The Parametric Ordinal-Recursive Complexity of Post Embedding Problems

Prateek Karandikar\textsuperscript{1,2} Sylvain Schmitz\textsuperscript{2}

\textsuperscript{1}CMI, Chennai, India
\textsuperscript{2}LSV, ENS Cachan & CNRS, France

Abstract

Post Embedding Problems are a family of decision problems based on the interaction of a rational relation with the subword embedding ordering, and are used in the literature to prove non multiply-recursive complexity lower bounds. We refine the construction of Chambart and Schnoebelen (LICS 2008) and prove parametric lower bounds depending on the size of the alphabet.

1 Introduction

Ordinal Recursive functions and subrecursive hierarchies (Rose, 1984; Fairtlough and Wainer, 1998) are employed in computability theory, proof theory, Ramsey theory, rewriting theory, etc. as tools for bounding derivation sizes and other objects of very high combinatorial complexity. A standard example is the ordinal-indexed extended Grzegorczyk hierarchy \( F_\alpha \) (Löb and Wainer, 1970), which characterizes classical classes of functions: for instance, \( F_2 \) is the class of elementary functions, \( \bigcup_{k<\omega} F_k \) of primitive-recursive ones, and \( \bigcup_{k<\omega} F_k \) of multiply-recursive ones. Similar tools are required for the classification of decision problems arising with verification algorithms and logics, prompting the still young investigation of fast-growing complexity classes \( F_\alpha \) and their associated complete problems (Friedman, 1999; Schmitz and Schnoebelen, 2012).

Post Embedding Problems (PEPs) have been introduced by Chambart and Schnoebelen (2007) as a tool to prove the decidability of safety and termination problems in unreliable channel systems. The most classical instance of a PEP is called “regular” by Chambart and Schnoebelen (2007), but we will follow Barceló et al., (2012) and rather call it rational in this paper:

\textbf{Rational Embedding Problem} (EP[Rat])

\textbf{input} A rational relation \( R \) in \( \Sigma^* \times \Sigma^* \).

\textbf{question} Is the relation \( R \cap \subseteq \) empty?

\*Research partially funded by the ANR ReacHard project (ANR 11 BS02 001 01). The first author is partially funded by the Tata Consultancy Service. Part of this research was conducted while the second author was visiting the Department of Computer Science at Oxford University thanks to a grant from the ESF Games for Design and Verification activity.
Here, the $\sqsubseteq$ relation denotes the subword embedding ordering, which relates two words $w$ and $w'$ if $w = c_1 \cdots c_n$ and $w' = w_0c_1w_1 \cdots w_nc_nw_{n+1}$ for some symbols $c_i$ in $\Sigma$ and words $w_i$ in $\Sigma^*$; in other words, $w$ can be obtained from $w'$ by “losing” some symbol occurrences (maybe none).

Although PEPs appear naturally in relation with channel systems (Chambart and Schnoebelen, 2007, 2008a; Jančar et al., 2012) and queries on graph databases (Barceló et al., 2012), their main interest lies in their use in lower bound proofs for other, sometimes seemingly distantly related problems (Ouaknine and Worrell, 2007; Lasota and Walukiewicz, 2008; Atig et al., 2010): in spite of their simple formulation, they are known to be of non multiply-recursive complexity in general. In fact, this motivation has been present from their inception in (Chambart and Schnoebelen, 2007): find a “master” decision problem complete for $F_{\omega^\omega}$, the class of hyper-Ackermannian problems, solvable with non multiply-recursive complexity, but no less—much like SAT is often taken as the canonical $\text{NPTime}$-complete problem, or the Post Correspondence Problem for $\Sigma^0_1$. This has also prompted a wealth of research into variants and related questions (Chambart and Schnoebelen, 2008b, 2010a,b; Barceló et al., 2012; Karandikar and Schnoebelen, 2012).

In this paper, we revisit and simplify the original proof of Chambart and Schnoebelen (2008c) that established the hardness of PEPs, and prove tight parameterized lower bounds when the size of the alphabet $\Sigma$ is fixed. More precisely, we show that the $(k + 2)$-rational embedding problem, i.e. the restriction of EP[Rat] to alphabets $\Sigma$ of size at most $k + 2$, is hard for $F_{\omega^k}$ the class of $k$-Ackermannian problems if $k \geq 2$. As the problem can be shown to be in $F_{\omega^{k+1}}$ (Schmitz and Schnoebelen, 2011; Karandikar and Schnoebelen, 2012), we argue this to be a rather tight bound. The hyper-Ackermannian lower bound of $F_{\omega^\omega}$ first proven by Chambart and Schnoebelen then arises when $|\Sigma|$ is not fixed but depends on the instance.

**Overview.** Technically, our results rely on an implementation of the computations for the Hardy functions $H_{\omega^k}$ and their inverses by successive applications of a relation with a fixed bounded length discrepancy. The main difficulty here is that this implementation should be robust for the symbol losses associated with the embedding relation. It requires in particular a robust encoding of ordinals below $\omega^\omega$ as sequences over an alphabet of $k + 2$ symbols, for which we adapt the constructions of (Chambart and Schnoebelen, 2008c; Haddad et al., 2012; see Section 3).

This allows us to show in Section 4 that the following problem is $F_{\omega^k}$-hard when $k \geq 2$, $|\Sigma| = k + 2$, and $R$ has a bounded length discrepancy of 1:

**Lossy Rewriting** (LR[Rat])

Figure 1: Relationships between PEPs and similar decision problems.
input A rational relation \( R \) in \( \Sigma^* \times \Sigma^* \) and two words \( w \) and \( w' \) in \( \Sigma^* \).

question Does \((w, w')\) belong to the reflexive transitive closure \( R^\sqcup \)?

Here \( R^\sqcup \) denotes the “lossy version” of the relation \( R \), defined formally as the composition \( \sqcup \circ R \circ \sqcup \). We denote the restricted problem when the rational relation \( R \) has a bounded length discrepancy of 1 by \( LR[1-bld] \).

We then show in Section 5 that \( LR[1-bld] \) can easily be reduced to \( EP[Rat] \) and other (parameterized) embedding problems—including \( EP[Sync] \), a restriction of \( EP[Rat] \) introduced by Barceló et al. (2012) where the relation \( R \) is synchronous (aka regular), and which required a complex lower bound proof.

Figure 1 summarizes the lower bounds presented in this paper. In a sense, \( LR \) is our own champion for the title of “master” problem for \( F_{\omega \omega} \). Besides its rather simple statement, note that the related question of whether \((w, w')\) belongs to \( R^\sqcup \) is undecidable by an easy reduction from the acceptance problem for Turing machines.

Let us now turn to the necessary formal background on PEPs in Section 2. Due to space constraints, some proof details will be found in the appendices.

2 Post Embedding Problems

Rational Relations (Elgot and Mezei, 1965) play an important role in the following, as they provide a notion of finitely presentable relations over strings more powerful than string rewrite systems, and come with a large body of theory and results (see e.g. Sakarovitch, 2009, Chap. IV). Let us quickly skim over the notations and definitions that will be needed in this paper.

We assume the reader to be familiar with the basic characterizations of rational relations \( R \) between two finite alphabets \( \Sigma \) and \( \Delta \) by

closure of the finite relations in \( \Sigma^* \times \Delta^* \) under union, concatenation, and Kleene star

finite transductions defined by normalized transducers \( T = \langle Q, \Sigma, \Delta, \delta, I, F \rangle \)

where \( Q \) is a finite set of states, \( \delta \subseteq Q \times ((\Sigma \times \{\varepsilon\}) \cup (\{\varepsilon\} \times \Delta)) \times Q \) (\( \varepsilon \) denoting the empty word of length \( |\varepsilon| = 0 \)), initial set of states \( I \subseteq Q \), and final set of states \( F \subseteq Q \),

decomposition into a regular language \( L \) over some finite alphabet \( \Gamma \) and two morphisms \( u: \Gamma^* \to \Sigma^* \) and \( v: \Gamma^* \to \Delta^* \) s.t. \( R = u^{-1} \circ \text{Id}_L \circ v \), where \( \text{Id}_L \) is the identity function over the restricted domain \( L \).

This last characterization is known as Nivat’s Theorem, and shows that \( EP[Rat] \) can be stated alternatively as taking as input a rational language \( L \) in \( \Gamma^* \) and two morphisms \( u \) and \( v \) from \( \Gamma^* \) to \( \Sigma^* \) and asking whether there exists some word \( x \) in \( L \) s.t. \( u(x) \sqsubseteq v(x) \) (Chambart and Schnoebelen, 2007). This justifies the name of “Post Embedding Problem”, as the related Post Correspondence Problem asks instead given \( u \) and \( v \) whether there exists \( x \) in \( \Gamma^+ \) s.t. \( u(x) = v(x) \).

---

1We use different symbols “\( \ast \)” and “\( \oplus \)” for Kleene star and Kleene plus, i.e. iteration of concatenation “\( \cdot \)” on the one hand, and “\( \circ \ast \)” and “\( \circ \oplus \)” for reflexive transitive closure and transitive closure, i.e. iteration of composition “\( \circ \)” on the other hand. Rational relations and length-preserving relations are closed under Kleene star, but none of the classes of relations we consider is closed under reflexive transitive closure.
Synchronous Relations are a restricted class of rational relations that display closure under intersection and complement, in addition to e.g. the closure under composition and inverse that all rational relations enjoy. A rational relation has \textit{\textbf{b}-bounded length discrepancy} if the absolute value of \(|u| - |v|\) is at most \(b\) for all \((u, v)\) in \(R\), and has \textit{\textbf{bounded length discrepancy}} (bld) if there exists such a finite \(b\). In particular, it is \textit{\textbf{length-preserving}} if \(|u| = |v|\), i.e. if it has bld 0. A synchronous relation is a finite union of relations of form \(\{(u, vw) \mid (u, v) \in R \land w \in L\}\) and \(\{(uw, v) \mid (u, v) \in R \land w \in L\}\) where \(R\) ranges over length-preserving rational relations and \(L\) over regular languages.

In terms of classes of relations in \(\Sigma^* \times \Delta^*\), we have the strict inclusions

\[
\text{lp} = 0\text{-bld} \subsetneq \ldots \subsetneq b\text{-bld} \subsetneq \ldots \subsetneq (b+1)\text{-bld} \subsetneq \text{bld} \subsetneq \text{Sync} \subsetneq \text{Rat} .
\]

Post Embedding Problems, as we have seen in the introduction, are concerned with the interplay of a rational relation \(R\) in \(\Sigma^* \times \Sigma^*\) with the subword embedding ordering \(\subsetneq\). The latter is a particular case of a (deterministic) rational relation that is not synchronous. Both EP[Rat] and LR[Rat] are particular instances of more general, undecidable problems: the emptiness of intersection of two rational relations for EP[Rat], and the word problem in the reflexive transitive closure of a rational relation for LR[Rat]. We can add another natural problem to the set of PEPs:

Lossy Termination (LT[Rat])

\textbf{input} A rational relation \(R\) over \(\Sigma\) and a word \(w\) in \(\Sigma^*\).

\textbf{question} Does \(R \sqcap \subsetneq \) terminate from \(w\), i.e. is every sequence \(w = w_0 R \sqcap w_1 R \sqcap \ldots R \sqcap w_i \sqcap \ldots\) with \(w_0, w_1, \ldots, w_i, \ldots\) in \(\Sigma^*\) finite?

Again, this is a variant of the termination problem, which is in general undecidable when the relation is not lossy.

Restrictions. We parameterize PEPs with the subclass of rational relations under consideration for \(R\) and the cardinal of the alphabet \(\Sigma\); for instance, \((k+2)\text{-EP[Sync]}\) is the variant of EP[Rat] where the relation is synchronous and \(|\Sigma| = k + 2\). We are interested in this paper in providing \(\mathbf{F}_\omega\) lower bounds with the smallest possible class of relations and smallest possible alphabet size, but we should also mention that some (rather strong) restrictions become tractable:

- Barceló et al. (2012) show that EP[Rec]—where a \textit{recognizable relation} is a finite union of products \(L \times L'\) where \(L \) and \(L'\) range over regular languages—is in NLOGSPACE, because the intersection \(R \cap \subseteq\) is rational, and can effectively be constructed and tested for emptiness on the fly.

- Chambart and Schnoebelen (2007) show that EP[2Morph]—where a \textit{2-morphic relation} (Latteux and Leguy, 1983) is the composition \(R = (u^{-1}v) \setminus \{(\varepsilon, \varepsilon)\}\) of two morphisms \(u\) and \(v\) from \(\Gamma^*\) to \(\Sigma^*\)—is in LOGSPACE, because it reduces to checking whether there exists \(a\) in \(\Gamma\) s.t. \(u(a) \subseteq v(a)\),

- the case EP[Rewr] of \textit{rewrite relations} is similarly in LOGSPACE: a rewrite relation \(R\) is defined from a \textit{semi-Thue system}, i.e. a finite set \(\Upsilon\) of rules \((u, v)\) in \(\Sigma^* \times \Sigma^*\), as \(\rightarrow_{\Upsilon} = \{(ww'u, wvw') \mid w, w' \in \Sigma^* \land (u, v) \in \Upsilon\}\), and EP[Rewr] reduces to checking whether \(u \subseteq v\) for some rule \((u, v)\) of \(\Upsilon\),

\[1\]
• the unary alphabet case of 1-EP[Rat] is in NLOGSPACE: this can be seen using Parikh images and Presburger arithmetic; see App. A for details

**Proposition 1.** The problem 1-EP[Rat] is in NLOGSPACE.

## 3 Hardy Computations

We use the **Hardy hierarchy** as our main subrecursive hierarchy (Löb and Wainer 1970, Rose 1984, Faitlough and Wainer 1998). Although we will only use the lower levels of this hierarchy, its general definition is worth knowing, as it is archetypal of ordinal-indexed subrecursive hierarchies; see (Schmitz and Schwoebelen 2012) for a self-contained presentation.

### 3.1 The Hardy Hierarchy

**Ordinal Terms.** Let \( \varepsilon_0 \) be the smallest solution of the equation \( \omega^x = x \). It is well-known that any ordinal \( \alpha < \varepsilon_0 \) can be written uniquely in Cantor Normal Form (CNF) as a sum

\[
\alpha = \omega^{\beta_1} + \cdots + \omega^{\beta_n}
\]

where \( \beta_1 \leq \cdots \leq \beta_n \leq \alpha \) and each \( \beta_i \) is itself in CNF. This ordinal \( \alpha \) is 0 if \( n = 0 \) in (2), a successor ordinal if \( \beta_n = 0 \), and a limit ordinal otherwise. Subrecursive hierarchies are defined through assignments of fundamental sequences \( (\lambda_n)_{n<\omega} \) for limit ordinals \( \lambda < \varepsilon_0 \), satisfying \( \lambda_n < \lambda \) for all \( n \) and \( \lambda = \sup_n \lambda_n \). A standard assignment is defined by:

\[
(\gamma + \omega^{\alpha+1})_n \overset{\text{def}}{=} \gamma + \omega^\alpha \cdot n,
\]

\[
(\gamma + \omega^\lambda)_n \overset{\text{def}}{=} \gamma + \omega^\lambda n,
\]

thus verifying \( \omega^\alpha n = n \). Let \( \Omega \overset{\text{def}}{=} \omega^{\omega^{\omega}} \); this yields for instance \( \Omega_k = \omega^{\omega^k} \) and, if \( k > 0 \), \( (\Omega_k)_n = \omega^{\omega^{k-1}n} \).

**Hardy Hierarchy.** The **Hardy hierarchy** \( (H^\alpha)_{\alpha < \varepsilon_0} \) is an ordinal-indexed hierarchy of functions \( H^\alpha : \mathbb{N} \to \mathbb{N} \) defined by

\[
H^0(n) \overset{\text{def}}{=} n, \quad H^{\alpha+1}(n) \overset{\text{def}}{=} H^\alpha(n+1), \quad H^\lambda(n) \overset{\text{def}}{=} H^{\lambda n}(n).
\]

Observe that \( H^1 \) is simply the successor function, and more generally \( H^\alpha \) is the \( \alpha \)th iterate of the successor function, using diagonalisation to treat limit ordinals. A related hierarchy is the fast growing hierarchy \( (F_\alpha)_{\alpha < \varepsilon_0} \), which can be defined by \( F_\alpha \overset{\text{def}}{=} H^{\omega^\alpha} \), resulting in \( F_0(n) = H^1(n) = n+1, F_1(n) = H^\omega(n) = \omega^n(n) = 2n, F_2(n) = H^{\omega^2}(n) = 2^n n \) being exponential, \( F_3 = H^{\omega^3} \) being non-elementary, \( F_\omega = H^{\omega^\omega} = H^{\Omega_1} \) being an Ackermannian function, \( F_{\omega^k} = H^{\Omega_k} \) a \( k \)-Ackermannian function, and \( F_{\omega^\omega} = H^{\Omega} \) an hyper-Ackermannian function.

**Fast-Growing Complexity Classes.** Our intention is to establish the “\( F_{\omega^k} \)-hardness” of Post embedding problems. In order to make this statement more precise, we define the class \( F_{\omega^k} \) of \( k \)-Ackermannian problems as a specific instance of the fast-growing complexity classes defined for \( \alpha \geq 3 \) by

\[
F_\alpha \overset{\text{def}}{=} \bigcup_{p \in U_{\beta < \alpha}} \text{DTIME}(F_\alpha(p(n))) , \quad \mathcal{F}_\alpha = \bigcup_{c < \omega} \text{FDTIME}(F_\alpha^c(n)) ,
\]
where $\mathcal{F}_\alpha$ defined above is the $\alpha$th level of the extended Grzegorczyk hierarchy (Löb and Wainer 1970) when $\alpha \geq 2$. The classes $\mathcal{F}_\alpha$ are naturally equipped with $\bigcup_{\beta < \alpha} \mathcal{F}_\beta$ as class of reductions. For instance, because $\bigcup_{\beta < \omega} \mathcal{F}_\omega$ is exactly the set of multiply-recursive functions, $\mathcal{F}_\omega$ captures the intuitive notion of hyper-Ackermannian problems closed under multiply-recursive reductions.

Hardy Computations. The fast-growing and Hardy hierarchies have been used in several publications to establish Ackermannian and higher complexity bounds (Chambart and Schnoebelen 2008c; Schmitz and Schnoebelen 2011; Haddad et al. 2012; Schmitz and Schnoebelen 2012). The principle in their use for lower bounds is to view (4), read left-to-right, as a rewrite system over $\mathbb{N}$, × $\mathbb{N}$, and later implement it in the targeted formalism. Formally, a (forward) Hardy computation is a sequence

$$a_0, n_0 \rightarrow a_1, n_1 \rightarrow a_2, n_2 \rightarrow \cdots \rightarrow a_\ell, n_\ell$$

of evaluation steps implementing the equations in (4) seen as left-to-right rewrite rules over Hardy configurations $a, n$. It guarantees $a_0 > a_1 > a_2 > \cdots$ and keeps $H^a(n)$ invariant. We say it is complete when $a_\ell = 0$ and then $n_\ell = H^a(n_0)$ (we also consider incomplete computations). A backward Hardy computation is obtained by using (4) as right-to-left rules. For instance,

$$\omega^k, n \rightarrow \omega^{k-1}, n \rightarrow \omega^{k-1}(n-1)+\omega^{k-2}, n, n$$

constitute the first three steps of the forward Hardy computation starting from $\Omega_k, n$ if $k > 1$ and $n > 0$.

Termination of Hardy Computations. Because $a_0 > a_1 > \cdots > a_\ell$ in a forward Hardy computation like (6), it necessarily terminates. For inverse computations, this is less immediate, and we introduce for this a norm $\|a\|$ of an ordinal $a$ in $e_0$ as its count of “$\omega$” symbols when written as an ordinal term: formally, $\|e_0\| = n$ if $k > 1$ and $n > 0$.

Observe that $\|a \cdot m\| = \|a\| \cdot m$. We can check by transfinite induction on $a > 0$ that, for any limit ordinal $\lambda$, $\|\lambda_n\| > \|\lambda\|$ whenever $n > 1$. Indeed, if $a = \beta + 1$, then $\|\lambda_n\| = \|\gamma + \omega^\alpha \cdot n\| = \|\gamma\| + (1 + \|\beta\|)n > \|\gamma\| + 2 + \|\beta\| = \|\lambda\|$, and in the limit case, $\|\lambda_n\| = \|\alpha\| + 1 + \|\alpha_n\| > \|\gamma\| + 1 + \|\alpha\| = \|\lambda\|$ by ind. hyp. Therefore, if $n$ is larger than 1 in a configuration $a, n$ of an inverse Hardy computation following (4) from right to left, either we apply the successor rule and reach $a + 1, n - 1$ with a decreased $n$, or we apply the limit rule and reach $a', n$ s.t. $a = a'_n$ with a decreased $\|a\|$: in a backward Hardy computation, the pair $(n, \|a\|)$ decreases for the lexicographic ordering over $\mathbb{N}^2$. As this is a well-founded ordering, we see that backward computations terminate if $n$ remains larger than 1—which is a reasonable hypothesis for the following.

Note that, at such high complexities, the usual distinctions between deterministic vs. nondeterministic, or time-bounded vs. space-bounded computations become irrelevant. In particular, $\mathcal{F}_2$ is the set of elementary functions, and $\mathcal{F}_3$ the class of problems with a tower of exponentials of height bounded by some elementary function of the input as an upper bound.
3.2 Encoding Hardy Configurations

Our purpose is now to encode Hardy computations as relations over $\Sigma^*$. This entails in particular (1) encoding configurations $\alpha, \beta$ in $\Omega_k \times \mathbb{N}$ of a Hardy computation as finite sequences using cumulative ordinal descriptions or "codes", which we do in this subsection, and (2) later in Section 3.3 designing a 1-blp relation that implements Hardy computation steps over codes. A constraint on codes is that they should be robust against losses, i.e. if $\pi(x)$ and $\pi(x')$ are the ordinals associated to the codes $x$ and $x'$, then $H^{\pi(x)}(n) \leq H^{\pi(x')}(n)$—pending some hygienic conditions on $x$ and $x'$, see Lem. 2.

**Finite Ordinals** below $k$ can be represented as single symbols $a_0, \ldots, a_{k-1}$ of an alphabet $\Sigma_k$ along with a bijection

$$\varphi(a_i) \overset{\text{def}}{=} i.$$  \hspace{1cm} (9)

**Small Ordinals** below $\omega^k$ are then easily encoded as finite words over $\Sigma_k$: given a word $w = b_1 \cdots b_n$ over $\Sigma_k$, we define its associated ordinal in $\omega^k$ as

$$\beta(w) \overset{\text{def}}{=} \omega^{\varphi(b_1)} + \cdots + \omega^{\varphi(b_n)}.$$  \hspace{1cm} (10)

Note that $\beta$ is surjective but not injective: for instance, $\beta(a_0a_1) = \beta(a_1) = \omega$. By restricting ourselves to pure words over $\Sigma_k$, i.e. words satisfying $\varphi(b_j) \geq \varphi(b_{j+1})$ for all $1 \leq j < n$, we obtain a bijection between $\omega^k$ and $p(\Sigma^*_{k})$ the set of pure finite words in $\Sigma^*_{k}$, because then (10) is the CNF of $\beta(w)$.

**Large Ordinals** below $\Omega_k$ are denoted by codes ([Chambart and Schnoebelen, 2008c][Haddad et al., 2012]), which are #-separated words over the extended alphabet $\Sigma_{k,#}$, where each $w_i$ is a word over $\Sigma_k$. Its associated ordinal $\pi(x)$ in $\Omega_k$ is then defined as

$$\pi(x) \overset{\text{def}}{=} \beta(w_1w_2 \cdots w_{p-1}w_p\#w_{p+1}w_{p+2} \cdots w_m),$$  \hspace{1cm} (11)

or inductively by

$$\pi(w) \overset{\text{def}}{=} 0, \quad \pi(w\#x) \overset{\text{def}}{=} \omega^{\beta(w)} \cdot \pi(x) + \omega^{\beta(w)}$$  \hspace{1cm} (12)

for $w$ a word in $\Sigma^*_{k}$ and $x$ a code. For instance, $\pi(a_0a_0\#) = \omega^{a_{0}+1} = \pi(a_0a_0a_0\# \#a_1)$, or, closer to our concerns, the initial ordinal in our computations is $\pi(a^m_{k-1} \#) = (\Omega_k)_\alpha$ when $k > 0$.

Observe that $\pi$ is surjective, but not injective. We can mend this by defining a pure code $x = w_1\# \cdots \# w_{p-1}\# w_{p+1}$ as one where $w_{p+1} = \varepsilon$ and every word $w_i$ for $1 \leq i \leq p$ is pure—note that it does not force the concatenation of two successive words $w_iw_{i+1}$ of $x$ to be pure. This is intended, as this is the very mechanism that allows $\pi$ to be a bijection between $\Omega_k$ and $p(\Sigma^*_{k,#})$ (see App. [B.1]).

**Lemma 1.** The function $\pi$ is a bijection from $p(\Sigma^*_{k,#})$ to $\Omega_k$.

We also define $p(x)$ to be the unique pure code $x'$ verifying $\pi(x) = \pi(x')$; then $p(x) \supset x$, and $x \supset x'$ implies $p(x) \supset p(x')$. 

7
Hardy Configurations. \( \alpha, n \) are finally encoded as sequences \( c = \pi^{-1}(\alpha) \vdash \#^n \) using a separator “\(|\)”, i.e., as sequences in the language \( \text{Confs} \defeq p(\Sigma^*_k \#) \cdot \{ | \} \cdot \{ \# \}^* \). This is a regular language over \( \Sigma_k \# \uplus \{ | \} \), but the most important fact about this encoding is that it is robust against symbol losses as far as the corresponding computed Hardy values are concerned. Robustness is a critical part of hardness proofs based on Hardy functions. The main difficulty rises from the fact that the Hardy functions are not monotone in their ordinal parameter: for instance, \( H_{\omega^2}(n) = H_\alpha(n) = 2n \) is less than \( H_{\omega^{\omega^2}+1}(n) = 2n + 1 \). Code robustness is addressed in (Chambart and Schnoebelen, 2008c, Prop. 4.3), and in (Haddad et al., 2012, Prop. 16) for a more complex encoding of ordinals below \( \omega^{\omega^{\omega^2}} \) as vector sequences. Robustness is the limiting factor that prevents us from reducing languages in \( F_\alpha \) for \( \alpha > \Omega \) into PEPs.

Lemma 2 (Robustness). Let \( c = x : \#^n \) and \( c' = x' : \#^{n'} \) be two sequences in \( \text{Confs} \). If \( c \sqsubseteq c' \), then \( H_{\pi(x)}(n) \leq H_{\pi(x')}(n') \).

3.3 Encoding Hardy Computations

It remains to present a 1-bld relation that implements Hardy computations over Hardy configurations encoded as sequences in \( \text{Confs} \). We translate the equations from (4) into a relation \( R_H = (R_0 \cup R_1 \cup R_2) \cap (\text{Confs} \times \text{Confs}) \), which can be reversed for backward computations:

\[
R_0 \defeq \{(\#x : \#^n, x : \#^{n+1}) \mid n \geq 0, x \in \Sigma^*_k \# \} \tag{13}
\]

\[
R_1 \defeq \{(wa_0\#x : \#^n, w\#^xp(a_0x) : \#^n) \mid n > 1, w \in \Sigma^*_k, x \in \Sigma^*_k \# \} \tag{14}
\]

\[
R_2 \defeq \{(wa_i\#x : \#^n, wa_{i-1}\#p(a_ix) : \#^n) \mid n > 1, i > 0, w \in \Sigma^*_k, x \in \Sigma^*_k \# \} \tag{15}
\]

The relation \( R_0 \) implements the successor case, while \( R_1 \) and \( R_2 \) implement the limit case of (3) for ordinals of the form \( \gamma + \omega^\alpha + 1 \) and \( \gamma + \omega^\lambda \). The restriction to \( n > 1 \) in \( R_1 \) and \( R_2 \) enforces termination for backward computations; it is not required for correctness. Because \( R_H \) is a direct translation of (4) over \( \text{Confs} \):

Lemma 3 (Correctness). For all \( \alpha, \alpha' \) in \( \Omega_k \) and \( n, n' > 1 \), \( (\pi^{-1}(\alpha) : \#^n)(R_H \cup R_H^{-1})^{(\pi^{-1}(\alpha') : \#^{n'})} \) iff \( H^\alpha(n) = H^{\alpha'}(n') \).

Unfortunately, although \( R_0 \) is a length-preserving rational relation, \( R_1 \) and \( R_2 \) are not 1-bld, nor even rational. However, they can easily be broken into smaller steps, which are rational—as we are applying a reflexive transitive closure, this is at no expense in generality. This requires more complex encodings of Hardy configurations, with some “finite state control” and a working space in order to keep track of where we are in our small steps. Because we do not want to spend new symbols in this encoding, given some finite set \( Q \) of states, we work on sequences in

\[
\text{Seqs} \defeq \{a_0, a_1\}^{\lceil \log |Q| \rceil} \cdot \{ | \} \cdot p(\Sigma^*_k \#) \cdot \{ \# \}^* \cdot \{ | \} \cdot p(\Sigma^*_k \#) \cdot \{ | \} \cdot \{ \#, a_0, a_1 \}^* \, . \tag{16}
\]

with four segments separated by “\(|\)”: a state, a working segment, an ordinal encoding, and a counter. Given a state \( q \) in \( Q \), we use implicitly its binary encoding as a sequence of fixed length over \( \{a_0, a_1\} \). Our sequences in “normal”
mode look like “q ⊙ π−1(α) : ♯m” with an empty working segment and only #’s as counter symbols.

We define two relations \( F_w \) and \( B_w \) with domain and range \( \text{Seqs} \) that implement forward and backward computations with \( R_H \). A typical case is that of computations with \( R_1 \), which can be implemented as the closure of the union:

\[
q_{Fw} \mid w a_0 \# x \mid \#^{n+2} \quad F_w^1 \quad q_{Fw_1} \mid w \# \mid p(a_0 x) \mid \#^{n+1} a_0
\]

\[
q_{Fw_1} \mid w \#^m \mid x \mid \#^{n+1} a_0^{p+1} \quad F_w^1 \quad q_{Fw_1} \mid w \#^{m+1} \mid x \mid \#^n a_0^{p+2}
\]

\[
q_{Fw_1} \mid w \#^{m+1} \mid x \mid \#^{n+1} a_0^{p+2} \quad F_w^1 \quad q_{Fw_1} \mid w \#^{m+1} \mid x \mid \#^{n+2}
\]

for \( m, n, p \in \mathbb{N} \), \( w \) in \( p(\Sigma_k^*) \), and \( x \) in \( p(\Sigma_k^w) \). Note that \( p(a_0 x) \) returns \( a_0 x \) if \( x \) begins with \( # \) or \( a_0 \), and \( x \) otherwise. The corresponding backward computation for \( R_1 \) inverses the relations in \( (17)-19 \) and substitutes \( q_{Bw} \) for \( q_{Fw_1} \). The reader should be able to convince herself that this is indeed feasible in a rational \( 1\text{-bld} \) fashion; for instance, \( (18) \) can be written as a rational expression

\[
\left[q_{Fw_1}\right], \text{Id}_{\Sigma_k^w} \cdot \left[\#^* \cdot [\varepsilon \#] \cdot [\varepsilon] \cdot \text{Id}_{\Sigma_k^w} \cdot \left[\#^* \cdot \#^* \cdot [\varepsilon \# \varepsilon] \cdot [\varepsilon] \cdot [\varepsilon \#] \cdot [\varepsilon] \right] \cdot \left[\#^* \cdot [\varepsilon \#] \cdot [\varepsilon] \cdot [\varepsilon] \right] \cdot \left[\#^* \cdot [\varepsilon \#] \cdot [\varepsilon] \right] \cdot [\varepsilon] \right].
\]

A full description of \( F_w \) and \( B_w \) can be found in App. B.2.

Observe that separators “\( \cdot \)” are reliable, and that losses cannot pass unnoticed in the constant-sized state segment of a sequence in \( \text{Seqs} \); thus we can use lemmas \( 2 \) and \( 3 \) to prove that \( F_w \) and \( B_w \) are “weak” implementations of \( H^\alpha \) and its inverse when \( \alpha \) is in \( \Omega_k \). Not any reformulation of \( R_H \) as the closure of a rational relation would work here: our relation also needs to be robust to losses; see App. B.2 for details.

**Lemma 4 (Weak Implementation).** The relations \( F_w \) and \( B_w \) are \( 1\text{-bld} \) and terminating. Furthermore, if \( k \geq 1 \), \( m, n > 1 \) and \( \alpha \in \Omega_k \),

\[
q_{Fw} \pi^{-1}(\alpha) \mid \#^m \quad F_w^m \quad q_{Fw} \pi^{-1}(\alpha) \mid \#^n \quad B_w^m \quad q_{Fw} \pi^{-1}(\alpha) \mid \#^n
\]

implies \( m \leq H^\alpha(n) \)

and there exists rewrites verifying \( m = H^\alpha(n) \) in both of the above cases.

### 4 The Parametric Complexity of LR[1-bld]

Now equipped with suitable encodings for Hardy computations, we can turn to the main result of the paper: Prop. 2 below shows the \( F_{\omega,k} \)-hardness of \((k+2)\text{-LR}[1\text{-bld}]\). As we obtain almost matching upper bounds in Section 4.2, we deem this to be rather tight.

#### 4.1 Lower Bound

Thanks to the relations over \( \Sigma_{k+1} \oplus \{1\} \) defined in Section 3, we know that we can weakly compute with \( F_w \) a “budget space” as a unary counter of size \( F_{\omega,k}(n) \), and later check that this budget has been maintained by running through \( B_w \). We are going to insert the simulation of an \( F_{\omega,k} \)-hard problem between these two phases of budget construction and budget verification, thereby constructing \( F_{\omega,k} \)-hard instances of \((k+2)\text{-LR}[1\text{-bld}]\).

**Proposition 2.** Let \( k \geq 2 \). Then \((k+2)\text{-LR}[1\text{-bld}]\) is \( F_{\omega,k} \)-hard.
**Bounded Semi-Thue Reachability.** The problem we reduce from is a space-bounded variant of the semi-Thue reachability problem (aka semi-Thue word problem): as already mentioned in Section 2, a semi-Thue system \( \mathcal{Y} \) over an alphabet is a finite set of rules \((u, v)\) in \(\Sigma^* \times \Sigma^*\), defining a rewrite relation \( \rightarrow_{\mathcal{Y}} \).

**Semi-Thue Reachability (R[Rewr])**

**input** A semi-Thue system \( \mathcal{Y} \) over an alphabet \( \Sigma \), and words \( y \) and \( y' \) in \( \Sigma^* \).

**question** Is it the case that \( y \rightarrow_{\mathcal{Y}}^n y' \)?

This problem is in general undecidable, as one can express the “next configuration” relation of a Turing machine as a semi-Thue system. Its \( F_{\omega k} \)-bounded version for some \( k \geq 1 \) takes as input an instance \((\mathcal{Y}, y, y')\) of size \( n \) where, if \( y \rightarrow_{\mathcal{Y}}^n x \), then \( |x| \leq F_{\omega k}(n) \). This is easily seen to be hard for \( F_{\omega k} \), even for a binary alphabet \( \Sigma \):

**Fact 1.** The \( F_{\omega k} \)-bounded semi-Thue reachability problem is \( F_{\omega k} \)-complete, already for \(|\Sigma| = 2\).

**Reduction.** Let \((\mathcal{Y}, y, y')\) be an instance of size \( n > 1 \) of \( F_{\omega k} \)-bounded R[Rewr] over the two-letters alphabet \(\{a_0, a_1\}\). We build a \((k + 2)\)-LR[1-bld] instance in which the rewrite relation \( R \) performs the following sequence:

1. Weakly compute a budget of size \( F_{\omega k}(n) \), using \( Fw \) described in Section 3.
2. In this allocated space, simulate the rewrite steps of \( \mathcal{Y} \) starting from \( y \).
3. Upon reaching \( y' \), perform a reverse Hardy computation using \( Bw \) and check that we obtain back the initial Hardy configuration. This check ensures that the lossy rewrites were in fact reliable (i.e., no symbols were lost).

For Phase 2, we define a \#-padded version \( \text{Sim} \) of \( \rightarrow_{\mathcal{Y}} \) that works over Seqs:

\[
\text{Sim} \triangleq \{(q_{\text{Sim}} \equiv u\#^p, q_{\text{Sim}} \equiv v\#^q) \mid u \rightarrow_{\mathcal{Y}} v, |u| + p = |v| + q \}.
\]  

(21)

This is a length-preserving rational relation. We define two more length-preserving relations \( \text{Init} \) and \( \text{Fin} \) that initialize the simulation with \( y \) on the budget space, and launch the verification phase if \( y' \) appears there, allowing to move from Phase 1 to Phase 2 and from Phase 2 to Phase 3 respectively:

\[
\text{Init} \triangleq \{(q_{Fw} \equiv y\#^0, q_{\text{Sim}} \equiv y\#^0) \mid |\ell| \geq 0 \}.
\]  

(22)

\[
\text{Fin} \triangleq \{(q_{\text{Sim}} \equiv y'\#^\ell, q_{Bw} \equiv y'\#^\ell) \mid |\ell| \geq 0 \}.
\]  

(23)

Finally, because \( F_{\omega k}(n) = H^{(10)_k}(n) \), we define our source and target by

\[
w \overset{\text{def}}{=} q_{Fw} \equiv a_k^{10} \# a_1^{n} \#,
\]  

\[
w' \overset{\text{def}}{=} q_{Bw} \equiv a_k^{n} \# a_1^{10} \#
\]  

(24)

and we let \( R \) be the 1-bld rational relation \( Fw \cup \text{Init} \cup \text{Sim} \cup \text{Fin} \cup Bw \).

**Claim 1.** The given R[Rewr] instance is positive if and only if \((R, w, w')\) is a positive instance of the \((k + 2)\)-LR[1-bld] problem.

**Proof.** Suppose \( w \overset{R_{\omega k}}{\rightarrow} w' \). It is easy to see that the separator symbol “\#” and the encodings of states from \( Q \) are reliable. Because of the way the relations treat the states, we in fact get

\[
w Fw_{\omega k} (q_{Fw} \equiv \#_k^{+}) \text{ Init}_{\omega k} (q_{Sim} \equiv z_1) \text{ Sim}_{\omega k} (q_{Sim} \equiv z_2) \text{ Fin}_{\omega k} (q_{Sim} \equiv \#_k^{+}) Bw_{\omega k} w' \]
for some strings $z_1, z_2$ and naturals $\ell_1, \ell_2 \in \mathbb{N}$. By Lem. 4, we have $\ell_1 \leq F_{\omega^k}(n)$ and $\ell_2 \geq F_{\omega^k}(n)$. Since $\text{Init}$, $\text{Sim}$, and $\text{Fin}$ are length-preserving, we get

$$F_{\omega^k}(n) \geq \ell_1 \geq |z_1| \geq |z_2| \geq \ell_2 \geq F_{\omega^k}(n) \quad (25)$$

Thus equality holds throughout, and therefore the lossy steps of $\text{Sim}_\emptyset$ in Phase 2 were actually reliable, i.e. were steps of $\text{Sim}$. This allows us to conclude that the original R[Rewr] instance was positive.

Suppose conversely that the R[Rewr] instance is positive. We can translate this into a witnessing run for $w R^{\wedge} w'$, in particular, for $w F w^{\wedge} \downarrow \text{Init} \downarrow \text{Sim}^{\wedge} \downarrow \text{Fin}^{\wedge} w'$, because any successful run from the R[Rewr] instance can be plugged into the $\text{Sim}^{\wedge}$ phase; Lem. 4 and the fact that the configurations of $\Upsilon$ are bounded by $F_{\omega^k}(n)$ together ensure that this can be done.

### 4.2 Upper Bound

#### Well-Structured Transition Systems.

As a preliminary, let us show that the lossy rewriting problem is decidable. Indeed, the relation $R_{\emptyset}$ can be viewed as the transition relation of an infinite transition system over the state space $\Sigma^*$. Furthermore, by Higman’s Lemma, the subword embedding ordering $\sqsubseteq$ is a well quasi ordering (wqo) over $\Sigma^*$, and the relation $R_{\emptyset}$ is compatible with it: if $u R_{\emptyset} v$ and $u \sqsubseteq u'$ for some $u, v, u'$ in $\Sigma^*$, then there exists $v'$ in $\Sigma^*$ s.t. $u' R_{\emptyset} v'$: here it suffices to use $v' = v$ by transitivity of $\sqsubseteq$.

A transition system $S = \langle S, \rightarrow, \sqsubseteq \rangle$ with a wqo $\langle S, \leq \rangle$ as state space and a compatible transition relation $\rightarrow$ is called a well-structured transition system (WSTS), and several problems are decidable on such systems under very light effectiveness assumptions (Abdulla et al., 2000; Finkel and Schnoebelen, 2001), among which the coverability problem, which asks given a WSTS $S$ and two states $s$ and $s'$ in $S$ whether there exists $s'' \geq s'$ s.t. $s \rightarrow^* s''$. The lossy rewrite problem when $w R_{\emptyset} w'$ can be restated as a coverability problem for the WSTS $\langle \Sigma^*, R_{\emptyset}, \sqsubseteq \rangle$ and $w$ and $w'$, since if there exists $w'' \sqsubseteq w'$ with $w R_{\emptyset}^{\wedge} w''$, then $w R_{\emptyset}^{\wedge} w'$ also holds by transitivity of $\sqsubseteq$.

#### Parameterized Upper Bound.

In many cases, a combinatorial algorithm can be employed instead of the classical backward coverability algorithm for WSTS: we can find a particular coverability witness $w' = w_0 \sqsubseteq R^{-1} u_1 \cdots w_{\ell-1} \sqsubseteq R^{-1} w_\ell \sqsubseteq w$ of length $\ell$ bounded by a function akin to $F_{\omega^{k-1}}$ using the Length Function Theorem of (Schmitz and Schnoebelen, 2011). This is a generic technique for coverability explained in (Schmitz and Schnoebelen, 2012), and the reader will find it instantiated for $(k + 2)$-LR[Rat] in App. C.1.

**Proposition 3 (Upper Bound).** The problem $(k + 2)$-LR[Rat] is in $F_{\omega^{k+1}}$.

The small gap of complexity we witness here with Prop. 2 stems from the encoding apparatus, which charges us with one extra symbol. We have not been able to close this gap; for instance, the encoding breaks if we try to work without our separator symbol “$\uparrow$”. 

---

11
5 Applications

We apply in this section the proof of Prop. 2 to prove parametric complexity lower bounds for several problems. In three cases (propositions 4, 5, and 7 below), we proceed by a reduction from the LR problem, but take advantage of the specifics of the instances constructed in the proof Prop. 2 to obtain tighter parameterized bounds. The hardness proof for the LT problem in Prop. 6 requires more machinery, which needs to be incorporated to the construction of Section 4.1 in order to obtain a reduction.

Rational Embedding. We first deal with the classical embedding problem:

We reduce from a \((k+2)\)-LR\([\text{Rat}]\) instance and use Prop. 2. The issue is to somehow convert an iterated composition into an iterated concatenation—the idea is similar to the one typically used for proving the undecidability of PCP.

**Proposition 4.** Let \(k \geq 2\). Then \((k+2)\)-EP\([\text{Rat}]\) is \(F_{\omega^k}\)-hard.

**Proof.** Assume without loss of generality that \(w \neq w'\) in a \((k+2)\)-LR\([\text{Rat}]\) instance \(\langle R, w, w' \rangle\). We consider sequences of consecutive configurations of \(\sqsubseteq \# (\sqsubseteq \# \sqsubseteq \#)\oplus\) of form

\[
(w = v_0 \sqsupset u_0 R v_1 \sqsupset u_1 R v_2 \sqsupset \cdots R v_n \sqsupset u_n = w')
\]

(26)

that prove the LR instance to be positive. Let $ be a fresh symbol; we construct a new relation \(R'\) that attempts to read the \(u_i\)'s on its first component and the \(v_i\)'s on the second, using the $'s for synchronization:

\[
R' \overset{\text{def}}{=} \left[\left[\begin{array}{c}
\$w'\$
\$
\$
\end{array}\right] \cdot \left(\begin{array}{c}
R \\
$
\$
\end{array}\right) \cdot \left[\begin{array}{c}
\$\$
\$
\end{array}\right] \right] + \left[\begin{array}{c}
\epsilon \\
\$
\$
\end{array}\right]
\]

(27)

Observe that in any pair of words \((u, v)\) of \(R'\), one finds the same number of occurrences of the separator $ in \(u\) and \(v\), i.e. we can write \(u = \$u_n\$ \cdots \$u_0\$\) and \(v = \$v_n\$ \cdots \$v_0\$\) with \(n > 0\), verifying \(v_0 = w, u_n = w',\) and \(u_i R v_{i+1}\) for all \(i\).

Assume \(u \sqsubseteq v\): the embedding ordering is restricted by the $ symbols to the factors \(u_i \sqsubseteq v_i\). We can therefore exhibit a sequence of form (26). Conversely, given a sequence of form (26), the corresponding pair \((u, v)\) belongs to \(R' \sqcap \sqsubseteq\).

In order to conclude, observe that we can set $ \overset{\text{def}}{=} \text{p in the proof of Prop. 2} and adapt the previous arguments accordingly, since “\(\text{p}\)” is preserved by \(R\) and appears in both \(w\) and \(w'\) in the particular instances we build.

Synchronous Embedding. Turning now to the case of synchronous relations, we proceed as in the previous proof for Prop. 4, but employ an extra padding symbol \(\perp\) to construct a length-preserving version of the relation \(R\) in an instance of \((k+2)\)-LR\([\text{Sync}]\), allowing us to apply the Kleene star operator while remaining regular.

**Proposition 5.** Let \(k \geq 2\). Then \((k+3)\)-EP\([\text{Sync}]\) is \(F_{\omega^k}\)-hard.

**Proof.** Let \(\langle R, w, w' \rangle\) be an instance of \((k+2)\)-LR\([\text{Sync}]\) with \(w \neq w'\) and let $ and \(\perp\) be two fresh symbols. Because \(R \cdot \{\$, \$\}\) is a synchronous relation, we can construct a padded length-preserving relation

\[
R_{\perp} \overset{\text{def}}{=} \{(u\perp^m, v\perp^p) \mid m, p \geq 0 \land (u, v) \in R \land |u\perp^m| = |v\perp^p|\}
\]

(28)
and define a relation similar to (27):
\[
R' \perp \overset{\text{def}}{=} \left[ w' \right] \cdot R \perp \cdot \left[ \varepsilon \right] \cdot \left[ \varepsilon \right] \cdot \left[ u \right] \cdot \left[ \perp \right].
\] (29)

Let us show that \( R' \perp \) is regular: \( \{ (w, \perp) \} \) and \( \{ (\varepsilon, w) \} \) are relations with bounded length discrepancy and \( R \perp \) is length preserving, thus their concatenation has bounded length discrepancy, and can be effectively computed by resynchronization (Sakarovitch, 2009). Suffixing \( \{ (\varepsilon, \perp) \} \) thus yields a synchronous relation.

As in the proof of Prop. 4, \( R' \perp \) preserves the \$ separators, thus if \((u, v)\) belongs to \( R' \perp \), then we can write
\[
u = \$ u_n \$ \perp_{m_n} u_{n-1} \$ \perp_{m_{n-1}} \cdots \$ \perp_{m_1} u_0 \$ \perp_{m_0},
v = \$ v_n \$ \perp_{p_n} v_{n-1} \$ \perp_{p_{n-1}} \cdots \$ \perp_{p_1} v_0 \$ \perp_{p_0}.\] (30)

with \( n > 0 \) and \( m_n = 0 \). Furthermore, \( v_0 = w, u_n = w' \), and \((u, s\perp_{m_i}, v_{i+1}\perp_{p_{i+1}})\) belongs to \( R \perp \), thus \( u_i R v_{i+1} \) for all \( i \). If the EP instance is positive, i.e. if \( u \sqsubseteq v \), then \( u_i \sqsubseteq v_i \) and \( m_i \leq p_i \) for all \( i \), and we can build a sequence of form (26) proving the LR instance to be positive. Conversely, if the LR instance is positive, there exists a sequence of form (26), and we can construct a pair \((u, v)\) as in (30) above by guessing a sufficient padding amount \( p_0 \) that will allow to carry the entire rewriting.

Finally, as in the proof of Prop. 4, we can set \$ \overset{\text{def}}{=} p_0.

Lossy Termination. In contrast with the previous cases, our hardness proof for the LT problem does not reduce from LR but directly from a semi-Thue word problem, by adapting the proof of Prop. 2 in such a way that \( R^\omega \) is guaranteed to terminate. The main difference is that we reduce from a semi-Thue system where the length of derivations is bounded, rather than the length of configurations—this is still \( F_{\omega, k} \)-hard since the distinction between time and space complexities is insignificant at such high complexities. The simulation of such a system then builds two copies of the initial budget in Phase 1: a space budget, where the derivation simulation takes place, and a time budget, which gets decremented with each new rewrite of Phase 2 and enforces its termination even in case of losses. See App. D.1 for details.

**Proposition 6.** Let \( k \geq 2 \). Then \((k+2)-LT[1-bld]\) is \( F_{\omega, k} \)-hard.

Lossy Channel Systems. By over-approximating the behaviours of a channel system by allowing uncontrolled, arbitrary message losses, Abdulla, Cécé, et al. (Cécé et al., 1996; Abdulla and Jonsson, 1996) obtain decidability results on an otherwise Turing-complete model. Many variants of this model have been studied in the literature (Chambart and Schnoebelen, 2007, 2008a; Jančar et al., 2012), but our interest here is that LCSs were originally used as the formal model for the \( F_{\omega, \omega} \) lower bound proof of Chambart and Schnoebelen (2008a), rather than a PEP.

Formally, a **lossy channel system** (LCS) is a finite labeled transition system \((Q, \Sigma, \delta)\) where transitions in \( \delta \subset Q \times \{?,!\} \times \Sigma \times Q \) read and write on an unbounded channel. An channel system defines an infinite transition system over its set of configurations \( Q \times \Sigma^* \)—holding the current state and channel
content—, with transition relation \( q, x \to q', x' \) if either \( \delta \) holds a read \((q, ?m, q')\) and \( x = mx' \), or if it holds a write \((q, !m, q')\) and \( xm = x' \). The operational semantics of an LCS then use the lossy version \( \to \sqsubseteq \) of this transition relation.

In the following, we consider a slightly extended model, where transitions carry sequences of instructions instead, i.e. \( \delta \) is a finite set included in \( Q \times ([?, !] \times \Sigma)^* \times Q \). The natural decision problem associated with a LCS is its reachability problem:

**Lossy Channel System Reachability (LCS)**

**input** A LCS \( C \) and two configurations \((q, x)\) and \((q', x')\) of \( C \).

**question** Is \((q', x')\) reachable from \((q, x)\) in \( C \), i.e. does \( q, x \to^{\sqsubseteq} q', x' \)?

The lossy rewriting problem easily reduces to a reachability problem in a LCS: the LCS cycles through the channel contents thanks to a distinguished symbol, and applies the rational relation at each cycle; see App. D.2 for details.

**Proposition 7.** Let \( k \geq 2 \). Then \((k + 2)\)-LCS is \( \text{F}_{\omega}^k \)-hard.

### 6 Concluding Remarks

Post embedding problems provide a high-level packaging of hyper-Ackermannian decision problems—and thanks to our parametric bounds, for \( k \)-Ackermannian problems—, compared to e.g. reachability in lossy channel systems (as used in [Chambart and Schnoebelen 2008c]). The lossy rewriting problem is a prominent example: because it is stated in terms of a rational relation instead of a machine definition, it beneficiates automatically from the theoretical toolkit and multiple characterizations associated with rational relations. For a simple example, the *increasing* rewriting problem, which employs \( R \sqcap \sqsubseteq \text{def} = \sqsubseteq R \sqsubseteq \) instead of \( R \sqsubseteq \), is immediately seen to be equivalent to LR, by substituting \( R^{-1} \) for \( R \) and exchanging \( w \) and \( w' \).

Interestingly, this inversion trick allows to show the equivalence of the lossy and increasing variants of all our problems, except for lossy termination:

**Increasing Termination (IT[Rat])**

**input** A rational relation \( R \) over \( \Sigma \) and a word \( w \) in \( \Sigma^* \).

**question** Does \( R \sqcap \sqsubseteq \) terminate from \( w \)?

A related problem, termination of increasing channel systems with emptiness tests, is known to be in \( \text{F}_3 \) ([Bouver et al., 2012]) instead of \( \text{F}_{\omega} \) for LCS termination, but IT[Rat] is more involved. Like LR[Rat] or EP[Rat], it provides a high-level description, this time of *fair termination* problems in increasing channel systems, which are known to be equivalent to satisfiability of *safety metric temporal logic* ([Ouaknine and Worrell, 2007, 2006; Jenkins, 2012]). The exact complexity of IT[Rat] is open, with a gigantic gap between the \( \text{F}_{\omega} \) upper bound provided by WSTS theory, and an \( \text{F}_4 \) lower bound by [Jenkins, 2012].

### References

Abdulla, P.A. and Jonsson, B., 1996. Verifying programs with unreliable channels. *Information and Computation*, 127(2):91–101. doi:10.1006/inco.1996.0053.
Abdulla, P.A., Čeransk, K., Jonsson, B., and Tsay, Y.K., 2000. Algorithmic analysis of programs with well quasi-ordered domains. *Information and Computation*, 160:109–127. [doi:10.1006/inco.1999.2843](https://doi.org/10.1006/inco.1999.2843).

Atig, M.F., Bouajjani, A., Burchhardt, S., and Musuvathi, M., 2010. On the verification problem for weak memory models. In *POPL 2010*, 37th Annual Symposium on Principles of Programming Languages, pages 7–18. ACM. [doi:10.1145/1706299.1706303](https://doi.org/10.1145/1706299.1706303).

Barceló, P., Figueira, D., and Libkin, L., 2012. Graph logics with rational relations and the generalized intersection problem. In *LICS 2012*, 27th Annual IEEE Symposium on Logic in Computer Science, pages 115–124. IEEE Press. [doi:10.1109/LICS.2012.23](https://doi.org/10.1109/LICS.2012.23).

Bouyer, P., Markey, N., Ouaknine, J., Schnoebelen, Ph., and Worrell, J., 2012. On termination and invariance for faulty channel machines. *Formal Aspects of Computing*, 24(4):95–106. [doi:10.1007/s00165-012-0234-7](https://doi.org/10.1007/s00165-012-0234-7).

Cécé, G., Finkel, A., and Purushothaman Iyer, S., 1996. Unreliable channels are easier to verify than perfect channels. *Information and Computation*, 124(1):20–31. [doi:10.1006/inco.1996.0003](https://doi.org/10.1006/inco.1996.0003).

Chambart, P. and Schnoebelen, Ph., 2007. Post embedding problem is not primitive recursive, with applications to channel systems. In Arvind, V. and Prasad, S., editors, *FSTTCS 2007*, 27th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, volume 4855 of Lecture Notes in Computer Science, pages 265–276. Springer. [doi:10.1007/978-3-540-77050-3_22](https://doi.org/10.1007/978-3-540-77050-3_22).

Chambart, P. and Schnoebelen, Ph., 2008a. Mixing lossy and perfect fifo channels. In van Breugel, F. and Chechik, M., editors, *CONCUR 2008*, 19th International Conference on Concurrency Theory, volume 5201 of Lecture Notes in Computer Science, pages 340–355. Springer. [doi:10.1007/978-3-540-85361-9_28](https://doi.org/10.1007/978-3-540-85361-9_28).

Chambart, P. and Schnoebelen, Ph., 2008b. The $\omega$-regular Post embedding problem. In Amadio, R., editor, *FoSSaCS 2008*, 11th International Conference on Foundations of Software Science and Computational Structures, volume 4962 of Lecture Notes in Computer Science, pages 97–111. Springer. [doi:10.1007/978-3-540-78499-9_8](https://doi.org/10.1007/978-3-540-78499-9_8).

Chambart, P. and Schnoebelen, Ph., 2008c. The ordinal recursive complexity of lossy channel systems. In *LICS 2008*, 23rd Annual IEEE Symposium on Logic in Computer Science, pages 205–216. IEEE Press. [doi:10.1109/LICS.2008.47](https://doi.org/10.1109/LICS.2008.47).

Chambart, P. and Schnoebelen, Ph., 2010a. Computing blocker sets for the regular Post embedding problem. In *DLT 2010*, International Conference on Developments in Language Theory, volume 6224 of Lecture Notes in Computer Science, pages 136–147. Springer. [doi:10.1007/978-3-642-14155-4_14](https://doi.org/10.1007/978-3-642-14155-4_14).

Chambart, P. and Schnoebelen, Ph., 2010b. Pumping and counting on the regular Post embedding problem. In Abramsky, S., Meyer auf der Heide, F., and Spirakis, P., editors, *ICALP 2010*, 37th International Colloquium on Automata, Languages and Programming, volume 6199 of Lecture Notes in Computer Science, pages 64–75. Springer. [doi:10.1007/978-3-642-14162-1_6](https://doi.org/10.1007/978-3-642-14162-1_6).

Elgot, C.C. and Mezei, J.E., 1965. On relations defined by generalized finite automata. *IBM Journal of Research and Development*, 9(1):47–68. [doi:10.1147/rd.91.0047](https://doi.org/10.1147/rd.91.0047).

Fairtlough, M. and Wainer, S.S., 1998. Hierarchies of provably recursive functions. In Buss, S., editor, *Handbook of Proof Theory*, volume 137 of Studies in Logic and the Foundations of Mathematics, chapter III, pages 149–207. Elsevier. [S0049-237X(98)80018-9](https://doi.org/10.1016/S0049-237X(98)80018-9).

Finkel, A. and Schnoebelen, Ph., 2001. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1–2):63–92. [doi:10.1016/S0304-3975(00)00102-X](https://doi.org/10.1016/S0304-3975(00)00102-X).
Friedman, H.M., 1999. Some decision problems of enormous complexity. In LICS 1999, 14th Annual IEEE Symposium on Logic in Computer Science, pages 2–13. IEEE Press. doi:10.1109/LICS.1999.782577.

Haddad, S., Schmitz, S., and Schnoebelen, Ph., 2012. The ordinal-recursive complexity of timed-arc Petri nets, data nets, and other enriched nets. In LICS 2012, 27th Annual IEEE Symposium on Logic in Computer Science, pages 355–364. IEEE Press. doi:10.1109/LICS.2012.46.

Jančar, P., Karandikar, P., and Schnoebelen, Ph., 2012. Unidirectional channel systems can be tested. In Baeten, J., Ball, T., and de Boer, F., editors, IFIP TCS 2012, 7th IFIP International Conference on Theoretical Computer Science, volume 7604 of Lecture Notes in Computer Science, pages 149–163. Springer. doi:10.1007/978-3-642-33475-7_11.

Jenkins, M., 2012. PhD thesis, Oxford University. In preparation.

Karandikar, P. and Schnoebelen, Ph., 2012. Cutting through regular Post embedding problems. In CSR 2012, 7th International Computer Science Symposium in Russia, volume 7353 of Lecture Notes in Computer Science, pages 229–240. Springer. doi:10.1007/978-3-642-30642-6_22.

Lasota, S. and Walukiewicz, I., 2008. Alternating timed automata. ACM Transactions on Computational Logic, 9(2):10. doi:10.1145/1342991.1342994.

Latteux, M. and Leguy, J., 1983. On the composition of morphisms and inverse morphisms. In Diaz, J., editor, ICALP 1983, 10th International Colloquium on Automata, Languages and Programming, volume 154 of Lecture Notes in Computer Science, pages 420–432. Springer. doi:10.1007/BFb0036926.

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

Nivat, M., 1968. Transduction des langages de Chomsky. Ann. Inst. Fourier, 18:339–455.

Ouaknine, J.O. and Worrell, J.B., 2006. Safety Metric Temporal Logic is fully decidable. In Hermanns, H. and Palsberg, J., editors, TACAS 2006, 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, volume 3920 of Lecture Notes in Computer Science, pages 411–425. Springer. doi:10.1007/11691372_27.

Ouaknine, J.O. and Worrell, J.B., 2007. On the decidability and complexity of Metric Temporal Logic over finite words. Logical Methods in Computer Science, 3(1):8. doi:10.2168/LMCS-3(1:8)2007.

Rose, H.E., 1984. Subrecursion: Functions and Hierarchies, volume 9 of Oxford Logic Guides. Clarendon Press.

Sakarovitch, J., 2009. Elements of Automata Theory. Cambridge University Press.

Schmitz, S. and Schnoebelen, Ph., 2011. Multiply-recursive upper bounds with Higman’s Lemma. In ICALP 2011, International Colloquium on Automata, Languages and Programming, volume 6756 of Lecture Notes in Computer Science, pages 441–452. Springer. doi:10.1007/978-3-642-22012-8_35.

Schmitz, S. and Schnoebelen, Ph., 2012. Algorithmic Aspects of WQO Theory. Lecture Notes. http://cel.archives-ouvertes.fr/cel-00727025.
A Unary Alphabet: Prop. 1

We prove here Prop. 1: 1-EP[Rat] is in NLogSpace.

Parikh Images. The proof employs the semilinear view of unary rational relations: a semilinear set $S$ is a subset of $\mathbb{Z}_k^2$ described by a finite union of linear sets $(b, P)$ defined as $\{ b + \sum_{i=1}^m x_i p_i \mid x_1, \ldots, x_m \in \mathbb{N} \}$ where $b$ is a base in $\mathbb{Z}_k$ and $P$ is a set of $m$ periods $p_1, \ldots, p_m$, each in $\mathbb{Z}_k$. It is well known that the Parikh image (aka commutative image) $\Psi(L)$ of a regular language $L$ over an alphabet $\Sigma$ is a semilinear set in $\mathbb{N}|\Sigma|$ telling for each symbol of $\Sigma$ how many times it occurs in a word of $L$. Formally, let $\Sigma = \{ a_1, \ldots, a_n \}$, then a vector $u$ is in $\Psi(L)$ iff there exists a word $u$ in $L$ s.t. for all $1 \leq i \leq n$, $u(i) = |u|_{a_i}$, the number of occurrences of $a_i$ in $u$.

Proof of Prop. 1. Given a rational relation $R$ over the unary alphabet $\Sigma = \{ a \}$, we can view its normalized transducer $T = \langle Q, \Sigma, \Sigma, \delta, I, F \rangle$ as a nondeterministic finite automaton $A = \langle Q, \Delta, \delta, I, F \rangle$ over the two-letters alphabet $\Delta = \{ (a, \varepsilon), (\varepsilon, a) \}$. The Parikh image of $L(A)$ is then a semilinear set $S \subseteq \mathbb{N}^2$ verifying

$$S = \{ (m, n) \mid (a^m, a^n) \in R \}. \quad (31)$$

Assume $R \cap \emptyset \neq \emptyset$, i.e. there exists a pair $(m, n)$ in $S$ with $m \leq n$. Then, there exists a linear set $(b, P)$ in $S$ s.t. either $b = (b_1, b_2)$ with $b_1 \leq b_2$, or $b_1 > b_2$ but there exists a period $p = (p_1, p_2)$ in $P$ verifying $p_1 < p_2$—and then there exists $x \in \mathbb{N}$ s.t. $b_1 + xp_1 \leq b_2 + xp_2$.

It therefore suffices to check in NLogSpace for the existence of such a non-decreasing basis $b$ or such an increasing period $p$ in the normalized transducer $T$ for $R$. This is rather straightforward:

- a basis $b$ is read along a simple accepting run in $T$, hence a run of length at most $|Q|$, while
- a period $p$ is read along a simple loop on some state $q$ of $Q$; we have to check that $q$ is both accessible and co-accessible, thus $q$ should lie on an accepting run of length at most $2|Q|$ and exhibit a loop of length at most $|Q|$.

In both cases it suffices to guess a suitable accepting run to find such a $b$ or $p$. 

\[\square\]

B Codes and Hardy Computations

B.1 Pure Codes: Lem. 1

Proof of Lem. 1. Remember that ordinals are supplied with a left subtraction operation, because if $\beta \leq \beta'$, then $\beta' - \beta$ can be defined as the unique ordinal verifying $\beta + (\beta' - \beta) = \beta'$. We define an inverse $\pi^{-1}$ by induction on the CNF of ordinals; this function yields pure codes exclusively:

$$\pi^{-1}(0) \overset{\text{def}}{=} \varepsilon, \quad \pi^{-1}\left(\sum_{i=1}^{p} \omega^{\beta_i}\right) \overset{\text{def}}{=} \beta^{-1}(\beta_p)\#\pi^{-1}\left(\sum_{i=1}^{p-1} \omega^{\beta_i - \beta_p}\right). \quad \square$$
B.2 Computing with Rational Relations: Lem. 5

Forward and Backward Rules. We present here the two relations $Fw$ and $Bw$ under the understanding that they are suitably restricted to sequences in $\text{Seqs}$. The relations below are rational and even 1-bld. It suffices to give the relations for $Fw$, as $Bw$ just reverses their direction and uses states $q_{Bw}$, $q_{Bw_1}$, and $q_{Bw_2}$ instead of $q_{Fw}$, $q_{Fw_1}$, and $q_{Fw_2}$. For $R_0$ given in (13),

$$q_{Fw} \perp \# x : \#^{n+1}$$

for all $n \in \mathbb{N}$ and $x$ in $\Sigma_{k\#}^\ast$. Let us repeat the rules for $R_1$ given in (17)-(19):

$$q_{Fw} \perp w a_{0} \# x : \#^{n+2}$$

(17)

$$q_{Fw_1} \perp w \#^m : \#^{n+1}$$

(18)

$$q_{Fw_1} \perp w \#^{m+1} : \#^{n+2}$$

(19)

where $m, n, p$ range over $\mathbb{N}$, $w$ over $\Sigma_k^\ast$, and $x$ over $\Sigma_{k\#}^\ast$. For $R_2$ defined in (15),

$$q_{Fw_2} \perp w a_{1} \# x : \#^{n+2}$$

(33)

$$q_{Fw_2} \perp w a_{1} \# x : \#^{n+1}$$

(34)

$$q_{Fw_2} \perp w a_{1} \# x : \#^{n+2}$$

(35)

for $i > 0$, $m, n, p$ in $\mathbb{N}$, $w$ in $\Sigma_k^\ast$, and $x$ in $\Sigma_{k\#}^\ast$.

We define $Fw = Fw_0 \cup Fw_1 \cup Fw_2$ and analogously for $Bw$. The first thing to check is that the reflexive transitive closures of $Fw$ and $Bw$ implement those of $R_{Bw}$ and $R_{Bw}^{-1}$ as advertised. A helpful notion is that of a phase of a state $q$, which is a sequence of rewrites of form

$$(q_R \perp c_0) \quad R (q \perp x_1 \perp c_1) \quad R \cdots \quad R (q \perp x_m \perp c_m) \quad R (q_R \perp c_{m+1})$$

(36)

for some $c_i$s in $\text{Confs}$ and $x_i$s in $\Sigma_{k\#}^\ast$, where $R$ is $Fw$ or $Bw$ and thus $q_R$ is the corresponding state $q_{Fw}$ or $q_{Bw}$, and $q$ is an intermediate state among $(q_{Fw_1}, q_{Fw_2}, q_{Bw_1}, q_{Bw_2})$. The idea is that a phase ought to simulate exactly the effect of a single step $c_0 R_{Bw} c_m$ or $c_0 R_{Bw}^{-1} c_m$.

Lemma 5 (Correctness of $Fw$ and $Bw$). Let $j$ be in $\{0, 1, 2\}$ and $c, c'$ be in $\text{Confs}$. Then $(q_{Fw} \perp c) \quad Fw_j \perp (q_{Fw} \perp c')$ iff $(q_{Bw} \perp c') \quad Bw_j \perp (q_{Bw} \perp c)$ iff $c \quad R_{Bw}^\circ j \perp c'$.

Proof. The proof is conducted by a case analysis. Because $Bw$ is the inverse of $Fw$ with substituted state names, it suffices to show the equivalence of $(q_{Fw} \perp c) \quad Fw_j \perp (q_{Fw} \perp c')$ with $c \quad R_{Bw}^\circ j \perp c'$. For $j = 0$, the correctness of $Fw_0 = \{(q_{w_1}, q_{Fw})\} \cdot R_0$ is immediate.

For $j = 1$, it suffices to consider a single step of $R_1$, i.e. a pair of $c = w a_{0} \# x : \#^{n+2}$ and $c' = w \#^{n+2} p(a_0 x) : \#^{n+2}$ with $n \in \mathbb{N}$, $w$ in $\Sigma_k^\ast$, and $x$ in $\Sigma_{k\#}^\ast$. Then we have a rewrite sequence

$$(q_{Fw} \perp c) \quad Fw_1 \perp (q_{Fw_1} \perp w \# : p(a_0 x) : \#^{n+1} a_0)$$

(17)

$$(q_{Fw_1} \perp w \# : p(a_0 x) : \#^{n+2} a_0)$$

(18)

$$(q_{Fw_2} \perp w \# : p(a_0 x) : \#^{n+2})$$

(19)

$$= (q_{Fw} \perp c').$$

ii
Conversely, it suffices to consider a phase of $q_{FW_1}$. It is necessarily of the form above, because for $[19]$ to be applicable, the counter segment must be in $\{\#\}^*$, but the first step $[17]$ puts an $a_0$ at the end of the segment. Thus $FW_1$ has to go through the appropriate number of applications of $[18]$. Therefore, a phase of $q_{FW_1}$ implies a rewrite of $R_1$.

We leave the case of $j = 2$ as an exercise for the reader, as it is very similar to that of $j = 1$. 

B.2.1 Proof of Lem. 4

The lemma contains several statements. The fact that $FW$ and $BW$ are rational 1-bld is by definition. That they are terminating is because they check that their counters are larger than 1 in limit steps. Regarding weak implementation, thanks to Lem. 2 and Lem. 3, we know that computations using $RH$ are weak implementations in the sense of Lem. 4. Therefore, it remains to prove that the small steps defined for $FW$ and $BW$ (i) correctly implement the rules of $RH$ and (ii) are “robust” to losses.

Point (i) was the topic of Lem. 5, which in combination with Lem. 3 proves the existence of rewrites

\[(q_{FW} \sqcup \pi^{-1}(\alpha) : \#^n) F_{1}^{\circ} (q_{FW} \sqcup \#^m)\]

(37)

\[(q_{BW} \sqcup \#^m) B_{1}^{\circ} (q_{BW} \sqcup \pi^{-1}(\alpha) : \#^n)\]

(38)

with $m = H^\alpha(n)$.

Turning to point (ii), in order to prove that a rewrite of form $[37]$ implies $m \leq H^\alpha(n)$, we want to transform it into a rewrite according to $(RH)^\ast$, which is known to imply $m \leq H^\alpha(n)$ thanks to Lem. 2 and Lem. 3. We conduct a similar proof later for $[38]$ with $(R_{j-1})^\ast$, proving $[38]$ to imply $m \geq H^\alpha(n)$.

Claim 2. If $(q_{FW} \sqcup c) F_{1}^{\circ} (q_{FW} \sqcup c')$, then $c \ (RH)^* c'$. 

Note that this holds trivially for $FW_0$, thus as in the proof of Lem. 5 we can focus on lossy phases of form

\[(q_{FW} \sqcup c_0) Fw (q \sqcup w_1 \sqcup c_1) Fw^{\circ} \cdots Fw^{\circ} (q \sqcup w_m \sqcup c_m) Fw (q_{FW} \sqcup c_{m+1})\]

(39)

for some intermediate state $q$. We will deal with lossy phases of $q_{FW_1}$ here; the proof of the claim for $q_{FW_2}$ is similar.

Proof of Claim for $FW_1$. First observe that in a lossy phase like $[39]$ for $q_{FW_1}$, because $[17]$ and $[19]$ are used at the beginning and the end of the phase, each intermediate $q \sqcup w_1 \sqcup c_1$ is necessarily in the language

\[L_1 \overset{\text{def}}{=} \{q_{FW_1} \sqcup w \#^{\ell+1} \sqcup x \sqcup \#^n a_0^p \mid w \in p(\Sigma_k^+), x \in p(\Sigma_k^{\#}), \ell \geq 0, p > 0, n + p \geq 2\}.

(40)

Define the atomic embedding relation over an alphabet $\Gamma$ as

\[\sqcup \overset{\text{def}}{=} \{(ww', waw') \mid a \in \Gamma, w, w' \in \Gamma^*\}.

(41)

Clearly,

\[\sqcup = \sqcup^*\]

(42)
Moreover, if \( y \) and \( y' \) are two sequences in \( L_1 \) with \( y \sqsubseteq y' \), then we can find \( y_0, y_1, \ldots, y_n \) all in \( L_1 \) s.t. \( y = y_0 \sqsubseteq y_1 \sqsubseteq \cdots \sqsubseteq y_n \), i.e. we can find suitable atomic embeddings while remaining in \( L_1 \).

Write \( F_{w_{18}} \) for the subrelation of \( F_w \) defined by (18) and \( F_{w_{19}} \) for that of (19). Let us show that these subrelations verify

\[
(\sqcap_1 \sqcup F_{w_{18}}) \subseteq (F_{w_{18}} \sqcup \sqcap_1) \quad (\sqcap_1 \sqcup F_{w_{19}}) \subseteq (F_{w_{19}} \sqcup \sqcap_1) \quad (\sqcap_1 \sqcup \sqcap_1) \subseteq (F_{w_{18}} \sqcup F_{w_{19}} \sqcup \sqcap_1) \quad (43)
\]

over \( L_1 \times L_1 \). This is immediate in most cases, but there is a non-trivial case that justifies the use of transitive closures in (43) for (19): \( \sqcap_1 \sqcup F_{w_{18}} \; \sqcup \; (F_{w_{19}} \sqcup (\sqcap_1 \sqcup \sqcap_1) \; \sqcup \; \sqcap_1) \; \sqcup \; \sqcap_1 \)

should be rewritten into

\[
q_{F_{w_{1}} \sqcup F_{w_{18}}} \; \sqcup \; (F_{w_{19}} \sqcup (\sqcap_1 \sqcup \sqcap_1) \; \sqcup \; \sqcap_1) \; \sqcup \; \sqcap_1
\]

To wrap up the proof of the claim, observe that we can apply repeatedly (43) to a lossy phase like (39) until we have obtained a proper phase of the form

\[
(q_{F_{w_{1}} \sqcup F_{w_{18}}} \sqcup \sqcap_1) \quad F_{w_{17}} \quad (q_{F_{w_{1}} \sqcup F_{w_{18}}} \sqcup \sqcap_1) \quad \cdots \quad (q_{F_{w_{1}} \sqcup F_{w_{18}}} \sqcup \sqcap_1) \quad (44)
\]

Therefore, by Lem. 5, \( c_0 \) \( R_1 \sqcup c_m \) as desired.

Let us turn to the backward version of the claim:

**Claim 3.** If \( (q_{B_w} \sqcup c) \; B_w \sqcap \; (q_{B_w} \sqcup c') \), then \( c \; (R_{1}^{1} \sqcup c') \).

**Proof.** We proceed as in the proof of the previous claim, by considering lossy phases and transforming them into reliable ones. Focusing on \( B_w \), the crux of the argument mirrors (43) with

\[
(B_{w_j} \sqcap \sqcap_1) \subseteq (\sqcap_1 \sqcap B_{w_j}) \quad (45)
\]

over \( L_1 \times L_1 \) for \( j \) in \( \{18, 19\} \). The cases can be solved rather easily thanks to the restriction to \( L_1 \) defined in (40). For instance,

\[
q_{B_w} \sqcup w \#^m x \#^n + 2 \quad B_{w_{18}} \quad q_{B_w} \sqcup w \#^m x \#^n + 2
\]

necessarily has \( m > 0 \) in order to belong to \( L_1 \), thus can be rewritten into

\[
q_{B_w} \sqcup w \#^m x \#^n + 2 \quad \sqcap_1 \quad q_{B_w} \sqcup w \#^m x \#^n + 2
\]

Similar arguments can be used to complete the proof.

\[\square\]
C Complexity Bounds

C.1 Upper Bounds: Prop. 3

Coverability Algorithm. The algorithm for deciding coverability in WSTS is known as the backward coverability algorithm: given an instance $\langle R, w, w' \rangle$ with $w \neq w'$, the algorithm starts with the upward-closure $\subseteq \{\{w'\}\}$ of $w'$ as initial set of potential targets $I_0$. The algorithm then builds the set of predecessors $I_1 = I_0 \cup R^{-1}(I_0) = I_0 \cup \subseteq (R^{-1}(I_0))$: any sequence that covers $w'$ has to go through $I_1$. 

This process is repeated with $I_{i+1} = I_i \cup (R^{-1}(I_i))$ until stabilization, which occurs since upward-closed subsets of a wqo display the ascending chain condition: there exists $n$ s.t. $I_{n+1} = I_n$. As this $I_n$ contains all the words in $\Sigma^*$ that can cover $w'$, it remains to check whether $w$ belongs to the set or not. This algorithm is effective because, although the sets $I_i$ are infinite, they can be represented by their $\subseteq$-minimal elements, which are in finite number thanks to the wqo.

Controlled Sequence. When moving from decidability issues to complexity ones, we need to measure the complexity of basic operations in the previous algorithm. The key computation here is that of a minimal element $u_{i+1}$ in $I_{i+1}$ given a minimal element $u_i$ of $I_i$. Since $u_{i+1}$ is minimal, it is produced from some $v_i \supseteq u_i$ s.t. $u_{i+1} R v_i$, i.e. $u_i = a_1 \cdots a_m$ and $v_i = v'_0 a_1 v_1 \cdots v'_{m-1} a_m v_m$ for some $a_j \in \Sigma$ and $v'_j \in \Sigma^*$.

Given $T = \langle Q, \Sigma, \Sigma, \delta, I, F \rangle$ a normalized transducer for $R$, we know the accepting run with $v_i$ as image is of form

$$ q_0 \xrightarrow{(u'_0,v'_0)} q'_0 \xrightarrow{(e,a)} q_1 \xrightarrow{(u'_1,v'_1)} q'_1 \cdots q_{m-1} \xrightarrow{(u'_{m-1},v'_{m-1})} q'_{m-1} \xrightarrow{(e,a)} q_m \xrightarrow{(u'_m,v'_m)} q'_m $$

with $q_0 \in I$, $q'_m \in F$, and $u_{i+1} = u'_0 u'_1 \cdots u'_{m-1} u'_m$ as input. Then none of the segments $q'_j \xrightarrow{(u'_j,v'_j)} q'_j$ can have length greater than $|Q|$, or $u_{i+1}$ would not be a minimal element of $I_{i+1}$. Therefore, $|u_{i+1}| \leq |Q| \cdot (|a| + 1)$, and any $u_{i+1}$ can be computed in NLogSpace.

Proof of Prop. 3. The idea of our combinatorial algorithm is to derive an upper bound on the length of a sequence proving reachability. A nondeterministic algorithm can then explore this search space for a witness.

Assume the $(k+2)$-LR[Rat] instance to be positive. We consider now a sequence of upward-closed sets $\subseteq \{\{w'\}\} = I_0 \subseteq I_1 \subseteq \cdots \subseteq I_\ell$ such that $w$ is in $I_\ell$ but not in $I_i$ for any $i < \ell$, i.e. we do not wait for saturation of the $I_i$’s but stop as soon as $w$ appears. We can extract a particular minimal element $u_{i+1}$ in each $I_{i+1} \setminus I_i$. Let $g(x) = |Q| \cdot (x + 1)$; by the previous analysis, $|u_{i+1}| \leq g(|u_i|)$, and of course $|v_0| = |w'|$. The sequence $u_0, u_1, \ldots, u_\ell$ is a bad sequence: for all $i < j$, $u_i \not\subseteq u_j$. By the Length Function Theorem (Schmitz and Schnoebelen [2011]), the length $\ell$ is bounded by the Cichón function $h_{\omega+\omega+1}((k-1)|w'|)$ relativized with $h(x) = x \cdot g(x) = |Q|x^2 + |Q|x$.

A nondeterministic algorithm can then set $w_0 = w'$ and guess one by one a sequence of $\ell$ words $w_i$ in $I_i$ with $w_{i+1} \subseteq (R^{-1}\{\{w_i\}\})$ until $w_\ell \subseteq w$. The space required at each step is logarithmic in $|w_i|$, which is bounded overall by the Hardy function $h_{\omega+\omega+1}((k-1)|w'|)$ for the same relativized $h$. 

\v
Finally, given an instance \( \langle R, w, w' \rangle \) of \((k + 2)\)-LR[Rat] of size \( n \), as \( n \geq |Q| \) we can use \( h(x) = x^3 + x^2 \) instead and bound the required space of each step by \( h^{\omega_{k+1}}(n) \). The space requisites of this algorithm place it in \( F_{\omega_{k+1}} \), as the function \( h \) is polynomial.

D Applications

D.1 Lossy Termination: Prop. 6

We prove in this section Prop. 6: \((k + 2)\)-LT[Rat] is \( F_{\omega_k} \)-hard.

Proof Sketch of Prop. 6. We need for this proof to examine more carefully the construction in Section 4.1. The following facts are decisive:

1. both \( F_w \) in Phase 1 and \( B_w \) in Phase 3 are terminating relations,
2. the simulation of the semi-Thue system \( \Upsilon \) in Phase 2 can be carried instead with a “time budget”: it employs sequences of the form \( \gamma \cdot \# \cdot t \), where \( \gamma \) encodes a sequence of Seqs and \( t \) tells how many steps are still allowed—initially the same budget allocated by Phase 1, but decremented by 1 at each rewrite. This allows to simulate a time-bounded semi-Thue system instead of a space-bounded one, but they are equivalent as far as \( F_{\omega_k} \) is concerned.

Let us detail a bit further the changes we carry. The new relation \( R' \) has to be modified to work on words in \( \text{Seqs} \cdot \{1\} \cdot \{\#\}^* \). The relation \( F_w' \) for Phase 1 needs to duplicate its counter increments on both sides of the last separator \( p \) in (32), which becomes

\[
q_{F_w} \equiv \#x \cdot \#^n \cdot \#^n \cdot F_w'_0 \equiv \#^{n+1} \cdot \#^{n+1}.
\]  

(47)

The other cases of \( F_w' \) are based on those of \( F_w \) and additionally duplicate the contents after the last “\#”: for instance, for (18):

\[
q_{F_{w_1}} \equiv w \cdot \#^m \cdot x \cdot \#^{n+1} \cdot a_{0}^{p+1} \cdot z \cdot F_w' \equiv q_{F_{w_1}} \equiv w \cdot \#^{m+1} \cdot x \cdot \#^{n+2} \cdot a_{0}^{p+2} \cdot z.
\]  

(48)

The simulation relation \( \text{Sim}' \) for Phase 2 now decrements the time budget:

\[
\text{Sim}' \equiv \text{Sim} \cdot \left[ \begin{array}{c} 1 \\ \# \\ \# \end{array} \right] \cdot \left[ \begin{array}{c} \#^* \\ \\ \end{array} \right] \cdot \left[ \begin{array}{c} \# \\ \# \end{array} \right].
\]  

(49)

The other relations can be taken to simply preserve the time budget:

\[
B_w' \equiv B_w \cdot \left[ \begin{array}{c} 1 \\ \# \end{array} \right] \cdot \left[ \begin{array}{c} \#^* \\ \# \end{array} \right],
\]  

(50)

\[
\text{Init}' \equiv \text{Init} \cdot \left[ \begin{array}{c} 1 \\ \# \end{array} \right] \cdot \left[ \begin{array}{c} \#^* \\ \# \end{array} \right],
\]  

(51)

\[
\text{Fin}' \equiv \text{Fin} \cdot \left[ \begin{array}{c} 1 \\ \# \end{array} \right] \cdot \left[ \begin{array}{c} \#^* \\ \# \end{array} \right].
\]  

(52)
We add a new relation $\text{End}$ that enters an infinite loop if the full simulation has been carried:

$$\text{End} \overset{\text{def}}{=} \left( \left[ q_{\text{End}} \uplus a_{k-1}^n \#_{\#}^n \right] \cdot \left[ \#_{\#}^n \right] \right) + \left( \left[ q_{\text{End}} \right] \cdot \text{Id}_{\Sigma^* \uplus \mathcal{W}^*{\{\}^*}} \right). \quad (53)$$

Finally, the source sequence becomes

$$w \overset{\text{def}}{=} q_{\text{Fw}} \uplus a_{k-1}^n \#_{\#}^n. \quad (54)$$

The reader can check that the defined relation $R'$ is 1-bld and rational, and that the constructed instance $\langle R', w \rangle$ terminates iff the $\text{R[Rewr]}$ instance $\langle \Upsilon, y, y' \rangle$ was positive.

D.2 Lossy Channel Systems: Prop. [7]

We prove in this section Prop. [7] $(k + 2)$-LCS is $\mathcal{F}_{\omega^k}$-hard.

**Proof.** We reduce from a $(k+2)$-LR[Rat] instance $\langle R, w, w' \rangle$ and use Prop. [2]. Let $\$ be a fresh symbol and $T = \langle Q, \Sigma, \Sigma, \delta, I, F \rangle$ the normalized finite transducer for $R$.

We construct a LCS $C = \langle Q \uplus \{q_i, q_f\}, \Sigma \uplus \{\$\}, \delta' \rangle$ that cycles through its channel content: it starts with $w\$ as initial channel contents in some initial state of $T$, applies the transitions $(q, u, v, q')$ of $T$ by reading $u$ from the channel and writing $v$ through a transition $(q, ?uv, q')$ of $\delta'$, and cycles back upon reading $\$ by transitions $(q, \$\$, q')$ in $\delta'$ for all initial states $q' \in I$ and final states $q \in F$ of $T$. Adding to $\delta'$ the transitions $(q_i, \varepsilon, q)$ for $q$ in $I$ and $(q, \varepsilon, q_f)$ for $q$ in $F$, then $(w, w')$ belongs to $R^{\omega}_{\$}$ iff $q_i, w\$ $\rightarrow_{\$}^* q_f, w'\$ in $C$. As in the proof of Prop. [4], we can tighten this construction by reusing $\varepsilon$ for $\$.