Code Problems on Traces

Volker Diekert Anca Muscholl*

Universität Stuttgart, Institut für Informatik
Breitwiesenstr. 20-22, D-70565 Stuttgart

Abstract. The topic of codes in the framework of trace monoids leads to interesting and challenging decision problems of combinatorial flavour. We give an overview of the current state of some basic questions in this field. Among these, we consider the existence problem for strong codings, clique-preserving morphisms and the unique decipherability problem (code problem).

1 Introduction

Free partially commutative monoids [7] offer a mathematically sound framework for modelling and analyzing concurrent systems. This was made popular by the work of Mazurkiewicz. He investigated originally the behaviour of safe 1-labelled Petri nets [17] and the computer science community quickly recognized the importance of this work. The basic concept is to consider a system as a finite set of actions Σ, together with a fixed symmetric independence relation $I \subseteq \Sigma \times \Sigma$, denoting pairs of actions which can be scheduled in parallel. In the setting defined by a pair (Σ, I) we identify sequential observations (i.e., strings over Σ) modulo identities $ab = ba$ for $(a, b) \in I$. This yields the partial commutation and the resulting monoid has been called *trace monoid* by Mazurkiewicz. Trace monoids have been successfully considered in classical theories like formal languages, automata and logic. Some challenging decision problems for trace monoids are still open. One of the fundamental open problems is whether or not there is an algorithm to decide the solvability of equations involving constants. For the word case of free monoids the positive answer is known to due a famous result of Makanin [15]. The solution of an equation is a homomorphism. We think that a better understanding of homomorphisms will help to attack a possible extension of Makanin's result, and our contribution is restricted to some questions about homomorphisms. The surprising fact is that even basic questions on codings (injective homomorphisms) between trace monoids are still open. We focus on two problems. First, given two trace monoids $\mathbb{M}_1, \mathbb{M}_2$, does an injective homomorphism $h\colon \mathbb{M}_1 \to \mathbb{M}_2$ exist? Second, given a homomorphism $h\colon \mathbb{M}_1 \to \mathbb{M}_2$, is h injective, i.e., a coding? For both questions only partial answers are known. The first question has a positive answer for strong codings, see Sect. 3 below.

* This research has been supported in part by the French-German research programme PROCOPE.

The second question is known to be undecidable, even if the first monoid is free, as soon as (Σ_2, I_2) contains a cycle on four vertices as induced subgraph [14, 8]. It is known to be decidable when the independence relation (Σ_2, I_2) does not contain any induced subgraph isomorphic to the path P_4 or the cycle C_4 on four vertices, [1]. Recently it has been shown that for (Σ_2, I_2) being equal to the P_4, the code problem is decidable. However, the question of the precise borderline for decidability is still open. We present in Sect. 4 the solution to the code problem for P_4, which has been exhibited independently in [13, 16]. Another natural instance of the question whether a given homomorphism $h\colon \mathbb{M}_1 \to \mathbb{M}_2$ is a coding or not is given when h is described by morphisms between independence alphabets. This special case arises when actions in a system are refined in such a way that the refined action is a product of independent elements which therefore can be performed in parallel. However even for such a restricted class of homomorphisms the injectivity problem turns out again to be undecidable, see Sect. 5.

2 Notations and Preliminaries

A dependence alphabet is a pair (Σ, D), where Σ is a finite alphabet and $D \subseteq \Sigma \times \Sigma$ is a reflexive and symmetric relation, called dependence relation. The complement $I = (\Sigma \times \Sigma) \setminus D$ is called independence relation; it is irreflexive and symmetric. The pair (Σ, I) is denoted independence alphabet. We view both (Σ, D) and (Σ, I) as undirected graphs. The difference is that (Σ, D) has self-loops.
Given an independence alphabet (Σ, I) (or a dependence alphabet (Σ, D) resp.) we associate the trace monoid $\mathbb{M}(\Sigma, I)$. This is the quotient monoid $\Sigma^*/\equiv_I$, where $\equiv_I$ denotes the congruence generated by the set $\{uabv = ubav \mid (a,b) \in I,\ u, v \in \Sigma^*\}$; an element $t \in \mathbb{M}(\Sigma, I)$ is called a trace, the length $|t|$ of a trace t is given by the length of any representing word. By $\mathrm{alph}(t)$ we denote the alphabet of a trace t, which is the set of letters occurring in t. The initial alphabet of t is the set $in(t) = \{x \in \Sigma \mid \exists t' \in \mathbb{M}(\Sigma, I) :\ t = xt'\}$. By 1 we denote both the empty word and the empty trace. Traces $s, t \in \mathbb{M}(\Sigma, I)$ are called independent, if $\mathrm{alph}(s) \times \mathrm{alph}(t) \subseteq I$. We simply write $(s,t) \in I$ in this case. A trace $t \neq 1$ is called a *root*, if $t = u^n$ implies $n = 1$, for every $u \in \mathbb{M}(\Sigma, I)$. A trace t is called connected, if $\mathrm{alph}(t)$ induces a connected subgraph of the dependence alphabet (Σ, D).

3 Existence problem for codings

A homomorphism $h\colon \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ is given by a mapping $h\colon \Sigma \to \mathbb{M}(\Sigma', I')$ such that $h(a)h(b) = h(b)h(a)$ for all $(a,b) \in I$. The homomorphism is called a *coding,* if it is injective. Representing each trace $h(a)$ by a word $\hat{h}(a) \in \Sigma'^*$ we obtain a homomorphism between free monoids $\hat{h}\colon \Sigma^* \to \Sigma'^*$. It may happen that the lifting $\hat{h}$ is injective, but h is not. (A trivial example is $\hat{h} = \mathrm{id}_{\Sigma'}$ with $I' \neq \emptyset$.) But if we know that $\hat{h}$ is not injective, then h

cannot be injective. This surprising fact has been recently announced in [5]: If $h\colon \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ is a coding, then every lifting $\hat{h}\colon \Sigma^* \to \Sigma'^*$ is a word coding. The result is far from being trivial and gives some flavour about the unexpected behaviour of homomorphisms between trace monoids. It is completely open whether the existence problem for codings between trace monoids is decidable. Only some few results and conjectures have been established [4, 12]. The situation changes if we restrict our attention to a naturally arising subclass of homomorphisms. The existence problem of strong codings has a nice graph characterization being discussed in this section.

Definition 1. [6] A homomorphism $h\colon \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ is called a *strong* homomorphism, if $(h(a), h(b)) \in I'$ holds for all $(a, b) \in I$. A *strong* coding is a strong homomorphism being injective.

Hence, strong homomorphisms map independent letters to independent traces. The perhaps most prominent example of a strong coding is given by the Projection Lemma [9, 10]. It can be rephrased as follows.

Example 1. Let $\Sigma' \subseteq \Sigma$ be a subalphabet such that $\Sigma' \times \Sigma' \subseteq D$. The canonical projection $\pi_{\Sigma'}\colon \mathbb{M}(\Sigma, I) \to \Sigma'^*$ is the homomorphism induced by setting $\pi_{\Sigma'}(a) = a$ for $a \in \Sigma'$ and $\pi_{\Sigma'}(a) = 1$ for $a \in \Sigma \setminus \Sigma'$. Consider (Σ, D) written as a union of cliques, i.e., $(\Sigma, D) = (\bigcup_{i=1}^{k} \Sigma_i, \bigcup_{i=1}^{k} \Sigma_i \times \Sigma_i)$. Then the canonical homomorphism

$$\pi\colon \mathbb{M}(\Sigma, I) \to \prod_{i=1}^{k} \Sigma_i^*, \qquad t \mapsto (\pi_{\Sigma_i}(t))_{1 \le i \le k}$$

is a strong coding.

The Projection Lemma leads to the upper bound in the next proposition. To derive the lower bound is left as an exercise to the reader.

Proposition 2. *Let (Σ, D) be a dependence alphabet and $|\Sigma_i| \geq 2$ for $i = 1, \ldots, k$. Then there exists a strong coding $h\colon \mathbb{M}(\Sigma, I) \to \prod_{i=1}^{k} \Sigma_i^*$ if and only if (Σ, D) allows a covering by k cliques. In particular, deciding the existence of strong codings into a k-fold direct product of free monoids is NP-complete.*

The following example suggests some of the differences between the existence problem for codings resp. strong codings.

Example 2. Let $(\Sigma, D) = C_n$ denote the cycle on $n > 3$ vertices, i.e., $\Sigma = \{a_1, \ldots, a_n\}$ and D is defined by the edges $\{a_m a_{m+1} \mid 1 \leq m \leq n\}$ (with addition modulo n). By the proposition above there exists a strong coding from $\mathbb{M}(\Sigma, I)$ into $\prod_{i=1}^{k} \Sigma_i^*$ (where $|\Sigma_i| \geq 2$ for all i) if and only if $k \geq n$.
If we consider codings instead of strong codings we obtain $k = n - 1$ as an upper bound. Let $h\colon \mathbb{M}(\Sigma, I) \to \prod_{i=1}^{n-1} \Sigma_i^*$ be given by $h(a_m) = x_{m-1} y_m x_{m+1}$ for $m < n$ and $h(a_n) = x_{n-1} x_1$ (where by convention $x_0 = x_{n-1}$ and $x_n = x_1$). It

is straightforward to check that h is a homomorphism. Suppose now $h(u) = h(v)$ for some u, v and let $|u|+|v|$ be minimal. The minimality implies $in(u) \cap in(v) = \emptyset$. Since $u \neq 1 \neq v$ we may assume by symmetry that for some $1 \leq m < n$ we have $a_m \in in(u) \setminus in(v)$. Then $y_m \in in(h(u))$. This implies $a_m \in \mathrm{alph}(v)$ and even $a_m \in in(v)$ (otherwise $x_m \in in(h(v))$). Hence a contradiction.
It can be shown that $k \geq n-1$ is also necessary for the existence of a coding for $n > 4$. On the other hand, $\mathbb{M}(\Sigma, I)$ with $(\Sigma, D) = C_4$ can be already embedded in the direct product of two free monoids, [4, 12].

No difference between the existence of codings and strong codings occurs however if the left-hand side monoid is free commutative (see also [18]). It also yields a lower bound for the complexity in Cor. 4 and Cor. 7 below.

Proposition 3. *Let (Σ, D) be a dependence alphabet and $k \geq 1$. The following assertions are equivalent:*

i) The dependence alphabet (Σ, D) contains an independent set of size k.
ii) There exists a strong coding $h\colon \mathbb{N}^k \to \mathbb{M}(\Sigma, I)$.
iii) There exists a coding $h\colon \mathbb{N}^k \to \mathbb{M}(\Sigma, I)$.

Corollary 4. *Given (Σ, D) and k, it is NP-complete to decide whether there exists a (strong) coding of $\mathbb{N}^k$ into $\mathbb{M}(\Sigma, I)$. Therefore, the problem whether there exists a (strong) coding between two given trace monoids is (at least) NP-hard.*

Strong homomorphisms are tightly connected to morphisms of dependence and independence alphabets, which we now define.

Definition 5. Let (V', E'), (V, E) be undirected graphs (possibly with self-loops), $H \subseteq V' \times V$ be a relation between vertices, and for $a' \in V'$ let $H(a')$ denote the set $\{a \in V \mid (a', a) \in H\}$. The relation H is called a *relational morphism*, if $(a', b') \in E'$ implies $H(a') \times H(b') \subseteq E$ for all $a', b' \in V'$.

Theorem 6 [12]. *Let (Σ, D) and (Σ', D') be dependence alphabets. The following assertions are equivalent:*

i) There exists a strong coding $h\colon \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$.
ii) There exists a relational morphism $H\colon (\Sigma', D') \to (\Sigma, D)$ of dependence alphabets such that for all $a \in \Sigma$ there exists $a' \in \Sigma'$ with $a \in H(a')$, and for all $(a, b) \in D$, $a \neq b$ there exists $(a', b') \in D'$, $a' \neq b'$, with $(a, b) \in H(a') \times H(b')$.

Furthermore there are effective constructions between h and H such that $H = \{(a', a) \in \Sigma' \times \Sigma \mid a' \in \mathrm{alph}(h(a))\}$.

Corollary 7. *The following problem is NP-complete:*
Instance: Independence alphabets (Σ, I), (Σ', I').
Question: Does there exist a strong coding from $\mathbb{M}(\Sigma, I)$ into $\mathbb{M}(\Sigma', I')$?

4 Code problem

Rational languages in trace monoids (built over finite sets using concatenation, union and Kleene star) strictly contain the family of recognizable languages and do not form a Boolean algebra, in general. More precisely, the family of rational trace languages over $\mathbb{M}(\Sigma, I)$ is a Boolean algebra if and only if the independence relation I is transitive [20]. This fact leads to nontrivial decision problems for rational trace languages, see [3, 1, 20]. It is known e.g. that the question whether the intersection of two rational trace languages is empty or not is undecidable, in general. More precisely, the intersection problem is undecidable if and only if the graph associated to the independence relation contains as induced subgraph either a cycle or a path on four vertices [1]. It is therefore decidable exactly for transitive forests [22].
We ask in this section a closely related question, namely whether a finite set X is a code, i.e., whether X freely generates X^+. More generally, we will consider the notion of trace code.

Definition 8. For a set $X \subseteq \mathbb{M}(\Sigma, I)$ let Σ_X be an alphabet being in bijection ψ with the set X. Define $I_X \subseteq \Sigma_X \times \Sigma_X$ by $(x, y) \in I_X$ if $x \neq y$ and $\psi(x)\psi(y) = \psi(y)\psi(x)$.
Then X is called a *trace code*, if the induced homomorphism $\psi\colon \mathbb{M}(\Sigma_X, I_X) \to \mathbb{M}(\Sigma, I)$ is a coding.

Definition 9. The *(trace) code* problem for (Σ, I) is to decide on the instance of a finite set $X \subseteq \mathbb{M}(\Sigma, I)$ whether or not it is a (trace) code.

The trace code problem can be reduced to the intersection problem of rational trace languages. To see this, assume that X is not a trace code, then we find two traces $u, v \in \mathbb{M}(\Sigma_X, I_X), u \neq v$ such that $\psi(u) = \psi(v)$. For u, v with $|u| + |v|$ minimal we obtain some $x \in \Sigma_X$ with $x \in \mathit{in}(u)$ and either $\mathrm{alph}(v) \subseteq I_X(x)$ or $v = v_1 y v_2$, for some $y \neq x$, $(x, y) \notin I_X$ and $\mathrm{alph}(v_1) \subseteq I_X(x)$ ($I_X(x)$ denoting the set of letters from Σ_X independent of x). Hence, X is not a trace code if and only if for some $x \in \Sigma_X$ the following holds:

$$\psi(x)X^* \cap \left(\psi(I_X(x)^*) \cup \psi(I_X(x)^*)\psi(D_X(x) \setminus x)X^*\right) \neq \emptyset\,.$$

Since rational sets are closed under homomorphisms, the claim immediately follows. Hence, we obtain the following

Proposition 10. *i) If the trace code problem is decidable for (Σ, I), then the code problem is decidable for (Σ, I).*
ii) Both problems are decidable if the independence relation is a transitive forest [1].
iii) Both problems are undecidable as soon as the independence relation I contains C_4 as induced subgraph [14, 8] (equivalently, $\mathbb{M}(\Sigma, I)$ contains a submonoid isomorphic to $\{a, b\}^* \times \{c, d\}^*$).

The precise borderline for the decidability of the (trace) code problem is currently open. A temptive idea would consist in conjecturing undecidability for independence alphabets containing P_4 as induced subgraph, since this would settle the problem. However, the decidability of the (trace) code problem for P_4 has been recently established, [13, 16]. Furthermore, not every independence alphabet which does not contain C_4 as induced subgraph has a decidable code problem, see [13, 16] for examples.

The code problem between free monoids is equivalent to the emptiness problem for finite automata. Given $X \subseteq A^*$, the states of the associated automaton $\mathcal{A}_X$ are suffixes of words in X. A transition $u \xrightarrow{1} v$ exists if $v \in X^{-1}u \cup u^{-1}X$. With $X^{-1}X \setminus \{1\}$ as initial states and 1 as unique final state one can verify that $1 \in L(\mathcal{A}_X)$ is equivalent to $L(\mathcal{A}_X) \neq \emptyset$, which holds if and only if X is not a code.

For the special case of free monoids, well-known techniques (see [2]) solve the code problem in polynomial time. More precisely, the code problem is complete for NL, the class of languages which can be recognized by nondeterministic Turing machines in logarithmic space (for the hardness see [19]). For the trace monoid $\mathbb{M} = \mathbb{M}(\Sigma, I)$, where the independence (or equivalently, the dependence) alphabet is P_4, the same complexity result holds, c.f. Thm. 13 below. However, we need a more complex device than finite automata.

We assume for the rest of this section that $\mathbb{M} = \mathbb{M}(\Sigma, I)$, where (Σ, I) is equal to P_4:

$$\Sigma = \{a, b, c, d\} \text{ and } I = \{(a, b), (b, a), (b, c), (c, b), (c, d), (d, c)\}.$$

We will use a one-counter automaton. Thus, the automaton can increment, decrement, and test the counter for zero. The value of the counter is an integer. By storing the sign in the finite control, we may also use a pushdown automaton where the pushdown alphabet contains besides the bottom symbol only one single symbol.

We consider below mainly the code problem, i.e., the question whether a finite set $X \subseteq \mathbb{M}$ freely generates X^+. The basic idea will be to guess two different factorizations of an element of X^+ by storing in the counter a certain number of b's resp. c's, while keeping a (finite) synchronization information in the finite control.

The technical lemma below explains how information can be stored in this way. We need some more notations. For traces $w_1, w_2, \ldots$ in $\mathbb{M}$ let $w[i]$ denote $w_1 \cdots w_i$. If u is a factor of v we also write $u \subseteq v$. (This means that $v \in \mathbb{M}u\mathbb{M}$.) For $u_1, \ldots, u_i, v_1, \ldots, v_j \in X$ we call the pair $(u[i], v[j])$ a partial solution, if $u[i]s = v[j]s'$ holds for some $s, s' \in \mathbb{M}$. Using Levi's Lemma [10] we can represent $u[i], v[j]$ as follows.

Fact 11 *Let $(u[i], v[j])$ be a partial solution. Then unique traces $r, \alpha_0, \alpha, \beta_0, \beta \in \mathbb{M}$ exist such that*

- $u[i] = r\alpha_0\alpha$, $v[j] = r\beta_0\beta$
- $(\alpha_0\alpha, \beta_0\beta) \in I$

- $\alpha_0 \subseteq u[i-1]$, $\alpha \subseteq u_i$, $\beta_0 \subseteq v[j-1]$, $\beta \subseteq v_j$.

Lemma 12. *Let $X \subseteq \mathbb{M}$ be given and let $w = u_1 \cdots u_n = v_1 \cdots v_m$ be such that $u_i, v_j \in X$ with $(u_1, \ldots, u_n) \neq (v_1, \ldots, v_m)$. Let $(u[i], v[j])$ be a partial solution with $r, \alpha_0, \alpha, \beta_0, \beta$ defined as above, $(i,j) \neq (n,m)$. Moreover, suppose $|\mathrm{alph}(\alpha_0\beta_0)| \leq 1$.*
Then there exist $s, t \geq 0$ with $s + t > 0$ satisfying the following properties:

i) Either $s = 0$ or $t = 0$. Moreover, if $s = 0$ then $\mathrm{alph}(v_{j+1} \cdots v_{j+t-1})$ is a clique in (Σ, I). Symmetrically, if $t = 0$ then $\mathrm{alph}(u_{i+1} \cdots u_{i+s-1})$ is a clique in (Σ, I).
ii) $u[i+s] = r'\alpha_0'\alpha'$ and $v[j+t] = r'\beta_0'\beta'$ hold for uniquely determined traces $r', \alpha_0', \alpha', \beta_0', \beta'$ satisfying $\alpha_0' \subseteq u[i+s-1]$, $\alpha' \subseteq u_{i+s}$, $\beta_0' \subseteq v[j+t-1]$ and $\beta' \subseteq v_{j+t}$. Moreover, we have $|\mathrm{alph}(\alpha_0'\beta_0')| \leq 1$.

Proof. First observe that $|\mathrm{alph}(\alpha_0\beta_0)| \leq 1$ together with $(\alpha_0, \beta_0) \in I$ implies that either $\alpha_0 = 1$ or $\beta_0 = 1$. Suppose therefore without loss of generality $\beta_0 = 1$. We distinguish the following cases:

i) Let $\alpha_0 = 1$:
Choose $(s,t) = (1,0)$ and let $p, \alpha_0', \alpha', \beta'$ be such that $\alpha u_{i+1} = p\alpha_0'\alpha'$, $\beta = p\beta'$, $(\alpha', \beta') \in I$ and $\alpha_0' \subseteq \alpha$, $\alpha' \subseteq u_{i+1}$. Symmetrically we can choose $(s,t) = (0,1)$ and define $p, \alpha', \beta_0', \beta'$ accordingly.
ii) Let $\beta = 1$:
Choose $(s,t) = (0,1)$. Then we have $\alpha_0\alpha = p\alpha_0'\alpha'$, $v_{j+1} = p\beta'$ for uniquely determined $p, \alpha_0', \alpha,' \beta'$ with $(\alpha_0'\alpha', \beta') \in I$, where $\alpha_0' \subseteq \alpha_0$, $\alpha' \subseteq \alpha$. Note that p, α', β' depend on α, v_{j+1} and on the comparison between $|\alpha_0|$ and a value bounded by α, v_{j+1} (thus bounded by X).
iii) Let $\alpha_0, \beta \neq 1$ and $|\mathrm{alph}(\alpha_0\alpha)| = 1$:
Choose $(s,t) = (1,0)$. Then we have $\alpha_0\alpha u_{i+1} = p\alpha_0'\alpha'$ and $\beta = p\beta'$ for uniquely determined $p, \alpha_0', \alpha', \beta'$ with $(\alpha_0'\alpha', \beta') \in I$, where $\alpha_0' \subseteq \alpha_0\alpha$ and $\alpha' \subseteq u_{i+1}$. Due to $(\alpha_0\alpha, \beta) \in I$ we have $p \subseteq u_{i+1}$, hence $\alpha_0' = \alpha_0\alpha$. Moreover, α' and β' can be directly computed from u_{i+1}, β.
iv) Finally let $|\mathrm{alph}(\alpha_0\alpha)| > 1$ and $\alpha_0, \beta \neq 1$:
We know in this case that $\mathrm{alph}(\alpha_0\alpha)$ is a clique of the dependence relation D. In fact, either $\mathrm{alph}(\alpha_0\alpha) \subseteq \{a, c\}$ or $\mathrm{alph}(\alpha_0\alpha) \subseteq \{b, d\}$. Let e.g. $\alpha_0\alpha \in b^+d\{b,d\}^*$ and consider the least $t > 0$ such that $\mathrm{alph}(v_{j+t}) \cap (D(b) \setminus \{b\}) \neq \emptyset$, i.e., $d \in \mathrm{alph}(v_{j+t})$. Clearly, every $e \in \mathrm{alph}(\beta v_{j+1} \cdots v_{j+t-1})$ satisfies $(e,b) \in I$ or $e = b$. Moreover, if $e \neq b$ then $(e,d) \in I$. Otherwise, there would exist an edge from an e-labelled vertex x in $v_{j+1} \cdots v_{j+t-1}$ to a d-labelled vertex y in $v_{j+t} \cdots v_m$, whereas y precedes x in $\alpha_0\alpha u_{i+1} \cdots u_n$. Therefore, $e = c$ and $\mathrm{alph}(\beta v_{j+1} \cdots v_{j+t-1}) \subseteq \{b, c\}$.
Let $p, \alpha_0', \alpha', \beta_0', \beta'$ be the unique traces satisfying $(\alpha_0'\alpha', \beta_0'\beta') \in I$ and $\alpha_0\alpha = p\alpha_0'\alpha'$ and $\beta v_{j+1} \cdots v_{j+t} = p\beta_0'\beta'$. Moreover, $\alpha_0' \subseteq \alpha_0$, $\alpha' \subseteq \alpha$, $\beta_0' \subseteq \beta v_{j+1} \cdots v_{j+t-1}$ and $\beta' \subseteq v_{j+t}$. Since both $\mathrm{alph}(\alpha) \cap D(b) \neq \emptyset$ and $\mathrm{alph}(v_{j+t}) \cap D(b) \neq \emptyset$ hold, we obtain $\alpha_0 \subseteq p$ and $\pi_b(v_{j+1} \cdots v_{j+t-1}) \subseteq p$. Hence, $\alpha_0' = 1$ and $\beta_0' \subseteq \beta\pi_c(v_{j+1} \cdots v_{j+t-1})$. Clearly we have $\beta \in c^+$ due to $(\alpha_0\alpha, \beta) \in I$, and the claim is satisfied.

Remark. 1. Note that Lem. 12 still holds if (Σ, I) contains no triangle and no induced C_4. Moreover, for $(\Sigma, I) = P_4$ we note that the additional assumption $\alpha_0\beta_0 \in b^* \cup c^*$ yields in Lem. 12 $\alpha_0'\beta_0' \in b^* \cup c^*$, too. It suffices to choose in the case $\alpha_0 = 1$: $(s,t) = (1,0)$, if $\alpha \in b^* \cup c^*$, resp. $(s,t) = (0,1)$, if $\beta \in b^* \cup c^*$.

2. In order to be a code, X may contain at most one element of the form $b^k c^l$, since any two such traces commute.

3. Let us take a closer look at the last case in the previous proof. Assume $\alpha_0 = b^q$ and $\beta = c^r$ for some $q \geq 0$, $r > 0$.
   Supposing that $X \cap b^*c^* = \{b^k c^l\}$ the following equations hold:

$$b^q \alpha = p\alpha' \,, \tag{1}$$

$$c^r \binom{b^k}{c^l}^{t-1} v_{j+t} = p\beta_0'\beta'. \tag{2}$$

   Note that q and $k(t-1)$ differ by a value depending on α and v_{j+t}, only. Hence, $k(t-1)$ is determined by q and a value depending on α and v_{j+t}, only (thus bounded by X). Moreover, due to $|\alpha|_c = 0$ we have $\beta_0' = c^{r+l(t-1)}$. Finally, with $p = b^q p'$ for some p' we have $\alpha = p'\alpha'$, $b^{k(t-1)} v_{j+t} = b^q p' \beta'$ and we see that p', α', β' can be computed using α and v_{j+t}, only.

Let $(u[i], v[j])$ be a partial solution with $u[i] = r\alpha_0\alpha$, $v[j] = r\beta_0\beta$ as in Lem. 12. A counter automaton can store the integer value $|\alpha_0| - |\beta_0|$ in the counter, whereas α, β are part of the finite control. The initial configurations (α, β), α_0, β_0 will be given by partial solutions $(u[i], v[j])$ of minimal length satisfying $x \in \mathrm{alph}(u_1) \cap \mathrm{alph}(v[j])$ for some $x \in \{a, d\}$ (thus, $\alpha_0\beta_0 \in b^* \cup c^*$). The automaton will accept if $\alpha = \beta = 1$ and the counter is empty. It remains to describe the transition relation corresponding to the one-step transition described in Lem. 12, i.e., from $(u[i], v[j])$ to $(u[i+s], v[j+t])$.
Updating α', β' according to the situations considered in Lem. 12 is not difficult. The only problem arises in the last case considered in Lem. 12, when $b^k c^l \in X$ with $kl \neq 0$ and the counter has to switch from b's to c's (or conversely). Roughly, the value of the counter has to be divided by k and multiplied by l (we may also have to increment/decrement the counter by a bounded value, i.e., a value depending on the finite state). Obviously, we cannot perform this combined operation using a single counter. The solution is to store instead of $|\alpha_0| - |\beta_0|$ the value $(|\alpha_0| - |\beta_0|) \operatorname{div} k$, if $\alpha_0\beta_0 \in b^*$. Of course, we keep $(|\alpha_0| - |\beta_0|) \bmod k$ in the finite control. (Analogously, we store $(|\alpha_0| - |\beta_0|) \operatorname{div} l$ in the counter, if $\alpha_0\beta_0 \in c^*$.)

Remark. In order to be a trace code, X may contain at most two elements x_1, x_2 with $\mathrm{alph}(x_1)$, $\mathrm{alph}(x_2) \subseteq \{b, c\}$. More precisely, if $x_i = b^{k_i} c^{l_i}$ then $k_1 l_2 \neq k_2 l_1$ should hold. The second equation in the previous remark has to be replaced then by

$$c^r \binom{b^{k_1}}{c^{l_1}}^{t_1} \binom{b^{k_2}}{c^{l_2}}^{t_2} v_{j+t} = p\beta_0'\beta', \tag{2}$$

for t_1, t_2 with $t_1 + t_2 = t - 1$. It can be shown in this case that if X is not a trace code, then we may suppose in the equation above that the difference $|t_1 - t_2|$ is bounded by X (i.e., there exists a suitable solution $u_1 \cdots u_n = v_1 \cdots v_m$ with this property). In this case, we use the same method as for the code problem, with $k_1 + k_2$ (resp. $l_1 + l_2$) replacing k (resp. l) and keeping $|t_1 - t_2|$ in the finite control.

Theorem 13 [13, 16]. *Let $\Sigma = \{a, b, c, d\}$ and $\mathbb{M}(\Sigma, I)$ be defined by three equations $ab = ba, bc = cb$, and $cd = dc$. Then the (trace) code problem for the independence alphabet (Σ, I) is decidable in polynomial time, actually it is NL-complete.*

Proof sketch. For the complexity result in the theorem above recall that the hardness is already provided by the case of free monoids (over two letters alphabets) [19]. The code problem is shown to belong to NL by noting e.g. that one can test the existence of two different factorizations over a given $X \subseteq \mathbb{M}$ by using a 2-way multi-head nondeterministic counter automaton. With the notations of Lem. 12 this automaton keeps track of α and β using two heads, which point at the corresponding elements of X in the input; the counter is used as in the proof of Thm. 13, whereas the modulo k, l values are handled by further heads on the input. Since the class of languages accepted by 2-way multi-head nondeterministic counter automata is known to coincide with NL [21], the result follows.

5 Clique-preserving morphisms

From the viewpoint of semantics codings may arise by refinement of actions. Assume we want to refine an action in such a way that it can be distributed to different parallel components. Then we may think of this as a homomorphism where a letter is mapped to a product of independent letters. This idea leads to the following definition, where (throughout this section) the notion of clique is meant w.r.t. independence alphabets.

Definition 14. A clique-preserving morphism of independence alphabets H, $H\colon (\Sigma, I) \to (\Sigma', I')$ is a relation $H \subseteq \Sigma \times \Sigma'$ such that $H(A) = \{\alpha \in \Sigma' \mid (a, \alpha) \in H,\ a \in A\}$ is a clique of (Σ', I') whenever $A \subseteq \Sigma$ is a clique of (Σ, I).

A clique-preserving morphism $H \subseteq \Sigma \times \Sigma'$ yields in a natural way a homomorphism $h\colon \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ by letting $h(a) = \prod_{\alpha \in H(a)} \alpha$ for $a \in \Sigma$. Note that the product is well-defined since $H(a)$ is (by definition) a clique, i.e., a set of commuting elements. Moreover, for $(a, b) \in I$ we have

$$h(ab) = h(ba) = \prod_{\alpha \in H(a) \cup H(b)} \alpha \cdot \prod_{\alpha \in H(a) \cap H(b)} \alpha$$

Remark. A clique-preserving morphism is not necessarily a morphism of undirected graphs as defined in Sect. 3. The reason is that for $(a,b) \in I$ we may have $H(a) \cap H(b) \neq \emptyset$. Therefore the induced homomorphisms of trace monoids are not strong, in general. On the other hand, the strong coding defined in the Projection Lemma (Ex. 1) is clique-preserving, too.

Due to the next proposition we have a decidability result in the case where the left-hand side is free. This positive situation is in major contrast to Thm. 16 below.

Proposition 15. *Let $H \subseteq \Sigma \times \Sigma'$ be a relation such that $H(a)$ is a clique of (Σ', I') for all $a \in \Sigma$. Then the induced homomorphism $h\colon \Sigma^* \to \mathbb{M}(\Sigma', I')$ with $h(a) = \prod_{\alpha \in H(a)} \alpha$ is injective if and only if for all $a, b \in \Sigma$, $a \neq b$ there exists some $(\alpha, \beta) \in D'$, $\alpha \neq \beta$ with $\alpha \in H(a), \beta \in H(b)$.*

The following result has been stated in [12] without proof:

Theorem 16. *Given a clique-preserving morphism of independence alphabets $H\colon (\Sigma, I) \to (\Sigma', I')$, it is undecidable whether the associated homomorphism $h\colon \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$, $h(a) = \prod_{\alpha \in H(a)} \alpha$ for $a \in \Sigma$, is a coding.*

The following proof uses the undecidability of Post's correspondence problem (PCP), stated as follows: given two homomorphisms $f, g\colon \Gamma^* \to \Gamma'^*$, does some $u \in \Gamma^+$ exist with $f(u) = g(u)$? For simplifying our reduction we impose following restrictions on the given PCP instances:

- $1 \leq |f(a)|, |g(a)| \leq 2$ for every $a \in \Gamma$
- There exist $x, y \in \Gamma$, $x \neq y$, such that $f(u) = g(u)$ with $u \in \Gamma^+$ implies $u \in x\Gamma^+ y$; moreover, if the instance (f, g) has a solution, then also one in $x(\Gamma_0\Gamma_0)^*\Gamma_0 y$, where $\Gamma_0 = \Gamma \setminus \{x, y\}$. Finally, for some $\alpha, \beta \in \Gamma'$: $f(x), g(x) \in \alpha\Gamma'^*$, $f(y), g(y) \in \Gamma'^*\beta$ and α, β occur in no $f(a), g(a)$, for $a \notin \{x, y\}$.

One can show that PCP with these additional restrictions remains undecidable. This can be performed e.g. by slightly modifying the PCP pairs obtained in the reduction from the word problem for semi-Thue systems [11] (resp. imposing suitable restrictions on the semi-Thue systems).
For the alphabet Σ_0 defined below we will denote by Σ_0' the alphabet $\{a' \mid a \in \Sigma_0\}$. Let

$$\Sigma_0 = \{a_b, \overline{a}_b \mid a, b \in \Gamma_0\} \cup \{a, \overline{a} \mid a \in \Gamma_0\} \cup \{x_1, x_2, y_1, y_2\} \cup \{\overline{a}_x, y_b \mid a, b \in \Gamma_0\}\,,$$

and $\Sigma_1 = \Sigma_0 \cup \Sigma_0'$. On Σ_1 we have the independence relation $I_1 \subseteq \Sigma_1 \times \Sigma_1$:

$$I_1 = \{(x_1, x_1'), (y_2, y_2'), (x_1', x_1), (y_2', y_2)\}.$$

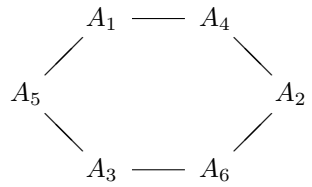
For the second independence alphabet (Σ_2, I_2) let

$$\Sigma_2 = \{A_i, \overline{A}_i \mid a \in \Gamma_0,\, 1 \leq i \leq 6\} \cup \{X_i, Y_i \mid 1 \leq i \leq 6\} \cup \{F, G, \mathfrak{c}, \$\} \cup \Gamma'\,,$$

where we associate to every letter $a \in \Gamma_0$ the capital letters A_i, $\overline{A}_i$ (resp. X_i for x, resp. Y_i for y) for all $1 \leq i \leq 6$. On Σ_2 we define the dependence (independence, resp.) relation $D_2 \subseteq \Sigma_2 \times \Sigma_2$ ($I_2 \subseteq \Sigma_2 \times \Sigma_2$, resp.) as symmetric relations satisfying:

(1) For all $a \in \Gamma$ let $D_2 \cap \{A_i \mid 1 \leq i \leq 6\}^2$ be (with self-loops omitted)

D_2:



respectively, for $a \in \Gamma_0$ and $1 \leq i, j \leq 6$:

$$(\overline{A}_i, \overline{A}_j) \in D_2 \quad \text{iff} \quad (A_i, A_j) \in D_2\,.$$

(2) For $a \neq b$ and i, j let $(A_i, B_j) \in D_2$ and $(\overline{A}_i, \overline{B}_j) \in D_2$.
(3) For all a, b, i, j we define $(A_i, \overline{B}_j) \in I_2$ if and only if

$$\{(i,j),(j,i)\} \cap (\{5,6\} \times \{1,2\} \cup \{4,6\} \times \{1,3\}) \neq \emptyset\,.$$

(4) Let $D_2(\gamma) = \Gamma'$ for all $\gamma \in \Gamma'$. Define further $D_2(F) = D_2(G) = \{F, G\}$, $I_2(\mathbb{C}) = \{X_1, X_2, X_3, F, G\} \cup \Gamma'$ and $I_2(\$) = \{Y_4, Y_5, Y_6, F, G\} \cup \Gamma'$.

We define now a clique-preserving morphism $H\colon (\Sigma_1, I_1) \to (\Sigma_2, I_2)$ based on the given PCP instance $f, g\colon \Gamma^* \to \Gamma'^*$. The associated homomorphism h will simulate the given (word) homomorphisms f, g using the control alphabet $A_i, \overline{A}_i, F, G, \ldots$. More precisely, the control alphabet enforces by synchronization the presence of f- resp. g-images (since $f(a)$ or $g(a)$ may have length 2, we cannot generate them directly by a homomorphism associated to a clique-preserving morphism.) We denote in the following by $h(a)_i$, $i = 1, 2$, $a \in \Gamma$, the ith letter of $h(a)$ (resp. 1, if $|h(a)| < i$), with $h \in \{f, g\}$.
In the definition below we have by convention $k(a)_i \in H(c)$ only if $k(a)_i \neq 1$, where $c \in \Sigma_1$, $a \in \Gamma$, and $k \in \{f, g\}$. However, for simplifying notations we write $f(a)_i$, $g(a)_i$ throughout.

(1) For all $a, b \in \Gamma_0$ let

$$\begin{aligned} H(a_b) &= \{\overline{B}_5, \overline{B}_6, A_1, A_2, f(a)_1, F\} & H(a) &= \{A_3, A_4, f(a)_2, G\}\,, \\ H(a'_b) &= \{\overline{B}_4, \overline{B}_6, A_1, A_3, g(a)_1, F\} & H(a') &= \{A_2, A_5, g(a)_2, G\}\,, \end{aligned}$$

and analogously

$$\begin{aligned} H(\overline{a}_b) &= \{B_5, B_6, \overline{A}_1, \overline{A}_2, f(a)_1, F\} & H(\overline{a}) &= \{\overline{A}_3, \overline{A}_4, f(a)_2, G\}\,, \\ H(\overline{a}'_b) &= \{B_4, B_6, \overline{A}_1, \overline{A}_3, g(a)_1, F\} & H(\overline{a}') &= \{\overline{A}_2, \overline{A}_5, g(a)_2, G\}\,. \end{aligned}$$

(2) Let for x_i, x_i' and $a \in \Gamma_0$:

$$\begin{aligned} H(x_1) &= \{\,\text{¢}, X_1, X_2, f(x)_1, F\} & H(x_2) &= \{X_3, X_4, f(x)_2, G\}\,, \\ H(x_1') &= \{\,\text{¢}, X_1, X_3, g(x)_1\} & H(x_2') &= \{X_2, X_5, g(x)_2\}\,, \\ H(\overline{a}_x) &= \{X_5, X_6, \overline{A}_1, \overline{A}_2, f(a)_1, F\} \\ H(\overline{a}_x') &= \{X_4, X_6, \overline{A}_1, \overline{A}_3, g(a)_1, F\}. \end{aligned}$$

(3) For $b \in \Gamma_0$ let:

$$\begin{aligned} H(y_b) &= \{\overline{B}_5, \overline{B}_6, Y_1, Y_2, f(y)_1, F\} \quad \text{and} \\ H(y_b') &= \{\overline{B}_4, \overline{B}_6, Y_1, Y_3, g(y)_1, F\}\,, \end{aligned}$$

together with

$$\begin{aligned} H(y_1) &= \{Y_3, Y_4, f(y)_2\} & H(y_2) &= \{Y_5, Y_6, \$\}\,, \\ H(y_1') &= \{Y_2, Y_5, g(y)_2, G\} & H(y_2') &= \{Y_4, Y_6, F, \$\}\,. \end{aligned}$$

With $I_1 = \{(x_1, x_1'), (y_2, y_2'), (x_1', x_1), (y_2', y_2)\}$ it is easily seen that H is a clique-preserving morphism of independence alphabets. Let $h\colon \mathbb{M}(\Sigma_1, I_1) \to \mathbb{M}(\Sigma_2, I_2)$ denote the associated homomorphism of trace monoids, $h(a) = \prod_{A \in H(a)} A$, $a \in \Sigma_1$. We defined H in such a way that every pair $(u, v) \in \mathbb{M}(\Sigma_1, I_1)^2$ of minimal length $|u| + |v|$ with $h(u) = h(v)$ and $u \neq v$ satisfies $\mathrm{alph}(u) \times \mathrm{alph}(v) \subseteq \Sigma_0 \times \Sigma_0' \cup \Sigma_0' \times \Sigma_0$.

Proposition 17. *The homomorphism $h\colon \mathbb{M}(\Sigma_1, I_1) \to \mathbb{M}(\Sigma_2, I_2)$ associated to the clique-preserving morphism H is not injective if and only if the PCP instance (f, g) has a solution.*

Proof. First, let us consider a solution $u \in x\Gamma_0^n y$ for (f, g) with n odd. We have $f(u) = g(u)$ with $u = x\, a(1) \cdots a(n)\, y$, $a(i) \in \Gamma_0$. For a trace $t = [a_1 \cdots a_n]$, $a_i \in \Sigma_0$, we will denote by t' the trace $[a_1' \cdots a_n']$. It is easy to see that with

$$z = [x_1 x_2\, \overline{a(1)}_x \overline{a(1)}\, a(2)_{a(1)} a(2) \cdots \overline{a(n)}_{a(n-1)} \overline{a(n)}\, y_{a(n)}\, y_1 y_2]$$

we obtain

$$h(z) = h(z') =$$

$$[\,\text{¢}\, X_1 \cdots X_6\, \overline{A(1)}_1 \cdots \overline{A(1)}_6 \ldots\ldots \overline{A(n)}_1 \cdots \overline{A(n)}_6\, Y_1 \cdots Y_6\, \$]$$

$$f(u)$$

$$F(GF)^{n+1}$$

(For the vector-like representation above we use the fact that $f(u)$, $F(GF)^{n+1}$ commute pairwise, resp. with the first component.)
For the converse, let us notice two properties of the clique-preserving morphism H used throughout the proof.

Fact 18 *1. $Zh(c) = h(c)Z$ for $c \in \Sigma_1$ and $Z \in \{A_i, \overline{A}_i \mid 1 \leq i \leq 6, a \in \Gamma\} \setminus \{X_2, X_3, Y_4, Y_5\}$ is equivalent with $(c, Z) \in H$, i.e., $Z \in \mathrm{alph}(h(c))$. For $Z \in \{X_2, X_3, Y_4, Y_5\}$ we have $Zh(c) = h(c)Z$ if and only if $(c, Z) \in H$ or $(c, Z) \in \{(x'_1, X_2), (x_1, X_3), (y_2, Y_4), (y'_2, Y_5)\}$.*

2. For all $Z \in \{A_i, \overline{A}_i \mid 2 \leq i \leq 5, a \in \Gamma_0\} \cup \{X_4, Y_2\}$ we have two distinct letters $c_1, c_2 \in \Sigma_1$ satisfying $(c_i, Z) \in H$. Additionally, either $F \in H(c_1)$ and $G \in H(c_2)$, or $F \in H(c_2)$ and $G \in H(c_1)$ holds. This property will allow to determine c_1 resp. c_2 in a unique way, using $(F, G) \in D_2$.

We denote in the following for $u \in \mathbb{M}(\Sigma_1, I_1)$ by $h^*(u)$ the projection of $h(u)$ on $\Sigma_2 \setminus \Gamma'$, i.e., we consider the control letters, only.

Assume now that $h\colon \mathbb{M}(\Sigma_1, I_1) \to \mathbb{M}(\Sigma_2, I_2)$ is not injective, and let $u, v \in \mathbb{M}(\Sigma_1, I_1)$, $u \neq v$, be of minimal length $|u| + |v|$ with $h(u) = h(v)$. Note that for every $(a, b) \in D_1 \setminus \mathrm{id}_{\Sigma_1}$ one has $(H(a) \times H(b)) \cap (D_2 \setminus \mathrm{id}_{\Sigma_2}) \neq \emptyset$. Moreover, by the minimality of $|u| + |v|$ the initial alphabets of u resp. v are pairwise independent, i.e., $in(u) \times in(v) \subseteq I_1$. Hence, two cases are possible: either $in(u) = \{x_1\}$ and $in(v) = \{x'_1\}$, or $in(u) = \{y_2\}$ and $in(v) = \{y'_2\}$.

We first consider the case where $in(u) = \{x_1\}$, $in(v) = \{x'_1\}$ and suppose $u = x_1^n u_1$, $v = {x'_1}^m v_1$ for some $u_1, v_1 \in \mathbb{M}(\Sigma_1, I_1) \setminus 1$ with $in(u_1) \neq \{x_1\}$, $in(v_1) \neq \{x'_1\}$, and $n, m \geq 1$. Hence, we have

$$h^*(u) = \underset{X_2^n}{\overset{X_1^n}{\not\subset^n}} F^n h^*(u_1) = \underset{X_3^m}{\overset{X_1^m}{\not\subset^m}} h^*(v_1) = h^*(v)\,.$$

Due to the condition imposed on the initial alphabets of u_1, v_1 we immediately follow $n = m$. Moreover, recalling Fact 18(1) and the minimality of $|u| + |v|$, we obtain $in(u_1) = \{x_2\}$ and $in(v_1) = \{x'_2\}$. Actually, since $\{X_4, X_5\} \subseteq D_2(\not\subset)$ we may observe that $u_1 = x_2^n u_2$ and $v_1 = {x'_2}^n v_2$ holds for some u_2, v_2 (neither $u_1 \in x_2^i x'_1 \mathbb{M}(\Sigma_1, I_1)$ nor $v_1 \in {x'_2}^i x_1 \mathbb{M}(\Sigma_1, I_1)$ can lead to a solution, if $i < n$).

Therefore, we obtain the following situation (we omit in the representations below dependence edges from $\not\subset$):

$$h^*(u) = \underset{F^n G^n}{\overset{X_1^n X_2^n X_3^n X_4^n}{\not\subset^n}} h^*(u_2) = \not\subset^n X_1^n X_3^n X_2^n X_5^n h^*(v_2) = h^*(v)\,.$$

Fact 18(2) applied to X_4 yields now directly $v_2 = {\overline{a}'_x}^n v_3$, for some v_3 and $a \in \Gamma_0$. Since $X_5 \in in(h(u_2))$ we have $\{\bar{a}_x, x'_2\} \cap in(u_2) \neq \emptyset$. Due to $(\overline{a}'_x, \overline{A}_1) \in H$ and $(\overline{A}_1, X_2) \in D_2$, we obtain $u_2 = \overline{a}_x^n u_3$, for some u_3. This yields the partial solution:

$$h(u) \;=\; \begin{array}{ccc} \cent^n & X_1^n \cdots X_6^n & \overline{A}_1^n \overline{A}_2^n \\ & F^n\, G^n & F^n \\ & (f(x)_1)^n\,(f(x)_2)^n & (f(a)_1)^n \end{array} \qquad h(u_3) \;=$$

$$h(v) \;=\; \begin{array}{ccc} \cent^n & X_1^n \cdots X_6^n & \overline{A}_1^n \overline{A}_3^n \\ & F^n & \\ & (g(x)_1)^n\,(g(x)_2)^n & (g(a)_1)^n \end{array} \qquad h(v_3)\,.$$

More generally, let us assume

$$h^*(u) \;=\; W \begin{array}{cc} \overline{A}_1^n & \overline{A}_2^n \\ G^n & F^n \end{array} h^*(u_1) \;=\; W \begin{array}{cc} \overline{A}_1^n & \overline{A}_3^n \end{array} h^*(v_1) \;=\; h^*(v)\,,$$

for $a \notin \{x, y\}$, $W \in \mathbb{M}(\Sigma_2, I_2)$, $u_1, v_1 \in \mathbb{M}(\Sigma_1, I_1)$.
Due to $\overline{A}_2 \in \mathit{in}(h(v_1))$, together with Fact 18(2), we immediately obtain $v_1 = \overline{a}'^n v_2$ for some v_2, hence $h^*(v) = W \begin{array}{cccc} \overline{A}_1^n & \overline{A}_3^n & \overline{A}_2^n & \overline{A}_5^n \\ & & G^n & \end{array} h^*(v_2)$. With $\overline{A}_3 \in \mathit{in}(h(u_1))$ we need a letter $c \in \Sigma$ satisfying $(c, \overline{A}_3) \in H$ and $\mathrm{alph}(h(c)) \cap D(\overline{A}_5) \subseteq \{\overline{A}_3\}$; hence, $c = \overline{a}$ and $u_1 = \overline{a}^n u_2$, for some u_2. This yields $h^*(u) = W \begin{array}{cccc} \overline{A}_1^n & \overline{A}_2^n & \overline{A}_3^n & \overline{A}_4^n \\ G^n & F^n & G^n & \end{array} h^*(u_2)$. Finally, $\overline{A}_4 \in \mathit{in}(h(v_2))$, together with Fact 18(2) yields $v_2 = b_a'^n v_3$ and

$$h^*(v) = W \begin{array}{cc} \overline{A}_1^n \cdots \overline{A}_6^n & B_1^n B_3^n \\ G^n F^n & \end{array} h^*(v_3)\,,$$

for some $b \neq x$ and $v_3 \in \mathbb{M}(\Sigma_1, I_1)$; on the other hand, the condition $\overline{A}_5 \in \mathit{in}(h(u_2))$ requires a letter $c \in \Sigma_1$ with $(c, \overline{A}_5) \in H$ and $B_1 h(c) = h(c) B_1$, hence $c = b_a$ and $u_2 = b_a^n u_3$, with $h^*(u) = W \begin{array}{cc} \overline{A}_1^n \cdots \overline{A}_6^n & B_1^n B_2^n \\ G^n F^n G^n F^n & \end{array} h^*(u_3)$, for some $u_3 \in \mathbb{M}(\Sigma_1, I_1)$.
The case $h^*(u) = W \begin{array}{cc} A_1^n & A_2^n \\ G^n & F^n \end{array} h^*(u_1) = h^*(v) = W\, A_1^n A_3^n h^*(v_1)$, $a \neq y$, is symmetric. Therefore, we suppose now $a = y$. We observe that the same arguments used above for $\overline{A}_2$, $\overline{A}_3$ hold (dually) for Y_2, Y_3 (in particular, by Fact 18(2)). Hence,

$$h^*(u) = W \begin{array}{cc} Y_1^n Y_2^n Y_3^n Y_4^n & \\ G^n F^n & \end{array} h^*(u_2) = W \begin{array}{cc} Y_1^n Y_3^n Y_2^n Y_5^n & \\ G^n & \end{array} h^*(v_2) = h^*(v)\,.$$

Thus we obtained $w = [x_1^n x_2^n\, \overline{a_1}_x^n \overline{a_1}^n\, a_{2a_1}^n a_2^n \cdots \overline{a_m}_{a_{m-1}}^n \overline{a_m}^n\, y_{a_m}^n\, y_1^n]$, $a_i \in \Gamma_0$, $n, m \geq 1$, such that $u = wu_2$ and $v = w'v_2$ for some u_2, v_2 (recall that w' denotes the Σ_0'-copy of $w \in \Sigma_0^*$). Recall that $f(y), g(y) \in \Gamma'^* \beta$ and β occurs in no $f(a), g(a)$, for $a \notin \{x, y\}$. Hence, we obtained

$$(f(x)_1)^n (f(x)_2)^n (f(a_1)_1)^n (f(a_1)_2)^n \cdots (f(a_m)_1)^n (f(a_m)_2)^n (f(y)_1)^n (f(y)_2)^n =$$
$$(g(x)_1)^n (g(x)_2)^n (g(a_1)_1)^n (g(a_1)_2)^n \cdots (g(a_m)_1)^n (g(a_m)_2)^n (g(y)_1)^n (g(y)_2)^n\,.$$

It is now easy to check that $z = x\, a_1 \cdots a_m\, y$ satisfies $f(z) = g(z)$, hence z is a solution for the PCP instance (f, g).

It remains to consider a pair $(u, v) \in \mathbb{M}(\Sigma, I)^2$, $u \neq v$, of minimal length $|u|+|v|$, with $h(u) = h(v)$, which satisfies $in(u) = \{y_2'\}$, $in(v) = \{y_2\}$. Assume $u = {y_2'}^n u_1$ and $v = y_2^m v_1$ with u_1, v_1 such that $in(u_1) \neq \{y_2'\}$, $in(v_1) \neq \{y_2\}$. Again, $m = n$ follows immediately. Hence, we obtain

$$h^*(u) = \$^n \underset{F^n}{Y_6^n Y_4^n}\, h^*(u_1) = \$^n Y_6^n Y_5^n\, h^*(v_1) = h^*(v)\,.$$

This case can now be handled as the first one, replacing $A_1, A_2, A_3, A_4, A_5, A_6$ by $A_6, A_4, A_5,\ A_2, A_3, A_1$ ($\overline{A}_i$ analogously) and interchanging x, y. This concludes the proof.

References

1. IJ. J. Aalbersberg and H. J. Hoogeboom. Characterizations of the decidability of some problems for regular trace languages. *Mathematical Systems Theory*, 22:1–19, 1989.
2. J. Berstel and D. Perrin. *Theory of Codes*. Pure and Applied Mathematics; 117. Academic Press, Orlando, Florida, 1985.
3. A. Bertoni, G. Mauri, and N. Sabadini. Equivalence and membership problems for regular trace languages. In *Proceedings of the 9th International Colloquium on Automata, Languages and Programming (ICALP'82)*, number 140 in Lecture Notes in Computer Science, pages 61–71, Berlin-Heidelberg-New York, 1982. Springer.
4. V. Bruyère and C. De Felice. Trace codings. In E. Mayr and C. Puech, editors, *Proceedings of the 12th Annual Symposium on Theoretical Aspects of Computer Science (STACS'95), 1995*, number 900 in Lecture Notes in Computer Science, pages 373–384, Berlin-Heidelberg-New York, 1995. Springer.
5. V. Bruyère and C. De Felice. Any lifting of a trace coding is a word coding. Submitted for publication, 1996.
6. V. Bruyère, C. De Felice, and G. Guaiana. Coding with traces. In P. Enjalbert, E. Mayr, and K. W. Wagner, editors, *Proceedings of the 11th Annual Symposium on Theoretical Aspects of Computer Science (STACS'94), 1994*, number 775 in Lecture Notes in Computer Science, pages 353–364, Berlin-Heidelberg-New York, 1994. Springer.
7. P. Cartier and D. Foata. *Problèmes combinatoires de commutation et réarrangements*. Number 85 in Lecture Notes in Mathematics. Springer, Berlin-Heidelberg-New York, 1969.
8. M. Chrobak and W. Rytter. Unique decipherability for partially commutative alphabets. *Fundamenta Informaticae*, X:323–336, 1987.
9. M. Clerbout and M. Latteux. Partial commutations and faithful rational transductions. *Theoretical Computer Science*, 34:241–254, 1984.
10. R. Cori and D. Perrin. Automates et commutations partielles. *R.A.I.R.O. — Informatique Théorique et Applications*, 19:21–32, 1985.
11. M. D. Davis and E. J. Weyuker. *Computability, complexity and languages*. Academic Press, New York, 1983.
12. V. Diekert, A. Muscholl, and K. Reinhardt. On codings of traces. In E. Mayr and C. Puech, editors, *Proceedings of the 12th Annual Symposium on Theoretical Aspects of Computer Science (STACS'95), 1995*, number 900 in Lecture Notes in Computer Science, pages 385–396, Berlin-Heidelberg-New York, 1995. Springer.

13. H. J. Hoogeboom and A. Muscholl. The code problem for traces – improving the boundaries. Submitted for publication.
14. G. Hotz and V. Claus. *Automatentheorie und Formale Sprachen, Band III.* Bibliographisches Institut, Mannheim, 1972.
15. G. S. Makanin. The problem of solvability of equations in free semigroups. *Math. USSR Izvestiya*, 21:483–546, 1983.
16. Yu. Matyiasevich. Cas décidables et indécidables du problème du codage pour les monoïdes partialement commutatifs. To appear in Quadrature.
17. A. Mazurkiewicz. Concurrent program schemes and their interpretations. DAIMI Rep. PB 78, Aarhus University, Aarhus, 1977.
18. E. Ochmański. On morphisms of trace monoids. In R. Cori and M. Wirsing, editors, *Proceedings of the 5th Annual Symposium on Theoretical Aspects of Computer Science (STACS'88)*, number 294 in Lecture Notes in Computer Science, pages 346–355, Berlin-Heidelberg-New York, 1988. Springer.
19. W. Rytter. The space complexity of the unique decipherability problem. *Information Processing Letters*, 23:1–3, 1986.
20. J. Sakarovitch. The "last" decision problem for rational trace languages. In I. Simon, editor, *Proceedings of the 1st Latin American Symposium on Theoretical Informatics (LATIN'92)*, number 583 in Lecture Notes in Computer Science, pages 460–473, Berlin-Heidelberg-New York, 1992. Springer.
21. K. Wagner and G. Wechsung. *Computational complexity.* VEB Deutscher Verlag der Wissenschaften, Berlin, 1986.
22. E. S. Wolk. A note on the comparability graph of a tree. *Proc. of the Amer. Math. Soc.*, (16):17–20, 1965.

This article was processed using the LaTeX macro package with LLNCS style