *finite basis property*: any non-empty subset contains at least one, and at most finitely many minimal elements (up to equivalence); thus if $U \subseteq X$ is upwards-closed, then $\min U$ is finite and $U = \uparrow(\min U)$.

For a basic example, consider any finite set Q along with the equality relation, which is a wqo $(Q, =)$ by the pigeonhole principle. Any well-order is a wqo, thus the set of natural numbers and any of its initial segments $[k] \stackrel{\text{def}}{=} \{0, \dots, k-1\}$ along with their natural ordering

are also wqos. More examples can be constructed using algebraic operations: for instance, if (X_0, \leq_{X_0}) and (X_1, \leq_{X_1}) are wqos, then so are:

- their disjoint sum $(X_0 \sqcup X_1, \leq)$ where $X_0 \sqcup X_1 \stackrel{\text{def}}{=} \{(x, 0) \mid x \in X_0\} \cup \{(x, 1) \mid x \in X_1\}$ and $(x, i) \leq (y, j)$ if $i = j$ and $x \leq_{X_i} y$;
- their Cartesian product $(X_0 \times X_1, \leq)$ where $(x_0, x_1) \leq (y_0, y_1)$ if $x_i \leq_{X_i} y_i$ for all $0 \leq i \leq 1$; in the case of (\mathbb{N}^d, \leq) , this result is also known as Dickson's Lemma [12].

Note that the set of configurations $(Q \times \mathbb{N}^c, \leq)$ of an LCM is a wqo for the product ordering.

3.2 Compatibility

An *ordered transition system* $\mathcal{S} = (S, \rightarrow, \leq)$ combines a set S of configurations with a transition relation $\rightarrow \subseteq S \times S$ and a quasi-ordering \leq of its configurations. An ordered transition system $\mathcal{S} = (S, \rightarrow, \leq)$ is *well-structured* if (S, \leq) is a wqo and

$$\forall s_1, s_2, t_1 \in S, (s_1 \rightarrow s_2 \text{ and } s_1 \leq t_1) \text{ implies } \exists t_2 \in S, (t_1 \rightarrow t_2 \text{ and } s_2 \leq t_2). \quad (1)$$

This property is also called *compatibility* (of the ordering with the transitions). Formally, it just means that \leq is a *simulation* relation for (S, \rightarrow) , in precisely the classical sense of [35]. The point of (1) is to ensure that a larger configuration can do at least as much as a smaller configuration. For instance, lossy steps in a LCM are visibly compatible with \leq according to (1), and thus the transition system $(Q \times \mathbb{N}^c, \rightarrow_\ell, \leq)$ defined by the lossy operational semantics of a LCM is a WSTS.

3.3 Coverability

We focus here on the *coverability problem*: given a WSTS (S, \rightarrow, \leq) and two configurations $s, t \in S$, does s *cover* t , i.e., does there exist $t' \geq t$ such that $s \rightarrow^* t'$? The decidability of this problem uses a set-saturation method first introduced by Arnold and Latteux [5] for reset Petri nets, but the algorithm was independently rediscovered by Abdulla et al. [2] for lossy channel systems and its generic formulation was popularised in the surveys [1, 17].

Backward Coverability. The algorithm computes $\text{Pre}_\exists^*(\uparrow t) \stackrel{\text{def}}{=} \{s' \in S \mid \exists t' \geq t, s' \rightarrow^* t'\}$, i.e., the set of configurations that cover t ; it only remains to check whether $s \in \text{Pre}_\exists^*(\uparrow t)$ in order to answer the coverability instance. More precisely, for a set of configurations $U \subseteq S$, let us define its (existential) *predecessor set* as $\text{Pre}_\exists(U) \stackrel{\text{def}}{=} \{s \in S \mid \exists s' \in U, s \rightarrow s'\}$. The algorithm computes the limit of the sequence $U_0 \subseteq U_1 \subseteq \dots$ defined by

$$U_0 \stackrel{\text{def}}{=} \uparrow t, \quad U_{n+1} \stackrel{\text{def}}{=} U_n \cup \text{Pre}_\exists(U_n). \quad (2)$$

Note that for all n , $U_n = \{s' \in S \mid \exists t' \geq t, s' \rightarrow^{\leq n} t'\}$ is the set of configurations that cover t in at most n steps, and that we can stop this computation as soon as $U_{n+1} \subseteq U_n$.

There is no reason for the chain defined by (2) to stabilise in general ordered transition systems, but it does in the case of a WSTS. Indeed, $\text{Pre}_\exists(U)$ is upwards-closed whenever $U \subseteq S$ is upwards-closed, thus the sequence defined by (2) stabilises to $\bigcup_{i \in \mathbb{N}} U_i = \text{Pre}_\exists^*(\uparrow t)$ after a finite amount of time thanks to the ascending chain condition. Moreover, the finite basis property ensures that all the sets U_i can be finitely represented using their minimal elements, and the union or inclusion of two upwards-closed sets can be computed on this representation. The last ingredients are two effectiveness assumptions:

- (S, \leq) should be effective, meaning that S is recursive and the ordering \leq is decidable,

- there exists an algorithm returning the set of minimal predecessors $\min \text{Pre}_\exists(\uparrow s')$ of any given configuration s' ; this is known as the *effective pred-basis* assumption.

These two assumptions hold in LCMs: $(Q \times \mathbb{N}^c, \leq)$ is certainly effective, and the minimal predecessors of a configuration $q'(\mathbf{v}')$ can be computed by

$$\min \text{Pre}_\exists(\uparrow q'(\mathbf{v}')) = \min\{q(\text{pre}_{c_{\text{op}}}(\mathbf{v}')) \mid (q, c, \text{op}, q') \in \delta \text{ and } \mathbf{v}'(c) = 0 \text{ if } \text{op} = \text{=0?}\} \quad (3)$$

where $\text{pre}_{c_{\text{op}}}(\mathbf{v}')$ is a vector in \mathbb{N}^c defined by $\text{pre}_{c_{\text{op}}}(\mathbf{v}')(c') \stackrel{\text{def}}{=} \mathbf{v}'(c')$ for all $c' \neq c$ in \mathbb{C} and

$$\text{pre}_{c_{\text{=0?}}}(\mathbf{v}')(c) \stackrel{\text{def}}{=} 0, \quad \text{pre}_{c_{\text{++}}}(\mathbf{v}')(c) \stackrel{\text{def}}{=} \max\{0, \mathbf{v}'(c) - 1\}, \quad \text{pre}_{c_{\text{--}}}(\mathbf{v}')(c) \stackrel{\text{def}}{=} \mathbf{v}'(c) + 1. \quad (4)$$

Let us consider again the lossy counter machine of Example 2. Observe that the backward coverability algorithm with target configuration $q_0(0, y, 0, 0)$ terminates with $q_0(0, 0, n, 0) \in \min \text{Pre}_\exists^*(\uparrow q_0(0, y, 0, 0))$ for $n = \text{tower}(y)$, and thus requires a non-elementary number of steps.

Coverability Pseudo-Witnesses. Let us reformulate the termination argument of the backward coverability algorithm in terms of bad sequences. We can extract a sequence of elements t_0, t_1, \dots from the ascending sequence $U_0 \subsetneq U_1 \subsetneq \dots$ defined by (2) before saturation: $t_0 \stackrel{\text{def}}{=} t$ and $t_{i+1} \in U_{i+1} \setminus U_i$ for all i . Note that if $i < j$, then $t_j \in U_j \setminus U_i$ and therefore $t_i \not\leq t_j$: the sequence t_0, t_1, \dots is bad and therefore finite. In fact, we can even pick t_{i+1} at each step among the minimal elements of $\text{Pre}_\exists(\uparrow t_i)$; we call such a bad sequence t_0, t_1, \dots, t_n with

$$t_0 \stackrel{\text{def}}{=} t, \quad t_{i+1} \in \min \text{Pre}_\exists(\uparrow t_i) \setminus U_i. \quad (5)$$

a *pseudo-witness* of the coverability of t . The maximal length of pseudo-witnesses is therefore equal to the number of steps of the backward coverability algorithm, and this is what we will bound in the upcoming Sections 4 and 5.

4 Controlled Bad Sequences and Antichains

As we have just discussed, the running time of the backward coverability algorithm is essentially bounded by the length of the bad sequences constructed by its termination argument. Though bad sequences over a wqo are finite, we cannot bound their lengths in general; e.g., $(0, n+1), (0, n), \dots, (0, 0)$ and $(1, 0), (0, n), (0, n-1), \dots, (0, 1), (0, 0)$ are bad sequences over \mathbb{N}^2 of length $n+2$ for all n . Nevertheless, a bad sequence produced by an algorithm like the backward coverability algorithm of Section 3.3 is not arbitrary, because its elements are determined by the algorithm's input and the complexity of its operations. We capture this intuition formally through *controlled sequences*.

4.1 Controlling Sequences

Norms. Given a wqo (X, \leq_X) , we posit a *norm* function $|\cdot|_X: X \rightarrow \mathbb{N}$; if $x \leq_X x'$ implies $|x|_X \leq |x'|_X$, we call this norm *monotone*. In order to be able to derive combinatorial statements, we require $X_{\leq n} \stackrel{\text{def}}{=} \{x \in X \mid |x|_X \leq n\}$ to be finite for every n ; we call the resulting structure $(X, \leq_X, |\cdot|_X)$ a *normed wqo* (nqo).

We will use the following monotone norms on the wqos we defined in Section 3.1: over a finite Q , all the elements have the same norm 0; over \mathbb{N} or $[d]$, n has norm $|n|_{\mathbb{N}} = |n|_{[d]} = n$; over disjoint sums $X_0 \sqcup X_1$, (x, i) uses the norm $|x|_{X_i}$ of its underlying set; finally, over Cartesian products $X \times Y$, (x, y) uses the infinite norm $\max(|x|_X, |y|_Y)$.

Controls. Let $n_0 \in \mathbb{N}$ and let $g: \mathbb{N} \rightarrow \mathbb{N}$ be a monotone and inflationary function, i.e., for all $x \leq x'$, $g(x) \leq g(x')$ and $x \leq g(x)$. We say that a sequence x_0, x_1, x_2, \dots of elements in X is *amortised* (g, n_0)-*controlled* if

$$\forall i. |x_i|_X \leq g^i(n_0), \quad (6)$$

where g^i denotes the i th iterate of g . We say that it is *strongly* (g, n_0)-*controlled* if

$$|x_0|_X \leq n_0 \quad \text{and} \quad \forall i. |x_{i+1}|_X \leq g(|x_i|_X). \quad (7)$$

By definition, a strongly controlled sequence is also amortised controlled: $|x_0|_X \leq g^0(n_0) = n_0$, which prompts the name of *initial norm* for n_0 , and amortised steps cannot grow faster than g the *control function*. A useful property of strongly (g, n_0)-controlled sequences x_0, x_1, \dots is that, for all i , the suffix x_i, x_{i+1}, \dots is also ($g, |x_i|_X$)-controlled; this might not hold in an amortised controlled sequence.

Previous works like [15, 42, 3] focused on the more general amortised controlled sequences, but strong controlled ones are actually more relevant in practice. For instance, in the case of LCM coverability, the computation of minimal predecessors in (3–4) shows that the pseudo-witnesses from (5) of the coverability of a target configuration $q_f(\mathbf{v}_f)$ are strongly ($H, |\mathbf{v}_f|$)-controlled by the initial norm $|\mathbf{v}_f| = \max_{c \in \mathbb{C}} \mathbf{v}_f(c)$ and the control function $H(x) \stackrel{\text{def}}{=} x + 1$.

Length, Norm, and Width Functions. The point of controlled sequences is that their length can be bounded. Consider for this the tree obtained by sharing the common prefixes of all the strongly (g, n_0)-controlled bad sequences over a normed wqo $(X, \leq_X, |\cdot|_X)$.

- This tree is finitely branching by (7)—its branching degree is bounded by the cardinal of $X_{\leq g^i(n_0)}$ for a node at depth i —, and
- it has no infinite branches since bad sequences over (X, \leq_X) are finite.

By König’s Lemma, this tree of bad sequences is therefore finite, of some height $L_{g,X}^s(n_0)$ representing the length of the maximal strongly (g, n_0)-controlled bad sequence(s) over X , and we also let $N_{g,X}^s(n_0)$ bound the norms encountered along such sequences; note that $N_{g,X}^s(n_0) \leq g^{L_{g,X}^s(n_0)}(n_0)$ since g is monotone inflationary. Similarly, there exists a maximal length, denoted by $W_{g,X}^a(n_0)$ (resp. $W_{g,X}^s(n_0)$), for amortised (resp. strongly) (g, n_0)-controlled antichains over X . We call $L_{g,X}^s$, $N_{g,X}^s$, and $W_{g,X}^s$ the *strong length*, *strong norm*, and *strong width* functions respectively, and $W_{g,X}^a$ the *width* function. By definition, a strongly controlled antichain is amortised controlled, i.e.,

$$W_{g,X}^s(n_0) \leq W_{g,X}^a(n_0), \quad (8)$$

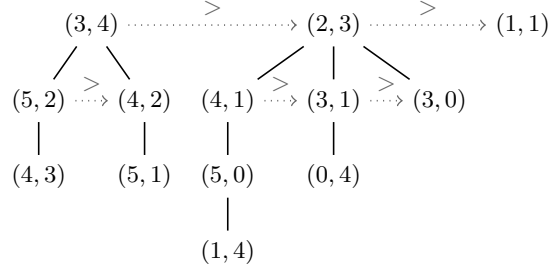
and the length of a bad sequence where all the elements are of norm at most $N = N_{g,X}^s(n_0)$ is bounded by the cardinal of $X_{\leq N}$, i.e.,

$$L_{g,X}^s(n_0) \leq |X_{\leq N_{g,X}^s(n_0)}|. \quad (9)$$

Observe that $L_{g,X}^s(|t|_X)$ bounds the number of steps required by the backward coverability algorithm for a WSTS over $(X, \leq_X, |\cdot|_X)$ with target configuration t where $s' \in \min \text{Pre}(\uparrow t')$ implies $|s'|_X \leq g(|t'|_X)$. In the case of LCMs, we are therefore interested in $L_{H, Q \times N^c}^s(|\mathbf{v}_f|)$.

4.2 Antichain Factorisations

Let $(X, \leq_X, |\cdot|_X)$ be a nwqo where $|\cdot|_X$ is monotone—i.e., $x \leq x'$ implies $|x|_X \leq |x'|_X$ —, and let $x_0, x_1, \dots, x_{\ell-1} \in X^*$ be a strongly (g, n_0)-controlled bad sequence over $(X, \leq_X, |\cdot|_X)$.



■ **Figure 2** The antichain factorisation of the strongly $(H, 4)$ -controlled bad sequence $(3, 4) (5, 2) (4, 3) (4, 2) (5, 1) (2, 3) (4, 1) (5, 0) (1, 4) (3, 1) (0, 4) (3, 0) (1, 1)$ over \mathbb{N}^2 for $H(x) \stackrel{\text{def}}{=} x + 1$.

Informally, the *antichain factorisation* of $x_0, x_1, \dots, x_{\ell-1}$ is an ordered forest A where all the branches are strongly (g, n_0) -controlled antichains, siblings are ordered left-to-right by the strict ordering $>_X$, and such that the pre-order traversal of A yields back the bad sequence. Consider for instance the example of Figure 2: this bad sequence has length 13, thus the norm of its elements is at most $H^{12}(4) = 16$, but because the height of its antichain factorisation is 4, we can actually bound the norm by $H^3(4) = 7$.

We can compute this factorisation from any strongly (g, n_0) -controlled bad sequence. Formally, $A \subseteq X^*$ is a prefix-closed finite set of antichains with the prefix ordering as vertical ordering. Two antichains u and v in A are *siblings* if $u = w \cdot x$ and $v = w \cdot y$ for some $w \in X^*$ and $x, y \in X$, and we order such siblings by letting $u >_X v$ if $x >_X y$. Given $x_0, \dots, x_{\ell-1}$, we let $A \stackrel{\text{def}}{=} \text{Fact}(x_0, 1)$ where

$$\text{Fact}(y_0 \cdots y_m, i) \stackrel{\text{def}}{=} \{y_0 \cdots y_m\} \cup \begin{cases} \emptyset & \text{if } i = \ell, \\ \text{Fact}(y_0 \cdots y_m x_i, i + 1) & \text{if } \forall j. y_j \not>_X x_i, \\ \text{Fact}(y_0 \cdots y_{k-1} x_i, i + 1) & \text{if } k = \min\{j \mid y_j >_X x_i\}. \end{cases} \quad (10)$$

This corresponds to scanning the elements x_i of the bad sequence from left to right while building the current ‘rightmost branch’ $y_0 \cdots y_m \in A$, which is (by induction on i) a strongly (g, n_0) -controlled antichain and a scattered subword of $x_0 \cdots x_{i-1}$ (thus $y_j \not>_X x_i$ for all $0 \leq j \leq m$) such that $y_m = x_{i-1}$. If $y_j \not>_X x_i$ for all $0 \leq j \leq m$, then $y_0 \cdots y_m x_i$ is a (g, n_0) -controlled antichain and a scattered subword of $x_0 \cdots x_i$. If otherwise $y_j >_X x_i$ for some y_j , we let k be the minimal such j and we start a new rightmost branch with x_i out of y_{k-1} (x_i is possibly a new root if $k = 0$); crucially, because $|\cdot|_X$ is monotone and $x_i <_X y_k$, we have $|x_i|_X \leq |y_k|_X \leq g(|y_{k-1}|_X)$ (or $|x_i|_X \leq |y_k|_X \leq n_0$ at the root), thus $y_0 \cdots y_{k-1} x_i$ is again a strongly (g, n_0) -controlled antichain and a scattered subword of $x_0 \cdots x_i$.

We deduce a bound on the strong norm function in terms of the strong width function.

► **Lemma 3 (Antichain Factorisation).** *Let $(X, \leq_X, |\cdot|_X)$ be a normed wqo with $|\cdot|_X$ monotone, n_0 in \mathbb{N} , and $g: \mathbb{N} \rightarrow \mathbb{N}$ monotone inflationary. Then $N_{g,X}^s(n_0) \leq g^{W_{g,X}^s(n_0)}(n_0)$.*

Lemma 3 combined with (8) shows that the strong norm function $N_{g,X}^s$ can be bounded in terms of the width function $W_{g,X}^s$. By (9), this will also yield a bound on the strong length function $L_{g,X}^s$. We focus therefore on the width function in the upcoming Section 5.

5 Width Function Theorem

As seen in Section 4, by suitably controlling how large the elements can grow in antichains, we can derive upper bounds on the time and space required by the backward coverability

algorithm of Section 3. We prove in this section a *width function theorem*, a combinatorial statement on the length of amortised controlled antichains over tuples of natural numbers, which will allow to derive a complexity upper bound for reachability in lossy counter machines.

The high complexities at play here require the use of ordinal-indexed *subrecursive functions* in order to denote non-elementary growths. We first recall the definitions of two families of such functions in Section 5.1. We then prove in Section 5.2 a bound on the width function W_{g, \mathbb{N}^d}^a using the framework of [41, 42].

5.1 Subrecursive Hierarchies

We employ notations compatible with those of Schwichtenberg and Wainer [47, Chap. 4], and refer the interested reader to their monograph and [42] for proofs and additional material.

Fundamental Sequences and Predecessors. Consider an ordinal term α in Cantor normal form $\omega^{\alpha_1} + \dots + \omega^{\alpha_p}$ where $\alpha_1 \geq \dots \geq \alpha_p$. Then $\alpha = 0$ if and only if $p = 0$, an ordinal α of the form $\alpha' + 1$ (i.e. with $p > 0$ and $\alpha_p = 0$) is called a *successor* ordinal, and otherwise if $\alpha_p > 0$ it is called a *limit* ordinal, and can be written as $\gamma + \omega^\beta$ by setting $\gamma = \omega^{\alpha_1} + \dots + \omega^{\alpha_{p-1}}$ and $\beta = \alpha_p$. We write ‘ λ ’ to denote a limit ordinal.

A *fundamental sequence* for a limit ordinal λ is a strictly increasing sequence $(\lambda(x))_{x < \omega}$ of ordinal terms with supremum λ . We use the standard assignment of fundamental sequences to limit ordinals below ε_0 in Cantor normal form, defined inductively by

$$(\gamma + \omega^{\beta+1})(x) \stackrel{\text{def}}{=} \gamma + \omega^\beta \cdot (x + 1), \quad (\gamma + \omega^\lambda)(x) \stackrel{\text{def}}{=} \gamma + \omega^{\lambda(x)}. \quad (11)$$

This particular assignment satisfies e.g. $0 < \lambda(x) < \lambda(y)$ for all $x < y$. For instance, $\omega(x) = x + 1$, $(\omega^{\omega^4} + \omega^{\omega^3 + \omega^2})(x) = \omega^{\omega^4} + \omega^{\omega^3 + \omega \cdot (x+1)}$.

The *predecessor* $P_x(\alpha)$ of an ordinal term $0 < \alpha < \varepsilon_0$ at $x \in \mathbb{N}$ is defined inductively by

$$P_x(\alpha + 1) \stackrel{\text{def}}{=} \alpha, \quad P_x(\lambda) \stackrel{\text{def}}{=} P_x(\lambda(x)). \quad (12)$$

In essence, the predecessor of an ordinal is obtained by repeatedly taking the x th element in the fundamental sequence of limit ordinals, until we finally reach a successor ordinal and may remove 1. For instance, $P_x(\omega^2) = P_x(\omega \cdot (x + 1)) = P_x(\omega \cdot x + x + 1) = \omega \cdot x + x$.

Hardy and Cichoń Hierarchies. Let $h: \mathbb{N} \rightarrow \mathbb{N}$ be a function. The *Hardy hierarchy* $(h^\alpha)_{\alpha \in \varepsilon_0}$ and the *Cichoń hierarchy* $(h_\alpha)_{\alpha \in \varepsilon_0}$ relative to h are defined for all $0 < \alpha < \varepsilon_0$ by

$$h^0(x) \stackrel{\text{def}}{=} x, \quad h^\alpha(x) \stackrel{\text{def}}{=} h^{P_x(\alpha)}(h(x)), \quad h_0(x) \stackrel{\text{def}}{=} 0, \quad h_\alpha(x) \stackrel{\text{def}}{=} 1 + h_{P_x(\alpha)}(h(x)). \quad (13)$$

Observe that $h^k(x) = h^{P_x(k)}(h(x)) = h^{k-1}(h(x))$ for some finite k is the k th iterate of h . This intuition carries over: h^α is a ‘transfinite’ iteration of the function h , using diagonalisation in the fundamental sequences to handle limit ordinals. A standard choice for the function h is the successor function, noted $H(x) \stackrel{\text{def}}{=} x + 1$; in that case, we see that a first diagonalisation yields $H^\omega(x) = H^x(x + 1) = 2x + 1$. The next diagonalisation occurs at $H^{\omega \cdot 2}(x) = H^{\omega+x}(x + 1) = H^\omega(2x + 1) = 4x + 3$. Fast-forwarding a bit, we get for instance a function of exponential growth $H^{\omega^2}(x) = 2^{x+1}(x + 1) - 1$, and later a non-elementary function $H^{\omega^3}(x)$ akin to a tower of exponentials of height x , and an ‘Ackermannian’ non primitive-recursive function H^{ω^ω} .

Both h^α and h_α are monotone and inflationary whenever h is monotone inflationary. Hardy functions are well-suited for expressing large iterates of a control function, and therefore for bounding the norms of elements in a controlled sequence. Cichoń functions are well-suited

for expressing the length of controlled sequences: we can compute how many times we should iterate h in order to compute $h^\alpha(x)$ using the corresponding Cichoń function [7]:

$$h^\alpha(x) = h^{h^\alpha(x)}(x). \quad (14)$$

5.2 Width Function for Dickson's Lemma

The starting point for our analysis is a *descent equation* for amortised controlled antichains through *residuals*, similar to the equations proven in [15, 41] for bad sequences (see Lemma 4). The key idea introduced in [15] is then to over-approximate the residuals of \mathbb{N}^d by working over *polynomial nwqos*, where disjoint sums are also allowed. Then, the notion of ‘over-approximation’ of residuals of polynomial nwqos is captured formally by showing the existence of a *normed reflection* into another polynomial nwqo. The final step lifts this analysis to ordinals, allowing to relate $W_{g,X}^a$ for a polynomial nwqo X to functions in the Cichoń hierarchy.

Strict Polynomial Normed wqos. Let us write $X \cdot p$ for $\overbrace{X \sqcup \dots \sqcup X}^{p \text{ times}}$, X^d for $\overbrace{X \times \dots \times X}^{d \text{ times}}$, and $\mathbf{0}$ for the empty nwqo. We call a nwqo of the form $\mathbb{N}^{d_1} \sqcup \dots \sqcup \mathbb{N}^{d_m}$ for some $m \geq 0$ and $d_1, \dots, d_m \geq 1$ a *strict polynomial nwqo*. The set of configurations $Q \times \mathbb{N}^c$ of an LCM with $|Q| = q$ locations and $|C| = d \geq 1$ counters, along with its ordering and infinite norm, is isomorphic to the strict polynomial nwqo $\mathbb{N}^d \cdot q$.

Residuals and a Descent Equation. Let $(X, \leq_X, |\cdot|_X)$ be a normed wqo and x be an element of X . We write $X_{\perp x} \stackrel{\text{def}}{=} \{y \in X \mid x \perp y\}$ for the *residual* of X in x . By the finite antichain condition, there cannot be infinite sequences of residuations $(\dots((X_{\perp x_0})_{\perp x_1})_{\perp x_2} \dots)_{\perp x_i}$ because $x_i \perp x_j$ for all $i < j$ and it would create an infinite antichain.

Consider now an amortised (g, n_0) -controlled antichain x_0, x_1, x_2, \dots over X . Assuming the sequence is not empty, then for all $i > 0$, $x_0 \perp x_i$, i.e. the suffix x_1, x_2, \dots is actually an antichain over $X_{\perp x_0}$. This suffix is now amortised $(g, g(n_0))$ -controlled, and thus of length bounded by $W_{g, X_{\perp x_0}}^a(g(n_0))$. This yields the following *descent equation* when considering all the possible amortised (g, n_0) -controlled antichains.

► **Lemma 4 (Descent Equation).** *Let $(X, \leq_X, |\cdot|_X)$ be a nwqo, $n_0 \in \mathbb{N}$ and $g: \mathbb{N} \rightarrow \mathbb{N}$. Then $W_{g,X}^a(n_0) = \max_{x \in X_{\leq n_0}} 1 + W_{g, X_{\perp x}}^a(g(n_0))$.*

Proof. Any amortised $(g, g(n_0))$ -controlled antichain x_1, x_2, \dots over $X_{\perp x}$ can be prefixed with any x such that $|x|_X \leq n_0$ to yield an amortised (g, n_0) -controlled antichain x, x_1, x_2, \dots , thus $W_{g,X}^a(n_0) \geq \max_{x \in X_{\leq n_0}} 1 + W_{g, X_{\perp x}}^a(g(n_0))$.

Conversely, let us pick an amortised (g, n_0) -controlled antichain x_0, x_1, x_2, \dots of maximal length $W_{g,X}^a(n_0)$; such a maximal antichain exists as discussed in Section 4.1. Then $x_0 \in X_{\leq n_0}$ and for all $i > 0$, $x_i \in X_{\perp x_0}$ and $|x_i|_X \leq g^{i-1}(g(n_0))$, thus $W_{g,X}^a(n_0) \leq \max_{x \in X_{\leq n_0}} 1 + W_{g, X_{\perp x}}^a(g(n_0))$. ◀

Reflecting Normed wqos. The descent equation, though it offers a way of computing the width function, quickly leads to complex residual expressions. We are going to over-approximate these $X_{\perp x}$'s using *nwqo reflections*, so that the computation can be carried out without leaving the realm of strict polynomial nwqos, leading to an *inductive* over-approximation of $X_{\perp x}$ over the structure of the strict polynomial nwqo X .

A *nwqo reflection* [41] is a mapping $r: X \rightarrow Y$ between two nwqos $(X, \leq_X, | \cdot |_X)$ and $(Y, \leq_Y, | \cdot |_Y)$ that satisfies the two following properties:

$$\forall x, x' \in X. \quad r(x) \leq_Y r(x') \quad \text{implies} \quad x \leq_X x', \quad (15)$$

$$\forall x \in X. \quad |r(x)|_Y \leq |x|_X. \quad (16)$$

In other words, a nwqo reflection is an order reflection that is not norm-increasing. This induces a quasi-ordering between nwqos, written $X \hookrightarrow Y$. Remark that reflections are compatible with disjoint sums and products [41, Prop. 3.5]:

$$X_0 \hookrightarrow Y_0 \text{ and } X_1 \hookrightarrow Y_1 \quad \text{imply} \quad X_0 \sqcup X_1 \hookrightarrow Y_0 \sqcup Y_1 \text{ and } X_0 \times X_1 \hookrightarrow Y_0 \times Y_1. \quad (17)$$

Crucially, nwqo reflections preserve amortised controlled antichains. Indeed, let $r: X \hookrightarrow Y$, and consider a sequence x_0, x_1, \dots over X . Then by (15), $r(x_0), r(x_1), \dots$ is an antichain when x_0, x_1, \dots is, and by (16), it is (g, n) -controlled when x_0, x_1, \dots is. Hence

$$X \hookrightarrow Y \quad \text{implies} \quad W_{g,X}^a(n) \leq W_{g,Y}^a(n) \text{ for all } g, n. \quad (18)$$

► **Remark 5.** By contrast with amortised controlled antichains, nwqo reflections do not preserve strongly controlled antichains. Consider for instance the strongly $(H, 4)$ -controlled antichain $(4, 3) (5, 2) (6, 1) (7, 0)$ over \mathbb{N}^2 where $H(x) \stackrel{\text{def}}{=} x + 1$. Let $\{e, o\}$ denote a set with two incomparable elements of norm zero, and consider the nwqo reflection $r: \mathbb{N}^2 \rightarrow (\mathbb{N}^2 \times \{e, o\})$ defined by $r(2n, m) \stackrel{\text{def}}{=} (2n, m, e)$ and $r(2n + 1, m) \stackrel{\text{def}}{=} (2n, m, o)$. The image of our antichain is the antichain $(4, 3, e) (4, 2, o) (6, 1, e) (6, 0, o)$, but it is not strongly $(H, 4)$ -controlled because $|(4, 2, o)| + 1 = 5 < 6 = |(6, 1, e)|$.

Inductive Reflection of Residuals. We provide a strict polynomial wqo reflecting $X_{\perp x}$ by induction over the structure of the strict polynomial nwqo X . The key difference compared to the analysis of bad sequences in [15, 42] occurs for $X = \mathbb{N}$: if $k \in \mathbb{N}$,

$$\mathbb{N}_{\perp k} = \mathbf{0}. \quad (19)$$

Regarding disjoint sums $X_0 \sqcup X_1$, it is plain that

$$(X_0 \sqcup X_1)_{\perp (x, i)} = (X_i)_{\perp x} \sqcup X_{1-i}. \quad (20)$$

Consider now $(\mathbb{N}^d)_{\perp \mathbf{v}}$ where $d > 1$ and $\mathbf{v} \in \mathbb{N}^d$. Observe that if $\mathbf{u} \in \mathbb{N}^d$ is such that $\mathbf{u} \perp \mathbf{v}$, then there exists $1 \leq i \leq d$ such that $\mathbf{u}(i) < \mathbf{v}(i)$, as otherwise we would have $\mathbf{u} \geq \mathbf{v}$. Thus

$$(\mathbb{N}^d)_{\perp \mathbf{v}} \hookrightarrow \bigsqcup_{1 \leq i \leq d} \mathbb{N}^{d-1} \times [\mathbf{v}(i)] \hookrightarrow \mathbb{N}^{d-1} \cdot \sum_{1 \leq i \leq d} \mathbf{v}(i). \quad (21)$$

Ordinal Notations. As it is more convenient to reason with ordinals than with polynomial nwqos, we use the following bijection between strict polynomial nwqos and ω^ω :

$$w(\mathbf{0}) \stackrel{\text{def}}{=} 0, \quad w(\mathbb{N}^d) \stackrel{\text{def}}{=} \omega^{d-1}, \quad w(X_0 \sqcup X_1) = w(X_0) \oplus w(X_1). \quad (22)$$

where ‘ \oplus ’ denotes the *natural sum* (aka *Hessenberg sum*) on ordinals: the natural sum $\alpha \oplus \beta$ of two ordinals with Cantor normal forms $\alpha = \sum_{i=1}^p \omega^{\alpha_i}$ and $\beta = \sum_{j=1}^m \omega^{\beta_j}$ is $\omega^{\gamma_1} + \dots + \omega^{\gamma_{p+m}}$ where the exponents $\gamma_1 \geq \dots \geq \gamma_{p+m}$ are a reordering of $\alpha_1, \dots, \alpha_p, \beta_1, \dots, \beta_m$. Given a strict polynomial nwqo $X = \bigsqcup_{i=1}^m \mathbb{N}^{d_i}$, its associated ordinal is $w(X) = \bigoplus_{i=1}^m \omega^{d_i-1}$. In the case of an LCM with $d \stackrel{\text{def}}{=} |C|$ counters and $q \stackrel{\text{def}}{=} |Q|$ locations, $w(Q \times \mathbb{N}^c) = \omega^{d-1} \cdot q$.

For each $n \in \mathbb{N}$, we define a relation ∂_n over ordinals in ω^ω that mirrors the inductive residuation and reflection operations on strict polynomial nwqos X over the ordinals $w(X)$:

$$\partial_n \alpha \stackrel{\text{def}}{=} \{ \gamma \oplus \partial_n \omega^d \mid \alpha = \gamma \oplus \omega^d \}, \quad \partial_n \omega^d \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } d = 0, \\ \omega^{d-1} \cdot n(d+1) & \text{otherwise.} \end{cases} \quad (23)$$

The intuition here is that $w(Y) \in \partial_n w(X)$ implies $X_{\perp x} \hookrightarrow Y$ for some $x \in X_{\leq n}$ (see Claim 6.1). Observe that $\alpha' \in \partial_n \alpha$ implies $\alpha' < \alpha$, thus $\bigcup_n \partial_n$ is a well-founded relation. This leads to the definition of an over-approximation of the width function $W_{g,X}^a$:

$$M_{g,\alpha}(n) \stackrel{\text{def}}{=} \max_{\alpha' \in \partial_n \alpha} \{1 + M_{g,\alpha'}(g(n))\}. \quad (24)$$

► **Proposition 6.** *Let $(X, \leq_X, | \cdot |_X)$ be a strict polynomial nwqo, $n_0 \in \mathbb{N}$, and $g: \mathbb{N} \rightarrow \mathbb{N}$. Then $W_{g,X}^a(n_0) \leq M_{g,w(X)}(n_0)$.*

Proof. The derivation relation ∂_n over ordinals was designed to satisfy the following.

▷ **Claim 6.1.** Let X be a strict polynomial nwqo and $x \in X_{\leq n}$ for some n . Then there exists a strict polynomial nwqo Y such that $w(Y) \in \partial_n w(X)$ and $X_{\perp x} \hookrightarrow Y$.

Proof. Let $X = \mathbb{N}^{d_1} \sqcup \dots \sqcup \mathbb{N}^{d_m}$ for some $d_1, \dots, d_m > 0$ and $x \in X_{\leq n}$; note that the existence of x rules out the case of $m = 0$. We proceed by induction over $m > 0$.

- The base case is $m = 1$, i.e., $X = \mathbb{N}^d$ for some $d > 0$.
 - If $d = 1$, then $X = \mathbb{N}$ and $w(X) = \omega^0$. Thus $X_{\perp x} = \mathbf{0}$ by (19) and $\partial_n w(X) = \{0\}$ by (23), and indeed $w(\mathbf{0}) = 0$.
 - Otherwise if $d > 0$, then $w(X) = \omega^{d-1}$ and x is a vector $\mathbf{v} \in (\mathbb{N}^d)_{\leq n}$. Equation (21) shows that $X_{\perp x} \hookrightarrow \mathbb{N}^{d-1} \cdot \sum_{1 \leq i \leq d} n = \mathbb{N}^{d-1} \cdot nd$, while $\partial_n w(X) = \{\omega^{d-2} \cdot nd\}$ by (23), and indeed $w(\mathbb{N}^{d-1} \cdot nd) = \omega^{d-2} \cdot nd$.
- For the induction step with $m > 1$, i.e., when $X = Y_0 \sqcup Y_1$ and $w(X) = w(Y_0) \oplus w(Y_1)$ for two non-empty strict polynomial nwqos Y_0 and Y_1 , then $x = (y, i)$ for some $i \in \{0, 1\}$ and $y \in (Y_i)_{\leq n}$. Equation (20) shows that $X_{\perp x} = (Y_i)_{\perp y} \sqcup Y_{1-i}$. By induction hypothesis, there exists Z_i such that $w(Z_i) \in \partial_n w(Y_i)$ and $(Y_i)_{\perp y} \hookrightarrow Z_i$, and the latter implies $X_{\perp x} \hookrightarrow Z_i \sqcup Y_{1-i}$ by (17). By (23), $w(Z_i \sqcup Y_{1-i}) = w(Z_i) \oplus w(Y_{1-i}) \in \partial_n w(X)$ as desired. ◀

Let us return to the proof of Proposition 6. Either $X_{\leq n_0}$ is empty and then $W_{g,X}^a(n_0) = 0 \leq M_{g,w(X)}(n_0)$, or there exists some $x \in X_{\leq n_0}$ that maximises $W_{g,X_{\perp x}}^a(g(n_0))$ in the descent equation from Lemma 4, i.e., such that

$$W_{g,X}^a(n_0) = 1 + W_{g,X_{\perp x}}^a(g(n_0)).$$

By Claim 6.1 there exists Y such that $w(Y) \in \partial_n w(X)$ and $X_{\perp x} \hookrightarrow Y$. By (18),

$$W_{g,X}^a(n_0) \leq 1 + W_{g,Y}^a(g(n_0)).$$

By well-founded induction on $w(Y) \in \partial_n w(X)$,

$$W_{g,Y}^a(g(n_0)) \leq M_{g,w(Y)}(g(n_0)).$$

Thus by definition of $M_{g,w(X)}$ in (24),

$$W_{g,X}^a(n_0) \leq 1 + M_{g,w(Y)}(g(n_0)) \leq M_{g,w(X)}(n_0). \quad \blacktriangleleft$$

It remains to compare $M_{g,\alpha}$ with standard subrecursive functions like the Cichoń functions, which was already done in [42, Sec. 2.4.3] for a very similar function.

► **Proposition 7.** *Let n be in \mathbb{N} , $g: \mathbb{N} \rightarrow \mathbb{N}$ a monotone inflationary function, and $0 < \alpha < \omega^d$. Then $M_{g,\alpha}(n) \leq 1 + M_{g,P_{nd}(\alpha)}(n)$.*

Proof. The function $M_{g,\alpha}(n)$ is very nearly the same as the one studied in [42, Sec. 2.4.3], except for the coefficient $n(d+1)$ instead of nd in the definition of $\partial_n \omega^d$ in (23). This minor difference turns out to have no impact on several of the arguments used in [42, Sec. 2.4.3], which we summarise here.

Let us start by introducing a *structural ordering* for ordinals in ω^ω in Cantor normal form [42, Eq. 2.69]:

$$\omega^{d_1} + \dots + \omega^{d_m} \sqsubseteq \omega^{d'_1} + \dots + \omega^{d'_k} \quad \text{if } m \leq k \text{ and } \forall 1 \leq i \leq m. d_i \leq d'_i. \quad (25)$$

Clearly, for all $\alpha, \gamma \in \omega^\omega$,

$$\alpha \sqsubseteq \alpha \oplus \gamma. \quad (26)$$

The $M_{g,\alpha}$ functions are monotone with respect to the structural ordering: if α and α' are two ordinals in ω^ω , then [42, Lem. 2.31]

$$\alpha \sqsubseteq \alpha' \quad \text{implies} \quad M_{g,\alpha}(n) \leq M_{g,\alpha'}(n). \quad (27)$$

This allows to prove that there exists a ‘maximising strategy’ when choosing how to decompose $\alpha = \omega^{d_1} + \dots + \omega^{d_m}$ as $\gamma \oplus \omega^{d_i}$ (where $\gamma = \sum_{j \neq i} \omega^{d_j}$), which is to pick $i = m$, i.e., a term with minimal exponent. Indeed, if $\alpha = \gamma + \omega^d$, then [42, Lem. 2.32]

$$M_{g,\alpha}(n) = 1 + M_{g,\gamma + \partial_n \omega^d}(g(n)). \quad (28)$$

We can now prove Proposition 7 by adapting the arguments of [42, Cor. 2.33]. Let us write α in Cantor normal form as $\gamma + \omega^{d'}$ for some $\gamma < \alpha$ and $d' < d$. By (28), $M_{g,\alpha}(n) = 1 + M_{g,\gamma + \partial_n \omega^{d'}}(g(n))$. If $d' = 0$, then by (23),

$$\gamma + \partial_n \omega^0 = \gamma = P_{nd}(\gamma + 1) = P_{nd}(\alpha)$$

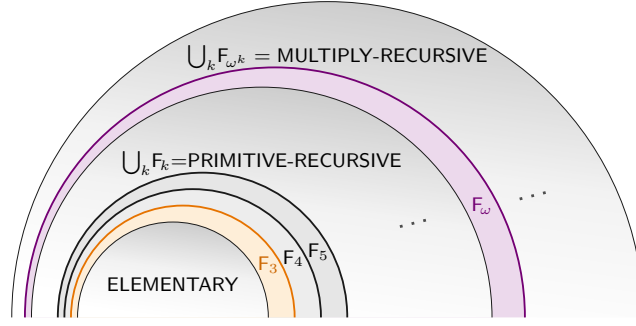
and the statement holds. Otherwise, by (23),

$$\begin{aligned} \gamma + \partial_n \omega^{d'} &= \gamma + \omega^{d'-1} \cdot n(d'+1) \\ &\sqsubseteq \gamma + \omega^{d'-1} \cdot nd && \text{by (26) since } d' < d \\ &\sqsubseteq \gamma + \omega^{d'-1} \cdot nd + P_{nd}(\omega^{d'-1}) && \text{by (26)} \\ &= P_{nd}(\gamma + \omega^{d'}) && \text{by definition of predecessors in (11–12)} \\ &= P_{nd}(\alpha), \end{aligned}$$

and the statement holds by (27). ◀

This extra twist of using a predecessor function different from the standard one from (12) can be avoided by instead over-approximating the control function g .

► **Theorem 8 (Width Function for Strict Polynomial nwqos).** *Let $d > 0$, $(X, \leq_X, | \cdot |_X)$ be a strict polynomial nwqo with $w(X) < \omega^d$, $n_0 \in \mathbb{N}$, $g: \mathbb{N} \rightarrow \mathbb{N}$ monotone inflationary, and let $h: \mathbb{N} \rightarrow \mathbb{N}$ be a monotone function such that $h(x \cdot d) \geq g(x) \cdot d$ for all x . Then $W_{g,X}^a(n_0) \leq h_{w(X)}(n_0 d)$.*



■ **Figure 3** Pinpointing $F_3 = \text{TOWER}$ and $F_\omega = \text{ACKERMANN}$ among the complexity classes beyond ELEMENTARY [39].

Proof. By Proposition 6, it suffices to show that $M_{g,w(X)}(n) \leq h_{w(X)}(nd)$, which we do by induction over $\alpha \stackrel{\text{def}}{=} w(X)$. If $\alpha = 0$, then $\partial_n \alpha = \emptyset$ thus $M_{g,\alpha}(n) = 0 \leq h_\alpha(nd)$. Otherwise, by Proposition 7, $M_{g,\alpha}(n) \leq 1 + M_{g,P_{nd}(\alpha)}(g(n))$. Because $P_{nd}(\alpha) < \alpha$, we can apply the induction hypothesis, yielding $M_{g,\alpha}(n) \leq 1 + h_{P_{nd}(\alpha)}(g(n)d) \leq 1 + h_{P_{nd}(\alpha)}(h(nd)) = h_\alpha(nd)$, where the last inequality follows from $h(nd) \geq g(n)d$ and the monotonicity of $h_{P_{nd}(\alpha)}$. ◀

Setting $h(x) \stackrel{\text{def}}{=} g(x)d$ always satisfies the conditions of the theorem. There are cases where setting $h \stackrel{\text{def}}{=} g$ suffices: e.g., $g(x) \stackrel{\text{def}}{=} 2x$, $g(x) \stackrel{\text{def}}{=} x^2$, $g(x) \stackrel{\text{def}}{=} 2^x$, and more generally whenever g is *super-homogeneous*, i.e. satisfies $g(dx) \geq g(x)d$ for all $d, x \geq 1$. In the case of LCMs, where $w(Q \times \mathbb{N}^c) < \omega^d$ if $d \stackrel{\text{def}}{=} |C| > 0$, a control function $g(x) \stackrel{\text{def}}{=} x + 1 = H(x)$ fits, thus setting $h(x) \stackrel{\text{def}}{=} x + d = H^d(x)$ satisfies $h(dx) = dx + d = (x + 1)d = g(x)d$.

By (8), Lemma 3, and (14), Theorem 8 also yields a bound on the strong norm function.

► **Corollary 9** (Strong Norm Function for Strict Polynomial nwqos). *Let d, X, n_0, g , and h be as in Theorem 8. Then $N_{g,X}^s(n_0) \leq h^{w(X)}(n_0d)$.*

6 Wrapping up

We have now all the ingredients needed to prove an F_d upper bound on LCM Reachability. Let us first recall the definition of the fast-growing complexity classes from [39].

Fast-Growing Complexity Classes. The *fast-growing* complexity classes [39] form a strict ordinal-indexed hierarchy of complexity classes $(F_\alpha)_{\alpha < \varepsilon_0}$ using the Hardy functions $(H^\alpha)_{\alpha < \varepsilon_0}$ relative to $H(x) \stackrel{\text{def}}{=} x + 1$ as a standard against which to measure high complexities. Let

$$\mathcal{F}_{<\alpha} \stackrel{\text{def}}{=} \bigcup_{\beta < \omega^\alpha} \text{FDTIME}(H^\beta(n)), \quad F_\alpha \stackrel{\text{def}}{=} \bigcup_{p \in \mathcal{F}_{<\alpha}} \text{DTIME}(H^{\omega^\alpha}(p(n))). \quad (29)$$

Then $\mathcal{F}_{<\alpha}$ is the class of functions computed by deterministic Turing machines in time $O(H^\beta(n))$ for some $\beta < \omega^\alpha$; this captures for instance the class of Kalmar elementary functions as $\mathcal{F}_{<3}$ and the class of primitive-recursive functions as $\mathcal{F}_{<\omega}$ [31, 49]. The class F_α is the class of decision problems solved by deterministic Turing machines in time $O(H^{\omega^\alpha}(p(n)))$ for some function $p \in \mathcal{F}_{<\alpha}$. The intuition behind this quantification of p is that, just like e.g. $\text{EXP} = \bigcup_{p \in \text{poly}} \text{DTIME}(2^{p(n)})$ quantifies over polynomial functions to provide enough ‘wiggle room’ to account for polynomial reductions, F_α is closed under $\mathcal{F}_{<\alpha}$ reductions [39, Thms. 4.7 and 4.8]. For instance, $\text{TOWER} \stackrel{\text{def}}{=} F_3$ defines the class of problems that can be solved in time bounded by a tower of exponentials of elementary height in the size of the input, $\bigcup_{k \in \mathbb{N}} F_k$

is the class of primitive-recursive decision problems, and $\text{ACKERMANN} \stackrel{\text{def}}{=} \mathbb{F}_\omega$ is the class of problems that can be solved in time bounded by the Ackermann function applied to some primitive-recursive function of the input size; see Figure 3 for a depiction.

Upper Bound. Recall from Section 3.3 that a pseudo-witness for coverability of a configuration $q_f(\mathbf{v}_f)$ in a LCM with $d \stackrel{\text{def}}{=} |\mathbb{C}| > 0$ counters and $q \stackrel{\text{def}}{=} |Q|$ locations is a strongly $(H, |\mathbf{v}_f|)$ -controlled bad sequence over $Q \times \mathbb{N}^{\mathbb{C}}$, which as discussed in Section 5.2 is a strict polynomial wqo with $w(Q \times \mathbb{N}^{\mathbb{C}}) = \omega^{d-1} \cdot q < \omega^d$, and that $h \stackrel{\text{def}}{=} H^d$ fits the conditions of Theorem 8 and Corollary 9. As stated in Theorem 1, together with the lower bounds from [40], the following entails the \mathbb{F}_d -completeness of LCM Reachability with a fixed number $d \geq 3$ of counters.

► **Proposition 10** (Upper Bound for LCM Reachability). *LCM Reachability is in \mathbb{F}_ω , and in \mathbb{F}_d if the number $d \geq 3$ of counters is fixed.*

Proof. Let $n_0 \stackrel{\text{def}}{=} |\mathbf{v}_f|$ be the infinite norm of the target configuration, $d \stackrel{\text{def}}{=} |\mathbb{C}| \geq 3$ be the number of counters, and $q \stackrel{\text{def}}{=} |Q| \geq 1$ the number of locations. By Corollary 9, the elements in a pseudo-witness of the coverability of $q_f(\mathbf{v}_f)$ are of norm at most $N \stackrel{\text{def}}{=} N_{H,Q \times \mathbb{N}^{\mathbb{C}}}^s(n_0) = h^{\omega^{d-1} \cdot q}(n_0)$ for $h(x) \stackrel{\text{def}}{=} H^d(x)$. Let $n \stackrel{\text{def}}{=} \max\{qd - 1, n_0\}$. As shown in Lemma 11 in Appendix A, this means that

$$N \leq H^{\omega^{d-1} \cdot qd}(n_0) \leq H^{\omega^{d-1} \cdot qd}(n) \leq H^{\omega^d}(n) \quad (30)$$

by monotonicity of the Hardy functions.

Note that there are at most $q(N+1)^d$ different configurations in $Q \times \mathbb{N}^{\mathbb{C}}$ of norm bounded by N , i.e., $|(Q \times \mathbb{N}^{\mathbb{C}})_{\leq N}| \leq q(N+1)^d$. By (9), this is also a bound on the strong length function $L_{H,Q \times \mathbb{N}^{\mathbb{C}}}^s(n_0)$. Thus the number of steps in the backward coverability algorithm is bounded by $q(N+1)^d$, and each step can be carried in time $O(N)$, hence the algorithm works in deterministic time $O(q(N+1)^{d+1}) = O(f(N)) = O(f(H^{\omega^d}(n)))$ for some elementary function $f \in \mathcal{F}_{<3}$. By [39, Cor. A.9], there exists an elementary inflationary function $p \in \mathcal{F}_{<3}$ such that $f(H^{\omega^d}(n)) \leq H^{\omega^d}(p(n))$: the backward coverability algorithm therefore works in deterministic time $O(H^{\omega^d}(p(n)))$ for some $p \in \mathcal{F}_{<3}$, which is an expression of the form (29).

Therefore, LCM Reachability is in \mathbb{F}_d when d is fixed, and in \mathbb{F}_ω otherwise because $p(n) \geq n \geq d - 1$ and thus $H^{\omega^d}(p(n)) \leq H^{\omega^\omega}(p(n))$. ◀

7 Concluding Remarks

We have shown the \mathbb{F}_d -completeness of reachability in lossy counter machines with a fixed number $d \geq 3$ of counters. The key novelty is that we analyse the length of controlled antichains over \mathbb{N}^d rather than that of controlled bad sequences. A possible explanation why this leads to improved upper bounds is that the *ordinal width* of \mathbb{N}^d , i.e., the ordinal rank of its antichains, is conjectured to be ω^{d-1} [13], while its *maximal order type*, i.e., the ordinal rank of its bad sequences, is well-known to be ω^d [9].

Our approach might be employed to tackle related parameterised complexity gaps, like the one between $\mathbb{F}_{\omega^{m-2}}$ -hardness [26] and $\mathbb{F}_{\omega^{m-1}+1}$ membership [41] of reachability in *lossy channel systems* with $m \geq 4$ channel symbols and a single channel. Those results rely however on the set of finite words over an alphabet of size m being a wqo for Higman's scattered subword ordering [22], for which the ordinal width and maximal order type coincide at $\omega^{\omega^{m-1}}$ [13, 9].

A Technical Appendix

The Hardy functions enjoy some nice identities (which justify the superscript notation): for all h , α, β , and x

$$h^\alpha \circ h^\beta(x) = h^{\alpha+\beta}(x), \quad (h^\alpha)^\beta(x) = h^{\alpha \cdot \beta}(x), \quad (31)$$

provided $\alpha + \beta$ (resp. $\alpha \cdot \beta$) satisfies specific conditions like being *tree-structured* [14, 7]; this will be the case in our applications of these identities. The condition on $\alpha \cdot \beta$ is necessary; for instance, (31) fails for $(H^2)^\omega(x) = (H^2)^x(x+2) = 3x+2$ because $2 \cdot \omega = \omega$ and $H^\omega(x) = 2x+1$.

Lemma 11 circumvents this limitation in a special case. Let us first mention that the Hardy functions $(h^\alpha)_{\alpha < \varepsilon_0}$ can also be defined by transfinite induction on α by

$$h^0(x) \stackrel{\text{def}}{=} x, \quad h^{\alpha+1}(x) \stackrel{\text{def}}{=} h^\alpha(h(x)), \quad h^\lambda(x) \stackrel{\text{def}}{=} h^{\lambda(x)}(x). \quad (32)$$

► **Lemma 11.** *Let $\alpha \in \varepsilon_0$, $d \in \mathbb{N}$, $x \in \mathbb{N}$, and $h: \mathbb{N} \rightarrow \mathbb{N}$ be monotone inflationary. Then $(h^d)^\alpha(x) \leq h^{\alpha \cdot d}(x)$.*

Proof. Let us first show by induction over $d \in \mathbb{N}$ that for all limit ordinals $\lambda < \varepsilon_0$,

$$x \leq y \quad \text{implies} \quad h^{\lambda(x) \cdot d}(y) \leq h^{\lambda \cdot d}(y). \quad (33)$$

For the base case $d = 0$, $h^{\lambda(x) \cdot 0}(y) = y = h^{\lambda \cdot 0}(y)$. For the induction step $d + 1$,

$$\begin{aligned} h^{\lambda(x) \cdot (d+1)}(y) &= h^{\lambda(x) \cdot d}(h^{\lambda(x)}(y)) && \text{by (31)} \\ &\leq h^{\lambda(x) \cdot d}(h^\lambda(y)) && \text{since } h^{\lambda(x)}(x) \stackrel{(32)}{=} h^\lambda(x) \leq h^\lambda(y) \text{ by monotonicity of } h^\lambda \\ &\leq h^{\lambda \cdot d}(h^\lambda(y)) && \text{by ind. hyp. since } x \leq y \leq h^\lambda(y) \\ &= h^{\lambda \cdot (d+1)}(y) && \text{by (31)}. \end{aligned}$$

Let us now prove Lemma 11 by transfinite induction over α . For the base case $\alpha = 0$,

$$(h^d)^0(x) = x = h^{\alpha \cdot 0}(x).$$

For the successor case $\alpha + 1$,

$$\begin{aligned} (h^d)^{\alpha+1}(x) &= (h^d)^\alpha(h^d(x)) && \text{by (32)} \\ &\leq h^{\alpha \cdot d}(h^d(x)) && \text{by ind. hyp. on } \alpha < \alpha + 1 \\ &\leq h^{\alpha \cdot d + d}(x) && \text{by (31)} \\ &= h^{(\alpha+1) \cdot d}(x). \end{aligned}$$

For the limit case λ ,

$$\begin{aligned} (h^d)^\lambda(x) &= (h^d)^{\lambda(x)}(x) && \text{by (32)} \\ &\leq h^{\lambda(x) \cdot d}(x) && \text{by ind. hyp. on } \lambda(x) < \lambda \\ &\leq h^{\lambda \cdot d}(x) && \text{by (33)}. \end{aligned} \quad \blacktriangleleft$$

References

- 1 Parosh A. Abdulla, Karlis Čerāns, Bengt Jonsson, and Yih-Kuen Tsay. Algorithmic analysis of programs with well quasi-ordered domains. *Information and Computation*, 160(1–2):109–127, 2000. doi:10.1006/inco.1999.2843.

- 2 Parosh A. Abdulla and Bengt Jonsson. Verifying programs with unreliable channels. *Information and Computation*, 127(2):91–101, 1996. doi:10.1006/inco.1996.0053.
- 3 Sergio Abriola, Santiago Figueira, and Gabriel Senno. Linearizing well-quasi orders and bounding the length of bad sequences. *Theoretical Computer Science*, 603:3–22, 2015. doi:10.1016/j.tcs.2015.07.012.
- 4 Roberto M. Amadio and Charles Meyssonnier. On decidability of the control reachability problem in the asynchronous π -calculus. *Nordic Journal of Computing*, 9(2):70–101, 2002.
- 5 André Arnold and M. Latteux. Récursivité et cones rationnels fermés par intersection. *CALCOLO*, 15(4):381–394, 1978. doi:10.1007/BF02576519.
- 6 Pierre Chambart and Philippe Schnoebelen. The ordinal recursive complexity of lossy channel systems. In *Proceedings of LICS 2008*, pages 205–216. IEEE, 2008. doi:10.1109/LICS.2008.47.
- 7 E. Adam Cichoń and Elias Tahhan Bittar. Ordinal recursive bounds for Higman’s Theorem. *Theoretical Computer Science*, 201(1–2):63–84, 1998. doi:10.1016/S0304-3975(97)00009-1.
- 8 Peter Clote. On the finite containment problem for Petri nets. *Theoretical Computer Science*, 43:99–105, 1986. doi:10.1016/0304-3975(86)90169-6.
- 9 Dick H. J. de Jongh and Rohit Parikh. Well-partial orderings and hierarchies. *Indagationes Mathematicae*, 39(3):195–207, 1977. doi:10.1016/1385-7258(77)90067-1.
- 10 Normann Decker and Daniel Thoma. On freeze LTL with ordered attributes. In *Proceedings of FoSSaCS 2016*, volume 9634 of *Lecture Notes in Computer Science*, pages 269–284. Springer, 2016. doi:10.1007/978-3-662-49630-5_16.
- 11 Stéphane Demri and Ranko Lazic. LTL with the freeze quantifier and register automata. *ACM Transactions on Computational Logic*, 10(3):16:1–16:30, 2009. doi:10.1145/1507244.1507246.
- 12 Leonard Eugene Dickson. Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors. *American Journal of Mathematics*, 35(4):413–422, 1913. doi:10.2307/2370405.
- 13 Mirna Džamonja, Sylvain Schmitz, and Philippe Schnoebelen. On ordinal invariants in well quasi orders and finite antichain orders. In Monika Seisenberger, Peter Schuster, and Andreas Weiermann, editors, *Well-Quasi Orders in Computation, Logic, Language and Reasoning*, Trends in Logic. Springer, 2019. To appear. arXiv:1711.00428[math.LO].
- 14 Matthew V. H. Fairtlough and Stanley S. Wainer. Hierarchies of provably recursive functions. In Sam Buss, editor, *Handbook of Proof Theory*, volume 137 of *Studies in Logic and the Foundations of Mathematics*, chapter III, pages 149–207. Elsevier, 1998. doi:10.1016/S0049-237X(98)80018-9.
- 15 Diego Figueira, Santiago Figueira, Sylvain Schmitz, and Philippe Schnoebelen. Ackermannian and primitive-recursive bounds with Dickson’s Lemma. In *Proceedings of LICS 2011*, pages 269–278. IEEE, 2011. doi:10.1109/LICS.2011.39.
- 16 Diego Figueira and Luc Segoufin. Future-looking logics on data words and trees. In *Proceedings of MFCS 2009*, volume 5734 of *Lecture Notes in Computer Science*, pages 331–343. Springer, 2009. doi:10.1007/978-3-642-03816-7_29.
- 17 Alain Finkel and Philippe Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1–2):63–92, 2001. doi:10.1016/S0304-3975(00)00102-X.
- 18 Radu Grigore and Nikos Tzevelekos. History-register automata. *Logical Methods in Computer Science*, 12(1):7:1–7:32, 2016. doi:10.2168/LMCS-12(1:7)2016.
- 19 Christoph Haase, Sylvain Schmitz, and Philippe Schnoebelen. The power of priority channel systems. *Logical Methods in Computer Science*, 10(4):4:1–4:39, 2014. doi:10.2168/LMCS-10(4:4)2014.
- 20 Serge Haddad, Sylvain Schmitz, and Philippe Schnoebelen. The ordinal recursive complexity of timed-arc Petri nets, data nets, and other enriched nets. In *Proceedings of LICS 2012*, pages 355–364. IEEE, 2012. doi:10.1109/LICS.2012.46.
- 21 Matthew Hague. Senescent ground tree rewriting systems. In *Proceedings of CSL-LICS 2014*, pages 48:1–48:10. ACM, 2014. doi:10.1145/2603088.2603112.

- 22 Graham Higman. Ordering by divisibility in abstract algebras. *Proceedings of the London Mathematical Society*, 3(2):326–336, 1952. doi:10.1112/plms/s3-2.1.326.
- 23 Piotr Hofman and Patrick Totzke. Trace inclusion for one-counter nets revisited. *Theoretical Computer Science*, 735:50–63, 2018. doi:10.1016/j.tcs.2017.05.009.
- 24 Rodney R. Howell, Louis E. Rosier, Dung T. Huynh, and Hsu-Chun Yen. Some complexity bounds for problems concerning finite and 2-dimensional vector addition systems with states. *Theoretical Computer Science*, 46:107–140, 1986. doi:10.1016/0304-3975(86)90026-5.
- 25 Petr Jančar and Sylvain Schmitz. Bisimulation equivalence of first-order grammars is ACKERMANN-complete. In *Proceedings of LICS 2019*. IEEE, 2019. To appear. arXiv:1901.07170[cs.LG].
- 26 Prateek Karandikar and Sylvain Schmitz. The parametric ordinal-recursive complexity of Post embedding problems. In *Proceedings of FoSSaCS 2013*, volume 7794 of *Lecture Notes in Computer Science*, pages 273–288. Springer, 2013. doi:10.1007/978-3-642-37075-5_18.
- 27 Joseph B. Kruskal. The theory of well-quasi-ordering: A frequently discovered concept. *Journal of Combinatorial Theory, Series A*, 13(3):297–305, 1972. doi:10.1016/0097-3165(72)90063-5.
- 28 Ranko Lazić, Joël O. Ouaknine, and James B. Worrell. Zeno, Hercules and the Hydra: Safety metric temporal logic is ACKERMANN-complete. *ACM Transactions on Computational Logic*, 17(3), 2016. doi:10.1145/2874774.
- 29 Ranko Lazić and Sylvain Schmitz. Non-elementary complexities for branching VASS, MELL, and extensions. *ACM Transactions on Computational Logic*, 16(3):20:1–20:30, 2015. doi:10.1145/2733375.
- 30 Ranko Lazić and Sylvain Schmitz. The complexity of coverability in ν -Petri nets. In *Proceedings of LICS 2016*, pages 467–476. ACM, 2016. doi:10.1145/2933575.2933593.
- 31 Martin H. Löb and Stanley S. Wainer. Hierarchies of number theoretic functions, I. *Archiv für Mathematische Logik und Grundlagenforschung*, 13:39–51, 1970. doi:10.1007/BF01967649.
- 32 Ernst W. Mayr and Albert R. Meyer. The complexity of the finite containment problem for Petri nets. *Journal of the ACM*, 28(3):561–576, 1981. doi:10.1145/322261.322271.
- 33 Richard Mayr. Undecidable problems in unreliable computations. *Theoretical Computer Science*, 297(1–3):337–354, 2003. doi:10.1016/S0304-3975(02)00646-1.
- 34 Kenneth McAloon. Petri nets and large finite sets. *Theoretical Computer Science*, 32(1–2):173–183, 1984. doi:10.1016/0304-3975(84)90029-X.
- 35 Robin Milner. Operational and algebraic semantics of concurrent processes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 19, pages 1201–1242. Elsevier, 1990. doi:10.1016/B978-0-444-88074-1.50024-X.
- 36 Angelo Montanari, Gabriele Puppis, and Pietro Sala. Maximal decidable fragments of Halpern and Shoham’s modal logic of intervals. In *Proceedings of ICALP 2010*, volume 6199 of *Lecture Notes in Computer Science*, pages 345–356. Springer, 2010. doi:10.1007/978-3-642-14162-1_29.
- 37 Guillermo A. Pérez. The fixed initial credit problem for partial-observation energy games is ACK-complete. *Information Processing Letters*, 118:91–99, 2017. doi:10.1016/j.ip1.2016.10.005.
- 38 Fernando Rosa-Velardo. Ordinal recursive complexity of unordered data nets. *Information and Computation*, 254(1):41–58, 2017. doi:10.1016/j.ic.2017.02.002.
- 39 Sylvain Schmitz. Complexity hierarchies beyond ELEMENTARY. *ACM Transactions on Computation Theory*, 8(1):3:1–3:36, 2016. doi:10.1145/2858784.
- 40 Sylvain Schmitz. *Algorithmic Complexity of Well-Quasi-Orders*. Habilitation thesis, École Normale Supérieure Paris-Saclay, 2017. tel.archives-ouvertes.fr:tel-01663266.
- 41 Sylvain Schmitz and Philippe Schnoebelen. Multiply-recursive upper bounds with Higman’s Lemma. In *Proceedings of ICALP 2011*, volume 6756 of *Lecture Notes in Computer Science*, pages 441–452. Springer, 2011. doi:10.1007/978-3-642-22012-8_35.
- 42 Sylvain Schmitz and Philippe Schnoebelen. Algorithmic aspects of wqo theory. Lecture notes, 2012. cel.archives-ouvertes.fr:cel-00727025.

- 43 Sylvain Schmitz and Philippe Schnoebelen. The power of well-structured systems. In *Proceedings of Concur 2013*, volume 8052 of *Lecture Notes in Computer Science*, pages 5–24. Springer, 2013. doi:10.1007/978-3-642-40184-8_2.
- 44 Philippe Schnoebelen. Verifying lossy channel systems has nonprimitive recursive complexity. *Information Processing Letters*, 83(5):251–261, 2002. doi:10.1016/S0020-0190(01)00337-4.
- 45 Philippe Schnoebelen. Lossy counter machines decidability cheat sheet. In *Proceedings of RP 2010*, volume 6227 of *Lecture Notes in Computer Science*, pages 51–75. Springer, 2010. doi:10.1007/978-3-642-15349-5_4.
- 46 Philippe Schnoebelen. Revisiting Ackermann-hardness for lossy counter machines and reset Petri nets. In *Proceedings of MFCS 2010*, volume 6281 of *Lecture Notes in Computer Science*, pages 616–628. Springer, 2010. doi:10.1007/978-3-642-15155-2_54.
- 47 Helmut Schwichtenberg and Stanley S. Wainer. *Proofs and Computation*. Perspectives in Logic. Cambridge University Press, 2012.
- 48 Alasdair Urquhart. The complexity of decision procedures in relevance logic II. *Journal of Symbolic Logic*, 64(4):1774–1802, 1999. doi:10.2307/2586811.
- 49 Stanley S. Wainer. Ordinal recursion, and a refinement of the extended Grzegorzcyk hierarchy. *Journal of Symbolic Logic*, 37(2):281–292, 1972. doi:10.2307/2272973.
- 50 Andreas Weiermann. Complexity bounds for some finite forms of Kruskal’s Theorem. *Journal of Symbolic Computation*, 18(5):463–488, 1994. doi:10.1006/jsc.1994.1059.