Revision 959, January 12, 2011.

Ackermannian and Primitive-Recursive Bounds with Dickson's Lemma*

Diego Figueira[1] Santiago Figueira[2] Sylvain Schmitz[3]
Philippe Schnoebelen[3]

[1] University of Warsaw
figueira@lsv.ens-cachan.fr
[2] Dept. of Computer Science, FCEyN, University of Buenos Aires & CONICET
santiago@dc.uba.ar
[3] LSV, ENS Cachan & CNRS
{schmitz,phs}@lsv.ens-cachan.fr

Abstract

Dickson's Lemma is a simple yet powerful tool widely used in decidability proofs, especially when dealing with counters or related data structures in algorithmics, verification and model-checking, constraint solving, logic, etc. While Dickson's Lemma is well-known, most computer scientists are not aware of the complexity upper bounds that are entailed by its use. This is mainly because, on this issue, the existing literature is not very accessible.

We propose a new analysis of the length of *bad sequences* over $(\mathbb{N}^k, \leq)$, improving on earlier results and providing upper bounds that are essentially tight. This analysis is complemented by a "user guide" explaining through practical examples how to easily derive complexity upper bounds from Dickson's Lemma.

1 Introduction

For some dimension k, let $(\mathbb{N}^k, \leq)$ be the set of k-tuples of natural numbers ordered with the natural product ordering

$$x = \langle x[1], \ldots, x[k] \rangle \;\leq\; y = \langle y[1], \ldots, y[k] \rangle \overset{\text{def}}{\Leftrightarrow} x[1] \leq y[1] \wedge \cdots \wedge x[k] \leq y[k] \; .$$

Dickson's Lemma is the statement that $(\mathbb{N}^k, \leq)$ is a well-quasi-ordering (a "wqo"). This means that there exist no infinite strictly decreasing sequences $x_0 > x_1 > x_2 > \cdots$ of k-tuples, and that there are no infinite antichains, i.e., sequences of pairwise incomparable k-tuples (Kruskal, 1972; Milner, 1985). Equivalently, every infinite sequence $\mathbf{x} = x_0, x_1, x_2, \ldots$ over $\mathbb{N}^k$ contains an *increasing pair* $x_{i_1} \leq x_{i_2}$ for some $i_1 < i_2$. We say that sequences with an increasing pair

*Work supported by the Agence Nationale de la Recherche, grant ANR-06-SETIN-001, and by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the FET-Open grant agreement FOX, number FP7-ICT-233599.

$x_{i_1} \leq x_{i_2}$ are *good* sequences. A sequence that is not good is bad. Dickson's Lemma states that every infinite sequence over $\mathbb{N}^k$ is good, i.e., that bad sequences are finite.

Using Dickson's Lemma "The most frequently rediscovered mathematical theorem" according to (Becker and Weispfenning, 1993, p. 184), Dickson's Lemma plays a fundamental role in several areas of computer science, where it is used to prove that some algorithmic constructions terminate, that some sets are finite, or semilinear, *etc.* In Section 7 later, we give examples dealing with counter machines and Petri nets because we are more familiar with this area, but many others exist.

Example 1.1. The following simple program is shown in (Podelski and Rybalchenko, 2004) to terminate for every input $\langle a, b\rangle \in \mathbb{N}^2$:

$\textsc{choice}\ (a, b)$
while $a > 0 \wedge b > 1$
  $\langle a, b\rangle \longleftarrow \langle a - 1, a\rangle$
 or
  $\langle a, b\rangle \longleftarrow \langle b - 2, a + 1\rangle$
end

We leave it to the reader to check that, in fact, any sequence of successive configurations $x_0 = \langle a, b\rangle, x_1, x_2, \ldots$ of this program is a bad sequence over $\mathbb{N}^2$, and is thus finite by Dickson's Lemma. Let $\textsc{Time}(a, b)$ be the maximal number of times the **while** loop of CHOICE can be executed—a natural complexity measure. If we could bound the length of bad sequences over $\mathbb{N}^2$ that start with $\langle a, b\rangle$, then we would have an upper-bound on $\textsc{Time}(a, b)$. □

In order to bound the running time of algorithms that rely on Dickson's Lemma, it is usually necessary to know (or to bound) the value of the index i_2 in the first increasing pair $x_{i_1} \leq x_{i_2}$. It is widely felt, at least in the field of verification and model-checking, that relying on Dickson's Lemma when proving decidability or finiteness does not give any useful information regarding complexity, or that it gives upper bounds that are not explicit and/or not meaningful. Indeed, bad sequences can be arbitrarily long.

The Length of Bad Sequences It is easy to construct arbitrarily long bad sequences, even when starting from a fixed first element. Consider $\mathbb{N}^2$ and fix $x_0 = \langle 0, 1\rangle$. Then the following

$$\langle 0, 1\rangle,\ \langle L, 0\rangle,\ \langle L - 1, 0\rangle,\ \langle L - 2, 0\rangle,\ \cdots\ \langle 2, 0\rangle,\ \langle 1, 0\rangle$$

is a bad sequence of length $L + 1$. What makes such examples possible is the "uncontrolled" jump from an element like x_0 to an *arbitrarily* large next element like here $x_1 = \langle L, 0\rangle$. Indeed, when one only considers bad sequences displaying some controlled behaviour (in essence, bad sequences of bounded complexity), upper bounds on their lengths certainly exist.

Let us fix a *control function* $f : \mathbb{N} \to \mathbb{N}$. We say that a sequence $\mathbf{x} = x_0, x_1, \ldots$ over $\mathbb{N}^k$ is *t-controlled* for some t in $\mathbb{N}$ if the infinity norm of the x_i verifies $|x_i|_\infty < f(i + t)$ for all indexes $i = 0, 1, \ldots$ Then, for fixed k, t, and f, there are only finitely many t-controlled bad sequences (by Dickson's Lemma *cum* Kőnig's Lemma) and a maximum length exists. This maximum length can even be computed if f is recursive.

In this paper, we write $L_{k,f}(t)$ for the maximal length of a t-controlled bad sequence (given f, and a dimension k) and bound it from above via a new decomposition approach. These results are especially useful when we study $L_{k,f}(t)$ as a function of t, i.e. when we prove that the function $L_{k,f}$ is majorized by a function in a given complexity class. The literature already contains upper bounds on $L_{k,f}$ (see Section 8) but these results are not widely known. Most prominently, McAloon (1984) shows that for linear f, $L_{k,f}$ is primitive-recursive for each fixed k, but is not primitive-recursive when k is not fixed. More precisely, for every k, $L_{k,f}$ is at level $\mathfrak{F}_{k+1}$ of the Fast Growing Hierarchy.[1] To quote Clote (1986), "This suggests the question whether $\mathfrak{F}_{k+1}$ is the best possible."

Our Contribution We present a self-contained and elementary proof, markedly simpler and more general than McAloon's, but yielding an improved upper bound: for linear control functions, $L_{k,f}$ is at level $\mathfrak{F}_k$, and more generally, for a control function f in $\mathfrak{F}_\gamma$, $L_{k,f}$ is at level $\mathfrak{F}_{\gamma+k-1}$.

Example 1.1 (continuing from p. 2)**.** Setting $f(x) = x+1$ makes every sequence of configurations of CHOICE (a, b) a $(\max(a, b))$-controlled bad sequence, for which our results incur an elementary length in $\mathfrak{F}_2$ as a function of $\max(a, b)$. □

That "TIME(a, b) is in $\mathfrak{F}_2$" is a very coarse bound, but as we will see in Section 6, allowing larger dimensions or more complex operations quickly yield huge complexities on very simple programs similar to CHOICE. In fact, we also answer Clote's question, and show that our upper bounds are optimal.

More precisely, our main technical contributions are

- We substantially simplify the problem by considering a richer setting for our analysis: all disjoint unions of powers of $\mathbb{N}$. This lets us provide finer and simpler decompositions of bad sequences (Section 3), from which one extracts upper bounds on their lengths (Section 5.1).
- We completely separate the decomposition issue (from complex to simple wqo's, where f is mostly irrelevant) from the question of locating the bounding function in the Fast Growing Hierarchy (where f becomes relevant); see Section 5.2.
- We obtain new bounds that are essentially tight in terms of the Fast Growing Hierarchy; see Section 6. Furthermore, these bounds are tight even when considering the coarser lexicographic ordering.
- We describe another benefit of our setting: it accomodates in a smooth and easy way an extended notion of bad sequences where the length of the forbidden increasing subsequences is a parameter (Section 4).

In addition we provide (in Section 7) a few examples showing how to use bounds on $L_{k,f}$ in practice. This section is intended as a short "user guide" showing via concrete examples how to apply our main result and derive upper bounds

[1] In truth, McAloon is not that explicit. The $\mathfrak{F}_{k+1}$ upper bound is extracted from his construction by Clote (1986), who also proposed a simple derivation for an upper bound at level $\mathfrak{F}_{k+6}$.

from one's use of Dickson's Lemma. We do not claim that we show new results for these examples, although the existence of the bounds we obtain is hardly known at all. The examples we picked are some of our favorites (many others exist, see Section 8 for a few references). In particular, they involve algorithms or proofs that do not directly deal with bad sequences over $(\mathbb{N}^k, \leq)$:

- programs shown to terminate using *disjunctive termination arguments* (Section 7.1),
- emptiness for *increasing counter automata* with applications to questions for XPath fragments on data words (Section 7.2), and
- *Karp and Miller coverability trees* and their applications, (Section 7.3).

2 WQO's Based on Natural Numbers

The disjoint union, or "sum" for short, of two sets A and B is denoted $A + B$, the sum of an I-indexed family $(A_i)_{i \in I}$ of sets is denoted $\sum_{i \in I} A_i$. While $A + B$ and $\sum_i A_i$ can be seen as, respectively, $A \times \{1\} \cup B \times \{2\}$ and $\bigcup_i A_i \times \{i\}$, we abuse notation and write x when speaking of an element (x, i) of $\sum_i A_i$.

Assume $(A_1, \leq_1)$ and $(A_2, \leq_2)$ are ordered sets. The product $A_1 \times A_2$ is equipped with the usual product ordering: $(x, y) \leq (x', y') \stackrel{\text{def}}{\Leftrightarrow} x \leq_1 x' \wedge y \leq_2 y'$. The sum $A_1 + A_2$ is equipped with the usual sum ordering given by

$$x \leq x' \stackrel{\text{def}}{\Leftrightarrow} \big(x, x' \in A_1 \wedge x \leq_1 x'\big) \vee \big(x, x' \in A_2 \wedge x \leq_2 x'\big)\,.$$

It is easy to see that $(A_1 \times A_2, \leq)$ and $(A_1 + A_2, \leq)$ are wqo's when $(A_1, \leq_1)$ and $(A_2, \leq_2)$ are. This immediately extends to $\prod_{i \in I} A_i$ and $\sum_{i \in I} A_i$ when the index set I is finite. Note that this allows inferring that $(\mathbb{N}^k, \leq)$ is a wqo (Dickson's Lemma) from the fact that $(\mathbb{N}, \leq)$ is.

A key ingredient of this paper is that we consider finite sums of finite powers of $\mathbb{N}$, i.e., sets like, e.g., $2 \times \mathbb{N}^3 + \mathbb{N}$ (or equivalently $\mathbb{N}^3 + \mathbb{N}^3 + \mathbb{N}^1$, and more generally of the form $\sum_{i \in I} \mathbb{N}^{k_i}$). With $S = \sum_{i \in I} \mathbb{N}^{k_i}$, we associate its *type* τ, defined as the multiset $\{k_i \mid i \in I\}$, and let $\mathbb{N}^\tau$ denote S (hence $\mathbb{N}^{\{k\}}$ is $\mathbb{N}^k$ and $\mathbb{N}^\emptyset$ is $\emptyset$).

Types such as τ can be seen from different angles. The multiset point of view has its uses, e.g., when we observe that $\mathbb{N}^{\tau_1} + \mathbb{N}^{\tau_2} = \mathbb{N}^{\tau_1 + \tau_2}$. But types can also be seen as functions $\tau : \mathbb{N} \to \mathbb{N}$ that associate with each power $k \in \mathbb{N}$ its multiplicity $\tau(k)$ in τ. We define the sum $\tau_1 + \tau_2$ of two types with $(\tau_1 + \tau_2)(k) \stackrel{\text{def}}{=} \tau_1(k) + \tau_2(k)$ and its multiple $p \times \tau$, for $p \in \mathbb{N}$, by $(p \times \tau)(k) \stackrel{\text{def}}{=} p.\tau(k)$. As expected, $\tau - \tau_1$ is only defined when τ can be written as some $\tau_1 + \tau_2$, and then one has $\tau - \tau_1 = \tau_2$.

There are two natural ways of comparing types: the inclusion ordering

$$\tau_1 \subseteq \tau_2 \stackrel{\text{def}}{\Leftrightarrow} \exists \tau' : \tau_2 = \tau_1 + \tau' \tag{1}$$

and the multiset ordering defined by transitivity and

$$\tau <_m \{k\} \quad \stackrel{\text{def}}{\Leftrightarrow} \quad k > l \text{ for all } l \in \tau, \tag{2}$$

$$\tau_1 + \tau <_m \tau_2 + \tau \stackrel{\text{def}}{\Leftrightarrow} \tau_1 <_m \tau_2\,. \tag{3}$$

Note how Eq. (2) entails $\emptyset <_m \{k\}$. Then Eq. (3) further yields $\emptyset \leq_m \tau$ for any τ (using transitivity). In fact, the multiset ordering is a well-founded linear extension of the inclusion ordering (see Dershowitz and Manna, 1979). This is the ordering we use when we reason "by induction over types".

3 Long Bad Sequences over $\mathbb{N}^\tau$

Assume a fixed, increasing, control function $f : \mathbb{N} \to \mathbb{N}$ with $f(0) > 0$; we keep f implicit to simplify notations, until Section 5.2 where the choice of control function will become important. For $t \in \mathbb{N}$, we say that a sequence $x_0, x_1, \ldots, x_l$ over $\mathbb{N}^\tau$ is *t-controlled* if $|x_i|_\infty < f(i+t)$ for all $i = 0, 1, \ldots, l$, where $|x_i|_\infty \stackrel{\text{def}}{=} \max\{x_i[j] \mid j = 1, \ldots, dim(x_i)\}$ is the usual infinity norm. Let $L_\tau(t)$ be the length of the longest t-controlled bad sequence over $\mathbb{N}^\tau$.

In simple cases, $L_\tau(t)$ can be evaluated exactly. For example consider $\tau = \{0\}$. Here $\mathbb{N}^\tau$, i.e., $\mathbb{N}^0$, only contains one element, the empty tuple $\langle\rangle$, whose norm is 0, so that every sequence over $\mathbb{N}^\tau$ is t-controlled because $f(0) > 0$, and is good as soon as its length is greater or equal to 2. Hence

$$L_{\{0\}}(t) = 1 \; , \tag{4}$$

and more generally for all $r \geq 1$

$$L_{r \times \{0\}}(t) = r \; . \tag{5}$$

Note that this entails $L_\emptyset(t) = L_{0\times\{0\}}(t) = 0$ as expected: the only sequence over $\mathbb{N}^\emptyset$ is the empty sequence.

The case $\tau = \{1\}$ is a little bit more interesting. A bad sequence $x_0, x_1, \ldots, x_l$ over $\mathbb{N}^{\{1\}}$, i.e., over $\mathbb{N}$, is a decreasing sequence $x_0 > x_1 > \cdots > x_l$ of natural numbers. Assuming that the sequence is t-controlled means that $x_0 < f(t)$. (It is further required that $x_i < f(t+i)$ for every $i = 1, \ldots, l$ but here this brings no additional constraints since f is increasing and the sequence must be decreasing.) It is plain that $L_{\{1\}}(t) \leq f(t)$, and in fact

$$L_{\{1\}}(t) = f(t) \tag{6}$$

since the longest t-controlled bad sequence is exactly

$$f(t)-1,\ f(t)-2,\ \ldots,\ 1,\ 0 \; .$$

Decomposing Bad Sequences over $\mathbb{N}^\tau$ After these initial considerations, we turn to the general case. It is harder to find *exact* formulae for $L_\tau(t)$ that work generally. In this section, we develop inequations providing upper bounds for $L_\tau(t)$ by induction over the structure of τ. These inequations are enough to prove our main theorem.

Assume $\tau = \{k\}$ and consider a t-controlled bad sequence $\mathbf{x} = x_0, x_1, \ldots, x_l$ over $\mathbb{N}^k$. Since $\mathbf{x}$ is t-controlled, x_0 is bounded and $x_0 \leq \langle f(t)-1, \ldots, f(t)-1\rangle$. Now, since $\mathbf{x}$ is bad, every x_i for $i > 0$ must have $x_i[j] < x_0[j]$ for at least one j in $1, \ldots, k$. In other words, every element of the *suffix sequence* $x_1, \ldots, x_l$ belongs to at least one region

$$R_{j,s} = \{x \in \mathbb{N}^k \mid x[j] = s\}$$

for some $1 \le j \le k$ and $0 \le s < f(t) - 1$. The number of regions is

$$N_k(t) \stackrel{\text{def}}{=} k \cdot (f(t) - 1) \; . \tag{7}$$

By putting every x_i in one of the regions, we decompose the suffix sequence into $N_k(t)$ subsequences, some of which may be empty.

We illustrate this on an example. Let $k = 2$ and consider the following bad sequence over $\mathbb{N}^2$

$$\mathbf{x} = \langle 2, 2\rangle,\ \langle 1, 5\rangle,\ \langle 4, 0\rangle,\ \langle 1, 1\rangle,\ \langle 0, 100\rangle,\ \langle 0, 99\rangle,\ \langle 3, 0\rangle \; .$$

The relevant regions are $R_{1,0}$, $R_{1,1}$, $R_{2,0}$, and $R_{2,1}$. We can put $x_3 = \langle 1, 1\rangle$ in either $R_{1,1}$ or $R_{2,1}$, but we have no choice for the other x_j's. Let us put x_3 in $R_{1,1}$; we obtain the following decomposition:

$$\langle 2,2\rangle, \begin{bmatrix} . & . & . & \langle 0,100\rangle, & \langle 0,99\rangle, & . & (R_{1,0} : x[1] = 0) \\ \langle 1,5\rangle, & . & \langle 1,1\rangle, & . & . & . & (R_{1,1} : x[1] = 1) \\ . & \langle 4,0\rangle, & . & . & . & \langle 3,0\rangle & (R_{2,0} : x[2] = 0) \\ . & . & . & . & . & . & (R_{2,1} : x[2] = 1) \end{bmatrix}$$

We have 4 subsequences, one per line. Each subsequence is bad (one is even empty). They are not $(t+1)$-controlled if we see them as *independent* sequences. For instance, the first subsequence, "$\langle 0, 100\rangle, \langle 0, 99\rangle$", is only controlled if $100 < f(t+1)$, while in the original sequence it was only required that $100 < f(t+4)$. But they are $(t + 1)$-controlled if we see them as a sequence over the sum type $4 \times \mathbb{N}^2$.

For the next step, we observe that every subsequence has all its elements sharing a same $x[j] = s$. By disregarding this fixed component, every subsequence can be seen as a bad sequence over $\mathbb{N}^{k-1}$. In our example, we get the following decomposition

$$\langle 2,2\rangle, \begin{bmatrix} . & . & . & \langle *,100\rangle, & \langle *,99\rangle, & . & (R_{1,0} : x[1] = 0) \\ \langle *,5\rangle, & . & \langle *,1\rangle, & . & . & . & (R_{1,1} : x[1] = 1) \\ . & \langle 4,*\rangle, & . & . & . & \langle 3,*\rangle & (R_{2,0} : x[2] = 0) \\ . & . & . & . & . & . & (R_{2,1} : x[2] = 1) \end{bmatrix}$$

This way, the suffix sequence $x_1, \ldots, x_l$ is seen as a bad sequence over $\mathbb{N}^{\tau'}$ for $\tau' \stackrel{\text{def}}{=} N_k(t) \times \{k-1\}$. Note that the decomposition of the suffix sequence always produces a bad, $(t + 1)$-controlled sequence over $\mathbb{N}^{\tau'}$. Hence we conclude that

$$L_{\{k\}}(t) \le 1 + L_{N_k(t)\times\{k-1\}}(t + 1) \; . \tag{8}$$

Observe that Eq. (8) applies even when $k = 1$, giving

$$\begin{aligned} L_{\{1\}}(t) &\le 1 + L_{(f(t)-1)\times\{0\}}(t + 1) \\ &= 1 + f(t) - 1 = f(t) \; . && \text{(by Eq. (5))} \end{aligned}$$

Eq. (8) still applies in the degenerate "$k = 0$" case: here $N_k(t) = 0$ and the meaningless type "$\{-1\}$" is made irrelevant.

Remark 3.1. When $k \ge 2$, the inequality in Eq. (8) cannot be turned into an equality. Indeed, a bad sequence over $N_k(t) \times \mathbb{N}^{k-1}$ cannot always be merged into a bad sequence over $\mathbb{N}^k$. As a generic example, take a bad sequence $\mathbf{x}$ of maximal length over $\mathbb{N}^k$. This sequence ends with $\langle 0, ..., 0\rangle$ (or is not maximal). If we now append another copy of $\langle 0, ..., 0\rangle$ at the end of $\mathbf{x}$, the sequence is not bad anymore. However, when $k \ge 2$ we can decompose its suffix as a bad sequence over $N_k(t) \times \mathbb{N}^{k-1}$ by putting the two final $\langle 0, ..., 0\rangle$'s in the different regions $R_{1,0}$ and $R_{2,0}$. □

The above reasoning, decomposing a sequence over $\mathbb{N}^k$ into a first element and a suffix sequence over $\mathbb{N}^{\tau'}$ for $\tau' = N_k(t) \times \{k-1\}$, applies more generally for decomposing a sequence over an arbitrary $\mathbb{N}^\tau$. Assume $\tau \neq \emptyset$, and let $\mathbf{x} = x_0, x_1, \ldots, x_l$ be a bad sequence over $\mathbb{N}^\tau$. The initial element x_0 of $\mathbf{x}$ belongs to $\mathbb{N}^k$ for some $k \in \tau$ and as above $\mathbf{x}$ can be seen as x_0 followed by a bad subsequence over $\tau' = N_{k,t} \times \{k-1\}$, hence the suffix of $\mathbf{x}$ can be seen as a bad subsequence over $\tau' + (\tau - \{k\})$. This calls for special notations: for $k \geq 1$ in τ and t in $\mathbb{N}$, we let

$$\tau_{\langle k,t\rangle} \stackrel{\text{def}}{=} \tau - \{k\} + N_k(t) \times \{k-1\} \ . \tag{9}$$

Observe that $\tau_{\langle 0,t\rangle}$ is simply $\tau - \{0\}$ since $N_0(t) = 0$.

We can now write down the main consequence of our decomposition:

Theorem 3.2. *For any τ*

$$L_\tau(t) \leq \max_{k\in\tau} \left\{1 + L_{\tau_{\langle k,t\rangle}}(t+1)\right\} .$$

See Appendix A.1 for a formal proof.

The "max" in Theorem 3.2 accounts for allowing a sequence over $\mathbb{N}^\tau$ to begin with a tuple x_0 from any $\mathbb{N}^k$ for $k \in \tau$. As usual, we let $\max\emptyset \stackrel{\text{def}}{=} 0$. Note that this entails $L_\emptyset(t) = 0$, agreeing with Equation 5.

4 Long r-Bad Sequences

We say that sequences with an increasing subsequence $x_{i_1} \leq x_{i_2} \leq \cdots \leq x_{i_{r+1}}$ of length $r+1$ are *r-good* (hence "good" is short for "1-good"). A sequence that is not r-good is r-bad. By Dickson's Lemma, every infinite sequence over $\mathbb{N}^k$ is r-good (for any r), i.e., r-bad sequences are finite. Bounding the length of r-bad sequences is helpful in applications where an algorithm does not stop at the first increasing pair.

Finding a bound on the length of controlled r-bad sequences can elegantly be reduced to the analysis of plain bad sequences, another benefit of our "sum of powers of $\mathbb{N}$" approach.

Write $L_{r,\tau}(t)$ for the maximum length of t-controlled r-bad sequences over $\mathbb{N}^\tau$. In this section we prove the following equality:

$$L_{r,\tau}(t) = L_{r\times\tau}(t) \ . \tag{10}$$

For a sequence $\mathbf{x} = x_0, x_1, \ldots, x_l$ over some $\mathbb{N}^\tau$, an index $i = 0, 1, \ldots, l$ and some $p = 1, \ldots, r$, we say that i is *p-good* if there is an increasing subsequence of length $p+1$ *that starts with* x_i, i.e., some increasing subsequence $x_{i_1} \leq x_{i_2} \leq \cdots \leq x_{i_{p+1}}$ with $i_1 = i$. The *goodness* of index i is the largest p such that i is p-good.

For example, consider the following sequence over $\mathbb{N}^2$

$$\mathbf{x} = \langle 3,1\rangle,\ \langle 5,0\rangle,\ \langle 3,5\rangle,\ \langle 2,4\rangle,\ \langle 2,6\rangle,\ \langle 3,1\rangle,\ \langle 4,5\rangle,\ \langle 2,8\rangle \ .$$

$\mathbf{x}$ can be arranged in layers according to goodness, as in

2-good indices:	$\langle 3,1\rangle$,	.	.	$\langle 2,4\rangle$,	.	.	.	.
1-good indices:	.	.	$\langle 3,5\rangle$,	.	$\langle 2,6\rangle$,	$\langle 3,1\rangle$,	.	.
0-good indices:	.	$\langle 5,0\rangle$,	.	.	.	.	$\langle 4,5\rangle$,	$\langle 2,8\rangle$

This transformation applies to sequences over any wqo. It has two properties:

Badness of layers: Assume that $x_i \leq x_j$ is an increasing pair in $\mathbf{x}$. If x_j is p-good then, by definition, x_i is at least $(p+1)$-good. Hence x_i and x_j cannot be in the same goodness layer and every layer is a bad subsequence of $\mathbf{x}$.

Number of layers: If $\mathbf{x}$ is r-bad, every index i is at most $(r-1)$-good and the decomposition requires at most r non-empty layers.

If we now see the decomposition as transforming a t-controlled r-bad sequence $\mathbf{x}$ over $\mathbb{N}^\tau$ into a sequence $\mathbf{x}'$ over $\mathbb{N}^{r\times\tau}$, then $\mathbf{x}'$ is t-controlled and, as we observed above, bad. Thus

$$L_{r,\tau}(t) \leq L_{r\times\tau}(t) \tag{11}$$

holds in general, proving one half of (10).

For the other half, let $\mathbf{x} = x_0, \ldots, x_l$ be some t-controlled sequence over $\mathbb{N}^{r\times\tau}$. By collapsing $\mathbb{N}^{r\times\tau}$ to $\mathbb{N}^\tau$ in the obvious way, $\mathbf{x}$ can be transformed into a sequence $\mathbf{y}$ over $\mathbb{N}^\tau$. The two sequences have same length and same control. Regarding badness, we can show that $\mathbf{y}$ is r-bad when $\mathbf{x}$ is bad, entailing $l+1 \leq L_{r,\tau}(t)$ and hence

$$L_{r\times\tau}(t) \leq L_{r,\tau}(t) \ . \tag{12}$$

For the proof, assume, by way of contradiction, that $\mathbf{y}$ is not r-bad, i.e., is r-good. Then it contains an increasing subsequence with $r+1$ elements. By the pigeonhole principle, two of these come from the same summand in $r\times\tau$, hence $\mathbf{x}$ contains an increasing pair and is good, contradicting our assumption.

5 Upper Bound

Theorem 3.2 gives a bounding function for L. Define

$$M_\tau(t) \stackrel{\text{def}}{=} \max_{k\in\tau}\left\{1 + M_{\tau_{\langle k,t\rangle}}(t+1)\right\} \ . \tag{13}$$

This inductive definition is well-formed since $\tau_{\langle k,t\rangle} <_m \tau$ and the multiset ordering is well-founded. Note that $M_\emptyset(t) = 0$ since $\max\emptyset = 0$. For all τ and t, it holds that $L_\tau(t) \leq M_\tau(t)$.

We first show that the maximum in Eq. (13) is reached by always choosing the smallest element of τ (Section 5.1), and then use this characterization to classify M in the Fast Growing Hierarchy (Section 5.2).

5.1 A Maximizing Strategy for M

The next Lemma shows that the maximum of all $1 + M_{\tau_{\langle k,t\rangle}}(t+1)$ used in Eq. (13) can always be obtained by taking $k = \min\tau$. This useful fact leads to a simplified definition of M.

Lemma 5.1 (see Appendix A.2)**.** *Let $k = \min\tau$ and $l \in \tau$. Then $M_{\tau_{\langle l,t\rangle}}(t+1) \leq M_{\tau_{\langle k,t\rangle}}(t+1)$ and, hence,*

$$\begin{aligned} &M_\emptyset(t) = 0 \\ &M_\tau(t) = 1 + M_{\tau_{\langle \min\tau,t\rangle}}(t+1) \quad \text{for } \tau \neq \emptyset \ . \end{aligned}$$

5.2 Classifying M in the Fast Growing Hierarchy

The bounding function M_τ grows very fast with the dimension k: $M_{\{3\}}$ is already non-elementary for $f(x) = 2x + 1$. Clote (1986) classified the upper bounds derived from both his construction and that of McAloon using the *Fast Growing Hierarchy* $(\mathfrak{F}_\alpha)_\alpha$ (Löb and Wainer, 1970) for finite ordinals α: for a linear control function, he claimed his bounding function to reside at the $\mathfrak{F}_{k+6}$ level, and McAloon's at the $\mathfrak{F}_{k+1}$ level. We show in this section a bounding function in $\mathfrak{F}_k$; the results of the next section entail that this is optimal, since we can find a lower bound for $L_{r\times\{k\}}$ which resides in $\mathfrak{F}_k \backslash \mathfrak{F}_{k-1}$ if $k \geq 2$.

The Fast Growing Hierarchy The class $\mathfrak{F}_k$ of the Fast Growing Hierarchy is the closure under substitution and limited recursion of the constant, sum, projections, and F_n functions for $n \leq k$, where F_n is defined recursively by[2]

$$F_0(x) \stackrel{\text{def}}{=} x + 1 \tag{14}$$

$$F_{n+1}(x) \stackrel{\text{def}}{=} F_n^{x+1}(x) \; . \tag{15}$$

The hierarchy is strict for $k \geq 1$, i.e. $\mathfrak{F}_k \subsetneq \mathfrak{F}_{k+1}$, because $F_{k+1} \notin \mathfrak{F}_k$. For small values of k, the hierarchy characterizes some well-known classes of functions:

- $\mathfrak{F}_0 = \mathfrak{F}_1$ contains all the linear functions, like $\lambda x.x + 3$ or $\lambda x.2x$,
- $\mathfrak{F}_2$ contains all the elementary functions, like $\lambda x.2^{2^x}$,
- $\mathfrak{F}_3$ contains all the tetration functions, like $\lambda x. \underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{x \text{ times}}$, etc.

The union $\bigcup_k \mathfrak{F}_k$ is the set of 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 $\mathfrak{F}_\omega \backslash \bigcup_k \mathfrak{F}_k$. Some further intuition on the relationship between the functions f in $\mathfrak{F}_k$ and F_k for $k \geq 1$ can be gained from the following fact: for each such f, there exists a finite p s.t. F_k^p majorizes f, i.e. for all $x_1, \ldots, x_n$, $f(x_1, \ldots, x_n) < F_k^p(\max(x_1, \ldots, x_n))$ (Löb and Wainer, 1970, Theorem 2.10).

Readers might be more accustomed to a variant $(A_k)_k$ of the $(F_k)_k$ called the *Ackermann Hierarchy* (see e.g. Friedman, 2001), and defined by

$$A_1(x) \stackrel{\text{def}}{=} 2x$$

$$A_{k+1}(x) \stackrel{\text{def}}{=} A_k^x(1) \text{ for } k \geq 1 \; .$$

These versions of the Ackermann functions correspond exactly to exponentiation of 2 and tetration of 2 for $k = 2$ and $k = 3$ respectively. One can check that for all $k, p \geq 1$, there exists $x_{k,p} \geq 0$ s.t., for all $x \geq x_{k,p}$, $A_k(x) > F_{k-1}^p(x)$, which contradicts A_k being in $\mathfrak{F}_{k-1}$ by (Löb and Wainer, 1970, Theorem 2.10). Conversely, $A_k(x) \leq F_k(x)$ for all $k \geq 1$ and $x \geq 0$, which shows that A_k belongs to $\mathfrak{F}_k \backslash \mathfrak{F}_{k-1}$ for $k \geq 2$.

[2] For simplicity's sake, we present here a version more customary in the recent literature, including McAloon (1984) and Clote (1986). Note however that it introduces a corner case at level 1: in Löb and Wainer (1970), $\mathfrak{F}_0 \subsetneq \mathfrak{F}_1$, the latter being the set of polynomial functions, generated by $F_1(x) \stackrel{\text{def}}{=} (x + 1)^2$.

$$\underbrace{f(t)-1\; f(t)-1\;\cdots\; f(t)-1}_{\ell_{k,f}(t)\text{ times}}\quad \underbrace{f(t)-2,\; f(t)-2,\;\cdots,\; f(t)-2}_{\ell_{k,f}(o_{k,f}(t))\text{ times}}\quad\cdots\quad \underbrace{0,\;0,\;\cdots,\;0}_{\ell_{k,f}\left(o_{k,f}^{f(t)-1}(t)\right)\text{ times}}$$

Figure 1: The decomposition of bad sequences for the lexicographic ordering.

Main Result In this section and in the following one, we focus on classifying in the Fast Growing Hierarchy the function $M_{r\times\{k\}}$ for some fixed r, k, and (implicit) f. Here the choice for the control function f becomes critical, and we prefer therefore the explicit notation $M_{r\times\{k\},f}$.

The main result of this section is then

Proposition 5.2 (see Appendix A.3)**.** *Let $k, r \geq 1$ be natural numbers and $\gamma \geq 1$ an ordinal. If f is a monotone unary function of $\mathfrak{F}_\gamma$ with $f(x) \geq \max(1, x)$ for all x, then $M_{r\times\{k\},f}$ is in $\mathfrak{F}_{\gamma+k-1}$.*

One can be more general in the comparison with McAloon's proof: his Main Lemma provides an upper bound of form $G'_{k,f}(d \cdot f(x)^2)$ for some constant d, where in turn his $G'_{k,f}$ function can be shown to be bounded above by a function in $\mathfrak{F}_{\gamma+k+1}$ when f is in $\mathfrak{F}_\gamma$. The $\mathfrak{F}_{k+1}$ bound for linear functions reported by Clote (1986) is the result of a specific analysis in McAloon's Main Corollary.

6 Lower Bound

We prove in this section that the upper bound of $\mathfrak{F}_{\gamma+k-1}$ for a control function f in $\mathfrak{F}_\gamma$ is tight if f grows fast enough.

Let $\leq_{\text{lex}}$ denote the lexicographic ordering over $\mathbb{N}^k$, defined by

$$\begin{aligned} x = \langle x[1], \ldots, x[k]\rangle \leq_{\text{lex}} y = \langle y[1], \ldots, y[k]\rangle &\overset{\text{def}}{\Leftrightarrow} x[1] \leq y[1] \\ &\vee (x[1] = y[1] \wedge \langle x[2], \ldots, x[k]\rangle \leq_{\text{lex}} \langle y[2], \ldots, y[k]\rangle)\ . \end{aligned}$$

This is a well linear ordering for finite k values, and is coarser than the natural product ordering. Let us fix a control function f; we denote by $\ell_{r,k,f}(t)$ the length of the longest t-controlled r-bad sequence for $\leq_{\text{lex}}$ on $\mathbb{N}^k$: this implies that for all t

$$\ell_{r,k,f}(t) \leq L_{r\times\{k\},f}(t)\ . \tag{16}$$

We derive in this section an *exact* inductive definition for ℓ in the case $r = 1$, and show that it yields large enough lower bounds for L in the case of $f = F_\gamma$.

An Inductive Definition for ℓ We define our strategy for generating the longest bad controlled sequence for $\leq_{\text{lex}}$ in $\mathbb{N}^k$ by induction on k. Assume as usual $f(0) > 0$; for $k = 1$, the longest t-controlled sequence is

$$f(t) - 1,\ f(t) - 2,\ \ldots,\ 1,\ 0$$

of length $f(t)$, and we define

$$\ell_{1,f}(t) = f(t)\ . \tag{17}$$

In dimension $k+1$, we consider the bad sequence where the projection on the first coordinate is segmented into $f(t)$ constant sections, such that the projection on the k remaining coordinates of each section is itself a bad sequence of dimension k following the same strategy.

Example 6.1. The sequence built by our strategy for $k=2$, $t=3$, and $f(x)=x+1$ is

i	0	1	2	3	4	5	$\cdots$	10	11	12	13	$\cdots$	26	27	28	29	$\cdots$	58	59
$x_i[1]$	3	3	3	3	2	2	$\cdots$	2	2	1	1	$\cdots$	1	1	0	0	$\cdots$	0	0
$x_i[2]$	3	2	1	0	7	6	$\cdots$	1	0	15	14	$\cdots$	1	0	31	30	$\cdots$	1	0
$f(i+t)$	4	5	6	7	8	9	$\cdots$	14	15	16	17	$\cdots$	30	31	32	33	$\cdots$	62	63

It is composed of four sections, one for each value of the first coordinate. The first section starts at $i=0$ and is of length $\ell_{1,f}(3)=4$, the second starts at $i=4$ and is of length $\ell_{1,f}(7)=8$, the third at $i=12$ with length $\ell_{1,f}(15)=16$, and the last at $i=28$ with length $\ell_{1,f}(31)=32$. The successive arguments of $\ell_{1,f}$ can be decomposed as sums $t+\ell_{1,f}(t)$ for the previously computed argument t:

$$\begin{aligned}
7 &= 3+4 = 3+\ell_{1,f}(3)\\
15 &= 7+8 = 7+\ell_{1,f}(7)\\
31 &= 15+16 = 16+\ell_{1,f}(15)
\end{aligned}$$

simply because at each step the starting index is increased by the length of the previous section. □

We define accordingly an offset function o by

$$o_{k,f}(t) \stackrel{\text{def}}{=} t+\ell_{k,f}(t)\;; \tag{18}$$

the strategy results in general in a sequence of the form displayed in Figure 1 on the first coordinate. The obtained sequence is clearly bad for $\leq_{\text{lex}}$; that it is the longest such sequence is also rather straightforward by induction: each segment of our decomposition is maximal by induction hypothesis, and we combine them using the maximal possible offsets. Hence

$$\ell_{k+1,f}(t)=\sum_{j=1}^{f(t)}\ell_{k,f}\left(o_{k,f}^{j-1}(t)\right)\;. \tag{19}$$

Remark 6.2. The lexicographic ordering really yields shorter bad sequences than the product ordering, i.e. we can have $\ell_{k,f}(t)<L_{\{k\},f}(t)$, as can be witnessed by the two following sequences for $f(x)=2x$ and $t=1$, which are bad for $\leq_{\text{lex}}$ and $\leq$ respectively:

$$\begin{aligned}
&\langle 1,1\rangle,\langle 1,0\rangle,\langle 0,5\rangle,\langle 0,4\rangle,\langle 0,3\rangle,\langle 0,2\rangle,\langle 0,1\rangle,\ \langle 0,0\rangle\\
&\langle 1,1\rangle,\langle 0,3\rangle,\langle 0,2\rangle,\langle 0,1\rangle,\langle 9,0\rangle,\langle 8,0\rangle,\langle 7,0\rangle,\langle 6,0\rangle,\langle 5,0\rangle,\ldots,\langle 0,0\rangle
\end{aligned}$$

The first sequence, of length $8=\ell_{2,f}(1)$, is maximal for $\leq_{\text{lex}}$, and shorter than the second, of length $14\leq L_{\{2\},f}(1)$. □

Lower Bound for r-Bad Sequences One can further extend this strategy to give a lower bound on the length of interleavings of r-bad sequences in $\mathbb{N}^k$, by simply concatenating r sequences, each starting with a higher offset. For instance, for $r = 2$, start with the sequence of length $\ell_{k,f}(t)$; arrived at this point, the next sequence reaches length $\ell_{k,f}(t + \ell_{k,f}(t))$. In general

$$\ell_{r,k,f}(t) \geq \sum_{j=1}^{r} \ell_{k,f}\left(o_{k,f}^{j-1}(t)\right) \ . \tag{20}$$

Proposition 6.3 (see Appendix A.4)**.** *Let $\gamma \geq 0$ be an ordinal and $k, r \geq 1$ natural numbers. Then, for all $t \geq 0$, $\ell_{r,k,F_\gamma}(t) \geq F^r_{\gamma+k-1}(t)$.*

Remark 6.4. Note that, since

$$\ell_{r,k,F_\gamma}(t) \leq L_{r\times\{k\},F_\gamma}(t) \leq M_{r\times\{k\},F_\gamma}(t) \ ,$$

Proposition 5.2 and Proposition 6.3 together show that $M_{r\times\{k\},F_\gamma}$ belongs to $\mathfrak{F}_{\gamma+k-1}\backslash\mathfrak{F}_{\gamma+k-2}$ if $\gamma \geq 1$ and $\gamma + k \geq 3$. One can see that the same holds for ℓ_{k,F_γ}, since it is defined by limited primitive recursion. □

Remark 6.5. In the case of the successor control function $f = F_0$, the F_{k-1} lower bound provided by Proposition 6.3 does not match the $\mathfrak{F}_k$ upper bound of Proposition 5.2 (indeed the statement of the latter does not allow $\gamma = 0$ and forces $\gamma = 1$). Tightness holds nevertheless, since Friedman (2001) proved in his Theorem 2.6 an A_k lower bound for this particular case of $f = F_0$. □

Concrete Example It is easy to derive a concrete program illustrating the intuition behind Proposition 6.3:

Example 6.6. Consider the following program with control $\lambda x.2^x + 1$ in $\mathfrak{F}_2$ for $t = \lceil \log_2 \max_{1\leq j\leq k} a_j \rceil$:

LEX $(a_1, \ldots, a_k)$
$c \longleftarrow 1$
while $\bigwedge_{1\leq j\leq k} a_j > 0$
  $\langle a_1, a_2, \ldots, a_{k-1}, a_k, c\rangle \longleftarrow \langle a_1 - 1, a_2, \ldots, a_{k-1}, a_k, 2c\rangle$
 or
  $\langle a_1, a_2, \ldots, a_{k-1}, a_k, c\rangle \longleftarrow \langle 2c, a_2 - 1, \ldots, a_{k-1}, a_k, 2c\rangle$
 or
  ⋮
 or
  $\langle a_1, a_2, \ldots, a_{k-1}, a_k, c\rangle \longleftarrow \langle 2c, 2c, \ldots, 2c, a_k - 1, 2c\rangle$
end

An analysis similar to that of $\ell_{k,f}$ shows that, for $k \geq 2$ and $m = \min_{1\leq j\leq k} a_j > 0$, LEX might run through its **while** loop more than $A_{k+1}(m)$ times, which is a function in $\mathfrak{F}_{k+1}\backslash\mathfrak{F}_k$. It matches the $\mathfrak{F}_{k+1}$ upper bound provided by Proposition 5.2 for this program, since the projection of any sequence of program configurations $\langle a_1, \ldots, a_k, c\rangle$ on the k first components is bad (c increases continuously and thus does not contribute to the sequence being bad). □

7 Applications

We observe that results on the length of bad sequences, like our Proposition 5.2, are rarely used in the verification literature. In this section we argue that Proposition 5.2 is very easy to use when one seeks complexity upper bounds, at least if one is content with the somewhat coarse bounds provided by the Fast Growing Hierarchy.

One might want to modify the choices of parametrization we made out of technical convenience: for instance

- controlling the sum of the vector components instead of their infinity norm, i.e. asking that $\sum_j x_i[j] < f(i+t)$: since $|x_i|_\infty \leq \sum_j x_i[j]$, Proposition 5.2 also works for this definition of control,
- controlling the bitsize of the successive vectors in a bad sequence similarly only induces a jump in the classification of f from $\mathfrak{F}_1$ to $\mathfrak{F}_2$ and leaves the other cases unchanged,
- using an "internal" view of the control, constraining how much the vector components can grow in the course of a single step of the algorithm, i.e. such that $|x_i|_\infty < f^i(t)$, leads to upper bounds one level higher in the Fast Growing Hierarchy, since $\lambda i.f^{i+1}(t)$ controls the sequence in our sense and belongs to $\mathfrak{F}_{\gamma+1}$ whenever f belongs to $\mathfrak{F}_\gamma$.

7.1 Disjunctive Termination Arguments

Program termination proofs essentially establish that the program's transition relation R is well-founded. The classical, "monolithic" way of proving well-foundedness is to exhibit a *ranking* function ρ from the set of program configurations $x_0, x_1, \ldots$ into a well-order such that $R \subseteq \{(x_i, x_j) \mid \rho(x_i) \not\leq \rho(x_j)\}$, like $\lambda a_1 \cdots a_k c.(\sum_{k\geq j\geq 1} \omega^{j-1}\cdot a_j)$, mapping $\mathbb{N}^{k+1}$ to ω^k for Example 6.6. That same ranking function could also be seen as mapping to $(\mathbb{N}^k, \leq_{\text{lex}})$, a linear extension of the product ordering. Our techniques easily apply to such termination proofs based on lexicographic orderings: one only needs to identify a control function. This is usually obtained by combining the computational complexities of the program operations and of the ranking function.

A different termination argument was proposed by Podelski and Rybalchenko (2004) (see also (Blass and Gurevich, 2008) for further details and an overview of earlier work on the subject): in order to prove R to be well-founded, they rather exhibit a finite set of well-founded relations $T_1, \ldots, T_k$ and prove that $R^+ \subseteq T_1 \cup \cdots \cup T_k$. In practice, each of the T_j, $1 \leq j \leq k$, is proved well-founded through a ranking function ρ_j, but these functions might be considerably simpler than a monolithic ranking function. In the case of Example 6.6, choosing $T_j = \{(\langle a_1, \ldots, a_j, \ldots, a_k, c\rangle, \langle a'_1, \ldots, a'_j, \ldots, a'_k, c'\rangle) \mid a_j > 0 \wedge a'_j < a_j\}$, yields such a *disjunctive termination argument*.

Although Podelski and Rybalchenko resort to Ramsey's Theorem in their termination proof, we can easily derive an alternative proof from Dickson's Lemma, which allows us to apply our results: if each of the T_j is proven well-founded thanks to a mapping ρ_j into some wqo $(X_j, \leq_j)$, then with a sequence $x_0, x_1, \ldots$ of program configurations one can associate the sequence of tuples $\langle \rho_1(x_0), \ldots, \rho_k(x_0)\rangle, \langle \rho_1(x_1), \ldots, \rho_k(x_1)\rangle, \ldots$ in $X_1 \times \cdots \times X_k$, the latter being a wqo for the product ordering by Dickson's Lemma. Since for any indices $i_1 < i_2$, $(x_{i_1}, x_{i_2}) \in R^+$ is in some T_j for some $1 \leq j \leq k$, we have $\rho_j(x_{i_1}) \not\leq_j \rho_j(x_{i_2})$ by definition of a ranking function. Therefore the sequence of tuples is bad for the product ordering and thus finite, and the program terminates.

If the range of the ranking functions is $\mathbb{N}$, one merely needs to provide a control on the ranks $\rho_j(x_i)$, i.e. on the composition of R^i with ρ_j, in order to apply Proposition 5.2. For instance, for all programs consisting of a loop with variables ranging over $\mathbb{Z}$ and updates of linear complexity (like CHOICE or LEX), Bradley et al. (2005) synthesize linear ranking functions into $\mathbb{N}$:

Question 7.1. What is the complexity of loop programs with linear operations proved terminating thanks to a k-ary disjunctive termination argument that uses linear ranking functions into $\mathbb{N}$?

The control on the ranks in such programs is at most exponential (due to the iteration of the loop) in $\mathfrak{F}_2$. With Proposition 5.2 one obtains an upper bound in $\mathfrak{F}_{k+1}$ on the maximal number of loop iterations (i.e., the running time of the program), where k is the number of transition invariants $T_1, \ldots, T_k$ used in the termination proof—in fact we could replace "linear" by "polynomial" in Question 7.1 and still provide the same answer. Example 6.6 shows this upper bound to be tight. Unsurprisingly, our bounds directly relate the complexity of programs with the number of disjunctive termination arguments required to prove their termination.

7.2 Reachability for Incrementing Counter Automata

Incrementing Counter Automata, or ICA's, are Minsky counter machines with a modified operational semantics (see Demri, 2006; Demri and Lazić, 2009). ICA's have proved useful for deciding logics on data words and data trees, like XPATH fragments (Figueira and Segoufin, 2009). The fundamental result in this area is that, for ICA's, the set of reachable configurations is a computable set (Mayr, 2003; Schnoebelen, 2010b).

Here we only introduce a few definitions and notations that are essential to our development (and refer to (Mayr, 2003; Schnoebelen, 2010b) for more details). The configuration of a k-counter machine $M = (Q, \Delta)$ is some tuple $v = \langle q, a_1, \ldots, a_k \rangle$ where q is a control-state from the finite set Q, and $a_1, \ldots, a_k \in \mathbb{N}$ are the current values of the k counters. Hence $Conf_M \stackrel{\text{def}}{=} Q \times \mathbb{N}^k$. The transitions between the configurations of M are obtained from its rules (in Δ). Now, whenever M seen as a Minsky machine has a transition $\langle q, a_1, \ldots, a_k \rangle \to_{\mathrm{M}} \langle p, b_1, \ldots, b_k \rangle$, the same M seen as an ICA has all transitions $\langle q, a_1, \ldots, a_k \rangle \to_{\mathrm{I}} \langle p, b'_1, \ldots, b'_k \rangle$ for $b'_1 \geq b_1 \wedge \cdots \wedge b'_k \geq b_k$: Informally, an ICA behaves as its underlying Minsky machine, except that counters may increment spuriously after each step. The consequence is that, if we order $Conf_M$ with the standard partial ordering (by seeing $Conf_M$ as the wqo $\sum_{q \in Q} \mathbb{N}^k$), then the reachability set of an ICA is upward-closed.

We now describe the forward-saturation algorithm that computes the reachability set from an initial configuration v_0.

Let $X_0, X_1, X_2, \ldots$ and $Y_0, Y_1, Y_2, \ldots$ be the sequences of subsets of $Conf_M$

defined by

$$X_0 \stackrel{\text{def}}{=} \{v_0\}, \qquad\qquad X_{i+1} \stackrel{\text{def}}{=} \mathit{Post}(X_i),$$
$$Y_0 \stackrel{\text{def}}{=} X_0, \qquad\qquad Y_{i+1} \stackrel{\text{def}}{=} Y_i \cup X_{i+1},$$

where $\mathit{Post}(X) \stackrel{\text{def}}{=} \{v' \in \mathit{Conf}_M \mid \exists v \in X : v \to_{\mathrm{I}} v'\}$. The reachability set is $\mathit{Reach}(M, v_0) \stackrel{\text{def}}{=} \bigcup_{i=1,2,\ldots} X_i$, i.e., $\lim_{i\to\omega} Y_i$. However, since every X_{i+1} is upward-closed, the sequence $(Y_i)_{i\in\mathbb{N}}$ stabilizes after finitely many steps, i.e., there is some l such that $Y_l = Y_{l+1} = \cdots = \mathit{Reach}(M, v_0)$.This method is effective once we represent (infinite) upward-closed sets by their finitely many minimal elements: it is easy to compute the minimal elements of X_{i+1} from the minimal elements of X_i, hence one can build the sequence $Y_0, Y_1, \ldots$ (again represented by minimal elements) until stabilization is detected.

Question 7.2. What is the computational complexity of the above forward-saturation algorithm for ICA's?

For this question, we start with the length of the sequence $Y_0 \subsetneq Y_1 \subsetneq Y_2 \subsetneq \cdots \subsetneq Y_l = Y_{l+1}$. For each $i = 1, \ldots, l$, let v_i be a minimal element in $Y_i \setminus Y_{i-1}$ (a *nonempty* subset of Conf_M). Note that $v_i \in X_i$, an upward-closed set, so that Y_i contains all configurations above v_i. Hence $v_j \not\geq v_i$ for $j > i$ (since $v_j \notin Y_i$) and the sequence $\mathbf{v} = v_1, v_2, \ldots$ is bad—this also proves the termination of the $(Y_i)_i$ sequence.

We now need to know how $\mathbf{v}$ is controlled. Consider a minimal element v of Y_i. Then $|v|_\infty \leq i + |v_0|_\infty$, which means that $\mathbf{v}$ is $|v_0|_\infty$-controlled for $f = F_0$ the successor function: We observe that f is independent of the ICA M at hand! Using Proposition 5.2 we conclude that, for fixed k, l is bounded by a function in $\mathfrak{F}_k$ with $|v_0|_\infty$ as argument. Now, computing X_{i+1} and Y_{i+1} (assuming representation by minimal elements) can be done in time linear in $|X_i|$ and $|Y_i|$ (and $|M|$ and $|v_0|_\infty$), so that the running time of the algorithm is in $O(|M| \cdot l)$, i.e., also in $\mathfrak{F}_k$ (see Schnoebelen, 2010a, for F_{k-2} lower bounds for the reachability problem in k-dimensional ICA's).

Here the main parameter in the complexity is the number k of counters, not the size of Q or the number of rules in M. For fixed k the complexity is primitive-recursive, and it is Ackermannian when k is part of the input—which is the case in the encoding of logical formulæ of Demri and Lazić (2009).

7.3 Coverings for Vector Addition Systems

Vector addition systems (VAS's) are systems where k counters evolve by non-deterministically applying k-dimensional translations from a fixed set. They can be seen as an abstract presentation of Petri nets, and are thus widely used to model concurrent systems, reactive systems with resources, etc.

Formally, an k-dimensional VAS is some $S = (\Delta, v_0)$ where $v_0 \in \mathbb{N}^k$ is an *initial configuration* and $\Delta \subseteq \mathbb{Z}^k$ is a finite set of *translations*. Unlike translations, configurations only contain non-negative values. A VAS S has a step $v \xrightarrow{\delta} v'$ whenever $\delta \in \Delta$ and $v + \delta \in \mathbb{N}^k$: we then have $v' = v + \delta$. Hence the negative values in δ are used to decrement the corresponding counters *on the condition that they do not become negative*, and the positive values are used to increment the other counters. A configuration v is reachable, denoted $v \in \mathit{Reach}(S)$, if there exists a sequence $v_0 \xrightarrow{\delta_1} v_1 \xrightarrow{\delta_2} v_2 \cdots \xrightarrow{\delta_n} v_n = v$. That reachability is decidable for VAS's is a major result of computer science but we are concerned here with computing a *covering* of the reachability set.

In order to define what is a "covering", we consider the completion $\mathbb{N}_\omega \stackrel{\text{def}}{=} \mathbb{N} \cup \{\omega\}$ of $\mathbb{N}$ and equip it with the obvious ordering. Tuples $w \in \mathbb{N}_\omega^k$, called *$\omega$-markings*, are ordered with the product ordering. While ω-markings are not proper configurations, it is convenient to extend the notion of steps and write $w \xrightarrow{\delta} w'$ when $w' = w + \delta$ (assuming $n + \omega = \omega$ for all n).

Let $C \subseteq \mathbb{N}_\omega^k$ be a set of ω-markings. We say that C is a *covering for S* if for any $v \in \mathit{Reach}(S)$, C contains some w with $v \leq w$, while any $w \in C$ is in the adherence of the reachability set, i.e., $w = \lim_{i=1,2,\ldots} v_i$ for some markings $v_1, v_2, \ldots$ in $\mathit{Reach}(S)$. Hence a covering is a rather precise approximation of the reachability set (precisely, the adherence of its downward-closure). A fundamental result is that *finite* coverings always exist and are computable. This entails several decidability results, e.g. whether a counter value remains bounded throughout all the possible runs.

A particular covering of S can be obtained from the KM tree,[3] introduced by Karp and Miller (1969). Formally, this tree has nodes labeled with ω-markings and edges labeled with translations. The root s_0 is labeled with v_0 and the tree is grown in the following way: Assume a node s of the tree is labeled with some w and let $(v_0 =)w_0, w_1, ..., w_n = w$ be the labels on the path from the root to s. For any translation $\delta \in \Delta$ such that there is a step $w \xrightarrow{\delta} w'$, we consider whether to grow the tree by adding a child node s' to s with a δ-labeled edge from s to s'.

1. If $w' \leq w_i$ for one of the w_i's on the path from s_0 to s, we do not add s' (the branch ends).
2. Otherwise, if $w' > w_i$ for some $i = 0, \ldots, n$, we build w'' from w' by setting, for all $j = 1, \ldots, k$, $w''[j] \stackrel{\text{def}}{=} \omega$ whenever $w'[j] > w_i[j]$, otherwise $w''[j]$ is just $w'[j]$. Formally, w'' can be thought as "$w_i + \omega \times (w' - w_i)$". We add s', the edge from s to s', and we label s' with w''.
3. Otherwise, w' is not comparable with any w_i: we simply add the edge and label s' with w'.

Theorem 7.3 ((Karp and Miller, 1969))**.** *The above algorithm terminates and the set of labels in the KM tree is a covering for S.*

Question 7.4. What is the complexity of the KM algorithm? What is the size of the KM tree? And the size of C?

Answering the above question requires understanding why the KM algorithm terminates. First observe that the KM tree is finitely branching (a node has at most $|\Delta|$ children), thus the tree can only be infinite by having an infinite branch (Kőnig's Lemma). Assume, for the sake of contradiction, that there is an infinite branch labeled by some $w_0, w_1, \ldots$ The sequence may be a good sequence, but any increasing pair $w_{i_1} \leq w_{i_2}$ requires w_{i_2} to be inserted at step 2

[3] The computation of the KM tree has other uses, e.g., with the finite containment problem Mayr and Meyer (1981). Results from Mayr and Meyer (1981) show Ackermanian lower bounds, and provided the initial motivation for the work of McAloon (1984) and Clote (1986).

of the KM algorithm. Hence w_{i_2} has more ω's than w_{i_1}. Finally, since an ω-marking has at most k ω's, the sequence is $(k+1)$-bad and cannot be infinite since $\mathbb{N}_\omega^k$ is a wqo.

Now, how is the sequence controlled? If we say that the ω's do not count in the size of an ω-marking, a branch $w_0, w_1, \ldots$ of the KM tree has $|w_{i+1}|_\infty \leq |w_i|_\infty + |\Delta| \leq |v_0|_\infty + i \cdot |\Delta|$. Hence the sequence is $|v_0|_\infty$-controlled for $f(x) = x \cdot |\Delta| + 1$, a control at level $\mathfrak{F}_1$ for fixed Δ. More coarsely, the sequence is $|S|$-controlled for a *fixed* $f(x) = x^2$, this time at level $\mathfrak{F}_2$. By Proposition 5.2 and Eq. (10), we deduce that the length of any branch is less than $l_{\max} = L_{(k+1)\times\{k\}}(|S|)$. The size of the KM tree, and of the resulting C, is bounded by $|\Delta|^{l_{\max}}$. Finally, the time complexity of the KM algorithm on k-dimensional VAS's
is in $\mathfrak{F}_{k+1}$: the complexity is primitive-recursive for fixed dimensions, but Ackermannian when k is part of the input.

The above result on the size of KM trees can be compared with the tight bounds that Howell et al. show for VAS's (Howell et al., 1986, Theorem 2.8). Their $\mathfrak{F}_{k-1}$ bound is two levels better than ours. It only applies to KM trees and is obtained via a rather complex analysis of the behaviour of VAS's, not a generic analysis of Dickson's Lemma. In particular it does not apply to VAS extensions, while our complexity analysis carries over to many classes of well-structured counter systems, like the strongly increasing affine nets of Finkel et al. (2004), for which both the KM tree algorithm and a $\mathfrak{F}_2$ control keep applying, and thus so does the $\mathfrak{F}_{k+1}$ bound.

8 Related Work

Bounds for $\mathbb{N}^k$ We are not the first ones to study the length of controlled bad sequences. Regarding Dickson's Lemma, both McAloon (1984) and Clote (1986) employ *large intervals* in a sequence and their associated Ramsey theory, showing that large enough intervals would result in good sequences. Unlike our elementary argument based on disjoint sums, we feel that the combinatorial aspects of McAloon's approach are rather complex, whereas the arguments of Clote rely on a long analysis performed by Ketonen and Solovay (1981) and is not parametrized by the control function f. Furthermore, as already mentioned on several occasions, both proofs result in coarser upper bounds. Friedman (2001) also shows that bad sequences over $\mathbb{N}^k$ are primitive-recursive in his Theorem 6.2, but the proof is given for the specific case of the successor function as control, and does not distinguish the dimension k as a parameter. One could also see the results of Howell et al. (1986) or Hofbauer (1992) as implicitly providing bounds on the bad sequences that can be generated resp. by VAS's and certain terminating rewrite systems; using these bounds for different problems can be cumbersome, since not only the control complexity is fixed, but it also needs to be expressed in the formal system at hand.

Beyond $\mathbb{N}^k$ Bounds on bad sequences for other wqo's have also been considered; notably Cichoń and Tahhan Bittar (1998) provide bounds for finite sequences with the embedding order (Higman's Lemma). Their bounds use a rather complex ordinal-indexed hierarchy. If we only consider tuples of natural numbers, their decomposition also reduces inductively from $\mathbb{N}^k$ to $\mathbb{N}^{k-1}$, but it uses the "badness" parameter (r, see Section 4) as a useful tool, as witnessed by their exact analysis of $L_{r,1,f}$. For arbitrary $k \in \mathbb{N}$, Cichoń and Tahhan Bittar have an elegant decomposition, somewhat similar to the large interval approach, that bounds $L_{r,k,f}$ by some $L_{r',k-1,f'}$ for some r' and f' obtained from r, f and k. However, r' and f', r'' and f'', ..., quickly grow very complex, and how to classify the resulting bounds in the Fast Growing Hierarchy is not very clear to us. By contrast, our approach lets us keep the same fixed control function f at all steps in our decomposition.

Weiermann proves another bound for Higman's Lemma (Weiermann, 1994, Corollary 6.3), but his main focus is actually to obtain bounds for Kruskal's Theorem (Weiermann, 1994, Corollary 6.4), i.e. for finite trees with the embedding ordering. The bounds are, as expected, very high, and only consider polynomial ranking functions.

Further Pointers The question of extracting complexity upper bounds from the use of Dickson's Lemma can be seen as an instance of a more general concern stated by Kreisel: "What more than its truth do we know if we have a proof of a theorem in a given formal system?" Our work fits in the field of implicit computational complexity in a broad sense, which employs techniques from linear logic, lambda calculus and typing, invariant synthesis, term rewriting, etc. that entail complexity properties. In most cases however, the scope of these techniques is very different, as the complexity classes under study are quite low, with PTime being the main object of focus (e.g. Leivant, 2002; Gulwani, 2009; Hoffmann and Hofmann, 2010, etc.). By contrast, our technnique is of limited interest for such low complexities, as the Fast Growing Hierarchy only provides very coarse bounds. But it is well suited for the very large complexities of many algorithmic issues, for well-structured transition systems (Finkel and Schnoebelen, 2001) working on tuples of naturals, Petri nets equivalences (Mayr and Meyer, 1981), Datalog with constraints (Revesz, 1993), Gröbner's bases (Gallo and Mishra, 1994), relevance logics (Urquhart, 1999), LTL with Presburger constraints (Demri, 2006), data logics (Demri and Lazić, 2009; Figueira and Segoufin, 2009), etc.

A related concept is the *order type* of a well partial order (de Jongh and Parikh, 1977), which corresponds to the maximal transfinite length of an uncontrolled bad sequence. Although order types do not translate into bounds on controlled sequences,[4] they are sometimes good indicators, a rule of thumb being that an upper bound in $\mathfrak{F}_\alpha$ is often associated with an order type of ω^α, which actually holds in our case. Such questions have been mostly investigated for the complexity of term rewriting systems (see Lepper, 2004, and the references therein), where for instance the maximal derivation length of a term rewriting system compatible with multiset termination ordering (of order ω^k for some finite k) was shown primitive-recursive by Hofbauer (1992) (however no precise bounds in terms of k were given).

[4] For instance, ω^k is the order type of both $(\mathbb{N}^k, \leq)$ and $(M(\Sigma_k), \subseteq)$, where $M(\Sigma_k)$ is the set of multisets over a finite set Σ_k with k elements, but one needs to be careful on how a control on one structure translates into a control for the other.

9 Conclusion

In spite of the prevalent use of Dickson's Lemma in various areas of computer science, the upper bounds it offers are seldom capitalized on. Beyond the optimality of our bounds in terms of the Fast Growing Hierarchy, our first and foremost hope is for our results to improve this situation, and reckon for this on

- an arguably simpler main proof argument, that relies on a simple decomposition using disjoint sums,
- a fully worked out classification for our upper bounds—a somewhat tedious task—, which is reusable because we leave the control function as an explicit parameter,
- three template applications where our upper bounds on bad sequences translate into algorithmic upper bounds. These are varied enough not to be a mere repetition of the exact same argument, and provide good illustrations of how to employ our results.

References

Becker, T. and Weispfenning, V., 1993. *Gröbner Bases: A Computational Approach to Commutative Algebra*, volume 141 of *Grad. Texts in Math.* Springer. ISBN 0-387-97971-9.

Blass, A. and Gurevich, Y., 2008. Program termination and well partial orderings. *ACM Transactions on Computational Logic*, 9:1–26. doi:10.1145/1352582.1352586.

Bradley, A.R., Manna, Z., and Sipma, H.B., 2005. Termination analysis of integer linear loops. In *CONCUR 2005*, volume 3653 of *Lecture Notes in Computer Science*, pages 488–502. Springer. doi:10.1007/11539452_37.

Cichoń, E.A. and Tahhan Bittar, E., 1998. Ordinal recursive bounds for Higman's Theorem. *Theoretical Computer Science*, 201:63–84. doi:10.1016/S0304-3975(97)00009-1.

Clote, P., 1986. On the finite containment problem for Petri nets. *Theoretical Computer Science*, 43:99–105. doi:10.1016/0304-3975(86)90169-6.

de Jongh, D.H.J. and Parikh, R., 1977. Well-partial orderings and hierarchies. *Indagationes Mathematicae*, 39:195–207. doi:10.1016/1385-7258(77)90067-1.

Demri, S., 2006. Linear-time temporal logics with Presburger constraints: An overview. *Journal of Applied Non-Classical Logics*, 16:311–347. doi:10.3166/jancl.16.311-347.

Demri, S. and Lazić, R., 2009. LTL with the freeze quantifier and register automata. *ACM Transactions on Computational Logic*, 10. doi:10.1145/1507244.1507246.

Dershowitz, N. and Manna, Z., 1979. Proving termination with multiset orderings. *Communications of the ACM*, 22:465–476. doi:10.1145/359138.359142.

Figueira, D. and Segoufin, L., 2009. Future-looking logics on data words and trees. In *MFCS 2009*, volume 5734 of *Lecture Notes in Computer Science*, pages 331–343. Springer. doi:10.1007/978-3-642-03816-7_29.

Finkel, A. and Schnoebelen, Ph., 2001. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256:63–92. doi:10.1016/S0304-3975(00)00102-X.

Finkel, A., McKenzie, P., and Picaronny, C., 2004. A well-structured framework for analysing Petri nets extensions. *Information and Computation*, 195:1–29.

Friedman, H.M., 2001. Long finite sequences. *Journal of Combinatorial Theory, Series A*, 95:102–144. doi:10.1006/jcta.2000.3154.

Gallo, G. and Mishra, B., 1994. A solution to Kronecker's Problem. *Applicable Algebra in Engineering, Communication and Computing*, 5:343–370.

Gulwani, S., 2009. SPEED: Symbolic complexity bound analysis. In *CAV 2009*, volume 5643 of *Lecture Notes in Computer Science*, pages 51–62. Springer. doi:10.1007/978-3-642-02658-4_7.

Hofbauer, D., 1992. Termination proofs by multiset path orderings imply primitive recursive derivation lengths. *Theoretical Computer Science*, 105:129–140. doi:10.1016/0304-3975(92)90289-R.

Hoffmann, J. and Hofmann, M., 2010. Amortized resource analysis with polynomial potential. In *ESOP 2010*, volume 6012 of *Lecture Notes in Computer Science*, pages 287–306. Springer. doi:10.1007/978-3-642-11957-6_16.

Howell, R.R., Rosier, L.E., Huynh, D.T., and Yen, H.C., 1986. Some complexity bounds for problems concerning finite and 2-dimensional vector addition systems with states. *Theoretical Computer Science*, 46:107–140. doi:10.1016/0304-3975(86)90026-5.

Karp, R.M. and Miller, R.E., 1969. Parallel program schemata. *Journal of Computer and System Sciences*, 3:147–195. doi:10.1016/S0022-0000(69)80011-5.

Ketonen, J. and Solovay, R., 1981. Rapidly growing Ramsey functions. *Annals of Mathematics*, 113:27–314. doi:10.2307/2006985.

Kruskal, J.B., 1972. The theory of well-quasi-ordering: A frequently discovered concept. *Journal of Combinatorial Theory, Series A*, 13:297–305. doi:10.1016/0097-3165(72)90063-5.

Leivant, D., 2002. Implicit computational complexity for higher type functionals. In *CSL 2002*, volume 2471 of *Lecture Notes in Computer Science*, pages 841–878. Springer. doi:10.1007/3-540-45793-3_25.

Lepper, I., 2004. Simply terminating rewrite systems with long derivations. *Archiv für Mathematische Logik und Grundlagenforschung*, 43:1–18. doi:10.1007/s00153-003-0190-2.

Löb, M. and Wainer, S., 1970. Hierarchies of number theoretic functions, I. *Archiv für Mathematische Logik und Grundlagenforschung*, 13:39–51. doi:10.1007/BF01967649.

Mayr, E.W. and Meyer, A.R., 1981. The complexity of the finite containment problem for Petri nets. *Journal of the ACM*, 28:561–576. doi:10.1145/322261.322271.

Mayr, R., 2003. Undecidable problems in unreliable computations. *Theoretical Computer Science*, 297:337–354. doi:10.1016/S0304-3975(02)00646-1.

McAloon, K., 1984. Petri nets and large finite sets. *Theoretical Computer Science*, 32:173–183. doi:10.1016/0304-3975(84)90029-X.

Milner, E.C., 1985. Basic WQO- and BQO-theory. In *Graphs and Order. The Role of Graphs in the Theory of Ordered Sets and Its Applications*, pages 487–502. D. Reidel Publishing.

Podelski, A. and Rybalchenko, A., 2004. Transition invariants. In *LICS 2004*, pages 32–41. IEEE. doi:10.1109/LICS.2004.1319598.

Revesz, P.Z., 1993. A closed-form evaluation for Datalog queries with integer (gap)-order constraints. *Theoretical Computer Science*, 116:117–149. doi:10.1016/0304-3975(93)90222-F.

Schnoebelen, Ph., 2010a. Revisiting Ackermann-hardness for lossy counter machines and reset Petri nets. In *MFCS 2010*, volume 6281 of *Lecture Notes in Computer Science*, pages 616–628. Springer. doi:10.1007/978-3-642-15155-2_54.

Schnoebelen, Ph., 2010b. Lossy counter machines decidability cheat sheet. In *RP 2010*, volume 6227 of *Lecture Notes in Computer Science*, pages 51–75. Springer. doi:10.1007/978-3-642-15349-5_4.

Urquhart, A., 1999. The complexity of decision procedures in relevance logic II. *Journal of Symbolic Logic*, 64:1774–1802. doi:10.2307/2586811.

Weiermann, A., 1994. Complexity bounds for some finite forms of Kruskal's Theorem. *Journal of Symbolic Computation*, 18:463–488. doi:10.1006/jsco.1994.1059.

A Proofs Omitted from the Main Text

A.1 Proof of Theorem 3.2

Theorem 3.2. *For any* τ

$$L_\tau(t) \leq \max_{k\in\tau} \left\{1 + L_{\tau\langle k,t\rangle}(t+1)\right\}.$$

We start with some necessary notation and basic facts: For two quasiorderings $(A_1, \leq_1)$ and $(A_2, \leq_2)$, a mapping $h : A_1 \to A_2$ is a *reflection* when

$$\forall a, b \in A_1 : h(a) \leq_2 h(b) \text{ implies } a \leq_1 b\,.$$

We further say that it is a *strong reflection* when $|h(x)|_\infty \leq |x|_\infty$ for all x. (NB: we only consider reflections between quasiorderings that are subsets of some $\mathbb{N}^\tau$, hence the notion of size is well-defined.) When h is a strong reflection, we write $h : A_1 \hookrightarrow A_2$ (or just $A_1 \hookrightarrow A_2$ when h is left implicit) and say that A_2 *strongly reflects* A_1.

Strong reflections preserve controlled bad sequences: assume $h : A_1 \hookrightarrow A_2$ and that $x_0, x_1, \ldots, x_l$ is a t-controlled bad sequence over A_1. Then $h(x_0), h(x_1), \ldots, h(x_l)$ is a t-controlled bad sequence over A_2.

This notion is compatible with the composition of orderings:

Fact A.1. *Let* A, A_1, A_2 *be quasiorderings:* $A_1 \hookrightarrow A_2$ *implies* $A+A_1 \hookrightarrow A+A_2$ *and* $A \times A_1 \hookrightarrow A \times A_2$.

For $a \in A$, we let $A/a \stackrel{\text{def}}{=} \{x \in A \mid a \not\leq x\}$ denote the subset of elements that are not above a. Note that $(A/b) \subseteq (A/a)$ when $a \leq b$.

When $(A, \leq)$ is a wqo, $(A/a, \leq)$ is clearly a wqo too, called a *residual wqo*. The point is that if $\mathbf{x} = x_0, x_1, \ldots$ is a bad sequence over some A, the suffix sequence $\mathbf{y} = x_1, \ldots$ is a bad sequence over A/x_0. In the following, we extend our notations and write $L_A(t)$ for the maximal length of a t-controlled bad sequence over A when A is a subset of some $\mathbb{N}^\tau$.

Here too, the notion of residuals is compatible with the composition of orderings: if a is in A_j, we have for a disjoint sum $\sum_{i\in I} A_i$ with $j \in I$

$$(\sum_{i\in I} A_i)/a = (A_j/a) + \sum_{i\in I\setminus\{j\}} A_i\,. \tag{21}$$

More crucially, the region-based decomposition of Section 3 relies on a reflection for products

$$\big((A \times B)/\langle a, b\rangle\big) \hookrightarrow \big((A/a) \times B \;+\; A \times (B/b)\big)\,. \tag{22}$$

An immediate corollary is

$$(A^k/\langle a, \ldots, a\rangle) \hookrightarrow k \times (A/a) \times A^{k-1}\,. \tag{23}$$

Lemma A.2. *Assume* $x \in \mathbb{N}^k$ *with* $k > 0$ *and* $|x|_\infty \leq f(t) - 1$*:*

$$\mathbb{N}^k/x \hookrightarrow k \times (f(t) - 1) \times \mathbb{N}^{k-1} \;\; (\textit{i.e., } N_k(t) \times \mathbb{N}^{k-1})\,.$$

Indeed, when $k = 1$, $\mathbb{N}/x = \{0, 1, \ldots, x-1\}$, which is isomorphic to $x \times \mathbb{N}^0$, in turn strongly reflected by $(f(t)-1) \times \mathbb{N}^0$, while for $k > 1$ we reduce to the 1-dimensional case using Eq. (23).

By definition of $\tau_{\langle k,t\rangle}$ (see Eq. 9), combining Lemma A.2 and Eq. (21) directly yields

Lemma A.3. *Assume $k \in \tau$ and $x \in \mathbb{N}^k$ with $|x|_\infty \leq f(t) - 1$:*

$$\mathbb{N}^\tau/x \hookrightarrow \mathbb{N}^{\tau_{\langle k,t\rangle}} .$$

Since strong reflections preserve controlled bad sequences, we deduce

$$A_1 \hookrightarrow A_2 \text{ implies } L_{A_1}(t) \leq L_{A_2}(t) \tag{24}$$

where, for $i = 1, 2$, A_i is some $\mathbb{N}^{\tau_i}$, or one of its residuals.

We are now sufficiently equipped.

Proof (of Theorem 3.2). The proof is by induction over τ, the base case $\tau = \emptyset$ holding trivially in view of $L_\emptyset(t) = 0$. For the inductive case, assume $\tau \neq \emptyset$ and let $\mathbf{x} = x_0, x_1, \ldots, x_l$ be a t-controlled bad sequence over $\mathbb{N}^\tau$ with maximal length, so that $L_\tau(t) = l + 1$. Write $\mathbf{y} = x_1, \ldots, x_l$ for the suffix sequence: $\mathbf{y}$ is a $(t+1)$-controlled bad sequence over $\mathbb{N}^\tau/x_0$. Since x_0 belongs to $\mathbb{N}^k$ for some $k \in \tau$, we deduce $l \leq L_{\tau_{\langle k,t\rangle}}(t+1)$ by combining Lemma A.3 and Eq. (24) and using the induction hypothesis. Which concludes our proof. ☐

A.2 Proof of Lemma 5.1

Let us first introduce a third, less standard, so-called "*dominance*" ordering on multisets, given by

$$\{a_1, \ldots, a_n\} \sqsubseteq \{b_1, \ldots, b_m\} \stackrel{\text{def}}{\Leftrightarrow} n \leq m \wedge a_1 \leq b_1 \wedge \ldots \wedge a_n \leq b_n \tag{25}$$

where it is assumed that elements are denoted in decreasing order, i.e., $a_1 \geq a_2 \geq \ldots \geq a_n$ and $b_1 \geq \ldots \geq b_m$. In other words, $\tau \sqsubseteq \tau'$ when every element in τ is dominated by its own sibling element in τ' (additionally τ' may have extra elements). For dominance, reflexivity and transitivity are clear. We let the reader check that the dominance ordering sits between the inclusion ordering and the multiset ordering.

In order to exploit Eq. (13), we need some basic properties of the operation that transforms τ into $\tau_{\langle k,t\rangle}$.

Lemma A.4 (About $\tau_{\langle k,t\rangle}$)**.**

1. *$\tau_{\langle k,t\rangle} <_m \tau$.*
2. *If $\tau \subseteq \tau'$ then $\tau_{\langle k,t\rangle} \subseteq \tau'_{\langle k,t\rangle}$.*
3. *If $\{k, l\} \subseteq \tau$ then $\tau_{\langle l,t\rangle\langle k,t'\rangle} = \tau_{\langle k,t'\rangle\langle l,t\rangle}$.*
4. *If $\{k, l\} \subseteq \tau$ with furthermore $k \leq l$ and $t \leq t'$, then $\tau_{\langle l,t\rangle\langle k,t'\rangle} \sqsubseteq \tau_{\langle k,t\rangle\langle l,t'\rangle}$.*
5. *If $\tau \sqsubseteq \tau'$ and $k \in \tau$ then there exists $l \in \tau'$ such that $k \leq l$ and $\tau_{\langle k,t\rangle} \sqsubseteq \tau'_{\langle l,t\rangle}$.*

Proof Sketch. For *3*, we note that $\tau_{\langle l,t\rangle\langle k,t'\rangle}$ and $\tau_{\langle k,t'\rangle\langle l,t\rangle}$ are obtained from τ by performing *exactly the same removals and additions of elements.* These are perhaps performed in different orders, but this does not change the end result.

For *4*, we note that $\tau_{\langle l,t\rangle\langle k,t'\rangle}$ is some $\tau - \{k,l\} + \tau_1$ for

$$\tau_1 = N_l(t) \times \{l-1\} + N_k(t') \times \{k-1\}$$

while $\tau_{\langle k,t\rangle\langle l,t'\rangle}$ is $\tau - \{k,l\} + \tau_2$ for

$$\tau_2 = N_l(t') \times \{l-1\} + N_k(t) \times \{k-1\} \ .$$

From $l \geq k$ and $f(t) \leq f(t')$ we deduce

$$N_l(t) + N_k(t') \leq N_l(t') + N_k(t) \ .$$

Hence τ_1 has less elements than τ_2. Furthermore, τ_1 has less of the larger "$l-1$" elements since $N_l(t) \leq N_l(t')$. Thus $\tau_1 \sqsubseteq \tau_2$, entailing $\tau - \{k,l\} + \tau_1 \sqsubseteq \tau - \{k,l\} + \tau_2$.

For *5*, we use the $l = b_i$ that corresponds to $k = a_i$ in the definition of dominance ordering. This ensures both $k \leq l$ (hence $N_k(t) \leq N_l(t)$ and $N_k(t) \times \{k-1\} \sqsubseteq N_l(t) \times \{l-1\}$) and $\tau - \{k\} \sqsubseteq \tau' - \{l\}$. Finally $\tau_{\langle k,t\rangle} \sqsubseteq \tau'_{\langle l,t\rangle}$. □

Lemma A.5 (Monotony w.r.t. dominance)**.** *If $\tau \sqsubseteq \tau'$ then $M_\tau(t) \leq M_{\tau'}(t)$.*

Proof. By induction over τ. The base case, $\tau = \emptyset$, is covered with $M_\emptyset(t) = 0$. For the inductive case, we assume that $\tau \neq \emptyset$ so that $M_\tau(t)$ is $1 + M_{\tau_{\langle k,t\rangle}}(t+1)$ for some $k \in \tau$. With Lemma A.4.*5*, we pick a $l \geq k$ such that $\tau_{\langle k,t\rangle} \sqsubseteq \tau'_{\langle l,t\rangle}$. Then

$$\begin{aligned} M_\tau(t) &= 1 + M_{\tau_{\langle k,t\rangle}}(t+1) && \text{(by assumption)} \\ &\leq 1 + M_{\tau'_{\langle l.t\rangle}}(t+1) && \text{(by ind. hyp., using Lemma A.4.}\mathit{5}\text{)} \\ &\leq M_{\tau'}(t) \ . && \text{(by Eq. (13), since } l \in \tau'\text{)} \end{aligned}$$

□

Lemma 5.1 (see Appendix A.2)**.** *Let $k = \min \tau$ and $l \in \tau$. Then $M_{\tau_{\langle l,t\rangle}}(t+1) \leq M_{\tau_{\langle k,t\rangle}}(t+1)$ and, hence,*

$$\begin{aligned} &M_\emptyset(t) = 0 \\ &M_\tau(t) = 1 + M_{\tau_{\langle \min\tau, t\rangle}}(t+1) \quad \textit{for } \tau \neq \emptyset \ . \end{aligned}$$

Proof. By induction over τ. The case where $l = k$ is obvious so we assume $l > k$ and hence $\{k,l\} \subseteq \tau$. Now

$$\begin{aligned} M_{\tau_{\langle k,t\rangle}}(t+1) &\geq 1 + M_{\tau_{\langle k,t\rangle\langle l,t+1\rangle}}(t+2) && \text{(by Eq. (13), since } l \in \tau_{\langle k,t\rangle}\text{)} \\ &\geq 1 + M_{\tau_{\langle l,t\rangle\langle k,t+1\rangle}}(t+2) && \text{(combining lemmata A.4.}\mathit{4}\text{ and A.5)} \\ &= M_{\tau_{\langle l,t\rangle}}(t+1) \ . && \text{(by ind. hyp., since } k = \min\tau_{\langle l,t\rangle}\text{)} \end{aligned}$$

□

Let us close this section on M with a consequence of Lemma 5.1:

Corollary A.6. *Let $\tau = \emptyset$ or $\tau' \leq_m \{\min \tau\}$. Then for all $t \geq 0$,*

$$M_{\tau+\tau'}(t) = M_{\tau'}(t) + M_\tau(t + M(\tau', t)) \; .$$

Proof. The statement is immediate if $\tau = \emptyset$. Otherwise, we prove it by induction over τ'. The base case, $\tau' = \emptyset$, is covered with

$$M_\tau(t) = 0 + M_\tau(t + 0) = M_\emptyset(t) + M_\tau(t + M_\emptyset(t)) \; .$$

For the inductive case, we assume $\tau' \neq \emptyset$, so that $k = \min \tau'$ exists and is no greater than $\min \tau$. Then by Lemma A.4.*1*, $\tau'_{\langle k,t\rangle} <_m \tau'$, and furthermore $\tau'_{\langle k,t\rangle} \leq_m \{\min \tau\}$. Thus

$$\begin{aligned}
&M_{\tau+\tau'}(t) \\
&\qquad = 1 + M_{\tau+\tau'_{\langle k,t\rangle}}(t+1) && \text{(by Lemma 5.1)} \\
&\qquad = 1 + M_{\tau'_{\langle k,t\rangle}}(t+1) + M_\tau(t + 1 + M_{\tau'_{\langle k,t\rangle}}(t+1)) && \text{(by ind. hyp.)} \\
&\qquad = M_{\tau'}(t) + M_\tau(t + M_{\tau'}(t)) \; . && \text{(by Lemma 5.1)}
\end{aligned}$$

□

A.3 Proof of Proposition 5.2

Proposition 5.2 (see Appendix A.3)**.** *Let $k, r \geq 1$ be natural numbers and $\gamma \geq 1$ an ordinal. If f is a monotone unary function of $\mathfrak{F}_\gamma$ with $f(x) \geq \max(1, x)$ for all x, then $M_{r\times\{k\},f}$ is in $\mathfrak{F}_{\gamma+k-1}$.*

Proof. We define in the next paragraph another function $G_{k,f}$, which is monotone and such that $G_{k,f}(x) \geq x$ (Lemma A.7). It further belongs to $\mathfrak{F}_{\gamma+k-1}$ by Lemma A.9, and is such that $M_{r\times\{k\},f}(x) = G^r_{k,f}(x) \dot{-} x$ according to Lemma A.8, i.e. $M_{r\times\{k\},f}$ is defined through finite substitution from $G_{k,f}$ and cut-off subtraction,[5] and therefore also belongs to $\mathfrak{F}_{\gamma+k-1}$. □

More about the Fast Growing Hierarchy Let us first give a few more details on the Fast Growing Hierarchy. The class of functions $\mathfrak{F}_k$ is the closure of $\{\lambda x.0, \lambda xy.x + y, \lambda x.x_i\} \cup \{F_n \mid n \leq k\}$ under the operations of

substitution if $h_0, h_1, \ldots, h_n$ belong to the class, then so does f if

$$f(x_1, \ldots, x_n) = h_0(h_1(x_1, \ldots, x_n), \ldots, h_n(x_1, \ldots, x_n))$$

limited recursion if h_1, h_2, and h_3 belong to the class, then so does f if

$$\begin{aligned}
f(0, x_1, \ldots, x_n) &= h_1(x_1, \ldots, x_n) \\
f(y + 1, x_1, \ldots, x_n) &= h_2(y, x_1, \ldots, x_n, f(y, x_1, \ldots, x_n)) \\
f(y, x_1, \ldots, x_n) &\leq h_3(y, x_1, \ldots, x_n) \; .
\end{aligned}$$

[5] Cut-off subtraction

$$x \dot{-} y \stackrel{\text{def}}{=} \begin{cases} x - y & \text{if } x \geq y \\ 0 & \text{otherwise,} \end{cases}$$

can be defined by limited primitive recursion in $\mathfrak{F}_0$.

Here are a few monotonicity properties that will be useful in the following:

- for each α and all n, x, y with $x > y$, $F_\alpha^n(x) > F_\alpha^n(y)$ (Löb and Wainer, 1970, Lemma 2.6.(iii)),
- for each α and all m, n, x, if $m > n$, $F_\alpha^m(x) > F_\alpha^n(x)$ (Löb and Wainer, 1970, Lemma 2.6.(iv)), and
- for each α and every $k \geq 1$ we have, for all n and x, $F_{\alpha+k}^n(x) \geq F_\alpha^n(x)$ (Löb and Wainer, 1970, Lemma 2.8).

A Simpler Version of M We consider a fast iteration hierarchy for $M_{r\times\{k\},f}$, given a monotone unary function f:

$$G_{1,f} \stackrel{\text{def}}{=} f(x) + x \tag{26}$$

$$G_{k+1,f}(x) \stackrel{\text{def}}{=} G_{k,f}^{N_{k+1}(x)}(x+1) \; . \tag{27}$$

Lemma A.7. *Let f be a monotone unary function such that $f(x) \geq x$ and let $n \geq 1$. Then the function $G_{n,f}$ is monotone and such that $G_{n,f}(x) \geq x$.*

We leave the previous proof to the reader, and turn to the main motivation for introducing $G_{k,f}$:

Lemma A.8. *Let $k \geq 1$. Then for all $r \geq 1$ and $x \geq 0$,*

$$M_{r\times\{k\},f}(x) = G_{k,f}^r(x) - x \; .$$

Proof. We proceed by induction on types τ of form $r \times \{k\}$. For the base case, which is $\tau = \{1\}$, we have for all x

$$M_{\{1\},f}(x) = f(x) = G_{1,f}(x) - x \; . \qquad \text{(by Def. (26))}$$

For the induction step, we first consider the case $\tau = \{k\}$. Then, for all x,

$$\begin{aligned} M_{\{k\},f}(x) &= 1 + M_{N_k(x)\times\{k-1\},f}(x+1) && \text{(by Lemma 5.1)} \\ &= 1 + G_{k-1,f}^{N_k(x)}(x+1) - x - 1 && \text{(by ind. hyp.)} \\ &= G_{k,f}(x) - x \; . && \text{(by Def. (27))} \end{aligned}$$

Finally, for the case $\tau = (r+1) \times \{k\}$, for all x,

$$
\begin{aligned}
&M_{r+1\times\{k\},f}(x)\\
&\quad = 1 + M_{r\times\{k\}+N_k(x)\times\{k-1\},f}(x+1) && \text{(by Lemma 5.1)}\\
&\quad = 1 + M_{N_k(x)\times\{k-1\},f}(x+1)\\
&\qquad + M_{r\times\{k\},f}\big(x+1+M_{N_k(x)\times\{k-1\},f}(x+1)\big) && \text{(by Corollary A.6)}\\
&\quad = 1 + G^{N_k(x)}_{k-1,f}(x+1) - x - 1\\
&\qquad + M_{r\times\{k\},f}\Big(x+1+G^{N_k(x)}_{k-1,f}(x+1)-x-1\Big)\\
&&& \text{(by ind. hyp. on } M_{N_k(x)\times\{k-1\},f})\\
&\quad = G^{N_k(x)}_{k-1,f}(x+1) - x + M_{r\times\{k\},f}\Big(G^{N_k(x)}_{k-1,f}(x+1)\Big)\\
&\quad = G^{N_k(x)}_{k-1,f}(x+1) - x + G^{r}_{k,f}\Big(G^{N_k(x)}_{k-1,f}(x+1)\Big)\\
&\qquad - G^{N_k(x)}_{k-1,f}(x+1) && \text{(by ind. hyp. on } M_{r\times\{k\},f})\\
&\quad = G^{r+1}_{k,f}(x) - x\;. && \text{(by Def. (27))}
\end{aligned}
$$

□

Placing $G_{n,f}$ in the Fast Growing Hierarchy We prove the following lemma:

Lemma A.9. *Let $\gamma \geq 1$ be an ordinal and f be a unary monotone function in $\mathfrak{F}_\gamma$ with $f(x) \geq \max(1,x)$ for all x. Then for all $k \geq 1$, $G_{k,f}$ belongs to $\mathfrak{F}_{\gamma+k-1}$.*

Proof. Since $\gamma \geq 1$, and because f is in $\mathfrak{F}_\gamma$, the function

$$h(x) \stackrel{\text{def}}{=} k \cdot f(x) + x + 1\;, \tag{28}$$

defined through finite substitution from f and addition, is monotone and also belongs to $\mathfrak{F}_\gamma$. Then, there exists $p \in \mathbb{N}$ such that, for all x (Löb and Wainer, 1970, Theorem 2.10):[6]

$$h(x) < F^p_\gamma(x)\;. \tag{29}$$

We start the proof of the lemma by several inequalities in Claims A.9.1 and A.9.2.

Claim A.9.1. For all $y \geq 1$, and $x, n \geq 0$

$$F^{y\cdot h(x)}_{\gamma+n}(x+1) \leq F^{y\cdot(p+1)}_{\gamma+n+1}(x)\;.$$

Proof. We proceed by induction on y for the proof of the claim. If $y = 1$, then

$$
\begin{aligned}
F^{h(x)}_{\gamma+n}(x+1) &\leq F^{h(x)}_{\gamma+n}(h(x)) && \text{(since } h(x) \geq x+1)\\
&\leq F^{h(x)+1}_{\gamma+n}(h(x)) && \text{(by monotonicity of } F_{\gamma+n})\\
&= F_{\gamma+n+1}(h(x)) && \text{(by Def. (15))}\\
&< F_{\gamma+n+1}\big(F^p_\gamma(x)\big) && \text{(by (29) and monotonicity of } F_{\gamma+n+1})\\
&\leq F^{p+1}_{\gamma+n+1}(x) && \text{(by (Löb and Wainer, 1970, Lemma 2.8))}
\end{aligned}
$$

[6]The theorem is actually stated for a different version of F_1, but it turns out to hold with ours as well.

and the claim holds. Quite similarly for the induction step,

$$
\begin{aligned}
F_{\gamma+n}^{(y+1)\cdot h(x)}(x+1) &= F_{\gamma+n}^{h(x)}\Big(F_{\gamma+n}^{y\cdot h(x)}(x+1)\Big)\\
&\leq F_{\gamma+n}^{h(x)}\Big(F_{\gamma+n+1}^{y\cdot(p+1)}(x)\Big)\\
&\qquad\qquad\text{(by ind. hyp. and monotonicity of } F_{\gamma+n}^{h(x)}\text{)}\\
&\leq F_{\gamma+n}^{h(x)}\Big(h(F_{\gamma+n+1}^{y\cdot(p+1)}(x))\Big)\\
&\qquad\qquad\text{(since } h(x)\geq x \text{ and by monotonicity of } F_{\gamma+n}^{h(x)}\text{)}\\
&\leq F_{\gamma+n}^{h(F_{\gamma+n+1}^{y\cdot(p+1)}(x))+1}\Big(h(F_{\gamma+n+1}^{y\cdot(p+1)}(x))\Big)\\
&\qquad\text{(since } F_{\gamma+n+1}^{y\cdot(p+1)}(x)\geq x \text{ and by monotonicity of } h \text{ and } F_{\gamma+n}^{y}(x)\text{)}\\
&= F_{\gamma+n+1}\Big(h(F_{\gamma+n+1}^{y\cdot(p+1)}(x))\Big) \qquad\qquad \text{(by Def. (15))}\\
&< F_{\gamma+n+1}\Big(F_{\gamma}^{p}(F_{\gamma+n+1}^{y\cdot(p+1)}(x))\Big)\\
&\qquad\qquad\text{(by (29) and monotonicity of } F_{\gamma+n+1}\text{)}\\
&\leq F_{\gamma+n+1}^{p+1}\Big(F_{\gamma+n+1}^{y\cdot(p+1)}(x)\Big)\\
&\qquad\qquad\text{(by (Löb and Wainer, 1970, Lemma 2.8))}\\
&= F_{\gamma+n+1}^{(y+1)\cdot(p+1)}(x)\;. \qquad\qquad \square
\end{aligned}
$$

Claim A.9.2. For all $1 \leq n \leq k$ and $x, y \geq 0$,

$$G_{n,f}^{y}(x) \leq F_{\gamma+n-1}^{y\cdot(p+1)^n}(x)\;.$$

Proof. Let us first show that, for all $1 \leq n \leq k$,

$$\forall x. G_{n,f}(x) \leq F_{\gamma+n-1}^{(p+1)^n}(x) \text{ implies } \forall x, y. G_{n,f}^{y}(x) \leq F_{\gamma+n-1}^{y\cdot(p+1)^n}(x). \tag{30}$$

By induction on y: for $y = 0$, $G_{n,f}^{0}(x) = x = F_{\gamma+n-1}^{0\cdot(p+1)^n}(x)$, and for the induction step on y, for any x, y,

$$
\begin{aligned}
G_{n,f}^{y+1}(x) &= G_{n,f}\Big(G_{n,f}^{y}(x)\Big)\\
&\leq G_{n,f}\Big(F_{\gamma+n-1}^{y\cdot(p+1)^n}(x)\Big) \qquad \text{(by ind. hyp. and monotonicity of } G_{n,f}\text{)}\\
&\leq F_{\gamma+n}^{(p+1)^n}\Big(F_{\gamma+n}^{y\cdot(p+1)^n}(x)\Big) \qquad\qquad \text{(by ind. hyp.)}\\
&= F_{\gamma+n}^{(y+1)\cdot(p+1)^n}(x)\;.
\end{aligned}
$$

It remains to prove that $G_{n,f}(x) \leq F_{\gamma+n-1}^{(p+1)^n}(x)$ by induction on n: for $n = 1$,

$$G_{1,f}(x) = f(x) + x \leq h(x) < F_{\gamma}^{p}(x) \leq F_{\gamma+1-1}^{p+1}(x)$$

by (29) and monotonicity of F_γ. For the induction step on n,

$$\begin{aligned}
G_{n+1,f}(x) &= G_{n,f}^{N_{n+1}(x)}(x+1) \\
&\leq F_{\gamma+n-1}^{N_{n+1}(x)\cdot(p+1)^n}(x+1) && \text{(by ind. hyp. and (30) for } y = N_{n+1}(x)) \\
&\leq F_{\gamma+n-1}^{h(x)\cdot(p+1)^n}(x+1) && \text{(since } n \leq k) \\
&\leq F_{\gamma+n}^{(p+1)\cdot(p+1)^n}(x) && \text{(by Claim A.9.1 for } y = (p+1)^n \geq 1) \\
&= F_{\gamma+n}^{(p+1)^{n+1}}(x)\ . && \square
\end{aligned}$$

The main proof consists in first proving that for all $1 \leq n \leq k$,

$$\lambda x.G_{n,f}(x) \in \mathfrak{F}_{\gamma+n-1} \text{ implies } \lambda xy.G_{n,f}^y(x) \in \mathfrak{F}_{\gamma+n}\ . \tag{31}$$

Indeed, for all x, y,

$$\begin{aligned}
G_{n,f}^y(x) &\leq F_{\gamma+n-1}^{y\cdot(p+1)^n}(x) && \text{(by Claim A.9.2)} \\
&\leq F_{\gamma+n-1}^{x+y\cdot(p+1)^n+1}(x+y\cdot(p+1)^n) && \text{(by monotonicity of } F_{\gamma+n-1}) \\
&= F_{\gamma+n}(x+y\cdot(p+1)^n)\ .
\end{aligned}$$

Thus $\lambda xy.G_{n,f}^y(x)$ is defined by a simple recursive definition from $G_{n,f}$, which is in $\mathfrak{F}_{\gamma+n-1} \subseteq \mathfrak{F}_{\gamma+n}$ by hypothesis, and is limited by a function in $\mathfrak{F}_{\gamma+n}$, namely $\lambda xy.F_{\gamma+n}(x+y\cdot(p+1)^n)$, clearly defined by finite substitution from addition and $F_{\gamma+n}$. It belongs therefore to $\mathfrak{F}_{\gamma+n}$.

It remains to prove that for all $1 \leq n \leq k$, $G_{n,f}$ is in $\mathfrak{F}_{\gamma+n-1}$. We proceed by induction on n; for the case $n = 1$, $G_{1,f} = f(x) + x$ is defined by finite substitution from f and addition, thus belongs to $\mathfrak{F}_\gamma$ by hypothesis. For the induction step on n, $\lambda x.G_{n+1,f}(x) = \lambda x.G_{n,f}^{N_{n+1}(x)}(x+1)$ is defined by substitution from

- addition,
- $\lambda x.N_{n+1}(x) = \lambda x.(n+1)\cdot(f(x) \dot{-} 1)$, which is defined through cut-off subtraction (recall that $f(x) \geq 1$ for all x), f, and addition, and thus belongs to $\mathfrak{F}_\gamma \subseteq \mathfrak{F}_{\gamma+n}$, and from
- $\lambda xy.G_{n,f}^y(x)$, which is by induction hypothesis and Eq. (31) in $\mathfrak{F}_{\gamma+n}$.

Thus $\lambda x.G_{n+1,f}(x)$ belongs to $\mathfrak{F}_{\gamma+n}$. $\square$

A.4 Proof of Proposition 6.3

Proposition 6.3 (see Appendix A.4)**.** *Let $\gamma \geq 0$ be an ordinal and $k, r \geq 1$ natural numbers. Then, for all $t \geq 0$, $\ell_{r,k,F_\gamma}(t) \geq F_{\gamma+k-1}^r(t)$.*

Proof. Let us first show that for all $k \geq 1$

$$\forall t\,.\,\ell_{k,F_\gamma}(t) \geq F_{\gamma+k-1}(t) \quad \text{implies } \forall r \geq 1, t\,.\,\ell_{k,F_\gamma}^r(t) \geq F_{\gamma+k-1}^r(t)\ . \tag{32}$$

By induction on r; the base case for $r = 1$ holds by hypothesis, and the induction step holds by monotonicity of $F_{\gamma+k-1}$.

It remains to prove $\ell_{k,F_\gamma}(t) \geq F_{\gamma+k-1}(t)$ by induction over $k \geq 1$. The base case is settled by $\ell_{1,F_\gamma}(t) = F_\gamma(t) = F_{\gamma+1-1}(t)$, and for the induction step, we have for all $t \geq 0$:

$$\begin{aligned}
\ell_{k+1,F_\gamma}(t) &= \sum_{j=1}^{F_\gamma(t)} \ell_{k,F_\gamma}\left(o^{j-1}_{k,F_\gamma}(t)\right) \\
&\geq \sum_{j=1}^{F_\gamma(t)} \ell^j_{k,F_\gamma}(t) && \text{(by monotonicity of } \ell) \\
&\geq \ell^{F_\gamma(t)}_{k,F_\gamma}(t) && \text{(still by monotonicity of } \ell) \\
&\geq F^{F_\gamma(t)}_{\gamma+k-1}(t) && \text{(by ind. hyp. and (32))} \\
&\geq F^{t+1}_{\gamma+k-1}(t) && \text{(by monotonicity of } F_{\gamma+k-1}) \\
&= F_{\gamma+k}
\end{aligned}$$

Finally, for all $r \geq 1$ and $t \geq 0$,

$$\begin{aligned}
\ell_{r,k,F_\gamma}(t) &\geq \sum_{j=1}^{r} \ell_{k,F_\gamma}\left(o^{j-1}_{k,F_\gamma}(t)\right) \\
&\geq \sum_{j=1}^{r} \ell^j_{k,F_\gamma}(t) && \text{(by monotonicity of } \ell) \\
&\geq \ell^r_{k,F_\gamma}(t) && \text{(still by monotonicity of } \ell) \\
&\geq F^r_{\gamma+k-1}(t)\ . && \text{(by (32) and the previous argument)}
\end{aligned}$$

□