Masaryk University Brno
Faculty of Sciences

Deciding Existence of Trace Codings (Abstract of Ph.D. thesis)

Michal Kunc

Brno, May 2003

Mathematics Subject Classification: 20M05, 20M15, 68Q85.

1 Introduction

In 1977 Mazurkiewicz [14] proposed free partially commutative monoids as a tool for describing behaviour of concurrent systems. In this approach one represents elementary actions with letters of a given alphabet; then an observation of a finite run of a system is just a word over the alphabet. Two words are regarded as describing the same behaviour whenever they can be obtained from each other by commuting adjacent occurrences of letters representing concurrent actions. The mathematical formalization for this concept is provided by considering finitely presented monoids with their defining relations expressing commutativity of some of their generators. Possible behaviours of a system then correspond to congruence classes modulo the defining relations, which are called traces. In this way causal order of actions is distinguished from the order arising from sequentiality of observations.

1 Definition. Let Σ be a finite set and let I be a symmetric and reflexive binary relation on Σ. We call I an *independence relation* on Σ and the undirected graph (Σ, I) an *independence alphabet*. The complement of this relation $D = (\Sigma \times \Sigma) \setminus I$ is called a *dependence relation* and the graph (Σ, D) a *dependence alphabet*. Let $\sim_I$ be the congruence of the free monoid Σ^* generated by the relation $\{(xy, yx) \mid (x,y) \in I\}$. The quotient monoid $\Sigma^*/\sim_I$ is denoted by $\mathbb{M}(\Sigma, I)$ and called a *trace monoid*. Elements of this monoid are called *traces*.

Traces $s, t \in \mathbb{M}(\Sigma, I)$ are called *independent* if $\mathrm{alph}(s) \times \mathrm{alph}(t) \subseteq I \setminus \mathrm{id}_\Sigma$ holds.

Let us recall that a *code* is a finite set of words satisfying no non-trivial relation. As codes naturally correspond to injective morphisms of free monoids, one can use this characterization to generalize the notion of codes to trace monoids. Therefore a morphism of trace monoids (trace morphism) is called a *coding* whenever it is injective.

The basic decision problem of trace codings is the problem of determining whether a given trace morphism is a coding. A decision procedure for injectivity of word morphisms is well-known and for free commutative monoids injectivity coincides with linear independence of images of letters. But for trace monoids the problem was proved undecidable already in [10]. By means of the theorem of Aalbersberg and Hoogeboom [1] the classical positive results can be extended to the case of codomain monoids whose independence graph forms a transitive forest. Further work on classifying monoids having the injectivity problem decidable has been done in [7, 9, 13, 15], but the exact borderline between decidable and undecidable cases is still unknown.

In 1988 Ochmański [16] formulated several problems about trace codings. One of his conjectures, which was proved true in 1996 by Bruyère and De Felice, can be stated as follows. It is easy to see that each trace morphism φ is determined by a word morphism ψ of the corresponding free monoids through the commutative diagram

$$\begin{array}{ccc} \Sigma^* & \xrightarrow{\psi} & (\Sigma')^* \\ {\scriptstyle \nu}\big\downarrow & & \big\downarrow{\scriptstyle \nu'} \\ \mathbb{M}(\Sigma,I) & \xrightarrow[\varphi]{} & \mathbb{M}(\Sigma',I') \end{array}$$

Such morphisms ψ are called *liftings* of φ and they satisfy the following claim.

2 Proposition ([3]). *For an arbitrary trace coding φ, every lifting of φ to the corresponding free monoids is a coding.*

Another Ochmański's question asked to give an algorithm deciding for any given pair of trace monoids whether there exists a coding between them, i.e. whether the first of the given monoids is isomorphic to a submonoid of the second one.

3 Definition. Let $\mathscr{C}$ be an arbitrary class of trace morphisms. The *trace coding problem* for the class $\mathscr{C}$ (in short $\mathscr{C}$-TCP) asks to decide for given two independence alphabets (Σ,I) and (Σ',I') whether there exists a coding from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$ belonging to $\mathscr{C}$. If $\mathscr{C}$ contains all trace morphisms, then we talk briefly about the trace coding problem (TCP).

The TCP appears to be rather intractable since there is no obvious enumeration procedure either for all submonoids of trace monoids being itself trace monoids (due to the undecidability of the injectivity of morphisms) or for the pairs of monoids where no coding exists (as there are usually infinitely many candidates for being codings).

The classical cases of the problem are simple: all finitely generated free monoids can be embedded into the one with two generators and for embedding a free commutative monoid into another one we need at least the same number of generators. These characterizations were generalized in [2] to all instances of the trace coding problem where the domain monoid is a direct product of free monoids. Some partial results about the case of domain monoids being free products of free commutative monoids were obtained in [4]. But in the full generality the problem remained completely open.

In connection with decision problems of trace codings, two particular classes of trace morphisms were already considered:

- strong morphisms, introduced in [5],
- cp-morphisms, which were introduced in [8] as morphisms associated with clique-preserving morphisms of independence alphabets.

In order to deal with the general case, we have generalized the latter notion and we refer to the arising morphisms as weak. Let $\mathrm{alph} : \Sigma^* \to 2^\Sigma$ denote the *content* mapping assigning to every word $u \in \Sigma^*$ the set of all letters occurring in u.

4 Definition (1.2.4). A morphism $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ is called *strong* if

$$\forall (x,y) \in I \setminus \mathrm{id}_\Sigma : \ \mathrm{alph}(\varphi(x)) \cap \mathrm{alph}(\varphi(y)) = \emptyset \ .$$

It is called *weak* if $\mathrm{alph}(\varphi(x)) \times \mathrm{alph}(\varphi(x)) \subseteq I'$ for all letters $x \in \Sigma$. And it is called a *cp-morphism* if it is weak and the image of every $x \in \Sigma$ contains at most one occurrence of each $a \in \Sigma'$.

We denote the classes of all strong and weak morphisms by $\mathscr{S}$, $\mathscr{W}$ respectively. The $\mathscr{S}$-TCP is known to be NP-complete due to the following result.

5 Proposition ([8]). *Let (Σ, I) and (Σ', I') be independence alphabets and let $H : \Sigma \to 2^{\Sigma'}$ be any mapping. Then there exists a strong coding from $\mathbb{M}(\Sigma, I)$ to $\mathbb{M}(\Sigma', I')$ satisfying $\mathrm{alph} \circ \varphi|_\Sigma = H$ if and only if for every $x, y \in \Sigma$:*

$$H(x) \times H(y) \subseteq I' \setminus \mathrm{id}_{\Sigma'} \iff (x,y) \in I \setminus \mathrm{id}_\Sigma \ ,$$
$$H(x) \times H(y) \subseteq I' \implies (x,y) \in I \ .$$

2 Overview

The thesis consists of five chapters. Basic definitions and known results are recalled in Chapter 1. In Chapter 2 we study properties of weak morphisms and the so-called co-strong morphisms. The main result of this section shows that the problem of existence of weak codings is even more complex than the original problem for general morphisms. The following Chapter 3 is devoted to proving the decidability of this problem for some classes of instances, which entails positive answers for the corresponding cases of the original question. On the other hand, in Chapter 4 we prove that in general the existence of codings from any given family of trace morphisms containing all weak codings is undecidable, thus answering the Ochmański's question negatively. The final Chapter 5 is devoted to summarizing the main results.

Most of the material presented in the thesis (except for Section 2.3) is contained in the paper [12] currently submitted for publication; main ideas of these results were briefly described in the extended abstract [11].

3 Restricted Classes of Morphisms

Chapter 2 is devoted to the study of classes of trace morphisms defined by additional requirements on contents of images of generators of the domain monoid.

In Section 2.1 we investigate properties of counter-examples to injectivity for weak trace morphisms, introduce some techniques for manipulating weak morphisms and develop several methods of disproving their injectivity.

The aim of Section 2.2 is to describe how the original trace coding problem is connected with its analogue for weak morphisms. To reveal connections between weak morphisms and general ones we employ the standard decomposition of traces into primitive roots of connected components.

A trace $s \in \mathbb{M}(\Sigma, I) \setminus \{1\}$ is called *connected* whenever the graph $(\mathrm{alph}(s), D)$ is connected. Clearly every trace $s \in \mathbb{M}(\Sigma, I)$ can be uniquely decomposed as a product of independent connected traces, which are referred to as *connected components* of s.

A trace $s \in \mathbb{M}(\Sigma, I)$ is *primitive* if it is connected and for every $t \in \mathbb{M}(\Sigma, I)$ and $n \in \mathbb{N}$, the equality $s = t^n$ implies $n = 1$. It is well-known that any connected trace s is a power of a unique primitive trace t, called the *primitive root* of s.

For a trace morphism $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$, we consider for every letter $x \in \Sigma$ the decomposition of the image $\varphi(x \sim_I)$ into primitive roots of connected components. It is a fundamental property of primitive traces that they do not commute unless they are equal or independent and therefore the substantial information characterizing their behaviour is their content. So, we introduce sufficiently many new letters for each possible content with the aim of replacing these primitive roots with them.

6 Definition (2.2.1). Let (Σ, I), (Σ', I') be independence alphabets. We define the independence alphabet $({\Sigma'}_\Sigma, {I'}_\Sigma)$ as follows. Let ${\Sigma'}_\Sigma = \Sigma' \cup (\overline{\mathscr{C}}(\Sigma', D') \times \Sigma)$, where $\overline{\mathscr{C}}(\Sigma', D')$ denotes the set of all connected subgraphs of (Σ', D') with at lest 2 vertices. For a word $u \in ({\Sigma'}_\Sigma)^*$, define its *extended content* $\mathrm{ealph}(u) \subseteq \Sigma'$ as

$$\mathrm{ealph}(u) = (\mathrm{alph}(u) \cap \Sigma') \cup \bigcup \{A \mid (A, x) \in \mathrm{alph}(u) \setminus \Sigma'\} \ .$$

Finally, for $\alpha, \beta \in {\Sigma'}_{\Sigma}$, set

$$(\alpha, \beta) \in {I'}_{\Sigma} \iff \mathrm{ealph}(\alpha) \times \mathrm{ealph}(\beta) \subseteq I' \text{ or } \alpha = \beta \ .$$

In this way we express every morphism φ as a composition of a weak morphism and a strong morphism.

7 Proposition (2.2.4). *Let $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ be a morphism of trace monoids. Then there exist a weak morphism $\psi : \mathbb{M}(\Sigma, I) \to \mathbb{M}({\Sigma'}_{\Sigma}, {I'}_{\Sigma})$ and a strong morphism $\sigma : \mathbb{M}({\Sigma'}_{\Sigma}, {I'}_{\Sigma}) \to \mathbb{M}(\Sigma', I')$ such that $\sigma \circ \psi = \varphi$.*

Clearly, if φ is a coding, the weak morphism ψ must be a coding as well. On the other hand, we can use Proposition 5 to find a strong coding for prolonging any coding to the new codomain monoid into a coding to the original one.

8 Proposition (2.2.5). *The following conditions are equivalent.*

(i) There exists a coding from $\mathbb{M}(\Sigma, I)$ to $\mathbb{M}(\Sigma', I')$.
(ii) There exists a weak coding from $\mathbb{M}(\Sigma, I)$ to $\mathbb{M}({\Sigma'}_{\Sigma}, {I'}_{\Sigma})$.
(iii) There exists a coding from $\mathbb{M}(\Sigma, I)$ to $\mathbb{M}({\Sigma'}_{\Sigma}, {I'}_{\Sigma})$.

As an immediate consequence of this result we obtain:

9 Theorem (5.1). *If $\mathscr{C}$ is any class of trace morphisms containing all weak codings, then there exists an effective reduction of the TCP to the $\mathscr{C}$-TCP.*

The following claim, which is proved by shifting calculations to the case of weak codings using Proposition 8, allows us to restrict to instances of the TCP whose domain monoids have connected dependence alphabets.

10 Proposition (2.2.9). *Let (Σ_i, I_i) for $i \in \{1, \ldots, n\}$ and (Σ', I') be independence alphabets. Then there exists a coding from $\prod_{i=1}^{n} \mathbb{M}(\Sigma_i, I_i)$ to $\mathbb{M}(\Sigma', I')$ if and only if there exist $\Sigma'_i \subseteq \Sigma'$, for every $i \in \{1, \ldots, n\}$, such that $\Sigma'_i \times \Sigma'_j \subseteq I' \setminus \mathrm{id}_{\Sigma'}$ holds for every $i, j \in \{1, \ldots, n\}$, $i \neq j$, and codings $\varphi_i : \mathbb{M}(\Sigma_i, I_i) \to \mathbb{M}(\Sigma'_i, I')$ for every $i \in \{1, \ldots, n\}$.*

11 Theorem (5.2). *The TCP is effectively reducible to instances with domain monoids defined by connected dependence alphabets.*

In Section 2.3 we consider the condition obtained by replacing the reference to the independence relation in the definition of strong trace morphisms with the reference to the corresponding dependence relation.

12 Definition (2.3.1). We call a morphism $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ *co-strong* if

$$\forall (x, y) \in D : \ \mathrm{alph}(\varphi(x)) \cap \mathrm{alph}(\varphi(y)) = \emptyset \ .$$

We denote by $\mathscr{C}\mathscr{S}$ the class of all co-strong trace morphisms.

We prove that the construction of Section 2.2 can be performed also for the problem of existence of co-strong codings, i.e. that the $\mathscr{C}\mathscr{S}$-TCP is effectively reducible to the $\mathscr{C}\mathscr{S}\cap\mathscr{W}$-TCP. There are two main differences between these situations. First, for co-strong codings the new independence alphabet can be constructed independently of the domain monoid. Second, in this case it is not enough to consider just one new monoid; we have to introduce a set of monoids such that every co-strong morphism factorizes through one of them.

We define the independence relation $\mathscr{P}(I')$ on the set $\mathscr{C}(\Sigma',D')$ of all connected subgraphs of (Σ',D') by the rule:

$$(A,B)\in\mathscr{P}(I') \iff A\times B\subseteq I' \text{ or } A=B\,.$$

13 Proposition (2.3.7). *Let $\varphi:\mathbb{M}(\Sigma,I)\to\mathbb{M}(\Sigma',I')$ be a co-strong morphism. Then there exist $\Sigma_1\subseteq\mathscr{C}(\Sigma',D')$ satisfying*

$$\forall A,B\in\Sigma_1 : A\cap B\neq\emptyset \implies A=B\,,$$

a co-strong and weak morphism $\psi:\mathbb{M}(\Sigma,I)\to\mathbb{M}(\Sigma_1,\mathscr{P}(I'))$ and a co-strong and strong morphism $\sigma:\mathbb{M}(\Sigma_1,\mathscr{P}(I'))\to\mathbb{M}(\Sigma',I')$ such that $\sigma\circ\psi=\varphi$.

14 Corollary (2.3.10). *For an arbitrary class $\mathscr{C}$ of co-strong trace morphisms containing all co-strong and weak codings, there exists an effective reduction of the $\mathscr{C}\mathscr{S}$-TCP to the $\mathscr{C}$-TCP.*

By means of proving the corresponding result for weak codings we also show that if the domain dependence graph does not contain cycles of length 3, then one can construct a co-strong coding whenever there exists an arbitrary coding.

15 Proposition (2.3.13). *Let (Σ,I) and (Σ',I') be any independence alphabets such that the graph (Σ,D) is C_3-free. Then there exists a coding from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$ if and only if there exists a co-strong coding from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$.*

4 Decidable Cases

In Chapter 3 we show that in some cases the existence of a weak coding between trace monoids $\mathbb{M}(\Sigma,I)$ and $\mathbb{M}(\Sigma',I')$ is equivalent to the existence of a choice of contents of images of generators of $\mathbb{M}(\Sigma,I)$ satisfying certain regularity conditions. This choice is provided by a mapping $f:\Sigma\to 2^{\Sigma'}$; besides putting requirements on f assuring that it allows us to define a weak morphism and guarding against linear dependence on free commutative submonoids, we introduce a condition ensuring unique decipherability on every submonoid of $\mathbb{M}(\Sigma,I)$ generated by a subset of Σ on which the dependence relation forms a tree. Mappings satisfying these conditions are called wlt-mappings.

A crucial role in our considerations is played by those letters of the codomain alphabet which occur in the image of exactly one generator of the domain monoid.

16 Definition (3.1.1). Let (Σ, I) and (Σ', I') be independence alphabets. Let $f : \Sigma \to 2^{\Sigma'}$ be an arbitrary mapping and let $x \in X \subseteq \Sigma$. The set of *central letters* for X in $f(x)$ with respect to f is defined as

$$\mathsf{C}_f^X(x) = \{a \in f(x) \mid (\exists y \in X,\ b \in f(y) :\ (a,b) \in D')\ \&\ \&\ (\forall y \in X :\ a \in f(y) \implies x = y)\}\ .$$

17 Definition (3.1.2). Let (Σ, I) and (Σ', I') be independence alphabets and $f : \Sigma \to 2^{\Sigma'}$ a mapping. We call f a *wlt-mapping* from (Σ, I) to (Σ', I') if it satisfies the following conditions (W), (L) and (T).

(W) — weakness:

For every $x, y \in \Sigma : (x,y) \in I \iff f(x) \times f(y) \subseteq I'$.

(L) — regularity on linear parts:

For all $X \subseteq \Sigma$ such that $X \times X \subseteq I$, there exists an injective mapping $\rho_X : X \to \Sigma'$ satisfying $\forall x \in X :\ \rho_X(x) \in f(x)$.

(T) — regularity on trees:

For all $X \subseteq \Sigma$ such that (X, D) is a tree, there exist a letter $x \in X$ and an injective mapping $\sigma_{X,x} : \{y \in X \mid (x,y) \in D\} \to \Sigma'$ such that for all $y \in X$ satisfying $(x,y) \in D$ it holds $\sigma_{X,x}(y) \in f(y)$ and there exists $a \in \mathsf{C}_f^X(x)$ satisfying $(a, \sigma_{X,x}(y)) \in D'$.

Section 3.1 is devoted to discussing properties of wlt-mappings.

In Section 3.2 we first prove that in general the existence of a wlt-mapping is always necessary for the existence of a weak coding.

18 Lemma (3.2.1). *Let $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ be any weak coding. Then* $\mathrm{alph} \circ \varphi|_\Sigma$ *is a wlt-mapping from (Σ, I) to (Σ', I').*

Conversely, for some cases we show that to every wlt-mapping f one can construct a weak coding φ such that $\mathrm{alph} \circ \varphi|_\Sigma = f$ and we obtain the following result.

19 Proposition (3.2.3). *Let (Σ, I) and (Σ', I') be independence alphabets such that either $\mathbb{M}(\Sigma, I)$ is a direct product of free monoids or the graph (Σ, D) is acyclic. Then there exists a weak coding from $\mathbb{M}(\Sigma, I)$ to $\mathbb{M}(\Sigma', I')$ if and only if there exists a wlt-mapping from (Σ, I) to (Σ', I').*

In Section 3.3 we present a solution of the $\mathscr{W}$-TCP for all instances which have the dependence alphabet of the domain monoid C_3, C_4-free (not containing cycles of length less than 5) and whose codomain monoid is a direct product of free monoids. Because the property of being a direct product of free monoids is preserved by the construction of Definition 6, we obtain also the corresponding positive result for the TCP by means of Proposition 8. But unlike in the cases covered by Proposition 19, in this situation it is not true that for every wlt-mapping f there exists a coding φ such that $\mathrm{alph} \circ \varphi|_\Sigma = f$. Our approach is based on calculating how many of the free submonoids of the codomain monoid have their elements employed by a given wlt-mapping. We show that there are in fact always enough letters for constructing some morphism whose injectivity is easy to prove.

20 Proposition (3.3.4). *Let (Σ, D) be an arbitrary C_3, C_4-free dependence alphabet and let $\mathbb{M}(\Sigma', I')$ be a direct product of m free monoids over at least two generators and n free one-generated monoids. Let M be the number of non-trivial connected components of the graph (Σ, D) and let N be the number of trivial ones. Let a_i, b_i for $i \in \{1, \ldots, m\}$ and c_i for $i \in \{1, \ldots, n\}$ be distinct letters and consider the monoid*

$$\mathbb{M}(\Sigma_1, I_1) = \prod_{i=1}^{m} \{a_i, b_i\}^* \times \prod_{i=1}^{n} \{c_i\}^* .$$

Then the following statements are equivalent.

(i) There exists a weak coding from $\mathbb{M}(\Sigma, I)$ to $\mathbb{M}(\Sigma', I')$.

(ii) There exists a coding from $\mathbb{M}(\Sigma, I)$ to $\mathbb{M}(\Sigma', I')$.

(iii) There exists a weak coding from $\mathbb{M}(\Sigma, I)$ to $\mathbb{M}(\Sigma_1, I_1)$.

(iv) There exists a wlt-mapping from (Σ, I) to (Σ_1, I_1).

(v) $|\Sigma| - M - N \le m$ and $|\Sigma| - M \le m + n$.

As a direct consequence of Propositions 19 and 20 we obtain:

21 Theorem (5.3). *The $\mathscr{W}$-TCP restricted to instances with independence alphabets (Σ, I) and (Σ', I') satisfying one of the following conditions is decidable.*

(i) $\mathbb{M}(\Sigma, I)$ is a direct product of free monoids.

(ii) The graph (Σ, D) is acyclic.

(iii) The graph (Σ, D) is C_3, C_4-free and $\mathbb{M}(\Sigma', I')$ is a direct product of free monoids.

Because the reduction described in Proposition 8 preserves the domain monoid, we immediately deduce the following statements about general codings.

22 Corollary ([2]). *The restriction of the TCP to instances whose domain monoids are direct products of free monoids is decidable.*

23 Corollary (5.5). *The TCP restricted to instances with domain monoids defined by acyclic dependence alphabets is decidable.*

Proposition 20 also partially answers the question of Diekert [6] about the number of free monoids needed for encoding a given trace monoid into their direct product.

24 Theorem (5.6). *Let (Σ, D) be a C_3, C_4-free dependence alphabet. Then there exists a coding from $\mathbb{M}(\Sigma, I)$ to $(\{a, b\}^*)^m$ if and only if $m \ge |\Sigma| - M$, where M is the number of non-trivial connected components of the graph (Σ, D).*

Finally, the aim of Section 3.4 is to show that none of the assumptions of Proposition 20 can be avoided.

5 The General Case

Chapter 4 is devoted to proving the undecidability of the TCP by a reduction of the complement of the Post's correspondence problem (coPCP), which is well-known not to be recursively enumerable. The proof proceeds in two steps via the problem of existence of weak codings with partially prescribed contents of images of letters.

The aim of Section 4.1 is to describe how the problem of existence of weak codings satisfying certain requirements on contents of images of letters can be effectively reduced to the TCP. We use two mappings $\mu, \nu : \Sigma \to 2^{\Sigma'}$ to specify these restrictions on contents. A weak morphism $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ is called (μ, ν)-*weak* if it satisfies for all $x \in \Sigma$ the condition $\mu(x) \subseteq \mathrm{alph}(\varphi(x)) \subseteq \nu(x)$.

First, we show how to specify mandatory letters defined by μ using only the mapping ν. There is nothing to take care of for $x \in \Sigma$ such that $|\nu(x)| = 1$ because $\mathrm{alph}(\varphi(x)) = \nu(x)$ is satisfied for every ν-weak coding φ. The idea of the construction is to enrich each of the original alphabets with the same set Θ of new letters and define $\nu(y) = \{y\}$ for every $y \in \Theta$; since the behaviour of an arbitrary ν-weak coding on these letters is obvious, they can serve as a skeleton for prescribing contents of images of other letters. More precisely, to ensure that the image of x under every ν-weak coding contains $a \in \Sigma'$, we introduce a letter $(x,a) \in \Theta$ dependent on x in the domain alphabet and dependent only on the letter a in the codomain alphabet.

In a similar way we show that one can manage the content requirements even without the mapping ν. This time, we add to the alphabets mutually dependent cliques of independent letters, each of them having sufficiently distinct size. Then one can verify that images of elements of a given clique in the domain alphabet under a weak coding use almost exclusively letters from the clique of the same size in the codomain alphabet. So, in order to deal with the requirements for a letter $x \in \Sigma$, we introduce a clique which has all of its elements independent on x in the domain alphabet and independent exactly on letters allowed in the image of x in the codomain alphabet. Because images of independent letters under a weak morphism always contain only independent ones, this ensures that prohibited letters are never used. And since the letters added to the codomain alphabet according to Definition 6 do not form in the independence alphabet any cliques bigger than those already existing, this construction functions in the same way even if we use Proposition 8 to pass to the TCP.

Altogether, we obtain the following result.

25 Proposition (4.1.2 + 4.1.3). *Let (Σ, I) and (Σ', I') be independence alphabets such that the monoid $\mathbb{M}(\Sigma, I)$ is a free product of at least two non-trivial free commutative monoids. Let $\mu, \nu : \Sigma \to 2^{\Sigma'}$ be mappings satisfying for all $x, y \in \Sigma$:*

$$x \, I \, y,\ x \neq y \implies \mu(x) = \mu(y) = \emptyset \,,$$
$$\nu(x) \times \nu(x) \subseteq I', \qquad x \, I \, y \implies \nu(x) = \nu(y) \,.$$

Then one can effectively construct independence alphabets (Σ_1, I_1) and (Σ_1', I_1') such that the following statements are equivalent.

(i) There exists a (μ,ν)-weak coding from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$.

(ii) There exists a weak coding from $\mathbb{M}(\Sigma_1,I_1)$ to $\mathbb{M}(\Sigma_1',I_1')$.

(iii) There exists a weak coding from $\mathbb{M}(\Sigma_1,I_1)$ to $\mathbb{M}((\Sigma_1')_{\Sigma_1},(I_1')_{\Sigma_1})$.

(iv) There exists a coding from $\mathbb{M}(\Sigma_1,I_1)$ to $\mathbb{M}(\Sigma_1',I_1')$.

In Section 4.2 we construct a reduction of the coPCP to the problem of existence of (μ,ν)-weak codings. It is enough to deal with those instances of the PCP where neighbouring letters are always distinct. This enables us not to care about the numbers of occurrences of letters in images under (μ,ν)-weak morphisms.

For every suitable instance $\mathscr{P}$ of the PCP, we construct two independence alphabets (Σ,I) and (Σ',I') and mappings $\mu,\nu : \Sigma \to 2^{\Sigma'}$ in such a way that all assumptions of Proposition 25 are satisfied. In the outcome of our construction, counter-examples to injectivity for (μ,ν)-weak morphisms correspond to solutions of the instance $\mathscr{P}$. The domain alphabet $\Sigma = \Omega \times \{1,2\}$ consists of one pair of letters for each element of a certain set Ω. Letters from these pairs appear on opposite sides of counter-examples to injectivity and correspond there to each other according to their first coordinates. For each element $\omega \in \Omega$, the images of $(\omega,1)$ and $(\omega,2)$ under μ and ν are together used as one rule for a computation.

Computations of counter-examples to injectivity we deal with consist in constructing and prolonging potential initial parts of such counter-examples, which we call semi-equalities. We say that $(u,v) \in \Sigma^* \times \Sigma^*$ is a *semi-equality* for a morphism $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ if there exist $s,t \in (\Sigma')^*$ such that $\varphi(u)s \sim_{I'} \varphi(v)t$. We call this semi-equality *non-trivial* if there do not exist words $w,r \in \Sigma^*$ such that $uw \sim_I vr$.

If (u,v) is a semi-equality for a morphism $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$, we can consider the traces $u',v' \in \mathbb{M}(\Sigma',I')$ consisting of those occurrences of letters in $\varphi(u)$ and $\varphi(v)$ which have no corresponding occurrence in $\varphi(v)$, $\varphi(u)$ respectively. Then the pair (u',v') is called the *state* of (u,v) and the pair $(\mathrm{red}(u'),\mathrm{red}(v'))$ the *reduced state* of (u,v), where $\mathrm{red}(s)$ denotes the trace obtained from s by removing, for every $a \in \Sigma'$, from each block of occurrences of a which can be moved together by interchanging neighbouring independent letters all but one occurrence.

All of the information we need to explore possible continuations of a semi-equality is contained in its state. And every semi-equality for a weak morphism whose state consists entirely of independent letters can be prolonged into a counter-example:

26 Lemma (2.1.22). *Let φ be a weak morphism from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$ such that there exists a non-trivial semi-equality (u,v) for φ with a state (u',v') which satisfies $\mathrm{alph}(u'v') \times \mathrm{alph}(u'v') \subseteq I'$. Then φ is not a coding.*

Because a counter-example has to be found regardless of numbers of occurrences of letters in the images of elements of Σ, the reduced state contains exactly the information common to all the possible cases. The original alphabet of the instance $\mathscr{P}$ is made part of the alphabet Σ'. A computation of a solution of $\mathscr{P}$ is simulated by appending the pairs of elements of Σ to an already constructed semi-equality in the way determined by its reduced state and by dependences between elements of Σ' until Lemma 26 can be applied; letters of the original alphabet are thus accumulated in the desired way.

Finally, we define a special (μ,ν)-weak morphism φ from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$ and prove the following result.

27 Proposition (4.2.1). *The following statements are equivalent.*

(i) *$\mathscr{P}$ has no solution.*

(ii) *φ is a coding.*

(iii) *There exists a (μ,ν)-weak coding from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$.*

As a side result, Proposition 27 immediately implies that injectivity is not decidable for cp-morphisms, which reproves the main result of [7].

Propositions 25 and 27 together give an effective reduction of the coPCP to the TCP and consequently the undecidability of the TCP.

28 Theorem (5.7). *The TCP is not recursively enumerable.*

The same assertion for certain classes of trace morphisms holds due to Theorem 9.

29 Corollary (5.8). *If $\mathscr{C}$ is any class of trace morphisms containing all weak codings, then the $\mathscr{C}$-TCP is not recursively enumerable.*

References

[1] IJ.J. Aalbersberg, H.J. Hoogeboom, Characterizations of the decidability of some problems for regular trace languages, *Math. Syst. Theory* **22**(1) (1989) 1–19.

[2] V. Bruyère, C. De Felice, Coding and strong coding in trace monoids, in: E.W. Mayr, C. Puech (Eds.), *Proc. 12th Annu. Sympos. on Theoretical Aspects of Computer Science (STACS'95)*, Lecture Notes in Comput. Sci., vol. 900, Springer, Berlin, 1995, pp. 373–384.

[3] V. Bruyère, C. De Felice, Any lifting of a trace coding is a word coding, *Inform. and Comput.* **130**(2) (1996) 183–193.

[4] V. Bruyère, C. De Felice, On the existence of codings between trace monoids, *J. Autom. Lang. Comb.* **4**(2) (1999) 87–100.

[5] V. Bruyère, C. De Felice, G. Guaiana, On some decision problems for trace codings, *Theoret. Comput. Sci.* **148**(2) (1995) 227–260.

[6] V. Diekert, Research topics in the theory of free partially commutative monoids, *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS* **40** (1990) 479–491.

[7] V. Diekert, A. Muscholl, Code problems on traces, in: W. Penczek, A. Szalas (Eds.), *Proc. 21st Internat. Sympos. on Mathematical Foundations of Computer Science (MFCS'96),* Lecture Notes in Comput. Sci., vol. 1113, Springer, Berlin, 1996, pp. 2–17.

[8] V. Diekert, A. Muscholl, K. Reinhardt, On codings of traces, in: E.W. Mayr, C. Puech (Eds.), *Proc. 12th Annu. Sympos. on Theoretical Aspects of Computer Science (STACS'95),* Lecture Notes in Comput. Sci., vol. 900, Springer, Berlin, 1995, pp. 385–396.

[9] H.J. Hoogeboom, A. Muscholl, The code problem for traces — improving the boundaries, *Theoret. Comput. Sci.* **172** (1997) 309–321.

[10] G. Hotz, V. Claus, *Automatentheorie und Formale Sprachen III,* Bibliographisches Institut, Mannheim, 1972.

[11] M. Kunc, The trace coding problem is undecidable, in: F. Orejas, P.G. Spirakis, J. van Leeuwen (Eds.), *Proc. 28th Internat. Colloq. on Automata, Languages and Programming (ICALP'01),* Lecture Notes in Comput. Sci., vol. 2076, Springer, Berlin, 2001, pp. 603–614.

[12] M. Kunc, Undecidability of the trace coding problem and some decidable cases, submitted.

[13] Y. Matiyasevich, Mots et codes: cas décidables et indécidables du problème du codage pour les monoïdes partiellement commutatifs, *Quadrature* **27** (1997) 23–33.

[14] A. Mazurkiewicz, Concurrent program schemes and their interpretations, DAIMI Rep. PB 78, Aarhus Univ., 1977.

[15] A. Muscholl, *Decision and complexity issues on concurrent systems,* Habilitationsschrift, Stuttgart Univ., 1999.

[16] E. Ochmański, On morphisms of trace monoids, in: R. Cori, M. Wirsing (Eds.), *Proc. 5th Annu. Sympos. on Theoretical Aspects of Computer Science (STACS'88),* Lecture Notes in Comput. Sci., vol. 294, Springer, Berlin, 1988, pp. 346–355.

Department of Mathematics, Masaryk University
Janáčkovo nám. 2a, 662 95 Brno, Czech Republic
e-mail: `kunc@math.muni.cz`
www: `http://www.math.muni.cz/~kunc`