The ω-Regular Post Embedding Problem*

P. Chambart and Ph. Schnoebelen

LSV, ENS Cachan, CNRS
61, av. Pdt. Wilson, F-94230 Cachan, France
email: {chambart|phs}@lsv.ens-cachan.fr

Abstract. Post's Embedding Problem is a new variant of Post's Correspondence Problem where words are compared with embedding rather than equality. It has been shown recently that adding regular constraints on the form of admissible solutions makes the problem highly non-trivial, and relevant to the study of lossy channel systems. Here we consider the infinitary version and its application to recurrent reachability in lossy channel systems.

1 Introduction

Post's correspondence problem, or shortly PCP, can be stated as the question whether two morphisms $u,v:\Sigma^*\to\Gamma^*$ agree non-trivially on some input, i.e., whether $u(\sigma)=v(\sigma)$ for some non-empty $\sigma\in\Sigma^+$. This undecidable problem plays a central role in computer science because it is very often easier and more natural to prove undecidability by reduction from PCP than from, say, the halting problem for Turing machines.

In a recent paper, we introduced PEP, the *Post Embedding Problem*, a variant of PCP where one asks whether $u(\sigma)$ is a (scattered) *subword* of $v(\sigma)$ for some σ [CS07]. The subword relation, also called embedding, is denoted "$\sqsubseteq$": $w\sqsubseteq w' \overset{\text{def}}{\Leftrightarrow} w$ can be obtained from w' by erasing some letters, possibly all of them, possibly none. We also introduced $\mathsf{PEP}^{\text{reg}}$, an extension of PEP where one adds the requirement that a solution σ belongs to a regular language $R\subseteq\Sigma^*$.

PEP is a trivial, hence not very interesting, problem. However, and quite surprisingly, $\mathsf{PEP}^{\text{reg}}$ behaves very differently. $\mathsf{PEP}^{\text{reg}}$ is decidable but it is not primitive recursive. In fact it is (non-trivially) equivalent to the reachability problem for lossy channel systems. Thus $\mathsf{PEP}^{\text{reg}}$ is a new representative of the strange computational niche that hosts lossy channel systems and other problems in timed automata and logics [LW05,ADOW05,OW06,OW07], concurrency models [AM02,Del07,LNO$^+$07], temporal and modal logic [DL06,GKWZ06,KWZ05,Kur06], and other areas [JL07]. We could also use $\mathsf{PEP}^{\text{reg}}$ to solve open problems on unidirectional channel systems combining one reliable and one lossy channel. These unidirectional systems, introduced in [CS07], are currently under our active scrutiny because of their fundamental role in the classification of channel systems that mix reliable and unreliable channels along arbitrary network topologies [Cha07].

* Work supported by the Agence Nationale de la Recherche, grant ANR-06-SETIN-001.

The ω*-regular Post Embedding Problem.* In this paper we consider infinitary extensions of $\mathsf{PEP}^{\text{reg}}$ [1], most prominently $\mathsf{PEP}^{\omega\text{-reg}}$, where one asks for an *infinite* $\sigma \in \Sigma^\omega$ such that $u(\sigma) \sqsubseteq v(\sigma)$, and where an ω-regular constraint can further be imposed upon σ. Our motivation is twofold. Firstly, we aim at deepening our understanding of PEP and $\mathsf{PEP}^{\text{reg}}$, two exciting new problems. Secondly, and based on the existing results for the finitary case, we expect that connections can be established between $\mathsf{PEP}^{\omega\text{-reg}}$ and recurrent reachability questions on channel systems.

Our contribution. In this paper, we show the equivalence between $\mathsf{PEP}^{\omega\text{-reg}}$ and recurrent reachability questions for unidirectional channel systems. This equivalence is shown using the *2-dimensional correspondence+embedding problem*, or $\mathsf{2PCEP}$, a new intermediary problem that leads to a clearer, more abstract and more modular approach. The approach handles both the finitary and the infinitary cases in a single way.

We also show that $\mathsf{PEP}^{\omega\text{-reg}}$ can be reduced to $\mathsf{PEP}^{\text{reg}}$, so that the two problems are equivalent. Hence $\mathsf{PEP}^{\omega\text{-reg}}$ is decidable. This has the surprising consequence that recurrent reachability for unidirectional channel systems is decidable. It further shows that the links we established between unidirectional channel systems and lossy channel systems (in [CS07]) do not carry over from reachability to recurrent reachability.

Finally, we show that recurrent reachability for lossy channel systems can be reduced to $\mathsf{PEP}_{\text{dir}}^{\omega\text{-reg}}$, the variant of $\mathsf{PEP}^{\omega\text{-reg}}$ where we look for *direct* solutions (informally, solutions where $v(\sigma)$ must be ahead of $u(\sigma)$ at all times when σ grows from ε to its final value). Hence $\mathsf{PEP}_{\text{dir}}^{\omega\text{-reg}}$ is undecidable (while $\mathsf{PEP}_{\text{codir}}^{\omega\text{-reg}}$ is decidable). Again, this contrasts with the finitary case, where $\mathsf{PEP}^{\text{reg}}$, $\mathsf{PEP}_{\text{dir}}^{\text{reg}}$ and $\mathsf{PEP}_{\text{codir}}^{\text{reg}}$ are equivalent.

Outline of the paper. Section 2 recalls the necessary definitions and notations on embeddings between finite or infinite words. Section 3 states the ω-regular Post embedding problem, solves it in the unconstrained case, and shows that restricting to short morphisms is no loss of generality. Section 4 shows the equivalence between $\mathsf{PEP}^{\text{reg}}$ and $\mathsf{PEP}^{\omega\text{-reg}}$, before Section 5 links $\mathsf{PEP}^{\omega\text{-reg}}$ and $\mathsf{PEP}^{\text{reg}}$ with reachability and recurrent reachability questions for unidirectional channel systems. Finally, section 6 solves the remaining case, $\mathsf{PEP}_{\text{dir}}^{\omega\text{-reg}}$, by linking it to recurrent reachability for lossy channel systems.

2 Notations and definitions

Words. We write $u, v, w, t, \sigma, \rho, \alpha, \beta, \ldots$ for words, i.e., finite or infinite (i.e., ω-length) sequences of letters such as $a, b, i, j, \ldots$ from alphabets $\Sigma, \Gamma, \ldots$. The *length* of $u \in \Sigma^* \cup \Sigma^\omega$ is written $|u|$, the set $alph(u)$ is the set of letters (a subset of Σ) that occur in u. We denote with $u.v$, or uv, the concatenation of u and v, with $uv = u$ when u has ω-length.

A *morphism* from Σ^* to Γ^* is a map $h : \Sigma^* \to \Gamma^*$ that respects the monoidal structure, i.e., with $h(\varepsilon) = \varepsilon$ and $h(\sigma.\rho) = h(\sigma).h(\rho)$. Its extension over Σ^ω is defined in the obvious way: note that, in general, it takes values in $\Gamma^* \cup \Gamma^\omega$ since $h(u) = \varepsilon$ for $u \neq \varepsilon$ is allowed. A morphism h is completely defined by its image $h(1)$, $h(2)$, ..., on

[1] Recall that the classic PCP problem is undecidable but r.e., while the infinitary extension, denoted PCP^ω, is Σ^1_1-complete.

$\Sigma = \{1,2,\ldots\}$. We often simply write $h_1, h_2, \ldots$, and h_σ, instead of $h(1), h(2), \ldots$, and $h(\sigma)$.

Embeddings. Given two words $u = a_1 \ldots a_n$ and $v = b_1 \ldots b_m$, we write $u \sqsubseteq v$ when u is a *subword* of v, i.e., when u can be obtained by erasing some letters (possibly none) from v. For example, $abba \sqsubseteq abracadabra$. Equivalently, $u \sqsubseteq v$ when u can be embedded in v, i.e., when there exists an order-preserving injective map (called an "*embedding*") $h : \{1,\ldots,n\} \to \{1,\ldots,m\}$ such that $a_i = b_{h(i)}$ for all $i = 1,\ldots,n$. Embeddings between ω-words are defined similarly, with a strictly increasing $h : \mathbb{N} \setminus 0 \to \mathbb{N} \setminus 0$. We explicitly allow the embedding of finite words into infinite ones.

It is well-known that the subword relation is a partial ordering on finite words. Observe that, between ω-words, embedding is only a (partial) *quasi*-ordering: $u \sqsubseteq v$ and $v \sqsubseteq u$ together do not imply $u = v$. For example, $(ab)^\omega \sqsubseteq (bba)^\omega \sqsubseteq (ab)^\omega$. We write $u \equiv v$ when $u \sqsubseteq v$ and $v \sqsubseteq u$.

Halving ω-words. For some $u \in \Sigma^\omega$, let $\mathit{inf}(u) \subseteq \Sigma$ denote the set of letters that occur infinitely many times in u. The word u can be decomposed under the form $u'.u''$ where u' is a finite prefix and the corresponding suffix $u'' \in \Sigma^\omega$, only contains letters from $\mathit{inf}(u)$. Such a decomposition is called a *halving* of u. There exists several (in fact, infinitely many) halvings of any $u \in \Sigma^\omega$: the *canonical halving* is obtained by selecting the shortest possible prefix u'.

The following lemma is a classic tool when considering embeddings between ω-words (see, e.g., [Fin85]).

Lemma 2.1. *Let $u, v \in \Sigma^\omega$ be two ω-words with $u'.u''$ and $v'.v''$ two arbitrary halvings of u and v. Then*

$$u \sqsubseteq v \text{ iff } \begin{cases} \mathit{alph}(u'') \subseteq \mathit{alph}(v''), \text{ and} \\ \text{there exists } x \in \mathit{alph}(v'')^* \text{ such that } u' \sqsubseteq v'x. \end{cases}$$

Furthermore, when $u \sqsubseteq v$, then x can be chosen with $|x| \leq |u'|$, and for any halving $u = u'.u''$ there exists a halving $v = v'.v''$ such that $u' \sqsubseteq v'$.

Corollary 2.2. *Let u_1, u_2 be two ω-words such that $\mathit{inf}(u_1) = \mathit{alph}(u_1) = \mathit{alph}(u_2) = \mathit{inf}(u_2)$. Then $u.u_1 \equiv u.u_2$ for all $u \in \Sigma^*$.*

3 Post embedding problems

Post embedding problems are variants of Post correspondence problems where correspondence (equality between words) is replaced by embedding, and where an additional regular constraint may be imposed over the solution.

Formally, given two morphisms $u, v : \Sigma^* \to \Gamma^*$ we say that $\sigma \in \Sigma^*$ is a *(finite) solution* to Post's embedding problem if $u_\sigma \sqsubseteq v_\sigma$. If $\sigma \in \Sigma^\omega$ and $u_\sigma \sqsubseteq v_\sigma$, then σ is an *infinite solution* (also called, an ω-solution).

We say that σ is a *direct* solution if $u_\rho \sqsubseteq v_\rho$ for every prefix ρ of σ. It is a *codirect solution* if $u_\rho \sqsubseteq v_\rho$ for every suffix ρ of σ. When considering finite solutions [CS07], there is a symmetry between the notions of direct and codirect solutions, since a direct solution for some u,v is a codirect solution for the mirror instance $\tilde{u},\tilde{v}$. This symmetry does not carry over to infinite solutions because the mirror of an ω-word is not an ω-word. Also, observe that the prefixes of a direct ω-solution are finite (direct) solutions, and that the suffixes of a codirect ω-solution are other infinite (codirect) solutions.

The Post embedding problems we considered in [CS07] are $\mathsf{PEP}^{\text{reg}}$, $\mathsf{PEP}^{\text{reg}}_{\text{dir}}$ and $\mathsf{PEP}^{\text{reg}}_{\text{codir}}$ that ask, given two morphisms u,v and a regular $R \subseteq \Sigma^*$, whether R contains a solution (respectively, a direct solution, a codirect solution). The infinitary extensions of these problems are $\mathsf{PEP}^{\omega\text{-reg}}$, $\mathsf{PEP}^{\omega\text{-reg}}_{\text{dir}}$ and $\mathsf{PEP}^{\omega\text{-reg}}_{\text{codir}}$, that ask, given u,v and an ω-regular $R \subseteq \Sigma^\omega$, whether there exists an ω-solution $\sigma \in R$ (resp., a direct ω-solution, a codirect ω-solution).

In the above definition, the regular constraint applies to σ but this is inessential and our results still hold when the constraint applies to u_σ, or v_σ, or both (see [CS07]).

For complexity issues, we assume that the constraint R is given as a nondeterministic automaton $\mathcal{A}_R$, that can be a FSA or a Büchi automaton depending on whether R is finitary or not. By a *reduction* between two decision problems, we mean a logspace many-one reduction, except when specified otherwise (as in Section 4). We say two problems are *equivalent* when they are inter-reducible.

3.1 General embedding for direct solutions

We now state a technical lemma that shows that the above definition of a direct solution, "$u_\rho \sqsubseteq v_\rho$ for all prefixes ρ of σ", can be replaced by a stronger requirement: that there exists an embedding of u_σ into v_σ that embeds any u_ρ into the corresponding v_ρ.

Let a PEP^ω instance be given by two morphisms u,v, and consider an infinite $\sigma \in \Sigma^\omega$, of the form $\sigma = i_1.i_2.i_3\ldots$

For $k = 0,1,2,\ldots$, we let l_k and l'_k denote respectively, the lengths $|u_{i_1 i_2 \ldots i_k}|$ and $|v_{i_1 i_2 \ldots i_k}|$.

Lemma 3.1. *The following are equivalent:*
(a). *σ is a* direct *solution,*
(b). *For all $k \in \mathbb{N}$, there exists an embedding $h_k : \{1,2,\ldots,l_k\} \to \{1,2,\ldots,l'_k\}$ that witnesses $u_{i_1 i_2 \ldots i_k} \sqsubseteq v_{i_1 i_2 \ldots i_k}$,*
(c). *There exists a general embedding $h : \mathbb{N} \to \mathbb{N}$ that witnesses $u_\sigma \sqsubseteq v_\sigma$ and such that its restriction to $\{1,2,\ldots,l_k\}$ witnesses $u_{i_1 i_2 \ldots i_k} \sqsubseteq v_{i_1 i_2 \ldots i_k}$.*

Proof (Sketch). (a) and (b) are equivalent by definition of being a direct solution. (c) obviously implies (b). We prove (c) from (b) by defining $h(i) \stackrel{\text{def}}{=} \min_{k=1,2,\ldots} h_k(i)$. □

3.2 The unrestricted problems

PEP and PEP^ω are the special case of $\mathsf{PEP}^{\text{reg}}$ and $\mathsf{PEP}^{\omega\text{-reg}}$ where $R = \Sigma^+$ (respectively, $R = \Sigma^\omega$), i.e., where there are no regularity constraints over the form of a solution. The remark that PEP is trivial extends to PEP^ω, $\mathsf{PEP}^\omega_{\text{dir}}$ and $\mathsf{PEP}^\omega_{\text{codir}}$:

Proposition 3.2. *Given two morphisms $u, v : \Sigma^* \to \Gamma^*$ defining a Post embedding problem:*
1. There is a solution in Σ^+ if and only if there is a direct ω-solution in Σ^ω if and only if there is some $i \in \Sigma$ such that $u_i \sqsubseteq v_i$.
2. There is an ω-solution in Σ^ω if and only if there is there is a codirect ω-solution if and only if there exists a non-empty subset Σ' of Σ s.t. $alph(u(\Sigma')) \subseteq alph(v(\Sigma'))$.

Proof. 1. Obviously, if $u_i \sqsubseteq v_i$ then $i \in \Sigma$ is a solution in Σ^+, and i^ω is a direct ω-solution. Conversely, if there is a direct solution $\sigma = i_1 i_2 i_3 \dots$ in Σ^ω, then $u_{i_1} \sqsubseteq v_{i_1}$ by definition of directness. If there is a finite solution $\sigma = i_1 i_2 i_3 \dots i_m$ in Σ^+, then either $u_{i_1} \sqsubseteq v_{i_1}$ and we are done, or $i_2 i_3 \dots i_m$ is a shorter finite solution, and we'll eventually encounter some $u_i \sqsubseteq v_i$.
2. Obviously, if $alph(u(\Sigma')) \subseteq alph(v(\Sigma'))$ for some non-empty $\Sigma' = \{i_1, \dots, i_m\}$, then $(i_1 \dots i_m)^\omega$ is an ω-solution, and even a codirect one. Conversely, given an ω-solution σ, Lemma 2.1 entails that, letting $\Sigma' \stackrel{\text{def}}{=} inf(\sigma)$, one has $alph(u(\Sigma')) \subseteq alph(v(\Sigma'))$. ☐

The corollary is:

Theorem 3.3. PEP^ω *and* $\mathsf{PEP}^\omega_{codir}$ *coincide, and are* PTime*-complete.*
$\mathsf{PEP}^\omega_{dir}$ *coincides with the finitary problems* PEP, PEP_{dir} *and* PEP_{codir}*, and these problems are in* $\mathsf{LogSpace}$.

Proof (Sketch). There exists a simple polynomial-time decision procedure for PEP^ω. It computes the largest Σ' satisfying $alph(u(\Sigma')) \subseteq alph(v(\Sigma'))$ and then checks that this Σ' is not empty. This largest Σ' is obtained by starting with $\Sigma' := \Sigma$ and then removing from Σ' every i for which $alph(u_i)$ is not included in the current Σ', until eventual stabilization (see Appendix B for PTime-hardness). Regarding $\mathsf{PEP}^\omega_{\text{dir}}$, one only needs deterministic logarithmic space to find whether $u_i \sqsubseteq v_i$ for some i. ☐

3.3 Short morphisms

$\mathsf{PEP}^{\text{reg}}_{\leq 1}$ (respectively $\mathsf{PEP}^{\omega\text{-reg}}_{\leq 1}$) is $\mathsf{PEP}^{\text{reg}}$ (respectively $\mathsf{PEP}^{\omega\text{-reg}}$) with the constraint that all images u_i's and v_i's have length ≤ 1, i.e., the morphisms can be seen as maps $u, v : \Sigma \to \Gamma \cup \{\varepsilon\}$.

Proposition 3.4.
1. PEP^{reg} *and* $\mathsf{PEP}^{reg}_{\leq 1}$ *are equivalent (inter-reducible).*
2. $\mathsf{PEP}^{\omega\text{-}reg}$ *and* $\mathsf{PEP}^{\omega\text{-}reg}_{\leq 1}$ *are equivalent (inter-reducible).*

Proof. It is enough to show that PEP reduces to $\mathsf{PEP}_{\leq 1}$. For this, let $u, v : \Sigma^* \to \Gamma^*$ be a PEP instance. Let $k > 0$ be large enough so that, for all $i \in \Sigma$, u_i and v_i have at most k letters. Then we can write each u_i under the form $u_i^1 \dots u_i^k$ with $u_i^j \in \Gamma \cup \{\varepsilon\}$, i.e., $|u_i^j| \leq 1$. Similarly, we write every v_i as some $v_i^1 \dots v_i^k$ with $|v_i^j| \leq 1$. We now define $\Sigma' \stackrel{\text{def}}{=} \Sigma \times \{1, \dots, k\}$ and two morphisms $u', v' : \Sigma'^* \to \Gamma^*$ with $u'(i, j) \stackrel{\text{def}}{=} u_i^j$ and $v'(i, j) \stackrel{\text{def}}{=} v_i^j$. Observe that u', v' defines a $\mathsf{PEP}_{\leq 1}$ instance. Now, with $R \subseteq \Sigma^*$ (or $R \subseteq \Sigma^\omega$) one associates a constraint $R' \subseteq \Sigma'^*$ (resp., $R' \subseteq \Sigma'^\omega$) by $R' \stackrel{\text{def}}{=} h(R)$ with $h : \Sigma^* \to \Sigma'^*$ given by $h(i) = (i, 1)(i, 2) \dots (i, k)$. R' is regular (resp., ω-regular) since R is, and u', v' admits a solution in R' iff u, v has one in R. ☐

4 Reducing $\mathsf{PEP}^{\omega\text{-reg}}$ to $\mathsf{PEP}^{\mathrm{reg}}$

Theorem 4.1 (Main result). *$\mathsf{PEP}^{\omega\text{-}reg}$ and PEP^{reg} are equivalent (modulo elementary reductions).*

Corollary 4.2. *$\mathsf{PEP}^{\omega\text{-}reg}$ is decidable (but not primitive-recursive).*

One direction of Theorem 4.1 is obvious: any $\mathsf{PEP}^{\mathrm{reg}}$ instance u, v, R can be seen as a $\mathsf{PEP}^{\omega\text{-reg}}$ instance by adding an extra symbol $\perp$ to Σ and Γ, replacing R with $R.\perp^\omega$, and letting $u(\perp) = v(\perp) = \perp$.

For the other direction, we consider a $\mathsf{PEP}^{\omega\text{-reg}}$ instance given by two morphisms $u, v : \Sigma^* \to \Gamma^*$ and an ω-regular $R \subseteq \Sigma^\omega$.

Lemma 4.3. *There exists $\sigma \in R$ such that $u_\sigma \sqsubseteq v_\sigma$ if and only if there exists two finite words ρ_1 and ρ_2 in Σ^* such that*

(a) $\rho_1.\rho_2^\omega \in R$,
(b) $u_{\rho_1} \sqsubseteq v_{\rho_1.\rho_2}$, *and*
(c) $alph(u_{\rho_2}) \subseteq alph(v_{\rho_2})$.

Proof. The "$\Leftarrow$" direction is easy since taking $\sigma = \rho_1.\rho_2^\omega$ is sufficient. For the "$\Rightarrow$" direction, we assume that $\sigma = a_1a_2a_3 \ldots \in R$ satisfies $u_\sigma \sqsubseteq v_\sigma$ and show how to build ρ_1 and ρ_2.

Let $\mathcal{A}_R = (Q, \Sigma, q_0, F, \delta)$ be a Büchi automaton for R, and $\pi = q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \xrightarrow{a_3} \cdots$ be an accepting run of $\mathcal{A}_R$ over σ. This run is an ω-sequence of transitions "$q_{i-1} \xrightarrow{a_i} q_i$", so that $\pi \in \delta^\omega$ can be halved under the form $\pi = \pi'.\pi''$. This gives rise to two halvings $u'.u''$ and $v'.v''$ of, respectively, u_σ and v_σ.

Let us pick a finite prefix θ of π'' that uses every transition from $inf(\pi)$ at least once, and that ends on the starting state of π''. Hence θ is some $q_n \xrightarrow{a_{n+1}} q_{n+1} \xrightarrow{a_{n+2}} \cdots \xrightarrow{a_{n+k}} q_{n+k}$ with $n = |\pi'|$, $q_n = q_{n+k}$, and $inf(\sigma) = \{a_{n+1}, a_{n+2}, \ldots, a_{n+k}\}$. Let now $\rho_1 \stackrel{\text{def}}{=} a_1a_2 \ldots a_n$ and $\rho \stackrel{\text{def}}{=} a_{n+1}a_{n+2} \ldots a_{n+k}$. Clearly $\rho_1.\rho^\omega \in R$ as witnessed by the ultimately periodic run $\pi'.\theta^\omega$. Furthermore, from $u' = u_{\rho_1}$ and $inf(u'') = alph(u'') = alph(u_\rho)$, we deduce $u_\sigma = u'.u'' \equiv u_{\rho_1.\rho^\omega}$ using Corollary 2.2. Similarly, $v_\sigma \equiv v_{\rho_1.\rho^\omega}$. Hence $u_\sigma \sqsubseteq v_\sigma$ entails $u_{\rho_1.\rho^\omega} \sqsubseteq v_{\rho_1.\rho^\omega}$. Using Lemma 2.1, we conclude that $u_{\rho_1} \sqsubseteq v_{\rho_1.\rho_2}$ can be obtained by picking for ρ_2 a large enough power $\rho_2 \stackrel{\text{def}}{=} \rho.\rho \ldots \rho$ of ρ. Such a ρ_2 further ensures $\rho_2^\omega = \rho^\omega$, so that requirements (a) and (c) are inherited from ρ. □

For the next step, we show how to state the existence of two finite ρ_1 and ρ_2 as in Lemma 4.3 under the form of a $\mathsf{PEP}^{\mathrm{reg}}$ problem.

Let $\mathcal{A}_R = (Q, \Sigma, q_0, F, \delta)$ be the Büchi automaton defining R. As is standard, for $q, q' \in Q$, we let $L_{q,q'} \subseteq \Sigma^*$ denote the (regular) language accepted by starting $\mathcal{A}_R$ in q and stopping in q'.

Let $\Sigma' = \{1', 2', \ldots\}$ be a copy of $\Sigma = \{1, 2, \ldots\}$ where letters have been primed: for $x \in \Sigma^*$ and $L \subseteq \Sigma^*$, we let $x' \in \Sigma'^*$ and $L' \subseteq \Sigma'^*$ denote primed versions of x and L.

We can now express condition (a) as a regularity constraint on $\rho_1.\rho_2'$: by definition, $\rho_1.\rho_2^\omega$ belongs to R iff for some $q \in Q$, $\rho_1 \in L_{q_0,q}$ and $\rho_2 \in (L_{q,q} \smallsetminus \varepsilon)$. That is, if and only if $\rho_1.\rho_2' \in R_1$ with

$$R_1 \stackrel{\text{def}}{=} \bigcup_{q \in Q} L_{q_0,q}.(L'_{q,q} \smallsetminus \varepsilon).$$

Condition (b) can be stated as an embedding property on $\rho_1.\rho_2'$: let $u', v' : (\Sigma \cup \Sigma')^* \to \Gamma^*$ be the extensions of u and v given by $u'_{i'} \stackrel{\text{def}}{=} \varepsilon$ and $v'_{i'} \stackrel{\text{def}}{=} v_i$. Then

$$u_{\rho_1} \sqsubseteq v_{\rho_1.\rho_2} \text{ if and only if } u'_{\rho_1.\rho_2'} \sqsubseteq v'_{\rho_1.\rho_2'}.$$

Finally, condition (c) can be expressed as another regularity constraint. Indeed, for $X \subseteq \Gamma$, $alph(u_{\rho_2}) \subseteq X$ and $alph(v_{\rho_2}) \subseteq X$ require $\rho_2 \in u^{-1}(X^*)$ and, respectively, $\rho_2 \in v^{-1}(X^*)$. These are regular conditions on ρ_2 since inverse morphisms preserve regularity. Let now

$$R_2 \stackrel{\text{def}}{=} \bigcup_{X \subseteq \Gamma} \Big(u^{-1}(X^*) \cap v^{-1}(X^*) \cap \bigcap_{a \in X} \overbrace{\Sigma^* \{ i \in \Sigma \mid a \in alph(v_i) \} \Sigma^*}^{a \in alph(v_{\rho_2})} \Big).$$

Clearly, $alph(u_{\rho_2}) \subseteq alph(v_{\rho_2})$ if and only if $\rho_2 \in R_2$. Hence $alph(u_{\rho_2}) \subseteq alph(v_{\rho_2})$ if, and only if, $\rho_1.\rho_2' \in \Sigma^*.(R_2)'$ where we observe that R_2, hence $\Sigma^*.(R_2)'$ too, are regular.

Finally, u, v has an ω-solution in R iff u', v' has a finite solution in $R_1 \cap (R_2)'$, which provides the reduction from $\mathsf{PEP}^{\omega\text{-reg}}$ to $\mathsf{PEP}^{\text{reg}}$.

Remark 4.4. The automaton for R_1 has size linear in $|\mathcal{A}_R|$. The automaton for R_2 has size exponential in $|\Sigma|$: this is because we consider all subsets $X \subseteq \Sigma$. Hence the reduction from $\mathsf{PEP}^{\omega\text{-reg}}$ to $\mathsf{PEP}^{\text{reg}}$ is not logspace when the constraint R is given by a non-deterministic FSA. It is polynomial-space, which is certainly fine enough to state "equivalence" by inter-reducibility between problems that are not primitive-recursive.

There exists other possible choices for the precise finitary way with which R is supposed to be provided in a PEP instance: for many of these choices, from various logical formalisms (e.g., MSO) to various automata-based framework (e.g., alternating automata), logspace reductions from $\mathsf{PEP}^{\omega\text{-reg}}$ to $\mathsf{PEP}^{\text{reg}}$ exist. $\square$

We conclude this section with the following observation:

Theorem 4.5. $\mathsf{PEP}^{\omega\text{-}reg}_{codir}$ *and* $\mathsf{PEP}^{reg}_{codir}$ *are equivalent (inter-reducible).*

This can be proved using the same techniques we used in this section, in particular one can state a version of Lemma 4.3 that accounts for codirect solutions (while this is not possible for direct solutions). Then a *codirect* infinite solution σ induces the existence of a *codirect* $\rho_1.\rho_2^\omega$, and the existence of such an infinite $\rho_1.\rho_2^\omega$ can be witnessed by a finite $\rho_1.\rho_2'$ that solves a derived $\mathsf{PEP}^{\text{reg}}_{\text{codir}}$ instance.

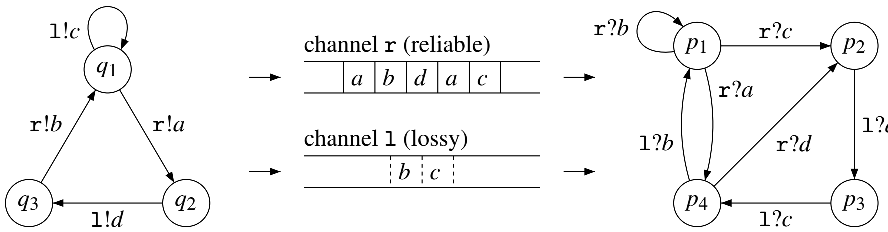

Fig. 1. A unidirectional channel system with one r eliable and one l ossy channel

5 Unidirectional channel systems

Unidirectional channel systems, shortly UCS, are systems composed of two finite-state machines that communicate *unidirectionally* via one reliable and one lossy channel, as illustrated in Fig. 1. No feedback communication from the receiver to the sender is possible. UCS's are a key ingredient in the complete classification of mixed channel systems according to their network topologies [Cha07].

Formally, a UCS has the form $S = (Q_1, Q_2, \mathtt{M}, \{\mathtt{r}, \mathtt{l}\}, \Delta_1, \Delta_2)$, where Q_1 and Δ_1 (respectively, Q_2 and Δ_2) are the finite set of states and set of rules of the sender (respectively, the receiver), $\mathtt{M}$ is the finite message alphabet, $\mathtt{r}$ and $\mathtt{l}$ are the names of, respectively, the reliable and the lossy channel. The sender's rules, Δ_1, is a subset of $Q_1 \times \{\mathtt{r}, \mathtt{l}\} \times \{!\} \times \mathtt{M}^* \times Q_1$, i.e., it contains rules of the form $q \xrightarrow{\mathtt{r}!u} q'$ or $q \xrightarrow{\mathtt{l}!u} q'$. The receiver's rules have the form $q \xrightarrow{\mathtt{r}?u} q'$ or $q \xrightarrow{\mathtt{l}?u} q'$ with $q, q' \in Q_2$.

A configuration of S is a tuple $\langle q_1, q_2, v_1, v_2 \rangle$ with control states q_1 and q_2 for the components, contents v_1 for channel $\mathtt{r}$, and v_2 for $\mathtt{l}$. The operational semantics is as expected. A rule $q \xrightarrow{\mathtt{r}!u} q'$ (resp. $q \xrightarrow{\mathtt{l}!u} q'$) from Δ_1 gives rise to all transitions $\langle q, q_2, v_1, v_2 \rangle \to \langle q', q_2, v_1 u, v_2 \rangle$ (resp. all $\langle q, q_2, v_1, v_2 \rangle \to \langle q', q_2, v_1, v_2 u' \rangle$ for $u' \sqsubseteq u$). A rule $q \xrightarrow{\mathtt{r}?u} q'$ (resp. $q \xrightarrow{\mathtt{l}?u} q'$) from Δ_2 gives rise to all transitions $\langle q_1, q, uv_1, v_2 \rangle \to \langle q_1, q', v_1, v_2 \rangle$ (resp. all $\langle q_1, q, v_1, uv_2 \rangle \to \langle q_1, q', v_1, v_2 \rangle$). Observe that message losses only occur when writing to channel $\mathtt{l}$. A run π is a sequence

$$\pi : \langle q_1^0, q_2^0, v_1^0, v_2^0 \rangle \to \langle q_1^1, q_2^1, v_1^1, v_2^1 \rangle \to \langle q_1^2, q_2^2, v_1^2, v_2^2 \rangle \to \cdots$$

of configurations linked by valid transitions.

We consider reachability and recurrent reachability problems for UCS's. Formally, given a UCS S, two initial states $q^1_{\mathrm{init}} \in Q_1$ and $q^2_{\mathrm{init}} \in Q_2$, two sets $F_1 \subseteq Q_1$ and $F_2 \subseteq Q_2$ of final states, the *reachability problem*, denoted $\mathsf{ReachUcs}$, asks whether there exists a run that starts from configuration $\langle q^1_{\mathrm{init}}, q^2_{\mathrm{init}}, \varepsilon, \varepsilon \rangle$ and ends in some configuration $\langle q^1_{\mathrm{final}}, q^2_{\mathrm{final}}, \varepsilon, \varepsilon \rangle$ with $(q^1_{\mathrm{final}}, q^2_{\mathrm{final}}) \in F_1 \times F_2$. The *recurrent reachability problem*, denoted $\mathsf{RecReachUcs}$, asks whether there exists an infinite run starting from $\langle q^1_{\mathrm{init}}, q^2_{\mathrm{init}}, \varepsilon, \varepsilon \rangle$ and visiting infinitely many configurations $\langle q_1^i, q_2^i, v_1^i, v_2^i \rangle$ with $(q_1^i, q_2^i) \in F_1 \times F_2$.

Remark 5.1. As explained in [CS07], requiring that our reachability questions have empty channels in the initial and the target configurations is just a technical simplification. More general reachability questions, including *control-state reachability*, where the channels contents in the target configuration are existentially quantified upon, reduce easily to $\mathsf{ReachUcs}$. $\square$

Theorem 5.2 (Equivalence between UCS and Post Embedding).
1. PEP^{reg} *and* $\mathsf{ReachUcs}$ *are equivalent (inter-reducible).*
2. $\mathsf{PEP}^{\omega\text{-}reg}$ *and* $\mathsf{RecReachUcs}$ *are equivalent (inter-reducible).*

The finitary case was first stated and proved in [CS07]. In the rest of this section, we develop a new and more modular proof that also applies to the ω-regular case.

We first introduce an abstract version of the UCS problems that is closer to PEP:

Definition 5.3 ($\mathsf{2PCEP}$)**.**
a. *The* 2-dimensional correspondence plus embedding problem *asks, given two pairs of morphisms* $f_1, g_1 : \Sigma_1^* \to \Gamma^*$ *and* $f_2, g_2 : \Sigma_2^* \to \Gamma^*$*, to find words* σ_1 *and* σ_2 *s.t.* $f_1(\sigma_1) = f_2(\sigma_2)$ *(correspondence) and* $g_1(\sigma_1) \sqsubseteq g_2(\sigma_2)$ *(embedding).*
b. $\mathsf{2PCEP}^{reg}$ *is the decision problem, where given* f_1, g_1, f_2, g_2 *and two regular languages* $R_1 \subseteq \Sigma_1^*$ *and* $R_2 \subseteq \Sigma_2^*$*, one asks whether there is a solution with* $\sigma_1 \in R_1$ *and* $\sigma_2 \in R_2$.
c. $\mathsf{2PCEP}^{\omega\text{-}reg}$ *is the infinitary version of* $\mathsf{2PCEP}^{reg}$*, where now* $R_1 \subseteq \Sigma_1^\omega$ *and* $R_2 \subseteq \Sigma_2^\omega$ *are two given* ω*-regular languages, and where one looks for* ω*-solutions with* $\sigma_1 \in R_1$ *and* $\sigma_2 \in R_2$.

Lemma 5.4 (See Appendix A).
1. $\mathsf{ReachUcs}$ *and* $\mathsf{2PCEP}^{reg}$ *are equivalent.*
2. $\mathsf{RecReachUcs}$ *and* $\mathsf{2PCEP}^{\omega\text{-}reg}$ *are equivalent.*

We now reduce 2-dim correspondence+embedding to Post embedding:

Lemma 5.5 (See Appendix A).
1. $\mathsf{2PCEP}^{reg}$ *reduces to* PEP^{reg}.
2. $\mathsf{2PCEP}^{\omega\text{-}reg}$ *reduces to* $\mathsf{PEP}^{\omega\text{-}reg}$.

We can now conclude the proof of Theorem 5.2: since $\mathsf{PEP}^{\mathrm{reg}}$ can be seen as a special case of $\mathsf{2PCEP}^{\mathrm{reg}}$ (let $f_1 = f_2 = Id$, $g_1 = u$, $g_2 = v$) and, similarly, $\mathsf{PEP}^{\omega\text{-}\mathrm{reg}}$ as a special case of $\mathsf{2PCEP}^{\omega\text{-}\mathrm{reg}}$, Lemmas 5.4 and 5.5 entail the equivalence of $\mathsf{PEP}^{\mathrm{reg}}$ and $\mathsf{ReachUcs}$ on the one hand, of $\mathsf{PEP}^{\omega\text{-}\mathrm{reg}}$ and $\mathsf{RecReachUcs}$ on the other hand.

6 Lossy channel systems

Systems composed of several finite-state components communicating via several channels (all of them lossy) can be simulated by systems with a single channel and a single component (see, e.g., [Sch02, Section 5]). Hence we define here a lossy channel system (a LCS) as a tuple $S = (Q, \mathtt{M}, \{\mathtt{c}\}, \Delta)$ as illustrated in Fig. 2. Rules read from, or write to, the single channel $\mathtt{c}$. Configurations of S are pairs $\langle q, v\rangle \in Q \times \mathtt{M}^*$ of a state and a channel contents. Transitions between configurations are obtained from the rules as expected, in the write-lossy spirit we just used for UCS's (see [CS07] for a formal

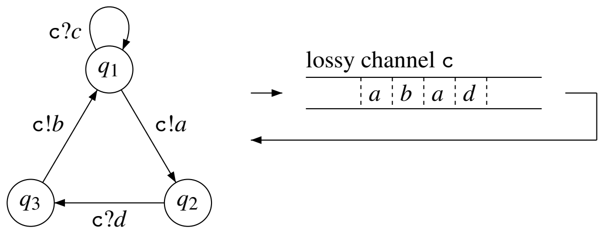

Fig. 2. A single-component system with a single lossy channel

definition).

ReachLcs, the *reachability problem for LCS's*, is the question, given a LCS S, an initial state $q_{\text{init}} \in Q$ and a set $F \subseteq Q$ of final states, whether S has a run that goes from $\langle q_{\text{init}}, \varepsilon\rangle$ to $\langle q, \varepsilon\rangle$ for some $q \in F$. RecReachLcs, the *recurrent reachability problem for LCS's*, is the question whether S has an infinite run $\langle q_{\text{init}}, \varepsilon\rangle \to \langle q_1, v_1\rangle \to \langle q_2, v_2\rangle \to \cdots$ with $q_k \in F$ for infinitely many $k \in \mathbb{N}$. Recall that ReachLcs is decidable [Pac87,AJ96b,BBS06] (albeit not primitive-recursive [Sch02]) while RecReachLcs is undecidable [AJ96a] (albeit r.e.).[2] Furthermore, ReachUcs and ReachLcs (and $\mathsf{PEP}^{\text{reg}}$) are inter-reducible [CS07].

In the rest of this section we prove the following theorem.

Theorem 6.1. $\mathsf{PEP}^{\omega\text{-}reg}_{dir}$ *and* RecReachLcs *are equivalent (inter-reducible).*

Corollary 6.2. $\mathsf{PEP}^{\omega\text{-}reg}_{dir}$ *is (r.e. but) undecidable.*

The two directions of Theorem 6.1 are given by Lemmas 6.3 and 6.4.

Lemma 6.3. $\mathsf{PEP}^{\omega\text{-}reg}_{dir}$ *reduces to* RecReachLcs.

Proof. The reduction from $\mathsf{PEP}^{\omega\text{-reg}}_{\text{dir}}$ to RecReachLcs is illustrated in Fig. 3, where the "rules" of the form $q \xrightarrow{\mathtt{c}!x\,\mathtt{c}?y} q'$ are just a shorthand description for two consecutive rules $q \xrightarrow{\mathtt{c}!x} q_?$ and $q_? \xrightarrow{\mathtt{c}?y} q'$ that traverse an anonymous intermediary state $q_?$. Simply put, the LCS $S_{u,v,R}$ mimics the Büchi automaton $\mathcal{A}_R$ that defines the constraint $R \subseteq \Sigma^\omega$. A run of the LCS that visits F infinitely often will performs steps $1, 2, 3, \ldots$, writing to the channel some v'_1, v'_2, v'_3, ..., that are subwords (because of message losses) of v_{i_1}, v_{i_2}, v_{i_3}, ...(the writes prescribed by the rules). During these same steps, it reads u_{i_1}, u_{i_2}, u_{i_3}, ..., from the channel. These read letters must have been written earlier, hence for $k = 1, 2, 3, \ldots$, $u_{i_1} \ldots u_{i_k}$ is a prefix of $v'_1 \ldots v'_k$, hence a subword of $v_{i_1} \ldots v_{i_k}$. Finally, $\sigma \stackrel{\text{def}}{=} i_1.i_2.i_3 \ldots$ is a direct solution.

Reciprocally, given a direct solution $\sigma = i_1.i_2.i_3 \ldots$, it is possible (using the general embedding provided by Lemma 3.1) to find subwords v'_1, v'_2, v'_3, ... of v_{i_1}, v_{i_2}, v_{i_3}, ... s.t., for all $k = 1, 2, \ldots$, $u_{i_1} \ldots u_{i_k}$ is a prefix of $v'_1 \ldots v'_k$. Using these v'_k, one easily obtains an infinite run of the LCS that shows the associated RecReachLcs is positive. □

[2] For Turing machines, the reachability problem is undecidable albeit r.e., while the recurrent reachability problem is Σ^1_1-complete.

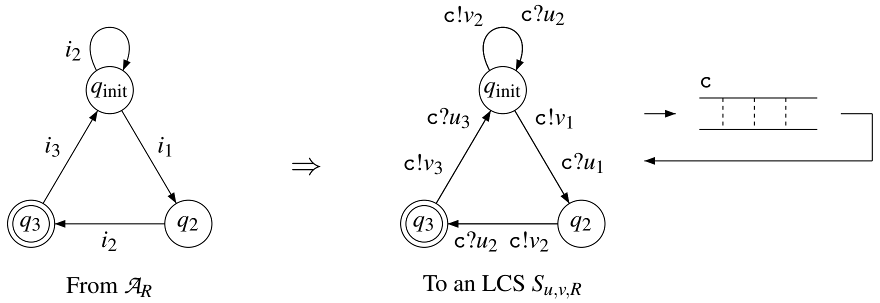

Fig. 3. Reductions between $\mathsf{PEP}^{\omega\text{-reg}}_{\text{dir}}$ and RecReachLcs

Lemma 6.4. RecReachLcs *reduces to* $\mathsf{PEP}^{\omega\text{-}reg}_{dir}$.

Proof. Consider a RecReachLcs instance $S = (Q, \mathtt{M}, \{\mathtt{c}\}, \Delta)$ with given q_{init} and F. With it, we associate a $\mathsf{PEP}^{\omega\text{-reg}}_{\text{dir}}$ instance where $\Sigma = \Delta$ and where $R \subseteq \Sigma^\omega$ is given by the Büchi automaton that is exactly like S, with the difference that any rule δ between some states q and q' is now a transition $q \xrightarrow{\delta} q'$ in $\mathcal{A}_R$. The morphisms u, v are defined by $u(\delta) \stackrel{\text{def}}{=}$ "what rule δ reads in channel c", $v(\delta) \stackrel{\text{def}}{=}$ "what δ writes in c". Since $u(\delta) = \varepsilon$ or $v(\delta) = \varepsilon$ for every rule (LCS's rules either read or write to c, not both), S (essentially) coincides with $S_{u,v,R}$ (Fig. 3). Hence the proof of Lemma 6.3 shows that u, v, R is a positive $\mathsf{PEP}^{\omega\text{-reg}}$ instance iff the original RecReachUcs instance is positive. $\square$

7 Concluding remarks

We introduced infinitary versions of $\mathsf{PEP}^{\text{reg}}$, a new and exciting variant of Post Correspondence Problem based on embedding rather than equality, which also is an abstract representative of the LCS complexity niche.

Our main result is that two such infinitary versions, $\mathsf{PEP}^{\omega\text{-reg}}$ and $\mathsf{PEP}^{\omega\text{-reg}}_{\text{codir}}$, are equivalent to the finitary $\mathsf{PEP}^{\text{reg}}$. Hence they are decidable albeit not in primitive-recursive time. Since one can link $\mathsf{PEP}^{\omega\text{-reg}}$ and RecReachUcs, the recurrent reachability problem for unidirectional channel systems, we obtain the decidability of RecReachUcs. In fact, and quite surprisingly, RecReachUcs and PEP or ReachLcs are equivalent. The last version, $\mathsf{PEP}^{\omega\text{-reg}}_{\text{codir}}$, is equivalent to RecReachLcs, the recurrent reachability problem for lossy channel systems, which is undecidable albeit r.e. Finally, the PTime-complete unconstrained PEP^{ω} is harder that the unconstrained PEP that can be solved in logspace.

References

[ADOW05] P. A. Abdulla, J. Deneux, J. Ouaknine, and J. Worrell. Decidability and complexity results for timed automata via channel machines. In *Proc. ICALP 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 1089–1101. Springer, 2005.

[AJ96a] P. A. Abdulla and B. Jonsson. Undecidable verification problems for programs with unreliable channels. *Information and Computation*, 130(1):71–90, 1996.

[AJ96b] P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. *Information and Computation*, 127(2):91–101, 1996.

[AM02] R. Amadio and Ch. Meyssonnier. On decidability of the control reachability problem in the asynchronous π-calculus. *Nordic Journal of Computing*, 9(2):70–101, 2002.

[BBS06] C. Baier, N. Bertrand, and Ph. Schnoebelen. On computing fixpoints in well-structured regular model checking, with applications to lossy channel systems. In *Proc. LPAR 2006*, volume 4246 of *Lecture Notes in Artificial Intelligence*, pages 347–361. Springer, 2006.

[Cha07] P. Chambart. Canaux fiables et non-fiables : frontières de la décidabilité. Rapport de Master, Master Parisien de Recherche en Informatique, Paris, France, September 2007.

[CS07] P. Chambart and Ph. Schnoebelen. Post embedding problem is not primitive recursive, with applications to channel systems. In *Proc. FST&TCS 2007*, volume 4855 of *Lecture Notes in Computer Science*, pages 265–276. Springer, 2007.

[Del07] G. Delzanno. Constraint-based automatic verification of abstract models of multithreaded programs. *Theory and Practice of Logic Programming*, 7(1–2):67–91, 2007.

[DL06] S. Demri and R. Lazić. LTL with the freeze quantifier and register automata. In *Proc. LICS 2006*, pages 17–26. IEEE Comp. Soc. Press, 2006.

[Fin85] A. Finkel. Une généralisation des théorèmes de Higman et de Simon aux mots infinis. *Theoretical Computer Science*, 38(1):137–142, 1985.

[GHR95] R. Greenlaw, H. J. Hoover, and W. L. Ruzzo. *Limits to Parallel Computation: P-Completeness Theory*. Oxford Univ. Press, 1995.

[GKWZ06] D. Gabelaia, A. Kurucz, F. Wolter, and M. Zakharyaschev. Non-primitive recursive decidability of products of modal logics with expanding domains. *Annals of Pure and Applied Logic*, 142(1–3):245–268, 2006.

[JL07] M. Jurdziński and R. Lazić. Alternation-free modal mu-calculus for data trees. In *Proc. LICS 2007*, pages 131–140. IEEE Comp. Soc. Press, 2007.

[Kur06] A. Kurucz. Combining modal logics. In P. Blackburn, J. van Benthem, and F. Wolter, editors, *Handbook of Modal Logics*, volume 3, chapter 15, pages 869–926. Elsevier Science, 2006.

[KWZ05] B. Konev, F. Wolter, and M Zakharyaschev. Temporal logics over transitive states. In *Proc. CADE 2005*, volume 3632 of *Lecture Notes in Computer Science*, pages 182–203. Springer, 2005.

[LNO+07] R. Lazić, T. Newcomb, J. Ouaknine, A. W. Roscoe, and J. Worrell. Nets with tokens which carry data. In *Proc. ICATPN 2007*, volume 4546 of *Lecture Notes in Computer Science*, pages 301–320. Springer, 2007.

[LW05] S. Lasota and I. Walukiewicz. Alternating timed automata. In *Proc. FOSSACS 2005*, volume 3441 of *Lecture Notes in Computer Science*, pages 250–265. Springer, 2005.

[OW06] J. Ouaknine and J. Worrell. On metric temporal logic and faulty Turing machines. In *Proc. FOSSACS 2006*, volume 3921 of *Lecture Notes in Computer Science*, pages 217–230. Springer, 2006.

[OW07] J. Ouaknine and J. Worrell. On the decidability and complexity of Metric Temporal Logic over finite words. *Logical Methods in Comp. Science*, 3(1):1–27, 2007.

[Pac87] J. K. Pachl. Protocol description and analysis based on a state transition model with channel expressions. In *Proc. PSTV 1987*, pages 207–219. North-Holland, 1987.

[Sch02] Ph. Schnoebelen. Verifying lossy channel systems has nonprimitive recursive complexity. *Information Processing Letters*, 83(5):251–261, 2002.

A Proofs for Section 5

A.1 Commuting UCS steps

We first state a trivial but important property about runs of unidirectional systems. Let $S = (Q_1, Q_2, \mathtt{M}, \{\mathtt{r}, \mathtt{l}\}, \Delta_1, \Delta_2)$ be some UCS, and $\langle q_1, q_2, x, y\rangle \xrightarrow{\delta_2} \langle q_1, q_2', x', y'\rangle \xrightarrow{\delta_1} \langle q_1', q_2', x'', y''\rangle$ be two consecutive steps with $\delta_1 \in \Delta_1$ and $\delta_2 \in \Delta_2$, i.e., where the receiver performs the first step, and the sender the second step. Then it is possible to fire δ_1 before δ_2 and reach the same configuration. More precisely, there exists x''' and y''' with $\langle q_1, q_2, x, y\rangle \xrightarrow{\delta_1} \langle q_1', q_2, x''', y'''\rangle \xrightarrow{\delta_1} \langle q_1', q_2', x'', y''\rangle$.

The corollaries are

Lemma A.1. *If S has a run $\langle q_1, q_2, x, y\rangle \xrightarrow{\Delta_1 \cup \Delta_2}{}^{*} \langle q_1', q_2', x', y'\rangle$ then it has one such run of the form*

$$\langle q_1, q_2, x, y\rangle \xrightarrow{\Delta_1}{}^{*} \langle q_1', q_2, x'', y''\rangle \xrightarrow{\Delta_2}{}^{*} \langle q_1', q_2', x', y'\rangle.$$

Lemma A.2. *If S has an infinite run from $\langle q_0^1, q_0^2, x_0, y_0\rangle$ of the form*

$$\langle q_0^1, q_0^2, x_0, y_0\rangle \to \langle q_1^1, q_1^2, x_1, y_1\rangle \to \langle q_2^1, q_2^2, x_2, y_2\rangle \to \cdots$$

with $q^1 = q_i^1$ for infinitely many i's, and $q^2 = q_i^2$ for infinitely many i's (not necessarily the same), then it has one such run with $(q^1, q^2) = (q_i^1, q_i^2)$ for infinitely many i's.

A.2 Proof of Lemma 5.4

2PCEP$^{\text{reg}}$ reduces to ReachUcs, and 2PCEP$^{\omega\text{-reg}}$ to RecReachUcs.

For this, consider a 2PCEP$^{\text{reg}}$ instance $f_1, g_1, f_2, g_2, R_1, R_2$ as in Definition 5.3.b. Further assume that, for $i = 1, 2$, R_i is given by some FSA $\mathcal{A}_i = (Q_i, \Sigma_i, q^i_{\text{init}}, F_i, \delta_i)$.

With this instance, we associate an UCS where the the sender is obtained from $\mathcal{A}_2$ by replacing transitions $q \xrightarrow{i} q' \in \delta_2$ with rules $q \xrightarrow{\mathtt{r}!f_2(i)\ \mathtt{l}!g_2(i)} q'$, and the receiver is obtained from $\mathcal{A}_1$ by replacing transitions $q \xrightarrow{i} q' \in \delta_1$ with rules $q \xrightarrow{\mathtt{r}?f_1(i)\ \mathtt{l}?g_1(i)} q'$.

If the 2PCEP$^{\text{reg}}$ instance is positive, then a solution σ_1, σ_2 can be used in a straightforward way to build, out of σ_2, a run in the UCS that will start from $\langle q^2_{\text{init}}, q^1_{\text{init}}, \varepsilon, \varepsilon\rangle$, will reach some $\langle q^2_{\text{final}}, q^1_{\text{init}}, f_2(\sigma_2), x\rangle$ for some $q^2_{\text{final}} \in F_2$, and where, using message losses, we can choose to reach any $x \sqsubseteq g_2(\sigma_2)$. By picking $x = g_1(\sigma_1)$, we can now continue the run, using σ_1, and reach $\langle q^1_{\text{final}}, q^2_{\text{final}}, \varepsilon, \varepsilon\rangle$ for some $q^1_{\text{final}} \in F_1$.

Reciprocally, using Lemma A.1, a run from $\langle q^2_{\text{init}}, q^1_{\text{init}}, \varepsilon, \varepsilon\rangle$ to some $\langle q^1_{\text{final}}, q^2_{\text{final}}, \varepsilon, \varepsilon\rangle$ can be reordered into some

$$\langle q^2_{\text{init}}, q^1_{\text{init}}, \varepsilon, \varepsilon\rangle \underbrace{\xrightarrow{r_1}\xrightarrow{r_2}\cdots\xrightarrow{r_n}}_{\text{rules from }\Delta_1} \langle q^2_{\text{final}}, q^1_{\text{init}}, x, y\rangle \underbrace{\xrightarrow{r_1'}\xrightarrow{r_2'}\cdots\xrightarrow{r_m'}}_{\text{rules from }\Delta_2} \langle q^1_{\text{final}}, q^2_{\text{final}}, \varepsilon, \varepsilon\rangle$$

where all sender's steps occur first, followed by the receiver steps. This translates into a path $q^2_{\text{init}} \xrightarrow{\sigma_2} q^2_{\text{final}}$ in $\mathcal{A}_2$, and $q^1_{\text{init}} \xrightarrow{\sigma_1} q^1_{\text{final}}$ in $\mathcal{A}_1$ where $f_2(\sigma_2) = x = f_1(\sigma_1)$, and where $g_2(\sigma_2) \sqsupseteq y = g_1(\sigma_1)$, solving the 2PCEP$^{\text{reg}}$ instance.

Finally, the $\mathsf{2PCEP}^{\mathrm{reg}}$ instance is positive iff the associated $\mathsf{ReachUcs}$ instance is. Hence $\mathsf{2PCEP}^{\mathrm{reg}}$ reduces to $\mathsf{ReachUcs}$.

The same association of an UCS with $f_1, g_1, f_2, g_2, \mathcal{A}_1, \mathcal{A}_2$ shows that $\mathsf{2PCEP}^{\omega\text{-reg}}$ reduces to $\mathsf{RecReachUcs}$.

Indeed, an infinite solution σ_1, σ_2 in some ω-regular languages R_1 and R_2, can be used to build an infinite run of the UCS that visit infinitely many configurations $\langle q^2_{\mathrm{final}}, q^1_i, x_i, y_i\rangle$ with some $q^2_{\mathrm{final}} \in F_2$, and infinitely many configurations $\langle q^2_i, q^1_{\mathrm{final}}, x'_i, y'_i\rangle$ with some $q^1_{\mathrm{final}} \in F_1$. Using Lemma A.2, this run can be reordered into a run visiting infinitely many configurations $\langle q^2_{\mathrm{final}}, q^1_{\mathrm{final}}, x''_i, y''_i\rangle$, showing the $\mathsf{RecReachUcs}$ instance is positive.

Reciprocally, from an infinite run of the UCS that visits infinitely many configurations of the form $\langle q^2_{\mathrm{final}}, q^1_{\mathrm{final}}, x''_i, y''_i\rangle$, one extracts two solutions σ_1, σ_2 that show that the $\mathsf{2PCEP}^{\omega\text{-reg}}$ instance is positive.

$\mathsf{ReachUcs}$ reduces to $\mathsf{2PCEP}^{\mathrm{reg}}$, and $\mathsf{RecReachUcs}$ to $\mathsf{2PCEP}^{\omega\text{-reg}}$.

Consider an $\mathsf{ReachUcs}$ instance with some UCS $S = (Q_1, Q_2, \mathtt{M}, \{\mathtt{r}, \mathtt{l}\}, \Delta_1, \Delta_2)$, some initial states $q^1_{\mathrm{init}}, q^2_{\mathrm{init}}$, and some sets of final states F_1, F_2.

With this instance, we associate a $\mathsf{2PCEP}^{\mathrm{reg}}$ instance where $\Sigma_1 \stackrel{\mathrm{def}}{=} \Delta_2$ and $\Sigma_2 \stackrel{\mathrm{def}}{=} \Delta_1$ are the set of rules. Automata $\mathcal{A}_1$ and $\mathcal{A}_2$ for R_1 and R_2 are obtained from the control graph of the receiver (resp., the sender) in the obvious way. (Note that we extract FSA's from an $\mathsf{ReachUcs}$ instance, and Büchi automata from an $\mathsf{RecReachUcs}$ instance.) The morphisms are defined in the obvious way:

$$f_1(\delta) \stackrel{\mathrm{def}}{=} x \text{ and } g_1(\delta) \stackrel{\mathrm{def}}{=} y \text{ for } \delta = q \xrightarrow{\mathtt{r}?x\,\mathtt{l}?y} r \text{ in } \Delta_2,$$
$$f_2(\delta) \stackrel{\mathrm{def}}{=} x \text{ and } g_2(\delta) \stackrel{\mathrm{def}}{=} y \text{ for } \delta = q \xrightarrow{\mathtt{r}!x\,\mathtt{l}!y} r \text{ in } \Delta_1.$$

A.3 Proof of Lemma 5.5

We consider a $\mathsf{2PCEP}$ instance f_1, g_1, f_2, g_2 where we assume that the morphisms are short, i.e., f_i and g_i can be seen as having type $(\Sigma_i \cup \{\varepsilon\}) \to (\Gamma_{\cup}\{\varepsilon\})$. For $\mathsf{2PCEP}^{\mathrm{reg}}$ and $\mathsf{2PCEP}^{\omega\text{-reg}}$, and thanks to the possibility offered by the regular constraints, this assumption is no loss of generality, as can be easily proved using the techniques from section 3.3.

Let $\Sigma \stackrel{\mathrm{def}}{=} (\Sigma_1 \cup \{\varepsilon\}) \times (\Sigma_2 \cup \{\varepsilon\})$ and define $X \subseteq \Sigma$ by

$$(i, j) \in X \text{ if and only if } f_1(i) = f_2(j).$$

Then $(i_1, j_1).(i_2, j_2)\ldots(i_n, j_n) \in X^*$ implies that $f_1(i_1.i_2\ldots i_n) = f_2(j_1.j_2\ldots j_n)$. Reciprocally, if $f_1(\sigma_1) = f_2(\sigma_2)$, then σ_1 and σ_2 can be decomposed under the form $\sigma_1 = i_1.i_2\ldots i_n$ and $\sigma_2 = j_1.j_2\ldots j_n$ such that $(i_k, j_k) \in X$ for $k = 1, \ldots, n$. Observe that in this decomposition, $n \geq |\sigma_i|$ is possible since $i_k = \varepsilon$ or $j_k = \varepsilon$ (or both) is allowed.

Now define projection morphisms $h_1 : \Sigma^* \to \Sigma_1^*$ and $h_2 : \Sigma^* \to \Sigma_2^*$ in the obvious way, and let $u, v : \Sigma^* \to \Gamma^*$ be two morphisms given by $u \stackrel{\mathrm{def}}{=} g_1 \circ h_1$ and $v \stackrel{\mathrm{def}}{=} g_2 \circ h_2$. Then $u_{(i_1,j_1).(i_2,j_2)\ldots(i_n,j_n)} \sqsubseteq v_{(i_1,j_1).(i_2,j_2)\ldots(i_n,j_n)}$ if and only if $g_1(i_1.i_2\ldots i_n) \sqsubseteq g_2(j_1.j_2\ldots j_n)$.

Finally, the 2PCEP$^{\mathrm{reg}}$ instance with regular constraints R_1, R_2 translates into an equivalent PEP$^{\mathrm{reg}}$ instance, with morphisms u and v as above, and with constraint

$$R \stackrel{\text{def}}{=} X^* \cap {h_1}^{-1}(R_1) \cap {h_2}^{-1}(R_2),$$

which is regular. Similarly, the 2PCEP$^{\omega\text{-reg}}$ instance with ω-regular constraints R_1, R_2 translates into an equivalent PEP$^{\omega\text{-reg}}$ instance, with same morphisms u and v, and with constraint

$$R \stackrel{\text{def}}{=} X^\omega \cap {h_1}^{-1}(R_1) \cap {h_2}^{-1}(R_2),$$

which is ω-regular.

B PEP$^\omega$ is PTime-hard

We reduce CircuitValue to PEP$^\omega$. Let $\mathcal{C} = (G_\vee, G_\wedge, G_\top, G_\perp, f_1, f_2, n_0)$ be an instance of CircuitValue, as illustrated in Fig 4. We assume, without loss of generality [GHR95,

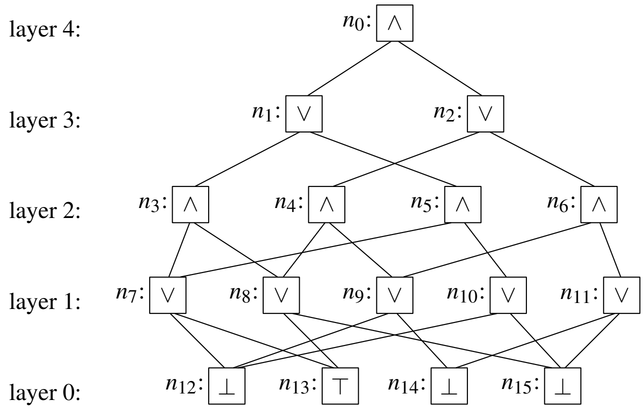

Fig. 4. An instance of CircuitValue.

problem A.1.6], that gates are arranged in layers, that layer 0 contains "constants" gates from $G_\top \cup G_\perp$, that, for any, $k \in \mathbb{N}$ layer $2k+1$ (resp. $2k+2$) contains OR-gates (resp. AND-gates) from $G_\vee$ (resp. $G_\wedge$), that any gate n in some layer $k > 0$ has exactly two inputs, $f_1(n)$ and $f_2(n)$, that belong to layer $k-1$ (NB: $f_1(n) = f_2(n)$ is allowed). Finally, we assume that the output n_0 of $\mathcal{C}$ belongs to $G_\wedge$.

Given a circuit $\mathcal{C}$, we define in the obvious way the value $val(n) \in \{0,1\}$ of gate $n \in G$, where $G \stackrel{\text{def}}{=} G_\vee \cup G_\wedge \cup G_\top \cup G_\perp$ is the set of gates. Let $G_{=1} \stackrel{\text{def}}{=} \{n \in G \mid val(n) = 1\}$. In our example, $G_{=1} = \{n_1, n_3, n_7, n_8, n_{13}\}$.

With $\mathcal{C}$ we associate two morphisms $u, v : \Sigma^* \to \Gamma^*$ as follows. Let $\Sigma \stackrel{\text{def}}{=} G_\wedge \cup (G_\vee \times \{1,2\}) \cup G_\top$ and $\Gamma \stackrel{\text{def}}{=} G$.

$$u(n) \stackrel{\text{def}}{=} f_1(n).f_2(n).n_0 \qquad v(n) \stackrel{\text{def}}{=} n \qquad \text{for } n \in G_\wedge, \tag{C1}$$

$$u(n,i) \stackrel{\text{def}}{=} f_i(n).n_0 \qquad v(n,i) \stackrel{\text{def}}{=} n \qquad \text{for } n \in G_\vee \times \{1,2\}, \tag{C2}$$

$$u(n) \stackrel{\text{def}}{=} n_0 \qquad v(n) \stackrel{\text{def}}{=} n \qquad \text{for } n \in G_\top. \tag{C3}$$

The reduction is clearly logspace. Its correctness is established by the following two lemmas.

Lemma B.1. *If $val(n_0) = 1$, then there is a non-empty Σ' with $alph(u(\Sigma')) \subseteq alph(v(\Sigma'))$.*

Proof. Let

$$\Sigma' \stackrel{\text{def}}{=} \begin{array}{l} \{n \in G_\wedge \cup G_\top \mid val(n) = 1\} \\ \cup \ \{(n,i) \in G_\vee \times \{1,2\} \mid val(f_i(n)) = 1\}. \end{array}$$

Σ' is not empty since it contains n_0. Observe that $alph(v(\Sigma'))$ is exactly $G_{=1}$. It remains to check, by inspecting (C1–3), that $x \in \Sigma'$ implies $alph(u(x)) \subseteq G_{=1}$. ☐

Lemma B.2. *Assume that $alph(u(\Sigma')) \subseteq alph(v(\Sigma'))$ for some non-empty $\Sigma' \subseteq \Sigma$. Then $val(n_0) = 1$.*

Proof. Since necessarily n_0 appears in $alph(u(\Sigma'))$, hence in $alph(v(\Sigma'))$, it is enough to show that $alph(v(\Sigma')) \subseteq G_{=1}$. We do this by induction on layers. Let $x \in \Sigma'$ and consider three cases. If $x \in G_\top$, then $x \in G_{=1}$ obviously. If $x \in G_\wedge$, then $alph(u(x)) \subseteq alph(v(\Sigma'))$ implies that both $f_1(x)$ and $f_2(x)$ belong to $alph(v(\Sigma'))$, hence evaluate to 1 by ind. hyp., so that $val(x) = 1$. Finally, if x is some $(n,i) \in G_\vee \times \{1,2\}$, then from $f_i(n) = u(x) \in alph(v(\Sigma'))$, we deduce that $f_i(n) \in G_{=1}$ by ind. hyp., hence $val(n) = 1$, proving $v(x) \in G_{=1}$. ☐