On Codings of Traces

Extended Abstract *

Volker Diekert Anca Muscholl Klaus Reinhardt

Institut für Informatik, Universität Stuttgart
Breitwiesenstr. 20-22, D-70565 Stuttgart
e-mail: {diekert,muscholl,reinhard}@informatik.uni-stuttgart.de

Abstract. The paper solves the main open problem of [BFG94]. We show that given two dependence alphabets (Σ, D) and (Σ', D'), it is decidable whether there exists a strong coding $h : M(\Sigma, D) \longrightarrow M(\Sigma', D')$ between the associated trace monoids. In fact, we show that the problem is NP-complete. (A coding is an injective homomorphism, it is strong if independent letters are mapped to independent traces.) We exhibit an example of trace monoids where a coding between them exists, but no strong coding. The decidability of codings remains open, in general. We have a lower and an upper bound, which show both to be strict. We further discuss encodings of free products of trace monoids and give almost optimal constructions.
In the final section, we state that the coding property is undecidable in a naturally arising class of homomorphisms.

Topics: Formal languages, concurrency.

1 Introduction

The theory of traces has been recognized as an important tool for investigations about concurrent systems, see [DR95]. The origins of the theory go back to the work of Mazurkiewicz (trace theory), Karp/Miller, Keller (parallel program schemata), and Cartier/Foata (combinatorics), see [Maz87, Kel73, CF69].
Traces are of particular interest for a description of concurrent processes, since for many algorithms one can obtain complexity bounds, which are close to the corresponding algorithms on words, e.g. test of equality [AG91], pattern matching [HY92], or word problems [Die89, Die90]. A convenient data structure for these algorithms is a representation of traces as a tuple of words. This is nothing but a coding of a trace monoid in a direct product of free monoids and leads to the general problem to find codings between trace monoids. This question has been raised by Ochmański in [Och88] and reconsidered recently by Bruyère et al. in [BFG94] (see also [BF95] in this volume). A natural class of codings of trace monoids is given by strong codings. These are injective homomorphisms

* This research has been supported by the ESPRIT Basic Research Action No. 6317 ASMICS 2, *Algebraic and Syntactic Methods In Computer Science*.

such that independent letters are mapped to independent traces (similar to a refinement of actions in a concurrent system). The main result of [BFG94] is the decidability of the existence of a strong coding for two large families of trace monoids, but the authors left open the general problem: given two trace monoids $M(\Sigma, D)$ and $M(\Sigma', D')$, is it decidable whether there exists a strong coding of one into the other. The main result of the present paper solves this problem. We give a graph theoretical criterion for the existence of a strong coding thereby showing its NP-completeness, see Thm. 9.

The paper contains several other results. As mentioned above, from the viewpoint of applications, the most important codings are those into a direct product of free monoids. Again, there is a characterization for strong codings, which is directly related to a covering by cliques and which shows the NP-completeness of this restricted problem, too. Much less is known if we consider codings (i.e., injective homomorphisms), instead of strong codings. For certain dependence alphabets we are able to compute the least k for the existence of an encoding of the trace monoid into a k-fold direct product (e.g., the path and the cycle of n vertices). We show that if the dependence alphabet is a cycle of four vertices, then the associated trace monoid has a coding into a direct product of free monoids with two components, whereas at least four are needed for a strong coding. The existence of a coding between trace monoids is NP-hard, but we even do not know its decidability in the restricted case where the second monoid is a direct product of free monoids. We have a lower bound for the minimal number of components we need by the size of a maximal independent set and an upper bound by the number of cliques which are needed to cover all vertices and edges. The exact value remains unknown.

We give an example of a dependence alphabet (which is in fact a cograph), where both the lower and the upper bounds are strict. Finally, we give almost optimal constructions for encoding free products of trace monoids into direct products of free monoids. The results can be used for encoding efficiently trace monoids with a cograph dependence alphabet.

In the final section we reconsider the well-known result on the undecidability of whether a given homomorphism of trace monoids is a coding. The proof [HC72, CR87] uses a reduction of the Post correspondence problem, and it does not apply to restricted classes of homomorphisms. In this paper we start with a so-called strict morphism between independence alphabets. These morphisms induce in a canonical way homomorphisms between the associated trace monoids. (The homomorphism of the Projection Lemma is of this kind.) Thm. 22 states that the coding property for this natural class of homomorphisms is undecidable.

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 (which however will be omitted in pictures). An undirected graph is simply a pair (V, E), where $E \subseteq V \times V$ is a symmetric relation. We use the following basic operations on graphs: complementation $\overline{(V, E)} = (V, (V \times V) \setminus E)$, disjoint union $(V_1, E_1) \dot{\bigcup} (V_2, E_2) = (V_1 \dot{\bigcup} V_2, E_1 \dot{\bigcup} E_2)$, and complex product $(V_1, E_1) * (V_2, E_2) = (V_1 \dot{\bigcup} V_2, E_1 \dot{\bigcup} E_2 \dot{\bigcup} (V_1 \times V_2) \dot{\bigcup} (V_2 \times V_1))$. (The smallest family of graphs containing the one-point graphs which is closed under these operations is the family of cographs, see e.g. [CLB81, CPS85].)
Given a dependence alphabet (Σ, D) (or an independence alphabet (Σ, I) resp.) we associate the trace monoid $M(\Sigma, D)$. This is the quotient monoid $\Sigma^*/\{ab = ba \mid (a, b) \in I\}$; an element $t \in M(\Sigma, D)$ 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 $\mathrm{in}(t) = \{x \in \Sigma \mid \exists t' : t = xt'\}$. By 1 we denote both the empty word and the empty trace. Traces $s, t \in M(\Sigma, D)$ 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. A trace t is called connected, if $\mathrm{alph}(t)$ induces a connected subgraph of the dependence alphabet (Σ, D).
The constructions disjoint union and complex product resp., on dependence alphabets correspond to the direct product and the free product (= direct sum in the category of monoids) resp., for the associated trace monoids. Thus, we have $M((\Sigma_1, D_1) \dot{\bigcup} (\Sigma_2, D_2)) = M(\Sigma_1, D_1) \times M(\Sigma_2, D_2)$ and $M((\Sigma_1, D_1) * (\Sigma_2, D_2)) = M(\Sigma_1, D_1) * M(\Sigma_2, D_2)$. The situation for independence alphabets is dual.
Let $\Sigma' \subseteq \Sigma$ be a subalphabet and $D' = (\Sigma' \times \Sigma') \cap D$ the induced dependence relation. The canonical projection $\pi_{\Sigma'} : M(\Sigma, D) \to M(\Sigma', D')$ is induced by $\pi_{\Sigma'}(a) = a$, if $a \in \Sigma'$ and $\pi_{\Sigma'}(a) = 1$ otherwise. 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 we have the following well-known Projection Lemma:

Proposition 1 [CL85, CP85]. *Let $(\Sigma, D) = (\bigcup_{i=1}^k \Sigma_i, \bigcup_{i=1}^k \Sigma_i \times \Sigma_i)$. Then the canonical homomorphism*

$$\pi : M(\Sigma, D) \to \prod_{i=1}^{k} \Sigma_i^*, \qquad t \mapsto (\pi_{\Sigma_i}(t))_{1 \leq i \leq k}$$

is injective.

In the following a homomorphism $h : M(\Sigma, D) \to M(\Sigma', D')$ is called a *coding*, if it is injective. It is called a *strong* homomorphism, if independent letters are mapped to independent traces, i.e., if $(a, b) \in I$ implies $(h(a), h(b)) \in I'$ for all $a, b \in \Sigma$. Of particular interest are strong codings, a notion which has been introduced in [BFG94].
The next proposition belongs probably to folklore:

Proposition 2. *Let (Σ, D) be a dependence alphabet and $k \geq 1$. The following assertions are equivalent:*

1. *The dependence alphabet (Σ, D) contains an independent set of size k.*
2. *There exists a strong coding $h : \mathbb{N}^k \to M(\Sigma, D)$.*
3. *There exists a coding $h : \mathbb{N}^k \to M(\Sigma, D)$.*

Proof. Since the implications 1) $\Rightarrow$ 2) $\Rightarrow$ 3) are obvious, we show 3) $\Rightarrow$ 1). Let $\{a_1, \ldots, a_k\}$ be a set of generators of $\mathbb{N}^k$ and let $t_i = h(a_i)$, $1 \leq i \leq k$. By a result of Duboc [Dub86] the equations $t_i t_j = t_j t_i$ $(1 \leq i \neq j \leq k)$ yield the existence of pairwise independent, connected roots $x_1, \ldots, x_l$, together with nonnegative integers n_{ij} $(1 \leq i \leq k,\ 1 \leq j \leq l)$ such that $t_i = x_1^{n_{i1}} \cdots x_l^{n_{il}}$ for $1 \leq i \leq k$. The set of roots $\{x_1, \ldots, x_l\}$ generates a commutative submonoid of $M(\Sigma, D)$, and the coding factorizes as $h : \mathbb{N}^k \to \mathbb{N}^l \to M(\Sigma, D)$. With h being injective we have $k \leq l$ by linear algebra. Finally, it suffices to choose some letter $b_i \in \mathrm{alph}(x_i)$, $1 \leq i \leq l$, in order to establish the result.

Corollary 3. *It is NP-complete to decide whether there exists a (strong) coding of $\mathbb{N}^k$ into $M(\Sigma, D)$. Therefore, the problem whether there exists a (strong) coding between two given trace monoids is (at least) NP-hard.*

3 Characterization of the existence of strong codings

Strong codings between trace monoids are closely related to morphisms of (in-)dependence alphabets.

Definition 4. Let (V, E) and (V', E') be undirected graphs. A *morphism* $H : (V, E) \to (V', E')$ is a relation between vertices $H \subseteq V \times V'$ such that $(a, b) \in E$ implies $H(a) \times H(b) \subseteq E'$ for all $a, b \in V$. (By $H(a)$ we mean the set $\{\alpha \in V' \mid (a, \alpha) \in H\}$ for $a \in V$.)

Obviously, undirected graphs with these morphisms form a category. For $H \subseteq V \times V'$ we denote by H^{-1} the inverse relation $H^{-1} = \{(\alpha, a) \in V' \times V \mid (a, \alpha) \in H\}$. A relation $H \subseteq V \times V'$ is a morphism $H : (V, E) \to (V', E')$ if and only if $H^{-1} : \overline{(V', E')} \to \overline{(V, E)}$ is a morphism on the complement graphs in the opposite direction. Hence, complementation yields a duality of the category.
The basic relation between strong homomorphisms of trace monoids and morphisms of undirected graphs is given by the next lemma, which follows directly from the above definitions.

Lemma 5. *Let $h : M(\Sigma, D) \to M(\Sigma', D')$ be a homomorphism of trace monoids. Define $H \subseteq \Sigma \times \Sigma'$ as*

$$H = \{(a, \alpha) \in \Sigma \times \Sigma' \mid \alpha \in alph(h(a))\}.$$

The following assertions are equivalent:

1. *$h : M(\Sigma, D) \to M(\Sigma', D')$ is a strong homomorphism.*
2. *$H : (\Sigma, I) \to (\Sigma', I')$ is a morphism of independence alphabets.*
3. *$H^{-1} : (\Sigma', D') \to (\Sigma, D)$ is a morphism of dependence alphabets.*

Definition 6. A morphism $G : (\Sigma', D') \to (\Sigma, D)$ of dependence alphabets is called *covering*, if for all $a \in \Sigma$ there exists $\alpha \in \Sigma'$ with $a \in G(\alpha)$ and if for all $(a, b) \in D$, $a \neq b$ there exists $(\alpha, \beta) \in D'$, $\alpha \neq \beta$ with $(a, b) \in G(\alpha) \times G(\beta)$.

The following lemma is stated with a different notation in [BFG94].

Lemma 7. *Let $h : M(\Sigma, D) \to M(\Sigma', D')$ be a strong coding. Define $H = \{(a, \alpha) \in \Sigma \times \Sigma' \mid \alpha \in alph(h(a))\}$. Then $H^{-1} : (\Sigma', D') \to (\Sigma, D)$ is a covering of dependence alphabets.*

Proof. By Lem. 5 the relation H^{-1} is a morphism. Let $a \in \Sigma$, then $h(a) \neq 1$, since h is a coding. Hence $a \in H^{-1}(\alpha)$ for some $\alpha \in \Sigma'$. Now, let $(a, b) \in D$, $a \neq b$. Since $h(ab) \neq h(ba)$ we find by Prop. 1 a pair $(\alpha, \beta) \in D'$, $\alpha \neq \beta$ such that $\pi_{\alpha,\beta} h(ab) \neq \pi_{\alpha,\beta} h(ba)$. Thus, either $(a, b) \in H^{-1}(\alpha) \times H^{-1}(\beta)$ or $(a, b) \in H^{-1}(\beta) \times H^{-1}(\alpha)$ holds. In any case H^{-1} is a covering.

Lemma 8. *Let $H \subseteq \Sigma \times \Sigma'$ be a relation such that $H^{-1} : (\Sigma', D') \to (\Sigma, D)$ is a covering of dependence alphabets.*
Then there exists a strong coding $h : M(\Sigma, D) \to M(\Sigma', D')$ such that $H = \{(a, \alpha) \in \Sigma \times \Sigma' \mid \alpha \in alph(h(a))\}$.

Proof. Let $\Sigma = \{a_1, a_2, ...\}$ and $\Sigma' = \{\alpha_1, \alpha_2, ...\}$. For each i define $\overrightarrow{H(a_i)} = \alpha_{i_1} \cdots \alpha_{i_k}$ and $\overleftarrow{H(a_i)} = \alpha_{i_k} \cdots \alpha_{i_1}$, where $H(a_i) = \{\alpha_{i_1}, \ldots, \alpha_{i_k}\}$ and $i_1 < \cdots < i_k$. For $i = 1, 2, \ldots$ let

$$h(a_i) = (\,\overrightarrow{H(a_i)}\,)^i\ \overleftarrow{H(a_i)}.$$

Lem. 5 states that h is a strong homomorphism. We have to show that h is injective, only. First note that $h(a) \neq 1$ for all $a \in \Sigma$. Assume by contradiction that $h(x) = h(y)$ for some $x \neq y$. Let x be of minimal length with this property. Then $x = ax'$ for some $a \in \Sigma$, $x' \in M(\Sigma, D)$. Since h is strong and $h(x) = h(y)$, there is some $b \in \mathrm{alph}(y)$ with $(a, b) \in D$. Hence we can write $y = zby'$ for some $b \in \Sigma, y', z \in M(\Sigma, D)$ such that $(a, z) \in I$ and $(a, b) \in D$.
We have $a \neq b$ since x is of minimal length. Since H^{-1} is a covering, we find $(\alpha, \beta) \in D'$, $\alpha \neq \beta$ such that $\alpha \in \mathrm{alph}(h(a))$ and $\beta \in \mathrm{alph}(h(b))$. For the contradiction it is enough to show $\pi_{\alpha,\beta} h(x) \neq \pi_{\alpha,\beta} h(y)$. Since h is strong, we have $\pi_{\alpha,\beta} h(z) = 1$. We may assume that α comes before β in Σ'. Let $a = a_i$ and $b = a_j$ $(i \neq j)$. We have $\pi_{\alpha,\beta} h(a) = (\alpha\beta^{\epsilon})^i\,(\beta^{\epsilon}\alpha)$ and $\pi_{\alpha,\beta} h(b) = (\alpha^{\delta}\beta)^j\,(\beta\alpha^{\delta})$ for some $\delta, \epsilon \in \{0, 1\}$. It follows that neither $\pi_{\alpha,\beta} h(a)$ is a prefix of $\pi_{\alpha,\beta} h(y)$ nor $\pi_{\alpha,\beta} h(zb)$ is a prefix of $\pi_{\alpha,\beta} h(x)$. Hence $h(x) = h(y)$ is impossible.

The following result solves the main open problem of [BFG94]. It follows by the conjunction of Lem. 7 and Lem. 8.

Theorem 9. *Let (Σ, D) and (Σ', D') be dependence alphabets. The following assertions are equivalent:*

1. *There exists a strong coding $h : M(\Sigma, D) \to M(\Sigma', D')$.*

2. There exists a covering $G : (\Sigma', D') \to (\Sigma, D)$ of dependence alphabets.

Furthermore there are effective constructions between h and G such that $G = H^{-1}$ and $H = \{(a, \alpha) \in \Sigma \times \Sigma' \mid \alpha \in alph(h(a))\}$.

Corollary 10. *The following problem is NP-complete:*
Input: Dependence alphabets (Σ, D), (Σ', D').
Question: Does there exist a strong coding from $M(\Sigma, D)$ into $M(\Sigma', D')$?

4 Codings into direct products of free monoids

In this section we investigate codings of a trace monoid into a k-fold direct product of free monoids. We are interested in the smallest possible value of k. For strong codings the situation is clear, as stated in Prop. 12 below. (This result follows also from [BFG94].)

Lemma 11. *Let $h : M(\Sigma, D) \to \prod_{i=1}^{k} \Sigma_i^*$ be a strong coding into a k-fold direct product of free monoids. Let $\pi_i = \pi_{\Sigma_i}$ denote the canonical projection onto the i-th component and $C_i = \{a \in \Sigma \mid \pi_i h(a) \neq 1\}$, $i = 1, \ldots, k$. Then C_i is a (dependence) clique for all $i = 1, \ldots, k$ and $(\Sigma, D) = (\bigcup_{i=1}^{k} C_i, \bigcup_{i=1}^{k} C_i \times C_i)$.*

Proof. Since h is injective, each $(a, b) \in D$ is contained in some $C_i \times C_i$. The set C_i is a clique, $i = 1, \ldots, k$, since h is strong.

Proposition 12. *Let (Σ, D) be a dependence alphabet and $|\Sigma_i| \geq 2$ for $i = 1, \ldots, k$. Then there exists a strong coding $h : M(\Sigma, D) \to \prod_{i=1}^{k} \Sigma_i^*$ if and only if (Σ, D) allows a covering by k cliques. In particular, deciding the existence of strong codings into k-fold direct product of free monoids is NP-complete.*

Proof. One direction is Lem. 11, the other follows from the Projection Lemma, Prop. 1. The question, whether the vertices *and* edges of a given graph can be covered by k cliques, is NP-complete, see [GJ78] for details.

4.1 A lower bound

We now turn to the problem of codings into direct products *without* the property of being a strong homomorphism. The following example shows a dependence alphabet where the existence of a coding into a k-fold direct product of free monoids also requires a covering by k cliques. Hence, in the example, the bound of Prop. 12 is optimal for codings, too.

Example 1. Let $(\Sigma, D) = P_n$ be the path of length $n \geq 1$, i.e., $\Sigma = \{a_1, \ldots, a_n\}$ with $D = \{(a_p, a_q) \mid 1 \leq p, q \leq n, |p - q| \leq 1\}$. Suppose $h : M(\Sigma, D) \to \prod_{i=1}^{k} \Sigma_i^*$ is a coding. Then we have $k \geq n - 1$.

Proof. Assume by contradiction that there is an embedding h into a k-fold direct product, $k \leq n - 2$. For $1 \leq m < n$ let $A_m = \{i \mid 1 \leq i \leq k,\ h_i(a_m a_{m+1}) \neq$

$h_i(a_{m+1}a_m)\}$, with $h_i = \pi_i h$. Clearly, $A_m \neq \emptyset$ and $|\bigcup_{m=1}^{n-1} A_m| \leq n-2$. Moreover, for every $1 \leq l < m < n$ with $m \geq l+2$ we have $A_m \cap A_l = \emptyset$ (since $\{a_{m+1}\} \times \{a_l, a_{l+1}\} \subseteq I$ and thus $h_i(a_{m+1}) = 1$ for $i \in A_l$).
Now, let $1 \leq p < q \leq n$ with $q-p$ minimal such that $|\bigcup_{m=p}^{q-1} A_m| \leq q-p-1$. Then $A_m \cap A_{m+1} \neq \emptyset$ for every $p \leq m < q-1$ follows by the minimality of $q-p$, together with the above observation. Furthermore, we have $|A_m \cup A_{m+1}| \geq 2$ (otherwise, it is easy to see that nonnegative integers $r, s \geq 1$ exist, such that $h(a_m^r a_{m+1} a_{m+2}^s) = h(a_{m+2}^s a_{m+1} a_m^r)$). We can now deduce that after renumbering we have $A_p = \{p\}$, $A_m = \{m-1, m\}$ $(p < m < q-1)$ and $A_{q-1} = \{q-2\}$.
The final step is to show the existence of integers $k_i \geq 1$ $(p \leq i \leq q)$ with $h(a_p^{k_p} \cdots a_q^{k_q}) = h(a_q^{k_q} \cdots a_p^{k_p})$ (note $q > p$), which yields the contradiction. For this, write $h_m(a_m) = r_m^{s_m}$ and $h_m(a_{m+2}) = r_m^{t_m}$, for some words r_m and integers $s_m, t_m \geq 1$, $p \leq m \leq q-2$. The claimed k_i, $p \leq i \leq q$, are now chosen as a (positive) solution of the system of $(q-p-2)$ equations $s_m \cdot k_m = t_m \cdot k_{m+2}$.

Remark. It is interesting to note that if (Σ, D) corresponds to C_n (the cycle of length n) and $n \geq 5$, then the optimal k for encoding $M(\Sigma, D)$ into a k-fold direct product of free monoids is again $k = n-1$. The lower bound can be seen using the above result on $|\bigcup_{m=1}^{n-1} A_m|$. For C_4 we can do better, see Ex. 2 below.

In the following we denote by $\alpha(\Sigma, D)$ the size of a maximal independent set of (Σ, D). Lem. 11 yields the following obvious lower bound.

Proposition 13. *Let $h : M(\Sigma, D) \to \prod_{i=1}^{k} \Sigma_i^*$ be a coding. Then we have $\alpha(\Sigma, D) \leq k$.*

Proof. $\mathbb{N}^{\alpha(\Sigma,D)}$ is a submonoid of $M(\Sigma, D)$.

4.2 Inductive methods

The following example (also used in [BF95]) provides two observations: For some codings we can achieve the lower bound $\alpha(\Sigma, D)$ of Prop. 13, and we may find a coding into a k-fold direct product, where no strong coding exists.

Example 2. Let (Σ, D) be a C_4, i.e., a cycle with four letters.

$$(\Sigma, D) \;=\; \begin{array}{ccc} a & - & d \\ | & & | \\ b & - & c \end{array}$$

Then there is a coding $h : M(\Sigma, D) \to \{a, b\}^* \times \{c, d\}^*$. Thus for a coding a 2-fold $(2 = \alpha(\Sigma, D))$ direct product of free monoids is enough, whereas for a strong coding we need four components by Prop. 12.

Proof. Choose any nonnegative integers $m_1, m_2, n_1, n_2 > 0$ such that the matrix $\begin{pmatrix} m_1 & n_1 \\ m_2 & n_2 \end{pmatrix}$ is non-singular. Define $h : M(\Sigma, D) \to \{a,b\}^* \times \{c,d\}^*$ by

$$h(a) = (a^{m_1}, c^{n_1}),\ h(b) = (b^{m_1}, d^{n_1}),\ h(c) = (a^{m_2}, c^{n_2}),\ h(d) = (b^{m_2}, d^{n_2}).$$

It is easily seen that h is injective. The basic observation is that $M(\Sigma, D)$ has the algebraic structure of a free product of $\mathbb{N}^2$ by $\mathbb{N}^2$.

In fact, Ex. 2 reveals a more general principle.

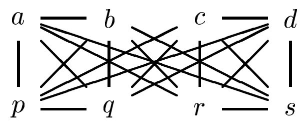
Proposition 14. *Let $h_j : M(\Sigma_j, D_j) \to \prod_{i=1}^{k} \Gamma_i^*$ be codings such that $|\Gamma_i| \geq 2$ and $\pi_i h_j(a) \neq 1$ for all $a \in \Sigma_j$, $i = 1, \ldots, k$, $j = 1, \ldots, m$. Then there exists a coding $h : \mathop{*}\limits_{j=1}^{m} M(\Sigma_j, D_j) \to \prod_{i=1}^{k} \Gamma_i^*$, where $\mathop{*}\limits_{j=1}^{m} M(\Sigma_j, D_j)$ denotes the m-fold free product. Furthermore, we find h such that $\pi_i h(a) \neq 1$ for all $a \in \dot{\bigcup}_{j=1}^{m} \Sigma_j$ and $i = 1, \ldots, k$.*

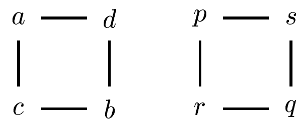
Proof. (Sketch) For each j replace Γ_i by some Γ_{ij} such that the alphabets become disjoint for different j. The codings h_j define a canonical homomorphism $\overline{h} : \mathop{*}\limits_{j=1}^{m} M(\Sigma_j, D_j) \to \prod_{i=1}^{k} (\dot{\bigcup}_{j=1}^{m} \Gamma_{ij})^*$. Since $\pi_i h_j(a) \neq 1$ it is easy to see that $\overline{h}$ is a coding. Furthermore, we have $\pi_i \overline{h}(a) \neq 1$ for all $a \in \dot{\bigcup}_{j=1}^{m} \Sigma_j$.
Finally, since $|\Gamma_i| \geq 2$ for all $i = 1, \ldots, k$, we can code $(\dot{\bigcup}_{j=1}^{m} \Gamma_{ij})^*$ into Γ_i^*.

Corollary 15. *Let $k, m \geq 1$ and a, b different letters, $a \neq b$. Then there exists a coding $h : \mathop{*}\limits_{i=1}^{m} \mathbb{N}^k \to \prod_{i=1}^{k} \{a,b\}^*$.*

Unfortunately, the hypothesis $\pi_i h_j(a) \neq 1$ of Prop. 14 is very strong. As soon as (Σ, D) contains three letters a, b, c such that $(a,b) \in D$, $(a,c) \in I$, and $(b,c) \in I$ it cannot be satisfied anymore. Even for cographs (Σ, D) the number $k = \alpha(\Sigma, D)$ is, in general, not large enough in order to allow a coding of $M(\Sigma, D)$ into a k-fold direct product of free monoids. We have the following example showing the strictness of our lower and upper bounds:

Example 3. Let the dependence alphabet be the following cograph:

$(\Sigma, D) =$ 

, i.e., $(\Sigma, I) =$ 

Then we have $\alpha(\Sigma, D) = 2$ and $M(\Sigma, D) = (\{a,b\}^* \times \{c,d\}^*) * (\{p,q\}^* \times \{r,s\}^*)$. The least k such that a coding $h : M(\Sigma, D) \to \prod_{1 \leq i \leq k} \Sigma_i^*$ exists is $k = 3$.

Proof. (Sketch) First assume $h : M(\Sigma, D) \to \Sigma_1^* \times \Sigma_2^*$ would be a coding. A combinatorial argument yields the existence of letters $x \in \{a,b,c,d\}$ and

$y \in \{p,q,r,s\}$ such that $h(x) = (u,1)$, $h(y) = (1,v)$ for some $u \in \Sigma_1^*$, $v \in \Sigma_2^*$. Then $h(xy) = h(yx) = (u,v)$ contradicting $xy \neq yx$.
Finally, consider the homomorphism $h : M(\Sigma, D) \to \{a,b,p,q\}^* \times \{r,s,e\}^* \times \{c,d,f\}^*$ given by the following table (where the columns give $h(x)$, $x \in \Sigma$):

h	a	b	c	d	p	q	r	s
$\{a,b,p,q\}$	a	b	1	1	p	q	1	1
$\{r,s,e\}$	e	e	e	e	1	1	r	s
$\{c,d,f\}$	1	1	c	d	f	f	f	f

To check injectivity, note that for $t \in M(\Sigma, D)$, $h(t)$ allows to decode the initial alphabet of t: if the 2nd (resp. 3rd) component starts in $\{r,s\}$ (resp. in $\{c,d\}$) then the corresponding letter belongs to $\mathrm{in}(t)$. Otherwise we have e and f in the last two components, which restricts $\mathrm{in}(t)$ to the set $\{a,b,p,q\}$, which in turn is identified by the first component of the encoding.

The next results concern the special case of embedding trace monoids with cograph dependence alphabets into direct products of free monoids. Consider two codings $h_1 : M_1 = M(\Sigma_1, D_1) \to \prod_{i=1}^k \Sigma_i^*$ and $h_2 : M_2 = M(\Sigma_2, D_2) \to \prod_{j=1}^l \Gamma_j^*$. Whereas the direct product $M_1 \times M_2$ can be embedded into a $(k+l)$-fold direct product (a tight bound), we are able to obtain a better upper bound for the free product $M_1 * M_2$. The next lemma presents an inductive method for embedding free products.

Lemma 16. *Let $M_1 = M(\Sigma_1, D_1)$, $M_2 = M(\Sigma_2, D_2)$ be given trace monoids. Let Σ_1, Σ_2 be pairwise disjoint, let Γ be an alphabet and $x \notin \Gamma$ a new letter. Then there exists a coding $h : M_1 * (M_2 \times \Gamma^*) \to (M_1 * M_2) \times (\Gamma \cup \{x\})^*$.*

Proof. Consider the homomorphism $h : M_1 * (M_2 \times \Gamma^*) \to (M_1 * M_2) \times (\Gamma \cup \{x\})^*$ given by

$$h(a) = \begin{cases} (a\,,\,x) & \text{for } a \in \Sigma_1 \\ (a\,,\,1) & \text{for } a \in \Sigma_2 \\ (1\,,\,a) & \text{for } a \in \Gamma \end{cases}$$

Note that h allows the decoding of initial alphabets as follows. The 2nd component of an encoded trace $h(z)$, $\pi_{\Gamma \cup \{x\}} h(z)$, starts by $a \in \Gamma$ if and only if a belongs to the initial alphabet of z, $\mathrm{in}(z)$. Otherwise, a letter $b \in (\Sigma_1 \cup \Sigma_2) \cap \mathrm{in}(z)$ can be determined using $\pi_{\Sigma_1 \cup \Sigma_2} h(z) = \pi_{\Sigma_1 \cup \Sigma_2} z$.

Theorem 17. *Let $h_1 : M(\Sigma_1, D_1) \to \prod_{i=1}^k \Sigma_i^*$ and $h_2 : M(\Sigma_2, D_2) \to \prod_{j=1}^l \Gamma_j^*$ be codings, $k, l \geq 1$. Then there exists a coding of $M(\Sigma_1, D_1) * M(\Sigma_2, D_2)$ into a $(k+l-1)$-fold direct product of free monoids. Moreover, this bound is optimal, in general.*

Proof. Clearly, for trace monoids M_i, N_i such that M_i can be embedded into N_i $(i = 1, 2)$, we can embed $M_1 * M_2$ canonically into $N_1 * N_2$. Thus, assume that $M_1 = M(\Sigma_1, D_1) = \prod_{i=1}^k \Sigma_i^*$ and $M_2 = M(\Sigma_2, D_2) = \prod_{j=1}^l \Gamma_j^*$ holds and let

w.l.o.g. $l \geq 2$. By Lem. 16 we can embed $(\prod_{i=1}^{k} \Sigma_i^*) * ((\prod_{j=1}^{l-1} \Gamma_j^*) \times \Gamma_l^*)$ into the monoid $((\prod_{i=1}^{k} \Sigma_i^*) * (\prod_{j=1}^{l-1} \Gamma_j^*)) \times (\Gamma_l \dot\cup \{x\})^*$. By induction, the left operand of the outer direct product can be embedded into a $(k+l-2)$-fold direct product, which yields the result.
The lower bound of this construction is obtained by generalizing Ex. 3 to the monoid $(\prod_{i=1}^{k}\{a,b\}^*) * (\prod_{j=1}^{l}\{p,q\}^*)$.

We consider in Thm. 19 below a nearly optimal method (with regard to the number of components) of encoding m-fold free products, $m \geq 3$, into direct products of free monoids. Let us start with a technical

Lemma 18. *Let $k, m \geq 2$ and let Σ_{ij} be pairwise disjoint alphabets, $i = 1, \ldots, k$, $j = 1, \ldots, m$. Then there exists a coding $h : \mathop{*}\limits_{j=1}^{m} (\prod_{i=1}^{k} \Sigma_{ij}^*) \to (\bigcup_{j=1}^{m} \Sigma_{1,j})^* \times (\mathop{*}\limits_{j=1}^{m} (\mathbb{N} \times \prod_{i=2}^{k} \Sigma_{ij}^*))$.*

Proof. Let x_j, $j = 1, \ldots, m$, be new letters. We identify $\{x_j\}^* \times \prod_{i=2}^{k} \Sigma_{ij}^*$ with $\mathbb{N} \times \prod_{i=2}^{k} \Sigma_{ij}^*$. The following homomorphism h is easily seen to be injective:

$$h(a) = \begin{cases} (a\,,\,(x_j,1)) & \text{if } a \in \Sigma_{1j} \text{ for some } 1 \leq j \leq m \\ (1\,,\,(1,a)) & \text{if } a \notin \bigcup_{j=1}^{m} \Sigma_{1j} \end{cases}$$

The theorem below can be applied in conjunction with Lem. 16 in order to encode m-fold free products efficiently for $m \geq 3$. (Lem. 16 is used for reducing the number of components.)

Theorem 19. *Let $h_j : M(\Sigma_j, D_j) \to \prod_{i=1}^{k} \Gamma_{ij}^*$ be codings, $j = 1, \ldots, m$, and $a \neq b$. Then there exists a coding $h : \mathop{*}\limits_{j=1}^{m} M(\Sigma_j, D_j) \to \prod_{i=1}^{2k}\{a,b\}^*$.*

Proof. Assume Γ_{ij} to be pairwise disjoint. By repeated application of Lem. 18 we obtain a coding $h' : \mathop{*}\limits_{j=1}^{m} (\prod_{i=1}^{k} \Gamma_{ij}^*) \to \prod_{i=1}^{k} \Theta_i^* \times (\mathop{*}\limits_{j=1}^{m} \mathbb{N}^k)$, with $\Theta_i = \bigcup_{j=1}^{m} \Gamma_{ij}$ for $i = 1, \ldots, k$. By standard methods we encode Θ_i^* into $\{a,b\}^*$ and, by Cor. 15, $\mathop{*}\limits_{j=1}^{m} \mathbb{N}^k$ into $\prod_{i=k+1}^{2k}\{a,b\}^*$. The result follows by composition with the coding of $\mathop{*}\limits_{j=1}^{m} M(\Sigma_j, D_j)$ into $\mathop{*}\limits_{j=1}^{m} (\prod_{i=1}^{k} \Gamma_{ij}^*)$.

Remark. Note that already for $m = 2$ the lower bound forThm. 19 is by Thm. 17 the value $2k - 1$. Hence, Thm. 19 gives an almost optimal construction.

5 Clique-preserving morphisms

Throughout this section the notion of clique is meant w.r.t. independence alphabets.

Definition 20. A clique-preserving morphism of independence alphabets $H : (\Sigma, I) \to (\Sigma', I')$ 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 $h : M(\Sigma, D) \to M(\Sigma', D')$ 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. (In fact, our construction is a faithful covariant functor from independence alphabets to trace monoids.)
The most prominent homomorphism of trace monoids arising this way is the coding used in the Projection Lemma, Prop. 1. Write $(\Sigma, D) = (\bigcup_{i=1}^{k} \Sigma_i, \bigcup_{i=1}^{k} \Sigma_i \times \Sigma_i)$ and let $\Sigma' = \dot{\bigcup}_{i=1}^{k} \Sigma_i$ be the disjoint union. The identity relations $id_{\Sigma_i} \subseteq \Sigma_i \times \Sigma_i$ induce in a natural way a relation $H \subseteq \Sigma \times \Sigma'$ which by abuse of language can be written as $H = \bigcup_{i=1}^{k} id_{\Sigma_i} \subseteq \Sigma \times \Sigma'$. The associated homomorphism h is exactly the strong coding of Prop. 1, $h : M(\Sigma, D) \to \prod_{i=1}^{k} \Sigma_i^*$.

Remark. Note that a clique-preserving morphism is not a morphism of undirected graphs as defined in Sect. 3, in general. 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.

The following proposition is in major contrast to the final result of Thm. 22 below.

Proposition 21. *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 $h : \Sigma^* \to M(\Sigma', D')$ 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 $\alpha \in H(a), \beta \in H(b)$.*

It is well-known that it is undecidable whether a homomorphism of trace monoids is injective, see [HC72, CR87]. The following theorem sharpens this result in the sense that we show the undecidability for a more natural class of homomorphism.

Theorem 22. *Given a clique-preserving morphism of independence alphabets $H : (\Sigma, I) \to (\Sigma', I')$, it is undecidable whether the associated homomorphism $h : M(\Sigma, D) \to M(\Sigma', D')$, $h(a) = \prod_{\alpha \in H(a)} \alpha$ for $a \in \Sigma$, is a coding.*

The proof of Thm. 22 is a technical and involved reduction of the halting problem of a two-counter machine. For lack of space we omit the details and refer to the full version of this paper.

6 Conclusion and open problems

In this paper we have solved the problem of the existence of strong codings for trace monoids by giving a NP-complete graph criterion. Whether or not the existence of codings is decidable remains an interesting open question. For

strong codings into a k-fold direct product of free monoids we know the smallest possible value of k. For codings, we know a lower and an upper bound for k, only. It is still possible that the smallest k is uncomputable.
We have extended a well-known undecidability result to a very natural class of homomorphism. The proof (not included in the present extended abstract) is very complicated. It would be interesting to find a simple direct proof.

References

[AG91] C. Àlvarez and J. Gabarró. The parallel complexity of two problems on concurrency. *Information Processing Letters*, 38:61–70, 1991.

[BF95] V. Bruyère and C. De Felice. Trace codings. *This volume.*

[BFG94] V. Bruyère, C. De Felice, and G. Guaiana. Coding with traces. In *Proc. of STACS'94*, LNCS 775, pp. 353–364. Springer, 1994.

[CF69] P. Cartier and D. Foata. *Problèmes combinatoires de commutation et réarrangements.* Lecture Notes in Mathematics 85. Springer, 1969.

[CL85] M. Clerbout and M. Latteux. Partial commutations and faithful rational transductions. *Theoretical Computer Science*, 35:241–254, 1985.

[CLB81] D. G. Corneil, H. Lerchs, and L. Stewart Burlingham. Complement reducible graphs. *Discrete Appl. Math.*, 3:163–174, 1981.

[CP85] R. Cori and D. Perrin. Automates et commutations partielles. *R.A.I.R.O. — Informatique Théorique et Applications*, 19:21–32, 1985.

[CPS85] D. G. Corneil, Y. Pearl, and L. K. Stewart. A linear recognition algorithm for cographs. *SIAM Journal of Computing*, 14:926–934, 1985.

[CR87] M. Chrobak and W. Rytter. Unique decipherability for partially commutative alphabets. *Fundamenta Informaticae*, X:323–336, 1987.

[Die89] V. Diekert. Word problems over traces which are solvable in linear time. *Theoretical Computer Science*, 74:3-18, 1990.

[Die90] V. Diekert. *Combinatorics on Traces.* LNCS 454. Springer, 1990.

[DR95] V. Diekert and G. Rozenberg, eds. *The Book of Traces.* World Scientific, Singapore, 1995. To appear.

[Dub86] C. Duboc. On some equations in free partially commutative monoids. *Theoretical Computer Science*, 46:159–174, 1986.

[GJ78] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* Freeman, San Francisco, 1978.

[HC72] G. Hotz and V. Claus. *Automatentheorie und Formale Sprachen, Band III.* Bibliographisches Institut, Mannheim, 1972.

[HY92] K. Hashiguchi and K. Yamada. String matching problems over free partially commutative monoids. *Information and Computation*, 101:131–149, 1992.

[Kel73] R. Keller. Parallel program schemata and maximal parallelism I. Fundamental results. *Journal of the ACM*, 20:514–537, 1973.

[Maz87] A. Mazurkiewicz. Trace theory. In *Petri Nets, Applications and Relationship to other Models of Concurrency*, LNCS 255. Springer, 1987.

[Och88] E. Ochmański. On morphisms of trace monoids. In *Proc. of STACS'88*, LNCS 294, pp. 346–355. Springer, 1988.

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