TOCS manuscript No.
(will be inserted by the editor)

Generalized Post Embedding Problems

P. Karandikar · Ph. Schnoebelen

Received: date / Accepted: date

Abstract The Regular Post Embedding Problem extended with partial (co)directness is shown decidable. This extends to universal and/or counting versions. It is also shown that combining directness and codirectness in Post Embedding problems leads to undecidability.

1 Introduction

The *Regular Post Embedding Problem* (PEP for short, named by analogy with Post's Correspondence Problem, aka PCP) is the problem of deciding, given two morphisms on words $u, v : \Sigma^* \to \Gamma^*$ and a regular language $R \in \mathsf{Reg}(\Sigma)$, whether there is $\sigma \in R$ such that $u(\sigma)$ is a (scattered) subword of $v(\sigma)$. One then calls σ a *solution* of the PEP instance.

We use "$\sqsubseteq$" to denote the *subword* relation, also called *embedding*: $u(\sigma) \sqsubseteq v(\sigma) \stackrel{\text{def}}{\Leftrightarrow} u(\sigma)$ can be obtained by erasing some letters from $v(\sigma)$, possibly all of them, possibly none. Equivalently, PEP is the question whether a rational relation, or a transduction, $T \subseteq \Gamma^* \times \Gamma^*$ intersects non-vacuously the subword relation [5], hence it is a special case of the intersection problem for two rational relations.

This problem, introduced in [8], is new and quite remarkable: it is decidable but surprisingly hard since it is not primitive-recursive.[1] The problem is in fact

Supported by Grant ANR-11-BS02-001. The first author was partially supported by Tata Consultancy Services. An extended abstract of this article appeared in [18].

P. Karandikar
Chennai Mathematical Institute and LSV, ENS Cachan

Ph. Schnoebelen
LSV – CNRS & ENS Cachan

[1] But the problem becomes easy, decidable in linear-time and logarithmic space [8], when restricted to $R = \Sigma^+$ as in PCP.

$\mathbf{F}_{\omega^\omega}$-complete [17], that is, it sits at the first level above multiply-recursive in the Ordinal-Recursive Complexity Hierarchy [25].

A variant problem was introduced in [8]: $\mathsf{PEP}_{\mathrm{dir}}$ asks for the existence of a *direct* solution, i.e., some $\sigma \in R$ such that $u(\tau) \sqsubseteq v(\tau)$ for every prefix τ of σ. It turns out that PEP and $\mathsf{PEP}_{\mathrm{dir}}$ are inter-reducible (though not trivially) [10] and have the same complexity.

In this article we introduce $\mathsf{PEP}^{\mathrm{partial}}_{\mathrm{dir}}$, or "$\mathsf{PEP}$ with *partial* directness": Instead of requiring $u(\tau) \sqsubseteq v(\tau)$ for all prefixes of a solution (as in $\mathsf{PEP}_{\mathrm{dir}}$), or for none (as in PEP), $\mathsf{PEP}^{\mathrm{partial}}_{\mathrm{dir}}$ lets us select, by means of a regular language, which prefixes should verify the requirement. Thus $\mathsf{PEP}^{\mathrm{partial}}_{\mathrm{dir}}$ generalizes both PEP and $\mathsf{PEP}_{\mathrm{dir}}$.

Our main result is that $\mathsf{PEP}^{\mathrm{partial}}_{\mathrm{dir}}$ and the mirror problem $\mathsf{PEP}^{\mathrm{partial}}_{\mathrm{codir}}$ are decidable. The proof combines two ideas. Firstly, by Higman's Lemma, a long solution must eventually contain "*comparable*" so-called cutting points, from which one deduces that the solution is not minimal (or unique, or ...). Secondly, the above notion of "*eventually*", that comes from Higman's Lemma, can be turned into an effective upper bound thanks to a Length Function Theorem [26].

The decidability of $\mathsf{PEP}^{\mathrm{partial}}_{\mathrm{dir}}$ not only generalizes the decidability of PEP and $\mathsf{PEP}_{\mathrm{dir}}$: it is also simpler than the earlier proofs for PEP or $\mathsf{PEP}_{\mathrm{dir}}$, and it easily leads to an $\mathbf{F}_{\omega^\omega}$ complexity upper bound.

In a second part of the article, we extend our main result and show the decidability of universal and/or counting versions of the extended $\mathsf{PEP}^{\mathrm{partial}}_{\mathrm{dir}}$ problem. We also explain how our attempts at further generalisation, most notably by considering the combination of directness and codirectness in a same instance, lead to undecidable problems.

Applications to channel machines. Our interest in PEP and its variants comes from their close connection with fifo channel machines, a family of computational models that play a central role in some areas of program and system verification (see [7,1,22,4]) and that also provide decidable automata models for problems on Real-Time and Metric Temporal Logic, modal logics, data logics, etc. [2,24,21,19,23,5]. Here, PEP and its variants provide abstract versions of problems on channel machines, bringing greater clarity and versatility in both decidability and undecidability (more generally, hardness) proofs.

In this context, a further motivation for considering $\mathsf{PEP}^{\mathrm{partial}}_{\mathrm{dir}}$ is that it allows solving the decidability of UCSTs, i.e., unidirectional channel systems (with one reliable and one lossy channel) *extended with the possibility of testing the contents of channels* [16]. We recall that PEP was introduced for UCSs, unidirectional channel systems where tests on channels are not supported [10, 9], and that $\mathsf{PEP}_{\mathrm{dir}}$ corresponds to LCSs, i.e., lossy channel systems, for which decidability use techniques from WSTS theory [3,14,11,6]. Fig. 1 depicts the resulting situation.

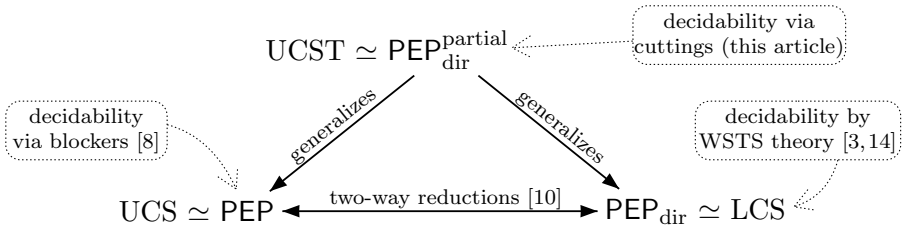

Fig. 1 Three decidability proofs for PEP and variants

Outline of the article. Section 2 recalls basic notations and definitions. In particular, it lists basic results about how the subword relation interacts with concatenations and factorization. Section 3 explains the Length Function Theorem for Higman's Lemma. Section 4 contains our main result, a direct decidability proof for $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$, a problem subsuming both PEP and $\mathsf{PEP}_{\text{dir}}$. Section 5 builds on this result and shows the decidability of counting problems on $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$. Section 6 further shows the decidability of universal variants of these questions. Section 7 contains undecidability results for some extensions of $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$.

2 Words and subwords

Words. Concatenation of words is denoted multiplicatively, with ε denoting the empty word. We write $|s|$ for the length of a word s, and $|\Gamma|$ for the size of a finite alphabet Γ. If s is a prefix of a word t, $s^{-1}t$ denotes the unique word s' such that $t = ss'$ (otherwise $s^{-1}t$ is not defined). Similarly, when s is a suffix of t, ts^{-1} is t with the s suffix removed. For a word $s = \mathtt{a}_0 \dots \mathtt{a}_{n-1}$, $\widetilde{s} \stackrel{\text{def}}{=} \mathtt{a}_{n-1} \dots \mathtt{a}_0$ is the mirrored word. The mirror of a language R is $\widetilde{R} \stackrel{\text{def}}{=} \{\widetilde{s} \mid s \in R\}$.

With a language $R \subseteq \Gamma^*$ one associates a congruence (wrt concatenation) given by $s \sim_R t \stackrel{\text{def}}{\Leftrightarrow} \forall x, y (xsy \in R \Leftrightarrow xty \in R)$ and called the Myhill congruence (also, the syntactic congruence). This equivalence has finite index if (and only if) R is regular. For regular R, let $\mu(R)$ denote this index: it satisfies $\mu\big(\widetilde{R}\big) = \mu(\Gamma^* \smallsetminus R) = \mu(R)$ and $\mu(R \cap R') \leq \mu(R)\,\mu(R')$. Also, $\mu(R)$ is computable from R, and in particular, $\mu(R) \leq m^m$ when R is recognized by a m-state complete DFA [15].

Subwords. We write $s \sqsubseteq t$ when s is a subword (subsequence) of t. Formally, $\mathtt{a}_0 \dots \mathtt{a}_{n-1} \sqsubseteq t$ iff t is some concatenation $t_0\mathtt{a}_0t_1\mathtt{a}_1 \dots \mathtt{a}_{n-1}t_n$. An *embedding* of $s = \mathtt{a}_0 \dots \mathtt{a}_{n-1}$ into $s' = \mathtt{a}'_0 \dots \mathtt{a}'_{m-1}$ is a strictly monotonic map $h : \{0, \dots, n-1\} \to \{0, \dots, m-1\}$ such that $\mathtt{a}_i = \mathtt{a}'_{h(i)}$ for all $0 \leq i < n$. Clearly, $s \sqsubseteq s'$ iff there exists an embedding of s into s'.

The subword relation is an ordering that is compatible with the monoid structure: $\varepsilon \sqsubseteq s$ for all words s, and $ss' \sqsubseteq tt'$ when $s \sqsubseteq t$ and $s' \sqsubseteq t'$.

Lemma 1 (Subwords and concatenation) *For all words y, z, s, t:*

(a) If $yz \sqsubseteq st$*, then* $y \sqsubseteq s$ *or* $z \sqsubseteq t$*.*
(b) If $yz \sqsubseteq st$ *and* $z \sqsubseteq t$ *and* x *is the longest suffix of* y *such that* $xz \sqsubseteq t$*, then* $yx^{-1} \sqsubseteq s$*.*
(c) If $yz \sqsubseteq st$ *and* $z \not\sqsubseteq t$ *and* x *is the shortest prefix of* z *such that* $x^{-1}z \sqsubseteq t$*, then* $yx \sqsubseteq s$*.*
(d) If $yz \sqsubseteq st$ *and* $z \sqsubseteq t$ *and* x *is the longest prefix of* t *such that* $z \sqsubseteq x^{-1}t$*, then* $y \sqsubseteq sx$*.*
(e) If $yz \sqsubseteq st$ *and* $z \not\sqsubseteq t$ *and* x *is the shortest suffix of* s *such that* $z \sqsubseteq xt$*, then* $y \sqsubseteq sx^{-1}$*.*
(f) If $sx \sqsubseteq yt$ *and* $t \sqsubseteq s$*, then* $sx^k \sqsubseteq y^k t$ *for all* $k \geq 1$*.*
(g) If $xs \sqsubseteq ty$ *and* $t \sqsubseteq s$*, then* $x^k s \sqsubseteq ty^k$ *for all* $k \geq 1$*.*

Proof Items (a–e) are easy (or see [13, Section 3]). Item (f) is proved by induction on k. The claim is true for $k = 1$, suppose it is true for $k = p$. Then $sx^{p+1} = sx^p x \sqsubseteq y^p tx \sqsubseteq y^p sx \sqsubseteq y^p yt = y^{p+1}t$. Item (g) is obtained from (f) by mirroring. □

3 Higman's Lemma and the length of bad sequences

It is well-known that for words over a finite alphabet, $\sqsubseteq$ is a well-quasi-ordering, that is, any infinite sequence of words $x_1, x_2, x_3, \ldots$ contains an infinite increasing subsequence $x_{i_1} \sqsubseteq x_{i_2} \sqsubseteq x_{i_3} \sqsubseteq \cdots$ [20]. This result is called Higman's Lemma.

For $n \in \mathbb{N}$, we say that a sequence (finite or infinite) of words is *n-good* if it contains an increasing subsequence of length n. It is *n-bad* otherwise. Higman's Lemma states that every infinite sequence is n-good for every n. Hence every n-bad sequence is finite.

It is often said that Higman's Lemma is "non-effective" or "non-constructive" since it does not come with any explicit information on the maximal length of bad sequences. Consequently, when one uses Higman's Lemma to prove that an algorithm terminates, no meaningful upper-bound on the algorithm's running time is derived from the proof. However, the length of bad sequences can be bounded if one takes into account the complexity of the sequences, or more precisely, of the process that generates bad sequences. The interested reader can consult [26,27] for more details. In this article we only use the simplest version of these results, i.e., the statement that when sequences only grow in a restricted way then the maximal length of bad sequences is computable, as we now explain.

For $k \in \mathbb{N}$, we say that a sequence of words $x_1, x_2, \ldots$ is *k-controlled* if $|x_i| \leq ik$ for all $i = 1, 2, \ldots$ Let $H(n, k, \Gamma)$ be the maximum length (if it exists) of an n-bad k-controlled sequence of words over a finite alphabet Γ.

Theorem 2 (Length Function Theorem) *H is a computable (total) function. Furthermore, H is monotonic in its three arguments.*

Proof Any prefix of a finite k-controlled n-bad sequence is k-controlled and n-bad. In particular, the empty sequence is. We arrange the set of all finite k-controlled n-bad sequences into a tree denoted $T_{n,k,\Gamma}$, or simply T, where the empty sequence is the root of T, and where a non-empty sequence of the form $x_1, \ldots, x_{l+1}$ is a child of its immediate prefix $x_1, \ldots, x_l$.

If T has an infinite path, this path is a chain of finite bad sequences linearly ordered by the prefix ordering and with which we can build an infinite k-controlled n-bad sequence by taking a limit. Thus T has no infinite paths since, by Higman's Lemma, Γ^* has no infinite bad sequences. Furthermore T is finitely branching, since the sequences it contains are k-controlled and Γ is finite. Thus, by Kőnig's Lemma, T is finite and $H(n, k, \Gamma)$ exists: it is the length of the longest sequence appearing in T, and also the length of T's longest path from the root.

H is computable since $T_{n,k,\Gamma}$ can be constructed effectively, starting from the root and listing the finitely many ways a current n-bad sequence can be extended in a k-controlled way. Finally, H is monotonic since, when $n' \leq n$ and $k' \leq k$, the n-bad k-controlled sequences over Γ include in particular all the n'-bad k'-controlled sequences over a subalphabet. □

Remark 3 Note that there is in general no maximum length of n-bad sequences over Γ if one does not restrict to k-controlled sequences. However, the proof of the Length Function Theorem can accommodate more liberal notions of controlled sequences, e.g., having $|x_i| \leq f(i)$ for all i, where f is a given computable function.

Note also that if $|\Gamma| = |\Gamma'|$ then $H(n, k, \Gamma) = H(n, k, \Gamma')$: only the number of different letters in Γ matters, and we sometimes write $H(n, k, p)$ for $H(n, k, \Gamma)$ where $p = |\Gamma|$. Upper bounds on $H(n, k, p)$ can be derived from the results given in [26] but these bounds are enormous, hard to express and hard to understand. In this article we content ourselves with the fact that H is computable. □

Below, we use the Length Function Theorem contrapositively: a k-controlled sequence of length greater than $H(n, k, \Gamma)$ is necessarily n-good, i.e., contains an increasing subsequence $x_{i_1} \sqsubseteq x_{i_2} \sqsubseteq \cdots \sqsubseteq x_{i_n}$ of length n.

4 Deciding $\mathsf{PEP}^{\mathbf{partial}}_{\mathbf{dir}}$, or PEP with partial directness

We introduce $\mathsf{PEP}^{\mathrm{partial}}_{\mathrm{dir}}$, a problem generalizing both PEP and $\mathsf{PEP}_{\mathrm{dir}}$, and show its decidability. This is proved by showing that if a $\mathsf{PEP}^{\mathrm{partial}}_{\mathrm{dir}}$ instance has a solution, then it has a solution whose length is bounded by a computable function of the input. This is simpler and more direct than the earlier decidability proof (for PEP only) based on blockers [8].

Definition 4 $\mathsf{PEP}^{\mathrm{partial}}_{\mathrm{dir}}$ *is the problem of deciding, given morphisms* $u, v : \Sigma^* \to \Gamma^*$ *and regular languages* $R, R' \in \mathsf{Reg}(\Sigma)$*, whether there is* $\sigma \in R$ *such that* $u(\sigma) \sqsubseteq v(\sigma)$ *and* $u(\tau) \sqsubseteq v(\tau)$ *for all prefixes* τ *of* σ *belonging to* R' *(in*

which case σ is called a solution*).*
$\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ *is the variant problem of deciding whether there is $\sigma \in R$ such that $u(\sigma) \sqsubseteq v(\sigma)$ and $u(\tau) \sqsubseteq v(\tau)$ for all suffixes τ of σ that belong to R'.*

Both PEP and $\mathsf{PEP}_{\text{dir}}$ are special cases of $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$, obtained by taking $R' = \varnothing$ and $R' = \Sigma^*$ respectively. Obviously $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ and $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ are two equivalent presentations, modulo mirroring, of a same problem. Given a $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ or $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ instance, we let $K_u \stackrel{\text{def}}{=} \max_{a\in\Sigma} |u(a)|$ denote the *expansion factor* of u and define

$$L \stackrel{\text{def}}{=} H(\mu(R)\,\mu(R') + 1, K_u, \Gamma)$$

(recall that $\mu(R)$ and $\mu(R')$ are the indexes of the Myhill congruences associated with R and R', while $H(n, k, \Gamma)$ is defined with the Length Function Theorem).

In this section we prove:

Theorem 5 *A $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ instance has a solution if, and only if, it has a solution of length at most $2L$.*
This entails that $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ is decidable.

Decidability is an obvious consequence since the length bound is computable, and since it is easy to check whether a candidate σ is a solution.

For the proof of Theorem 5, we consider an arbitrary $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ instance $(\Sigma, \Gamma, u, v, R, R')$ and a solution σ. Write $N = |\sigma|$ for its length, $\sigma[0, i)$ and $\sigma[i, N)$ for, respectively, its prefix of length i and its suffix of length $N - i$. Two indices $i, j \in [0, N]$ are *congruent* if $\sigma[i, N) \sim_R \sigma[j, N)$ and $\sigma[i, N) \sim_{R'} \sigma[j, N)$. When σ is fixed, as in the rest of this section, we use shorthand notations like $u_{0,i}$ and $v_{i,j}$ to denote the images, here $u(\sigma[0, i))$ and $v(\sigma[i, j))$, of factors of σ.

We prove two "cutting lemmas" giving sufficient conditions for "cutting" a solution $\sigma = \sigma[0, N)$ along certain indices $a < b$, yielding a shorter solution $\sigma' = \sigma[0, a)\sigma[b, N)$, i.e., σ with the factor $\sigma[a, b)$ cut out. Here the following notation is useful. We associate, with every suffix τ of σ', a corresponding suffix, denoted $S(\tau)$, of σ: if τ is a suffix of $\sigma[b, N)$, then $S(\tau) \stackrel{\text{def}}{=} \tau$, otherwise, $\tau = \sigma[i, a)\sigma[b, N)$ for some $i < a$ and we let $S(\tau) \stackrel{\text{def}}{=} \sigma[i, N)$. In particular $S(\sigma') = \sigma$.

An index $i \in [0, N]$ is said to be *blue* if $u_{i,N} \sqsubseteq v_{i,N}$, it is *red* otherwise. In particular, N is blue trivially, 0 is blue since σ is a solution, and i is blue whenever $\sigma[i, N) \in R'$. If i is a blue index, let $l_i \in \Gamma^*$ be the longest suffix of $u_{0,i}$ such that $l_i\, u_{i,N} \sqsubseteq v_{i,N}$ and call it the *left margin* at i.

Lemma 6 (Cutting lemma for blue indices) *Let $a < b$ be two congruent and blue indices. If $l_a \sqsubseteq l_b$, then $\sigma' = \sigma[0, a)\sigma[b, N)$ is a solution (shorter than σ).*

Proof Clearly $\sigma' \in R$ since $\sigma \in R$ and a and b are congruent. Also, for all suffixes τ of σ', $S(\tau) \in R'$ iff $\tau \in R'$.

We claim that, for any suffix τ of σ', if $u(S(\tau)) \sqsubseteq v(S(\tau))$ then $u(\tau) \sqsubseteq v(\tau)$. This is obvious when $\tau = S(\tau)$, so we assume $\tau \neq S(\tau)$, i.e., $\tau = \sigma[i,a)\sigma[b,N)$ and $S(\tau) = \sigma[i,N)$ for some $i < a$. Assume $u(S(\tau)) \sqsubseteq v(S(\tau))$, i.e., $u_{i,N} \sqsubseteq v_{i,N}$. Now both $u_{i,a}$ and l_a are suffixes of $u_{0,a}$, so that one is a suffix of the other, which gives two cases.

1. If $u_{i,a}$ is a suffix of l_a, then

$$\begin{aligned} u(\tau) = u_{i,a}\, u_{b,N} &\sqsubseteq l_a\, u_{b,N} && \text{since } u_{i,a} \text{ is a suffix of } l_a, \\ &\sqsubseteq l_b\, u_{b,N} && \text{since } l_a \sqsubseteq l_b \text{ by assumption}, \\ &\sqsubseteq v_{b,N} && \text{by definition of } l_b, \\ &\sqsubseteq v_{i,a}\, v_{b,N} = v(\tau)\,. \end{aligned}$$

2. Otherwise, $u_{i,a} = x\, l_a$ for some x, as illustrated in Fig. 2 where slanted arrows follow the rightmost embedding of $u(\sigma)$ into $v(\sigma)$. Here $u_{i,N} \sqsubseteq v_{i,N}$

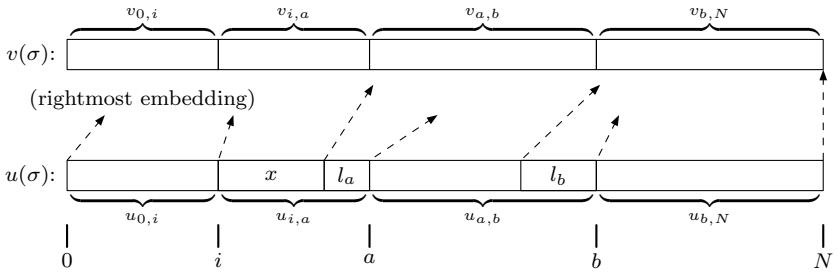

Fig. 2 Schematics for Lemma 6, with $l_a \sqsubseteq l_b$

rewrites as $x\, l_a\, u_{a,N} \sqsubseteq v_{i,a}\, v_{a,N}$. Now, and since l_a is (by definition) the longest suffix for which $l_a\, u_{a,N} \sqsubseteq v_{a,N}$, Lemma 1.b entails $x \sqsubseteq v_{i,a}$. Then

$$\begin{aligned} u(\tau) = u_{i,a}\, u_{b,N} &= x\, l_a\, u_{b,N} \\ &\sqsubseteq v_{i,a}\, l_b\, u_{b,N} && \text{since } x \sqsubseteq v_{i,a} \text{ and } l_a \sqsubseteq l_b, \\ &\sqsubseteq v_{i,a}\, v_{b,N} = v(\tau) && \text{by definition of } l_b. \end{aligned}$$

We can now infer $u(\tau) \sqsubseteq v(\tau)$ for any suffix $\tau \in R'$ (or for $\tau = \sigma'$) from the corresponding $u(S(\tau)) \sqsubseteq v(S(\tau))$. This shows that σ' is a solution. □

If i is a red index, i.e., if $u_{i,N} \not\sqsubseteq v_{i,N}$, let $r_i \in \Gamma^*$ be the shortest prefix of $u_{i,N}$ such that $r_i^{-1} u_{i,N} \sqsubseteq v_{i,N}$ (equivalently $u_{i,N} \sqsubseteq r_i\, v_{i,N}$) and call it the *right margin* at i.

Lemma 7 (Cutting lemma for red indices) *Let $a < b$ be two congruent and red indices. If $r_b \sqsubseteq r_a$, then $\sigma' = \sigma[0,a)\sigma[b,N)$ is a solution (shorter than σ).*

Proof Write x for $r_b^{-1} u_{b,N}$. Then $u_{b,N} = r_b\, x$ and $x \sqsubseteq v_{b,N}$. We proceed as for Lemma 6 and show that $u(S(\tau)) \sqsubseteq v(S(\tau))$ implies $u(\tau) \sqsubseteq v(\tau)$ for all suffixes

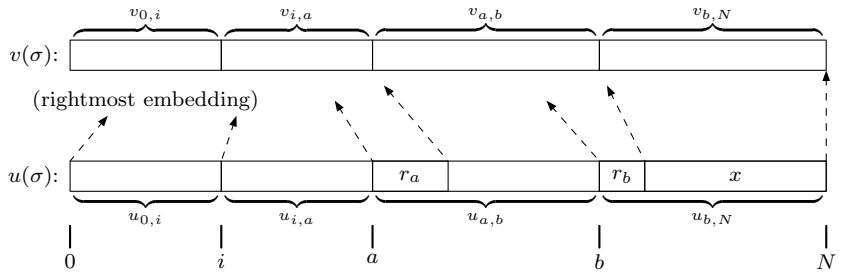

Fig. 3 Schematics for Lemma 7, with $r_b \sqsubseteq r_a$

τ of σ'. Assume $u(S(\tau)) \sqsubseteq v(S(\tau))$ for some τ. The only interesting case is when $\tau \neq S(\tau)$, i.e., when $\tau = \sigma[i,a)\sigma[b,N)$ for some $i < a$ (see Fig. 3).

From $u_{i,N} = u_{i,a}\,u_{a,N} \sqsubseteq v_{i,a}\,v_{a,N} = v_{i,N}$, i.e., $u(S(\tau)) \sqsubseteq v(S(\tau))$, and $u_{a,N} \not\sqsubseteq v_{a,N}$ (since a is a red index), Lemma 1.c entails $u_{i,a}\,r_a \sqsubseteq v_{i,a}$ by definition of r_a. Then

$$\begin{aligned} u(\tau) = u_{i,a}\,u_{b,N} = u_{i,a}\,r_b\,x &\sqsubseteq u_{i,a}\,r_a\,v_{b,N} && \text{since } r_b \sqsubseteq r_a \text{ and } x \sqsubseteq v_{b,N},\\ &\sqsubseteq v_{i,a}\,v_{b,N} = v(\tau) && \text{since } u_{i,a}\,r_a \sqsubseteq v_{i,a}. \end{aligned}$$ □

For the next step let $g_1 < g_2 < \cdots < g_{N_1}$ be all the blue indices in σ, and let $b_1 < b_2 < \cdots < b_{N_2}$ be the red indices. Observe that $N_1 + N_2 = N + 1$ since each index in $0, \ldots, N$ is either blue or red. We consider the corresponding sequences $(l_{g_i})_{i=1,\ldots,N_1}$ of left margins and $(r_{b_i})_{i=1,\ldots,N_2}$ of right margins.

Lemma 8 *$|l_{g_i}| \leq (i-1)\times K_u$ for all $i = 1, \ldots, N_1$, and $|r_{b_i}| \leq (N_2 - i + 1)\times K_u$ for all $i = 1, \ldots, N_2$. In other words, the sequence of left margins and the* reversed *sequence of right margins are K_u-controlled.*

Proof We prove that $|l_{g_i}| \leq (i-1) \times K_u$ by induction on i, showing $|l_{g_1}| = 0$ and $|l_{g_i}| - |l_{g_{i-1}}| \leq K_u$ for $i > 1$.

The base case $i = 1$ is easy: obviously $g_1 = 0$ since 0 is a blue index, and $l_0 = \varepsilon$ since it is the only suffix of $u_{0,0} = \varepsilon$, so that $|l_{g_1}| = 0$.

For the inductive step $i > 1$, write p for g_{i-1} and q for g_i. By definition, l_p is the longest suffix of $u_{0,p}$ with $l_p\,u_{p,N} = l_p\,u_{p,q}\,u_{q,N} \sqsubseteq v_{p,N}$. Since $l_q\,u_{q,N} \sqsubseteq v_{q,N} \sqsubseteq v_{p,N}$, l_q must be a suffix of $l_p\,u_{p,q}$, hence $|l_q| \leq |l_p| + |u_{p,q}| \leq |l_p| + K_u(q-p)$. This proves the claim in the case where $q = p+1$, i.e., when p and $p+1$ are blue.

There remains the case where $q > p+1$ and where all the indices from $p+1$ to $q-1$ are red. Thus in particular $u_{q-1,N} = u_{q-1,q}\,u_{q,N} \not\sqsubseteq v_{q-1,N}$. On the other hand q is blue and $l_q\,u_{q,N} \sqsubseteq v_{q,N} \sqsubseteq v_{q-1,N}$. We conclude that l_q must be a suffix of $u_{q-1,q}$, so that $|l_q| \leq K_u$ which proves the claim.

The reasoning for $|r_{b_i}|$ is similar:

If $b_{i+1} = b_i + 1$, then both b_i and the next index are red. Then r_{b_i} is a prefix of $u_{b_i,b_i+1}\,r_{b_{i+1}}$ so that $|r_{b_i}| \leq K_u + |r_{b_{i+1}}|$.

If $b_{i+1} > b_i + 1$, then $b_i + 1$ is blue and r_{b_i} is a prefix of u_{b_i,b_i+1} so that $|r_{b_i}| \leq K_u$.

For the base case, we have $b_{N_2} < N$ since N is blue. Hence $b_{N_2} + 1$ is blue and $|r_{b_{N_2}}| \leq K_u$ as above.

Finally, $|r_{b_i}| \leq (N_2 + 1 - i) \times K_u$ for all $i = 1, \ldots, N_2$. □

We are now ready to conclude the proof of Theorem 5. Let $N_c \stackrel{\text{def}}{=} \mu(R)\,\mu(R')+1$ and $L \stackrel{\text{def}}{=} H(N_c, K_u, \Gamma)$ and assume that $N > 2L$. Since $N_1 + N_2 = N + 1$, either σ has at least $L + 1$ blue indices and, by definition of L and H, there exist N_c blue indices $a_1 < a_2 < \cdots < a_{N_c}$ with $l_{a_1} \sqsubseteq l_{a_2} \sqsubseteq \cdots \sqsubseteq l_{a_{N_c}}$, or σ has at least $L + 1$ red indices and there exist N_c red indices $a'_1 < a'_2 < \cdots < a'_{N_c}$ with $r_{a'_{N_c}} \sqsubseteq \cdots \sqsubseteq r_{a'_2} \sqsubseteq r_{a'_1}$ (since it is the reversed sequence of right margins that is controlled). Out of $N_c = \mu(R)\,\mu(R')+1$ indices, two must be congruent, fulfilling the assumptions of either Lemma 6 or Lemma 7. Therefore σ can be cut to obtain a shorter solution.

Since $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ and $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ are equivalent problems modulo mirroring of R, u and v, we deduce that $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ too is decidable, and more precisely:

Corollary 9 *A* $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ *instance has a solution if, and only if, it has a solution of length at most* $2L$.

5 Counting the number of solutions

We consider two counting questions: $\exists^{\infty}\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ is the question whether a $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ instance has infinitely many solutions (a decision problem), while $\#\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ is the problem of computing the number of solutions of the instance (a number in $\mathbb{N} \cup \{\infty\}$). For technical convenience, we often deal with the (equivalent) codirected versions, $\exists^{\infty}\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ and $\#\mathsf{PEP}^{\text{partial}}_{\text{codir}}$.

For an instance $(\Sigma, \Gamma, u, v, R, R')$, we let $K_v \stackrel{\text{def}}{=} \max_{a\in\Sigma} |v(a)|$ and define

$$M \stackrel{\text{def}}{=} H(\mu(R)\,\mu(R') + 1, K_v, \Gamma)\,, \quad M' \stackrel{\text{def}}{=} H\big((2M+2)\mu(R)\,\mu(R') + 1, K_u, \Gamma\big)\,.$$

In this section we prove:

Theorem 10 *For a* $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ *or* $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ *instance, the following are equivalent:*

(a) it has infinitely many solutions;
(b) it has solution of length N *with* $2M < N$*;*
(c) it has a solution of length N *with* $2M < N \leq 2M'$.

This entails the decidability of $\exists^{\infty}\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ *and* $\exists^{\infty}\mathsf{PEP}^{\text{partial}}_{\text{codir}}$*, and the computability of* $\#\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ *and* $\#\mathsf{PEP}^{\text{partial}}_{\text{codir}}$.

As with Theorem 5, the length bounds $2M$ and $2M'$ are computable, so that $\exists^{\infty}\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ and $\exists^{\infty}\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ can be decided by finite enumeration. When the number of solutions is finite, counting them can also be done by finite enumeration since we know all solutions have then length at most $2M$.

For the proof of Theorem 10, we first observe that if the instance has a solution of length $N > 2M$, it has a solution with R replaced by $R^{>} \stackrel{\text{def}}{=} R \cap \Sigma^{2M+1}\Sigma^*$. The syntactic congruence associated with $R^{>}$ has index at most $(2M+2)\mu(R)$. From Theorem 5, we deduce that the modified instance has a solution of length at most $2M'$. Hence (b) and (c) are equivalent.

It remains to show that (b) implies (a) since obviously (a) implies (b). For this we fix an arbitrary $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ instance $(\Sigma, \Gamma, u, v, R, R')$ and consider a solution σ, of length N. We develop two so-called "iteration lemmas" that are similar to the cutting lemmas from Section 4, with the difference that they expand σ instead of reducing it.

As before, an index $i \in [0, N]$ is said to be *blue* if $u_{i,N} \sqsubseteq v_{i,N}$, and *red* otherwise. With a blue (resp., a red) index $i \in [0, N]$ we associate a word s_i (resp., t_i) in Γ^*. The s_i's and t_i's are analogous to the l_i's and r_i's from Section 4, however they are factors of $v(\sigma)$, not of $u(\sigma)$ like l_i or r_i, and this explains the difference between M and L. The terms "left margin" and "right margin" will be reused here for these factors.

We start with blue indices. For a blue index $i \in [0, N]$, let s_i be the longest prefix of $v_{i,N}$ such that $u_{i,N} \sqsubseteq s_i^{-1} v_{i,N}$ (equivalently, such that $s_i\, u_{i,N} \sqsubseteq v_{i,N}$) and call it the *right margin* at i.

Lemma 11 *Suppose $a < b$ are two blue indices with $s_b \sqsubseteq s_a$. Then for all $k \geq 1$, $s_a (u_{a,b})^k \sqsubseteq (v_{a,b})^k s_b$.*

Proof $s_a\, u_{a,N} \sqsubseteq v_{a,N}$ expands as $(s_a\, u_{a,b}) u_{b,N} \sqsubseteq v_{a,b}\, v_{b,N}$. Since b is blue, $u_{b,N} \sqsubseteq v_{b,N}$ and, by definition of s_b, Lemma 1.d further yields $s_a\, u_{a,b} \sqsubseteq v_{a,b}\, s_b$. One concludes with Lemma 1.f, using $s_b \sqsubseteq s_a$. □

Lemma 12 (Iteration lemma for blue indices) *Let $a < b$ be two congruent blue indices. If $s_b \sqsubseteq s_a$, then for every $k \geq 1$, $\sigma' = \sigma[0, a).\sigma[a, b)^k.\sigma[b, N)$ is a solution.*

Proof Let τ be any suffix of σ'. We show that $u(\tau) \sqsubseteq v(\tau)$ when $\tau \in R'$ or $\tau = \sigma'$, which will complete the proof. There are three cases, depending on how long τ is.

- τ is a suffix of $\sigma[a, N)$. Then τ is a suffix of σ itself, and this case is trivial since σ is a solution.
- τ is $\sigma[i, b)\sigma[a, b)^p\sigma[b, N)$ for some $p \geq 1$ and $a < i \leq b$. Since a and b are congruent, $\tau \in R'$ implies $\sigma[i, N) \in R'$. Thus $u_{i,N} \sqsubseteq v_{i,N}$, hence

$u_{i,b} \sqsubseteq v_{i,b}\, s_b$ (since $u_{b,N} \sqsubseteq v_{b,N}$).

$$\begin{aligned} u(\tau) &= u_{i,b}(u_{a,b})^p\, u_{b,N} \\ &\sqsubseteq v_{i,b}\, s_b(u_{a,b})^p\, u_{b,N} \\ &\sqsubseteq v_{i,b}\, s_a(u_{a,b})^p\, u_{b,N} && \text{since } s_b \sqsubseteq s_a \\ &\sqsubseteq v_{i,b}(v_{a,b})^p\, s_b\, u_{b,N} && \text{by Lemma 11} \\ &\sqsubseteq v_{i,b}(v_{a,b})^p\, v_{b,N} && \text{by definition of } s_b \\ &= v(\tau)\,. \end{aligned}$$

– τ is $\sigma[i,a)\sigma[a,b)^k\sigma[b,N)$ for some $0 \le i < a$. Since a and b are congruent, $\tau \in R'$ (or $\tau = \sigma$) implies $u_{i,N} \in R'$ (or $u_{i,N} = \sigma$) so that $u_{i,N} \sqsubseteq v_{i,N}$, from which we deduce $u_{i,a} \sqsubseteq v_{i,a}\, s_a$ as in the previous case. Then, using Lemma 11 and $s_b\, u_{b,N} \sqsubseteq v_{b,N}$, we get

$$\begin{aligned} u(\tau) &= u_{i,a}(u_{a,b})^k\, u_{b,N} \\ &\sqsubseteq v_{i,a}\, s_a(u_{a,b})^k\, u_{b,N} \\ &\sqsubseteq v_{i,a}(v_{a,b})^k\, s_b\, u_{b,N} && \text{by Lemma 11} \\ &\sqsubseteq v_{i,a}(v_{a,b})^k\, v_{b,N} && \text{by definition of } s_b \\ &= v(\tau)\,. \end{aligned}$$

□

Now to red indices. For a red index $i \in [0,N]$, let t_i be the shortest suffix of $v_{0,i}$ such that $u_{i,N} \sqsubseteq t_i\, v_{i,N}$. This is called the *left margin* at i. Thus, for a blue j such that $j < i$, $u_{j,N} \sqsubseteq v_{j,N}$ implies $u_{j,i}\, t_i \sqsubseteq v_{j,i}$ by Lemma 1.e.

Lemma 13 (Iteration lemma for red indices) *Let $a < b$ be two congruent red indices. If $t_a \sqsubseteq t_b$, then for every $k \ge 1$, $\sigma' = \sigma[0,a).\sigma[a,b)^k.\sigma[b,N)$ is a solution.*

Proof Let τ be any suffix of σ'. We show that $u(\tau) \sqsubseteq v(\tau)$ when $\tau \in R'$ or $\tau = \sigma'$, which will complete the proof. There are three cases, depending on how long τ is.

– τ is a suffix of $\sigma[a,N)$. Then τ is a suffix of σ itself, and this case is trivial since σ is a solution.
– τ is $\sigma[i,b)\sigma[a,b)^p\sigma[b,N)$ for some $p \ge 1$ and $a < i \le b$. Since a and b are congruent, $\tau \in R'$ implies $\sigma[i,N) \in R'$ and so $u_{i,N} \sqsubseteq v_{i,N}$. By definition of t_a, we have $u_{a,b}u_{b,N} \sqsubseteq (t_a v_{a,b})v_{b,N}$. Using Lemma 1.e and the definition of t_b we get $u_{a,b}\, t_b \sqsubseteq t_a\, v_{a,b}$, and then $(u_{a,b})^p\, t_b \sqsubseteq t_a(v_{a,b})^p$ with Lemma 1.g. Then

$$\begin{aligned} u(\tau) &= u_{i,b}(u_{a,b})^p\, u_{b,N} \\ &\sqsubseteq u_{i,b}(u_{a,b})^p\, t_b\, v_{b,N} && \text{by definition of } t_b \\ &\sqsubseteq u_{i,b}\, t_a(v_{a,b})^p\, v_{b,N} && \text{as above} \\ &\sqsubseteq u_{i,b}\, t_b(v_{a,b})^p\, v_{b,N} && \text{since } t_a \sqsubseteq t_b \\ &\sqsubseteq v_{i,b}(v_{a,b})^p\, v_{b,N} && \text{since } u_{i,N} \sqsubseteq v_{i,N},\ b \text{ is red, Lemma 1.e} \\ &= v(\tau)\,. \end{aligned}$$

– τ is $\sigma[i,a)\sigma[a,b)^k\sigma[b,N)$ for some $0 \leq i < a$ and $k \geq 1$. Since a and b are congruent, $\tau \in R'$ (or $\tau = \sigma$) implies $u_{i,N} \in R'$ (or $u_{i,N} = \sigma$) so that $u_{i,N} \sqsubseteq v_{i,N}$, from which we deduce $u_{i,a}\, t_a \sqsubseteq v_{i,a}$ as in the previous case. Then

$$\begin{aligned} u(\tau) &= u_{i,a}(u_{a,b})^k\, u_{b,N} \\ &\sqsubseteq u_{i,a}(u_{a,b})^k\, t_b\, v_{b,N} && \text{by definition of } t_b \\ &\sqsubseteq u_{i,a}\, t_a\, (v_{a,b})^k\, v_{b,N} && \text{as before} \\ &\sqsubseteq v_{i,a}(v_{a,b})^k\, v_{b,N} && \text{as above} \\ &= v(\tau)\,. \end{aligned}$$ □

We may now prove that the $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ instance has infinitely many solutions if it has solution of length $N > 2M$, i.e., that (b) implies (a) in Theorem 10.

Suppose there are N_1 blue indices in σ, say $g_1 < g_2 < \cdots < g_{N_1}$; and N_2 red indices, say $b_1 < b_2 < \cdots < b_{N_2}$.

Lemma 14 *$|s_{g_i}| \leq (N_1 - i + 1) \times K_v$ for all $i = 1, \ldots, N_1$, and $|t_{b_i}| \leq (i-1) \times K_v$ for all $i = 1, \ldots, N_2$. That is, the* reversed *sequence of right margins and the sequence of left margins are K_v-controlled.*

Proof We start with blue indices and right margins.

Lemma 15 *Suppose $a < b$ are two blue indices. Then s_a is a prefix of $v_{a,b}\, s_b$.*

Proof Both s_a and $v_{a,b}s_b$ are prefixes of $v_{a,N}$, hence one of them is a prefix of the other. Assume, by way of contradiction, that $v_{a,b}\, s_b$ is a proper prefix of s_a, say $s_a = v_{a,b}\, s_b\, x$ for some $x \neq \varepsilon$. Then $s_a\, u_{a,N} \sqsubseteq v_{a,N}$ rewrites as $v_{a,b}\, s_b\, x\, u_{a,N} \sqsubseteq v_{a,b}\, v_{b,N}$. Cancelling $v_{a,b}$ on both sides gives $s_b\, x\, u_{a,N} \sqsubseteq v_{b,N}$, i.e., $(s_b\, x\, u_{a,b})u_{b,N} \sqsubseteq v_{b,N}$, which contradicts the definition of s_b. □

We now show that $s_{g_{N_1}}, \ldots, s_{g_1}$ is K_v-controlled. N is a blue index, and $|s_N| = 0$. For $i \in [0, N)$, if both i and $i+1$ are blue indices, then by Lemma 15, $|s_i| \leq |s_{i+1}| + K_v$. If i is blue and $i+1$ is red, then it is easy to see that s_i is a prefix of $v(\sigma_i)$, and hence $|s_i| \leq K_v$. So we get that $s_{g_{N_1}}, \ldots, s_{g_1}$ is K_v-controlled.

Now to red indices and left margins. 0 is not a red index. For $i \in [0, N)$, if both i and $i+1$ are red, then it is easy to see that t_{i+1} is a suffix of $t_i\, v(\sigma_i)$, and so $|t_{i+1}| \leq |t_i| + K_v$. If i is blue and $i+1$ is red, then t_{i+1} is a suffix of $v(\sigma_i)$, and so $|t_{i+1}| \leq K_v$. So we get that $t_{b_1}, \ldots, t_{b_{N_2}}$ is K_v-controlled. □

Assume that σ is a long solution of length $N > 2M$. At least $M+1$ indices among $[0, N]$ are blue, or at least $M+1$ are red. We apply one of the two above claims, and from either $s_{g_{N_1}}, \ldots, s_{g_1}$ (if $N_1 > M$) or $t_{b_1}, \ldots, t_{b_{N_2}}$ (if $N_2 > M$) we get an increasing subsequence of length $\mu(R)\,\mu(R') + 1$. Among these there must be two congruent indices. Then we get infinitely many solutions by Lemma 12 or Lemma 13.

6 Universal variants of $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$

We consider universal variants of $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ (or rather $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ for the sake of uniformity). Formally, given instances $(\Sigma, \Gamma, u, v, R, R')$ as usual, $\forall\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ is the question whether *every* $\sigma \in R$ is a solution, i.e., satisfies both $u(\sigma) \sqsubseteq v(\sigma)$ and $u(\tau) \sqsubseteq v(\tau)$ for all suffixes τ that belong to R'. Similarly, $\forall^\infty\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ is the question whether "almost all", i.e., *all but finitely many*, σ in R are solutions, and $\#\neg\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ is the associated counting problem that asks how many $\sigma \in R$ are not solutions.

These universal questions can also be seen as Post *non*-embedding problems, asking whether there exists some $\sigma \in R$ such that $u(\sigma) \not\sqsubseteq v(\sigma)$? Introduced in [13] with $\forall\mathsf{PEP}$, they are significantly less challenging than the standard PEP problems, and decidability is easier to establish. For this reason, we just show in this article how $\forall\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ and $\forall^\infty\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ reduce to $\forall^\infty\mathsf{PEP}$ whose decidability was shown in [13]. The point is that partial codirectness constraints can be eliminated since universal quantifications commute with conjunctions (and since the codirectness constraint is universal itself).

Lemma 16 *$\forall\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ and $\forall^\infty\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ many-one reduce to $\forall^\infty\mathsf{PEP}$.*

Corollary 17 *$\forall\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ and $\forall^\infty\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ are decidable, $\#\neg\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ is computable.*

We now prove Lemma 16. First, $\forall\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ easily reduces to $\forall^\infty\mathsf{PEP}^{\text{partial}}_{\text{codir}}$: add an extra letter $\mathtt{z}$ to Σ with $u(\mathtt{z}) = v(\mathtt{z}) = \varepsilon$ and replace R and R' with $R.\mathtt{z}^*$ and $R'.\mathtt{z}^*$. Hence the second half of the lemma entails its first half by transitivity of reductions.

For reducing $\forall^\infty\mathsf{PEP}^{\text{partial}}_{\text{codir}}$, it is easier to start with the negation of our question:

$$\exists^\infty\sigma \in R : \big(u(\sigma) \not\sqsubseteq v(\sigma) \text{ or } \sigma \text{ has a suffix } \tau \text{ in } R' \text{ with } u(\tau) \not\sqsubseteq v(\tau)\big)\,. \qquad (*)$$

Call $\sigma \in R$ a *type 1 witness* if $u(\sigma) \not\sqsubseteq v(\sigma)$, and a *type 2 witness* if it has a suffix $\tau \in R'$ with $u(\tau) \not\sqsubseteq v(\tau)$. Statement $(*)$ holds if, and only if, there are infinitely many type 1 witnesses or infinitely many type 2 witnesses. The existence of infinitely many type 1 witnesses (call that "case 1") is the negation of a $\forall^\infty\mathsf{PEP}$ question. Now suppose that there are infinitely many type 2 witnesses, say $\sigma_1, \sigma_2, \ldots$ For each i, pick a suffix τ_i of σ_i such that $\tau_i \in R'$ and $u(\tau_i) \not\sqsubseteq v(\tau_i)$. The set $\{\tau_i \mid i = 1, 2, \ldots\}$ of these suffixes can be finite or infinite. If it is infinite ("case 2a"), then

$$u(\tau) \not\sqsubseteq v(\tau) \text{ for infinitely many } \tau \in (\overrightarrow{R} \cap R')\,, \qquad (**)$$

where $\overrightarrow{R}$ is short for $\overrightarrow{{}^{\geq 0}R}$ and for $k \in \mathbb{N}$, $\overrightarrow{{}^{\geq k}R} \stackrel{\text{def}}{=} \{y \mid \exists x : (|x| \geq k \text{ and } xy \in R)\}$ is the set of the suffixes of words from R one obtains by removing at least k letters. Observe that, conversely, $(**)$ implies the existence of infinitely many type 2 witnesses (for a proof, pick $\tau_1 \in \overrightarrow{R} \cap R'$ satisfying the above, choose

$\sigma_1 \in R$ of which τ_1 is a suffix. Then choose τ_2 such that $|\tau_2| > |\sigma_1|$, and proceed similarly).

On the other hand, if $\{\tau_i \mid i = 1, 2, \ldots\}$ is finite ("case 2b"), then there is a $\tau \in R'$ such that $u(\tau) \not\sqsubseteq v(\tau)$ and $\sigma'\tau \in R$ for infinitely many σ'. By a standard pumping argument, the second point is equivalent to the existence of some such σ' with also $|\sigma'| > k_R$, where k_R is the size of a NFA for R (taking $k_R = \mu(R)$ also works). Write now $\hat{R}$ for $\overrightarrow{>k_R R}$: if $\{\tau_i \mid i = 1, 2, \ldots\}$ is finite, then $u(\tau) \not\sqsubseteq v(\tau)$ for some τ in $(R' \cap \hat{R})$, and conversely this implies the existence of infinitely many type 2 witnesses.

To summarize, and since $\overrightarrow{R}$ and $\hat{R}$ are regular and effectively computable from R, we have just reduced $\forall^\infty\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ to the following conjunction

$$\begin{aligned} &\forall^\infty \sigma \in R : u(\sigma) \sqsubseteq v(\sigma) && \texttt{(not case 1)} \\ \bigwedge\ &\forall^\infty \tau \in (\overrightarrow{R} \cap R') : u(\tau) \sqsubseteq v(\tau) && \texttt{(not case 2a)} \\ \bigwedge\ &\forall \tau \in (\hat{R} \cap R') : u(\tau) \sqsubseteq v(\tau)\,. && \texttt{(not case 2b)} \end{aligned}$$

This is now reduced to a single $\forall^\infty\mathsf{PEP}$ instance by rewriting the $\forall\mathsf{PEP}$ into a $\forall^\infty\mathsf{PEP}$ (as explained in the beginning of this proof) and relying on a distributivity property of the form

$$\bigwedge_{i=1}^{n} \Big[\forall^\infty \sigma \in R_i : u(\sigma) \sqsubseteq v(\sigma)\Big] \;\equiv\; \forall^\infty \sigma \in \Big[\bigcup_{i=1}^{n} R_i\Big] : u(\sigma) \sqsubseteq v(\sigma)$$

to handle the resulting conjunction of 3 $\forall^\infty\mathsf{PEP}$ instances.

7 Undecidability for $\mathsf{PEP}_{\text{co\&dir}}$ and other extensions

The decidability of $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ is a non-trivial generalization of previous results for PEP. It is a natural question whether one can further generalize the idea of partial directness and maintain decidability. In this section we describe two attempts that lead to undecidability, even though they remain inside the regular PEP framework.[2]

Allowing non-regular R'. One direction for extending $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ is to allow *more expressive R' sets* for partial (co)directness. Let $\mathsf{PEP}^{\text{partial[DCFL]}}_{\text{codir}}$ and $\mathsf{PEP}^{\text{partial[Pres]}}_{\text{codir}}$ be like $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ except that R' can be any deterministic context-free $R' \in \mathsf{DCFL}(\Sigma)$ (resp., any Presburger-definable $R' \in \mathsf{Pres}(\Sigma)$, i.e., a language consisting of all words whose Parikh image lies in a given Presburger, or semilinear, subset of $\mathbb{N}^{|\Sigma|}$). Note that $R \in \mathsf{Reg}(\Sigma)$ is still required.

Theorem 18 (Undecidability) $\mathsf{PEP}^{\text{partial[DCFL]}}_{\text{codir}}$ *and* $\mathsf{PEP}^{\text{partial[Pres]}}_{\text{codir}}$ *are* Σ^0_1*-complete.*

[2] PEP is undecidable if we allow constraint sets R outside $\mathsf{Reg}(\Sigma)$ [8]. Other extensions, like $\exists x \in R_1 : \forall y \in R_2 : u(xy) \sqsubseteq v(xy)$, for $R_1, R_2 \in \mathsf{Reg}(\Sigma)$, have been shown undecidable [12].

Since both problems clearly are in Σ_1^0, one only has to prove hardness by reduction, e.g., from PCP, Post's Correspondence Problem. Let (Σ, Γ, u, v) be a PCP instance (where the question is whether there exists $x \in \Sigma^+$ such that $u(x) = v(x)$). Extend Σ and Γ with new symbols: $\Sigma' \stackrel{\text{def}}{=} \Sigma \cup \{1, 2\}$ and $\Gamma' \stackrel{\text{def}}{=} \Gamma \cup \{\#\}$. Now define $u', v' : \Sigma'^* \to \Gamma'^*$ by extending u, v on the new symbols with $u'(1) = v'(2) = \varepsilon$ and $u'(2) = v'(1) = \#$. Define now $R = 12\Sigma^+$ and $R' = \{\tau 2 \tau' \mid \tau, \tau' \in \Sigma^* \text{ and } |u(\tau\tau')| \neq |v(\tau\tau')|\}$. Note that R' is deterministic context-free and Presburger-definable.

Lemma 19 *The* PCP *instance* (Σ, Γ, u, v) *has a solution if and only if the* $\mathsf{PEP}_{\text{codir}}^{\text{partial[Pres]}}$ *and* $\mathsf{PEP}_{\text{codir}}^{\text{partial[DCFL]}}$ *instance* $(\Sigma', \Gamma', u', v', R, R')$ *has a solution.*

Proof Suppose σ is a solution to the PCP problem. Then $\sigma \neq \varepsilon$ and $u(\sigma) = v(\sigma)$. Now $\sigma' \stackrel{\text{def}}{=} 12\sigma$ is a solution to the partially codirected problem since $12\sigma \in R$, $u'(12\sigma) = \#u(\sigma) \sqsubseteq v'(12\sigma) = \#v(\sigma)$, and σ' has no suffix in R' (indeed $2\sigma \notin R'$ since $|u(\sigma)| = |v(\sigma)|$).

Conversely, suppose σ' is a solution to the partially codirected problem. Then $\sigma' = 12\sigma$ for some $\sigma \neq \varepsilon$. Since $u'(\sigma') = \#u(\sigma) \sqsubseteq v'(\sigma') = \#v(\sigma)$, we have $u(\sigma) \sqsubseteq v(\sigma)$. If $|u(\sigma)| \neq |v(\sigma)|$, then $2\sigma \in R'$, and so we must have $u'(2\sigma) = \#u(\sigma) \sqsubseteq v'(2\sigma) = v(\sigma)$. This is not possible as # does not occur in $v(\sigma)$. So $|u(\sigma)| = |v(\sigma)|$, and $u(\sigma) = v(\sigma)$. Thus σ is a solution to the PCP problem. □

Combining directness and codirectness. Another direction is to allow *combining* directness and codirectness constraints. Formally, $\mathsf{PEP}_{\text{co\&dir}}$ is the problem of deciding, given Σ, Γ, u, v, and $R \in \mathsf{Reg}(\Sigma)$ as usual, whether there exists $\sigma \in R$ such that $u(\tau) \sqsubseteq v(\tau)$ and $u(\tau') \sqsubseteq v(\tau')$ for all decompositions $\sigma = \tau.\tau'$. In other words, σ is both a direct and a codirect solution.

Note that $\mathsf{PEP}_{\text{co\&dir}}$ has no R' parameter (or, equivalently, has $R' = \Sigma^*$) and requires directness and codirectness at all positions. However, this restricted combination is already undecidable:

Theorem 20 (Undecidability) $\mathsf{PEP}_{\text{co\&dir}}$ *is* Σ_1^0*-complete.*

Membership in Σ_1^0 is clear and we prove hardness by reducing from the Reachability Problem for length-preserving semi-Thue systems.

A semi-Thue system $S = (\Upsilon, \Delta)$ has a finite set $\Delta \subseteq \Upsilon^* \times \Upsilon^*$ of string rewrite rules over some finite alphabet Υ, written $\Delta = \{l_1 \to r_1, \ldots, l_k \to r_k\}$. The one-step rewrite relation $\to_\Delta \subseteq \Upsilon^* \times \Upsilon^*$ is defined as usual with $x \to_\Delta y \stackrel{\text{def}}{\Leftrightarrow} x = zlz'$ and $y = zrz'$ for some rule $l \to r$ in Δ and strings z, z' in Υ^*. We write $x \stackrel{m}{\to}_\Delta y$ and $x \stackrel{*}{\to}_\Delta y$ when x can be rewritten into y by a sequence of m (respectively, any number, possibly zero) rewrite steps.

The *Reachability Problem* for semi-Thue systems is "Given $S = (\Upsilon, \Delta)$ and two regular languages $P_1, P_2 \in \mathsf{Reg}(\Upsilon)$, is there $x \in P_1$ and $y \in P_2$ s.t. $x \stackrel{*}{\to}_\Delta y$?". It is well-known (or easy to see by encoding Turing machines in semi-Thue systems) that this problem is undecidable (in fact, Σ_1^0-complete)

even when restricted to *length-preserving systems*, i.e., systems where $|l| = |r|$ for all rules $l \to r \in \Delta$.

We now construct a many-one reduction to $\mathsf{PEP}_{\text{co\&dir}}$. Let $S = (\Upsilon, \Delta)$, P_1, P_2 be a length-preserving instance of the Reachability Problem. W.l.o.g., we assume $\varepsilon \notin P_1$ and we restrict to reachability via an even and non-zero number of rewrite steps. With any such instance we associate a $\mathsf{PEP}_{\text{co\&dir}}$ instance $u, v : \Sigma^* \to \Gamma^*$ with $R \in \mathsf{Reg}(\Sigma)$ such that the following Correctness Property holds:

$$\begin{aligned} &\exists x \in P_1,\ \exists y \in P_2,\ \exists m \text{ s.t. } x \xrightarrow{m}_\Delta y \text{ (and } m > 0 \text{ is even)} \\ \text{iff }\ &\exists \sigma \in R \text{ s.t. } \sigma = \tau\tau' \text{ implies } u(\tau) \sqsubseteq v(\tau) \text{ and } u(\tau') \sqsubseteq v(\tau')\,. \end{aligned} \tag{CP}$$

The reduction uses letters like $\mathtt{a}$, $\mathtt{b}$ and $\mathtt{c}$ taken from Υ, and adds † as an extra letter. We use six copies of each such "plain" letter. These copies are obtained by priming and double-priming letters, and by overlining. Hence the six copies of $\mathtt{a}$ are $\mathtt{a}, \mathtt{a}', \mathtt{a}'', \overline{\mathtt{a}}, \overline{\mathtt{a}'}, \overline{\mathtt{a}''}$. As expected, for a "plain" word (or alphabet) x, we write x' and $\overline{x}$ to denote a version of x obtained by priming (respectively, overlining) all its letters. Formally, letting $\Upsilon_\dagger$ being short for $\Upsilon \cup \{\dagger\}$, one has $\Sigma = \Gamma \stackrel{\text{def}}{=} \Upsilon_\dagger \cup \Upsilon'_\dagger \cup \Upsilon''_\dagger \cup \overline{\Upsilon_\dagger} \cup \overline{\Upsilon'_\dagger} \cup \overline{\Upsilon''_\dagger}$.

We define and explain the reduction by running it on the following example:

$$\Upsilon = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\} \text{ and } \Delta = \{\mathtt{ab} \to \mathtt{bc},\ \mathtt{cc} \to \mathtt{aa}\}. \tag{S_{exmp}}$$

Assume that $\mathtt{abc} \in P_1$ and $\mathtt{baa} \in P_2$. Then $P_1 \xrightarrow{*}_\Delta P_2$ since $\mathtt{abc} \xrightarrow{*}_\Delta \mathtt{baa}$ as witnessed by the following (even-length) derivation $\pi =$ "$\mathtt{abc} \to_\Delta \mathtt{bcc} \to_\Delta \mathtt{baa}$". In our reduction, a rewrite step like "$\mathtt{abc} \to_\Delta \mathtt{bcc}$" appears in the PEP solution σ as the letter-by-letter interleaving $\mathtt{a}\overline{\mathtt{b}}\mathtt{b}\overline{\mathtt{c}}\mathtt{c}\overline{\mathtt{c}}$, denoted $\mathtt{abc} \,|||\, \mathtt{bcc}$, of a plain string and an overlined copy of a same-length string.

Write $T_{\blacktriangleright}(\Delta)$, or just $T_{\blacktriangleright}$ for short, for the set of all $x \,|||\, y$ such that $x \to_\Delta y$. Obviously, and since we are dealing with length-preserving systems, $T_{\blacktriangleright}$ is a regular language, as seen by writing it as $T_{\blacktriangleright} = \big(\sum_{a \in \Upsilon} a\overline{a}\big)^* . \big\{ l \,|||\, r \mid l \to r \in \Delta \big\} . \big(\sum_{a \in \Upsilon} a\overline{a}\big)^*$, where $\{l \,|||\, r \mid l \to r \in \Delta\}$ is a finite, hence regular, language.

$T_{\blacktriangleright}$ accounts for odd-numbered steps. Symmetrically, for even-numbered steps like $\mathtt{bcc} \to_\Delta \mathtt{baa}$ in π above, we use $\mathtt{b}\overline{\mathtt{b}}\mathtt{a}\overline{\mathtt{a}}\mathtt{c}\overline{\mathtt{c}}$, i.e., $\mathtt{baa} \,|||\, \mathtt{bcc}$. Here too $T_{\blacktriangleleft} \stackrel{\text{def}}{=} \{y \,|||\, x \mid x \to_\Delta y\}$ is regular. Finally, a derivation π of the general form

$$x_0 \to_\Delta x_1 \to_\Delta x_2 \ldots \to_\Delta x_{2k},$$

where $K \stackrel{\text{def}}{=} |x_0| = \ldots = |x_{2k}|$, is encoded as a solution σ_π of the form $\sigma_\pi = \rho_0\sigma_1\rho_1\sigma_2 \ldots \rho_{2k-1}\sigma_{2k}\rho_{2k}$ that alternates between the encodings of steps (the σ_i's) in $T_{\blacktriangleright} \cup T_{\blacktriangleleft}$, and *fillers*, (the ρ_i's) defined as follows:

$$\sigma_i \stackrel{\text{def}}{=} \begin{cases} x_{i-1} \,|||\, x_i \text{ for odd } i\,, \\ x_i \,|||\, x_{i-1} \text{ for even } i\,, \end{cases}$$

$$\rho_0 \stackrel{\text{def}}{=} x''_0 \,|||\, \dagger''^K\,, \qquad \rho_{2k} \stackrel{\text{def}}{=} x''_{2k} \,|||\, \dagger''^K\,, \qquad \rho_i \stackrel{\text{def}}{=} \begin{cases} \dagger'^K \,|||\, x'_i \text{ for odd } i\,, \\ x'_i \,|||\, \dagger'^K \text{ for even } i \neq 0, 2k\,. \end{cases}$$

Note that the extremal fillers ρ_0 and ρ_{2k} use double-primed letters, when the internal fillers use primed letters. Continuing our example, the σ_π associated with the derivation $\mathtt{abc}\to_\Delta\mathtt{bcc}\to_\Delta\mathtt{baa}$ is

$$\sigma_\pi = \underbrace{\mathtt{a}''\overline{\dagger''}\mathtt{b}''\overline{\dagger''}\mathtt{c}''\overline{\dagger''}}_{\mathtt{a}''\mathtt{b}''\mathtt{c}''\,|||\,\dagger''\dagger''\dagger''} \underbrace{\mathtt{a}\overline{\mathtt{b}}\mathtt{b}\overline{\mathtt{c}}\mathtt{c}\overline{\mathtt{c}}}_{\mathtt{abc}\,|||\,\mathtt{bcc}} \underbrace{\dagger'\overline{\mathtt{b}'}\dagger'\overline{\mathtt{c}'}\dagger'\overline{\mathtt{c}'}}_{\dagger'\dagger'\dagger'\,|||\,\mathtt{b}'\mathtt{c}'\mathtt{c}'} \underbrace{\mathtt{b}\overline{\mathtt{b}}\mathtt{a}\overline{\mathtt{c}}\mathtt{a}\overline{\mathtt{c}}}_{\mathtt{baa}\,|||\,\mathtt{bcc}} \underbrace{\mathtt{b}''\overline{\dagger''}\mathtt{a}''\overline{\dagger''}\mathtt{a}''\overline{\dagger''}}_{\mathtt{b}''\mathtt{a}''\mathtt{a}''\,|||\,\dagger''\dagger''\dagger''}\,.$$

The point with primed and double-primed copies is that u and v associate them with different images. Precisely, we define

$$\begin{array}{lllll} u(a) = a, & u(a') = \dagger, & u(\dagger') = \dagger, & u(a'') = \varepsilon, & u(\dagger'') = \varepsilon, \\ v(a) = \dagger, & v(a') = a, & v(\dagger') = w_\Upsilon, & v(a'') = a, & v(\dagger'') = w_\Upsilon, \end{array}$$

where a is any letter in Υ, and where w_Υ is a word listing all letters in Υ. E.g., $w_{\{\mathtt{a},\mathtt{b},\mathtt{c}\}} = \mathtt{abc}$ in our running example. The extremal fillers use special double-primed letters because we want $u(\rho_0) = u(\rho_{2k}) = \varepsilon$ (while v behaves the same on primed and double-primed letters). Finally, overlining is preserved by u and v: $u(\overline{x}) \stackrel{\text{def}}{=} \overline{u(x)}$ and $v(\overline{x}) \stackrel{\text{def}}{=} \overline{v(x)}$.

This ensures that, for $i > 0$, $u(\sigma_i) \sqsubseteq v(\rho_{i-1})$ and $u(\rho_i) \sqsubseteq v(\sigma_i)$, so that a σ_π constructed as above is a direct solution. It also ensures $u(\sigma_i) \sqsubseteq v(\rho_i)$ and $u(\rho_{i-1}) \sqsubseteq v(\sigma_i)$ for all $i > 0$, so that σ_π is also a codirect solution. One can check it on our running example by writing $u(\sigma_\pi)$ and $v(\sigma_\pi)$ alongside:

$$\sigma_\pi = \overbrace{\mathtt{a}''\overline{\dagger''}\mathtt{b}''\overline{\dagger''}\mathtt{c}''\overline{\dagger''}}^{\rho_0} \overbrace{\mathtt{a}\overline{\mathtt{b}}\mathtt{b}\overline{\mathtt{c}}\mathtt{c}\overline{\mathtt{c}}}^{\sigma_1} \overbrace{\dagger'\overline{\mathtt{b}'}\dagger'\overline{\mathtt{c}'}\dagger'\overline{\mathtt{c}'}}^{\rho_1} \overbrace{\mathtt{b}\overline{\mathtt{b}}\mathtt{a}\overline{\mathtt{c}}\mathtt{a}\overline{\mathtt{c}}}^{\sigma_2} \overbrace{\mathtt{b}''\overline{\dagger''}\mathtt{a}''\overline{\dagger''}\mathtt{a}''\overline{\dagger''}}^{\rho_2}$$

$$\begin{array}{lllllll} u(\sigma_\pi) = & & & \mathtt{a}\overline{\mathtt{b}}\mathtt{b}\overline{\mathtt{c}}\mathtt{c}\overline{\mathtt{c}} & \dagger\overline{\dagger}\dagger\overline{\dagger}\dagger\overline{\dagger} & \mathtt{b}\overline{\mathtt{b}}\mathtt{a}\overline{\mathtt{c}}\mathtt{a}\overline{\mathtt{c}} & \\ v(\sigma_\pi) = & \mathtt{a}\,\overline{\mathtt{abc}}\,\mathtt{b}\,\overline{\mathtt{abc}}\,\mathtt{c}\,\overline{\mathtt{abc}} & \dagger\overline{\dagger}\dagger\overline{\dagger}\dagger\overline{\dagger} & \mathtt{abc}\,\overline{\mathtt{b}}\,\mathtt{abc}\,\overline{\mathtt{c}}\,\mathtt{abc}\,\overline{\mathtt{c}} & \dagger\overline{\dagger}\dagger\overline{\dagger}\dagger\overline{\dagger} & \mathtt{b}\,\overline{\mathtt{abc}}\,\mathtt{a}\,\overline{\mathtt{abc}}\,\mathtt{a}\,\overline{\mathtt{abc}} & \end{array}$$

There remains to define R. Since $\rho_0 \in \left(\Upsilon''\overline{\dagger''}\right)^+$, since $\sigma_i \in T_\blacktriangleright$ for odd i, etc., we let

$$R \stackrel{\text{def}}{=} \left(\Upsilon''\overline{\dagger''}\right)^+ . T_\blacktriangleright^{\cap P_1} . \left(\dagger'\overline{\Upsilon'}\right)^+ . \left(T_\blacktriangleleft . \left(\Upsilon'\overline{\dagger'}\right)^+ . T_\blacktriangleright . \left(\dagger'\overline{\Upsilon'}\right)^+\right)^* . T_\blacktriangleleft^{\cap P_2} . \left(\Upsilon''\overline{\dagger''}\right)^+\,,$$

where $T_\blacktriangleright^{\cap P_1} \stackrel{\text{def}}{=} \{x \,|||\, y \mid x \to_\Delta y \wedge x \in P_1\} = T_\blacktriangleright \cap \{x \,|||\, y \mid x \in P_1 \wedge |x| = |y|\}$ is clearly regular when P_1 is, and similarly for $T_\blacktriangleleft^{\cap P_2} \stackrel{\text{def}}{=} \{y \,|||\, x \mid x \to_\Delta y \wedge y \in P_2\}$. Since $\sigma_\pi \in R$ when π is an even-length derivation from P_1 to P_2, we deduce that the left-to-right implication in (CP) holds.

We now prove the right-to-left implication, which concludes the proof of Theorem 20.

Assume that there is a $\sigma \in R$ such that $u(\tau) \sqsubseteq v(\tau)$ and $u(\tau') \sqsubseteq v(\tau')$ for all decompositions $\sigma = \tau\tau'$. By definition of R, σ must be of the form

$$\sigma = \rho_0\sigma_1\rho_1(\sigma_2\rho_2\sigma_3\rho_3)\dots(\dots\sigma_{2k-1}\rho_{2k-1})\sigma_{2k}\rho_{2k}$$

for some $k > 0$, with $\rho_0 \in \left(\Upsilon''\overline{\dagger''}\right)^+$, with $\sigma_i \in T_{\blacktriangleright}$ for odd i and $\sigma_i \in T_{\blacktriangleleft}$ for even i, etc. These $4k+1$ non-empty factors, $(\sigma_i)_{1\leq i\leq 2k}$ and $(\rho_i)_{0\leq i\leq 2k}$, are called the "*segments*" of σ, and numbered $s_0, \dots, s_{4k}$ in order.

Lemma 21 *$u(s_p) \sqsubseteq v(s_{p-1})$ and $u(s_{p-1}) \sqsubseteq v(s_p)$ for all $p = 1, \dots, 4k$.*

Proof First note that the definition of u and v ensures that $u(s_p)$ and $v(s_p)$ use disjoint alphabets. More precisely, all $u(\sigma_i)$'s and $v(\rho_i)$'s are in $(\Upsilon\overline{\Upsilon})^*$, while the $v(\sigma_i)$'s and the $u(\rho_i)$'s are in $(\dagger\overline{\dagger})^*$, with the special case that $u(\rho_0) = u(\rho_{2k}) = \varepsilon$ since ρ_0 and ρ_{2k} are made of double-primed letters.

Since σ is a direct solution, $u(s_0 \dots s_p) \sqsubseteq v(s_0 \dots s_p)$ for any p, and even

$$u(s_0 \dots s_p) \sqsubseteq v(s_0 \dots s_{p-1}), \tag{A_p}$$

since $v(s_p)$ has no letter in common with $u(s_p)$. We now claim that, for all $p = 1, \dots, 4k$

$$u(s_0 s_1 \dots s_p) \not\sqsupseteq v(s_0 s_1 \dots s_{p-2}), \tag{B_p}$$

as we prove by induction on p. For the base case, $p = 1$, the claim is just the obvious $u(s_0 s_1) \not\sqsupseteq \varepsilon$. For the inductive case $p > 1$, one combines $u(s_0 \dots s_{p-1}) \not\sqsupseteq v(s_0 \dots s_{p-3})$ (ind. hyp.) with $u(s_p) \not\sqsupseteq v(s_{p-2})$ (different alphabets) and gets $u(s_0 \dots s_p) \not\sqsupseteq v(s_0 \dots s_{p-2})$.

We now combine (A_p), i.e., $u(s_0 \dots s_p) \sqsubseteq v(s_0 \dots s_{p-1})$, and (B_{p-1}), i.e., $u(s_0 s_1 \dots s_{p-1}) \not\sqsupseteq v(s_0 s_1 \dots s_{p-3})$, yielding $u(s_p) \sqsubseteq v(s_{p-2}s_{p-1})$, hence $u(s_p) \sqsubseteq v(s_{p-1})$ since $u(s_p)$ and $v(s_{p-2})$ share no letter: we have proved one half of the Lemma. The other half is proved symmetrically, using the fact that σ is also a codirect solution. □

Lemma 22 $|s_1| = |s_2| = \dots = |s_{4k-1}|$.

Proof For any p with $0 < p < 4k$, $u(s_p) \sqsubseteq v(s_{p-1})$ (Lemma 21) implies $|s_p| \leq |s_{p-1}|$, as can easily be seen either when s_p is some $x \,|||\, y$ or when s_p is some filler like $\dagger'^L \,|||\, x'$. Thus $|s_0| \geq |s_1| \geq \dots \geq |s_{4k-1}|$. Similarly, the other half of Lemma 21, i.e., $u(s_{p-1}) \sqsubseteq v(s_p)$, entails $|s_1| \leq |s_2| \leq \dots \leq |s_{4k}|$. □

Now pick any $i \in \{1, \dots, 2k\}$. If i is odd, then by definition of R, $\sigma_i \in T_{\blacktriangleright}$ is some $x_{i-1} \,|||\, y_i$ with $x_{i-1} \to_\Delta y_i$ and $\sigma_{i+1} \in T_{\blacktriangleleft}$ is some $y_{i+1} \,|||\, x_i$ with $x_i \to_\Delta y_{i+1}$. Furthermore, ρ_i is some $\dagger'^{|z_i|} \,|||\, z_i'$. With Lemma 21, we deduce $y_i \sqsubseteq z_i$ and $x_i \sqsubseteq z_i$. With Lemma 22, we further deduce $|y_i| = |z_i| = |x_i|$, hence $y_i = x_i$. A similar reasoning shows that $y_i = x_i$ also holds when i is even, so that the steps $x_{i-1} \to_\Delta y_i$ can be chained. Finally, we deduce from σ the existence of a derivation $x_0 \to_\Delta x_1 \to_\Delta \dots \to_\Delta x_{2k}$. Since $\sigma_0 \in T_{\blacktriangleright}^{\cap P_1}$ and $\sigma_{2k} \in T_{\blacktriangleleft}^{\cap P_2}$, we further deduce $x_0 \in P_1$ and $x_{2k} \in P_2$. Hence the existence of σ entails $P_1 \xrightarrow{2k}_\Delta P_2$, which concludes the proof.

8 Concluding remarks

We introduced partial directness in Post Embedding Problems and proved the decidability of $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ and $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ by showing that an instance has a solution if, and only if, it has a solution of length bounded by a computable function of the input. (Furthermore, from Theorem 5, one may directly derive upper bounds on the complexity of $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ and $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ using the bounds on the Length Function H provided in [26].)

This generalizes and simplifies earlier proofs for PEP and $\mathsf{PEP}_{\text{dir}}$. The added generality is non-trivial and leads to decidability for UCST, or UCS (that is, unidirectional channel systems) extended with tests [16]. The simplification lets us deal smoothly with counting or universal versions of the problem. Finally, we showed that *combining* directness and codirectness constraints leads to undecidability.

References

1. Abdulla, P.A., Collomb-Annichini, A., Bouajjani, A., Jonsson, B.: Using forward reachability analysis for verification of lossy channel systems. Formal Methods in System Design **25**(1), 39–65 (2004). DOI 10.1023/B:FORM.0000033962.51898.1a
2. Abdulla, P.A., Deneux, J., Ouaknine, J., Worrell, J.: Decidability and complexity results for timed automata via channel machines. In: Proc. ICALP 2005, *Lecture Notes in Computer Science*, vol. 3580, pp. 1089–1101. Springer (2005). DOI 10.1007/11523468_88
3. Abdulla, P.A., Jonsson, B.: Verifying programs with unreliable channels. Information and Computation **127**(2), 91–101 (1996). DOI 10.1006/inco.1996.0053
4. Atig, M.F., Bouajjani, A., Touili, T.: On the reachability analysis of acyclic networks of pushdown systems. In: Springer (ed.) Proc. CONCUR 2008, *Lecture Notes in Computer Science*, vol. 5201, pp. 356–371. Springer (2008). DOI 10.1007/978-3-540-85361-9
5. Barceló, P., Figueira, D., Libkin, L.: Graph logics with rational relations. Logical Methods in Comp. Science **9**(3) (2013). DOI 10.2168/LMCS-9(3:1)2013
6. Bertrand, N., Schnoebelen, Ph.: Computable fixpoints in well-structured symbolic model checking. Formal Methods in System Design **43**(2), 233–267 (2013). DOI 10.1007/s10703-012-0168-y
7. Cécé, G., Finkel, A., Purushothaman Iyer, S.: Unreliable channels are easier to verify than perfect channels. Information and Computation **124**(1), 20–31 (1996). DOI 10.1006/inco.1996.0003
8. Chambart, P., Schnoebelen, Ph.: Post Embedding Problem is not primitive recursive, with applications to channel systems. In: Proc. FST&TCS 2007, *Lecture Notes in Computer Science*, vol. 4855, pp. 265–276. Springer (2007). DOI 10.1007/978-3-540-77050-3_22
9. Chambart, P., Schnoebelen, Ph.: Mixing lossy and perfect fifo channels. In: Proc. CONCUR 2008, *Lecture Notes in Computer Science*, vol. 5201, pp. 340–355. Springer (2008). DOI 10.1007/978-3-540-85361-9_28
10. Chambart, P., Schnoebelen, Ph.: The ω-Regular Post Embedding Problem. In: Proc. FOSSACS 2008, *Lecture Notes in Computer Science*, vol. 4962, pp. 97–111. Springer (2008). DOI 10.1007/978-3-540-78499-9_8
11. Chambart, P., Schnoebelen, Ph.: The ordinal recursive complexity of lossy channel systems. In: Proc. LICS 2008, pp. 205–216. IEEE Comp. Soc. Press (2008). DOI 10.1109/LICS.2008.47
12. Chambart, P., Schnoebelen, Ph.: Computing blocker sets for the Regular Post Embedding Problem. In: Proc. DLT 2010, *Lecture Notes in Computer Science*, vol. 6224, pp. 136–147. Springer (2010). DOI 10.1007/978-3-642-14455-4_14

13. Chambart, P., Schnoebelen, Ph.: Pumping and counting on the Regular Post Embedding Problem. In: Proc. ICALP 2010, *Lecture Notes in Computer Science*, vol. 6199, pp. 64–75. Springer (2010). DOI 10.1007/978-3-642-14162-1_6
14. Finkel, A., Schnoebelen, Ph.: Well-structured transition systems everywhere! Theoretical Computer Science **256**(1–2), 63–92 (2001). DOI 10.1016/S0304-3975(00)00102-X
15. Holzer, M., König, B.: On deterministic finite automata and syntactic monoid size. Theoretical Computer Science **327**(3), 319–347 (2004). DOI 10.1016/j.tcs.2004.04.010
16. Jančar, P., Karandikar, P., Schnoebelen, Ph.: Unidirectional channel systems can be tested. In: Proc. IFIP TCS 2012, *Lecture Notes in Computer Science*, vol. 7604, pp. 149–163. Springer (2012). DOI 10.1007/978-3-642-33475-7_11
17. Karandikar, P., Schmitz, S.: The parametric ordinal-recursive complexity of Post embedding problems. In: Proc. FOSSACS 2013, *Lecture Notes in Computer Science*, vol. 7794, pp. 273–288. Springer (2013). DOI 10.1007/978-3-642-37075-5_18
18. Karandikar, P., Schnoebelen, Ph.: Cutting through regular Post embedding problems. In: Proc. CSR 2012, *Lecture Notes in Computer Science*, vol. 7353, pp. 229–240. Springer (2012). DOI 10.1007/978-3-642-30642-6_22
19. Konev, B., Kontchakov, R., Wolter, F., Zakharyaschev, M.: Dynamic topological logics over spaces with continuous functions. In: Advances in Modal Logic, vol.6, pp. 299–318. College Publications (2006)
20. Kruskal, J.B.: The theory of well-quasi-ordering: A frequently discovered concept. Journal of Combinatorial Theory, Series A **13**(3), 297–305 (1972). DOI 10.1016/0097-3165(72)90063-5
21. Kurucz, A.: Combining modal logics. In: P. Blackburn, J.v. Benthem, F. Wolter (eds.) Handbook of Modal Logics, vol. 3, chap. 15, pp. 869–926. Elsevier Science (2006). DOI 10.1016/S1570-2464(07)80018-8
22. La Torre, S., Madhusudan, P., Parlato, G.: Context-bounded analysis of concurrent queue systems. In: Proc. TACAS 2008, *Lecture Notes in Computer Science*, vol. 4963, pp. 299–314. Springer (2008). DOI 10.1007/978-3-540-78800-3_21
23. Lasota, S., Walukiewicz, I.: Alternating timed automata. ACM Trans. Computational Logic **9**(2) (2008). DOI 10.1145/1342991.1342994
24. Ouaknine, J., Worrell, J.: On metric temporal logic and faulty Turing machines. In: Proc. FOSSACS 2006, *Lecture Notes in Computer Science*, vol. 3921, pp. 217–230. Springer (2006). DOI 10.1007/11690634_15
25. Schmitz, S.: Complexity hierarchies beyond elementary (2013). Preprint, `arxiv:1312.5686[cs.CC]`
26. Schmitz, S., Schnoebelen, Ph.: Multiply-recursive upper bounds with Higman's lemma. In: Proc. ICALP 2011, *Lecture Notes in Computer Science*, vol. 6756, pp. 441–452. Springer (2011). DOI 10.1007/978-3-642-22012-8_35
27. Schmitz, S., Schnoebelen, Ph.: Algorithmic aspects of WQO theory (2012). Lecture notes, `http://cel.archives-ouvertes.fr/cel-00727025`