Coverability Trees for Petri Nets with Unordered Data

Piotr Hofman[1,i], Sławomir Lasota[2,ii], Ranko Lazić[3,iii], Jérôme Leroux[4], Sylvain Schmitz[1,iv], and Patrick Totzke[3,iii]

[1] LSV, ENS Cachan & CNRS, Université Paris-Saclay, France
[2] University of Warsaw, Poland
[3] DIMAP, Department of Computer Science, University of Warwick, UK
[4] LaBRI, CNRS, France

Abstract. We study an extension of classical Petri nets where tokens carry values from a countable data domain, that can be tested for equality upon firing transitions. These Unordered Data Petri Nets (UDPN) are well-structured and therefore allow generic decision procedures for several verification problems including coverability and boundedness.
We show how to construct a finite representation of the coverability set in terms of its ideal decomposition. This not only provides an alternative method to decide coverability and boundedness, but is also an important step towards deciding the reachability problem. This also allows to answer more precise questions about the reachability set, for instance whether there is a bound on the number of tokens on a given place (place boundedness), or if such a bound exists for the number of different data values carried by tokens (place width boundedness).
We provide matching Hyper-Ackermann bounds on the size of coverability trees and on the running time of the induced decision procedures.

1 Introduction

Unordered data Petri nets (UDPN [15]) extend Petri nets by decorating tokens with data values taken from some countable data domain $\mathbb{D}$. These values act as pure names: they can only be compared for equality or non-equality upon firing transitions. Such systems can model for instance distributed protocols where process identities need to be taken into account [21]. UDPNs also coincide with the natural generalisation of Petri nets in the framework of sets with atoms [3]. In spite of their high expressiveness, UDPNs fit in the large family of Petri net extensions among the *well-structured* ones [1, 7]. As such, they still enjoy decision

[i] Partially funded by the Polish National Science Centre grant 2013/09/B/ST6/01575 and by Labex Digicosme, Université Paris-Saclay, project VERICONISS.
[ii] Partially funded by the Polish National Science Centre grant 2012/07/B/ST6/01497.
[iii] Partially supported by the EPSRC grant EP/M011801/1.
[iv] Partially supported by the ANR grant ANR-14-CE28-0005 prodaq and by the Leverhulme Trust Visiting Professorship VP1-2014-041.

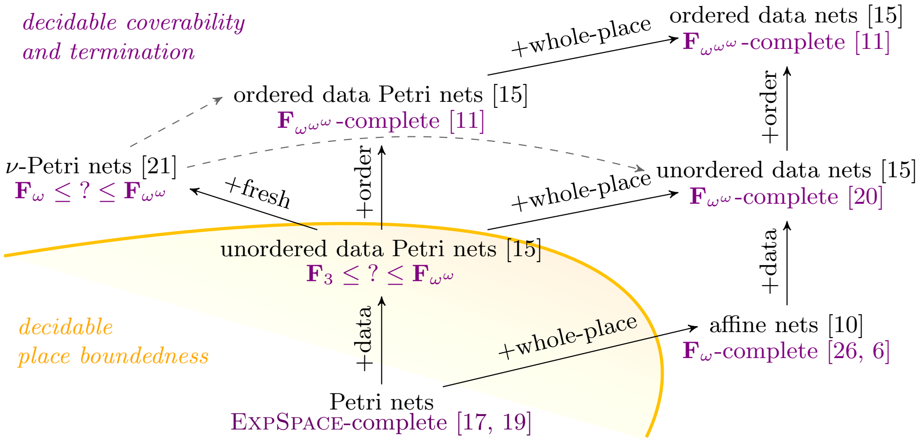


Fig. 1. A short taxonomy of some well-structured extensions of Petri nets. Complexities in violet refer to the coverability and termination problems, and can be taken as proxies for expressiveness; the exact complexities of coverability and termination in ν-Petri nets and UDPNs are unknown at the moment. Place boundedness is decidable below the yellow line and undecidable above. As indicated by the dashed arrows, freshness can be enforced using a dense linear order or whole-place operations.

procedures for several verification problems, prominently safety (through the *coverability* problem) and termination.

Unordered data Petri nets have an interesting position in the taxonomy of well-structured Petri net extensions (see Fig. 1). Indeed, all their extensions forgo the decidability of the *reachability* problem (whether a target configuration is reachable) and of the *place boundedness* problem (whether the number of tokens in a given place can be bounded along all runs): this is the case of *ν-Petri nets* [21] that allow to create fresh data values, of *ordered data Petri nets* [15] that posit a dense linear ordering on $\mathbb{D}$, and of *unordered data nets* [15] that allow to perform 'whole-place' operations, which move and/or duplicate all the tokens from a place to another. By contrast, it is currently open whether reachability is decidable in UDPNs, and a consequence of our results in this paper is that place boundedness is decidable—which is a significant first step if we wish to adapt to UDPNs some of the known algorithms for reachability in Petri nets [18, 13].

Contributions. In this paper, we show how to construct finite *coverability trees* for UDPNs, adapting the existing construction of Karp and Miller [12] for Petri nets. Such trees are constructed forward from an initial configuration like reachability trees, but approximate the latter by *accelerating* sequences of transitions and explicitly manipulating limits of reachable configurations as downwards-closed sets (see Sec. 4.1). We rely for all this on a general theory for representing downwards-closed sets as finite unions of *ideals* developed by Finkel and Goubault-Larrecq [9] for this exact purpose (see Sec. 3).

Coverability trees contain a wealth of information about the system at hand, and allow to answer various coverability and boundedness questions—allowing us to derive a new result: the place boundedness problem is decidable for UDPNs, and so are its variants, like place width- and place depth boundedness (see Sec. 2). We also establish in Sec. 5 matching 'hyper-Ackermannian' lower and upper bounds on the size of UDPNs coverability trees. This yields $\mathbf{F}_{\omega^\omega}$ upper bounds for the already mentioned decidable problems in UDPNs, in terms of the fast-growing complexity classes $(\mathbf{F}_\alpha)_\alpha$ from [23]. These complexity results rely largely on the work of Rosa-Velardo [20] on the complexity of coverability in unordered data nets.

Related Work. A coverability tree construction has already been undertaken by Rosa-Velardo, Martos-Salgado, and de Frutos-Escrig [22] in the case of ν-Petri nets, and inescapably there are many similarities between their work and ours. Our construction does however not merely remove freshness constraints from theirs: (1) we start anew and rely on a strong invariant on the form of ideals in UDPN coverability trees, which leads to significant simplifications but would be markedly difficult to extract from Rosa-Velardo et al.'s construction, and (2) coverability trees are not necessarily finite for ν-Petri nets (place boundedness is indeed undecidable [21]), which means that our termination argument and complexity bounds are entirely new considerations.

Like Rosa-Velardo et al. [22] we rely on the work of Finkel and Goubault-Larrecq [9] on forward analysis of well-structured systems. Finkel and Goubault-Larrecq provide in particular an abstract generic procedure, but without guarantee of termination in general, and their framework needs to be instantiated for each specific class of systems.

2 Model

Our presentation of unordered data Petri nets differs from the original one [15] on two counts: we work with an equivalent formalism with more of a *vector addition system* [12] flavour, and because we need to work with ideals we define the syntax and the semantics on extended configurations, which allow for infinitely many different data values and infinite counts.

Let $\mathbb{Z}$ and $\mathbb{N}$ denote the sets of integers and non-negative integers respectively, and complete them as $\mathbb{Z}_\omega \stackrel{\text{def}}{=} \mathbb{Z} \uplus \{\omega\}$ and $\mathbb{N}_\omega \stackrel{\text{def}}{=} \mathbb{N} \uplus \{\omega\}$ with a new top element ω with $\omega > z$ and $z + \omega = \omega + z = \omega$ for all z in $\mathbb{Z}$. Given a dimension k in $\mathbb{N}$, we denote the projection into the ith component of a vector $\boldsymbol{v} \in \mathbb{Z}_\omega^k$ by $\boldsymbol{v}[i]$ and define the product ordering and sum over $\mathbb{Z}_\omega^k$ componentwise: $\boldsymbol{u} \leq \boldsymbol{v}$ if $\boldsymbol{u}[i] \leq \boldsymbol{v}[i]$ for all $1 \leq i \leq k$, and $(\boldsymbol{u} + \boldsymbol{v})[i] \stackrel{\text{def}}{=} \boldsymbol{u}[i] + \boldsymbol{v}[i]$ for all $1 \leq i \leq k$. We write $\mathbf{0}$ for the vector with 0 on all components.

Data Vectors. Fix some countable domain $\mathbb{D}$ of data values and a dimension k in $\mathbb{N}$. A *data vector* is a function $f: \mathbb{D} \to \mathbb{Z}_\omega^k$. Data vectors can be partially ordered and summed pointwise: $f \leq g$ if $f(d) \leq g(d)$ for all d in $\mathbb{D}$, and $(f+g)(d) \stackrel{\text{def}}{=} f(d)+g(d)$ for all d in $\mathbb{D}$. As usual we write $f < g$ if $f \leq g$ and $f(d) < g(d)$ for some d in $\mathbb{D}$. For a subset $K \subseteq \{1,\ldots,k\}$ we write $f{\restriction}_K$ for the projection of f into components in K: for all d in $\mathbb{D}$, $f{\restriction}_K(d) \stackrel{\text{def}}{=} (f(d)){\restriction}_K$. The *support* of a data vector f is $\mathit{Supp}_{\mathbf{0}}(f) \stackrel{\text{def}}{=} \{d \in \mathbb{D} \mid f(d) \neq \mathbf{0}\}$; f is *finitely supported* if $\mathit{Supp}_{\mathbf{0}}(f)$ is finite. We say that a data vector f is *non-negative* if $f(d)$ belongs to $\mathbb{N}^k_\omega$ for all d in $\mathbb{D}$. It is *finite* if $f(d)$ belongs to $\mathbb{Z}^k$ for all d and it is finitely supported.

We call bijections $\sigma : \mathbb{D} \to \mathbb{D}$ *(data) permutations* and write $f\sigma$ for the composition of a permutation σ and a data vector f. If two data vectors f, g satisfy $f = g\sigma$ for some permutation σ, we say f and g are equal *up to permutation* and write $f \equiv g$.

In the sequel we will consider sets of data vectors that are *finite up to permutation*: a set X of data vectors is finite up to permutation if there is a finite subset X' of X such that every $f \in X$ is equal up to permutation to some $f' \in X'$. Note that if X is closed under permutations then X' can be used as its finite presentation; any such finite set X' we call *representative* of X.

Definition 2.1 (UDPN). *An* unordered data Petri net *(UDPN) is a finite set $\mathcal{T}$ of finite data vectors. A* transition *is a data vector $t \stackrel{\text{def}}{=} f\sigma$, where $f \in \mathcal{T}$ and σ is a data permutation. There is a* step *$f \xrightarrow{t} g$ between non-negative data vectors f, g if $g = f + t$ for some transition t. Note that this enforces that $f(d)+t(d) \geq 0$ for all d in $\mathbb{D}$ since g is non-negative. We simply write $f \to g$ if $f \xrightarrow{t} g$ for some transition t and let $\xrightarrow{*}$ denote the transitive and reflexive closure of $\to$.*

A configuration *is a finite non-negative data vector f, i.e. $f(d)$ belongs to $\mathbb{N}^k$ for all d in $\mathbb{D}$, and $f(d) = \mathbf{0}$ for almost all d in $\mathbb{D}$. We write Confs for the set of configurations and note that Confs is closed under UDPN steps. The* reachability *set from a given vector f is defined as*

$$\mathit{Reach}(f) \stackrel{\text{def}}{=} \{g \in \mathit{Confs} \mid f \xrightarrow{*} g\}. \tag{1}$$

Observe that UDPNs over any domain with cardinality $|\mathbb{D}| = 1$ are classical vector addition systems [12]. Notice also that, the set of transitions in an UDPN is finite up to permutations of $\mathbb{D}$ and that the step relation is closed under permutations: For every non-negative data vector f, transition t and permutation σ we have that

$$f \xrightarrow{t} g \text{ implies } f\sigma \xrightarrow{t\sigma} g\sigma. \tag{2}$$

Example 2.2. For the domain $\mathbb{D} \stackrel{\text{def}}{=} \mathbb{N}$ and $k \stackrel{\text{def}}{=} 2$, consider a 2-dimensional UDPN $\mathcal{T} = \{t_1, t_2\}$, with vectors t_1, t_2 defined as $t_1 : 0 \mapsto (1,0)$ and $t_1 : n \mapsto (0,0)$ for all $n > 0$, and $t_2 : 0 \mapsto (-1,-1)$, $t_2 : 1 \mapsto (1,1)$ and $t_2 : n \mapsto (0,0)$ for all $n > 1$.

The configuration f_0 with $f_0 : 0 \mapsto (1,1)$ and $f_0 : n \mapsto (0,0)$ for $n > 0$, has infinitely many t_1-successors. Namely, $g_i \stackrel{\text{def}}{=} f_0 + t_1\sigma_i$ for every permutation

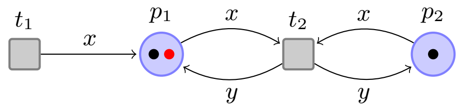


Fig. 2. A place/transition representation of the UDPN of Ex. 2.2 in the style of [21, 11]. Different data values are depicted though differently coloured tokens in places (the circles), and through differently named variables in transitions (the boxes and arrows).

σ_i that swaps 0 and $i \in \mathbb{D}$. However, there are only two such successors up to permutation because $g_i \equiv g_j$ for all $i, j > 0$. The reachability set of f_0 is

$$\mathit{Reach}(f_0) = \left\{ g \;\middle|\; \begin{array}{l} \exists d_1, d_2, \ldots, d_m \in \mathbb{D} \; \exists n_1, n_2, \ldots, n_m \in \mathbb{N} \\ g(d_1) = (n_1, 1) \text{ and } \forall 1 < i \leq m,\; g(d_i) = (n_i, 0), \\ \forall d \in \mathbb{D} \setminus \{d_1, d_2, \ldots, d_m\} \; g(d) = (0,0) \end{array} \right\}. \tag{3}$$

In the sequel, we present UDPNs only up to permutation in matrix form by juxtaposing the vectors from their finite supports: we write $t_1 \stackrel{\text{def}}{=} \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$, $t_2 \stackrel{\text{def}}{=} \left[\begin{smallmatrix}-1 & 1\\-1 & 1\end{smallmatrix}\right]$ and $f_0 \stackrel{\text{def}}{=} \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$. The t_1-successors of f_0 are $g_0 = \left[\begin{smallmatrix}2\\1\end{smallmatrix}\right]$ and $g_i = \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right] + \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right] = \left[\begin{smallmatrix}1 & 1\\1 & 0\end{smallmatrix}\right]$ for $i \neq 0$. The latter is depicted in Fig. 2 in the style of coloured place/transition nets. The reachability set of f_0 can be written as $\mathit{Reach}(f_0) = \left\{ \left[\begin{smallmatrix}n_0 & n_1 & \cdots & n_m\\1 & 0 & \cdots & 0\end{smallmatrix}\right] \mid m \geq 0, n_0, \ldots, n_m \geq 1 \right\}$.

Embeddings. We say that a data vector f *embeds* into a data vector g and write $f \sqsubseteq g$ (resp. $f \sqsubset g$) if there exists an injection $\pi\colon \mathbb{D} \to \mathbb{D}$ such that $f \leq g\pi$ (resp. $f < g\pi$). The injection π itself is called an *embedding* (of f into g) and a *permutation embedding* in case it is bijective. Given a set of configurations C, its *downward-closure* $\downarrow C$ is $\{f \in \mathit{Confs} \mid \exists g \in C \,.\, f \sqsubseteq g\}$, and as usual a set C is *downwards-closed* if $\downarrow C = C$.

Finitely supported data vectors are isomorphic to finite multisets of vectors in $\mathbb{Z}_\omega^k$ when working up to data permutation. Moreover, on permutation classes of finitely supported data vectors, the embedding ordering coincides with the usual embedding ordering over finite multisets of vectors; in consequence, UDPN configurations are well-quasi-ordered by the embedding ordering. Thus an UDPN defines a quasi-ordered transition system $(\mathit{Confs}, \to, \sqsubseteq)$, which satisfies a (strong) *compatibility* condition as shown in the following lemma. Together with the fact that $(\mathit{Confs}, \sqsubseteq)$ is a wqo, this entails that it is a *well-structured* transition system in the sense of [1, 7].

Lemma 2.3 (Strong Strict Compatibility). *Let f, f', g be configurations. If $f \sqsubseteq f'$ (resp. $f \sqsubset f'$) and $f \to g$, then there exists a configuration g' with $f' \to g'$ and $g \sqsubseteq g'$ (resp. $g \sqsubset g'$).*

Proof. Consider a finite data vector t such that $f \xrightarrow{t} g$, and a permutation π of $\mathbb{D}$ such that $f \leq f'\pi$ (recall that, when working with finitely supported data vectors, embeddings can be assumed to be permutations). We claim that

$g' \stackrel{\text{def}}{=} f' + t\pi^{-1}$ satisfies $f' \xrightarrow{t\pi^{-1}} g'$ and $g \leq g'\pi$. Indeed, for all d in $\mathbb{D}$, noting $e \stackrel{\text{def}}{=} \pi(d)$,

$$g(d) = f(d) + t(d) \leq f'(\pi(d)) + t(d) = f'(e) + t(\pi^{-1}(e)) = g'(e) = g'(\pi(d)) \ .$$

Furthermore, assuming $f < f'\pi$, for at least one d in $\mathbb{D}$ the above inequality is strict. $\square$

Decision Problems. For the purpose of verification, we are interested in standard decision problems for UDPN, including *reachability* (does $f \xrightarrow{*} g$ hold for given configurations f and g?), *coverability* (given configurations f, g, does there exists $g' \sqsupseteq g$ s.t. $f \xrightarrow{*} g'$?), and *boundedness* (is $\mathit{Reach}(f)$ finite up to permutation?).

While the decidability of reachability remains open, well-structuredness of UDPN (and some basic effectiveness assumptions) implies that the coverability and boundedness problems are decidable using the generic algorithms from [1, 7]. In fact, decidability holds more generally for ordered data Petri nets [15, Thm. 4.1]. For the coverability problem, Rosa-Velardo [20, Thm. 1] proved an HyperAckermann upper bound ($\mathbf{F}_{\omega^\omega}$ in the hierarchy from [23]; see Sec. 5), while Lazić et al. [15, Thm. 5.2] proved a Tower lower bound ($\mathbf{F}_3$ in the same hierarchy). The complexity of the boundedness problem has not been studied before. Furthermore, the following, more precise variant of boundedness called *place boundedness* was not known to be decidable:

Input: An UDPN, a configuration f, and a set of coordinates $K \subseteq \{1, \ldots, k\}$.
Question: Is $\{g{\restriction}_K \mid g \in \mathit{Reach}(f)\}$ finite up to permutation?

In the presence of an infinite data domain $\mathbb{D}$, place boundedness can be further refined: even if infinitely many configurations are reachable, the system can still be bounded in the sense that there exists a bound on the number of different data values in reachable configurations; Rosa-Velardo and de Frutos-Escrig [21] call this *bounded width*. Similarly, there may exist some bound on the multiplicities with which any data value occurs, while the number of different data values is unbounded; Rosa-Velardo and de Frutos-Escrig [21] call this *bounded depth*.

We formalise the resulting decision problems in our notation as follows. The *place width boundedness* problem is given as:

Input: An UDPN, a configuration f, and a set of coordinates $K \subseteq \{1, \ldots, k\}$.
Question: Is $\{|\mathit{Supp}_{\mathbf{0}}(g{\restriction}_K)| \mid g \in \mathit{Reach}(f)\}$ finite?

The *place depth boundedness* problem is the following:

Input: An UDPN, a configuration f, and a set of coordinates $K \subseteq \{1, \ldots, k\}$.
Question: Is $\{g{\restriction}_K(d) \mid g \in \mathit{Reach}(f), d \in \mathbb{D}\}$ finite?

If the answer to the depth (width) boundedness problem is positive we call those components $i \in K$ depth (width) bounded.

Example 2.4. From the initial configuration $f_0 = \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$, the UDPN from Ex. 2.2 can reach any configuration of the form $\left[\begin{smallmatrix}n\\1\end{smallmatrix}\right]$ in n many t_1-steps, all exercised on the same data value. Similarly, any configuration of the form $\left[\begin{smallmatrix}1 & 1 & \cdots & 1\\1 & 0 & \cdots & 0\end{smallmatrix}\right]$ with a support of size $n+1$ can be reached after a sequence of n transitions t_1, each exercised on a different data value. Consequently, the first component is neither depth nor width bounded.

However, any reachable configuration g will satisfy $\sum_{d \in \mathbb{D}} g(d)[2] = 1$. In Petri net parlance, there is always exactly one token in the second place. The system is thus place bounded for $K \stackrel{\text{def}}{=} \{2\}$.

The main contribution of this paper is the effective computability of a suitable abstraction of the classical coverability tree construction [12] for UDPNs. This provides a way to decide all variants of the boundedness problem mentioned above. We summarise the consequences of our construction below.

Theorem 2.5. *In UDPNs, place depth boundedness implies place width boundedness. In consequence, place depth boundedness coincides with place boundedness.*

Theorem 2.6. *The boundedness, place boundedness, and place width boundedness problems for UDPNs are in $\mathbf{F}_{\omega^\omega}$, i.e. in* HyperAckermann.

Let us emphasise the importance of decidability of place boundedness: first, the problem is undecidable in all the extensions of UDPNs in Fig. 1. Moreover, in the case of Petri nets, the decidability of place boundedness plays a crucial role in the decidability proofs for reachability [18, 13, 14, 16], hence Thm. 2.6 provides one of the basic building blocks for future attempts at proving the decidability of reachability for UDPNs.

3 Simple Ideals

A key observation about all decision problems mentioned in the previous section is that they do not require computing the reachability set: they can all be solved given some suitable representation of the *cover* [9], defined as

$$Cover(f) \stackrel{\text{def}}{=} {\downarrow} Reach(f), \tag{4}$$

for f the initial configuration. Indeed, coverability reduces to checking whether $g \in Cover(f)$, boundedness to checking whether $Cover(f)$ is finite up to permutation of $\mathbb{D}$, and place boundedness to checking whether $\{g{\restriction_K} \mid g \in Cover(f)\}$ is finite up to permutation of $\mathbb{D}$. The main property of the coverability tree we construct in Sec. 4 is that we can extract a suitable representation of $Cover(f)$.

Ideals and Clovers. We refer the reader to the work of Finkel and Goubault-Larrecq [8, 9] for details; it suffices to say that downwards-closed sets of configurations can be represented as finite unions of so-called *configuration ideals*. Formally, a configuration ideal J is a non-empty, downwards-closed, and *directed* set of configurations; this last condition means that, if f and f' are configurations in J, then there exists h in J with $f \sqsubseteq h$ and $f' \sqsubseteq h$. Crucially for algorithmic considerations, a configuration ideal J can in turn be *represented* as the downward-closure

$$J = {\downarrow}g \stackrel{\text{def}}{=} \{h \in \mathit{Confs} \mid h \sqsubseteq g\} \tag{5}$$

of a non-negative data vector g having a finite range: $g(\mathbb{D})$ is a finite subset of $\mathbb{N}_\omega^k$. We can check that every such ${\downarrow}g$ is a configuration ideal (see [8, 9] for the converse): it is non-empty and downwards-closed by definition, and we can check it is also directed. Indeed, if f, f' are configurations and π, π' are injections with $f \leq g\pi$ and $f' \leq g\pi'$, then since f and f' are finitely supported we can assume π and π' to be permutations, and we can define a configuration $h \leq g$ such that $f \leq h\pi$ and $f' \leq h\pi'$: set h as the pointwise least upper bound $h(d)[i] \stackrel{\text{def}}{=} \max(f(\pi^{-1}(d))[i], f'({\pi'}^{-1}(d))[i]) \leq g(d)[i]$ for all d and $1 \leq i \leq k$.

Cover(f), being downwards-closed, is represented by a finite set of representations of configuration ideals, called *Clover*(f) by Finkel and Goubault-Larrecq:

$$\mathit{Cover}(f) = \bigcup\{{\downarrow}g \mid g \in \mathit{Clover}(f)\} .$$

Clover(f) is determined uniquely up to permutation, and contains $\sqsubseteq$-maximal data vectors g satisfying ${\downarrow}g \subseteq \mathit{Cover}(f)$; for further details see [8, 9]. In the following we identify a configuration ideal $J = {\downarrow}g$ with its representation g.

Remark 3.1 (Ideals for Petri nets). For readers familiar with Karp and Miller's coverability trees for Petri nets, observe that configuration ideal representations generalise the notion of 'extended markings', which are vectors in $\mathbb{N}_\omega^k$. Also, *Clover*(f) for a Petri net can be computed as the set of vertex labels in its coverability tree—this will also be our case.

Simple Ideals. Crucially, it turns out that we do not need general configuration ideals for our coverability trees for UDPNs. We only need to consider the downward-closures ${\downarrow}g$ of non-negative vectors g (cf. (5)), where the set of vectors appearing infinitely often as $g(d)$, when d ranges over $\mathbb{D}$, is a singleton $\{\boldsymbol{I}\}$ for some vector $\boldsymbol{I}$ in $\{0, \omega\}^k$ (instead of a finite subset of $\mathbb{N}_\omega^k$ for general configuration ideals). Put differently, given such a vector $\boldsymbol{I}$, we define the $\boldsymbol{I}$*-support* of a data vector f as $\mathit{Supp}_{\boldsymbol{I}}(f) \stackrel{\text{def}}{=} \{d \in \mathbb{D} \mid f(d) \neq \boldsymbol{I}\}$, and define an $\boldsymbol{I}$*-simple ideal (representation)* as a non-negative data vector with finite $\boldsymbol{I}$-support. In particular, a finitely supported non-negative data vector is a $\mathbf{0}$-simple ideal. We write M, N, ... to denote simple ideals. A simple ideal M can be represented concretely as a pair $M = \langle m, \boldsymbol{I}\rangle$ where m is the finite multiset of vectors in $\mathbb{N}_\omega^k$ obtained from M by restriction to its $\boldsymbol{I}$-support.

Example 3.2. We represent simple ideals similarly as configurations, using the additional last column for the $\boldsymbol{I}$ part. Continuing with the UDPN of Ex. 2.2, its cover is the downward-closure of a single $\boldsymbol{I}$-simple ideal:

$$\mathit{Clover}(f_0) = \left[\begin{smallmatrix}\omega\\1\end{smallmatrix}\middle|\begin{smallmatrix}\omega\\0\end{smallmatrix}\right] , \qquad\qquad \mathit{Cover}(f_0) = {\downarrow}\left[\begin{smallmatrix}\omega\\1\end{smallmatrix}\middle|\begin{smallmatrix}\omega\\0\end{smallmatrix}\right] ,$$

where $\boldsymbol{I} \stackrel{\text{def}}{=} (\omega, 0)$. The $\boldsymbol{I}$-support of the ideal has one element, mapped to $(\omega, 1)$.

Note that UDPN steps map $\boldsymbol{I}$-simple ideals to $\boldsymbol{I}$-simple ideals. Lemma 3.3 formally states the relation between steps of ideals and steps of configurations in the downward closures. The next lemma shows that $\boldsymbol{I}$-simple ideals can only have finitely many successors up to permutation. This property will later be used to define coverability trees of finite branching degree.

Lemma 3.3. *Let M, M' be $\boldsymbol{I}$-simple ideals such that $M \to M'$. Then for every configuration $c' \in \downarrow M'$ there exist configurations $c \in \downarrow M$ and $c'' \in \downarrow M'$ with $c \to c''$ and $c' \sqsubseteq c''$.*

Proof. Suppose $M \xrightarrow{t} M'$ for a finite data vector t. The data vector $f \stackrel{\text{def}}{=} c' - t$ satisfies $f \leq M$ but $f(d)$ can possibly be negative for some data value d; therefore we can not simply put $c \stackrel{\text{def}}{=} f$. A way to fix this is to define c by

$$c(d)[i] \stackrel{\text{def}}{=} \max(0, x(d)[i]), \qquad \text{for all } d \in \mathbb{D} \text{ and all coordinates } i.$$

Thus defined, c satisfies $c \leq M$, and setting $c'' \stackrel{\text{def}}{=} c + t$ satisfies $c' \leq c''$ as required. □

Lemma 3.4. *Let M be a simple ideal. The set $\{N \mid M \to N\}$ of successors of M is finite up to permutation and has a representative with cardinality bounded by $(|Supp_{\boldsymbol{I}}(M)| + \max_{t \in \mathcal{T}} |Supp_{\mathbf{0}}(t)|)! \cdot |\mathcal{T}|^2$.*

Proof. Consider an UDPN defined by the finite set $\mathcal{T}$ of data vectors. Fix an $\boldsymbol{I}$-simple ideal M, and denote by S the $\boldsymbol{I}$-support of M. We will be now considering *S-permutations*, by which we mean those data permutations π that satisfy $\pi(d) = d$ for all $d \in S$. Equality and finiteness up to S-permutation can be defined exactly as for plain permutations.

A crucial but simple observation is that the set of transitions of the UDPN is finite up to S-permutation. Indeed, assume wlog. that S is disjoint from the supports of all vectors in $\mathcal{T}$. Consider the finite set $\mathcal{T}'$ that contains all data vectors $t\sigma$, where $t \in \mathcal{T}$ and permutation σ swaps some subset of S with some subset of the support of t. Then every transition of the UDPN is of the form $t'\pi$, where $t' \in \mathcal{T}'$ and π is an S-permutation. Regarding the size of this new UDPN, there are at most $\sum_{t \in \mathcal{T}} (|S| + |Supp_{\mathbf{0}}(t)|)!$ such permutations σ, hence $|\mathcal{T}'| \leq \sum_{t \in \mathcal{T}} (|S| + |Supp_{\mathbf{0}}(t)|)! \cdot |\mathcal{T}|$.

Now we use the extension, to simple ideals, of the closure of the step relation under permutations, cf. Eq. (2), to derive a strengthening of our claim, namely finiteness of the successors of M up to S-permutations. Consider an arbitrary step $M \xrightarrow{t'\pi} N$ of M; by Eq. (2) we get

$$M = M\pi^{-1} \xrightarrow{t'} N\pi^{-1}$$

(the equality holds as π is an S-permutation). Therefore N is equal up to S-permutation to some $\mathcal{T}'$-successor of M. As $\mathcal{T}'$ is finite, the set of $\mathcal{T}'$-successors of M is finite and bounded by $|\mathcal{T}'|$, which implies our claim. □

A consequence of our construction of coverability trees in Sec. 4, and of the complexity analysis conducted in Sec. 5, is the following core result:

Theorem 3.5. *Given an UDPN and an initial configuration f, an ideal representation $Clover(f)$ of $Cover(f)$ is computable in $\mathbf{F}_{\omega^\omega}$. Furthermore, $Clover(f)$ contains only simple ideals.*

Theorem 3.5, together with the following proposition, easily imply Theorems 2.5 and 2.6 (below $Clover(f)\restriction_K \stackrel{\text{def}}{=} \{g\restriction_K \mid g \in Clover(f)\}$):

Proposition 3.6. *Fix $K \subseteq \{1, \ldots, k\}$. An UDPN is width-bounded iff $Clover(f)\restriction_K$ contains only finitely supported vectors. An UDPN is depth-bounded iff $Clover(f)\restriction_K$ contains only finite vectors.*

Proof. The former equivalence, as well as the if direction of the latter one, follow by finiteness of $Clover(f)\restriction_K$. It remains to argue that place depth-boundedness forces $Clover(f)\restriction_K$ to contain only finite vectors. Indeed, a non-finite simple ideal has necessarily ω at some component, which implies depth-unboundedness. □

The remaining part of the paper is devoted to the proof of Thm. 3.5. In Sec. 4, we present an algorithmic construction of the coverability tree, and show its termination and correctness. Then in Sec. 5 we provide upper and lower bounds on the size of the coverability tree.

4 Representing a Cover

We will show that, analogously to the classical construction of Karp and Miller [12] for vector addition systems, the cover set of any UDPN configuration can be effectively represented in the form of a finite *coverability tree*, where nodes are labelled by simple ideals.

For a given initial ideal the construction of a coverability tree amounts to iteratively computing successors (up to permutation), applying symbolic *acceleration* steps when a strictly dominating pair $M \sqsubset M'$ appears on a branch, and terminating a branch if a label embeds into one of its ancestors.

4.1 Accelerations

The idea behind acceleration steps is that due to monotonicity (Lem. 2.3), any finite sequence of steps

$$M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} \ldots \xrightarrow{t_k} M_k \tag{6}$$

such that $M_0 \sqsubset M_k$ can be extended indefinitely. Such an unfolding may have two distinct kinds of effect: Firstly, it may unboundedly increase components in data values already contained in the initial $\boldsymbol{I}$-support (we call this effect *depth acceleration*). Secondly, it may increase an unbounded number of 'fresh' data values, outside of the initial $\boldsymbol{I}$-support (we call this effect *width acceleration*).

Our construction only accelerates increasing sequences as above when there is a permutation (i.e. bijective) embedding of M_0 into M_k.

As a building block we shall use the usual vector acceleration: for two non-negative vectors $\boldsymbol{v}, \boldsymbol{v}' \in \mathbb{N}_\omega^k$ with $\boldsymbol{v}' \leq \boldsymbol{v}$, define a new vector $\mathrm{acc}(\boldsymbol{v}', \boldsymbol{v})$, for $1 \leq i \leq k$ by:

$$\mathrm{acc}(\boldsymbol{v}', \boldsymbol{v})[i] \stackrel{\text{def}}{=} \begin{cases} \boldsymbol{v}[i], & \text{if } \boldsymbol{v}'[i] = \boldsymbol{v}[i], \\ \omega, & \text{if } \boldsymbol{v}'[i] < \boldsymbol{v}[i]. \end{cases}$$

Definition 4.1 (Depth and Width Acceleration). *For $\boldsymbol{I}$-simple ideals M', M and a permutation π with $M' < M\pi$, or equivalently $M'\pi^{-1} < M$, the* depth acceleration *of M', M, π is the $\boldsymbol{I}$-simple ideal defined by*

$$M_{depth}(d) \stackrel{def}{=} acc(M'(\pi^{-1}(d)), M(d)), \qquad \text{for all data values } d \in \mathbb{D}.$$

For $d \in \mathbb{D}$ such that $M'(\pi^{-1}(d)) = \boldsymbol{I} < M(d)$, put $\boldsymbol{I}_d \stackrel{def}{=} acc(M'(\pi^{-1}(d)), M(d))$; the width acceleration *of M', M, π, d is the $\boldsymbol{I}_d$-simple ideal defined by*

$$M_{width}(d) \stackrel{def}{=} \begin{cases} \boldsymbol{I}_d, & \text{if } M(d) = \boldsymbol{I} \\ M(d), & \text{otherwise,} \end{cases}$$

By definition, $M < M_{\text{depth}}, M_{\text{width}}$.

4.2 Coverability Trees

By Lem. 3.4 we can compute for any $\boldsymbol{I}$-simple ideal M a *successor representative*, namely a finite set such that every successor of M is equal up to permutation to some element of this set.

For the sake of simplicity, we choose a conservative policy of application of accelerations: first, a proper nesting is imposed, in the sense that two different accelerated paths are either disjoint, or contained one in the other; second, a depth-accelerated path can not contain another accelerated path, while a width-accelerated path can. However, as width accelerations strictly increase the $\boldsymbol{I}$ part, a width-accelerated path is never contained in another accelerated path. Therefore the only allowed inclusion is when a depth-accelerated path is included in a width-accelerated one.

Definition 4.2 (Coverability Tree). *A* coverability tree *is a tree with nodes labelled by simple ideals such that the following criteria are satisfied.*

1. *A node with label N is a leaf iff it has an ancestor with label $N' \sqsupseteq N$.*
2. *Otherwise, suppose an interior node N has an ancestor N' such that both N', N are $\boldsymbol{I}$-simple and $N' \sqsubset N$. Let $\mathcal{P}$ denote the path from N' to N in the tree, including N' and N.*
   (a) *Suppose $N'(\pi^{-1}(d)) = \boldsymbol{I} < N(d)$ for some permutation π with $N'\pi < N$ and $d \in \mathbb{D}$; and for every node in $\mathcal{P}$ that is a depth acceleration of some nodes M', M, both M' and M belong to $\mathcal{P}$. Then N has exactly one child labelled by the width acceleration of N', N, π, d.*

(b) Otherwise, if $\mathcal{P}$ contains no acceleration then N has exactly one child labelled by the depth acceleration of N', N, π, for some permutation π with $N'\pi < N$.

3. Otherwise, if a node N satisfies none of the above criteria then its set of children is the successor representative of N.

Remark 4.3. Note that Def. 4.2 does not determine the coverability tree unambiguously: the choice of a permutation π in points 2(a) and 2(b) is not unique.

Remark 4.4. The condition in point 1 in Def. 4.2 implies that no branch of a coverability tree contains two different nodes with the same label. We identify a node with its label in the sequel.

Example 4.5. We pick some coverability tree for the UDPN from Ex. 2.2 rooted in the configuration f_0. There is a branch with labels (up to permutation)

$$f_0 = \left[\begin{smallmatrix}1\\1\end{smallmatrix}\middle|\begin{smallmatrix}0\\0\end{smallmatrix}\right],\ f_1 = \left[\begin{smallmatrix}2\\1\end{smallmatrix}\middle|\begin{smallmatrix}0\\0\end{smallmatrix}\right],\ f_2 = \left[\begin{smallmatrix}\omega\\1\end{smallmatrix}\middle|\begin{smallmatrix}0\\0\end{smallmatrix}\right],\ f_3 = \left[\begin{smallmatrix}\omega & 1\\0 & 1\end{smallmatrix}\middle|\begin{smallmatrix}0\\0\end{smallmatrix}\right],\ f_4 = \left[\begin{smallmatrix}\omega & 1\\0 & 1\end{smallmatrix}\middle|\begin{smallmatrix}\omega\\0\end{smallmatrix}\right],\ f_5 = \left[\begin{smallmatrix}\omega & 1 & 1\\0 & 1 & 0\end{smallmatrix}\middle|\begin{smallmatrix}\omega\\0\end{smallmatrix}\right],$$

where f_2 is a depth acceleration (of f_0, f_1), f_4 is a width acceleration (of f_0, f_3), and all other nodes are the result of successor steps from their parent. The node f_5 is a leaf because $f_5 \sqsubseteq f_4$. A full coverability tree can be found in App. A.

Correctness. A coverability tree is finite (termination), and represents the cover of its root node (completeness and soundness). These required properties are proven in detail in App. B:

- *Finiteness* is proven by first exhibiting a wqo for the specific type of $\boldsymbol{I}$-simple ideals that appears on coverability trees. This wqo depends on the existence of permutation embeddings, a property that on its own does *not* induce a well-quasi-ordering over the set of all $\boldsymbol{I}$-simple ideals; see Rem. D.2. Our termination argument is further refined in App. D.4 to derive complexity bounds; see Sec. 5.2.
- The *completeness* proof relies on the monotonicity of steps over simple ideals, and shows that all the elements in $\mathit{Cover}(f)$ are covered by some simple ideal in any coverability tree.
- *Soundness* is the most delicate property to establish. Its crux is that neither width nor depth accelerations may take us outside the cover of the initial configuration.

5 Complexity Bounds

In the section, we prove lower and upper bounds on the resources needed by the construction of the coverability tree. We refer the reader to [25, 24] for gentle introductions to the techniques employed to prove these results. The enormous complexities involved in our construction require to use *fast-growing complexity* classes [23], which we present succinctly in Sec. 5.1 and in more details in App. C, before showing hyper-Ackermannian upper and lower bounds in sections 5.2 and 5.3.

5.1 Fast-Growing Complexity

In order to express the non-elementary functions required for our complexity statements, we shall employ a family of subrecursive functions $(h^\alpha)_\alpha$ indexed by ordinals α known as the *Hardy hierarchy*.

Ordinal Terms. We use ordinal terms α in *Cantor Normal Form* (CNF), which can be written as terms $\alpha = \omega^{\alpha_1} + \cdots + \omega^{\alpha_n}$ where $\alpha_1 \geq \cdots \geq \alpha_n$ are themselves written in CNF. Using such notations, we can express any ordinal below ε_0, the minimal fixpoint of $x = \omega^x$. The ordinal 0 is obtained when $n = 0$; otherwise if $\alpha_n = 0$ the ordinal α is a *successor* ordinal $\omega^{\alpha_1} + \cdots + \omega^{\alpha_{n-1}} + 1$, and if $\alpha_n > 0$ the ordinal α is a *limit* ordinal. We usually write λ to denote limit ordinals.

Fundamental Sequences. For all x in $\mathbb{N}$ and limit ordinals λ, we use a standard assignment of fundamental sequences $\lambda(0) < \lambda(1) < \cdots < \lambda(x) < \cdots < \lambda$ with supremum λ. Fundamental sequences are defined by transfinite induction by:

$$(\gamma + \omega^{\beta+1})(x) \stackrel{\text{def}}{=} \gamma + \omega^\beta \cdot (x+1) \;, \qquad (\gamma + \omega^{\lambda'})(x) \stackrel{\text{def}}{=} \gamma + \omega^{\lambda'(x)} \;. \tag{7}$$

For instance, $\omega(x) = x + 1$, $\omega^2(x) = \omega \cdot (x+1)$, $\omega^\omega(x) = \omega^{x+1}$, etc.

The Hardy Hierarchy. Let $h\colon \mathbb{N} \to \mathbb{N}$ be a strictly increasing function. The Hardy functions $(h^\alpha\colon \mathbb{N} \to \mathbb{N})_\alpha$ are defined by transfinite induction on their ordinal indices by

$$h^0(x) \stackrel{\text{def}}{=} x \;, \qquad h^{\alpha+1}(x) \stackrel{\text{def}}{=} h^\alpha(h(x)) \;, \qquad h^\lambda(x) \stackrel{\text{def}}{=} h^{\lambda(x)}(x) \;. \tag{8}$$

Observe that $h^k(x)$ for a finite k is simply the kth iterate of h. For limit ordinals λ, $h^\lambda(x)$ performs a form of diagonalisation: for instance, setting $H(x) \stackrel{\text{def}}{=} x + 1$ the successor function, $H^\omega(x) = H^{x+1}(x) = 2x + 1$, $H^{\omega^2}(x) = 2^{x+1}(x+1) - 1$ is a function of exponential growth, while H^{ω^3} is a non elementary function akin to a tower of exponentials of height x, H^{ω^ω} is a non primitive-recursive function with growth similar to the Ackermann function, and $H^{\omega^{\omega^\omega}}$ is a non multiply-recursive function characteristic of hyper-Ackermannian complexity.

Complexity Classes. Following [23], we can define complexity classes for computations with time or space resources bounded by Hardy functions of the size of the input. We concentrate in this paper on the HyperAckermann complexity class. Let FMR denote the set of multiply-recursive functions and let h be any multiply-recursive strictly increasing function, then [23, Thm. 4.2]:

$$\textsc{HyperAckermann} \stackrel{\text{def}}{=} \mathbf{F}_{\omega^\omega} = \bigcup_{m \in \mathrm{FMR}} \mathrm{D}\textsc{Time}(h^{\omega^{\omega^\omega}}(m(n))) \tag{9}$$

is the set of decision problems solvable with resources bounded by an hyper-Ackermannian function applied to a multiply-recursive function m of the size of the input. This class is closed under multiply-recursive reductions, and several problems are known to be complete for it (see Sec. 6.2 of [23] for a survey), including coverability in unordered data nets [20].

5.2 Upper Bounds

We focus on the worst-case *norm* of the constructed simple ideals, from which bounds on the total size of the coverability tree and the complexity upper bound in Thm. 3.5 can both be derived.

Norms of Simple Ideals. For a vector $\boldsymbol{u}$ in $\mathbb{Z}_\omega^k$, its *norm* is its maximal finite absolute value: $\|\boldsymbol{v}\| \stackrel{\text{def}}{=} \max\{|\boldsymbol{v}[i]| \mid 1 \leq i \leq k \wedge \boldsymbol{v}[i] \neq \omega\}$. Observe that, if $\boldsymbol{I}$ is in $\{0,\omega\}^k$, then $\|\boldsymbol{I}\| = 0$. For an $\boldsymbol{I}$-simple ideal M, and thus for finitely supported ones in particular, we define its norm as the maximum between the cardinality of its support and the maximal norm of its vectors: $\|M\| \stackrel{\text{def}}{=} \max\{|\mathit{Supp}_{\boldsymbol{I}}(M)|, \|M(d)\| \mid d \in \mathbb{D}\}$. Note that the vectors for data d outside the support have all norm 0. In App. D, we exhibit a bound $B \stackrel{\text{def}}{=} h^{\omega^{\omega^{k+3}}}(\|f_0\|)$ on the norms of all the simple ideals constructed in a coverability tree rooted by f_0 as defined in Def. 4.2, where h also depends on the UDPN:

Theorem 5.1. *The norms of the simple ideals in a coverability tree rooted in a configuration f_0 for a k-dimensional UDPN $\mathcal{T}$ are bounded by $h^{\omega^{\omega^{k+3}}}(\|f_0\|)$, where $h(x)$ is an elementary function of x, k, and $\|\mathcal{T}\|$.*

The main technical ingredients for Thm. 5.1 are combinatorial statements on the lengths of so-called *controlled bad sequences* proven by Rosa-Velardo [20, App. A] for finite multisets of vectors of natural numbers. Our proofs require however a substantial amount of work on top of that of Rosa-Velardo's for two reasons: we work with extended vectors in $\mathbb{N}_\omega^k$, and use permutation embeddings rather than just plain embeddings.

Relating Norms with Sizes and Complexity. The norm $\|M\| \leq B$ of a simple ideal M is directly related to the size of its concrete binary representation: the latter needs at most $\|M\| \cdot k \cdot (\lceil \log \|M\| \rceil + 1)$ bits for the $\boldsymbol{I}$ supported part of the ideal and k bits for the $\boldsymbol{I}$ vector itself. We can also bound the length of the branches in our coverability trees: there are indeed at most $(B+2)^{kB} \cdot 2^k$ different simple ideals with norm $\leq B$, and no two interior nodes on a branch are labelled by the same ideal due to condition 1 in Def. 4.2 (see Rem. 4.4). Finally, by Lem. 3.4, the branching degree of the coverability tree is bounded by an exponential function $(B + \|\mathcal{T}\|)! \cdot |\mathcal{T}|^2$ in B and the size of $\mathcal{T}$. These three observations combined allow to bound the size of the coverability tree:

Theorem 5.2 (Size of Coverability Trees). *The size of a coverability tree built from an initial configuration f_0 for a k-dimensional UDPN $\mathcal{T}$ is bounded by an elementary function of B, k, and the size of $\mathcal{T}$.*

Theorem 5.2 along with Eq. (9) and the completeness and soundness of coverability trees as shown in lemmata B.4 and B.10 yields the proof of Thm. 3.5, using the fact that $\mathbf{F}_{\omega^\omega}$ is closed under elementary reductions [23, Thm. 4.7].

5.3 Lower Bounds

The sheer complexity bounds we just obtained on the size of coverability trees beg the question whether they are the best possible. We show in Thm. 5.3 that, indeed, the size of coverability trees for a family of UDPNs is provably non multiply-recursive, matching essentially the statement of Thm. 5.2:

Theorem 5.3 (Hyper-Ackermannian Coverability Trees). *There exists families of $O(k)$-sized UDPNs $(\mathcal{T}_k)_k$ and $O(k+\log n)$-sized initial configurations $(f_{k,n})_{k,n}$, whose coverability trees are of size at least $H^{\omega^{\omega^k}}(n)$.*

Hardy Computations. As detailed in App. E, we prove Thm. 5.3 by 'implementing' the computation of Hardy functions $H^{\omega^{\omega^k}}$ in nets $\mathcal{T}_k$. The main idea, first developed in [11, 24], is to see the equations in (8) for $0 < \alpha$ as rewriting rules operating on pairs (α, n):

$$(\alpha+1, n) \to (\alpha, n+1)\ , \qquad\qquad (\lambda, n) \to (\lambda(n), n)\ . \tag{10}$$

Note that a sequence $(\alpha_0, n_0) \to (\alpha_1, n_1) \to \cdots \to (\alpha_i, n_i) \to \cdots$ of rewriting steps maintains $H^{\alpha_i}(n_i) = H^{\alpha_0}(n_0)$ for all i, and must eventually terminate at some rank ℓ with $\alpha_\ell = 0$ since $\alpha_i > \alpha_{i+1}$ for all i, and then $n_\ell = H^{\alpha_0}(n_0)$.

Using a natural representation of ordinals $\alpha < \omega^{\omega^k}$ as finite multisets of vectors also employed by Rosa-Velardo [20], a pair (α, n) can be encoded as a configuration of $\mathcal{T}_k$, and the rewriting rules of (10) can be implemented on such codes by steps of $\mathcal{T}_k$. This is however not a perfect implementation: many incorrect computations yielding results different from $H^{\omega^{\omega^k}}(n) = H^{\omega^{\omega^{k-1}\cdot(n+1)}}(n)$ are possible. The crucial point is that there exists a *perfect* computation in $\mathcal{T}_k$, of length at least $H^{\omega^{\omega^k}}(n)$. Furthermore, this computation does not allow any acceleration step, and has therefore to occur as such in any coverability tree.

6 Concluding Remarks

In this paper, we have presented a procedure to construct coverability trees for UDPNs in the style of Karp and Miller [12]. This yields decision procedures for coverability and several variants of the boundedness problem including place-boundedness, depth- and width place-boundedness. Besides its interest for the formal verification of UDPNs, this paves the way towards future attempts at proving the decidability of reachability along the lines developed for Petri nets in [18, 13, 14, 16].

We have derived hyper-Ackermannian upper bounds on the complexity of our construction, and shown that such enormous complexities are actually attained on some UDPNs. Note that this however does *not* provide a lower bound

- on the size of $\mathit{Clover}(f)$, for which the best known bound is an Ackermannian lower bound adapted from the case of Petri nets [4], nor

– on the complexity of the various boundedness problems on UDPNs, for which the best lower bound is hardness for TOWER = $\mathbf{F}_3$, adapted from the coverability problem [15].

We actually suspect that much lower complexities that HYPERACKERMANN could be obtained for the coverability and boundedness problems. For instance, in the case of Petri nets, coverability trees have a worst-case Ackermannian size [4, 6], but coverability, boundedness, and place-boundedness are all EXPSPACE-complete [17, 19, 2, 5].

References

1. Abdulla, P.A., Čerāns, K., Jonsson, B., Tsay, Y.K.: Algorithmic analysis of programs with well quasi-ordered domains. Inform. and Comput. 160(1–2), 109–127 (2000)
2. Blockelet, M., Schmitz, S.: Model-checking coverability graphs of vector addition systems. MFCS 2011. LNCS, vol. 6907, pp. 108–119. Springer (2011)
3. Bojańczyk, M., Klin, B., Lasota, S.: Automata theory in nominal sets. Logic. Meth. in Comput. Sci. 10(3:4), 1–44 (2014)
4. Cardoza, E., Lipton, R.J., Meyer, A.R.: Exponential space complete problems for Petri nets and commutative semigroups: Preliminary report. STOC'76. pp. 50–54. ACM (1976)
5. Demri, S.: On selective unboundedness of VASS. J. Comput. Syst. Sci. 79(5), 689–713 (2013)
6. Figueira, D., Figueira, S., Schmitz, S., Schnoebelen, Ph.: Ackermannian and primitive-recursive bounds with Dickson's Lemma. LICS 2011. pp. 269–278. IEEE Press (2011)
7. Finkel, A., Schnoebelen, Ph.: Well-structured transition systems everywhere! Theor. Comput. Sci. 256(1–2), 63–92 (2001)
8. Finkel, A., Goubault-Larrecq, J.: Forward analysis for WSTS, part I: Completions. STACS 2009. LIPIcs, vol. 3, pp. 433–444. LZI (2009)
9. Finkel, A., Goubault-Larrecq, J.: Forward analysis for WSTS, part II: Complete WSTS. Logic. Meth. in Comput. Sci. 8(3:28), 1–35 (2012)
10. Finkel, A., McKenzie, P., Picaronny, C.: A well-structured framework for analysing Petri net extensions. Inform. and Comput. 195(1–2), 1–29 (2004)
11. Haddad, S., Schmitz, S., Schnoebelen, Ph.: The ordinal recursive complexity of timed-arc Petri nets, data nets, and other enriched nets. LICS 2012. pp. 355–364. IEEE Press (2012)
12. Karp, R.M., Miller, R.E.: Parallel program schemata. J. Comput. Syst. Sci. 3(2), 147–195 (1969)
13. Kosaraju, S.R.: Decidability of reachability in vector addition systems. Proc. STOC'82. pp. 267–281. ACM (1982)
14. Lambert, J.L.: A structure to decide reachability in Petri nets. Theor. Comput. Sci. 99(1), 79–104 (1992)
15. Lazić, R., Newcomb, T., Ouaknine, J., Roscoe, A., Worrell, J.: Nets with tokens which carry data. Fund. Inform. 88(3), 251–274 (2008)
16. Leroux, J., Schmitz, S.: Demystifying reachability in vector addition systems. LICS 2015. pp. 56–67. IEEE Press (2015)

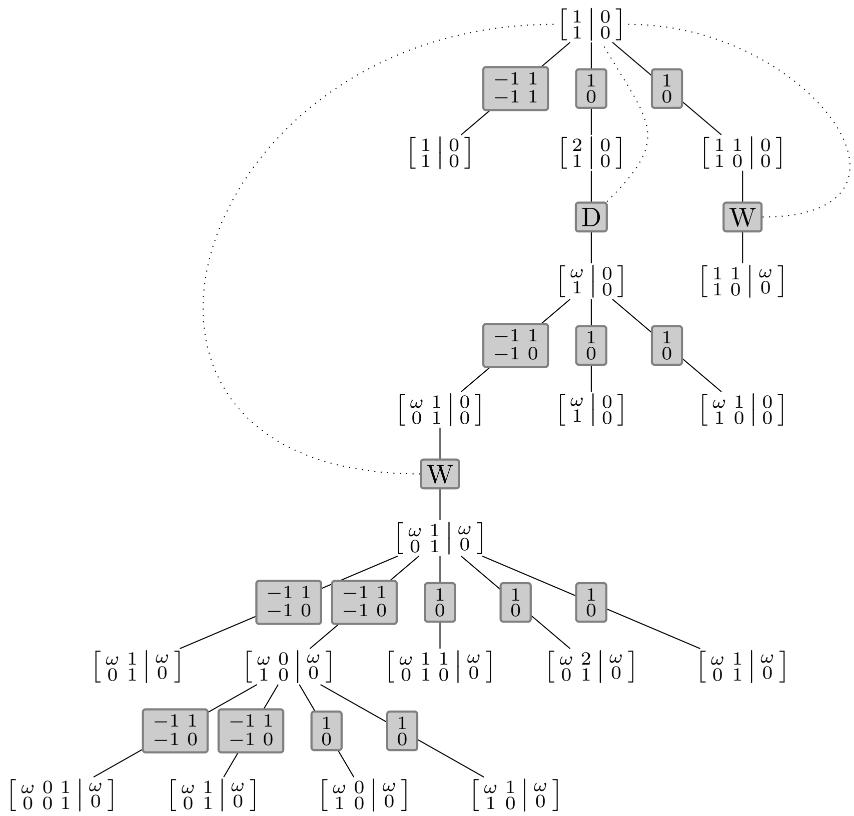
17. Lipton, R.: The reachability problem requires exponential space. Tech. Rep. 62, Yale University (1976)
18. Mayr, E.W.: An algorithm for the general Petri net reachability problem. Proc. STOC'81. pp. 238–246. ACM (1981)
19. Rackoff, C.: The covering and boundedness problems for vector addition systems. Theor. Comput. Sci. 6(2), 223–231 (1978)
20. Rosa-Velardo, F.: Ordinal recursive complexity of unordered data nets. Tech. Rep. TR-4-14, Departamento de Sistemas Informáticos y Computación, Universidad Complutense de Madrid (2014), http://antares.sip.ucm.es/frosa/docs/complexityUDN.pdf
21. Rosa-Velardo, F., de Frutos-Escrig, D.: Decidability and complexity of Petri nets with unordered data. Theor. Comput. Sci. 412(34), 4439–4451 (2011)
22. Rosa-Velardo, F., Martos-Salgado, M., de Frutos-Escrig, D.: Accelerations for the coverability set of Petri nets with names. Fund. Inform. 113(3–4), 313–341 (2011)
23. Schmitz, S.: Complexity hierarchies beyond Elementary. ACM Trans. Comput. Theory (2016), http://arxiv.org/abs/1312.5686, to appear
24. Schmitz, S., Schnoebelen, Ph.: Algorithmic aspects of WQO theory. Lecture notes (2012), http://cel.archives-ouvertes.fr/cel-00727025
25. Schmitz, S., Schnoebelen, Ph.: The power of well-structured systems. Concur 2013. LNCS, vol. 8052, pp. 5–24. Springer (2013)
26. Schnoebelen, Ph.: Revisiting Ackermann-hardness for lossy counter machines and reset Petri nets. MFCS 2010. LNCS, vol. 6281, pp. 616–628. Springer (2010)

A A Coverability Tree

We consider again the 2-dimensional UDPN over the domain $\mathbb{D} \stackrel{\text{def}}{=} \mathbb{N}$ from Ex. 2.2: it is given by vectors t_1, t_2 defined as $t_1 : 0 \mapsto (1,0)$ and $t_1 : n \mapsto (0,0)$ for all $n > 0$, and $t_2 : 0 \mapsto (-1,-1)$, $t_2 : 1 \mapsto (1,1)$ and $t_2 : n \mapsto (0,0)$ for all $n > 1$. Up to permutation, these are represented by $t_1 \stackrel{\text{def}}{=} \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$ and $t_2 \stackrel{\text{def}}{=} \left[\begin{smallmatrix}-1 & 1\\-1 & 1\end{smallmatrix}\right]$.

We are interested in the cover of the configuration f_0, with $f_0 : 0 \mapsto (1,1)$ and $f_0 : n \mapsto (0,0)$ for $n > 0$, which is a $\mathbf{0}$-simple ideal and represented (up to permutations) by $f_0 = \left[\begin{array}{c|c}1 & 0\\1 & 0\end{array}\right]$.

Depicted below is a coverability tree rooted in f_0. The grey edge labels indicate the used transition in case the edge is due to a successor step, and W/D in case the next node is a width/depth acceleration. The dotted lines indicate the beginning of accelerated path. For ease of presentation, nodes that embed into some lexicographically smaller node are not further explored.



Notice that the parent $\left[\begin{array}{cc|c}\omega & 1 & \omega\\0 & 1 & 0\end{array}\right]$ of the ideal $\left[\begin{array}{cc|c}\omega & 0 & \omega\\1 & 0 & 0\end{array}\right]$ in the second row from the bottom strictly embeds in $\left[\begin{array}{cc|c}\omega & 0 & \omega\\1 & 0 & 0\end{array}\right]$. However, there is no *permutation* embedding of $\left[\begin{array}{cc|c}\omega & 1 & \omega\\0 & 1 & 0\end{array}\right]$ into $\left[\begin{array}{cc|c}\omega & 0 & \omega\\1 & 0 & 0\end{array}\right]$ and thus no acceleration happens at this point.

B Proofs of Correctness

We gather in this appendix the omitted proofs of correctness of the construction of coverability trees.

B.1 Termination

For the termination argument we require the following lemma, which ensures that along any sufficiently long sequence of successor steps one can find ideals that are related by a permutation embedding; see App. D.2 for a proof.

Lemma B.1. *For any finite $S \subseteq \mathbb{D}$ and $\boldsymbol{I} \in \{0, \omega\}^k$, permutation embeddings well-quasi-order the set $\mathcal{M}_{S,\boldsymbol{I}} \stackrel{\text{def}}{=} \{M$ an $\boldsymbol{I}$-simple ideal $\mid M(d) \geq \boldsymbol{I}$ for all $d \notin S\}$.*

Lemma B.2 (Termination). *Every coverability tree is finite.*

Proof. By contraposition. Assume an infinite coverability tree and (by Kőnig's Lemma) an infinite branch with labels $N_0, N_1, \ldots$. First observe that for every label N_i there is exactly one vector $\boldsymbol{I}_i \in \{0, \omega\}^k$ such that N_i is $\boldsymbol{I}_i$-simple. Further, these vectors are monotone in the index: $i < j$ implies that $\boldsymbol{I}_i \leq \boldsymbol{I}_j$. Since every width acceleration strictly increases this vector, the sequence of $\boldsymbol{I}_i$'s eventually stabilises to some $\boldsymbol{I}$ and from some index n_0 onwards, all N_i for $i \geq n_0$ are $\boldsymbol{I}$-simple ideals.

We show that there are also only finitely many depth accelerations on this sequence. Notice that all nodes from index n_0 onwards are greater or equal to $\boldsymbol{I}$ outside of $\mathit{Supp}_{\boldsymbol{I}}(N_{n_0})$: For every $d \in \mathbb{D}$ and $i \geq n_0$ it holds that

$$d \notin \mathit{Supp}_{\boldsymbol{I}}(N_{n_0}) \implies N_i(d) \geq \boldsymbol{I}. \tag{11}$$

Suppose there are infinitely many depth accelerations. By Lem. B.1 we can therefore extract an infinite subsequence $(M_i)_{i \geq 0}$ of $(N_i)_{i \geq 0}$ such that every M_i is a depth acceleration and moreover, for all indices i there is some permutation embedding π_i with $M_i \leq M_{i+1}\pi_i$.

We observe that the cardinality of $\boldsymbol{I}$-supports of elements of M_i has to stabilise. Indeed, if $|\mathit{Supp}_{\boldsymbol{I}}(M_i)| < |\mathit{Supp}_{\boldsymbol{I}}(M_j)|$ for $i < j$ then there is a width acceleration of M_i, M_j, contrary to the assumption that $\boldsymbol{I}$ has stabilised.

Notice that every depth acceleration introduces at least one new ω-component because otherwise it would be a leaf node, contradicting the assumption that the branch is infinite. Formally, if we let

$$W_i \stackrel{\text{def}}{=} \{(d, l) \mid d \in \mathit{Supp}_{\boldsymbol{I}}(M_i),\ M_i(d)[l] = \omega\} \tag{12}$$

then for all $i < j$ it holds that $|W_i| < |W_j|$. This again yields a contradiction with the assumption that the sequence $(M_i)_{i \geq 0}$ is infinite, because clearly $|\mathit{Supp}_{\boldsymbol{I}}(M_0)| \cdot k$ bounds the cardinality of any W_i. We conclude that there are only finitely many depth accelerations on the sequence $(N_i)_{i \geq n_0}$.

Let us pick $n_1 \in \mathbb{N}$ large enough such that no accelerations happen after index n_1. That is, $n_1 \leq i$ implies that N_{i+1} is a successor of N_i. Now again, by Lem. B.1 we can find indices $n_1 \leq i < j$ such that $N_i \leq N_j\pi$ for some permutation embedding π. By definition of the coverability tree, this means that N_{j+1} must be the result of an acceleration. Contradiction. $\square$

B.2 Completeness

The proof of Lem. 2.3 essentially works for the more general case of arbitrary non-negative data vectors:

Lemma B.3. *Let f, f', g be non-negative data vectors such that $f \leq f'\pi$ for an embedding π. If $f \to g$, then there exists g' with $f' \to g'$ and $g \leq g'\pi$.*

Lemma B.4 (Completeness). *Let $R \xrightarrow{*} N$ for two simple ideals R, N. Every coverability tree with root R contains a node $N' \sqsupseteq N$.*

Proof. Pick any coverability tree with the root R. The tree *defines* a downward closed set $\mathcal{C}$ of configurations, namely the union of downward closures $\downarrow M$ of all simple ideals M appearing in the tree. We need to prove that $Cover(R)$ is included in $\mathcal{C}$. It is enough to prove that the set $\mathcal{C}$ (1) contains R, (2) is closed under successors, and (3) is downward closed, as the $Cover(R)$ is the smallest set satisfying the conditions (1)–(3). Conditions (1) and (3) are readily verified by $\mathcal{C}$, thus we concentrate on (2).

Let N be an arbitrary configuration in $\mathcal{C}$; therefore some node $N' \sqsupseteq N$ appears in the tree. Choose N' to be $\sqsubseteq$-maximal, and among the $\sqsubseteq$-maximal ones choose N' to be at the minimal depth in the tree. We claim that N' is not the parent of an acceleration – indeed, this would violate the $\sqsubseteq$-maximality of N', as the acceleration is strictly larger than its parent. Further, N' is not a leaf either, as in this case there would be a node $N'' \sqsupseteq N'$ on the branch from the root to N' thus violating either maximality or minimal depth of N'. Therefore the set of children of N' in the tree is a representative of the successor set of N' and for every successor M of N, by Lem. B.3 the tree contains a successor M' of N' with $M \sqsubseteq M'$. Thus $M \in \mathcal{C}$, which proves condition (2) to holds. $\square$

B.3 Soundness

We split the proof of soundness (Lem. B.10) into few separate lemmata. Lemmata B.5 and B.7 deal with depth accelerations; then Lemmata B.8 and B.9 are their analogs that deal with width accelerations.

Lemma B.5. *Let $N \xrightarrow{*} N'$ be a finite path of simple ideals such that $N \leq N'\pi$ for some permutation embedding π. Then there exists a finite path $N \xrightarrow{*} N''$ such that $N \leq N''$ and for all $d \in \mathbb{D}$ and $1 \leq i \leq k$,*

$$N(d)[i] < N'\pi(d)[i] \implies N'(d)[i] < N''(d)[i]$$

Proof. Let $N_0 \to N_1 \to \cdots \to N_l$ be a finite path of simple ideals such that $N_0 \leq N_l\pi$ for a permutation embedding π. Note that all N_i are $\boldsymbol{I}$ simple ideals for the same vector $\boldsymbol{I}$. We claim that $N_0 \leq N_l\pi'$ for some permutation embedding π' that is identity outside a finite subset. Indeed, there is only finitely many transition steps between N_0 and N_l thus $N_0(d) \neq N_l(d)$ only for finitely many data values d. Therefore π can be made identity outside of $Supp_{\boldsymbol{I}}(N_0) \cup Supp_{\boldsymbol{I}}(N_l)$, yielding π' as required.

Knowing that $N_0 \leq N_l\pi'$, we apply Lem. B.3 to all the steps in the assumed sequence $N_0 \xrightarrow{*} N_l$ to get a path $N_l \to N_{1+1} \to \cdots \to N_{2l}$ with $N_l \leq N_{2l}\pi'$. Repeating this process we get, for every $n > 0$, a simple ideal $N_{l\cdot n}$ satisfying $N_{l\cdot n} \xrightarrow{*} N_{l\cdot(n+1)}$ and $N_l \leq N_{l\cdot n}(\pi')^{n-1}$. For some n, permutation $(\pi')^n$ is necessarily the identity, and then $N'' \stackrel{\text{def}}{=} N_{l\cdot(n+1)}$ satisfies the claim. □

For two simple ideals we write $M =_\omega N$ if for every $d \in \mathbb{D}$ and $1 \leq i \leq k$, it holds that $M(d)[i] = \omega \iff N(d)[i] = \omega$.

Lemma B.6. *Suppose N is the depth acceleration of some A, P, π in the coverability tree. Then for every simple ideal C such that $A =_\omega C \leq N\pi$ there exists a simple ideal $C' \geq A$ such that $A \xrightarrow{*} C'$ and $C \sqsubseteq C'$.*

Proof. Both A and P must be $\boldsymbol{I}$-simple for the same $\boldsymbol{I}$ and there exists a finite sequence $A \xrightarrow{*} P$ of steps between them. Moreover, $A \leq P\pi \leq N\pi$. By Lem. B.5 there exists some finite sequence of steps, starting in A and ending in some $A_1 \geq A$, that strictly increases those components where N is strictly larger than A. Formally, let W be the set of pairs $(d, i) \in \mathbb{D} \times \{1, \ldots, k\}$ such that $A(d)[i] < N\pi(d)[i] = \omega$. Then for every $(d, i) \in W$ it holds that $A(d)[i] < A_1(d)[i]$. Since $A \leq A_1$, the same sequence of steps can be executed again, leading from A_1 to some $A_2 \geq A_1$ and so on. Let us denote by ρ_n the result of repeating the assumed sequence for n times. That is, ρ_n is a finite path from A to some A_n where for all $(d, i) \in W$ it holds that $A_n(d)[i] > A(d)[i] + n$. To show the claim, consider some simple ideal C with $A =_\omega C \leq N\pi$ and pick n large enough such that $C(d)[i] < n$ for all $(d, i) \in W$. Clearly this is possible because all $C(d)[i]$ are finite. Then the path ρ_n leads to a simple ideal C' that satisfies the claim. □

Lemma B.7 (Soundness of Depth Acceleration). *Suppose N is the depth acceleration of some A, P, π in the coverability tree. Then for every $c \in {\downarrow}N$ there exist configurations $a \in {\downarrow}A$ and $c' \sqsupseteq c$ such that $a \xrightarrow{*} c'$.*

Proof. Both A and P must be $\boldsymbol{I}$-simple for the same $\boldsymbol{I}$, and $A \leq P\pi \leq N\pi$. Consider a configuration $c \in {\downarrow}N$. Wlog. asume $c \leq N\pi$; indeed, every configuration $c' \in {\downarrow}N$ embeds into N by a permutation embedding, say $c' \leq N\sigma$, and thus $c' \equiv c$ for $c = c'\sigma^{-1}\pi \leq N\pi$. Make c into a simple ideal $C =_\omega A$ by putting $C(d)[i] = \omega$ whenever $A(d)[i] = \omega$, and $C(d)[i] = c(d)[i]$ otherwise. By Lem. B.6 we get a path $A \xrightarrow{*} C'$, for $C \sqsubseteq C'$. As $c \in {\downarrow}C'$, we apply Lem. 3.3 to all the steps in the path and obtain a configuration $a \in {\downarrow}A$ from which the path is executable and leads to a configuration $c' \in {\downarrow}C'$ such that $c \sqsubseteq c'$. □

Lemma B.8. *For every width acceleration of some A, P, π, d in the coverability tree, there exists a finite path $A \xrightarrow{*} A'$ such that $A \leq A'$ and for all $d \in \mathbb{D}$ and $1 \leq i \leq k$,*

$$A(d)[i] < P\pi(d)[i] \implies A(d)[i] < A'(d)[i].$$

Proof. Both A and P must be $\boldsymbol{I}$-simple for the same $\boldsymbol{I}$, and there is a finite sequence of $\boldsymbol{I}$-simple ideals $N_0, N_1, \ldots, N_l$ such that $N_0 = A$, $N_l = P$, and for every $i > 0$, the ideal N_i is either a successor of N_{i-1}, or a depth acceleration of N_{i-2}, N_{i-1} and some permutation π_i. Let A'_{min} be the smallest ideal such that $A \leq A'_{min}$ and for all $d \in \mathbb{D}$ $A(d)[i] < P\pi(d)[i] \implies A(d)[i] < A'(d)[i]$.

To build a finite path $A \xrightarrow{*} A'$ for some $A' \geq A'_m in$, we start with the empty path for N_l and successively extend the path (at the beginning) by traversing backwards through $N_{l-1}, \ldots, N_0$. Suppose we have already a path ρ for N_i. If the current node N_i is a successor of N_{i-1} then we can just extend the path ρ by the step $N_{i-1} \to N_i$. Otherwise, the current node N_i is a depth acceleration of ancestors N_{i-2}, N_{i-1} and some permutation σ, and $N_{i-2} \leq N_{i-1}\sigma \leq N_i\sigma$. Let W be the set of pairs $(d, i) \in \mathbb{D} \times \{1, \ldots, k\}$ such that $N_{i-2}(d)[i] < N_i\sigma(d)[i] = \omega$. Construct a simple ideal C from $N_i\sigma$ by replacing, for all $(d, i) \in W$, the values $N_i\sigma(d)[i] = \omega$, by non-negative numbers sufficiently large to execute the renamed path $\rho\sigma$ from C. Furthermore, the numbers should be large enough so that the resulting final value after executing ρ from C is larger than $A'_m in$. Then $N_{i-2} =_\omega C \leq N_i\sigma$ and therefore Lem. B.6 applies and yields a path $N_{i-2} \xrightarrow{*} C'$ for some $C' \geq N_{i-2}$ such that $C \sqsubseteq C'$. The path $\rho\sigma\sigma'$ obtained form $\rho\sigma$ by further renaming via some permutation σ', is thus executable from C'. Composing the just obtained path $N_{i-2} \xrightarrow{*} C'$ with $\rho\sigma\sigma'$ yields a new path from N_{i-2}.

The so constructed path from $A = N_0$ to some A' satisfies the claim. □

Lemma B.9 (Soundness of Width Acceleration). *Suppose N is the width acceleration of some A, P, π, d in the coverability tree. Then for every $c \in {\downarrow}N$ there exist configurations $a \in {\downarrow}A$ and $c' \sqsupseteq c$ such that $a \xrightarrow{*} c'$.*

Proof. Then both A and P must be $\boldsymbol{I}$-simple for the same $\boldsymbol{I}$, and N is $\boldsymbol{I}_*$-simple for some $\boldsymbol{I}_* > \boldsymbol{I}$. Moreover, it holds that $A \leq P\pi$ for some permutation and there exists $d_* \in Supp_{\boldsymbol{I}}(P\pi) \setminus Supp_{\boldsymbol{I}}(A)$ with $A(d_*)[i] < P\pi(d_*)[i]$ for those $1 \leq i \leq k$ where $\boldsymbol{I}[i] < \boldsymbol{I}_*[i]$. Lemma B.8 thus guarantees, for any $n \in \mathbb{N}$, a finite path ρ_n from A to some $A_n \geq A$ such that $\boldsymbol{I}[i] < \boldsymbol{I}_*[i]$ implies $A_n(d_*)[i] > A(d_*)[i] + n$.

Now fix some enumeration $\{d_0, d_1, \ldots\}$ of data values outside of the $\boldsymbol{I}$-support of $P\pi$ and let B_i be defined by $B_i(d_i) \stackrel{\text{def}}{=} \boldsymbol{I}_*$ and $B_i(d) \stackrel{\text{def}}{=} \mathbf{0}$ for $f \neq d_i$. This way we can write $N\pi$ as the infinite sum $N\pi = P\pi + \sum_{i \geq 0} B_i$. Notice that every configuration $c \in {\downarrow}N$ has an equivalent $c \equiv c' \leq P\pi + \sum_{i=0}^{n} B_i$ for some n. It therefore suffices to show that for every configuration $c \leq P\pi + \sum_{i=0}^{n} B_i$ there exists configurations $r \leq A$ and c' such that $r \xrightarrow{*} c' \geq c$. The crucial observation is, intuitively, that the sequence of transitions leading from A to A_n is also executable if one increases some datum d_i instead of d_*. More precisely, as steps are closed under permutations (cf. Eq. (2)), if $A \xrightarrow{t_1}\xrightarrow{t_2} \cdots \xrightarrow{t_n} A_n$, then also

$A\sigma_i \xrightarrow{t_1\sigma_i}\xrightarrow{t_2\sigma_i} \cdots \xrightarrow{t_n\sigma_i} A_n\sigma_i$, for a permutation σ_i that exchanges d_* and d_i. Let's write $\rho_{n,i}$ for the path from above. Notice that $A\sigma_i \equiv A$ and the endpoint $A_n\sigma_i$ is pointwise larger than A. Consequently, we can successively execute any sequence of such paths (for different n and i), starting from A. In particular, for any $n, m \in \mathbb{N}$ there exists a finite sequence of steps, executable from A, that ends in some $A_{n,m} \geq A$ where $A_{n,m}(d_j)[i] \geq m$ for those $j \leq n$ and coordinates $1 \leq i \leq k$ with $\boldsymbol{I}[i] = 0 < \omega = \boldsymbol{I}_*[i]$.

To show the claim, take a configuration $c \leq P\pi + \sum_{i=0}^{n} B_i$ and let $m \in \mathbb{N}$ be large enough such that $c(d)[i] \leq m$ for all $d \in \mathbb{D}$ and $1 \leq i \leq k$. Consider the finite path as constructed above for our chosen n, m. Again, since this path is finite, we find a bound on the negative effect of this path on any coordinate and thus find some configuration $a \leq A$ from which the path is executable and leads to a configuration $a_{n,m} \sqsupseteq c$. $\square$

Lemma B.10 (Soundness). *Let N be a node on a coverability tree with root R. For every configuration $c \in {\downarrow}N$ there exist configurations $r \in {\downarrow}R$ and c' such that $r \xrightarrow{*} c'$ and $c \sqsubseteq c'$.*

Proof. By induction on the depth of N in the tree. The claim is trivial for $N = R$. Assume the claim holds for all ancestors of N. There are three cases, where N is either a successor of its parent, a depth or a width acceleration. The claim follows by lemmata 3.3, B.7 and B.9, respectively. $\square$

C Subrecursive Hierarchies

We recall some basic definitions and facts about subrecursive hierarchies and fast-growing complexity classes. The reader is referred to [24, 23] for more details.

C.1 Ordinal Terms

Ordinal terms below ε_0 in Cantor normal form can also be written by grouping equal summands as

$$\alpha = \omega^{\alpha_1} \cdot c_1 + \cdots + \omega^{\alpha_n} \cdot c_n \tag{13}$$

for $\alpha > \alpha_1 > \cdots > \alpha_n$ and coefficients $0 < c_1, \ldots, c_n < \omega$. Using this representation, we can define the *leanness* or ordinal *norm* $N\alpha$ as its largest coefficient: $N\alpha \stackrel{\text{def}}{=} \max\{c_i, N\alpha_i \mid 1 \leq i \leq n\}$.

Besides the familiar direct sum and direct products of ordinal arithmetic, we will use the *natural sum* of two ordinals, defined on their CNF by

$$(\omega^{\alpha_1} + \cdots + \omega^{\alpha_n}) \oplus (\omega^{\beta_1} + \cdots + \omega^{\beta_m}) \stackrel{\text{def}}{=} \omega^{\gamma_1} + \cdots + \omega^{\gamma_{n+m}} \tag{14}$$

where $\gamma_1 \geq \cdots \geq \gamma_{n+m}$ is a reordering of the α_is and β_js, and their *natural product*

$$(\omega^{\alpha_1} + \cdots + \omega^{\alpha_n}) \otimes (\omega^{\beta_1} + \cdots + \omega^{\beta_m}) \stackrel{\text{def}}{=} \bigoplus_{1 \leq i \leq n, 1 \leq j \leq m} \omega^{\alpha_i \oplus \beta_j} . \tag{15}$$

For instance, although $\omega + \omega^\omega = \omega^\omega$, their natural sum is $\omega \oplus \omega^\omega = \omega^\omega + \omega$; similarly, $\omega \cdot \omega^\omega = \omega^\omega$ but $\omega \otimes \omega^\omega = \omega^{\omega+1}$.

C.2 Hardy and Cichoń Hierarchies

Cichoń Hierarchy. Given a strictly increasing $h\colon \mathbb{N} \to \mathbb{N}$, another hierarchy of subrecursive functions of interest is the *Cichoń hierarchy* $(h_\alpha\colon \mathbb{N} \to \mathbb{N})_\alpha$ defined by [27]:

$$h_0(x) \stackrel{\text{def}}{=} 0\,, \qquad h_{\alpha+1}(x) \stackrel{\text{def}}{=} 1 + h_\alpha(h(x))\,, \qquad h_\lambda(x) \stackrel{\text{def}}{=} h_{\lambda(x)}(x)\,. \tag{16}$$

This hierarchy is closely related to the Hardy hierarchy. In particular, for all α and x,

$$h^\alpha(x) = h^{h_\alpha(x)}(x)\,, \tag{17}$$

meaning that the Cichoń function measures the finite number of iterations of h involved in computing h^α. We typically employ the Cichoń hierarchy to bound the length of computations, while the Hardy hierarchy allows to bound the size of the intermediate configurations. We also have for all α and x

$$h^\alpha(x) \geq h_\alpha(x) + x\,. \tag{18}$$

Composition and Relativisation. An interesting consequence of the definition of the Hardy functions is that we can 'internalise' compositions and relativisations in the ordinal index. By transfinite induction on β, one can check that, assuming $\alpha + \beta$ is a term in CNF,

$$h^\alpha \circ h^\beta = h^{\alpha+\beta}\,. \tag{19}$$

Note that (19) does not hold if $\alpha+\beta$ is not a term in CNF; for instance, $1+\omega = \omega$ but $H^1(H^\omega(x)) = H(2x+1) = 2x+2 > H^\omega(x)$. We can however bound such cases with

$$h^\alpha \circ h^\beta \leq h^{\alpha\oplus\beta}\,, \tag{20}$$

and indeed in the previous example $1 \oplus \omega = \omega + 1$ and $H^1(H^\omega(x)) = 2x + 2 \leq H^{\omega+1}(x) = H^{x+2}(x+1) = 2x+3$.

Similarly, by transfinite induction on β, one can check that, if $\alpha \cdot \beta$ is a term in CNF,

$$(h^\alpha)^\beta = h^{\alpha\cdot\beta}\,. \tag{21}$$

Again, (21) fails if $\alpha \cdot \beta$ is not a term in CNF. For instance, $2 \cdot \omega = \omega$ but $H^2(x) = x + 2$ and $(H^2)^\omega(x) = (H^2)^{x+1}(x) = 3x + 2 > H^\omega(x) = 2x + 1$. Again, we can bound such cases by

$$(h^\alpha)^\beta \leq h^{\alpha\otimes\beta}\,, \tag{22}$$

and for instance for $2 \otimes \omega = \omega \cdot 2$, $(H^2)^\omega(x) = 3x+2 \leq H^{\omega\cdot 2}(x) = H^{\omega+x+1}(x) = H^\omega(2x+1) = H^{2x+2}(2x+1) = 4x+3$.

C.3 Fast-Growing Complexity Classes

We use the complexity classes from [23] for computations with time or space resources bounded by Hardy functions of the size of the input. In order to obtain robust classes closed under reductions using similarly bounded resources, we employ functions from the *extended Grzegorczyk* hierarchy $(\mathscr{F}_{<\alpha})_\alpha$ of Löb and Wainer [29]: for all $\alpha > 2$,

$$\mathscr{F}_{<\alpha} = \bigcup_{\beta<\omega^\alpha} \text{FDT}\textsc{ime}(H^\beta(n)) \tag{23}$$

is the set of functions computable in deterministic time bounded by some $H^\beta(n)$ for $\beta < \omega^\alpha$. This results notably in $\mathscr{F}_{<3}$ being the set of elementary functions, $\mathscr{F}_{<\omega}$ the set of primitive-recursive functions, and $\mathscr{F}_{<\omega^\omega}$ the set of multiply-recursive functions.

The *fast-growing complexity* classes $(\mathbf{F}_\alpha)_\alpha$ are classes of decision problems that require time or space resources (at such high complexities, the distinction between time and space is indeed moot) bounded by H^{ω^α} [23]:

$$\mathbf{F}_\alpha = \bigcup_{p\in\mathscr{F}_{<\alpha}} \text{DT}\textsc{ime}(H^{\omega^\alpha}(p(n))) \,. \tag{24}$$

For instance, $\mathbf{F}_3$ is the set of decision problems solvable with resources bounded by a tower of exponentials of height elementary in the size of the input, $\mathbf{F}_\omega$ with resources bounded by an Ackermannian applied to some primitive-recursive function on the size of the input, and $\mathbf{F}_{\omega^\omega}$ with resources bounded by an hyper-Ackermannian function applied to a multiply-recursive function on the size of the input.

We can replace H by other functions h in (24), provided that h does not grow too fast. Theorem 4.2 in [23] allows for instance to derive Eq. (9) for multiply-recursive h.

D Normed Well-Quasi-Orders and Complexity

The purpose of this appendix is to provide the details of the proof of Thm. 5.1. The main ingredient to that end is a *length function theorem* for controlled bad sequences over the wqo defined in Lem. B.1; see Cor. D.6. We rely for this on a similar theorem proven by Rosa-Velardo [20, App. A] for so-called *exponential wqos*, which include finite multisets of vectors in $\mathbb{N}^k$ ordered by embedding as a particular case. Our length function theorem requires however a substantial amount of work on top of that of Rosa-Velardo's for two reasons: we work with extended vectors in $\mathbb{N}_\omega^k$ (see Lem. D.1), and use permutation embeddings rather than just plain embeddings (see Lem. B.1).

D.1 Exponential Well-Quasi-Orders

We define formally a *finite multiset* over some quasi-order $(X, \leq_X)$ as a permutation class of partial functions m from $\mathbb{D}$ to X with finite domain $\mathrm{Dom}(m) \subseteq \mathbb{D}$; we shall work directly with canonical representatives m for each permutation class. An *embedding* $m \leq_{\mathbb{M}(X)} m'$ occurs between two finite multisets if there exists an injection π from $\mathrm{Dom}(m)$ to $\mathrm{Dom}(m')$ such that $m(d) \leq_X m'(d)$ for all d in $\mathrm{Dom}(m)$. Thus a finitely supported data vector f can be equated with a finite multiset m of vectors in $\mathbb{Z}_\omega^k \setminus \{\mathbf{0}\}$ with $\mathit{Supp}_{\mathbf{0}}(f) = \mathrm{Dom}(m)$ and $f(d) = m(d)$ for all d in $\mathit{Supp}_{\mathbf{0}}(f)$, and the embedding relations also coincide.

Let us denote by Γ_0 the empty wqo and by Γ_1 the singleton set $\{\bullet\}$ well-quasi-ordered with equality. An *exponential* wqo is one that can be constructed from Γ_0, Γ_1, and $\mathbb{M}(\mathbb{N}^k)$ through Cartesian products and disjoint unions, using respectively the product and sum orderings [20, App. A]. In other words, our wqos of interest are produced by the abstract syntax

$$X ::= \Gamma_0 \mid \Gamma_1 \mid \mathbb{M}(\mathbb{N}^k) \mid X \times X \mid X \sqcup X \qquad \text{(exponential wqos)}$$

where k ranges over $\mathbb{N}$. For instance, any finite set $\Gamma_n \stackrel{\text{def}}{=} \{\bullet_1, \ldots, \bullet_n\}$ ordered by equality is an exponential wqo by building disjoint sums of Γ_1. Observe furthermore that $\mathbb{N}^0$ contains a single element, namely the zero-dimensional vector, and is isomorphic with Γ_1. Hence $\mathbb{N}$ with its natural ordering is isomorphic to $\mathbb{M}(\mathbb{N}^0)$ with the embedding ordering, and is therefore an exponential wqo. This in turn entails that $\mathbb{N}^k$ is an exponential wqo for any k by building Cartesian products.

Normed WQOs. A *normed* quasi-order associates a *norm* function $|.|_X \colon X \to \mathbb{N}$ to a quasi-order $(X, \leq)$, such that $X_{\leq n} \stackrel{\text{def}}{=} \{x \in X \mid |x|_X \leq n\}$ is finite for every n. For exponential wqos, we use

- the zero norm on Γ_1, i.e. $|\bullet|_{\Gamma_1} \stackrel{\text{def}}{=} 0$,
- $|x|_{X \sqcup Y} \stackrel{\text{def}}{=} |x|_X$ if $x \in X$ and $|x|_Y$ otherwise, and thus $|\bullet_j|_{\Gamma_n} = 0$ for all $0 \leq j \leq n$,
- the infinity norm on Cartesian products, i.e. $|(x, y)|_{X \times Y} \stackrel{\text{def}}{=} \max(|x|_X, |y|_Y)$, and
- $|m|_{\mathbb{M}(X)} \stackrel{\text{def}}{=} \max\{|\mathrm{Dom}(m)|, |m(d)|_X \mid d \in \mathrm{Dom}(m)\}$ for finite multisets, which gives rise to the expected norm $|n|_{\mathbb{N}} = n$ on $\mathbb{N} \equiv \mathbb{M}(\mathbb{N}^0)$, $|\boldsymbol{v}|_{\mathbb{N}^k} = \max_{1 \leq i \leq k} \boldsymbol{v}[i] = \|\boldsymbol{v}\|$, and coincides for multisets in $\mathbb{M}(\mathbb{N}^k)$ with the norm on finitely supported data vectors defined in Sec. 5.2.

D.2 Reflections.

A *normed order reflection* between two normed quasi-orders $(Y, \leq_Y, |.|_Y)$ and $(Y, \leq_Y, |.|_Y)$ is a function $r \colon X \to Y$ such that for all x and x' from X,

1. $r(x) \leq_Y r(x')$ implies $x \leq_X x'$, and
2. $|r(x)|_Y \leq |x|_X$.

We write $X \hookrightarrow Y$ if such a normed reflection exists. If $(Y, \leq_Y, |.|_Y)$ is a normed wqo and there is a normed order reflection from $(X, \leq_X, |.|_X)$ to it, then $(X, \leq_X, |.|_X)$ is also a normed wqo. Normed order reflections allow us to reduce questions for various normed wqos to the same questions for exponential wqos. As a first example, although $\mathbb{N}_\omega$ is not an exponential wqo, it reflects into $\mathbb{N} \sqcup \Gamma_1$ when using the norm defined in Sec. 5.2, as can be seen with the reflection that maps ω to $\bullet$ and every n in $\mathbb{N}$ to itself.

Reflecting Data Vectors. Another reflection allows to handle finitely supported data vectors with images in $\mathbb{N}_\omega^k$:

Lemma D.1. *There is a normed reflection from* $\mathbb{M}(\mathbb{N}_\omega^k)$ *into* $\big(\mathbb{M}(\mathbb{N}^k)\big)^{2^k}$.

Proof. Let us start by considering vectors in $\mathbb{N}_\omega^k$. Since $\mathbb{N}_\omega \hookrightarrow \mathbb{N} \sqcup \Gamma_1$ and using the fact that $\hookrightarrow$ is a precongruence for Cartesian products [30, Rem. 2.17], we already know that

$$\mathbb{N}_\omega^k \hookrightarrow (\mathbb{N} \sqcup \Gamma_1)^k \,. \tag{25}$$

We are going to over-approximate this further and show that

$$\mathbb{N}_\omega^k \hookrightarrow \mathbb{N}^k \times \Gamma_{2^k} \,. \tag{26}$$

Indeed, given a vector $\boldsymbol{v}$ in $\mathbb{N}_\omega^k$, we can associate to it a pair $r_1(\boldsymbol{v})$ in $(\mathbb{N} \times \{0, \omega\})^k$ where $\{0, \omega\}$ is ordered by equality, defined for $1 \leq i \leq k$ by

$$r_1(\boldsymbol{v})[i] \stackrel{\text{def}}{=} \begin{cases} (\boldsymbol{v}[i], 0) & \text{if } \boldsymbol{v}[i] < \omega \\ (0, \omega) & \text{otherwise.} \end{cases} \tag{27}$$

This is clearly a normed reflection, and we obtain (26) by observing that $(\mathbb{N} \times \{0, \omega\})^k$ is isomorphic with $\mathbb{N}^k \times \Gamma_{2^k}$.

In order to complete the proof of Lem. D.1, we shall prove more generally that for all normed wqos X,

$$\mathbb{M}(X \times \Gamma_n) \hookrightarrow \big(\mathbb{M}(X)\big)^n \,, \tag{28}$$

which combined with (26) will yield the result. Let us write $\Gamma_n = \{\bullet_1, \ldots, \bullet_n\}$ and let us consider the function r_2 defined for every finite multiset m of pairs in $X \times \Gamma_n$ and for every $1 \leq j \leq n$ as a function $r_2(m)[j]$ with domain $\mathrm{Dom}(r_2(m)[j]) \stackrel{\text{def}}{=} \{d \in \mathrm{Dom}(m) \mid \exists x \in X \,.\, m(d) = (x, \bullet_j)\}$, and mapping any d from this domain to x such that $m(d) = (x, \bullet_j)$.

If now $r_2(m) \leq_{\mathbb{M}(X)^n} r_2(m')$, it is straightforward to assemble the n injections π_j from $\mathrm{Dom}(r_2(m)[j])$ to $\mathrm{Dom}(r_2(m')[j])$ into a suitable single injection π from $\mathrm{Dom}(m)$ to $\mathrm{Dom}(m')$ since the various domains $\mathrm{Dom}(r_2(m)[j])$ are disjoint and $\mathrm{Dom}(m) = \bigcup_{1 \leq j \leq n} \mathrm{Dom}(r_2(m)[j])$, thereby proving $m \leq_{\mathbb{M}(X \times \Gamma_n)} m'$.

Finally, $|m|_{\mathbb{M}(X \times \Gamma_n)} = |r_2(m)|_{\mathbb{M}(X)^n}$ for all m since the elements in Γ_n do not contribute to this norm. $\square$

Reflecting Simple Ideals. In order to obtain a length function theorem on simple ideals ordered by permutation embeddings and normed as defined in Sec. 5.2, we shall exhibit normed order reflections into an exponential wqo.

Remark D.2. Note that permutation embeddings do not give rise to a wqo on the set of *all* $\boldsymbol{I}$-simple ideals.

Let $k \stackrel{\text{def}}{=} 1$ and $\boldsymbol{I} \stackrel{\text{def}}{=} \omega$, and define for every i in $\mathbb{N}$ the simple $\boldsymbol{I}$-ideal M_i with $\boldsymbol{I}$-support of cardinal i that maps every element in $\mathit{Supp}_{\boldsymbol{I}}(M_i)$ to 0. Thus $M_0 = [|\omega]$, $M_1 = [0|\omega]$, $M_2 = [0,0|\omega]$, etc. We claim that the sequence $M_0, M_1, \ldots$ is an infinite bad sequence for permutation embeddings. Indeed, assume for the sake of contradiction that there exist indices $i < j$ and a permutation π of $\mathbb{D}$ such that $M_i \leq M_j\pi$. Then the i elements of $\mathit{Supp}_{\boldsymbol{I}}(M_i)$ must be mapped to i elements of $\mathit{Supp}_{\boldsymbol{I}}(M_j)$. There remains $j - i > 0$ elements d in $\pi^{-1}(\mathit{Supp}_{\boldsymbol{I}}(M_j))$ for which $M_i(d) = \omega$ but $M_j(\pi(d)) = 0$, a contradiction.

However, when we restrict ourselves to specific classes of simple ideals, we can recover well-quasi-orders. We start with sets of $\boldsymbol{I}$-simple ideals with supports of cardinal bounded by some s in $\mathbb{N}$:

$$\mathcal{B}_{s,\boldsymbol{I}} \stackrel{\text{def}}{=} \{M \text{ an } \boldsymbol{I}\text{-simple ideal} \mid |\mathit{Supp}_{\boldsymbol{I}}(M)| \leq s\} . \tag{29}$$

The following lemma shows that such $\boldsymbol{I}$-simple ideals can be treated as vectors in $\mathbb{N}_\omega^{sk}$:

Lemma D.3. *For any s in $\mathbb{N}$ and $\boldsymbol{I}$ in $\{0,\omega\}^k$, $\mathcal{B}_{s,\boldsymbol{I}}$ ordered by permutation embeddings reflects into $Y_{s,k} \stackrel{\text{def}}{=} \mathbb{N}^{ks} \times \Gamma_{2^{ks}}$.*

Proof. By Eq. (26), it suffices to exhibit a reflection into $(\mathbb{N}_\omega^k)^s$ ordered by the product ordering. For each $\boldsymbol{I}$-simple ideal M in $\mathcal{B}_{s,\boldsymbol{I}}$, we fix an arbitrary total ordering of $\mathit{Supp}_{\boldsymbol{I}}(M)$ to obtain a vector in $\mathbb{N}_\omega^{|\mathit{Supp}_{\boldsymbol{I}}(M)|}$ and pad it with $s - |\mathit{Supp}_{\boldsymbol{I}}(M)|$ copies of $\boldsymbol{I}$ to obtain a vector $r(M)$ in $(\mathbb{N}_\omega^k)^s$. It is straightforward to check that r is a normed order reflection into $(\mathbb{N}_\omega^k)^s$; recall that the copies of $\boldsymbol{I}$ do not contribute to the norm of $r(M)$. ☐

Another instance of a set of simple ideals well-quasi-ordered by permutation embeddings is $\mathcal{M}_{S,\boldsymbol{I}}$. We shall actually consider *S-permutation embedddings*, by which we mean those data permutations π that satisfy $\pi(d) = d$ for all $d \in S$, and prove the stronger statement that those embeddings give rise to a wqo. We exhibit a normed order reflection into an exponential wqo; this will simultaneously prove Lem. B.1:

Lemma B.1. *For any finite $S \subseteq \mathbb{D}$ and $\boldsymbol{I} \in \{0,\omega\}^k$, permutation embeddings well-quasi-order the set $\mathcal{M}_{S,\boldsymbol{I}} \stackrel{\text{def}}{=} \{M$ an $\boldsymbol{I}$-simple ideal $\mid M(d) \geq \boldsymbol{I}$ for all $d \notin S\}$.*

Proof. We are going to prove a stronger property, namely that $\mathcal{M}_{S,\boldsymbol{I}}$ ordered by S-permutation embeddings reflects into the exponential wqo

$$X_{S,\boldsymbol{I}} \stackrel{\text{def}}{=} \mathbb{N}^{k|S|} \times \Gamma_{2^{k|S|}} \times \left(\mathbb{M}(\mathbb{N}^k)\right)^{2^k} . \tag{30}$$

For this purpose observe that an S-permutation π can be split into the identity on S and a permutation of $\mathbb{D} \setminus S$. We reflect therefore every $\boldsymbol{I}$-simple ideal M into a pair (M_1, M_2) where

– the first component M_1 is the function from S to $\mathbb{N}_\omega^k$ defined by $M_1(d) \stackrel{\text{def}}{=} M(d)$ for all d in S and ordered pointwise; this can also be seen as a single vector in $\mathbb{N}_\omega^{k|S|}$ along with the product ordering, and by (26) this can be further reflected into $\mathbb{N}^{k|S|} \times \Gamma_{2^{k|S|}}$;
– the second component M_2 is the non-negative data vector defined for all d in S by $M_2(d) \stackrel{\text{def}}{=} \mathbf{0}$, and for all d in $\mathbb{D} \setminus S$ and $1 \le i \le k$ by

$$M_2(d)[i] \stackrel{\text{def}}{=} \begin{cases} 0, & \text{if } \boldsymbol{I}[i] = \omega \\ M(d)[i], & \text{otherwise,} \end{cases} \tag{31}$$

where we employ the usual embedding ordering.
By definition of $\mathcal{M}_{S,\boldsymbol{I}}$, for all d in $\mathbb{D} \setminus S$, $M(d) \ge \boldsymbol{I}$, and the inequality is strict for only finitely many values d by definition of an $\boldsymbol{I}$-simple ideal. Hence M_2 is finitely supported. By Lem. D.1, this can be further reflected into $\big(\mathbb{M}(\mathbb{N}^k)\big)^{2^k}$.

Consider two simple ideals M and N in $\mathcal{M}_{S,\boldsymbol{I}}$ and assume $(M_1, M_2) \le (N_1, N_2)$, i.e. $M_1 \le N_1$ for the pointwise ordering and $M_2 \sqsubseteq N_2$. Since M_2 and N_2 are finitely supported, the latter embedding yields the existence of a permutation π such that $M_2 \le N_2\pi$. Since $M_2(d) = (N_2(\pi(d)) = \mathbf{0}$ for all d in S, we can furthermore assume that π is an S-permutation. Then, for all d in S, $M(d) = N(\pi(d))$ since $M_1 \le N_1$ pointwise on S, and for all d in $\mathbb{D} \setminus S$ and all $1 \le i \le k$, if $\boldsymbol{I}[i] = \omega$ then $M(d)[i] = N(\pi(d))[i] = \omega$ and otherwise $M(d)[i] = M_2(d)[i] = N_2(\pi(d))[i] = N(\pi(d))[i]$. Overall, $M \le N\pi$ as desired.

Finally, the conditions on norms are straightforward to check. □

D.3 Length Function Theorems for Permutation Embeddings

A *length function theorem* is a combinatorial statement that provides upper bounds on the length of so-called *controlled bad sequences* in a normed wqo. In App. D.4 we are going to apply such a theorem, proven by Rosa-Velardo [20], to revisit the proof of termination in Lem. B.2 and extract bounds on the length of branches in our coverability tree construction.

Controlled Bad Sequences. Consider some quasi-order $(X, \le_X)$. Recall that a finite or infinite sequence $x_0, x_1, \ldots, x_i, \ldots$ of elements from X is *good* if there exist two indices $i < j$ such that $x_i \le_X x_j$, and it is otherwise *bad*. Also recall that a wqo can be defined as a quasi-order where all the bad sequences are finite. There is however in general no bound on the length of bad sequences over a wqo; for instance, the sequences $n, n-1, n-2, \ldots, 1, 0$ can be constructed for all n over the wqo $(\mathbb{N}, \le)$.

The idea in order to extract complexity bounds from the use of wqos is to restrict our attention to bad sequences where the elements cannot grow arbitrarily fast. Let g be a monotone function $\mathbb{N} \to \mathbb{N}$ and n be a non-negative integer in $\mathbb{N}$. A finite or infinite sequence $x_0, x_1, \ldots, x_i, \ldots$ of elements from a normed quasi-order $(X, \leq_X, |.|_X)$ is *(g, n)-controlled* if $|x_i|_X \leq g^i(n)$ for all indices i in $\mathbb{N}$. Note that this entails in particular that $|x_0|_X \leq n$ for the initial element in the sequence, while g measures the amortised growth of the elements in the sequence. The point of this miniaturisation is that there is now a *maximal* length for (g, n)-controlled bad sequences over a normed wqo $(X, \leq_X, |.|_X)$ [30, Prop. 2.5], which we denote by $L_{g,X}(n)$.

Observe finally that, if $X \hookrightarrow Y$, then (g, n)-controlled bad sequences in $(X, \leq_X, |.|_X)$ are of at most the same length as in $(Y, \leq_Y, |.|_Y)$ [30, Prop. 2.16]. Thus the proof of Lem. B.1 shows that $L_{g,\mathcal{M}_{S,\boldsymbol{I}}}(n) \leq L_{g,X_{S,\boldsymbol{I}}}(n)$ for all n.

Maximal Order Types. We shall need in the following a way to map exponential wqos to ordinals indexing the subrecursive functions of App. C. As it turns out, a suitable index for exponential wqos $(X, \leq_X)$ is found in their *maximal order type* $o(X)$, which measures the order type of their maximal linearisation. By results of de Jongh and Parikh [28] and Weiermann [31], we have that o is a bijection between exponential wqos and ordinals below ω^{ω^ω}:

$$
\begin{aligned}
o(\varGamma_0) &= 0\,, & o(\varGamma_1) &= 1\,,\\
o(\mathbb{M}(\mathbb{N}^k)) &= \omega^{\omega^k}\,, & &\\
o(X \times Y) &= o(X) \otimes o(Y)\,, & o(X \sqcup Y) &= o(X) \oplus o(Y)\,.
\end{aligned}
$$

Applying these equations, we see e.g. that

$$
\begin{aligned}
o(\varGamma_n) &= n\,,\\
o(\mathbb{N}^k) &= \omega^k\,,\\
o(Y_{s,k}) &= o(\mathbb{N}^{ks} \times \varGamma_{2^{ks}}) = \omega^{ks} \cdot 2^{ks}\,,\\
o(\mathbb{M}(\mathbb{N}^k)^{2^k}) &= \omega^{\omega^k \cdot 2^k}\,,\\
o(X_{S,\boldsymbol{I}}) &= o\left(\mathbb{N}^{k|S|} \times \varGamma_{2^{k|S|}} \times \left(\mathbb{M}(\mathbb{N}^k)\right)^{2^k}\right) = \omega^{\omega^k \cdot k 2^k |S|} \cdot 2^{k|S|}\,.
\end{aligned}
$$

Theorem D.4 (Rosa-Velardo [20], Cor. 1). *Let $(A, \leq)$ be a normed exponential wqo of maximal order type α. If $\alpha < \omega^{\omega^{d+1}}$ and $N\alpha \leq \ell$ for some d and ℓ in $\mathbb{N}$, then $L_{g,A}(n) \leq h_\alpha(\ell d n)$ where $h(x) \stackrel{\text{def}}{=} x \cdot g(x)$.*

Corollary D.5 (Length Function Theorem for $\boldsymbol{\mathcal{B}_{s,I}}$). *For any S in $\mathbb{N}$ and $\boldsymbol{I} \in \{0, \omega\}^k$, $L_{g,\mathcal{B}_{s,\boldsymbol{I}}}(n) \leq h_{\omega^{ks} \cdot 2^{ks}}\big(ks2^{ks}n\big)$ where $h(x) \stackrel{\text{def}}{=} x \cdot g(x)$.*

Proof. We apply Thm. D.4 to $Y_{s,k}$ defined in Lem. D.3: its maximal order type is $o(Y_{s,k}) = \omega^{ks} \cdot 2^{ks} < \omega^{ks+1}$ with norm $No(Y_{s,k}) = 2^{ks}$. Hence, setting $h(x) \stackrel{\text{def}}{=}$

$x \cdot g(x)$, for all n,

$$\begin{aligned} L_{g,\mathcal{B}_{s,\boldsymbol{I}}}(n) &\leq L_{g,Y_{s,k}}(n) && \text{using Lem. D.3 and [30, Prop. 2.16],} \\ &\leq h_{\omega^{ks}\cdot 2^{ks}}(ks2^{ks}n) && \text{by Thm. D.4.} \end{aligned}$$
□

Corollary D.6 (Length Function Theorem for $\mathcal{M}_{S,\boldsymbol{I}}$). *For any finite set $S \subseteq \mathbb{D}$ and $\boldsymbol{I} \in \{0,\omega\}^k$, $L_{g,\mathcal{M}_{S,\boldsymbol{I}}}(n) \leq h_{\omega^{\omega^{k+1}}}\left(k^2 2^{k|S|}n\right)$ where $h(x) \stackrel{def}{=} x \cdot g(x)$.*

Proof. It suffices to apply Thm. D.4 to $X_{S,\boldsymbol{I}}$ defined in (30): its maximal order type is $o(X_{S,\boldsymbol{I}}) = \omega^{\omega^k \cdot k2^{k|S|} \cdot 2^{k|S|}} < \omega^{\omega^{k+1}}$ with norm $No(X_{S,\boldsymbol{I}}) \leq k2^{k|S|}$. Hence, setting $h(x) \stackrel{\text{def}}{=} x \cdot g(x)$, for all n,

$$\begin{aligned} L_{g,\mathcal{M}_{S,\boldsymbol{I}}}(n) &\leq L_{g,X_{S,\boldsymbol{I}}}(n) && \text{using Lem. B.1 and [30, Prop. 2.16],} \\ &\leq h_{\omega^{\omega^k \cdot k2^{k|S|} \cdot 2^{k|S|}}}(k^2 2^{k|S|}n) && \text{by Thm. D.4,} \\ &\leq h_{\omega^{\omega^{k+1}}}(k^2 2^{k|S|}n) && \text{by [30, Claim B.1.1 and Lem. C.9].} \end{aligned}$$
□

D.4 Proof of Thm. 5.1

We have now with corollaries D.5 and D.6 the main ingredients for a proof of Thm. 5.1. We are going to prove the theorem ‘bottom-up’ in a sequence of lemmata, by bounding the norms of simple ideals along branches of a coverability tree, first in acceleration-free sequences of steps with bounded supports (Lem. D.7), then in sequences with bounded supports allowing for depth acceleration but forbidding width accelerations (Lem. D.8), then in sequences allowing for depth acceleration but forbidding width acceleration (Lem. D.9), and finally for arbitrary branches, thereby proving the theorem.

Lemma D.7. *If $M_1, \ldots, M_\ell$ is an acceleration-free segment of a branch of a coverability tree with $|Supp_{\boldsymbol{I}}(M_i)| \leq |Supp_{\boldsymbol{I}}(M_1)|$ for all i, then*

$$\|M_\ell\| \leq b^{\omega^{k\|M_1\|} \cdot 2^{k\|M_1\|}}(k\|M_1\|^2 2^{k\|M_1\|})$$

for $b(x) \stackrel{def}{=} x(x + \|\mathcal{T}\|)$.

Proof. Since no width acceleration occurs in the segment, we have that the ideals M_i share a single vector $\boldsymbol{I}$. Then $M_1, \ldots, M_\ell$ is a sequence of elements in $\mathcal{B}_{|Supp_{\boldsymbol{I}}(M_1)|,\boldsymbol{I}}$. Because no acceleration occurs in the segment, this sequence is a bad sequence for permutation embeddings. Furthermore, the norm of M_{i+1} can only grow by $\|\mathcal{T}\|$ in a single step, meaning that the sequence $M_1, \ldots, M_\ell$ is controlled by $g(x) \stackrel{\text{def}}{=} x + \|\mathcal{T}\|$ and $\|M_1\|$. By Cor. D.5, $\ell \leq b_{\omega^\omega}(k\|M_1\|^2 2^{k\|M_1\|})$ for $b(x) = x(x + \|\mathcal{T}\|)$, yielding the result. □

Lemma D.8. *If $M_1, \ldots, M_\ell$ is a segment of a branch of a coverability tree with $|Supp_{\boldsymbol{I}}(M_i)| \leq |Supp_{\boldsymbol{I}}(M_1)|$ for all i, and no width acceleration occurs, then $\|M_\ell\| \leq B^{\omega^\omega}(\|M_1\|)$ for $B(x)$ an elementary function of x, k, and $\|\mathcal{T}\|$.*

Proof. As in the proof of Lem. D.7, $M_1, \dots, M_\ell$ is a sequence of elements in $\mathcal{B}_{|Supp_{\boldsymbol{I}}(M_1)|,\boldsymbol{I}}$. Following the argument of Lem. B.2, there can only be finitely many depth accelerations between M_1 and M_ℓ: indeed, we can introduce at most $k|Supp_{\boldsymbol{I}}(M_1)|$ ωs in the supports, hence there are at most $k\|M_1\|$ depth accelerations in the segment. Observe that depth accelerations do not increase the norms of the simple ideals involved, thus we can bound $\|M_\ell\|$ using $k\|M_1\|+1$ compositions of the function provided by Lem. D.7:

$$\begin{aligned}\|M_\ell\| &\leq b^{\omega^{k\|M_1\|}\cdot 2^{k\|M_1\|}(k\|M_1\|+1)}\big(k\|M_1\|^2 2^{k\|M_1\|}\big) && \text{byEq. (19)},\\ &\leq b^{\omega^\omega}\big((k\|M_1\|+1)\|M_1\|2^{k\|M_1\|}\big)\;.\end{aligned}$$

We can simplify this expression by setting $B(x) \stackrel{\text{def}}{=} b\big((kx+1)x2^{kx}\big)$; then

$$\|M_\ell\| \leq B^{\omega^\omega}(\|M_1\|)\;. \qquad \square$$

Lemma D.9. *If $M_1, \dots, M_\ell$ is a segment of a branch of a coverability tree where no width acceleration occurs, then $\|M_\ell\| \leq A^{\omega^{\omega^{k+2}}}(\|M_1\|)$ for $A(x)$ an elementary function of x, k, and $\|\mathcal{T}\|$.*

Proof. Because no width acceleration occurs in the segment, all the nodes in the segment are $\boldsymbol{I}$-ideals for a fixed $\boldsymbol{I}$. Moreover, because only transition steps and depth accelerations are allowed, the sequence $M_1, \dots, M_\ell$ is a sequence in $\mathcal{M}_{S,\boldsymbol{I}}$ for $S \stackrel{\text{def}}{=} Supp_{\boldsymbol{I}}(M_1)$. Consider the sizes $s_i \stackrel{\text{def}}{=} |Supp_{\boldsymbol{I}}(M_i)|$ of the supports in the sequence. We split this sequence into successive segments $M_1 = M_{i_0}, \dots, M_{i_1}, \dots, M_{i_2}, \dots, M_{i_n}, \dots, M_\ell$ whenever a larger support size is encountered for the first time: formally, there is a split starting at index i_j if $s_{i_j} > \max_{r<i_j} s_r$.

Each segment $M_{i_j}, \dots, M_{i_{j+1}-1}$ is such that $|Supp_{\boldsymbol{I}}(M_r)| \leq |Supp_{\boldsymbol{I}}(M_{i_j})|$ for all $i_j \leq r < i_{j+1}$, hence $\|M_{i_{j+1}}\| \leq B^{\omega^\omega}(\|M_{i_j}\|)$ according to Lem. D.8, which also yields

$$\|M_\ell\| \leq B^{\omega^\omega}(\|M_{i_n}\|)\;. \tag{32}$$

Since $s_{i_0} < s_{i_1} < \cdots < s_{i_n}$ but no width acceleration occurs in the segment, $M_1, \dots, M_{i_r}$ is therefore a bad sequence in $\mathcal{M}_{S,\boldsymbol{I}}$ controlled by $(B^{\omega^\omega}, \|M_1\|)$. Hence, by Cor. D.6,

$$r \leq c_{\omega^{\omega^{k+1}}}\big(k^2 2^{k\|M_1\|}\|M_1\|\big) \tag{33}$$

for $c(x) \stackrel{\text{def}}{=} x \cdot B^{\omega^\omega}(x)$, and therefore

$$\begin{aligned}\|M_\ell\| &\leq c(\|M_{i_n}\|) && \text{by (32)},\\ &\leq c^{\omega^{\omega^{k+1}}+1}\big(k^2 2^{k\|M_1\|}\|M_1\|\big) && \text{by (33) and (19)}.\end{aligned}$$

Let $a(x) \stackrel{\text{def}}{=} x \cdot B(x)$; then $c(x) \leq a^{\omega^\omega}(x)$ and

$$\begin{aligned}\|M_\ell\| &\leq (a^{\omega^\omega})^{\omega^{\omega^{k+1}}+1}\big(k^2 2^{k\|M_1\|}\|M_1\|\big) && \text{by (22)},\\ &\leq a^{\omega^{\omega^{k+1}+\omega}+\omega^\omega}\big(k^2 2^{k\|M_1\|}\|M_1\|\big)\;.\end{aligned}$$

Setting $A(x) \stackrel{\text{def}}{=} a(k^2 2^{kx} x)$, we obtain

$$\|M_\ell\| \leq A^{\omega^{\omega^{k+1}+\omega}+\omega^\omega}(\|M_1\|) \leq A^{\omega^{\omega^{k+2}}}(\|M_1\|) . \qquad \square$$

Theorem 5.1. *The norms of the simple ideals in a coverability tree rooted in a configuration f_0 for a k-dimensional UDPN $\mathcal{T}$ are bounded by $h^{\omega^{\omega^{k+3}}}(\|f_0\|)$, where $h(x)$ is an elementary function of x, k, and $\|\mathcal{T}\|$.*

Proof. Consider any maximal branch $f_0 = M_0, \ldots, M_\ell$. As explained in the proof of Lem. B.2, there can be at most $W \leq |\{0,\omega\}^k| = 2^k$ width accelerations along this branch, say at nodes $M_{i_1}, \ldots, M_{i_W}$; also define $i_0 \stackrel{\text{def}}{=} 0$ and $i_{W+1} \stackrel{\text{def}}{=} L$. Then Lem. D.9 shows that, if $i_j \leq i < i_{j+1}$ where $0 \leq j \leq W$, then

$$\|M_i\| \leq A^{\omega^{\omega^{k+2}}}(\|M_{i_j}\|) . \tag{34}$$

Note that width acceleration do not increase the norm of simple ideals, hence

$$\begin{aligned}\|M_\ell\| &\leq A^{\omega^{\omega^{k+2}}\cdot(2^k+1)}(\|M_1\|) && \text{by (34) and (19),}\\ &\leq A^{\omega^{\omega^{k+3}}}(\|M_1\| + 2^k + 1) ,\\ &\leq h^{\omega^{\omega^{k+3}}}(\|M_1\|)\end{aligned}$$

when setting $h(x) \stackrel{\text{def}}{=} A(x + 2^k + 1)$. $\square$

E Implementing Hardy Computations

We provide in this section the details for Thm. 5.3:

Theorem 5.3 (Hyper-Ackermannian Coverability Trees). *There exists families of $O(k)$-sized UDPNs $(\mathcal{T}_k)_k$ and $O(k + \log n)$-sized initial configurations $(f_{k,n})_{k,n}$, whose coverability trees are of size at least $H^{\omega^{\omega^k}}(n)$.*

As explained in the main text, we implement for this *Hardy computations* of $H^{\omega^{\omega^k}}(n) = H^{\omega^{\omega^{k-1}\cdot(n+1)}}(n)$ in UDPNs $\mathcal{T}_k$, starting from initial configurations $f_{k,n}$ encoding $(\omega^{\omega^{k-1}\cdot(n+1)}, n)$.

The construction of $\mathcal{T}_k$ can be seen as a simplification over similar constructions by Haddad et al. [11] and Rosa-Velardo [20] for ordered data Petri nets and unordered nets respectively. These constructions on more powerful classes of systems were for *weak Hardy computers* for $H^\alpha(n)$ (and their inverses), which provide more guarantees than what we aim for here: in addition to the existence of a perfect computation ending with $n' = H^\alpha(n)$, a weak Hardy computer also ensures that all the imperfect computations yield some $n'' \leq H^\alpha(n)$.

E.1 Encoding Ordinals as Configurations

In order to implement Hardy computations of $H^{\omega^{\omega^k}}(n)$ according to Eq. (10), we need to encode pairs (α, n) of ordinals α below ω^{ω^k} and natural numbers n into configurations of an UDPN. We use the same encoding of ordinals as the one employed by Rosa-Velardo [20].

Natural Numbers. The natural number n of our pairs (α, n) can simply be represented by a dedicated coordinate that we shall call p_n; we will ensure that a single datum d_n uses this coordinate in all reachable configurations C along the perfect computation. We write $n(C) \stackrel{\text{def}}{=} C(d_n)[p_n]$ for the natural number encoded by a configuration C.

Small Ordinals below ω^k can be written as $\beta = \omega^{k-1} \cdot c_{k-1} + \cdots + \omega^0 \cdot c_0$ where $0 \leq c_{k-1}, \ldots, c_0 < \omega$. Such an ordinal term can be readily seen as a k-dimensional vector of natural numbers $\boldsymbol{v}(\beta)$ indexed by $K \stackrel{\text{def}}{=} \{\omega^0, \ldots, \omega^{k-1}\}$ with $\boldsymbol{v}(\beta)[\omega^i] \stackrel{\text{def}}{=} c_i$ for all $\omega^i \in K$. Conversely, any vector $\boldsymbol{v}$ of natural numbers indexed by K codes an ordinal $\beta(\boldsymbol{v}) \stackrel{\text{def}}{=} \omega^{k-1} \cdot \boldsymbol{v}[\omega^{k-1}] + \cdots + \omega^0 \cdot \boldsymbol{v}[\omega^0]$. These mappings are bijective inverses when projecting vectors onto K. The ordinal ordering then corresponds to the lexicographic ordering over such vectors.

The zero vector $\mathbf{0}$ encodes $0 = \beta(\mathbf{0})$; a successor small ordinal $\beta + 1$ has $\boldsymbol{v}(\beta + 1)[\omega^0] > 0$ and a limit small ordinal $\beta + \omega^{m+1}$ with $m + 1 < k$ has $\boldsymbol{v}(\beta + \omega^{m+1})[\omega^{m+1}] > 0$ and $\boldsymbol{v}(\beta + \omega^{m+1})[\omega^i] = 0$ for all $0 \leq i < m + 1$.

Large Ordinals below ω^{ω^k} can in turn be written in CNF as $\alpha = \omega^{\beta_1} + \cdots + \omega^{\beta_r}$ for some small ordinals $\omega^k > \beta_1 \geq \cdots \geq \beta_r$.

Such an ordinal could be encoded as a finite configuration with each $\boldsymbol{v}(\beta_j)$ being the image of a distinct datum from $\mathbb{D}$. There would however be the issue that $\boldsymbol{v}(0) = \mathbf{0}$ would be lost out of the support. We therefore add an extra dimension o (for 'ordinal') to our vectors, with value 0 or 1 denoting respectively that the vector can be ignored or that it should be taken into account.

Then the encoding of α is the configuration $C(\alpha)$ with support $\mathit{Supp}_{\mathbf{0}}(C(\alpha)) = \{d_1, \ldots, d_r\}$, and for $1 \leq j \leq r$, $C(\alpha)(d_j){\restriction_K} \stackrel{\text{def}}{=} \boldsymbol{v}(\beta_j)$ (and $C(\alpha)(d_j)[o] \stackrel{\text{def}}{=} 1$). Note that this encoding loses the ordering information present in the ordinal α. The converse mapping is $\alpha(C) \stackrel{\text{def}}{=} \bigoplus_{d \in \mathbb{D}, C(d)[o]=1} \omega^{\beta(C(d){\restriction_K})}$ for a configuration C—it recovers the ordering information thanks to the use of natural sums.

For instance, for $k = 2$ and showing only the projections on K of our vectors, 0 is now encoded as the configuration $\left[\ \right]$ with empty support. A successor large ordinal like $\omega^{\omega \cdot 5+3} + \omega^{\omega \cdot 4+6} + 2$ is encoded by $\left[\begin{smallmatrix} 5 & 4 & 0 & 0 \\ 3 & 6 & 0 & 0 \end{smallmatrix}\right]$.

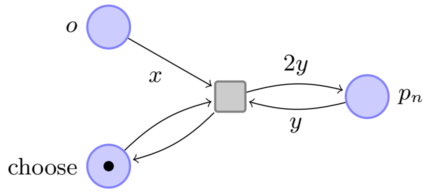


Fig. 3. Implementing (R_{+1}).

E.2 Hardy Steps on Codes

Because we only care about ordinals below ω^{ω^k} for some k, the rules in Eq. (10) can be refined to distinguish between the two different limit cases:

$$(\alpha+1, n) \to (\alpha, n+1)\ , \tag{R_{+1}}$$

$$(\alpha+\omega^{\beta+1}, n) \to (\alpha+\omega^{\beta}\cdot(n+1), n)\ , \tag{R_β}$$

$$(\alpha+\omega^{\beta+\omega^{m+1}}, n) \to (\alpha+\omega^{\beta+\omega^{m}\cdot(n+1)}, n)\ , \tag{R_m}$$

where $\alpha < \omega^{\omega^k}$ is a large ordinal, $\beta < \omega^k$ is a small ordinal, and $m+1 < k$ is finite.

To ease reading, we depict $\mathcal{T}_k$ as a place/transition net in the style of [21, 11]; such a system can be translated into an UDPN defined in VAS-style by increasing the dimension. Its structure revolves around a single $\{0,1\}$-valued coordinate 'choose' where the rule to apply and the data value involved are picked nondeterministically. A configuration C with some datum d such that $C(d)[\text{choose}] = 1$ is called *stable*.

Rule (R_{+1}) can be implemented by guessing that some data from $\mathbb{D}$ is mapped to $C(d)\restriction_K = \mathbf{0}$ in the current stable configuration C. In order to decrement $\alpha(C)$, we should remove d from the support, i.e. decrement $C(d)[o]$; in order to increment n, we should increment $C(d_n)[p_n]$. See Fig. 3 for a depiction.

Rule (R_m) guesses the datum d with minimal $\beta(C(d)\restriction_K)$, checks $C(d)[\omega^{m+1}] > 0$ by decrementing it, and enters a cycles to add n to $C(d)[\omega^m]$ thanks to an auxiliary coordinate p'_n. It finally cycles back p'_n into p_n and returns to the choice of the next rule to apply; see Fig. 4.

Rule (R_β) is implemented by guessing the datum d with minimal associated $\beta(C(d)\restriction_K)$ and checking that $C(d)[\omega^0] > 0$ by decrementing it. It then enters two nested systems: an outer loop through p_n into p'_n also guesses nondeterministically a fresh datum d_j at each go j through the loop and makes it active by incrementing $C(d_j)[o]$ (this datum is selected by z in Fig. 5), before entering the inner system. The latter consists of k successive stages, one for every i in K, and also employs auxiliary coordinates $\omega^{i'}$. Each pass through the inner system attempts to build a copy of $C(d)\restriction_K$ into $C(d_j)\restriction_K$. For each stage i, the

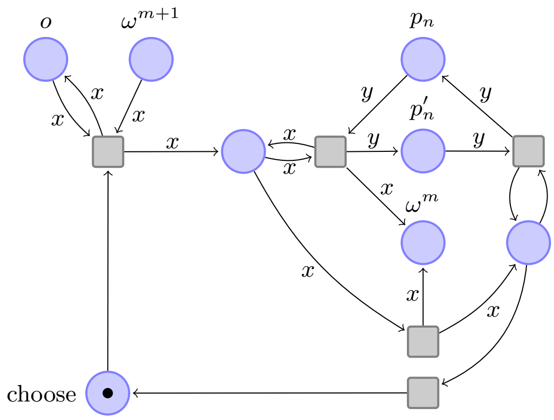


Fig. 4. Implementing (R_m).

inner system first cycles $C(d)[\omega^i]$ into $C(d_j)[\omega^i]$ and $C(d)[\omega^{i'}]$ and then cycles $C(d)[\omega^{i'}]$ back into $C(d)[\omega^i]$. Once the outer cycle is finished, we need to cycle back p'_n into p_n and thereafter return to the choice of the next rule to apply. We leave the details of the implementation of the inner system as an exercise for the reader.

E.3 Perfect and Imperfect Computations

Sources of Imperfection. We face three main of issues when implementing $(R_{+1}$–$R_m)$ on codes in an UDPN:

1. When implementing (R_β) and (R_m), we need to cycle through n. We do this by having an extra coordinate p'_n which we increment while decrementing n, and then cycle in reverse to reestablish the value of n. However, we have no means of ensuring that this cycling and its reverse have been performed in full. A similar issue arises with (R_β), which needs to copy n times the encoding $\boldsymbol{v}(\beta)$, thus to cycle through the components in K and back for each of these n times.
2. Because there is no explicit ordering in our encodings of ordinals, we only nondeterministically guess the lexicographically smallest vector in our configuration, which is the one on which $(R_{+1}$–$R_m)$ should always act. For instance, we cannot prevent applying (R_β) on a configuration that actually encodes a successor ordinal and for which (R_{+1}) should have been applied instead. Conversely, when implementing (R_{+1}), we cannot check that the datum d we picked was indeed such that $C(d)\restriction_K = \mathbf{0}$.
3. When creating new datum in the implementation of (R_β), we might reuse some datum already in use.

Issue (1) would actually be innocuous if we were trying to build a weak Hardy computer. Issue (2) is also innocuous for weak Hardy computers but is a major

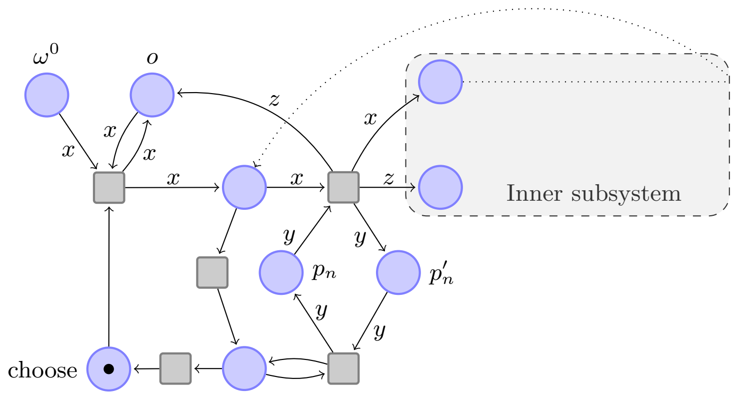


Fig. 5. Implementing the outer system for (R_β).

issue for weak *inverse* Hardy computers; Haddad et al. [11] and Rosa-Velardo [20] employ much more complex constructions, that rely on order or whole-place operations, in order to avoid it. Issue (3) is the main reason why our systems $\mathcal{T}_k$ are *not* weak Hardy computers, but can be tackled with order or whole-place operations [11, 20].

Perfect Computation. The important point is that the implementation allows a *perfect* computation $C_0, C_1, \ldots, C_L$ starting in $C_0 = f_{k,n}$ where none of the nondeterministic errors pointed above is made by $\mathcal{T}_k$. This computation implements the completed Hardy computation $(\alpha_0, n_0) \to \cdots \to (\alpha_\ell, n_\ell)$ with $\alpha_0 = \omega^{\omega^{k-1}\cdot(n+1)}$, $n_0 = n$, $\alpha_\ell = 0$, and therefore $n_\ell = H^{\omega^{\omega^{k-1}\cdot(n+1)}}(n)$.

Let us extract from $C_0, C_1, \ldots, C_L$ the indices of stable configurations $i_0 = 0 < i_1 < \cdots < i_\ell \leq L$ in the perfect computation. Formally, we ask from the perfect computation that each pair (α_j, n_j) of the completed Hardy computation is faithfully encoded into C_{i_j}, i.e. $\alpha(C_{i_j}) = \alpha_j$ and $n(C_{i_j}) = n_j$, that no two indices $i_j \leq r < p < i_{j+1}$ with $C_r \sqsubseteq C_p$ exist, and that if $C_r \sqsubseteq C_p$ for some $i_j \leq r < i_{j+1} \leq i_{j'} \leq p < i_{j'+1}$, then $C_{i_j} \sqsubseteq C_{i_{j'}}$; these conditions can be checked on the implementations of each rule individually.

In such a perfect computation, no acceleration is possible. Indeed, the ordinal encoding is *robust*, meaning here that $C_i \sqsubseteq C_j$ for $i < j$ implies $\alpha(C_i) \leq \alpha(C_j)$ [20, Prop. 1]. Thus, assuming for the sake of contradiction that $C_r \sqsubseteq C_p$ for $r < p$ in the perfect computation, necessarily r and p must belong to two different segments, say $j < j'$ with $i_j \leq r < i_{j+1}$ and $i_{j'} \leq p < i_{j'+1}$. But by robustness this would imply $\alpha_j \leq \alpha_{j'}$, contradicting the invariant of Hardy computations that $\alpha_j > \alpha_{j'}$ for all $j < j'$. Therefore a coverability tree for $\mathcal{T}_k$ started in $f_{k,n}$ has a branch, corresponding to the perfect computation, of length at least $H^{\omega^{\omega^k}}(n)$.

Additional References

27. Cichoń, E.A., Tahhan Bittar, E.: Ordinal recursive bounds for Higman's Theorem. Theor. Comput. Sci. 201(1–2), 63–84 (1998)
28. de Jongh, D.H.J., Parikh, R.: Well-partial orderings and hierarchies. Indag. Math. 39(3), 195–207 (1977)
29. Löb, M.H., Wainer, S.S.: Hierarchies of number theoretic functions, I. Arch. Math. Logic 13, 39–51 (1970)
30. Schmitz, S., Schnoebelen, Ph.: Multiply-recursive upper bounds with Higman's Lemma. ICALP 2011. LNCS, vol. 6756, pp. 441–452. Springer (2011), http://arxiv.org/abs/1103.4399
31. Weiermann, A.: A computation of the maximal order type of the term ordering on finite multisets. CiE 2009. LNCS, vol. 5635, pp. 488–498. Springer (2009)