Masarykova univerzita v Brně
Přírodovědecká fakulta

Deciding Existence of Trace Codings

disertační práce

Vypracoval: Mgr. Michal Kunc
Školitel: doc. RNDr. Libor Polák, CSc.
Obor: Algebra

Brno, květen 2003

Prohlašuji, že jsem tuto disertační práci vypracoval samostatně a že jsem čerpal pouze z uvedených pramenů.

V Brně

Acknowledgements

I am obliged to my supervisor Libor Polák for constantly presenting me with various interesting topics and for his close support and deep confidence in my independent research; his comments always provided me with motivations for future work.

It was a real pleasure for me to collaborate with Ondřej Klíma, I appreciate our numerous valuable discussions as well as his remarks on preliminary versions of this thesis. I would also like to thank other members of our department, in particular the lecturers of our excellent graduate courses, and to my friends and colleagues for maintaining the great atmosphere in our workplace.

Further, I wish to express many thanks to all the people who took part in creating my pleasant stay at the Universidade de Coimbra under the Socrates/Erasmus programme in the spring of 2000, during which main ideas of the results presented in the thesis were formulated.

Finally, I am grateful to my parents for their continuous support through all years of my studies; without it this work would hardly be ever done.

Abstract

This thesis deals with the problem of deciding the existence of codings (injective morphisms) between trace monoids. We introduce the notion of weak morphisms of trace monoids and show that the problem of existence of weak codings is even more complex than the original problem for general morphisms. Further we investigate properties of weak morphisms and use them to prove the decidability of this problem for some classes of instances, which entails positive answers for the corresponding cases of the original problem. In particular, we show the decidability for instances whose domain monoids are defined by acyclic dependence graphs. We also partially answer the question of Diekert from 1990 about the number of free monoids needed for encoding a given trace monoid into their direct product. On the other hand, we prove that in general the problem of existence of codings from any given family of trace morphisms containing all weak codings is not recursively enumerable, which answers the question raised by Ochmański in 1988.

Contents

Introduction

Trace Theory

One of the central topics in contemporary computer science is the specification and automatic verification of parallel and concurrent programs. As such a work has to be based on a rigorous mathematical model, several algebraic formalisms for describing concurrent computations (often called process algebras) have been introduced, among others Milner's Calculus of Communicating Systems (CCS) [30], Bergstra and Klop's Algebra of Communicating Processes (ACP) [3], Hoare's Communicating Sequential Processes (CSP) [22] or the ISO standard Formal Description Technique LOTOS. There were also proposed many different representations of behavioural semantics for these models of various levels of coarseness, e.g. labeled transition systems, Petri nets or event structures.

In 1977 Mazurkiewicz [29] used free partially commutative monoids (often called trace monoids) for an intuitive and elegant coarse description of the behaviour of concurrent systems, and he showed that in this way the behaviour of elementary net systems (1-safe Petri nets) can be captured faithfully. In this approach one represents elementary actions with letters of a given alphabet; then an observation of a finite run of a system is nothing but a word over the alphabet. Two words are regarded as describing the same behaviour if and only if they can be obtained from each other by commuting adjacent occurrences of letters representing concurrent (or independent) 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 the 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.

Recently trace monoids were also successfully applied in the theoretical analysis of Message Sequence Charts (MSC), which is a standardized formalism for graphical specification of message exchange scenarios used in the design of communication protocols. Many basic decision problems concerning MSCs appeared to be closely related to some problems for semi-commutations and were settled by direct reductions to results from trace theory (see e.g. [31]).

From the mathematical point of view, trace monoids are a common generalization of finitely generated free and free commutative monoids. They are usually defined using symmetric binary relations on the sets of generators (so-called independence relations), which contain just the pairs of commuting letters.

The most important class of trace monoids are direct products of free monoids; already in the beginnings of computability theory, certain subsets of these monoids (such as rational and enumerable relations) were studied [34, 20]. General partial commutations were probably first employed in 1969 by Cartier and Foata [11] as a tool for the study of Möbius functions. In the following decades trace monoids appeared in connection with different research fields and a self-contained theory of traces has also gradually developed. Finally, in 1995 the first monograph on trace theory [18] was published, presenting an overview of various directions of research on partial commutativity, and two years later one chapter of the Handbook of Formal Languages [15] was devoted exclusively to trace monoids.

Topics considered in the framework of trace theory belong to many areas of both mathematics and theoretical computer science: algebraic structure of trace monoids as well as their combinatorial properties are studied, the theory of formal languages over partially commutative alphabets and logics corresponding to partially commuting variables were developed, solutions of unification and rewriting decision problems under partial commutativity are sought etc. Methods employed within the theory also have various origins, ranging from combinatorics on words to automata theory, logic or graph theory. In several branches of trace theory the research produced interesting results and challenging problems, many of which are still open.

Rational Trace Languages and Trace Codings

Since all behaviours of a concurrent system can be represented as a subset of some trace monoid, i.e. a language over a partially commutative alphabet, it is not surprising that a significant part of the research on trace monoids has been done in the framework of formal language theory. Attention of researchers focused mainly on the concepts of recognizable (i.e. recognized by finite automata) and rational (i.e. defined by means of rational expressions) trace languages, which led to the development of a common generalization of the classical theories of regular languages, (semi-)linear sets and rational relations [4]. Similarly to the theory of regular languages, the relationships between descriptions of languages using finite monoids, rational operations, standard automata, asynchronous automata and logic were established (see [15]).

It is well known that for regular word languages all basic problems are decidable. But this nice property is not shared by the class of rational trace languages, which contains also non-recognizable languages and where many problems — like deciding universality or recognizability of a language or equality of two languages — become non-recursive. This results in undecidability of some questions about trace monoids expressible in terms of rational expressions. One of such questions is the problem of determining whether a given morphism of trace monoids is injective, which was proved undecidable already in [24]. The interest in such morphisms stems from the fact that injective trace morphisms (often called codings) are a natural generalization of the classical notion of codes (see [5]) to partially commutative alphabets. Another motivation for studying properties of morphisms of trace monoids comes from the theory of concurrent systems since they arise when simulations of these systems are considered.

For most of the fundamental decision problems about rational trace languages a characterization of the monoids where these problems are decidable was already established [25, 2, 6, 21, 35, 1, 36] (for an overview see e.g. [15]). A lot of work has been done also in the study of trace codings, but the status of decision problems of trace codings turned out to be more complicated. The first task concerning trace codings is to give an effective procedure for deciding whether a given morphism is injective. A decision procedure for injectivity of word morphisms is well known [37] and for free commutative monoids the injectivity of a morphism coincides with the linear independence of images of letters, which is easily decidable using elementary results from linear algebra. As the injectivity problem can be easily reduced to the disjointness problem for rational languages in the codomain monoid, by means of the theorem of Aalbersberg and Hoogeboom [1] these classical positive results can be directly extended to the case of codomain monoids whose independence graph forms a transitive forest (for free domain monoids this was proved in [12]). Further work towards classification of monoids having the injectivity problem decidable has been done in [16, 23, 28, 32]; several examples of classes of monoids with decidable and undecidable injectivity problems have been found. It also turned out that for some trace monoids the problem of deciding the freeness of their submonoids is equivalent to the reachability problem for certain naturally arising classes of abstract machines, for instance Matiyasevich's $\mathbb{Q}$-machines, where the decidability of the reachability problem is still unknown. In [32] complexity issues of deciding injectivity of trace morphisms were considered.

In 1988 Ochmański [33] formulated several problems about trace codings. One of his conjectures was proved true in 1996 by Bruyère and De Felice: as trace monoids are defined by means of presentations, all their morphisms are determined by word morphisms of the corresponding free monoids; and in [8] they demonstrated that it is possible to obtain an injective morphism of trace monoids only if one starts with an injective word morphism. 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 a submonoid of the second one. It is usually referred to as the trace coding problem. This problem 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 two 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 [7] to all instances of the trace coding problem where the domain monoid is a direct product of free monoids. In [17] the existence problem was solved for so-called strong codings of trace monoids (defined in [10]) and several approximations for encoding of trace monoids into direct products of free monoids were proved. Some partial results about the case of domain monoids being free products of free commutative monoids were obtained in [9]. But in the full generality the problem remained completely open.

Overview

Most of the material presented in this thesis (except for Section 2.3) is contained in the paper [27] currently submitted for publication; main ideas of these results were briefly described in the extended abstract [26].

In order to deal with the problem of deciding the existence of codings between trace monoids, we consider a particular family of trace morphisms, whose members we call weak morphisms. We show that the analogous problem of existence of weak codings is even more complex than the original one for general morphisms and we prove its decidability for some classes of instances, which entails positive answers for the corresponding cases of the original question. On the other hand, we prove that in general the existence of codings from any given family of trace morphisms containing all weak codings is undecidable.

The thesis is organized as follows.

Basic definitions and results are recalled in Chapter 1; we refer the reader to [15, 32] for a more comprehensive overview of the theory. In Section 1.1 we deal with general notions of trace theory and in Section 1.2 we present elementary facts and known results about trace morphisms and codings and introduce the notions of strong and weak morphisms.

Chapter 2 is devoted to the study of certain natural classes of trace morphisms and codings. In Section 2.1 we demonstrate several characteristic properties of weak morphisms significant for our purposes. The aim of Section 2.2 is to describe how the original trace coding problem is connected with its equivalent for weak morphisms. And in the following Section 2.3 we use the same technique to find such a connection for another class of trace morphisms, so-called co-strong morphisms.

In Chapter 3 we consider certain restrictions on pairs of input monoids for the problem of existence of weak codings which are sufficient to obtain its decidability. The main tool employed there is the notion of wlt-mappings. These mappings are used to describe those choices of contents of images of generators of the domain monoid which allow to construct a weak coding. In Section 3.1 wlt-mappings are introduced and their basic properties are studied. Then they are utilized in Sections 3.2 and 3.3 to solve the trace coding problem for domain monoids with acyclic dependence graphs and for codomain monoids which are direct products of free monoids provided the dependence graph of the domain monoid does not contain cycles of length 3 or 4. Finally, Section 3.4 demonstrates that once the above assumptions on instances are weakened, the claims presented there are no longer valid.

The undecidability result for the general case of the trace coding problem is proved in Chapter 4. First we show that when properly restricted, the problem of existence of weak codings with partially prescribed contents of images of generators can be reduced to the trace coding problem; this is the aim of Section 4.1. In the rest of the chapter we prove the undecidability of this problem by constructing a reduction of the Post's correspondence problem (PCP). The actual construction is performed in Section 4.2, then Section 4.3 describes its main ideas on a particular instance of the PCP and in Section 4.4 we demonstrate that the construction really produces the desired outcome.

The final Chapter 5 is devoted to summarizing the main results.

General Notation

We mean by $\mathbb{Z}$, $\mathbb{N}_0$, $\mathbb{N}$ and $\mathbb{Q}_0^+$ the sets of all integers, non-negative integers, positive integers and non-negative rational numbers respectively. For $N \subseteq \mathbb{N}$, $\operatorname{lcm} N$ stands for the least common multiple of all numbers in N. The cardinality of an arbitrary set A is written as $|A|$. For sets A_1 and A_2, we denote by $\mathsf{p}_i : A_1 \times A_2 \to A_i$, for $i \in \{1,2\}$, the projection mappings. For sets A and B, a mapping $\varphi : A \to B$ and a subset $C \subseteq A$, the notation $\varphi|_C$ stands for the restriction of φ to C. The symmetric closure of any binary relation ρ is denoted by $\operatorname{sym} \rho$. For an $(m \times n)$-matrix K, $M \subseteq \{1,\dots,m\}$ and $N \subseteq \{1,\dots,n\}$, we mean by $K(M,N)$ the submatrix of K consisting of the rows with indices in the set M and columns with indices in the set N. For a decision problem P, coP is the *complement* of P, i.e. the problem for which the answer to every instance is the negation of the original answer.

The neutral element of any monoid is written as 1. A monoid morphism $\varphi : M \to M'$ is termed *non-erasing* when $\varphi(x) \neq 1$ for every $x \in M \setminus \{1\}$. We denote by Σ^* the monoid of *words* (*free monoid*) over a finite set Σ. In this context, Σ is often called an *alphabet* and its elements *letters*. Many times in our constructions, we enrich some alphabet with additional letters; in these situations we always implicitly assume that all new letters are different from the old ones. Let $\operatorname{alph} : \Sigma^* \to 2^{\Sigma}$ denote the *content* mapping assigning to every word $u \in \Sigma^*$ the set of all letters occurring in u. The symbol $\preceq$ is used for the *prefix* ordering on Σ^*. Let $u \in \Sigma^*$ be any word. We refer to the first letter of u as $\mathrm{first}(u)$, to its last letter as $\mathrm{last}(u)$, to the length of u as $|u|$ and to its mirror image as $\overleftarrow{u}$. For $X \subseteq \Sigma$, let $\pi_X : \Sigma^* \to X^*$ be the projection morphism defined by $\pi_X(x) = x$ for $x \in X$ and by $\pi_X(x) = 1$ for $x \in \Sigma \setminus X$. Finally, let $|u|_X$ denote $|\pi_X(u)|$. In the above notation, we often write instead of X a list of its elements.

By an *(undirected) graph* we mean a pair (V,E), where V is a finite set of vertices and $E \subseteq V \times V$ is a symmetric adjacency relation on V. For $X \subseteq V$, the *subgraph* of (V,E) *induced* by X is the graph $(X, E \cap (X \times X))$ and we denote it by (X,E). A *clique* in the graph (V,E) is a subset $X \subseteq V$ such that $X \times X \subseteq E$. A *path* in (V,E) of length $n \geq 0$ between vertices $x,y \in V$ is a sequence $x = z_0, z_1, \ldots, z_n = y$ of vertices satisfying $(z_{i-1}, z_i) \in E$ for every $i \in \{1, \ldots, n\}$. It is called *simple* if the vertices $z_0, \ldots, z_n$ are pairwise distinct. The *distance* between two vertices of the graph is the length of the shortest paths between them. A graph is called *connected* if every pair of its vertices is connected by a path. A *cycle* of length $n \geq 1$ in (V,E) is a sequence $x_0, x_1, \ldots, x_n = x_0$ of vertices satisfying $(x_{i-1}, x_i) \in E$ for every $i \in \{1, \ldots, n\}$. It is called *simple* if $n \geq 3$ and the vertices $x_0, \ldots, x_{n-1}$ are pairwise distinct. A *tree* is a non-empty acyclic and connected graph.

For $n \in \mathbb{N}$, $n \geq 3$, let C_n be a graph which forms a chordless cycle on n vertices and let P_n be a graph forming a simple path on n vertices. We say that a graph is *C_n-free* if it contains no induced subgraph isomorphic to C_n.

Chapter 1

Traces and Codes

This chapter contains the preliminary material. Section 1.1 is devoted to the basic notions and results of trace theory used in this thesis and in Section 1.2 we deal with elementary properties of trace morphisms and codings and recall fundamental facts about their decision problems. In the second section we also introduce the key notion of our considerations — weak trace morphisms.

1.1 Basic Notions and Properties

Definition 1.1.1. 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*.

Remark 1.1.2. Usually independence relations are defined as irreflexive, but we adopt this notation since it faithfully corresponds to the behaviour of weak morphisms. In fact, the difference between the key notions of strong and weak morphisms lies exactly in this modification.

For a letter $x \in \Sigma$, we mean by $D(x)$ the set of all letters from Σ dependent on x, i.e. $D(x) = \{y \in \Sigma \mid (x,y) \in D\}$. For $X \subseteq \Sigma$, we define $D(X) = \bigcup\{D(x) \mid x \in X\}$.

For the rest of this chapter, let (Σ, I), (Σ', I') be a pair of independence alphabets. The corresponding dependence relations will be denoted by D and D' and we will derive their names in this way also further on.

Now we can define the notion of trace monoids, which is a common generalization of the classical concepts of free and free commutative monoids.

Definition 1.1.3. 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 *free partially commutative monoid* or a *trace monoid*. Elements of this monoid are called *traces*.

Remark 1.1.4. Observe that the above construction establishes (up to isomorphisms) a one-to-one correspondence between independence alphabets and trace monoids.

For $X \subseteq \Sigma$, the submonoid of $\mathbb{M}(\Sigma, I)$ generated by X is clearly the trace monoid with independence graph (X, I) and we denote it simply by $\mathbb{M}(X, I)$.

Example 1.1.5. Finite direct products of finitely generated free monoids are exactly trace monoids $\mathbb{M}(\Sigma, I)$ which have the graph $(\Sigma, D \cup \mathrm{id}_\Sigma)$ transitive. Dually, finite free products of finitely generated free commutative monoids are just trace monoids defined by transitive independence alphabets.

A solution of the word problem for trace monoids is provided by the following so-called *Projection Lemma*.

Lemma 1.1.6 ([13]). *Let $u, v \in \Sigma^*$. Then $u \sim_I v$ if and only if*

$$\forall x \in \Sigma : \ |u|_x = |v|_x \quad \& \quad \forall (x,y) \in D : \ \pi_{x,y}(u) = \pi_{x,y}(v) \ .$$

In particular, trace monoids are cancellative.

As the content and the length of a word and the number of occurrences of a letter in a word are preserved by the congruence $\sim_I$, it makes sense to consider all these notions also for traces. The same can be deduced for projection morphisms since $u \sim_I v$ implies $\pi_X(u) \sim_I \pi_X(v)$ for any $X \subseteq \Sigma$ by Lemma 1.1.6.

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. Notice that in such a case they satisfy $st = ts$.

Definition 1.1.7. For $s \in \mathbb{M}(\Sigma, I)$ we define its *initial alphabet* and its *final alphabet* as $\mathrm{init}(s) = \{\mathrm{first}(u) \mid u \in s\}$, $\mathrm{fin}(s) = \{\mathrm{last}(u) \mid u \in s\}$ if $s \neq 1$ and $\mathrm{init}(1) = \mathrm{fin}(1) = \emptyset$.

Observe that two occurrences of a letter $x \in \Sigma$ in a trace $s \in \mathbb{M}(\Sigma, I)$ can be put together using the allowed commutations if and only if no letter dependent on x occurs between them. Thus occurrences of x in s are partitioned into blocks of mutually interchangeable occurrences, which can be formally defined as follows.

Definition 1.1.8. Let $s \in \mathbb{M}(\Sigma, I)$ be a trace and $n \in \mathbb{N}$. For a letter $x \in \Sigma$, an *x-block* of length n in s is a triple (t, x^n, t'), where the traces $t, t' \in \mathbb{M}(\Sigma, I)$ are such that $s = tx^n t'$, $\mathrm{fin}(t) \subseteq D(x)$ and $x \notin \mathrm{init}(t')$.

Example 1.1.9. Consider the dependence graph $(\Sigma, D) = x - y - z - p$ isomorphic to the graph P_4. Then the trace $(xzyxzpxz \sim_I) \in \mathbb{M}(\Sigma, I)$ contains three z-blocks but only two x-blocks.

Clearly each occurrence of a letter x in a trace s belongs to exactly one x-block. In the case of free monoids, let us denote, for $u \in \Sigma^*$ and $m \in \mathbb{N}$, by $u\langle m \rangle$ the word consisting of the first m blocks of u.

Using Lemma 1.1.6 it is easy to see that the following definition is correct.

Definition 1.1.10. Let $s \in \mathbb{M}(\Sigma, I)$. We denote by $\mathrm{red}(s)$ the trace in $\mathbb{M}(\Sigma, I)$ obtained from s by removing, for every $x \in \Sigma$, from each x-block all but one occurrence of x. We call $\mathrm{red}(s)$ the *reduct* of s. If $s = \mathrm{red}(s)$, we say that s is *reduced*.

Example 1.1.11. For the trace introduced in Example 1.1.9 the definition spells:

$$\mathrm{red}(xzyxzpxz \sim_I) = xzyzpxz \sim_I \ .$$

Remark 1.1.12. Notice that every reduct is a reduced trace.

The central construction of the paper is based on appending additional letters to two traces in order to achieve their equality. This can be done if the parts of these traces which do not belong to their common prefix are independent. Let us state this well-known fact in more detail. All claims of the next lemma can be easily verified using Lemma 1.1.6.

Lemma 1.1.13. *Let $u, v \in \Sigma^*$ satisfy*

$$\forall (x,y) \in D : \ \underset{x,y}{\pi}(u) \preceq \underset{x,y}{\pi}(v) \ \textit{or} \ \underset{x,y}{\pi}(v) \preceq \underset{x,y}{\pi}(u) \ . \tag{1.1}$$

Let $\bar{u}$ and $\bar{v}$ be the words resulting from u and v respectively when we take just the first $\min\{|u|_x, |v|_x\}$ occurrences of each letter $x \in \Sigma$ and denote the words consisting of the remaining occurrences of letters in u and v by $v \backslash u$ and $u \backslash v$ respectively. Then

$$u \sim_I \bar{u} \cdot (v \backslash u), \qquad v \sim_I \bar{v} \cdot (u \backslash v), \qquad \bar{u} \sim_I \bar{v}, \qquad \mathrm{alph}(v \backslash u) \times \mathrm{alph}(u \backslash v) \subseteq I \setminus \mathrm{id}_\Sigma$$

and consequently $u \cdot (u \backslash v) \sim_I v \cdot (v \backslash u)$. □

Remark 1.1.14. For $u, v, w, r \in \Sigma^*$ satisfying $u \sim_I v$ and $w \sim_I r$, due to Lemma 1.1.6 one can see that $u \backslash w \sim_I v \backslash r$.

From Lemmas 1.1.6 and 1.1.13 we immediately obtain the following fact which will be used implicitly throughout the paper.

Lemma 1.1.15. *Let $u, v \in \Sigma^*$. Then there exist $w, r \in \Sigma^*$ such that $uw \sim_I vr$ if and only if (1.1) is satisfied.* □

To reveal connections between weak morphisms and general ones we will employ in Section 2.2 the standard decomposition of traces into primitive roots. Let us now recall basic facts about this construction.

Definition 1.1.16. A trace $s \in \mathbb{M}(\Sigma,I) \setminus \{1\}$ is called *connected* whenever the graph $(\mathrm{alph}(s),D)$ is connected. It is called *primitive* if it is connected and for every trace $t \in \mathbb{M}(\Sigma,I)$ and $n \in \mathbb{N}$, the equality $s = t^n$ implies $n = 1$.

Proposition 1.1.17 ([19]). *Any connected trace is a power of a unique primitive trace.*

If a connected trace $s \in \mathbb{M}(\Sigma,I)$ is a power of a primitive trace t, then t is referred to as the *primitive root* of s. It is clear that 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 the trace s. Let us denote by $\mathscr{PR}(s)$ the set of all primitive roots of connected components of s.

Example 1.1.18. Let the dependence graph (Σ,D) be isomorphic to the path P_7 and let us call x_i the i-th letter on this path, for $1 \le i \le 7$. Then the decomposition of the trace $s = x_1x_2x_5x_6x_7x_1x_3x_7x_2x_6x_3x_5 \sim_I$ into primitive roots of its connected components can be written as $s = (x_1x_2x_3)^2 \cdot (x_5x_6x_7^2x_6x_5) \sim_I$ and therefore

$$\mathscr{PR}(s) = \{x_1x_2x_3 \sim_I, x_5x_6x_7^2x_6x_5 \sim_I\} .$$

The fundamental property of primitive roots is that primitive roots of commuting traces are always either equal or independent:

Proposition 1.1.19 ([19]). *Let traces $s_1, s_2 \in \mathbb{M}(\Sigma,I)$ satisfy $s_1s_2 = s_2s_1$. Then for all $t_1 \in \mathscr{PR}(s_1)$ and $t_2 \in \mathscr{PR}(s_2)$ either $t_1 = t_2$ or $\mathrm{alph}(t_1) \times \mathrm{alph}(t_2) \subseteq I \setminus \mathrm{id}_\Sigma$.*

1.2 Trace Morphisms and Codings

Since trace monoids are defined by presentations, every morphism of trace monoids (briefly called *trace morphism*) $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ is uniquely determined by an arbitrary mapping $\varphi_0 : \Sigma \to (\Sigma')^*$ such that $\varphi_0(x) \in \varphi(x)$ for each letter $x \in \Sigma$. Such a mapping always satisfies

$$\forall (x,y) \in I : \ \varphi_0(x)\varphi_0(y) \sim_{I'} \varphi_0(y)\varphi_0(x) . \tag{1.2}$$

Conversely, any mapping $\varphi_0 : \Sigma \to (\Sigma')^*$ satisfying (1.2) extends to a trace morphism. Often, when considering a morphism $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$, we actually work with a morphism $\psi : \Sigma^* \to (\Sigma')^*$ defined by a fixed mapping φ_0, i.e. such a morphism ψ that the diagram

$$\begin{tikzcd} \Sigma^* \arrow[r, "\psi"] \arrow[d, "\nu"'] & (\Sigma')^* \arrow[d, "\nu'"] \\ \mathbb{M}(\Sigma,I) \arrow[r, "\varphi"'] & \mathbb{M}(\Sigma',I') \end{tikzcd}$$

commutes, where the mappings ν, ν' are the natural morphisms to quotient monoids. The morphism ψ is called a *lifting* of φ and we will denote it by φ too. That is, we allow φ to be applied also to words, the image of any word under φ is always considered to be a word and the equality sign between words always means their equality in a free monoid. We adopt this convention in order to strictly differentiate algebraic considerations from combinatorics on words.

Let us recall that a *code* is a finite set of words satisfying no non-trivial relation. In other words, a finite subset $C \subseteq \Sigma^*$ is a code if and only if the submonoid generated by C in Σ^* is free over C, i.e. the morphism $\varphi : C^* \to \Sigma^*$ defined for all $u \in C$ by the rule $\varphi(u) = u$ is injective. Therefore an injective morphism of free monoids is sometimes called a coding.

As a natural generalization of the notion of codes to trace monoids we obtain the notion of trace codes.

Definition 1.2.1. If a submonoid M of $\mathbb{M}(\Sigma, I)$ is isomorphic to a trace monoid, then its minimal set of generators $(M \setminus 1) \setminus (M \setminus 1)^2$ is called a *trace code*.

Because trace codes are exactly images of sets of generators under injective trace morphisms, the terminology of the classical theory of codes can be adopted.

Definition 1.2.2. We say that a trace morphism $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ is a *coding* if it is injective.

The following theorem, which was proved by Bruyère and De Felice, shows that in order to obtain by the above construction a trace coding we have to start with a word coding on the corresponding free monoids.

Proposition 1.2.3 ([8]). *For an arbitrary trace coding φ, every lifting of φ to the corresponding free monoids is a coding.*

But in essence this interesting result says a lot about morphisms of free monoids rather than about trace morphisms, so we do not have to be conscious of it in our considerations. In fact, the result is deeply based on the defect effect of non-injective morphisms, which is a specific property of free monoids.

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

- strong morphisms, introduced in [10],
- cp-morphisms, which were introduced in [17] 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. This approach also suggests us to use an alternative definition of cp-morphisms.

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

$$\forall x \in \Sigma : \ \mathrm{alph}(\varphi(x)) \times \mathrm{alph}(\varphi(x)) \subseteq I' \ .$$

It is called a *cp-morphism* if it is weak and satisfies $\forall x \in \Sigma, \ a \in \Sigma' : \ |\varphi(x)|_a \leq 1$, i.e. every letter is mapped to a reduced trace.

To obtain for strong and weak morphisms descriptions analogous to the above one for general morphisms, it is enough to replace the condition (1.2) with respectively

$$\forall (x,y) \in I \setminus \mathrm{id}_\Sigma : \ \mathrm{alph}(\varphi_0(x)) \times \mathrm{alph}(\varphi_0(y)) \subseteq I' \setminus \mathrm{id}_{\Sigma'}$$

and

$$\forall (x,y) \in I : \ \mathrm{alph}(\varphi_0(x)) \times \mathrm{alph}(\varphi_0(y)) \subseteq I' \ .$$

Some useful properties of strong and weak morphisms are satisfied also by their common generalization naturally arising from these characterizations.

Definition 1.2.5. A morphism $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ is termed an *sw-morphism* if

$$\forall (x,y) \in I \setminus \mathrm{id}_\Sigma : \ \mathrm{alph}(\varphi(x)) \times \mathrm{alph}(\varphi(y)) \subseteq I' \ .$$

Remark 1.2.6. Notice that every trace morphism from a free monoid is strong and dually every trace morphism to a free commutative monoid is weak. Further, it is clear that a composition of strong (weak) morphisms is always strong (weak respectively); but this is far from being true for sw-morphisms (see Proposition 2.2.4 below).

The following simple observation suggests how rich the class of strong codings is.

Definition 1.2.7. We say that a morphism $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ is *connected* if for every $x \in \Sigma$ the trace $\varphi(x \sim_I)$ is connected.

Lemma 1.2.8 ([10]). *Every connected trace coding is strong.*

Proof. Let $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ be a connected coding and let $(x,y) \in I \setminus \mathrm{id}_\Sigma$. Then we have $|\mathscr{PR}(\varphi(x \sim_I))| = |\mathscr{PR}(\varphi(y \sim_I))| = 1$. Due to the injectivity of φ, Proposition 1.1.19 implies $\mathrm{alph}(\varphi(x)) \times \mathrm{alph}(\varphi(y)) \subseteq I' \setminus \mathrm{id}_{\Sigma'}$. Therefore the coding is strong. □

Let us denote the classes of all strong and weak morphisms by $\mathscr{S}$, $\mathscr{W}$ respectively.

Definition 1.2.9. The *trace code problem* asks to decide for a given trace morphism $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ whether it is a coding.

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 the class $\mathscr{C}$ contains all trace morphisms, then the question is just whether $\mathbb{M}(\Sigma,I)$ is isomorphic to a submonoid of $\mathbb{M}(\Sigma',I')$ and we call it briefly the trace coding problem (TCP).

The trace code problem is well-known to be undecidable even for strong morphisms when both monoids are fixed and $\mathbb{M}(\Sigma,I)$ is free (see e.g. [15]). The undecidability result in the case of cp-morphisms was established in [16] using substantially more complex construction; we generalize this statement in Proposition 4.2.1, which forms one step of the proof of the main result. On the other hand, some positive results were also achieved; most notable are the decidability of the trace code problem for connected morphisms and for codomain monoids whose independence alphabet is either a transitive forest [1] or acyclic [28, 32].

However, when we consider the problems of existence of codings, the situation is entirely different. In the first place, unlike for the trace code problem, it is not clear whether the complement of the problem is recursively enumerable.

The two classical cases of the TCP are simple: all finitely generated free monoids can be embedded into the one with two generators and for free commutative monoids injectivity of a morphism coincides with linear independence of images of letters. These characterizations were generalized in [7] to all instances of the TCP where the domain monoid is a direct product of free monoids. In both classical cases, there exists a strong coding as soon as there exists an arbitrary coding; but a weak coding between free monoids can be constructed only if the codomain alphabet has at least the same number of elements as the domain alphabet.

The $\mathscr{S}$-TCP turned out to be NP-complete due to the following result.

Proposition 1.2.10 ([17]). *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 $x,y \in \Sigma$:*

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

The reason for the relative simplicity of the $\mathscr{S}$-TCP is that a strong coding can be easily constructed as soon as reasonable contents of images of letters are chosen (this choice is provided by a mapping H). To see this, notice that in the image of a letter dependent letters may occur, which enables us to encode all of the information needed for deciphering whenever we can do it independently for all letters of the domain alphabet. And the defining condition of strong morphisms provides us with this freedom since the only properties we have to satisfy are commutativity and linear independence of images of independent letters, which are trivial in this case as their contents are disjoint. On the other hand, if we consider weak morphisms, the image of any letter consists entirely of independent letters and at the same time images of independent letters may contain common letters. So we have less opportunities to encode some information into images of letters under weak morphisms than under general morphisms and that is why the problem of existence of weak codings becomes even more complex than the one for general codings (see Theorem 5.1).

The main motive for considering weak trace morphisms is that, compared with general trace morphisms, they possess many properties substantially simplifying their manipulation. For instance, the following simple observations are very useful.

Lemma 1.2.11. *Let $\varphi : \mathbb{M}(\Sigma, \Sigma \times \Sigma) \to \mathbb{M}(\Sigma', I')$ be an arbitrary weak coding from a free commutative monoid. Then the set $A = \bigcup\{\mathrm{alph}(\varphi(x)) \mid x \in \Sigma\}$ forms a clique in the graph (Σ', I') and there exists an injective mapping $\rho : \Sigma \to A$ which satisfies $\rho(x) \in \mathrm{alph}(\varphi(x))$ for every $x \in \Sigma$. In particular, $|A| \geq |\Sigma|$.*

Proof. As φ is an injective linear mapping, it is defined by a matrix with its rank equal to $|\Sigma|$, which allows us to construct a desired mapping ρ. $\square$

Lemma 1.2.12. *Let $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ be a weak morphism and $u \in \Sigma^*$ a word. Then $\varphi(\overleftarrow{u}) \sim_{I'} \overleftarrow{\varphi(u)}$.*

Proof. We just calculate

$$\varphi(\overleftarrow{u}) = \varphi(x_n) \cdots \varphi(x_1) \sim_{I'} \overleftarrow{\varphi(x_n)} \cdots \overleftarrow{\varphi(x_1)} = \overleftarrow{\varphi(x_1) \cdots \varphi(x_n)} = \overleftarrow{\varphi(u)} \; ,$$

where $u = x_1 \cdots x_n$, $n \in \mathbb{N}_0$ and $x_1, \ldots, x_n \in \Sigma$. $\square$

Chapter 2

Restricted Classes of Morphisms

In this chapter we concentrate on classes of trace morphisms defined by additional requirements on contents of images of generators of the domain monoid. Section 2.1 is devoted to the study of properties of weak morphisms; we also develop there some methods of manipulating weak morphisms and codings and introduce the notation used in the subsequent chapters when constructing counter-examples to injectivity for weak morphisms. In Sections 2.2 and 2.3 we reveal connections between the existence of general codings and the existence of weak codings and so-called co-strong codings respectively. We also apply the calculus of weak codings to show in Section 2.2 that in order to decide the existence of codings between trace monoids it is enough to deal separately with all connected components of the dependence alphabet of the domain monoid and in Section 2.3 that when the dependence alphabet of the domain monoid is C_3-free there exists a co-strong coding between given trace monoids every time an arbitrary coding exists.

2.1 Weak codings

A trace morphism $\varphi : \mathbb{M}(\Sigma, I) \longrightarrow \mathbb{M}(\Sigma', I')$ is not a coding if and only if there exist two words $u, v \in \Sigma^*$ such that $u \nsim_I v$ and $\varphi(u) \sim_{I'} \varphi(v)$. It is often useful to consider just minimal words u and v satisfying these conditions, i.e. those possessing the least number $|u| + |v|$. For such counter-examples

$$\operatorname{init}(u \sim_I) \cap \operatorname{init}(v \sim_I) = \operatorname{fin}(u \sim_I) \cap \operatorname{fin}(v \sim_I) = \emptyset \tag{2.1}$$

always holds; otherwise a smaller counter-example can be obtained by cancellation due to Lemma 1.1.6.

If a morphism φ is a coding, then in particular

$$\forall (x,y) \in D : \ \varphi(xy) \nsim_{I'} \varphi(yx) \ , \tag{2.2}$$

which is equivalent to saying that the domain dependence relation D is induced by the dependence relation D' via the mapping φ. Observe that a morphism φ is weak and satisfies (2.2) if and only if

$$\forall x,y \in \Sigma : \ \mathrm{alph}(\varphi(x)) \times \mathrm{alph}(\varphi(y)) \subseteq I' \iff (x,y) \in I \, . \tag{2.3}$$

In this section 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.

Let us start with one observation about the form of counter-examples to injectivity for morphisms satisfying (2.3).

Lemma 2.1.1. *Let $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ be a trace morphism satisfying (2.3) and let $u, v \in \Sigma^*$ be any words such that $\varphi(u) \sim_{I'} \varphi(v)$. Then* $\mathrm{init}(u \sim_I) \times \mathrm{init}(v \sim_I) \subseteq I$.

Proof. Suppose that there exists $(x,y) \in D$ such that $x \in \mathrm{init}(u \sim_I)$ and $y \in \mathrm{init}(v \sim_I)$. By (2.3) there are $a \in \mathrm{alph}(\varphi(x))$, $b \in \mathrm{alph}(\varphi(y))$ satisfying $(a,b) \in D'$. As φ is weak, $a \in \mathrm{init}(\varphi(u) \sim_{I'})$ and $b \in \mathrm{init}(\varphi(v) \sim_{I'})$, which contradicts $\varphi(u) \sim_{I'} \varphi(v)$. $\square$

The following lemma shows that if a weak morphism φ is not injective, then it can be verified by a counter-example of one of two special forms — one of them based purely on linear dependence of words and the other on independence of letters of the codomain alphabet.

Lemma 2.1.2. *Let $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ be a weak trace morphism which is not a coding. Then at least one of the following cases arises:*

(i) There exists $X \subseteq \Sigma$ such that $X \times X \subseteq I$ and the system $(\varphi(x))_{x \in X}$ of elements of the free commutative monoid generated by the set

$$\bigcup \{\mathrm{alph}(\varphi(x)) \mid x \in X\}$$

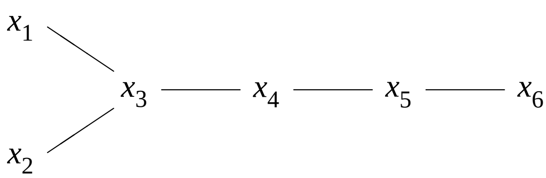
is linearly dependent.

(ii) There exists $u \in \Sigma^*$ such that $u \not\sim_I \overleftarrow{u}$, $\varphi(u) \sim_{I'} \varphi(\overleftarrow{u})$, $u \sim_I$ is connected and $\mathrm{init}(u \sim_I) \cap \mathrm{fin}(u \sim_I) = \emptyset$.

Moreover, if $D(x) \neq \emptyset$ for every $x \in \Sigma$, i.e. $\Sigma = D(\Sigma)$, the second claim is always true.

Before presenting the proof of this claim, let us give an example.

Example 2.1.3. Consider the relation D on the alphabet $\Sigma = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ defined by the graph



and let

$$\mathbb{M}(\Sigma',I') = \{a_1,a_2\}^* \times \{b_1,b_2\}^* \times \{c_1,c_2\}^* \times \{d_1,d_2\}^* .$$

Then one can disprove injectivity of any weak morphism φ from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$ with the following contents of images:

$$\begin{array}{lll} \mathrm{alph}(\varphi(x_1)) = \{a_1,b_1\} & \mathrm{alph}(\varphi(x_3)) = \{a_2,b_2,c_1\} & \mathrm{alph}(\varphi(x_5)) = \{c_1,d_2\} \\ \mathrm{alph}(\varphi(x_2)) = \{a_1,b_1\} & \mathrm{alph}(\varphi(x_4)) = \{a_1,b_1,c_2,d_1\} & \mathrm{alph}(\varphi(x_6)) = \{d_1\} . \end{array}$$

Since each of the letters a_2, b_2, c_2 and d_2 occurs just in one of the images of letters, without loss of generality we can assume that φ has the form:

$$\begin{array}{lll} \varphi(x_1) = a_1^i b_1^j & \varphi(x_3) = a_2 b_2 c_1^m & \varphi(x_5) = c_1^q d_2 \\ \varphi(x_2) = a_1^k b_1^l & \varphi(x_4) = a_1^n b_1^o c_2 d_1^p & \varphi(x_6) = d_1^r . \end{array}$$

Let us consider for instance the case $ln > ko$, $jn > io$ and $il > jk$. Then the word

$$u = x_1^{r(ln-ko)} x_3^q x_2^{r(jn-io)} x_4^{r(il-jk)} x_5^m x_6^{p(il-jk)}$$

verifies the condition (ii) of the lemma. In all the other cases such a word u can be constructed similarly — the inequalities determine the positions of the letters x_1, x_2 and x_4 with respect to x_3.

Proof of Lemma 2.1.2. We can assume that (2.2) holds, otherwise the condition (ii) is true. Take some words $v,w \in \Sigma^*$ satisfying $v \not\sim_I w$ and $\varphi(v) \sim_{I'} \varphi(w)$ such that the number $|v|+|w|$ is minimal possible. Consider the word $s = v\overleftarrow{w}$. Then we have $\overleftarrow{s} = w\overleftarrow{v}$ and therefore $\varphi(s) \sim_{I'} \varphi(\overleftarrow{s})$ by Lemma 1.2.12.

If $s \not\sim_I \overleftarrow{s}$ then we repeatedly employ cancellation to remove from the word s those letters which are simultaneously initial and final letters of the trace $s\sim_I$ until the set $\mathrm{init}(s\sim_I) \cap \mathrm{fin}(s\sim_I)$ is empty. More precisely, for any $x \in \mathrm{init}(s\sim_I) \cap \mathrm{fin}(s\sim_I)$, if $|s|_x = 1$ then $s \sim_I xt$ and $\overleftarrow{s} \sim_I x\overleftarrow{t}$, and if $|s|_x \geq 2$ then $s \sim_I xtx$ and $\overleftarrow{s} \sim_I x\overleftarrow{t}x$, for some word $t \in \Sigma^*$, so we can use this t as a new word s; the properties $s \not\sim_I \overleftarrow{s}$ and $\varphi(s) \sim_{I'} \varphi(\overleftarrow{s})$ are preserved thanks to Lemma 1.1.6. Now Lemma 1.1.6 guarantees the existence of some $(x,y) \in D$ such that $\pi_{x,y}(s) \neq \pi_{x,y}(\overleftarrow{s})$. Let the word $u \in \Sigma^*$ represent the connected component of $s\sim_I$ which contains the letters x and y. Then both $u \not\sim_I \overleftarrow{u}$ and $\varphi(u) \sim_{I'} \varphi(\overleftarrow{u})$ hold by Lemma 1.1.6 since two dependent letters from Σ' can occur in the images of elements of Σ under a weak morphism only within one component. This proves the second condition.

It remains to deal with the case $v\overleftarrow{w} \sim_I w\overleftarrow{v}$. Due to the minimality of v and w, we can use (2.1) to obtain $v = w\backslash v$ and $w = v\backslash w$, and therefore these words represent independent traces by Lemma 1.1.13. This implies $\mathrm{alph}(\varphi(v)) \times \mathrm{alph}(\varphi(w)) \subseteq I'$ since the morphism φ is weak. Hence the set $\mathrm{alph}(\varphi(v)) = \mathrm{alph}(\varphi(w))$ forms a clique in the graph (Σ', I'), and thus also $\mathrm{alph}(v) \cup \mathrm{alph}(w)$ is a clique in (Σ, I) by (2.3) and the case (i) arises. Under the additional assumption, we can now take any $x \in \mathrm{alph}(vw)$ and $y \in D(x)$ to obtain $vyw \not\sim_I wyv \sim_I \overleftarrow{vyw}$ using the minimality of v and w and then choose the only non-trivial connected component to get the validity of the second condition as in the previous paragraph. □

With Lemma 2.1.2 in hand, it is easy to find out that the status of the existence of weak codings often remains unchanged by adding new completely independent letters into the codomain alphabet.

Lemma 2.1.4. *Let (Σ, I) and $(\Sigma' \cup \Gamma, I')$ be two independence alphabets such that $\Sigma' \cap \Gamma = \emptyset$, $\Gamma \times (\Sigma' \cup \Gamma) \subseteq I'$ and $\Sigma = D(\Sigma)$. Let $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma' \cup \Gamma, I')$ be an arbitrary weak morphism. Then φ is a coding if and only if the weak morphism $\pi_{\Sigma'} \circ \varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ is a coding.*

Proof. Notice that any counter-example of the second form of Lemma 2.1.2 for the morphism $\pi_{\Sigma'} \circ \varphi$ is also a counter-example for the morphism φ. □

In the following, the task of deciding the existence of weak codings between given trace monoids is decomposed into separate tasks for the connected components of the domain dependence alphabet. First, we deal with letters of the domain alphabet independent on all the others.

Lemma 2.1.5. *There exists a weak coding*

$$\varphi : \mathbb{M}(\Sigma, I) \times \{x\}^* \to \mathbb{M}(\Sigma', I')$$

if and only if there exist a letter $a \in \Sigma'$ and a weak coding

$$\psi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma' \setminus (\{a\} \cup D'(a)), I') .$$

Proof. The converse implication of this claim is easily obtained by setting $\varphi(x) = a$ and $\varphi(y) = \psi(y)$ for all $y \in \Sigma$. In order to prove the direct one, observe first that $\mathrm{alph}(\varphi(x)) \times \mathrm{alph}(\varphi(s)) \subseteq I'$ for each $s \in \mathbb{M}(\Sigma, I)$ as φ is weak. We have to choose a suitable letter a in $\mathrm{alph}(\varphi(x))$. For the sake of contradiction, let us assume that for every $a \in \mathrm{alph}(\varphi(x))$ there exists a non-empty set $X_a \subseteq \Sigma$ such that $X_a \times X_a \subseteq I$ and the system $\left(\pi_{\Sigma' \setminus \{a\}}(\varphi(y))\right)_{y \in X_a}$ is linearly dependent. Provided X_a was chosen minimal possible, we can write

$$\pi_{\Sigma' \setminus \{a\}}\left(\varphi\left(\prod_{y \in X_a} y^{i_{ay}}\right)\right) \sim_{I'} \pi_{\Sigma' \setminus \{a\}}\left(\varphi\left(\prod_{y \in X_a} y^{j_{ay}}\right)\right)$$

for some $i_{ay}, j_{ay} \in \mathbb{N}_0$ satisfying $i_{ay} = 0 \iff j_{ay} \neq 0$. If there exist $a, b \in \mathrm{alph}(\varphi(x))$ and $z \in X_b$ such that $\{z\} \times X_a \nsubseteq I$, then

$$\varphi\left(\prod_{y \in X_a} y^{i_{ay}} \cdot z \cdot \prod_{y \in X_a} y^{j_{ay}}\right) \sim_{I'} \varphi\left(\prod_{y \in X_a} y^{j_{ay}} \cdot z \cdot \prod_{y \in X_a} y^{i_{ay}}\right)$$

contradicts the injectivity of φ. Otherwise $X_a \times X_b \subseteq I$ holds for all $a, b \in \mathrm{alph}(\varphi(x))$. Let us denote

$$i_a = \left|\varphi\left(\prod_{y \in X_a} y^{i_{ay}}\right)\right|_a \quad \text{and} \quad j_a = \left|\varphi\left(\prod_{y \in X_a} y^{j_{ay}}\right)\right|_a .$$

Because φ is a coding, we can assume that $i_a > j_a$ and consider the positive integer $k = \prod_{a \in \mathrm{alph}(\varphi(x))} (i_a - j_a)$. Then we also get a contradiction with the injectivity of φ :

$$\varphi\left(\prod_{\substack{a \in \mathrm{alph}(\varphi(x)) \\ y \in X_a}} y^{\frac{k \cdot |\varphi(x)|_a \cdot i_{ay}}{i_a - j_a}}\right) \sim_{I'} \varphi\left(x^k \cdot \prod_{\substack{a \in \mathrm{alph}(\varphi(x)) \\ y \in X_a}} y^{\frac{k \cdot |\varphi(x)|_a \cdot j_{ay}}{i_a - j_a}}\right) .$$

Thus we can choose a letter $a \in \mathrm{alph}(\varphi(x))$ such that, for every subset $X \subseteq \Sigma$ which satisfies $X \times X \subseteq I$, the system $\left(\pi_{\Sigma' \setminus \{a\}}(\varphi(y))\right)_{y \in X}$ is linearly independent. Let us define $\psi(y) = \pi_{\Sigma' \setminus \{a\}}(\varphi(y))$ for all $y \in \Sigma$. As for each $y \in \Sigma$ the emptiness of the set $\mathrm{alph}(\varphi(y)) \cap D'(a)$ follows from the weakness of φ, the weak morphism ψ really leads to the desired monoid. Clearly no counter-example of the form (i) of Lemma 2.1.2 exists for ψ due to our choice of the letter a. Let a word $u \in \Sigma^*$ satisfy the condition (ii) of Lemma 2.1.2 for ψ. Then $\psi(u) \sim_{I'} \psi(\overleftarrow{u})$ implies $\varphi(u) \sim_{I'} \varphi(\overleftarrow{u})$ since $\{a\} \times \mathrm{alph}(\varphi(u)) \subseteq I'$. But this contradicts the injectivity of φ. Hence ψ is a weak coding. □

Lemma 2.1.6. *There exists a weak coding from $\mathbb{M}(\Sigma, I)$ to $\mathbb{M}(\Sigma', I')$ if and only if there exists a weak coding $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ such that for every $x \in \Sigma \setminus D(\Sigma)$ the alphabet Σ' contains some letter a satisfying*

$$\mathrm{alph}(\varphi(x)) = \{a\} \quad \& \quad \forall y \in \Sigma \setminus \{x\} : \ \{a\} \times \mathrm{alph}(\varphi(y)) \subseteq I' \setminus \mathrm{id}_{\Sigma'} .$$

Proof. It is obtained by applying Lemma 2.1.5 inductively. □

Now we can state the general version of the decomposition result.

Proposition 2.1.7. *Let (Σ_i, I_i) for $i \in \{1, \ldots, n\}$ and (Σ', I') be arbitrary independence alphabets. Then there exists a weak coding*

$$\varphi : \prod_{i=1}^{n} \mathbb{M}(\Sigma_i, I_i) \to \mathbb{M}(\Sigma', I')$$

if and only if there exist subalphabets $\Sigma'_i \subseteq \Sigma'$*, for every* $i \in \{1,\dots,n\}$*, such that* $\Sigma'_i \times \Sigma'_j \subseteq I' \setminus \mathrm{id}_{\Sigma'}$ *holds for every* $i, j \in \{1,\dots,n\}$*,* $i \neq j$*, and weak codings*

$$\varphi_i : \mathbb{M}(\Sigma_i, I_i) \to \mathbb{M}(\Sigma'_i, I')$$

for every $i \in \{1,\dots,n\}$*.*

Proof. To get the converse implication, it is enough to define φ as the product of all codings φ_i. When considering the direct one, we can assume that φ satisfies the condition of Lemma 2.1.6. Let $\mathbb{M}(\Sigma, I)$ denote the domain product monoid, where Σ is the disjoint union of Σ_i and $D = \bigcup_{i=1}^n D_i$. Then in particular

$$\forall i \in \{1,\dots,n\} : \ D(\Sigma_i) \subseteq \Sigma_i \ \& \ (\Sigma_i \setminus D(\Sigma_i)) \times \Sigma \subseteq I$$

and $D(\Sigma) = \bigcup_{i=1}^n D(\Sigma_i)$. Let us consider the alphabets

$$\Sigma'_0 = \bigcup \{\mathrm{alph}(\varphi(x)) \mid x \in D(\Sigma)\} \quad \text{and} \quad X' = \Sigma'_0 \cap D'(\Sigma'_0) \ .$$

Applying Lemma 2.1.4 to the restriction of φ to $\mathbb{M}(D(\Sigma), I)$, we obtain a weak coding

$$\psi = \pi_{X'} \circ \varphi|_{\mathbb{M}(D(\Sigma),I)} : \mathbb{M}(D(\Sigma), I) \to \mathbb{M}(X', I')$$

satisfying $X' = \bigcup\{\mathrm{alph}(\psi(x)) \mid x \in D(\Sigma)\}$. Now every $a \in X'$ appears in the images of letters under ψ only within one connected component of the graph $(D(\Sigma), D)$; indeed, if $a \in \mathrm{alph}(\psi(x)) \cap \mathrm{alph}(\psi(y))$ then there exists a letter $b \in D'(a) \cap X'$ and consequently also $z \in D(\Sigma)$ with $b \in \mathrm{alph}(\psi(z))$, which satisfies $x\,D\,z$ and $y\,D\,z$ due to the weakness of ψ. Therefore $\mathrm{alph}(\psi(x)) \cap \mathrm{alph}(\psi(y)) = \emptyset$ holds for each $x \in D(\Sigma_i)$ and $y \in D(\Sigma_j)$ whenever $i \neq j$. So if we take

$$\Sigma'_i = \bigcup\{\mathrm{alph}(\psi(x)) \mid x \in D(\Sigma_i)\} \cup \bigcup\{\mathrm{alph}(\varphi(x)) \mid x \in \Sigma_i \setminus D(\Sigma_i)\}$$

and define

$$\varphi_i(x) = \begin{cases} \psi(x) & \text{for } x \in D(\Sigma_i) \ , \\ \varphi(x) & \text{for } x \in \Sigma_i \setminus D(\Sigma_i) \ , \end{cases}$$

we reach the desired conclusion due to our initial assumption on φ. □

Lemma 2.1.2 can be also used to deduce that if a weak morphism φ is strong too, then verifying (2.3) suffices for concluding that φ is a coding.

Lemma 2.1.8. *Every non-erasing strong morphism* $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ *which satisfies* (2.3) *is a coding. In particular, if* $|\Sigma| > 1$ *then every morphism* φ *from* Σ^* *to* $\mathbb{M}(\Sigma', I')$ *satisfying* (2.3) *is a coding.*

Proof. We prove that none of the cases of Lemma 2.1.2 can occur. The first condition does not hold since φ is non-erasing and contents of images of distinct independent letters are disjoint because φ is strong. So assume that $u \in \Sigma^*$ satisfies the case (ii) of Lemma 2.1.2. Take $x \in \operatorname{init}(u\sim_I)$ and let y be the last letter in u such that $(x,y) \in D$. By (2.3) there exist $a \in \operatorname{alph}(\varphi(x))$ and $b \in \operatorname{alph}(\varphi(y))$ with $(a,b) \in D'$. Due to the weakness of φ, $a \in \operatorname{init}(\varphi(u)\sim_{I'}) = \operatorname{fin}(\varphi(u)\sim_{I'})$. Because $a \notin \operatorname{alph}(\varphi(y))$, there is a letter z behind the last occurrence of y in u satisfying $a \in \operatorname{alph}(\varphi(z))$. As φ is strong, either $(x,z) \in D$ or $x = z$. But the former case is impossible by the choice of y and the latter case implies $x \in \operatorname{init}(u\sim_I) \cap \operatorname{fin}(u\sim_I)$, which contradicts the assumptions.

The second claim now follows from Remark 1.2.6. □

Now we are going to prove one assertion useful for showing the injectivity of a weak morphism by induction on the structure of the domain dependence alphabet. First, we state a technical lemma about word morphisms.

Lemma 2.1.9. *Let $\varphi : \Sigma^* \to (\Sigma')^*$ be a morphism. Let $X \subseteq \Sigma$, $a \in \Sigma'$ and $u,v \in \Sigma^*$ satisfy $\varphi(u) = \varphi(v)$, $|u|_x = |v|_x$ for every $x \in X$ and*

$$\forall x \in X\ \forall y \in \Sigma :\ \operatorname{alph}(\varphi(y)) \not\subseteq \{a\} \implies \pi_{x,y}(u) = \pi_{x,y}(v)\ .$$

Then $\varphi(\pi_{\Sigma\setminus X}(u)) = \varphi(\pi_{\Sigma\setminus X}(v))$.

Proof. Observe that after removing the i-th occurrence of a letter $x \in X$ from both words u and v, for $i \in \{1,\dots,|u|_x\}$, all assumptions of the lemma remain preserved. More precisely, if $\operatorname{alph}(\varphi(x)) \not\subseteq \{a\}$ then the occurrences of letters eliminated in this way from the words $\varphi(u)$ and $\varphi(v)$ are exactly the same and if $\operatorname{alph}(\varphi(x)) = \{a\}$ then all occurrences of a eliminated from $\varphi(u)$ and $\varphi(v)$ belong to the same a-block. □

Lemma 2.1.10. *Let $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ be an sw-morphism and $x \in \Sigma$ such that $\varphi|_{\mathbb{M}(\Sigma\setminus\{x\},I)}$ is a coding. Let words $u,v \in \Sigma^*$ satisfy $\varphi(u) \sim_{I'} \varphi(v)$, $|u|_x = |v|_x$ and $\pi_{x,y}(u) = \pi_{x,y}(v)$ for all $y \in D(x)$. Then $u \sim_I v$.*

Proof. To conclude $u \sim_I v$, it remains to show $\pi_{\Sigma\setminus\{x\}}(u) \sim_I \pi_{\Sigma\setminus\{x\}}(v)$. We verify the fact $\varphi(\pi_{\Sigma\setminus\{x\}}(u)) \sim_{I'} \varphi(\pi_{\Sigma\setminus\{x\}}(v))$ using Lemma 1.1.6, which is enough because the restriction $\varphi|_{\mathbb{M}(\Sigma\setminus\{x\},I)}$ is injective. It is clear from the equality $|u|_x = |v|_x$ that every letter has the same number of occurrences in $\varphi(\pi_{\Sigma\setminus\{x\}}(u))$ and $\varphi(\pi_{\Sigma\setminus\{x\}}(v))$. Consider $(a,b) \in D'$. If $a,b \notin \operatorname{alph}(\varphi(x))$ then

$$\pi_{a,b}\left(\varphi\left(\pi_{\Sigma\setminus\{x\}}(u)\right)\right) = \pi_{a,b}(\varphi(u)) = \pi_{a,b}(\varphi(v)) = \pi_{a,b}\left(\varphi\left(\pi_{\Sigma\setminus\{x\}}(v)\right)\right)\ .$$

Otherwise say $a \in \operatorname{alph}(\varphi(x))$. Then for every $y \in \Sigma\setminus\{x\}$ such that $b \in \operatorname{alph}(\varphi(y))$, we have $y\,D\,x$ since φ is an sw-morphism. Therefore Lemma 2.1.9 can be applied to the morphism $\pi_{a,b}\circ\varphi : \Sigma^* \to \{a,b\}^*$ for the set $\{x\}$, the letter a and the words u and v. We obtain the desired equality $\pi_{a,b}\big(\varphi(\pi_{\Sigma\setminus\{x\}}(u))\big) = \pi_{a,b}\big(\varphi(\pi_{\Sigma\setminus\{x\}}(v))\big)$. □

Notice that if the morphism φ is not assumed to be an sw-morphism, Lemma 2.1.10 does not hold:

Example 2.1.11. Let

$$\Sigma = \{x,y,z,p,q,r\},\ I = \mathrm{id}_\Sigma \cup \{(x,y),(y,x)\} \text{ and}$$
$$\mathbb{M}(\Sigma',I') = \{a_1,a_2\}^* \times \{b_1,b_2\}^* \times \{c_1,c_2\}^* .$$

Consider the morphism $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ given by the rules:

$$\begin{array}{lll} \varphi(x) = a_1a_2a_1 & \varphi(y) = a_1a_2a_1b_1b_2^2c_1c_2 & \varphi(z) = a_2a_1a_2c_1 \\ \varphi(p) = a_1b_1 & \varphi(q) = a_2a_1^2b_2c_2c_1^2c_2 & \varphi(r) = a_1a_2b_2c_2c_1 . \end{array}$$

It is not hard to verify that the restriction of φ to the submonoid $\mathbb{M}(\Sigma \setminus \{x\},I)$ is a coding; this is performed in detail as Example 2.1.21. But φ is not injective because $\varphi(yzypxqzr) \sim_{I'} \varphi(pzxqry^2z)$ holds although $yzypxqzr \not\sim_I pzxqry^2z$. Moreover, the morphism φ can be easily modified to ensure that any counter-example $\varphi(u) \sim_{I'} \varphi(v)$ to the injectivity of φ, where $u,v \in \Sigma^*$, satisfies $|u|_x = |v|_x$ and $\pi_{x,\bar{x}}(u) = \pi_{x,\bar{x}}(v)$ for all letters $\bar{x} \in D(x)$; it is sufficient to introduce a new letter x' into the φ-image of x and new letters z', p', q' and r', all of them dependent exactly on x', into the images of the corresponding letters in Σ.

The aim of the following considerations is to describe one method of manipulating numbers of occurrences of letters in the images under a weak morphism in order to simplify the morphism before starting any computations.

Definition 2.1.12. For a morphism $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ and a mapping $N : \Sigma \to \mathbb{N}$, let φ^N denote the morphism from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$ defined, for all $x \in \Sigma$, by the rule $\varphi^N(x) = \varphi(x)^{N(x)}$.

Lemma 2.1.13. *If $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ is a coding and $N : \Sigma \to \mathbb{N}$ is an arbitrary mapping, then φ^N is a coding as well.*

Proof. It is clear that $\mathrm{id}^N_{\mathbb{M}(\Sigma,I)}$ is a coding and $\varphi^N = \varphi \circ \mathrm{id}^N_{\mathbb{M}(\Sigma,I)}$. □

For any alphabet Σ and $n \in \mathbb{N}$, let us denote by N_n the corresponding constant mapping, i.e. $N_n(x) = n$ for all $x \in \Sigma$.

As the construction of Definition 2.1.12 preserves contents of images of letters, it is clear that φ^N is a weak morphism whenever φ is. Now we state a simple observation and then we use it to prove that for weak morphisms the converse of Lemma 2.1.13 also holds.

Lemma 2.1.14. *If $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ is a weak morphism and $n \in \mathbb{N}$ any positive integer, then $\varphi^{N_n} = \varphi \circ \mathrm{id}^{N_n}_{\mathbb{M}(\Sigma,I)} = \mathrm{id}^{N_n}_{\mathbb{M}(\Sigma',I')} \circ \varphi$.* □

Lemma 2.1.15. *Let $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ be a weak morphism. If φ^N is a coding for some $N : \Sigma \to \mathbb{N}$, then φ is a coding as well.*

Proof. Let $n = \operatorname{lcm}\{N(x) \mid x \in \Sigma\}$ and define $N' : \Sigma \to \mathbb{N}$ by the rule $N'(x) = n/N(x)$ for all $x \in \Sigma$. Then the morphism $\varphi^{N_n} = \varphi^N \circ \mathrm{id}^{N'}_{\mathbb{M}(\Sigma,I)}$ is a coding. Since we also have $\varphi^{N_n} = \mathrm{id}^{N_n}_{\mathbb{M}(\Sigma',I')} \circ \varphi$ by Lemma 2.1.14, this gives us the injectivity of φ. □

The previous lemma is a typical property of weak morphisms, once again it is not true in general:

Example 2.1.16. Take $\Sigma = \{x, y, z\}$ and $\Sigma' = \{a, b\}$. Let a morphism $\varphi : \Sigma^* \to (\Sigma')^*$ be given by the rules $\varphi(x) = a$, $\varphi(y) = b$, $\varphi(z) = ab$ and consider $N(x) = 2$ and $N(y) = N(z) = 1$. Then φ^N is a coding, but φ is not.

The following statement asserts that every weak coding can be modified without violating its injectivity to achieve that in the image of any letter there is at most one occurrence of an element from a reasonably chosen subset of the codomain alphabet.

Lemma 2.1.17. *Let $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ be a weak coding and $A \subseteq \Sigma'$ such that*

$$\forall x \in \Sigma : \ |\operatorname{alph}(\varphi(x)) \cap A| \leq 1 \, .$$

Then there exists a weak coding $\psi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ satisfying

$$\forall x \in \Sigma : \ \operatorname{alph}(\psi(x)) = \operatorname{alph}(\varphi(x)) \ \& \ |\psi(x)|_A \leq 1$$

and for all $x, y \in \Sigma$, $a, b \in \operatorname{alph}(\varphi(x)) \cap \operatorname{alph}(\varphi(y))$:

$$\frac{|\psi(x)|_a}{|\psi(y)|_a} : \frac{|\psi(x)|_b}{|\psi(y)|_b} = \frac{|\varphi(x)|_a}{|\varphi(y)|_a} : \frac{|\varphi(x)|_b}{|\varphi(y)|_b} \, .$$

Proof. Let

$$n = \operatorname{lcm}\{|\varphi(x)|_A \mid x \in \Sigma, \ \operatorname{alph}(\varphi(x)) \cap A \neq \emptyset\}$$

and define $N : \Sigma \to \mathbb{N}$ and $N' : \Sigma' \to \mathbb{N}$ by the rules:

$$N(x) = \begin{cases} n/|\varphi(x)|_A & \text{if } \operatorname{alph}(\varphi(x)) \cap A \neq \emptyset \, , \\ 1 & \text{otherwise} \, , \end{cases}$$

$$N'(a) = \begin{cases} n & \text{for } a \in A \, , \\ 1 & \text{for } a \notin A \, . \end{cases}$$

Let ψ be the morphism given by the formula:

$$\psi(x) = \begin{cases} a \cdot \pi_{\Sigma' \setminus A}(\varphi^N(x)) & \text{if } a \in \operatorname{alph}(\varphi(x)) \cap A \, , \\ \varphi^N(x) & \text{if } \operatorname{alph}(\varphi(x)) \cap A = \emptyset \, . \end{cases}$$

Then $\varphi^N = \mathrm{id}^{N'}_{\mathbb{M}(\Sigma',I')} \circ \psi$ is a coding by Lemma 2.1.13 and thus ψ is also injective. The remaining conditions are easy to verify. □

The following notions are introduced in order to formalize reasoning in the course of a construction of a counter-example to injectivity.

Definition 2.1.18. Let $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ be a morphism. Let $u, v \in \Sigma^*$. We say that the pair (u, v) is a *semi-equality* for φ if $\mathrm{init}(u \sim_I) \cap \mathrm{init}(v \sim_I) = \emptyset$ and there exist words $s, t \in (\Sigma')^*$ 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 $uw \sim_I vr$.

The next definition makes sense thanks to the defining properties of semi-equalities due to Lemma 1.1.15.

Definition 2.1.19. If (u, v) is a semi-equality for a morphism $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$, then the pair

$$(u', v') = \big((\varphi(v) \backslash \varphi(u)) \sim_{I'}, (\varphi(u) \backslash \varphi(v)) \sim_{I'}\big) \in \mathbb{M}(\Sigma', I') \times \mathbb{M}(\Sigma', I')$$

is called the *state* of (u, v) and the pair $(\mathrm{red}(u'), \mathrm{red}(v'))$ the *reduced state* of (u, v).

When dealing with semi-equalities, we often omit the reference to the morphism provided it is clear from the context. Recall that by Lemma 1.1.13 every state (u', v') satisfies $\mathrm{alph}(u') \times \mathrm{alph}(v') \subseteq I' \setminus \mathrm{id}_{\Sigma'}$.

The following lemma states that semi-equalities arise as initial parts of minimal counter-examples to injectivity.

Lemma 2.1.20. *Let $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ be a morphism and $u, v \in \Sigma^*$ be words satisfying $u \not\sim_I v$ and $\varphi(u) \sim_{I'} \varphi(v)$ such that $|u| + |v|$ is minimal. If $w \preceq u$ and $r \preceq v$, then (w, r) is a semi-equality.* □

All of the information we need to explore possible continuations of a semi-equality is contained in its state. Let us demonstrate this by an example.

Example 2.1.21. The diagram below shows that the restriction of the morphism φ introduced in Example 2.1.11 to the submonoid $\mathbb{M}(\Sigma \setminus \{x\}, I) \cong \{y, z, p, q, r\}^*$ is really injective. The idea of the calculation is to find a counter-example to injectivity by starting from the pair $(1, 1)$ and successively adding letters y, z, p, q and r from the right to both components to build new semi-equalities with the aim of reaching some semi-equality possessing the state $(1, 1)$. The states of the semi-equalities obtained by this construction are depicted here together with the letters from Σ used to acquire them; for each state, the underlined component is the one we try to extend in every possible way without violating (1.1).

$$
\begin{tikzcd}
(\underline{1},\underline{1}) \arrow[r, "{(y,p)}"] & (a_2a_1b_2^2c_1c_2,\underline{1}) \arrow[r, "{(1,z)}"] & (b_2^2c_2,\underline{a_2}) \arrow[r, "{(1,q)}"] \arrow[d, "{(1,r)}"] & (\underline{b_2},a_2^2a_1^2c_1^2c_2) \\
(\underline{1},a_2a_1^2c_2c_1^2c_2) \arrow[r, shift right, "{(q,1)}"'] & (b_2,\underline{1}) \arrow[l, shift right, "{(1,q)}"'] \arrow[r, shift right, "{(1,z)}"'] \arrow[d, shift left, "{(1,r)}"] & (\underline{b_2},a_2a_1a_2c_1) \arrow[l, shift right, "{(z,1)}"'] & (a_1b_2c_1c_2,\underline{1}) \\
 & (\underline{1},a_1a_2c_2c_1) \arrow[r, "{(p,1)}"] \arrow[u, shift left, "{(r,1)}"] & (\underline{b_1},a_2c_2c_1) \arrow[r, "{(q,1)}"] & (a_1^2b_1b_2c_1c_2,\underline{1}) \arrow[u, "{(1,p)}"']
\end{tikzcd}
$$

For weak morphisms, Lemma 2.1.20 can be partially reversed, namely, if we have a semi-equality whose state consists entirely of independent letters, then it can be prolonged into a counter-example.

Lemma 2.1.22. *Let $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ be a weak morphism 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.

Proof. Using Lemma 1.2.12 one calculates

$$\varphi(u\overleftarrow{v}) \sim_{I'} \varphi(u)\overleftarrow{\varphi(v)} \sim_{I'} \varphi(v)\overleftarrow{\varphi(u)} \sim_{I'} \varphi(v\overleftarrow{u}) \, ,$$

where the equivalence in the middle is a consequence of Lemma 1.1.13 since

$$(\varphi(v)\backslash\varphi(u))(\overleftarrow{\varphi(u)\backslash\varphi(v)}) \sim_{I'} (\varphi(u)\backslash\varphi(v))(\overleftarrow{\varphi(v)\backslash\varphi(u)})$$

due to the assumption. □

Let us now justify the consideration of reduced states. In Section 4.4 we need to find a counter-example to injectivity under certain assumptions. In order to do this, we construct some semi-equality and then inductively extend it until Lemma 2.1.22 can be applied. 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 first prerequisite for this construction is the ability to multiply the lengths of blocks in the state of the current semi-equality by an arbitrary fixed positive integer.

Lemma 2.1.23. *Let $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ be a weak morphism and let (u,v) be a semi-equality for φ with a state (u',v'). Then, for every $n \in \mathbb{N}$, the pair*

$$\left(\mathrm{id}^{N_n}_{\mathbb{M}(\Sigma,I)}(u), \mathrm{id}^{N_n}_{\mathbb{M}(\Sigma,I)}(v)\right)$$

is a semi-equality for φ with the state

$$\left(\mathrm{id}^{N_n}_{\mathbb{M}(\Sigma',I')}(u'), \mathrm{id}^{N_n}_{\mathbb{M}(\Sigma',I')}(v')\right) .$$

Proof. It follows directly from Lemma 2.1.14, Remark 1.1.14 and from the fact

$$\mathrm{id}^{N_n}_{\mathbb{M}(\Sigma',I')}(\varphi(v)) \backslash \mathrm{id}^{N_n}_{\mathbb{M}(\Sigma',I')}(\varphi(u)) = \mathrm{id}^{N_n}_{\mathbb{M}(\Sigma',I')}(\varphi(v) \backslash \varphi(u)) ,$$

which is easy to verify. □

Often we have to append a new pair of elements of Σ to a semi-equality in order to remove a given letter from the state, which in our constructions usually results in a replacement of this letter with another one together with some effect on the rest of the current state. This situation is described in general by the following lemma.

Lemma 2.1.24. *Let $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ be a weak morphism and let (u,v) be a semi-equality for φ with a state (u',v'). In addition, let $x,y \in \Sigma$ and*

$$a \in \mathrm{init}(v') \cap \mathrm{alph}(\varphi(x)) \setminus \mathrm{alph}(\varphi(y)) \tag{2.4}$$

be letters satisfying

$$\left(\mathrm{alph}(u' \cdot \varphi(x)) \setminus \{a\}\right) \times \mathrm{alph}(v' \cdot \varphi(y)) \subseteq I' , \tag{2.5}$$

$$x \in D(\mathrm{alph}(u)) \;\&\; y \in D(\mathrm{alph}(v)) . \tag{2.6}$$

Let m be the length of the first a-block in v' and $n = |\varphi(x)|_a$. Then the pair

$$\left(\mathrm{id}^{N_n}_{\mathbb{M}(\Sigma,I)}(u) \cdot x^m, \mathrm{id}^{N_n}_{\mathbb{M}(\Sigma,I)}(v) \cdot y^m\right) \tag{2.7}$$

is a semi-equality for φ and its state $(\overline{u},\overline{v})$ satisfies $a \notin \mathrm{init}(\overline{v})$.

Proof. Due to the assumption (2.6), the initial alphabets of the new pair are the same as those of (u,v), which verifies the first condition of Definition 2.1.18. We are going to check the validity of the second condition using Lemma 1.1.15. By Lemma 2.1.23, it is enough to verify that (1.1) holds for the traces

$$u_0 = \mathrm{id}^{N_n}_{\mathbb{M}(\Sigma',I')}(u') \cdot (\varphi(x))^m \quad \text{and} \quad v_0 = \mathrm{id}^{N_n}_{\mathbb{M}(\Sigma',I')}(v') \cdot (\varphi(y))^m .$$

It is clear for letters different from a due to (2.5). Since $\mathrm{alph}(u') \times \mathrm{alph}(v') \subseteq I' \setminus \mathrm{id}_{\Sigma'}$ and φ is weak, no letters dependent on a appear in u_0, and because $a \notin \mathrm{alph}(\varphi(y))$, the number of occurrences of a in u_0 is equal to the length of the first a-block in v_0. Hence (1.1) holds also for a and $a \notin \mathrm{init}(\overline{v})$ as desired. □

Remark 2.1.25. The construction of Lemma 2.1.24 will be in fact occasionally used even when the condition (2.5) is not satisfied; in such a case one has to ensure that (2.7) is really a semi-equality independently.

2.2 Reduction to the Weak Coding Equivalent

In this section we describe the fundamental connection between the problems TCP and $\mathscr{W}$-TCP. For an arbitrary 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. By Proposition 1.1.19 primitive traces 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 and replace these primitive roots with them. Since in each of the images there is at most one primitive root with a given content, for fixed alphabets (Σ, I) and (Σ', I') we can manage with a finite number of new letters. In this way we express every morphism φ as a composition of a weak morphism and a strong morphism. Clearly, if φ is a coding, the weak morphism constructed must be a coding as well. On the other hand, we can use Proposition 1.2.10 to find a strong coding for prolonging any coding to the new codomain monoid into a coding to the original one.

Let us perform this construction in detail. For a graph (V, E), we denote

$$\overline{\mathscr{C}}(V,E) = \{X \subseteq V \mid |X| \geq 2 \text{ and } (X,E) \text{ is connected}\} .$$

Definition 2.2.1. Let (Σ, I), (Σ', I') be independence alphabets. We extend (Σ', I') into a new independence alphabet $({\Sigma'}_\Sigma, {I'}_\Sigma)$ as follows. Let

$${\Sigma'}_\Sigma = \Sigma' \cup (\overline{\mathscr{C}}(\Sigma', D') \times \Sigma)$$

and, for a word $u \in ({\Sigma'}_\Sigma)^*$, define its *extended content* $\operatorname{ealph}(u) \subseteq \Sigma'$ as

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

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

$$(\alpha,\beta) \in {I'}_\Sigma \iff \operatorname{ealph}(\alpha) \times \operatorname{ealph}(\beta) \subseteq I' \text{ or } \alpha = \beta . \tag{2.8}$$

Then the pair of independence alphabets (Σ, I), $({\Sigma'}_\Sigma, {I'}_\Sigma)$ is called *saturated*.

Remark 2.2.2. It is easy to verify that $\operatorname{ealph}(\alpha) \cap \operatorname{ealph}(\beta) \neq \emptyset$ implies $(\alpha, \beta) \in {D'}_\Sigma$ for any distinct elements $\alpha, \beta \in {\Sigma'}_\Sigma$; therefore the relation ${I'}_\Sigma$ can be equivalently defined by the condition

$$(\alpha,\beta) \in {I'}_\Sigma \setminus \mathrm{id}_{{\Sigma'}_\Sigma} \iff \operatorname{ealph}(\alpha) \times \operatorname{ealph}(\beta) \subseteq I' \setminus \mathrm{id}_{\Sigma'} . \tag{2.9}$$

Notice also that if a word $u \in ({\Sigma'}_\Sigma)^*$ satisfies $\operatorname{ealph}(u) \times \operatorname{ealph}(u) \subseteq I'$, then $u \in (\Sigma')^*$.

Remark 2.2.3. If $\mathbb{M}(\Sigma',I') \cong \prod_{i=1}^{n} \mathbb{M}(\Sigma'_i,I'_i)$, where $n \in \mathbb{N}$, and Σ is an arbitrary finite alphabet, then $\mathbb{M}({\Sigma'}_\Sigma,{I'}_\Sigma) \cong \prod_{i=1}^{n} \mathbb{M}((\Sigma'_i)_\Sigma,(I'_i)_\Sigma)$.

As the relation $\sim_{{I'}_\Sigma}$ preserves extended content, the notion of extended content can be used also for traces.

Proposition 2.2.4. *Let $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ be any 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 $\sigma \circ \psi = \varphi$. Moreover, if the morphism φ is strong then the weak morphism ψ can be chosen strong too.*

Proof. For $x \in \Sigma$ let us denote $P_x = \mathscr{PR}(\varphi(x\sim_I))$ and

$$Q_x = \{s \in P_x \mid |\mathrm{alph}(s)| \geq 2\} , \qquad R_x = \{a \in \Sigma' \mid (a \sim_{I'}) \in P_x\} .$$

We construct a weak morphism ψ first. For every $A \in \overline{\mathscr{C}}(\Sigma',D')$ we choose any mapping $e_A : \Sigma \to \Sigma$ such that for all $x,y \in \Sigma$, $x \neq y$:

$$e_A(x) = e_A(y) \iff \exists s \in P_x \cap P_y : \ \mathrm{alph}(s) = A .$$

Such mappings e_A certainly exist because the defining condition always determines an equivalence relation on Σ. Now for every letter $x \in \Sigma$, consider the decomposition of the trace $\varphi(x\sim_I)$ into primitive roots of its connected components

$$\varphi(x\sim_I) = \left(\prod_{s\in Q_x} s^{i_s}\right) \cdot \left(\pi_{R_x}(\varphi(x\sim_I))\right)$$

and define

$$\psi(x) = \left(\prod_{s\in Q_x} \left(\mathrm{alph}(s), e_{\mathrm{alph}(s)}(x)\right)^{i_s}\right) \cdot \left(\pi_{R_x}(\varphi(x))\right) .$$

We prove that this mapping ψ really extends to a weak morphism. Let $(x,y) \in I$. Then $\varphi(x)\varphi(y) \sim_{I'} \varphi(y)\varphi(x)$ and Proposition 1.1.19 applied to the traces $\varphi(x\sim_I)$ and $\varphi(y\sim_I)$ gives

$$\forall s \in P_x,\ t \in P_y : \ s \neq t \implies \mathrm{alph}(s) \times \mathrm{alph}(t) \subseteq I' . \tag{2.10}$$

In order to verify $\mathrm{alph}(\psi(x)) \times \mathrm{alph}(\psi(y)) \subseteq {I'}_\Sigma$, take any elements $s \in Q_x$, $t \in Q_y$, $a \in R_x$, $b \in R_y$. We immediately deduce $(a,b) \in {I'}_\Sigma$ and $\left(a, \left(\mathrm{alph}(t), e_{\mathrm{alph}(t)}(y)\right)\right) \in {I'}_\Sigma$ from (2.10). If $s \neq t$ then $\mathrm{alph}(s) \times \mathrm{alph}(t) \subseteq I'$ holds by (2.10) and if $s = t$ then $e_{\mathrm{alph}(s)}(x) = e_{\mathrm{alph}(t)}(y)$ by the definition of $e_{\mathrm{alph}(s)}$. In both cases we obtain

$$\left(\left(\mathrm{alph}(s), e_{\mathrm{alph}(s)}(x)\right), \left(\mathrm{alph}(t), e_{\mathrm{alph}(t)}(y)\right)\right) \in {I'}_\Sigma .$$

Altogether, we get $\mathrm{alph}(\psi(x)) \times \mathrm{alph}(\psi(y)) \subseteq I'_\Sigma$. Trivially, the morphism ψ is also strong whenever φ is.

Now we define a strong morphism $\sigma : \mathbb{M}(\Sigma'_\Sigma, I'_\Sigma) \to \mathbb{M}(\Sigma', I')$. For $a \in \Sigma'$ let $\sigma(a) = a$ and for $(A,x) \in \Sigma'_\Sigma \setminus \Sigma'$ let

$$\sigma((A,x) \sim_{I'_\Sigma}) = \begin{cases} s & \text{if } \exists y \in \Sigma : \ s \in P_y, \ \mathrm{alph}(s) = A, \ e_A(y) = x \ , \\ 1 & \text{otherwise} \ . \end{cases}$$

Notice that the second rule is unambiguous due to the definition of e_A. Since we have $\mathrm{alph}(\sigma(\alpha)) \subseteq \mathrm{ealph}(\alpha)$ for every $\alpha \in \Sigma'_\Sigma$, by (2.9) this assignment defines a strong morphism. Finally, it is clear that $\varphi = \sigma \circ \psi$ holds. □

Proposition 2.2.5. *Let (Σ, I), (Σ', I') be independence alphabets. Then 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 $\mathbb{M}(\Sigma'_\Sigma, I'_\Sigma)$.

Proof. (i) $\Longrightarrow$ (ii). If $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ is a coding, by Proposition 2.2.4 there is a weak morphism $\psi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma'_\Sigma, I'_\Sigma)$ such that $\varphi = \sigma \circ \psi$ for some strong morphism $\sigma : \mathbb{M}(\Sigma'_\Sigma, I'_\Sigma) \to \mathbb{M}(\Sigma', I')$. Hence ψ is also injective.

(ii) $\Longrightarrow$ (iii) is trivial.

(iii) $\Longrightarrow$ (i). Let $\psi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma'_\Sigma, I'_\Sigma)$ be any coding. Because the mapping

$$H = \mathrm{ealph}|_{\Sigma'_\Sigma} : \Sigma'_\Sigma \to 2^{\Sigma'}$$

satisfies both conditions in Proposition 1.2.10 due to (2.8) and (2.9), there exists a strong coding $\sigma : \mathbb{M}(\Sigma'_\Sigma, I'_\Sigma) \to \mathbb{M}(\Sigma', I')$. Therefore we obtain a desired coding φ as the composition $\sigma \circ \psi$. □

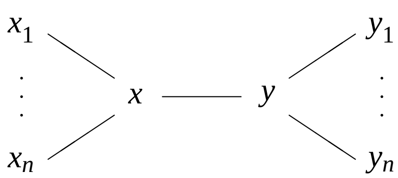
Example 2.2.6. As an example, let us employ Proposition 2.2.5 to characterize up to isomorphism all trace submonoids of the monoid $\mathbb{M}(\Sigma', I')$, where $\Sigma' = \{a,b,c,d\}$ and the dependence graph (Σ', D') is $a — b — c — d$ (hence the independence graph (Σ', I') is $c — a — d — b$). By Proposition 2.2.5 this task is the same as to find those trace monoids $\mathbb{M}(\Sigma, I)$ for which a weak coding $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma'_\Sigma, I'_\Sigma)$ exists.

First observe that the pairs of independent elements of Σ'_Σ are just (a,c), (a,d), $(a,(\{c,d\},x))$, (b,d) and $(d,(\{a,b\},x))$ for every $x \in \Sigma$. Since only two elements of Σ'_Σ, namely a and d, are independent on at least two other elements, there can be only two letters in Σ independent on at least two other letters and their images can contain only the letter a (d respectively); in particular, the graph (Σ, I) is acyclic.

Suppose there are $x,y,z \in \Sigma$ satisfying $\mathrm{alph}(\varphi(x)) = \{a\}$, $x\,D\,y$, $x\,D\,z$ and $y\,I\,z$. Then there is $\alpha \in \Sigma'_\Sigma$ independent on d such that $\mathrm{alph}(\varphi(y)) \cup \mathrm{alph}(\varphi(z)) \subseteq \{d,\alpha\}$ and $\alpha \in \mathrm{alph}(\varphi(y)) \cap \mathrm{alph}(\varphi(z))$. Consequently the word $u = y^{|\varphi(z)|_\alpha} x z^{|\varphi(y)|_\alpha}$ verifies the condition (ii) of Lemma 2.1.2 and so φ is not injective. Next, if there are $x,y,z,r \in \Sigma$ satisfying $x\,I\,y$, $z\,I\,r$ and $\{x,y\} \times \{z,r\} \subseteq D$, then $\mathrm{alph}(\varphi(xy)) \cap \mathrm{alph}(\varphi(zr)) = \emptyset$. Indeed, if $\alpha \in \mathrm{alph}(\varphi(xy)) \cap \mathrm{alph}(\varphi(zr))$ holds, then we have $\mathrm{alph}(\varphi(xy)) = \{\alpha,\beta\}$, $\beta \in \mathrm{alph}(\varphi(x)) \cap \mathrm{alph}(\varphi(y))$ and $\mathrm{alph}(\varphi(zr)) = \{\alpha,\gamma\}$ for some $\beta,\gamma \in \Sigma'_\Sigma$ and thus the condition (ii) of Lemma 2.1.2 holds for $u = x^{|\varphi(y)|_\beta} z y^{|\varphi(x)|_\beta}$.

Altogether, the independence graph (Σ, I) is of one of the following forms:

(i) One connected component of (Σ, I) is a subgraph of a graph of the form



and the other components are trivial.

(ii) The graph (Σ, I) consists of two connected components with two elements and arbitrarily many trivial ones.

On the other hand, in both cases some weak coding φ really exists: it is enough to define in the first case $\varphi(x) = a$, $\varphi(y) = d$, $\varphi(x_i) = (\{c,d\}, x_i)$ and $\varphi(y_i) = (\{a,b\}, y_i)$ and in the second case map the letters of the non-trivial components to the words b, bd, c and ac, and for any $z \in \Sigma$ forming a trivial component define $\varphi(z) = (\Sigma', z)$.

Corollary 2.2.7. *Let $\mathscr{C}$ be an arbitrary class of trace morphisms containing all weak codings. Then the $\mathscr{C}$-TCP restricted to saturated pairs of independence alphabets is equivalent to the TCP modulo effective reductions.*

Proof. Due to Proposition 2.2.5, given an instance of the TCP one can equivalently consider the corresponding saturated pair, which is easy to construct; and in the case of saturated pairs there exists a coding if and only if there exists a coding belonging to the class $\mathscr{C}$. □

Now we state the analogue of Proposition 2.2.5 for strong codings, which provides a reformulation of Proposition 1.2.10 characterizing the existence of strong codings using the notion of saturated pairs.

Proposition 2.2.8. *Let (Σ, I), (Σ', I') be independence alphabets. Then the following conditions are equivalent.*

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

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

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

(iv) *There exists a mapping* $H:\Sigma\to 2^{{\Sigma'}_\Sigma}$ *satisfying for every* $x,y\in\Sigma$:

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

Moreover, if H *is any such mapping then every mapping* $\varphi_0:\Sigma\to({\Sigma'}_\Sigma)^*$ *satisfying* $\mathrm{alph}(\varphi_0(x))=H(x)$ *for all* $x\in\Sigma$ *extends to a strong and weak coding from* $\mathbb{M}(\Sigma,I)$ *to* $\mathbb{M}({\Sigma'}_\Sigma,{I'}_\Sigma)$.

Proof. This can be proved by the same arguments as Proposition 2.2.5; the last claim and the equivalence of the conditions (ii) and (iv) follow from Lemma 2.1.8. □

Notice that in Proposition 2.2.5 there is the same domain monoid in each of the three conditions. This makes it suitable for showing decidability of the TCP for some classes of instances specified by properties of their domain monoids by means of proving the corresponding result for weak morphisms, which is the aim of Chapter 3. Let us now illustrate the usage of Proposition 2.2.5 by transferring Proposition 2.1.7 to the case of general codings.

Proposition 2.2.9. *Let* (Σ_i,I_i) *for* $i\in\{1,\ldots,n\}$ *and* (Σ',I') *be arbitrary 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 subalphabets* $\Sigma'_i\subseteq\Sigma'$, *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 $i\in\{1,\ldots,n\}$.

Proof. We have to prove only the direct implication, the converse is clear. Let Σ be the disjoint union of Σ_i. By Proposition 2.2.5 there exists a weak coding

$$\psi:\prod_{i=1}^n\mathbb{M}(\Sigma_i,I_i)\to\mathbb{M}({\Sigma'}_\Sigma,{I'}_\Sigma)\ .$$

Proposition 2.1.7 provides us with subsets $X'_i\subseteq{\Sigma'}_\Sigma$ such that $X'_i\times X'_j\subseteq{I'}_\Sigma\setminus\mathrm{id}_{{\Sigma'}_\Sigma}$ for every $i\neq j$ and with weak codings $\psi_i:\mathbb{M}(\Sigma_i,I_i)\to\mathbb{M}(X'_i,{I'}_\Sigma)$. Let us consider the alphabets $\Sigma'_i=\bigcup\{\mathrm{ealph}(\alpha)\mid\alpha\in X'_i\}$. Then $\Sigma'_i\times\Sigma'_j\subseteq I'\setminus\mathrm{id}_{\Sigma'}$ for $i\neq j$ due to (2.9) and $X_i' \subseteq (\Sigma_i')_\Sigma \subseteq \Sigma'_\Sigma$. Because the relation I'_Σ is defined on elements of $\Sigma'_\Sigma \setminus \Sigma'$ according to their first components, elements of $\Sigma'_\Sigma \setminus \Sigma'$ which have the same first components are mutually interchangeable in the alphabet $(\Sigma'_\Sigma, I'_\Sigma)$. Therefore we can replace second components of all elements of $\Sigma'_\Sigma \setminus \Sigma'$ occurring in some ψ_i-images with letters from Σ_i to obtain weak codings $\sigma_i : \mathbb{M}(\Sigma_i, I_i) \to \mathbb{M}((\Sigma_i')_{\Sigma_i}, I'_{\Sigma_i})$; notice that the relations I'_Σ and I'_{Σ_i} coincide on $(\Sigma_i')_{\Sigma_i}$ and that there are enough letters in $(\Sigma_i')_{\Sigma_i}$ to perform this replacement since the codings ψ_i are weak and $(A,x)\ D'_\Sigma\ (A,y)$ for every $(A,x),(A,y) \in \Sigma'_\Sigma \setminus \Sigma'$ with $x \neq y$. Applying Proposition 2.2.5 in the reverse direction, we get the required codings $\varphi_i : \mathbb{M}(\Sigma_i, I_i) \to \mathbb{M}(\Sigma_i', I')$. □

2.3 Co-strong Codings

Besides strong and weak morphisms, other kinds of trace morphisms can be naturally defined by introducing certain conditions on contents of images of letters. Studying restrictions of the TCP to these morphisms may shed some light on how the complex instances of this problem look like. This point of view, for instance, again underlines the simplicity of the $\mathscr{S}$-TCP: having an arbitrary additional effective condition on contents of images under strong morphisms, existence of such codings can be easily decided using Proposition 1.2.10.

In this section 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.

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.

Remark 2.3.2. Notice that performing the same construction for the definition of weak morphisms is not interesting with respect to the TCP due to the previously mentioned property of strong morphisms and Lemma 1.2.8.

Let us start with two simple observations about co-strong morphisms.

Lemma 2.3.3. *Let $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ and $\psi : \mathbb{M}(\Sigma', I') \to \mathbb{M}(\Sigma'', I'')$ be two co-strong morphisms. If the morphism ψ is in addition strong, then the composition $\psi \circ \varphi$ is co-strong too.* □

The next claim can be easily verified by means of Lemma 1.1.6.

Lemma 2.3.4. *Every co-strong and strong trace morphism φ which is non-erasing and satisfies (2.2) is a coding.* □

In the following, we prove that the construction of Section 2.2 can be performed as well for the problem of existence of co-strong codings, i.e. that the $\mathscr{CS}$-TCP is effectively reducible to the $\mathscr{CS} \cap \mathscr{W}$-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.

For a graph (V,E), we denote

$$\mathscr{C}(V,E) = \{X \subseteq V \mid X \neq \emptyset \text{ and } (X,E) \text{ is connected}\} .$$

Definition 2.3.5. Let (Σ,I) be an independence alphabet. We define the independence relation $\mathscr{P}(I)$ on the set $\mathscr{C}(\Sigma,D)$ by the rule:

$$(X,Y) \in \mathscr{P}(I) \iff X \times Y \subseteq I \text{ or } X = Y .$$

Remark 2.3.6. Notice that the independence graph $(\mathscr{C}(\Sigma',D'),\mathscr{P}(I'))$ is isomorphic to a subgraph of $({\Sigma'}_\Sigma,{I'}_\Sigma)$; singletons correspond to the copy of Σ' and each set $A \subseteq \Sigma'$ with $|A| \geq 2$ corresponds to (A,x) for an arbitrary $x \in \Sigma$.

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

$$\forall A,B \in \Sigma_1 : A \cap B \neq \emptyset \implies A = B , \tag{2.11}$$

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 $\sigma \circ \psi = \varphi$.

Remark 2.3.8. Instead of considering Σ_1 as a subset of $\mathscr{C}(\Sigma',D')$ we can view it as a subset of ${\Sigma'}_\Sigma$; the condition (2.11) should then be rephrased in the form

$$\forall \alpha,\beta \in \Sigma_1 : \mathrm{ealph}(\alpha) \cap \mathrm{ealph}(\beta) \neq \emptyset \implies \alpha = \beta . \tag{2.12}$$

Proof. It can be directly verified using Proposition 1.1.19 that the objects defined as follows possess the required properties. For every $x \in \Sigma$ denote $P_x = \mathscr{PR}(\varphi(x\sim_I))$ and if the decomposition of the trace $\varphi(x\sim_I)$ into primitive roots of its connected components is

$$\varphi(x\sim_I) = \prod_{s \in P_x} s^{i_s} ,$$

then assign:

$$\psi(x) = \prod_{s \in P_x} (\mathrm{alph}(s))^{i_s} .$$

Let the subalphabet be

$$\Sigma_1 = \{\mathrm{alph}(s) \mid \exists x \in \Sigma : s \in P_x\}$$

and for every $x \in \Sigma$ and $s \in P_x$ let $\sigma(\mathrm{alph}(s)\sim_{\mathscr{P}(I')}) = s$. $\square$

Proposition 2.3.9. *Let (Σ,I), (Σ',I') be independence alphabets. Then the following conditions are equivalent.*

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

(ii) *There exist some subalphabet $\Sigma_1 \subseteq \mathscr{C}(\Sigma',D')$ satisfying (2.11) and a co-strong and weak coding from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma_1,\mathscr{P}(I'))$.*

(iii) *There exist some subalphabet $\Sigma_1 \subseteq \mathscr{C}(\Sigma',D')$ satisfying (2.11) and a co-strong coding from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma_1,\mathscr{P}(I'))$.*

Proof. (i) $\Longrightarrow$ (ii) follows from Proposition 2.3.7.

(ii) $\Longrightarrow$ (iii) is trivial.

(iii) $\Longrightarrow$ (i). Let $\psi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma_1,\mathscr{P}(I'))$ be an arbitrary co-strong coding and let $\sigma : \mathbb{M}(\Sigma_1,\mathscr{P}(I')) \to \mathbb{M}(\Sigma',I')$ be any morphism such that $\mathrm{alph}(\sigma(A)) = A$ for every $A \in \Sigma_1$. Then σ is clearly non-erasing, co-strong, strong and satisfies (2.2), hence it is a coding by Lemma 2.3.4. Therefore we obtain a desired co-strong coding as the composition $\sigma \circ \psi$ due to Lemma 2.3.3. □

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.* □

Now we are going to show that if the domain dependence graph is C_3-free, then one can construct a co-strong coding whenever there exists an arbitrary coding. First we prove the analogue of this claim for weak codings.

Lemma 2.3.11. *Let (Σ,I) and (Σ',I') be independence alphabets such that the graph (Σ,D) is C_3-free. Then there exists a weak coding from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$ if and only if there exists a co-strong and weak coding from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$.*

Proof. Let $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ be an arbitrary weak coding. If we consider the decomposition of the domain monoid

$$\mathbb{M}(\Sigma,I) \cong \mathbb{M}(D(\Sigma),I) \times \mathbb{M}(\Sigma \setminus D(\Sigma),I) ,$$

then by Proposition 2.1.7 there exist some subalphabets $\Sigma'_1,\Sigma'_2 \subseteq \Sigma'$ of the codomain alphabet such that $\Sigma'_1 \times \Sigma'_2 \subseteq I' \setminus \mathrm{id}_{\Sigma'}$ and weak codings $\varphi_1 : \mathbb{M}(D(\Sigma),I) \to \mathbb{M}(\Sigma'_1,I')$ and $\varphi_2 : \mathbb{M}(\Sigma \setminus D(\Sigma),I) \to \mathbb{M}(\Sigma'_2,I')$. Without loss of generality, we can in addition assume $\Sigma'_1 = \bigcup\{\mathrm{alph}(\varphi_1(x)) \mid x \in D(\Sigma)\}$. Due to Lemma 2.1.4, the weak morphism $\pi_{D'(\Sigma'_1)} \circ \varphi_1 : \mathbb{M}(D(\Sigma),I) \to \mathbb{M}(D'(\Sigma'_1),I')$ is a coding. Moreover, it is also co-strong. With the aim of showing this fact by contradiction, let $x,y \in \Sigma$ satisfy $(x,y) \in D$ and let $a \in \mathrm{alph}(\varphi_1(x)) \cap \mathrm{alph}(\varphi_1(y)) \cap D'(\Sigma'_1)$. Then there is a letter $b \in \Sigma'_1 \cap D'(a)$ and consequently there is $z \in D(\Sigma)$ such that $b \in \mathrm{alph}(\varphi_1(z))$. But the weakness of the coding φ_1 implies $(x,z) \in D$ and $(y,z) \in D$, which is impossible as (Σ, D) is C_3-free. Since the weak coding φ_2 is trivially co-strong too, we obtain a required co-strong and weak coding from $\mathbb{M}(\Sigma, I)$ to $\mathbb{M}(\Sigma', I')$ as the product of $\pi_{D'(\Sigma_1')} \circ \varphi_1$ and φ_2. $\square$

In order to shift this result to the case of general morphisms we need a technical lemma.

Lemma 2.3.12. *Let (Σ, I) and (Σ', I') be independence alphabets and let $\Sigma_2 \subseteq \Sigma'_\Sigma$ be a subset such that the graph (Σ_2, D'_Σ) is C_3-free. Then there exist a subset $\Sigma_1 \subseteq \Sigma'_\Sigma$ satisfying (2.12) and a co-strong and strong coding from $\mathbb{M}(\Sigma_2, I'_\Sigma)$ to $\mathbb{M}(\Sigma_1, I'_\Sigma)$.*

Proof. The construction of the desired coding proceeds inductively with respect to the number $\sum_{\alpha \in \Sigma_2} |\mathrm{ealph}(\alpha)|$ using the following claim.

Claim 1. Let $\Sigma_2 \subseteq \Sigma'_\Sigma$ be a subset not satisfying (2.12) such that the graph (Σ_2, D'_Σ) is C_3-free. Then there exist $\Sigma_2' \subseteq \Sigma'_\Sigma$ such that (Σ_2', D'_Σ) is C_3-free and

$$\sum_{\alpha \in \Sigma_2'} |\mathrm{ealph}(\alpha)| < \sum_{\alpha \in \Sigma_2} |\mathrm{ealph}(\alpha)|$$

and a co-strong and strong coding from $\mathbb{M}(\Sigma_2, I'_\Sigma)$ to $\mathbb{M}(\Sigma_2', I'_\Sigma)$.

Proof. Consider any letters $\alpha, \beta \in \Sigma_2$ satisfying $\alpha \neq \beta$ and $\mathrm{ealph}(\alpha) \cap \mathrm{ealph}(\beta) \neq \emptyset$. We distinguish two cases.

First assume $\mathrm{ealph}(\alpha) = \mathrm{ealph}(\beta)$. Then $\alpha = (A, x)$ and $\beta = (A, y)$ for some letters $x, y \in \Sigma$, $x \neq y$, and some subset $A \subseteq \Sigma'$ such that $|A| \geq 2$ and the graph (A, D') is connected. Hence we can split the set A into disjoint non-empty subsets B and C such that both graphs (B, D') and (C, D') are connected. Now define $\alpha_1 = (B, x)$ if $|B| \geq 2$ and $\alpha_1 = b$ if $B = \{b\}$ and analogously $\beta_1 = (C, y)$ if $|C| \geq 2$ and $\beta_1 = c$ if $C = \{c\}$. Then consider the alphabet $\Sigma_2' = (\Sigma_2 \cup \{\alpha_1, \beta_1\}) \setminus \{\alpha, \beta\}$. Since we have $(\gamma, \alpha) \in I'_\Sigma$ and $(\gamma, \beta) \in I'_\Sigma$ for every $\gamma \in \Sigma_2 \setminus \{\alpha, \beta\}$ because (Σ_2, D'_Σ) is C_3-free, the graphs (Σ_2, D'_Σ) and (Σ_2', D'_Σ) are easily seen to be isomorphic.

Second, assume $\mathrm{ealph}(\alpha) \neq \mathrm{ealph}(\beta)$; say $\mathrm{ealph}(\alpha) \not\supseteq \mathrm{ealph}(\beta)$. Let us denote by B_i for $i = 1, \ldots, n$ the connected components of the graph $(\mathrm{ealph}(\beta) \setminus \mathrm{ealph}(\alpha), D')$. Let $x \in \Sigma$ be an arbitrary letter and let $\beta_i = (B_i, x)$ if $|B_i| \geq 2$ and $\beta_i = b_i$ if $B_i = \{b_i\}$. Take the subalphabet $\Sigma_2' = (\Sigma_2 \cup \{\beta_i \mid i = 1, \ldots, n\}) \setminus \{\beta\}$ and consider the morphism $\varphi : \mathbb{M}(\Sigma_2, I'_\Sigma) \to \mathbb{M}(\Sigma_2', I'_\Sigma)$ defined by the rule $\varphi(\beta) = \prod_{i=1}^n \beta_i$ and identical on the set $\Sigma_2 \setminus \{\beta\}$. The graph (Σ_2', D'_Σ) is C_3-free because for every $\gamma \in \Sigma_2 \setminus \{\beta\}$ we have $(\gamma, \beta) \in D'_\Sigma$ whenever $(\gamma, \beta_i) \in D'_\Sigma$ for some $i \in \{1, \ldots, n\}$. Due to the definition of β_i there exist letters $a_i \in \mathrm{ealph}(\alpha) \cap \mathrm{ealph}(\beta)$ and $b_i \in \mathrm{ealph}(\beta_i)$ such that $a_i \, D' \, b_i$. Therefore $(\alpha, \beta_i) \in D'_\Sigma$, and since plainly also $(\alpha, \beta), (\beta, \beta_i) \in D'_\Sigma$ holds and the graph (Σ_2, D'_Σ) is C_3-free, it follows that $\beta_i \notin \Sigma_2$. Thus φ is co-strong and strong. Now consider $\gamma \in \Sigma_2 \setminus \{\alpha, \beta\}$ satisfying $(\beta, \gamma) \in D'_\Sigma$. Then there exists some letter

$b \in \mathrm{ealph}(\beta) \cap D'(\mathrm{ealph}(\gamma))$. Because (Σ_2, D'_Σ) is C_3-free, $(\alpha, \gamma) \in I'_\Sigma$ holds. Hence $b \notin \mathrm{ealph}(\alpha)$ and for a certain $i \in \{1, \ldots, n\}$ we have $b \in \mathrm{ealph}(\beta_i)$, consequently $(\beta_i, \gamma) \in D'_\Sigma$ and $\varphi(\beta\gamma) \not\sim_{I'_\Sigma} \varphi(\gamma\beta)$. That is why φ fulfils (2.2). Altogether, φ is a coding by Lemma 2.3.4. □

Applying Claim 1 repeatedly and composing the codings constructed in each step, we eventually obtain a required subset $\Sigma_1 \subseteq \Sigma'_\Sigma$ satisfying (2.12) and a strong coding from $\mathbb{M}(\Sigma_2, I'_\Sigma)$ to $\mathbb{M}(\Sigma_1, I'_\Sigma)$, which is also co-strong due to Lemma 2.3.3. □

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 $\mathbb{M}(\Sigma', I')$.*

Proof. If there is a coding from $\mathbb{M}(\Sigma, I)$ to $\mathbb{M}(\Sigma', I')$, then by Proposition 2.2.5 there is a weak coding from $\mathbb{M}(\Sigma, I)$ to $\mathbb{M}(\Sigma'_\Sigma, I'_\Sigma)$. Using Lemma 2.3.11 we obtain a co-strong and weak coding $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma'_\Sigma, I'_\Sigma)$. As φ is weak and (Σ, D) is C_3-free, we in fact have $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma_2, I'_\Sigma)$ for a certain subset $\Sigma_2 \subseteq \Sigma'_\Sigma$ such that the graph (Σ_2, D'_Σ) is C_3-free. Now we employ Lemma 2.3.12 to get a co-strong and strong coding $\psi : \mathbb{M}(\Sigma_2, I'_\Sigma) \to \mathbb{M}(\Sigma_1, I'_\Sigma)$ for some subalphabet $\Sigma_1 \subseteq \Sigma'_\Sigma$ which satisfies (2.12). Then the composition $\psi \circ \varphi$ is a co-strong coding from $\mathbb{M}(\Sigma, I)$ to $\mathbb{M}(\Sigma_1, I'_\Sigma)$ due to Lemma 2.3.3. Finally, according to Remarks 2.3.6 and 2.3.8 we can apply Proposition 2.3.9 to obtain a co-strong coding from $\mathbb{M}(\Sigma, I)$ to $\mathbb{M}(\Sigma', I')$. □

Chapter 3

Decidable Cases

In this chapter 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 the monoid $\mathbb{M}(\Sigma,I)$ satisfying certain regularity conditions. This choice will be 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 the monoid $\mathbb{M}(\Sigma,I)$ generated by a subset of Σ on which the dependence relation forms a tree. Mappings satisfying these conditions will be called wlt-mappings.

We start by defining the notion of wlt-mappings and demonstrating some of their basic properties. In Section 3.2 we use wlt-mappings to deal with domain monoids whose dependence alphabets are acyclic and then in Section 3.3 we generalize this result, in the case of codomain monoids which are direct products of free monoids, to all C_3,C_4-free domain dependence alphabets. The final Section 3.4 is devoted to proving that in these results none of the assumptions on instances can be avoided.

3.1 Wlt-mappings

A crucial role in our considerations will be played by those letters of the codomain alphabet which occur in the image of exactly one generator of the domain monoid. Recall that according to Lemma 2.1.4 letters of the codomain alphabet independent on all letters occurring in the images are not significant for injectivity; that is why in the following definition of central letters they are excluded. Actually, since the process of reconstructing a word from its image under a weak coding by means of central letters is inductive, we have to consider central letters also for each subset of the domain alphabet.

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

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

and the set of central letters for X with respect to f as the (disjoint) union

$$\mathrm{C}_f^X = \bigcup_{x \in X} \mathrm{C}_f^X(x) \ .$$

Now we are ready to present the definition of wlt-mappings.

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 $\sigma_{X,x} : D(x) \cap X \to D'(\mathrm{C}_f^X(x))$ satisfying $\forall y \in D(x) \cap X :\ \sigma_{X,x}(y) \in f(y)$. Such letter x will be called X-*deciphering* for f.

Remark 3.1.3. The condition (W) ensures that every morphism constructed according to the mapping f satisfies (2.3). In what follows, when referring to (W) we mostly utilize only its direct implication; the converse implication is in fact a special case of (T) for 2-element subtrees.

Notice that the condition (L) in particular guarantees that $f(x) \neq \emptyset$ for every $x \in \Sigma$; this corresponds to the morphism property of being non-erasing.

Remark 3.1.4. It is clear from the definition that the restriction of any wlt-mapping from (Σ, I) to (Σ', I') to an arbitrary subalphabet of Σ is again a wlt-mapping.

Example 3.1.5. Take any $n \in \mathbb{N}$, $n \geq 3$, and let the dependence alphabet (Σ, D) be isomorphic to P_n with the i-th letter on the path denoted by x_i, for $i \in \{1, \ldots, n\}$. Further, consider the monoid $\mathbb{M}(\Sigma', I') = \{a_1, b_1\}^* \times \cdots \times \{a_{n-1}, b_{n-1}\}^*$. Then the rules $a_i \in f(x_i)$ and $b_i \in f(x_{i+1})$, for each $i \in \{1, \ldots, n-1\}$, define a wlt-mapping f from (Σ, I) to (Σ', I') where all vertices of every subtree X of the graph (Σ, D) are X-deciphering for f. Now choose any index $k \in \{2, \ldots, n-1\}$ and construct from f

a new mapping $g : \Sigma \to 2^{\Sigma'}$ by adding the letter a_i into $f(x_{i+2})$ for $i \in \{1,\ldots,k-1\}$ and b_i into $f(x_{i-1})$ for $i \in \{k,\ldots,n-1\}$. Then g is also a wlt-mapping from (Σ,I) to (Σ',I'), but only one Σ-deciphering letter exists for g, namely x_k. As we will see in Lemma 3.1.13, this behaviour of our wlt-mapping is not just a mere coincidence.

Example 3.1.6. Consider the dependence alphabet (Σ,D) isomorphic to the graph C_n, where $n \geq 5$, and let $\mathbb{M}(\Sigma',I') = \{a_1,b_1\}^* \times \cdots \times \{a_{n-1},b_{n-1}\}^*$. It is not hard to show that there are, up to symmetry, just two wlt-mappings from (Σ,I) to (Σ',I'); namely, if we denote the letters on the cycle (Σ,D) by $x_1,\ldots,x_n$ in a natural order, one of the mappings is defined by setting $a_i \in f(x_i)$ for all $i \in \{1,\ldots,n-1\}$, $b_1 \in f(x_n)$ and $b_i \in f(x_{i-1}) \cap f(x_{i+1})$ for all $i \in \{2,\ldots,n-1\}$, and the other one can be obtained by adding the letter b_1 into $f(x_2)$.

We start the detailed examination of the notions introduced in the above definitions with two simple observations.

Lemma 3.1.7. *Let (Σ,I), (Σ',I') be independence alphabets, $f : \Sigma \to 2^{\Sigma'}$ a mapping satisfying (W) and $X \subseteq \Sigma$ a subset such that the graph (X,D) is C_3,C_4-free.*

(i) *Let $x,y,z \in X$, $y \neq z$, be arbitrary letters and let letters $a \in f(x)$, $b \in f(y)$ and $c \in f(z)$ satisfy $(a,b),(a,c) \in D'$. Then $a \in \mathrm{C}^X_f(x)$.*

(ii) *If letters $a \in f(x)$ and $b \in f(y)$, where $x,y \in X$, satisfy $(a,b) \in D'$, then either $a \in \mathrm{C}^X_f(x)$ or $b \in \mathrm{C}^X_f(y)$.*

Proof. (i). Notice that if $a \in f(r)$ for some $r \in X$, then $(x,y),(x,z),(r,y),(r,z) \in D$ due to (W) and thus $r = x$ since (X,D) is C_3,C_4-free.

(ii). Assume $b \in \mathrm{C}^X_f(y)$ does not hold. Then there is $z \in X$, $z \neq y$, such that $b \in f(z)$. Hence $a \in \mathrm{C}^X_f(x)$ by (i). □

Lemma 3.1.8. *Let (Σ,I) and (Σ',I') be independence alphabets and $f : \Sigma \to 2^{\Sigma'}$ any mapping satisfying (W). Let $x \in X \subseteq \Sigma$ and let $Y \subseteq X$ be the subset consisting of all elements of X whose distance from the vertex x in the graph (X,D) is at most 2. Then $\mathrm{C}^X_f(x) = \mathrm{C}^Y_f(x)$ holds and therefore x is X-deciphering for f if and only if it is Y-deciphering for f.*

Proof. Consider $a \in \mathrm{C}^X_f(x)$. Then there exist $y \in X$ and $b \in f(y)$ such that $(a,b) \in D'$. Due to (W) we have $(x,y) \in D$ and so $y \in Y$. Hence $a \in \mathrm{C}^Y_f(x)$.

Conversely, if $a \in \mathrm{C}^Y_f(x)$ then for any $y \in X$ with $a \in f(y)$ we employ the existence of a letter $z \in Y$ with some $b \in f(z)$ satisfying $(a,b) \in D'$ to conclude $x\,D\,z$ and $z\,D\,y$ using (W). Thus $y \in Y$ and consequently $x = y$. □

Remark 3.1.9. From Lemma 3.1.8 one can see that if $X \subseteq \Sigma$ is an arbitrary subset and Y is a connected component of the graph (X,D), then $\mathrm{C}_f^X(x) = \mathrm{C}_f^Y(x)$ for all $x \in Y$. Therefore in the condition (T) of Definition 3.1.2 we can equivalently require (X,D) to be acyclic instead of a tree.

Now we state a lemma which provides two reformulations of the condition (T) of Definition 3.1.2. In contrast to (T), where the mappings $\sigma_{X,x}$ are defined locally on the set $D(x) \cap X$, these conditions require the existence of a suitable simultaneous choice of dependent letters from Σ' for all pairs of dependent letters from X.

Lemma 3.1.10. *If (Σ,I) and (Σ',I') are independence alphabets and $f : \Sigma \to 2^{\Sigma'}$ is any mapping satisfying (W), then the following statements are equivalent.*

(i) The mapping f satisfies (T).

(ii) For every $X \subseteq \Sigma$ such that (X,D) is a tree, there exists an injective mapping $\delta_X : D \cap (X \times X) \to D'$ satisfying the conditions:

(a) $\forall x,y \in X,\ x\,D\,y :\ \delta_X(x,y) = (a,b) \implies a \in f(x),\ b \in f(y)$.

(b) $\forall x,y \in X,\ x\,D\,y :\ \delta_X(x,y) = (a,b) \implies \delta_X(y,x) = (b,a)$.

(c) The dependence alphabet $(\Sigma', \mathrm{Im}(\delta_X))$ is acyclic.

(d) $\forall a \in \Sigma' \setminus \mathrm{C}_f^X :\ |\mathrm{Im}(\delta_X)(a)| \leq 1$, i.e. every letter from Σ' with at least two neighbours in the graph $(\Sigma', \mathrm{Im}(\delta_X))$ is central for X.

(iii) For every $X \subseteq \Sigma$ such that (X,D) is a tree, there exists an injective mapping $\tau_X : \{\{x,y\} \subseteq X \mid x\,D\,y\} \to \Sigma'$ satisfying for all $x,y \in X$, $x\,D\,y$:

$$\tau_X(\{x,y\}) \in f(x) \cap D'(\mathrm{C}_f^X(y)) \quad \textit{or} \quad \tau_X(\{x,y\}) \in f(y) \cap D'(\mathrm{C}_f^X(x))\,. \qquad (3.1)$$

Proof. (i) $\implies$ (ii). Let f satisfy (T). We construct required mappings δ_X inductively with respect to the cardinality of the set X. So, let $X \subseteq \Sigma$ be such that (X,D) is a tree. Choose an X-deciphering letter $x \in X$ for f and define δ_X for $y \in X$, $y\,D\,x$, by the rules $\delta_X(x,y) = (a, \sigma_{X,x}(y))$ and $\delta_X(y,x) = (\sigma_{X,x}(y), a)$, where $a \in \mathrm{C}_f^X(x) \cap D'(\sigma_{X,x}(y))$, and for $z,r \in X \setminus \{x\}$, $z\,D\,r$, by the rule $\delta_X(z,r) = \delta_Y(z,r)$, where Y is the connected component of $(X \setminus \{x\}, D)$ containing z and r.

The validity of the conditions (a) and (b) for δ_X is clear. In order to show that δ_X is injective, assume $\delta_X(y,z) = \delta_X(r,s) = (a,b)$ for some $y,z,r,s \in X$, $y\,D\,z$, $r\,D\,s$. If $a \in \mathrm{C}_f^X(x)$ then $y = r = x$ and $b = \sigma_{X,x}(z) = \sigma_{X,x}(s)$, which implies $z = s$ due to the injectivity of $\sigma_{X,x}$. Similarly one can deal with the case $b \in \mathrm{C}_f^X(x)$. Finally, if we have $a,b \notin \mathrm{C}_f^X(x)$ then $y,z,r,s \in X \setminus \{x\}$ and since $y\,D\,s$ follows from $a \in f(y)$, $b \in f(s)$ and $a\,D'\,b$ using (W), the letters y, z, r and s belong to the same connected component Y of $(X \setminus \{x\}, D)$ and we obtain $(y,z) = (r,s)$ from the injectivity of δ_Y.

With the aim of proving that the dependence alphabet $(\Sigma', \mathrm{Im}(\delta_X))$ is acyclic by means of contradiction, let us suppose there is a simple cycle $a_0, a_1, \ldots, a_n = a_0$ in $(\Sigma', \mathrm{Im}(\delta_X))$. Because $\sigma_{X,x}$ is injective, not all edges of this cycle lie in the set $\delta_X\big(D \cap ((X \times \{x\}) \cup (\{x\} \times X))\big)$. They also do not all belong to the same relation $\delta_X(D \cap (Y \times Y))$ for a connected component Y of the graph $(X \setminus \{x\}, D)$ thanks to the property (c) of δ_Y. Therefore we can assume without loss of generality that for some $i \in \mathbb{N}$, $1 \leq i \leq n-1$, and some connected component Y of $(X \setminus \{x\}, D)$ we have $(a_i, a_{i+1}), \ldots, (a_{n-1}, a_n) \in \delta_X(D \cap (Y \times Y))$ and $(a_0, a_1), (a_{i-1}, a_i) \notin \delta_X(D \cap (Y \times Y))$. Take the letters $y, z \in X$, $y \, D \, z$, and $r, s \in Y$, $r \, D \, s$, such that $\delta_X(y,z) = (a_0, a_1)$ and $\delta_X(r,s) = (a_{n-1}, a_n)$, Then the conditions (W) and (a) give $y \, D \, r$ and $z \, D \, s$ since $a_0 \, D' \, a_{n-1}$ and $a_1 \, D' \, a_n$. Thus $y, z \in X \cap D(Y) \subseteq Y \cup \{x\}$. The equality $y = x$ would imply $a_0 \in \mathrm{C}^X_f(x)$ due to the definition of δ_X, contradicting $a_0 \in f(s)$. Hence $z = x$ and y is the only element of $D(x) \cap Y$. The same arguments can be applied also to the pair (a_{i-1}, a_i), yielding $(a_{i-1}, a_i) = \delta_X(x,y) = (a_1, a_0)$. In particular $a_i = a_0$, which is impossible as the cycle is simple.

It remains to verify (d). Assume $(a,b), (a,c) \in \mathrm{Im}(\delta_X)$, where $a \in \Sigma' \setminus \mathrm{C}^X_f$. Then $(a,b) = \delta_X(y,z)$ and $(a,c) = \delta_X(r,s)$ for some $y, z, r, s \in X$, $y \, D \, z$, $r \, D \, s$. Because $a \notin \mathrm{C}^X_f$, $a \in f(y)$, $b \in f(z)$ and $c \in f(s)$, Lemma 3.1.7 (i) gives $z = s$. Since $a \notin \mathrm{C}^X_f$, according to the definition of the mapping δ_X we have neither $y = x$ nor $r = x$. If $z = x$ then $\sigma_{X,x}(y) = a = \sigma_{X,x}(r)$ by the definition of δ_X, which means $y = r$ due to the injectivity of $\sigma_{X,x}$ and we get $b = c$. Finally, in the case $y, z, r \in Y$ for a connected component Y of $(X \setminus \{x\}, D)$, either $y = r$, immediately leading to $b = c$, or $y \neq r$, which implies $a \notin \mathrm{C}^Y_f$ and consequently $b = c$ by the condition (d) for δ_Y.

(ii) $\Longrightarrow$ (iii). Let $X \subseteq \Sigma$ be a subset such that (X, D) is a tree. For each non-trivial connected component K of the dependence alphabet $(\Sigma', \mathrm{Im}(\delta_X))$ choose and fix an arbitrary letter $a_K \in K \cap \mathrm{C}^X_f$, which always exists due to Lemma 3.1.7 (ii). In order to define the mapping τ_X, take any $x, y \in X$ satisfying $x \, D \, y$. Then by the property (b) we have $\{\delta_X(x,y), \delta_X(y,x)\} = \{(b,c), (c,b)\}$ for some $(b,c) \in D'$. Let K be the connected component of $(\Sigma', \mathrm{Im}(\delta_X))$ containing b and c. Without loss of generality assume that c is the predecessor of b on the simple path from a_K to b in the tree $(K, \mathrm{Im}(\delta_X))$ and define $\tau_X(\{x,y\}) = b$. Since either $c = a_K$ holds or c has at least two neighbours in the graph $(\Sigma', \mathrm{Im}(\delta_X))$, the choice of the letter a_K and the condition (d) imply $c \in \mathrm{C}^X_f$. Therefore we get (3.1) from (a).

It remains to show that the mapping τ_X is injective. Let letters $x, y, z, r \in X$ satisfy $\tau_X(\{x,y\}) = \tau_X(\{z,r\})$. Then up to symmetry $\delta_X(x,y) = \delta_X(z,r) = (\tau_X(\{x,y\}), c)$, where c is the predecessor of $\tau_X(\{x,y\})$ on the simple path from a_K to $\tau_X(\{x,y\})$ in $(\Sigma', \mathrm{Im}(\delta_X))$. The injectivity of δ_X now gives $(x,y) = (z,r)$.

(iii) $\Longrightarrow$ (i). Let $X \subseteq \Sigma$ be such that (X, D) is a tree and let τ_X be a mapping which meets the requirements of (iii). Since the number of edges in (X, D) is less than the number of elements of X, we can choose a letter $x \in X$ satisfying $\tau_X(\{x,y\}) \in f(y)$ for all $y \in D(x) \cap X$ and define $\sigma_{X,x}(y) = \tau_X(\{x,y\})$. The injectivity of $\sigma_{X,x}$ is clear from the injectivity of τ_X and because $f(y) \cap D'(\mathrm{C}_f^X(y)) = \emptyset$ due to (W), the condition (3.1) implies $\sigma_{X,x}(y) \in f(y) \cap D'(\mathrm{C}_f^X(x))$. Hence x is an X-deciphering letter for f. $\square$

The following claim presents a simplification of the conditions of Lemma 3.1.10 in the case of codomain monoids which are direct products of free monoids.

Lemma 3.1.11. *Let (Σ, I) be an independence alphabet and let the monoid $\mathbb{M}(\Sigma', I')$ be isomorphic to the product $\prod_{i=1}^{n}(\Sigma_i')^*$ for pairwise disjoint alphabets $\Sigma_1', \dots, \Sigma_n'$. Let $f : \Sigma \to 2^{\Sigma'}$ be any mapping satisfying both conditions (W) and (L). Then f is a wlt-mapping from (Σ, I) to (Σ', I') if and only if for every subset $X \subseteq \Sigma$ such that (X, D) is a tree there exists an injective mapping*

$$\xi_X : \{\{x,y\} \subseteq X \mid x\,D\,y\} \to \{1, \dots, n\}$$

satisfying for all $x, y \in X$, $x\,D\,y$:

$$|(f(x) \cup f(y)) \cap \Sigma'_{\xi_X(\{x,y\})}| = 2\ . \tag{3.2}$$

Proof. Assuming validity of (W), we prove the equivalence of the condition (ii) of Lemma 3.1.10 and the condition given in this lemma. First notice that whenever (X, D) is C_3-free, the graph $(\bigcup f(X), D')$ is C_3-free too by (W), which means

$$\forall i \in \{1, \dots, n\} : \ \left|\bigcup f(X) \cap \Sigma_i'\right| \leq 2\ . \tag{3.3}$$

In particular, the graph $(\bigcup f(X), D')$ is acyclic and both conditions (c) and (d) are immediate consequences of (a).

"$\Longrightarrow$" Let $X \subseteq \Sigma$ be an arbitrary subset such that (X, D) is a tree and consider a mapping δ_X verifying the condition (ii) of Lemma 3.1.10. For any $x, y \in X$, $x\,D\,y$, we have $\delta_X(x,y) = (a,b)$, where $a, b \in \Sigma_i'$ for some $i \in \{1, \dots, n\}$. Then (b) ensures that we can correctly define $\xi_X(\{x,y\}) = i$. The condition (3.2) holds for this ξ_X thanks to (3.3) and (a). In order to show the injectivity of ξ_X, take any letters $x, y, z, r \in X$ satisfying $x\,D\,y$, $z\,D\,r$ and $\xi_X(\{x,y\}) = \xi_X(\{z,r\})$. Now the images $\delta_X(x,y)$, $\delta_X(z,r)$, $\delta_X(r,z)$ belong to $\Sigma'_{\xi_X(\{x,y\})} \times \Sigma'_{\xi_X(\{x,y\})}$ as well as to $\bigcup f(X) \times \bigcup f(X)$ by (a) and at the same time $\delta_X(z,r) \neq \delta_X(r,z)$ due to (b). Therefore the inequality (3.3) gives $\delta_X(x,y) \in \{\delta_X(z,r), \delta_X(r,z)\}$, which implies $\{x,y\} = \{z,r\}$ as δ_X is injective.

"$\Longleftarrow$" Assume we have a mapping ξ_X which meets the requirements given in the lemma and take any $x, y \in X$ satisfying $x\,D\,y$. Due to (3.2) and (W) there exist letters $a \in f(x) \cap \Sigma'_{\xi_X(\{x,y\})}$ and $b \in f(y) \cap \Sigma'_{\xi_X(\{x,y\})}$, $a\,D'\,b$, and we define $\delta_X(x,y) = (a,b)$. Then (a) is trivially valid and (b) is clear from (3.2). Finally, the injectivity of δ_X follows from the injectivity of ξ_X because of (b). $\square$

Remark 3.1.12. A significant advantage of the conditions stated in Lemmas 3.1.10 and 3.1.11 is that it suffices to verify the existence of δ_X, τ_X and ξ_X respectively only for maximal subtrees of (Σ,D) since the restrictions of these mappings to any subtree Y of X also satisfy the requirements.

Before proceeding to study relationships between wlt-mappings and weak codings, let us state a useful consequence of Lemma 3.1.10, which essentially says that once the condition (T) is violated, one cannot put it right by removing letters from the images. This provides significant help when one tries to verify non-existence of wlt-mappings.

Lemma 3.1.13. *Let f be a wlt-mapping from (Σ,I) to (Σ',I') and let $g:\Sigma\to 2^{\Sigma'}$ be any mapping satisfying (W) and $f(x)\subseteq g(x)$ for all $x\in\Sigma$. Then g is a wlt-mapping as well.*

Proof. It is clear that g satisfies (L). We verify the condition (ii) of Lemma 3.1.10 for g using the same mappings δ_X as for f. The only non-trivial task is to prove the validity of (d). Assume $a\in\Sigma'\setminus \mathsf{C}_g^X$ for a subtree X of (Σ,D) and let $(a,b),(a,c)\in \mathrm{Im}(\delta_X)$. Then $(a,b)=\delta_X(x,y)$ and $(a,c)=\delta_X(z,r)$ for certain letters $x,y,z,r\in X$. Because $a\in g(x)$, $b\in g(y)$, $c\in g(r)$, $a\notin \mathsf{C}_g^X$ and g satisfies (W), Lemma 3.1.7 (i) gives us that $y=r$ holds. If $a\in \mathsf{C}_f^X$ then we have $x=z$ as well and therefore $b=c$. And finally in the case $a\notin \mathsf{C}_f^X$ we obtain the desired equality $b=c$ from the condition (d) for f. $\square$

3.2 Acyclic Domain Dependence Alphabets

We start our calculations aiming to deal with instances of the TCP whose domain monoids have acyclic dependence alphabets by proving that in general the existence of a wlt-mapping is always necessary for the existence of a weak coding. In view of Proposition 2.2.5, the following claim is a generalization of Proposition 11 from [7], which states this fact for the simplest non-trivial case when $\Sigma=\{a_1,\dots,a_k,b\}$ and $D=\mathrm{sym}\{(a_1,b),\dots,(a_k,b)\}$. The construction performed in its proof is illustrated by Example 2.1.3.

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').*

Proof. Let us denote $f=\mathrm{alph}\circ\varphi|_\Sigma$. The condition (W) is clear from (2.3) and (L) is just a reformulation of Lemma 1.2.11. Suppose (T) does not hold. We will show that φ is not a coding. Let $X\subseteq\Sigma$ be a subset falsifying (T). Take an arbitrary $x\in X$. Then the system

$$\left(\pi_{D'(\mathsf{C}_f^X(x))}(\varphi(y\sim_I))\right)_{y\in D(x)\cap X} \tag{3.4}$$

of elements of the submonoid of $\mathbb{M}(\Sigma', I')$ generated by $\bigcup f(D(x) \cap X)$, which is free commutative, is linearly dependent. Therefore we can split $D(x) \cap X$ into disjoint sets L_x and R_x such that there exist $n_x^y \in \mathbb{N}_0$, for $y \in D(x) \cap X$, not all of them equal to 0, satisfying

$$\prod_{y \in L_x} \left(\pi_{D'(\mathrm{C}_f^X(x))} (\varphi(y)) \right)^{n_x^y} \sim_{I'} \prod_{y \in R_x} \left(\pi_{D'(\mathrm{C}_f^X(x))} (\varphi(y)) \right)^{n_x^y} . \tag{3.5}$$

Suppose we have fixed, for every $x \in X$ and $y \in D(x) \cap X$, some L_x, R_x and n_x^y satisfying (3.5). In addition, since the dependence relation restricted to X is acyclic, it is easy to fulfil the condition

$$\forall x, y \in X : \ y \in L_x \iff x \in R_y$$

by interchanging L_x with R_x for appropriate letters $x \in X$. Now we choose a subtree Y of (X, D), $|Y| \geq 2$, satisfying the condition

$$\forall x \in Y, \ y \in D(x) \cap X : \ n_x^y \neq 0 \iff y \in Y ; \tag{3.6}$$

if we understand X as a directed graph, where there is an arrow from x to $y \in D(x)$ if and only if $n_x^y \neq 0$, then we obtain Y as a terminal strongly connected component of this graph. Further, we take numbers $m_x \in \mathbb{N}$ for $x \in Y$ such that

$$\forall x, y, z \in Y : \ y, z \in D(x) \implies \frac{m_y}{m_z} = \frac{n_x^y}{n_x^z} ; \tag{3.7}$$

such numbers can be easily constructed inductively while adding vertices of the tree one by one, namely, when we add to the current subtree $Z \subseteq Y$ a vertex $y \in Y \setminus Z$ dependent on $x \in Z$, then we already have $m_z = (k/l) \cdot n_x^z$ for every $z \in D(x) \cap Z$ and for fixed integers $k, l \in \mathbb{N}$, so it is enough to multiply the values m_z for all $z \in Z$ by l and set $m_y = k \cdot n_x^y$.

Let $u \in Y^*$ be an arbitrary word consisting of exactly one x-block of length m_x for each letter $x \in Y$ such that for every $x, y \in Y$, if $(x, y) \in D$, then y precedes x in u if and only if $y \in L_x$. Clearly $u \not\sim_I \overleftarrow{u}$ since $|Y| \geq 2$. We will verify by means of Lemma 1.1.6 that $\varphi(u) \sim_{I'} \varphi(\overleftarrow{u})$ holds, thus reaching a contradiction with the injectivity of φ. Let $a, b \in \bigcup f(Y)$ satisfy $(a, b) \in D'$. Then at least one of these letters (say a) belongs to $\mathrm{C}_f^Y(x)$ for a certain $x \in Y$ by Lemma 3.1.7 (ii). Hence, the trace $\varphi(u) \sim_{I'}$ contains only one a-block and the number of occurrences of b preceding (succeeding) this block in $\varphi(u) \sim_{I'}$ is $\sum_{y \in L_x \cap Y} (m_y \cdot |\varphi(y)|_b)$, $\sum_{y \in R_x \cap Y} (m_y \cdot |\varphi(y)|_b)$ respectively. Due to (3.5), (3.6) and (3.7) these two numbers are equal and therefore $\pi_{a,b}(\varphi(u)) = \pi_{a,b}(\varphi(\overleftarrow{u}))$. $\square$

To be able to prove that in some cases the existence of a wlt-mapping suffices for producing a weak coding, we need a procedure converting the local regularity conditions given in Definition 3.1.2 into a global construction fully exploiting their potential. This is the aim of the following technical lemma.

Lemma 3.2.2. *Let $m, n \in \mathbb{N}$ and let $Z \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$ be any subset. Then there exists an $(m \times n)$-matrix $A = (a_{ij})$ over $\mathbb{N}_0$ such that*

(i) $a_{ij} = 0$ if and only if $(i, j) \in Z$,

(ii) for every $M \subseteq \{1, \ldots, m\}$ and $N \subseteq \{1, \ldots, n\}$ with $|M| = |N|$, the submatrix $A(M, N)$ of A is regular if and only if there exists a bijection $\tau : M \to N$ which satisfies $(i, \tau(i)) \notin Z$ for all $i \in M$.

Proof. Without loss of generality we can assume $m \leq n$. Consider the increasing sequence S of natural numbers $s_k = m^{m^{2k}}$, for $k \in \mathbb{N}_0$, and complete the matrix A on non-Z coordinates with arbitrary elements of this sequence, taking each of them only once. The "only if" part of the condition (ii) is clear since if there is not such a bijection, then $\det(A(M, N)) = 0$ due to the first condition. For the converse, let us consider the products $p_\tau = \prod_{i \in M} a_{i\tau(i)}$ which make up the determinant of $A(M, N)$, for all bijections $\tau : M \to N$. By the assumption, not all of these products are zero. Take any two distinct bijections $\sigma, \tau : M \to N$ satisfying $p_\sigma \neq 0 \neq p_\tau$. Let $k \in \mathbb{N}_0$ be the greatest integer such that the number s_k lies in one of the sets $\{a_{i\sigma(i)} \mid i \in M\}$ and $\{a_{i\tau(i)} \mid i \in M\}$ (say in the former one) but not in the other. Then $k \geq 1$ and we can easily calculate $p_\sigma / p_\tau \geq s_k / s_{k-1}^m \geq m!$. As there are at most $m!$ non-zero products summed in the determinant, the greatest one is bigger than the sum of the others and thus $\det(A(M, N)) \neq 0$. $\square$

In order to employ this lemma for our purposes, we have to understand every weak morphism $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ as a $(\Sigma \times \Sigma')$-matrix $(|\varphi(x)|_a)_{x \in \Sigma, a \in \Sigma'}$ over $\mathbb{N}_0$. Since $\mathrm{alph} \circ \varphi|_\Sigma = f$ can be rephrased as

$$|\varphi(x)|_a \neq 0 \iff a \in f(x) ,$$

a wlt-mapping f just determines non-zero entries of this matrix and the conditions (L) and (T) provide regularity of certain submatrices when φ is constructed using Lemma 3.2.2.

Proposition 3.2.3. *Let (Σ, I) and (Σ', I') be independence alphabets satisfying one of the following conditions.*

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

(ii) 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').

Remark 3.2.4. The first case was already solved for general trace morphisms in [7]; notice that in this case the requirement (T) is redundant.

Proof. The "only if" part is a direct consequence of Lemma 3.2.1.

Let f be any wlt-mapping from (Σ, I) to (Σ', I'). We prove the converse implication in the case (i) first. Let us assume $\mathbb{M}(\Sigma, I) = (\Sigma_1)^* \times \cdots \times (\Sigma_n)^*$. Due to (W) and (L), we can use Lemma 3.2.2 to construct some weak morphism $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ satisfying $\mathrm{alph} \circ \varphi|_\Sigma = f$ such that for all subsets $X \subseteq \Sigma$ with $X \times X \subseteq I$, the system $(\varphi(x))_{x \in X}$ of elements of the free commutative monoid generated by the set $\bigcup f(X)$ is linearly independent. This shows that the case (i) of Lemma 2.1.2 cannot happen. Because (W) makes the case (ii) of Lemma 2.1.2 also impossible by Lemma 2.1.8 applied to the monoids $(\Sigma_i)^*$ for $i \in \{1, \ldots, n\}$ with $|\Sigma_i| > 1$, we conclude that φ is a coding.

In the case (ii), we employ all conditions of Definition 3.1.2 and Lemma 3.2.2 to construct a weak morphism $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ such that the same conditions as in the previous case are satisfied and in addition for all $X \subseteq \Sigma$ such that (X, D) is a tree, there is $x \in X$ for which the system (3.4) is linearly independent. As these conditions are valid also for each induced subgraph of (Σ, D), to prove that φ is a coding we can use induction with respect to the number of vertices of the graph. So, let us assume that φ is injective on $\mathbb{M}(Y, I)$ for each proper subset Y of Σ. If the graph (Σ, D) has no edges, then φ is a coding due to the condition for independence cliques. If it is not the case, let X be one of its non-trivial connected components and consider $x \in X$ such that (3.4) is linearly independent. Let $u, v \in \Sigma^*$ satisfy $\varphi(u) \sim_{I'} \varphi(v)$. Since X is a connected component of (Σ, D), due to (W) letters from $\mathsf{C}^X_f(x)$ occur just in the image of x and letters from $D'(\mathsf{C}^X_f(x))$ just in the images of elements of $D(x) \subseteq X$. As $\mathsf{C}^X_f(x) \neq \emptyset$, the equality $|u|_x = |v|_x$ holds. Further we deduce

$$\pi_{\mathsf{C}^X_f(x) \cup D'(\mathsf{C}^X_f(x))}\left(\varphi\left(\pi_{\{x\} \cup D(x)}(u)\right)\right) \sim_{I'} \pi_{\mathsf{C}^X_f(x) \cup D'(\mathsf{C}^X_f(x))}\left(\varphi\left(\pi_{\{x\} \cup D(x)}(v)\right)\right)$$

and the independence of the system (3.4) gives us $\pi_{x,y}(u) = \pi_{x,y}(v)$ for every $y \in D(x)$. Because φ is injective on the submonoid $\mathbb{M}(\Sigma \setminus \{x\}, I)$, we can apply Lemma 2.1.10 to conclude $u \sim_I v$. Thus φ is a coding. □

3.3 Encoding into Direct Products of Free Monoids

Now we are going to 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 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 2.2.1, we consequently obtain the corresponding positive result for the TCP by applying Proposition 2.2.5. As the following example shows, unlike in the cases covered by Proposition 3.2.3, in this situation it is not true that for every wlt-mapping f there exists a coding φ such that $\mathrm{alph} \circ \varphi|_\Sigma = f$, therefore a modification of the wlt-mapping in order to make it suitable for a construction of a coding is unavoidable.

Example 3.3.1. On the alphabet $\Sigma = \{x, y, z, r, s, t\}$ define the dependence relation D by the graph

$$\begin{tikzcd}
 & & z \arrow[r, dash] & r \arrow[dd, dash] \\
x \arrow[r, dash] & y \arrow[ur, dash] \arrow[dr, dash] & & \\
 & & t \arrow[r, dash] & s
\end{tikzcd}$$

and let

$$\mathbb{M}(\Sigma', I') = \{a_1, a_2\}^* \times \{b_1, b_2\}^* \times \{c_1, c_2\}^* \times \{d_1, d_2\}^* \times \{e_1, e_2\}^* .$$

Consider the wlt-mapping f from (Σ, I) to (Σ', I') given by the rules:

$$\begin{array}{lll}
f(x) = \{a_1, b_1\} & f(y) = \{a_2, b_2, c_1, d_1\} & f(z) = \{a_1, c_2, e_1\} \\
f(r) = \{c_1, e_2\} & f(s) = \{d_1, e_1\} & f(t) = \{b_1, d_2\} .
\end{array}$$

When proving that every morphism $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ satisfying $\mathrm{alph} \circ \varphi|_\Sigma = f$ fails to be a coding, we can assume

$$|\varphi(x)|_{a_1} = |\varphi(y)|_{d_1} = |\varphi(z)|_{a_1} = |\varphi(s)|_{d_1} = 1$$

by Lemma 2.1.17 and

$$|\varphi(y)|_{a_2} = |\varphi(y)|_{b_2} = |\varphi(z)|_{c_2} = |\varphi(r)|_{d_2} = |\varphi(t)|_{e_2} = 1$$

since all of these letters occur only in one of the images. Thus

$$\begin{array}{lll}
\varphi(x) = a_1 b_1^i & \varphi(y) = a_2 b_2 c_1^j d_1 & \varphi(z) = a_1 c_2 e_1^k \\
\varphi(r) = c_1^l e_2 & \varphi(s) = d_1 e_1^m & \varphi(t) = b_1^n d_2
\end{array}$$

for some $i, j, k, l, m, n \in \mathbb{N}$. Then $u = x^{lmn} y^{kln} z^{lmn} r^{jkn} t^{ilm} s^{kln}$ satisfies $\varphi(u) \sim_{I'} \varphi(\overleftarrow{u})$, which shows that φ is not injective.

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. Let us first introduce a requirement on wlt-mappings which is sufficient to avoid counter-examples of the second form of Lemma 2.1.2.

Lemma 3.3.2. *Let $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ be an arbitrary weak morphism and let us denote $f = \mathrm{alph} \circ \varphi|_\Sigma$. If every connected component of the graph (Σ, D) contains at most one vertex $x \in \Sigma$ such that $f(y) \cap D'(\mathrm{C}_f^\Sigma(x)) = \emptyset$ for some vertex $y \in D(x)$, then the condition (ii) of Lemma 2.1.2 does not hold.*

Proof. Assume that a word $u \in \Sigma^*$ verifies the condition (ii) of Lemma 2.1.2. Then for every $x \in \mathrm{init}(u\sim_I)$, the last occurrence of x in u is succeeded by an occurrence of some $y \in D(x)$, because $\mathrm{init}(u\sim_I) \cap \mathrm{fin}(u\sim_I) = \emptyset$. If there exist letters $a \in \mathrm{C}_f^\Sigma(x)$ and $b \in f(y) \cap D'(a)$, then $\pi_{a,b}(\varphi(u)) \neq \pi_{a,b}(\varphi(\overleftarrow{u}))$, contradicting $\varphi(u) \sim_{I'} \varphi(\overleftarrow{u})$. Therefore $f(y) \cap D'(\mathrm{C}_f^\Sigma(x)) = \emptyset$ holds. Due to the symmetry, the same fact can be deduced also for $x \in \mathrm{fin}(u\sim_I)$. As $u\sim_I$ is connected, by the assumption of the lemma this implies that only one element of $\mathrm{alph}(u)$ can belong to $\mathrm{init}(u\sim_I) \cup \mathrm{fin}(u\sim_I)$, which is impossible since $\mathrm{init}(u\sim_I) \cap \mathrm{fin}(u\sim_I) = \emptyset$. $\square$

For counting of letters in the images we use the following property of graphs with vertices valuated by non-negative integers.

Lemma 3.3.3. *Let (V,E) be a connected undirected graph together with a valuation $\eta : V \to \mathbb{N}_0$ of its vertices. Assume that for every $X \subseteq V$ such that (X,E) is a tree, there exists a vertex $x \in X$ satisfying $\eta(x) \geq |E_X(x)|$, where $E_X(x) = \{y \in X \mid (x,y) \in E\}$. Then $\sum_{x\in V} \eta(x) \geq |V| - 1$.*

Moreover, for every undirected graph (V,E), such a valuation $\eta : V \to \mathbb{N}_0$ satisfying the equality $\sum_{x\in V} \eta(x) = |V| - 1$ exists.

Proof. The second claim can be obtained by setting $\eta(x) = 1$ for all vertices $x \in V$ except one; then every non-trivial subtree of (V,E) contains at least one leaf x with the value $\eta(x) = 1$.

We are going to prove the first claim through a contradiction. For this purpose, let us consider some graph (V,E) and its valuation $\eta : V \to \mathbb{N}_0$ falsifying the claim such that the number $|\eta^{-1}(0)|$ is the smallest possible, where $\eta^{-1}(0)$ stands for the set $\{x \in V \mid \eta(x) = 0\}$. Clearly $|\eta^{-1}(0)| \geq 2$. Additionally, assume that this graph possesses the minimal shortest distance between distinct zero-valuated vertices among such counter-examples. Take some vertices $y,z \in V$ satisfying $\eta(y) = \eta(z) = 0$ whose distance d is minimal. Then $d \geq 2$; otherwise $X = \{y,z\}$ contradicts the assumption. Let s be the successor of y on some shortest path to z and consider the valuation $\vartheta : V \to \mathbb{N}_0$ defined by the rules $\vartheta(y) = 1$, $\vartheta(s) = \eta(s) - 1 \geq 0$ and $\vartheta(x) = \eta(x)$ for every $x \in V \setminus \{y,s\}$. Then either $\vartheta(s) \geq 1$ and therefore $|\vartheta^{-1}(0)| < |\eta^{-1}(0)|$ or $\vartheta(s) = \vartheta(z) = 0$ and the distance between s and z is $d-1$. We will show that the valuation ϑ satisfies the assumptions of the lemma, thus contradicting the choice of η since $\sum_{x\in V} \vartheta(x) = \sum_{x\in V} \eta(x)$.

Let $X \subseteq V$ be such that (X,E) is a tree. It is enough to deal with the case $s \in X$. First assume $y \in X$. If y is a leaf of X, it is a required vertex. Otherwise a required vertex can be obtained using the condition for η on the maximal subtree of X which contains the vertex y and does not contain s. In the case $y \notin X$, we also distinguish two situations. If $E_X(y) = \{s\}$ then $Y = X \cup \{y\}$ is a tree and each vertex $x \in Y$ satisfying $\eta(x) \geq |E_Y(x)|$ lies in X and satisfies $\vartheta(x) \geq |E_X(x)|$. If $|E_X(y)| \geq 2$, take any vertex $t \in E_X(y)$ having the maximum distance from s in (X,E) and let Y be the subtree of X consisting of all vertices such that the shortest paths connecting them with s in (X,E) contain t. Then we get a required vertex by applying the condition for the valuation η to the tree $Y \cup \{y\}$. □

Now we are ready to prove the main result of this section.

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 $i \in \{1, \dots, m\}$ and c_i for $i \in \{1, \dots, n\}$ 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) $|\Sigma| - M - N \leq m$ and $|\Sigma| - M \leq m + n$.

Proof. (i) $\Longrightarrow$ (ii) and (iii) $\Longrightarrow$ (i) are trivial.

(ii) $\Longrightarrow$ (iii). First notice that $\mathbb{M}(\Sigma'_\Sigma, I'_\Sigma)$ is also a direct product of m free monoids over at least two generators and n free one-generated monoids. By Proposition 2.2.5 there exists some weak coding from $\mathbb{M}(\Sigma, I)$ to $\mathbb{M}(\Sigma'_\Sigma, I'_\Sigma)$. And because a weak morphism from a trace monoid having C_3-free dependence alphabet cannot employ three mutually dependent letters of the codomain alphabet, the codomain monoid in the condition (iii) is sufficient.

(iii) $\Longrightarrow$ (iv) follows from Lemma 3.2.1.

(iv) $\Longrightarrow$ (v). Let f be a wlt-mapping from (Σ, I) to (Σ_1, I_1). First, we have to slightly modify f. For every $i \in \{1, \dots, m\}$ such that $a_i, b_i \in \bigcup f(\Sigma)$, we assume that $a_i \in \mathsf{C}_f^\Sigma(x)$ for a certain letter $x \in \Sigma$ (this is possible due to Lemma 3.1.7 (ii) since a_i and b_i are interchangeable) and then we add b_i into the f-images of all letters from $D(x)$ using Lemma 3.1.13. Let us now denote $A_x = \{a_i \mid i = 1, \ldots, m\} \cap \mathsf{C}^{\Sigma}_f(x)$ for every $x \in \Sigma$.

Let Y be any connected component of the graph (Σ, D). Consider an arbitrary subset $X \subseteq Y$ such that (X, D) is a tree and let x be an X-deciphering letter for f. If there exists some $b_i \in \mathsf{C}^X_f(x)$, then the letter $y \in X$ whose f-image contains a_i is a leaf of (X, D) thanks to the initial modifications of our wlt-mapping. Otherwise we have $\mathsf{C}^X_f(x) \subseteq A_x$ and consequently $|A_x| \geq |D'(\mathsf{C}^X_f(x))| \geq |D(x) \cap X|$ because x is X-deciphering for f. In both cases we find a letter $y \in X$ such that $|A_y| \geq |D(y) \cap X|$. Therefore Lemma 3.3.3 can be employed to conclude $\sum_{x \in Y} |A_x| \geq |Y| - 1$. Summing these inequalities for all connected components of (Σ, D), we obtain

$$|\Sigma| - M - N \leq \sum_{x \in \Sigma} |A_x| \tag{3.8}$$

and the first inequality of the condition (v) follows from $\sum_{x \in \Sigma} |A_x| \leq m$, which holds since $A_x \cap A_y = \emptyset$ for $x \neq y$. Next, notice that every $x \in \Sigma$ such that $D(x) = \emptyset$ satisfies $f(x) \cap \{a_i, b_i \mid a_i \in \mathsf{C}^{\Sigma}_f\} = \emptyset$. So the number of trivial components of the graph (Σ, D) is at most $m + n - \sum_{x \in \Sigma} |A_x|$ by Lemma 1.2.11, which gives the second inequality of (v) due to (3.8).

(v) $\Longrightarrow$ (iii). Let us first construct a suitable wlt-mapping f from (Σ, I) to (Σ_1, I_1). We put each of the letters a_i and c_i into the f-image of at most one element of Σ to satisfy the following:

- For every non-trivial connected component Y of the graph (Σ, D), the image of all but one element of Y contains exactly one letter a_i, the remaining one contains neither a_i nor c_i.
- The image of each $x \in \Sigma$ satisfying $D(x) = \emptyset$ contains just one letter a_i or c_i.

The condition (v) ensures that this construction can be done. Then for every index $i \in \{1, \ldots, m\}$ and $y \in \Sigma$ such that $a_i \in f(y)$, we add b_i into $f(x)$ for all $x \in D(y)$. Notice that f is really a wlt-mapping since for every clique X in (Σ, I) one can define a mapping ρ_X by setting $\rho_X(x)$ equal to the letter a_i or c_i contained in $f(x)$ if such a letter exists, and equal to some b_i if x is the exceptional vertex of a non-trivial connected component of the graph (Σ, D), and the condition (T) is valid because every non-trivial subtree of (Σ, D) possesses a leaf with a central letter in its f-image.

Now we use Lemma 3.2.2 to construct a weak morphism $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma_1, I_1)$ satisfying $\mathrm{alph} \circ \varphi|_{\Sigma} = f$ in the same way as in the proof of Proposition 3.2.3. Clearly the case (i) of Lemma 2.1.2 cannot occur for the morphism φ. And the case (ii) of Lemma 2.1.2 is also impossible by Lemma 3.3.2 since all a's used in non-trivial connected components of (Σ, D) are central for Σ. Hence φ is a coding. $\square$

The problem of existence of trace codings into direct products of free monoids (posed in [14]) was already tackled in [17], where it was solved in the case of (Σ, D)

being a path or a cycle. Proposition 3.3.4 settles this problem for all domain monoids with C_3, C_4-free dependence alphabets. But as we will see in the next section, in more general situations the technique of wlt-mappings fails.

3.4 Counter-Examples

The aim of this section is to show that none of the assumptions of Proposition 3.3.4 can be removed. The first example demonstrates that the restriction on the codomain monoid cannot be avoided because the existence of a coding is not guaranteed by the existence of a wlt-mapping on the corresponding saturated pair of independence alphabets even for domain monoids which have C_3, C_4-free dependence alphabets.

Example 3.4.1. Let

$$\Sigma_0 = \{x_1, x_2, x_3, x_4, x_5, r\} \cup \{y_A, z_A \mid A \subset \{1,2,3,4,5\},\ |A| = 2\}$$

and consider the alphabet

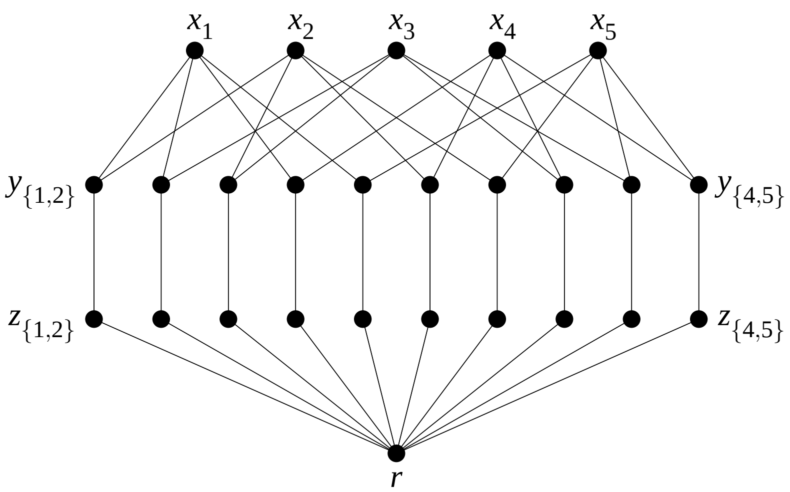
$$\Sigma = \Sigma_0 \cup \big(\{r, y_A\} \times \{1,2,3,4\}\big) \cup \big(\{z_A\} \times \{1,2\}\big)\ ,$$

where A always runs through the same values as above. Define a dependence relation on Σ as follows. For $i = 1, \dots, 5$:

$$x_i\, D\, y_A \iff i \in A\ ,$$

$$\begin{array}{cccc} y_A\, D\, z_A, & z_A\, D\, r, & z_A\, D\, (z_A, 1), & (z_A,1)\, D\, (z_A,2)\ , \\ y_A\, D\, (y_A,1), & (y_A,1)\, D\, (y_A,2), & y_A\, D\, (y_A,3), & (y_A,3)\, D\, (y_A,4)\ , \\ r\, D\, (r,1), & (r,1)\, D\, (r,2), & r\, D\, (r,3), & (r,3)\, D\, (r,4)\ . \end{array}$$

The remaining pairs are left independent. The picture of the Σ_0-part of the dependence graph is



with at most two simple paths of length 2 attached to each vertex. Let the codomain monoid $\mathbb{M}(\Sigma',I')$ be isomorphic to $\mathbb{M}(\Sigma_0',I_0')^{32}$, where $|\Sigma_0'| = 4$ and the relation I_0' (and hence also D_0') is defined by the graph P_4, i.e. the codomain dependence graph (Σ',D') consists of 32 copies of P_4.

We will prove that there exists a wlt-mapping from (Σ,I) to $({\Sigma'}_\Sigma,{I'}_\Sigma)$ but still there is no coding from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$. In our arguments we implicitly use Lemma 3.2.1 and instead of weak morphisms we often talk about the corresponding wlt-mappings. First observe that every C_3-free subgraph of the graph $((\Sigma_0')_\Sigma,(D_0')_\Sigma)$ is isomorphic to some subgraph of P_4 (see Example 2.2.6). Because (Σ,D) is C_3-free, the graph $(\bigcup f(\Sigma),{D'}_\Sigma)$ is C_3-free for every wlt-mapping f from (Σ,I) to $({\Sigma'}_\Sigma,{I'}_\Sigma)$ and so we can employ Remark 2.2.3 to find out that any weak coding from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}({\Sigma'}_\Sigma,{I'}_\Sigma)$ can be turned into a weak coding from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$ by renaming letters in the images. Therefore, in order to conclude that a coding from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$ does not exist, it is sufficient to verify non-existence of weak codings between these monoids and then apply Proposition 2.2.5.

Consider any wlt-mapping f from (Σ,I) to (Σ',I'). The condition (W) implies that the letters from $D'(\bigcup f(\Sigma))$ belonging to the same copy of P_4 in (Σ',D') are used in f-images just in some connected subgraph of (Σ,D) with maximal distance of its vertices no more than 3. Consequently, for distinct paths attached to elements of Σ_0 in the graph (Σ,D), the pairs of dependent letters from Σ' creating dependences between the two vertices on these paths come from different copies of P_4. As the number of attached paths equals to the number of copies of P_4, the set $\bigcup f(\Sigma_0)\cap D'(\bigcup f(\Sigma_0))$ can contain at most two letters from each copy of P_4. Moreover, as soon as two dependent letters from one copy of P_4 are used in f-images of some elements of Σ_0, one of them occurs in the image of the element of Σ_0 the corresponding path in (Σ,D) leads to and the other in the images of some of the neighbouring letters. Hence, if a weak coding from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$ exists, by Lemma 2.1.4 there is also a weak coding from $\mathbb{M}(\Sigma_0,I)$ to $(\{a,b\}^*)^{32}$ with the number of a's from distinct copies of $\{a,b\}^*$ occurring in the image of a given element of Σ_0 bounded by the number of 2-element paths attached to this element in (Σ,D). From now on, we consider only weak codings of this form.

Let us now summarize the properties of the restriction of the wlt-mapping f to Σ_0 we have discovered so far, together with introducing more transparent notation for elements of Σ' :

$$\begin{aligned} f(x_i) &\subseteq \{a_A,b_A \mid i\in A\} & f(y_A) &\subseteq \{\bar{a}_A,\bar{b}_A,c_A\} \\ f(z_A) &\subseteq \{a_A,b_A,\bar{c}_A,d,e\} & f(r) &\subseteq \{c_A,\bar{d},\bar{e}\}\ , \end{aligned} \tag{3.9}$$

where the pairs of dependent letters in Σ' are $a_A\ D'\ \bar{a}_A$, $b_A\ D'\ \bar{b}_A$, $c_A\ D'\ \bar{c}_A$, $d\ D'\ \bar{d}$ and $e\ D'\ \bar{e}$. It is not hard to verify that there really exists a wlt-mapping from (Σ,I) to (Σ',I') by taking equalities for all of these inclusions and extending to Σ naturally.

But in the following we prove that none of the morphisms

$$\varphi : \mathbb{M}(\Sigma_0, I) \to \prod_A \left(\{a_A, \overline{a}_A\}^* \times \{b_A, \overline{b}_A\}^* \times \{c_A, \overline{c}_A\}^* \right) \times \{d, \overline{d}\}^* \times \{e, \overline{e}\}^*$$

satisfying $\mathrm{alph} \circ \varphi|_{\Sigma_0} = f|_{\Sigma_0}$ is injective.

First observe that (T) is not valid when either $c_A \notin f(y_A)$ or $c_A \notin f(r)$ for some A; for instance, in the case $c_{\{1,2\}} \notin f(r)$ we consider the tree

$$X = \{x_1, x_2, x_3, x_4, y_{\{1,3\}}, y_{\{2,4\}}, z_{\{1,2\}}, z_{\{1,3\}}, z_{\{2,4\}}, r\} ,$$

verify that it is a counter-example for the mapping f defined by taking equalities in (3.9) except $f(r) = \{c_A, \overline{d}, \overline{e} \mid A \neq \{1,2\}\}$ and apply Lemma 3.1.13. Let us denote:

$$k_{ij} = |\varphi(y_{\{i,j\}})|_{c_{\{i,j\}}} \in \mathbb{N} , \qquad\qquad l_{ij} = |\varphi(r)|_{c_{\{i,j\}}} \in \mathbb{N} .$$

If the morphism φ is a coding, then for every $i, j \in \{1,2,3,4,5\}$, $i \neq j$:

$$|\varphi(x_i)|_{a_{\{i,j\}}} \cdot |\varphi(x_j)|_{b_{\{i,j\}}} \neq |\varphi(x_j)|_{a_{\{i,j\}}} \cdot |\varphi(x_i)|_{b_{\{i,j\}}} . \tag{3.10}$$

So we can assume that there exist $\alpha_{ij} \in \mathbb{N}$ and $\beta_{ij}, \gamma_{ij} \in \mathbb{Z}$ satisfying

$$\begin{aligned}
\alpha_{ij} \cdot |\varphi(z_{\{i,j\}})|_{a_{\{i,j\}}} + \beta_{ij} \cdot |\varphi(x_i)|_{a_{\{i,j\}}} + \gamma_{ij} \cdot |\varphi(x_j)|_{a_{\{i,j\}}} &= 0 , \\
\alpha_{ij} \cdot |\varphi(z_{\{i,j\}})|_{b_{\{i,j\}}} + \beta_{ij} \cdot |\varphi(x_i)|_{b_{\{i,j\}}} + \gamma_{ij} \cdot |\varphi(x_j)|_{b_{\{i,j\}}} &= 0 .
\end{aligned}$$

Now consider the ratios

$$n_A = \frac{|\varphi(z_A)|_d}{|\varphi(z_A)|_e} \in \mathbb{Q}_0^+ \cup \{\infty\}$$

(if both numbers in one of these fractions are zero, the condition (T) is violated). Whenever two of the ratios corresponding to disjoint sets are equal, d and e behave like one letter there and (T) can be also considered unsatisfied. Otherwise careful examination shows that, up to symmetries of the graph (Σ_0, D), one of the situations

$$n_{\{1,2\}} < n_{\{4,5\}} < n_{\{1,3\}} \leq n_{\{2,3\}} \quad \text{and} \quad n_{\{1,3\}} \leq n_{\{2,3\}} < n_{\{4,5\}} < n_{\{1,2\}}$$

arises. In other words, there exist $\delta, \varepsilon, \zeta \in \mathbb{N}$ such that

$$\begin{aligned}
\delta \cdot |\varphi(z_{\{1,2\}})|_d + \varepsilon \cdot |\varphi(z_{\{1,3\}})|_d &= \zeta \cdot |\varphi(z_{\{4,5\}})|_d , \\
\delta \cdot |\varphi(z_{\{1,2\}})|_e + \varepsilon \cdot |\varphi(z_{\{1,3\}})|_e &= \zeta \cdot |\varphi(z_{\{4,5\}})|_e ;
\end{aligned}$$

and similarly for the set $\{2,3\}$ instead of $\{1,3\}$.

In the case

$$|\varphi(z_{\{1,2\}})|_{a_{\{1,2\}}} = |\varphi(z_{\{1,2\}})|_{b_{\{1,2\}}} = 0\ , \tag{3.11}$$

we can additionally assume $\beta_{13} \neq 0$, for if $\beta_{13} = 0$ then φ is not injective on the tree

$$\{x_3, x_4, x_5, y_{\{1,2\}}, y_{\{1,3\}}, y_{\{4,5\}}, z_{\{1,2\}}, z_{\{1,3\}}, z_{\{4,5\}}, r\}\ .$$

Now define a word

$$\begin{aligned} u = {} & x_1^{\alpha_{45}\operatorname{pos}(-\beta_{13})\varepsilon} \cdot y_{\{1,2\}}^{k_{13}k_{45}l_{12}\operatorname{pos}(-\operatorname{sgn}(\beta_{13}))} \cdot x_1^{\alpha_{45}\operatorname{pos}(-\beta_{13})\varepsilon} \cdot x_3^{2\alpha_{45}\operatorname{pos}(-\gamma_{13})\varepsilon} \cdot \\ & y_{\{1,3\}}^{k_{12}k_{45}l_{13}} \cdot x_1^{\alpha_{45}\operatorname{pos}(\beta_{13})\varepsilon} \cdot y_{\{1,2\}}^{k_{13}k_{45}l_{12}\operatorname{pos}(\operatorname{sgn}(\beta_{13}))} \cdot x_1^{\alpha_{45}\operatorname{pos}(\beta_{13})\varepsilon} \cdot \\ & x_3^{2\alpha_{45}\operatorname{pos}(\gamma_{13})\varepsilon} \cdot z_{\{1,2\}}^{2\alpha_{13}\alpha_{45}\delta} \cdot z_{\{1,3\}}^{2\alpha_{13}\alpha_{45}\varepsilon} \cdot r^{k_{12}k_{13}k_{45}} \cdot z_{\{4,5\}}^{2\alpha_{13}\alpha_{45}\zeta} \cdot x_4^{2\alpha_{13}\operatorname{pos}(\beta_{45})\zeta} \cdot \\ & x_5^{2\alpha_{13}\operatorname{pos}(\gamma_{45})\zeta} \cdot y_{\{4,5\}}^{k_{12}k_{13}l_{45}} \cdot x_4^{2\alpha_{13}\operatorname{pos}(-\beta_{45})\zeta} \cdot x_5^{2\alpha_{13}\operatorname{pos}(-\gamma_{45})\zeta}\ , \end{aligned}$$

where $\operatorname{pos}(m)$ denotes m if $m > 0$ and 0 otherwise. Then we have $u \nsim_I \overleftarrow{u}$ since $\pi_{x_1,y_{\{1,3\}}}(u) \neq \pi_{x_1,y_{\{1,3\}}}(\overleftarrow{u})$, but at the same time $\varphi(u) \sim_{I'} \varphi(\overleftarrow{u})$.

If (3.11) does not hold, due to (3.10) the numbers

$$m_1 = \frac{|\varphi(x_1)|_{a_{\{1,2\}}}}{|\varphi(x_1)|_{b_{\{1,2\}}}} \qquad m_2 = \frac{|\varphi(x_2)|_{a_{\{1,2\}}}}{|\varphi(x_2)|_{b_{\{1,2\}}}} \qquad m_3 = \frac{|\varphi(z_{\{1,2\}})|_{a_{\{1,2\}}}}{|\varphi(z_{\{1,2\}})|_{b_{\{1,2\}}}}$$

in $\mathbb{Q}_0^+ \cup \{\infty\}$ are correctly defined and by possibly interchanging the indices 1 and 2 we can achieve that m_2 is either strictly larger or strictly smaller than both m_1 and m_3. Under this assumption, let us denote

$$\begin{aligned} p_{ij} &= |\varphi(x_i)|_{a_{\{i,j\}}} & q_{ij} &= |\varphi(x_i)|_{b_{\{i,j\}}} \\ s_{ij} &= |\varphi(z_{\{i,j\}})|_{a_{\{i,j\}}} & t_{ij} &= |\varphi(z_{\{i,j\}})|_{b_{\{i,j\}}} \end{aligned}$$

and consider the numbers

$$\begin{aligned} \eta &= (p_{31}q_{13} - p_{13}q_{31})(p_{21}t_{12} - q_{21}s_{12}) \\ \vartheta &= |(p_{31}q_{13} - p_{13}q_{31})(p_{21}q_{12} - p_{12}q_{21})| \\ \iota &= \operatorname{sgn}(\eta)\alpha_{45}\big(\varepsilon(p_{31}t_{13} - q_{31}s_{13})(p_{21}q_{12} - p_{12}q_{21}) - \delta\eta\big) \\ \kappa &= \operatorname{sgn}(\eta)(p_{31}q_{13} - p_{13}q_{31})(q_{12}s_{12} - p_{12}t_{12}) \\ \lambda &= \operatorname{sgn}(\eta)(q_{13}s_{13} - p_{13}t_{13})(p_{21}q_{12} - p_{12}q_{21})\ . \end{aligned}$$

Then the word

$$\begin{aligned}u = {} & x_1^{\alpha_{45}\delta|\eta|} \cdot x_2^{\alpha_{45}\delta \operatorname{pos}(\kappa)} \cdot x_3^{\alpha_{45}\varepsilon \operatorname{pos}(\lambda)} \cdot y_{\{1,2\}}^{k_{13}k_{45}l_{12}} \cdot x_1^{\operatorname{pos}(\iota)} \cdot \\ & y_{\{1,3\}}^{2k_{12}k_{45}l_{13}} \cdot x_1^{\operatorname{pos}(-\iota)} \cdot y_{\{1,2\}}^{k_{13}k_{45}l_{12}} \cdot x_2^{\alpha_{45}\delta \operatorname{pos}(-\kappa)} \cdot x_3^{\alpha_{45}\varepsilon \operatorname{pos}(-\lambda)} \cdot \\ & z_{\{1,2\}}^{\alpha_{45}\delta\vartheta} \cdot z_{\{1,3\}}^{\alpha_{45}\varepsilon\vartheta} \cdot r^{2k_{12}k_{13}k_{45}} \cdot z_{\{4,5\}}^{\alpha_{45}\zeta\vartheta} \cdot x_4^{\operatorname{pos}(\beta_{45})\zeta\vartheta} \cdot x_5^{\operatorname{pos}(\gamma_{45})\zeta\vartheta} \cdot \\ & y_{\{4,5\}}^{2k_{12}k_{13}l_{45}} \cdot x_4^{\operatorname{pos}(-\beta_{45})\zeta\vartheta} \cdot x_5^{\operatorname{pos}(-\gamma_{45})\zeta\vartheta}\end{aligned}$$

does not satisfy $u \sim_I \overleftarrow{u}$ because $\pi_{x_1,y_{\{1,2\}}}(u) \neq \pi_{x_1,y_{\{1,2\}}}(\overleftarrow{u})$. But $\varphi(u) \sim_{I'} \varphi(\overleftarrow{u})$. Altogether, φ is not a coding.

Proposition 3.3.4 states that a coding from the monoid $\mathbb{M}(\Sigma,I)$ to a direct product of free monoids exists if and only if the condition (v) holds provided the dependence alphabet (Σ,D) is C_3,C_4-free. On the other hand, it is easy to see that when (Σ,D) is either C_3 or C_4, this condition is not necessary for a coding to exist; as for the case of the graph C_4, the submonoid of the monoid $\{a,b\}^* \times \{c,d\}^*$ generated by the set $\{ac, ac^2, bd, bd^2\}$ is isomorphic to $\mathbb{M}(\Sigma,I)$. Moreover, in the rest of this section we show that for domain dependence alphabets which contain subgraphs isomorphic to either C_3 or C_4, the existence of a wlt-mapping does not guarantee the existence of a weak coding. Let us start with the graph C_3.

Example 3.4.2. Let

$$\begin{aligned}&\Sigma = \{x,y,z,r,s,t\},\ I = \mathrm{id}_\Sigma \cup \{(x,y),(y,x)\} \text{ and}\\ &\mathbb{M}(\Sigma',I') = \{a_1,a_2\}^* \times \{b_1,b_2\}^* \times \{c_1,c_2\}^* .\end{aligned}$$

First, we demonstrate that wlt-mappings from (Σ,I) to (Σ',I') are (up to symmetry) exactly the mappings $f : \Sigma \to 2^{\Sigma'}$ satisfying

$$\begin{aligned}&f(x) \in \big\{\{a_1,b_1\},\{a_1,b_1,c_1\}\big\}, && f(z) = \{a_2,b_2,c_2\}, && f(s) = \{a_2,b_1,c_2\},\\ &f(y) \in \big\{\{a_1,c_1\},\{a_1,b_1,c_1\}\big\}, && f(r) = \{a_1,b_2,c_2\}, && f(t) = \{a_2,b_2,c_1\}.\end{aligned}$$

Without loss of generality assume $f(x) \cup f(y) \subseteq \{a_1,b_1,c_1\}$. Due to the condition of Lemma 3.1.11 for $X = \{x,y,z\}$, the set $f(z)$ contains (up to symmetry) both letters a_2 and b_2. Suppose $f(z) = \{a_2,b_2\}$. Then one of the letters a_1 and b_1 belongs to at least two of the sets $f(r)$, $f(s)$ and $f(t)$, say $a_1 \in f(r) \cap f(s)$. By Lemma 3.1.11 for $\{x,y,r\}$ and $\{x,y,s\}$ we have $b_2,c_2 \in f(r) \cap f(s)$, which means that $f(r) = f(s)$, contradicting $r\,D\,s$. Therefore either $f(z) = \{a_2,b_2,c_1\}$ or $f(z) = \{a_2,b_2,c_2\}$. Using the same arguments as for z also for r, s and t we deduce that f is of the required form.

Consider any weak morphism $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ such that $\mathrm{alph} \circ \varphi|_\Sigma = f$. We are going to prove that it is not injective. Thanks to symmetries in a_1, b_1 and c_1, we can assume

$$\frac{|\varphi(x)|_{b_1}}{|\varphi(y)|_{b_1}} \geq \frac{|\varphi(x)|_{a_1}}{|\varphi(y)|_{a_1}} \geq \frac{|\varphi(x)|_{c_1}}{|\varphi(y)|_{c_1}} .$$

Then, applying Lemma 2.1.17 for $A = \{a_1, a_2\}$, we modify φ to satisfy

$$|\varphi(x)|_{a_1} = |\varphi(y)|_{a_1} = |\varphi(s)|_{a_2} = |\varphi(t)|_{a_2} = 1$$

together with $|\varphi(x)|_{b_1} \geq |\varphi(y)|_{b_1}$ and $|\varphi(x)|_{c_1} \leq |\varphi(y)|_{c_1}$. If one of these inequalities holds as an equality, we obtain $\varphi(xty) \sim_{I'} \varphi(ytx)$ and $\varphi(xsy) \sim_{I'} \varphi(ysx)$ respectively. If it is not the case, we have

$$\varphi(x) = a_1 b_1^i c_1^j \qquad \varphi(y) = a_1 b_1^k c_1^l \qquad \varphi(s) = a_2 b_1^m c_2^n \qquad \varphi(t) = a_2 b_2^o c_1^p$$

for some $j, k \in \mathbb{N}_0$, $i, l, m, n, o, p \in \mathbb{N}$ and the word $u = x^{mp} t^{m(l-j)} s^{p(i-k)} y^{mp}$ satisfies $\varphi(u) \sim_{I'} \varphi(\overleftarrow{u})$.

Remark 3.4.3. Another interesting case of the ($\mathscr{W}$-)TCP which still remains open is obtained by allowing for domain monoids only free products of free commutative monoids. In [9] the condition analogous to the existence of a wlt-mapping was proved accurate for general codings provided all of these free commutative monoids have at most two generators. But Example 3.4.2 demonstrates that once we try to generalize this result to weak morphisms, such a condition becomes insufficient.

Finally, we consider domain dependence alphabets containing C_4.

Example 3.4.4. On the alphabet $\Sigma = \{x, y, z, p, q, r, s, t\}$ let the dependence relation D be defined by the diagram

$$\begin{tikzcd}
x \arrow[r, dash] \arrow[d, dash] & y \arrow[r, dash] \arrow[d, dash] & z \arrow[r, dash] \arrow[d, dash] & p \arrow[d, dash] \\
t \arrow[r, dash] & s \arrow[r, dash] & r \arrow[r, dash] & q
\end{tikzcd}$$

and let

$$\mathbb{M}(\Sigma', I') = \{a_1, a_2\}^* \times \{b_1, b_2\}^* \times \{c_1, c_2\}^* \times \{d_1, d_2\}^* \times \{e_1, e_2\}^* .$$

By applying the condition of Lemma 3.1.11 to maximal subtrees of the graph (Σ, D), one can show that there exists (up to symmetry) just one wlt-mapping f from (Σ, I)

to (Σ', I'), namely

$$\begin{array}{lll} f(x) = \{a_1, b_1\}\ , & f(y) = \{a_2, b_2, c_2\}\ , & f(z) = \{c_1, d_1, e_1\}\ , \\ f(p) = \{d_2, e_2\}\ , & f(q) = \{d_1, e_1\}\ , & f(r) = \{c_2, d_2, e_2\}\ , \\ f(s) = \{a_1, b_1, c_1\}\ , & f(t) = \{a_2, b_2\}\ . & \end{array}$$

But there is no weak coding $\varphi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ such that $\mathrm{alph} \circ \varphi|_\Sigma = f$ since we may assume

$$|\varphi(y)|_{c_2} = |\varphi(z)|_{c_1} = |\varphi(r)|_{c_2} = |\varphi(s)|_{c_1} = 1$$

due to Lemma 2.1.17 and then we obtain $\varphi(ysrz) \sim_{I'} \varphi(rzys)$.

Chapter 4

The General Case

This chapter is devoted to proving the undecidability of the TCP. The proof proceeds in two steps. In Section 4.1 we consider the problem of existence of weak codings with partially prescribed contents of images of letters and show that many instances of this problem can be effectively encoded into instances of the trace coding problem. Then, in Section 4.2, we construct a reduction of the PCP to this problem. The proof of the correctness of the reduction is given in Section 4.4 after presenting its main ideas on a particular instance of the PCP in Section 4.3.

4.1 Content Fixation

The aim of this section 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 to specify these restrictions on contents — one of them to express which letters are compulsory and the other to express which are allowed.

Definition 4.1.1. Let $\mu,\nu : \Sigma \to 2^{\Sigma'}$ be any mappings. We say that a weak morphism $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ is (μ,ν)-*weak* if it satisfies for all $x \in \Sigma$ the condition

$$\mu(x) \subseteq \mathrm{alph}(\varphi(x)) \subseteq \nu(x) \ .$$

We call it ν-*weak* whenever $\mathrm{alph}(\varphi(x)) \subseteq \nu(x)$ for all $x \in \Sigma$.

First, we are going to 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 order to make it possible to extend any (μ,ν)-weak coding to the monoids with the additional generators Θ without violating its injectivity, we add into the codomain alphabet another set of new letters Γ, which enables us to encode relative positions of letters from Θ in a word from $(\Sigma \cup \Theta)^*$ with respect to the original letters Σ.

In addition, we have to ensure that the resulting alphabets satisfy all assumptions of Proposition 4.1.3 which provides the second step of the reduction — this is the first claim of the following proposition.

Proposition 4.1.2. *Let (Σ,I) and (Σ',I') be independence alphabets such that $D \neq \emptyset$ and I is transitive, i.e. 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 $x,y \in \Sigma$:*

$$x\,I\,y,\ x \neq y \implies \mu(x) = \mu(y) = \emptyset\ , \tag{4.1}$$
$$\nu(x) \times \nu(x) \subseteq I'\ , \tag{4.2}$$
$$x\,I\,y \implies \nu(x) = \nu(y)\ . \tag{4.3}$$

Define new independence alphabets (Σ_1,I_1) and (Σ_1',I_1') as follows. Let

$$\Gamma = \bigcup\big\{\{y\} \times \mu(y) \times (\Sigma \setminus \{y\}) \mid y \in \Sigma\big\},$$
$$\Theta = \bigcup\big\{\{x\} \times \mu(x) \mid x \in \Sigma\big\},\quad \Sigma_1 = \Sigma \cup \Theta,\quad \Sigma_1' = \Sigma' \cup \Theta \cup \Gamma$$

and let the independence relations be given by the conditions:

$$I_1 \cap (\Sigma \times \Sigma) = I \qquad\qquad I_1 \cap (\Theta \times \Sigma_1) = \mathrm{id}_\Theta$$
$$I_1' \cap (\Sigma' \times \Sigma') = I' \qquad\qquad \Gamma \times (\Sigma' \cup \Gamma) \subseteq I_1'$$

and for all $x,y,z \in \Sigma$, $y \neq z$, $a \in \mu(x)$, $b \in \mu(y)$, $c \in \Sigma'$:

$$(x,a)\,I_1'\,c \iff a \neq c\ , \tag{4.4}$$
$$(x,a)\,I_1'\,(y,b,z) \iff (x,a) \neq (y,b)\ , \tag{4.5}$$
$$(x,a)\,I_1'\,(y,b) \iff (x,a) = (y,b)\ . \tag{4.6}$$

Further, define a mapping $\nu_1 : \Sigma_1 \to 2^{\Sigma_1'}$ by the rules:

$$\nu_1(x) = \nu(x) \cup \bigcup\big\{\{y\} \times \mu(y) \times \{z\} \mid y,z \in \Sigma,\ y \neq z,\ z\,I\,x\big\}\ ,$$
$$\nu_1((x,a)) = \{(x,a)\}\ ,$$

for $x \in \Sigma$ and $a \in \mu(x)$. Then the following assertions hold.

(i) *The new independence relation I_1 is transitive and the mapping ν_1 satisfies the corresponding modifications of* (4.2) *and* (4.3).

(ii) *There exists a (μ,ν)-weak coding from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$ if and only if there exists a ν_1-weak coding from $\mathbb{M}(\Sigma_1,I_1)$ to $\mathbb{M}(\Sigma_1',I_1')$.*

Proof. The first claim is evident. Let us prove the second one.

"$\Longrightarrow$" Let $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ be any (μ,ν)-weak coding. For all letters $x \in \Sigma$ and $a \in \mu(x)$, set

$$\psi(x) = \varphi(x) \cdot \prod_{\substack{y \in \Sigma \setminus \{x\} \\ b \in \mu(y)}} (y,b,x) \,, \qquad\qquad \psi((x,a)) = (x,a) \,.$$

It is clear that this defines a ν_1-weak morphism $\psi : \mathbb{M}(\Sigma_1,I_1) \to \mathbb{M}(\Sigma_1',I_1')$. In order to prove that ψ is injective, take two words $u,v \in (\Sigma_1)^*$ which satisfy $\psi(u) \sim_{I_1'} \psi(v)$. Then $\pi_\Sigma(u) \sim_{I_1} \pi_\Sigma(v)$ since

$$\varphi(\pi_\Sigma(u)) = \pi_{\Sigma'}(\psi(u)) \sim_{I'} \pi_{\Sigma'}(\psi(v)) = \varphi(\pi_\Sigma(v)) \,.$$

One also derives

$$\pi_\Theta(u) = \pi_\Theta(\psi(u)) = \pi_\Theta(\psi(v)) = \pi_\Theta(v) \,.$$

For $x,y \in \Sigma$, $x \neq y$, $a \in \mu(x)$, the word $\pi_{(x,a),y}(u)$ can be obtained from the word $\pi_{(x,a),(x,a,y)}(\psi(u))$ by substituting y for (x,a,y). As the same holds also for v and because $(x,a)\ D_1'\ (x,a,y)$, we have $\pi_{(x,a),y}(u) = \pi_{(x,a),y}(v)$.

To get $u \sim_{I_1} v$, it remains to show $\pi_{(x,a),x}(u) = \pi_{(x,a),x}(v)$ for all $x \in \Sigma$ and $a \in \mu(x)$. Let us employ Lemma 2.1.9 for the word morphism $\pi_{(x,a),a} \circ \psi : (\Sigma_1)^* \to \{(x,a),a\}^*$, the set $\Sigma_1 \setminus \{(x,a),x\}$, the letter a and the words u and v. We obtain

$$\pi_{(x,a),a}\Big(\psi\Big(\pi_{(x,a),x}(u)\Big)\Big) = \pi_{(x,a),a}\Big(\psi\Big(\pi_{(x,a),x}(v)\Big)\Big)$$

and $\pi_{(x,a),x}(u) = \pi_{(x,a),x}(v)$ now follows from $a \in \mathrm{alph}(\psi(x))$. Hence ψ is a coding.

"$\Longleftarrow$" Let $\psi : \mathbb{M}(\Sigma_1,I_1) \to \mathbb{M}(\Sigma_1',I_1')$ be a ν_1-weak coding. We are going to prove that the coding $\sigma = \psi|_{\mathbb{M}(\Sigma,I)}$ is (μ,ν_1)-weak. Consider arbitrary $x \in \Sigma$ and $a \in \mu(x)$. Then $\mathrm{alph}\big(\psi((x,a))\big) = \{(x,a)\}$ since ψ is a ν_1-weak coding. Because $x\, D_1\ (x,a)$ and ψ is a coding, it implies that there is $\alpha \in \mathrm{alph}(\psi(x))$ such that $\alpha\ D_1'\ (x,a)$. Since

$$\mathrm{alph}(\psi(x)) \subseteq \nu_1(x) \subseteq \Sigma' \cup \bigcup \big\{ \{y\} \times \mu(y) \times \{x\} \mid y \in \Sigma \setminus \{x\} \big\}$$

due to the assumption (4.1), using the conditions (4.4) and (4.5) we conclude $\alpha = a$. Hence a belongs to $\mathrm{alph}(\sigma(x))$. This means that σ is a (μ,ν_1)-weak coding from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma'\cup\Gamma,I_1')$. Therefore the morphism $\pi_{\Sigma'}\circ\sigma : \mathbb{M}(\Sigma,I)\to\mathbb{M}(\Sigma',I')$ is a desired (μ,ν)-weak coding by Lemma 2.1.4 since the fact $D\neq\emptyset$ and the transitivity of the relation I together imply $\Sigma = D(\Sigma)$. □

In the rest of this section we show that we can manage our content requirements even without the mapping ν. This time, we add to the original independence alphabets mutually dependent cliques of independent letters, each of them having sufficiently distinct size. Then we can employ Lemma 1.2.11 to 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.

In effect the construction functions in the same way even if we add new letters to the codomain alphabet according to Definition 2.2.1 in order to pass to the TCP. It is due to the fact that these new letters do not form in the codomain independence alphabet any cliques bigger than those already existing.

Proposition 4.1.3. *Let (Σ,I) and (Σ',I') be independence alphabets with I transitive and $\nu:\Sigma\to 2^{\Sigma'}$ a mapping satisfying (4.2) and (4.3). Let $\star\notin\Sigma$ be a new letter and $\tau:\Sigma\cup\{\star\}\to\{1,\dots,n\}$ a bijection such that $\tau(\star)=1$. Define new independence alphabets (Σ_1,I_1) and (Σ_1',I_1') as follows. Set*

$$\Theta = \big\{(x,i)\mid x\in\Sigma\cup\{\star\},\ i\in\mathbb{N},\ 1\le i\le \tau(x)\cdot(|\Sigma'|+2)\big\}$$

and let $\Sigma_1=\Sigma\cup\Theta$ and $\Sigma_1'=\Sigma'\cup\Theta$. Let the independence relations be given by the conditions:

$$I_1\cap(\Sigma\times\Sigma)=I, \qquad I_1'\cap(\Sigma'\times\Sigma')=I',$$

for $(x,i),(y,j)\in\Theta$, $x,y\in\Sigma\cup\{\star\}$:

$$(x,i)\ I_1\ (y,j) \iff (x,i)\ I_1'\ (y,j) \iff x=y$$

and for $(x,i),(\star,j)\in\Theta$, $x,y\in\Sigma$, $a\in\Sigma'$:

$$(x,i)\ I_1\ y \iff x\ I\ y \qquad (\star,j)\ I_1\ y$$
$$(x,i)\ I_1'\ a \iff a\in\nu(x) \qquad (\star,j)\ I_1'\ a\,.$$

Then the following statements are equivalent.

(i) *There exists a 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 $\mathbb{M}(\Sigma_1',I_1')$.*

Proof. (i) $\Longrightarrow$ (ii). Let $\varphi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ be a ν-weak coding. Then we can extend φ into a weak morphism $\psi : \mathbb{M}(\Sigma_1,I_1) \to \mathbb{M}(\Sigma_1',I_1')$ by taking the identity on Θ. Indeed, when verifying that this defines a weak morphism, everything is clear except for the case of letters $(x,i)\ I_1\ y$ with $x,y \in \Sigma$; for such letters we have $x\ I\ y$ and $\mathrm{alph}(\psi(y)) = \mathrm{alph}(\varphi(y)) \subseteq \nu(y) = \nu(x)$ because φ is ν-weak and ν satisfies (4.3), and therefore $\mathrm{alph}\big(\psi((x,i))\big) \times \mathrm{alph}(\psi(y)) \subseteq I_1'$.

Let us prove that ψ is a coding. Suppose $u,v \in (\Sigma_1)^*$, $\psi(u) \sim_{I_1'} \psi(v)$. It is easy to see that $\pi_\Sigma(u) \sim_{I_1} \pi_\Sigma(v)$ and $\pi_\Theta(u) \sim_{I_1} \pi_\Theta(v)$ since both φ and the identity are injective. Take $(x,i) \in \Theta$ with $x \in \Sigma$. Consider the weak morphisms

$$\rho : \mathbb{M}(\Sigma \cup \{(x,i)\}, I_1) \to \mathbb{M}(\Sigma, I)$$

identical on Σ and mapping (x,i) to x and

$$\rho' : \mathbb{M}(\Sigma' \cup \{(x,i)\}, I_1') \to \mathbb{M}(\Sigma', I')$$

identical on Σ' and mapping (x,i) to $\varphi(x) = \psi(x)$. Notice that ρ' is really a weak morphism because for $a\ I_1'\ (x,i)$, where $a \in \Sigma'$, we have $a \in \nu(x)$, using (4.2) we get $\{a\} \times \nu(x) \subseteq I'$ and consequently $\{a\} \times \mathrm{alph}(\psi(x)) \subseteq I'$ as φ is ν-weak.

Then the following diagram commutes.

$$\begin{array}{ccc}
\mathbb{M}(\Sigma \cup \{(x,i)\}, I_1) & \xrightarrow{\ \rho\ } & \mathbb{M}(\Sigma, I) \\
\psi \big\downarrow & & \big\downarrow \varphi \\
\mathbb{M}(\Sigma' \cup \{(x,i)\}, I_1') & \xrightarrow[\ \rho'\ ]{} & \mathbb{M}(\Sigma', I')
\end{array} \tag{4.7}$$

Let us consider the words $\overline{u} = \pi_{\Sigma \cup \{(x,i)\}}(u)$ and $\overline{v} = \pi_{\Sigma \cup \{(x,i)\}}(v)$. Since

$$\psi(\overline{u}) = \pi_{\Sigma' \cup \{(x,i)\}}(\psi(u)) \sim_{I_1'} \pi_{\Sigma' \cup \{(x,i)\}}(\psi(v)) = \psi(\overline{v})\ ,$$

from (4.7) it follows that $(\varphi \circ \rho)(\overline{u}) \sim_{I'} (\varphi \circ \rho)(\overline{v})$ and hence $\rho(\overline{u}) \sim_I \rho(\overline{v})$ due to the injectivity of the morphism φ. If a letter $y \in \Sigma$ satisfies $y\ D_1\ (x,i)$, then $y\ D\ x$ and thus $\pi_{x,y}(\overline{u}) = \pi_{x,y}(\overline{v})$. Therefore Lemma 2.1.9 can be applied to the morphism

$\pi_{x,y} \circ \rho : (\Sigma \cup \{(x,i)\})^* \to \{x,y\}^*$ for the set $\{x\}$, the letter x and the words $\overline{u}$ and $\overline{v}$. We obtain

$$\pi_{x,y}\left(\rho\left(\pi_{(\Sigma\cup\{(x,i)\})\setminus\{x\}}(\overline{u})\right)\right) = \pi_{x,y}\left(\rho\left(\pi_{(\Sigma\cup\{(x,i)\})\setminus\{x\}}(\overline{v})\right)\right) ,$$

which is clearly equivalent to $\pi_{y,(x,i)}(u) = \pi_{y,(x,i)}(v)$. Altogether, we have verified the fact $u \sim_{I_1} v$. We conclude that ψ is a coding.

(ii) $\Longrightarrow$ (iii) is trivial.

(iii) $\Longrightarrow$ (i). Let $\varphi : \mathbb{M}(\Sigma_1, I_1) \to \mathbb{M}((\Sigma_1')_{\Sigma_1}, (I_1')_{\Sigma_1})$ be an arbitrary weak coding. First, we need to describe all large cliques in the codomain independence alphabet. So, consider a clique K in the graph $((\Sigma_1')_{\Sigma_1}, (I_1')_{\Sigma_1})$ having at least $|\Sigma'|+2$ elements. If we choose for every $\alpha \in K$ one element of $\mathrm{ealph}(\alpha)$, we get a clique K' in (Σ_1', I_1') of the same size as K by (2.9). Therefore K' contains at least two elements of Θ. Moreover, each of these elements is of the form (x,i) for a unique $x \in \Sigma \cup \{\star\}$, hence the other elements of K' lie in $\nu(x)$ (for $x = \star$, consider $\nu(\star) = \Sigma'$). Since there are always at least two elements of Θ in K', we can also see that for any choice of the clique K' this x is the same. Thus, if we write, for $x \in \Sigma$,

$$L_x = \{(x,i) \mid i \in \mathbb{N},\ 1 \le i \le \tau(x)\cdot(|\Sigma'|+2)\} \cup \nu(x) ,$$
$$L_\star = \{(\star,i) \mid i \in \mathbb{N},\ 1 \le i \le |\Sigma'|+2\} \cup \Sigma'_{\Sigma_1} ,$$

using (4.2) we obtain $K \subseteq L_x$ and $K \cap \Theta \neq \emptyset$ for some $x \in \Sigma \cup \{\star\}$.

Now we are going to prove that the set

$$K_x = \bigcup\left\{\mathrm{alph}\big(\varphi((x,i))\big) \mid i \in \mathbb{N},\ 1 \le i \le \tau(x)\cdot(|\Sigma'|+2)\right\}$$

satisfies the conditions $K_x \subseteq L_x$ and $K_x \cap \Theta \neq \emptyset$ for every $x \in \Sigma \cup \{\star\}$. We proceed by induction starting from the largest cliques. Assume we are already done with all letters $x \in \{\tau^{-1}(n), \ldots, \tau^{-1}(k+1)\}$ and let us consider $x = \tau^{-1}(k)$, where $k \in \{1, \ldots, n\}$. By Lemma 1.2.11, the set K_x is a clique in the graph $((\Sigma_1')_{\Sigma_1}, (I_1')_{\Sigma_1})$ of cardinality at least $\tau(x)\cdot(|\Sigma'|+2)$. Using the results of the previous paragraph, we can see that $K_x \subseteq L_y$ and $K_x \cap \Theta \neq \emptyset$, for some letter $y \in \Sigma \cup \{\star\}$. Due to the cardinality of K_x, we have $y \in \{\tau^{-1}(n), \ldots, \tau^{-1}(k+1), x\}$. If $y \neq x$ then $K_x \cup K_y \subseteq L_y$, which is a clique in $((\Sigma_1')_{\Sigma_1}, (I_1')_{\Sigma_1})$ since $y \neq \star$. But this contradicts (2.3) for any $(x,i), (y,j) \in \Theta$ because $(x,i)\ D_1\ (y,j)$ holds. Therefore $y = x$ and the fact is proved also for x.

For every letter $x \in \Sigma \cup \{\star\}$, as $K_x \cap \Theta \neq \emptyset$, there exist $(x,i_x), (x,j_x) \in \Theta$ such that $(x,i_x) \in \mathrm{alph}\big(\varphi((x,j_x))\big)$. Now consider any $y \in \Sigma$. Then $y\ I_1\ (\star, j_\star)$ and $y\ I_1\ (y, j_y)$, which implies

$$\mathrm{alph}(\varphi(y)) \times \{(\star, i_\star), (y, i_y)\} \subseteq (I_1')_{\Sigma_1} ,$$

consequently $\mathrm{ealph}(\varphi(y)) \subseteq \nu(y)$ and so $\mathrm{alph}(\varphi(y)) \subseteq \nu(y)$ by the assumption (4.2). Hence, the restriction $\varphi|_{\mathbb{M}(\Sigma,I)} : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ is a required ν-weak coding.

(iii) $\Longleftrightarrow$ (iv) follows immediately from Proposition 2.2.5. $\square$

4.2 Encoding of the Post's Correspondence Problem

It is well-known that the coPCP is not recursively enumerable. In this section we construct a reduction of this problem to the problem of deciding the existence of (μ,ν)-weak codings. Because our construction should be based only on contents of images of letters, we have to impose a certain restriction on instances of the PCP which enables us not to care about the numbers of occurrences of letters in these images.

Let $\mathscr{P}$ denote the following instance of the PCP. Let $n \in \mathbb{N}$. We are given n pairs $(w_1,\overline{w}_1),\dots,(w_n,\overline{w}_n)$ of non-empty words over some finite alphabet Ξ such that each two consecutive letters in any product of the words $w_1,\overline{w}_1,\dots,w_n,\overline{w}_n$ are different. The problem asks to decide whether there exists a finite sequence $i_1,\dots,i_m$ of integers from the set $\{1,\dots,n\}$ satisfying $w_1 w_{i_1} w_{i_2} \cdots w_{i_m} = \overline{w}_1 \overline{w}_{i_1} \overline{w}_{i_2} \cdots \overline{w}_{i_m}$ (observe that we require the initial pair of a solution to be equal to the first pair on the list).

Notice that this restriction on instances of the PCP causes no loss of generality since for every instance of the PCP we can obtain an equivalent instance of the above form by introducing a new letter # into Ξ and performing the substitution $x \mapsto x\#$ for all $x \in \Xi$.

For $i \in \{1,\dots,n\}$ and $j \in \{1,\dots,|w_i|\}$ and $k \in \{1,\dots,|\overline{w}_i|\}$, we refer to the j-th letter of the word w_i as x_{ij} and to the k-th letter of $\overline{w}_i$ as $\overline{x}_{ik}$. For the rest of this section, when writing indices i, ij or ik, we implicitly assume that they run through all values as in the previous sentence.

Now we define two independence alphabets (Σ,I) and (Σ',I'). Let us first introduce a set of new letters:

$$\begin{aligned}\Omega = \{&\alpha,\alpha_0,\beta_1,\dots,\beta_8,\gamma_1,\gamma_2,\gamma_3,\delta_{i1},\delta_{i2},\varepsilon_{ij1},\varepsilon_{ij2},\overline{\varepsilon}_{ik1},\overline{\varepsilon}_{ik2},\\ &\zeta_{ij1},\zeta_{ij2},\overline{\zeta}_{ik1},\overline{\zeta}_{ik2},\eta_{ij1},\eta_{ij2},\eta_{ij3},\overline{\eta}_{ik1},\overline{\eta}_{ik2},\overline{\eta}_{ik3},\vartheta_{ij},\overline{\vartheta}_{ik},\\ &\iota_i,\kappa_1,\kappa_2,\lambda_{ij1},\lambda_{ij2},\lambda_{ij3},\overline{\lambda}_{ik1},\overline{\lambda}_{ik2},\overline{\lambda}_{ik3},\xi_{i1},\xi_{i2},\xi_{i3}\}\ .\end{aligned}$$

The domain alphabet $\Sigma = (\Omega \setminus \{\alpha_0\}) \times \{1,2\}$ consists of one pair of letters for each element of the set $\Omega \setminus \{\alpha_0\}$. Letters from these pairs should appear on opposite sides of a counter-example to the injectivity of our morphism and correspond there to each other according to their first coordinates. Let the independence relation be

$$I = \mathrm{id}_\Sigma \cup \big\{\big((\alpha,1),(\alpha,2)\big),\big((\alpha,2),(\alpha,1)\big)\big\}\ .$$

In the desired outcome of our construction, counter-examples to the injectivity should correspond to solutions of the instance $\mathscr{P}$. A computation of a solution of $\mathscr{P}$ will be simulated by appending the pairs of elements of Σ to an already constructed semi-equality by means of Lemma 2.1.24 in the way determined by its state. Just one pair of letters in Σ is set independent to ensure that there is only one way to start this computation due to Lemma 2.1.1.

The alphabet Σ' is divided into several disjoint subsets according to the role of letters in the encoding:

$$\Sigma' = \{l_1, l_2, r_1, r_2, b, c, d\} \cup S \cup A \cup E \cup F \cup G \cup P \cup Q \cup \Xi \cup \{\star\} \cup \Omega \ .$$

Letters from the set $S = \{s, s_{\mathrm{f}}, s_i, s_{ij}, \bar{s}_{ik}, t_{ij}, \bar{t}_{ik}\}$ will control a computation during its initial and its final phase except for the letter s, which will keep it from a premature termination. The process of composing the words w_i and $\overline{w}_i$ in a semi-equality will be controlled by elements of the set $A = \{a, a_{ij}, \overline{a}_{ik}\}$, whose occurrences characterize individual steps of a computation: a semi-equality having the letter a_{ij} ($\overline{a}_{ik}$) in its state will be extended by pairs of elements of Σ introducing x_{ij} to the left component ($\overline{x}_{ik}$ to the right one respectively) and a semi-equality with a in the state will allow us to choose the next pair of words $(w_i, \overline{w}_i)$ or to finish a computation. During an addition of the pair $(w_i, \overline{w}_i)$, all letters of the word w_i will be placed before proceeding to place the letters of the word $\overline{w}_i$.

The progress of a computation will be determined by dependences between a's and letters from the sets

$$E = \{e, e_{ij}, \overline{e}_{ik}\}, \ F = \{f_i, f_{ij}, \overline{f}_{ik}\} \text{ and } G = \{g_i, g_{ij}, \overline{g}_{ik}\} \ .$$

More precisely, each of these letters can appear in the images of letters appended to a semi-equality only if it is independent on the letter from A in the state of this semi-equality. The letter e will serve for terminating a computation and the other elements of E for introducing letters from Ξ. In the same way, each letter from F will be used for manipulating the corresponding letter from G. Elements of G will guard against undesirable letters remaining in the state from one step of a computation to the next one. Letters from $E \setminus \{e\}$ and F are paired with auxiliary letters from the sets

$$P = \{p_{ij}, \overline{p}_{ik}\} \text{ and } Q = \{q_i, q_{ij}, \overline{q}_{ik}\} \ .$$

The letter $\star$ will behave just as the letters of the original alphabet Ξ and it will mark the end of a solution of $\mathscr{P}$. Letters from Σ will be placed on the appropriate sides of a semi-equality thanks to l's and r's and the pairs of letters in Σ will be fixed using elements of Ω.

Dependences between elements of Σ' are set by the following rules; all pairs not mentioned below are considered independent:

$$l_1 \, D' \, l_2, \ r_1 \, D' \, r_2, \ b \, D' \, c, \ b \, D' \, d \ ,$$
$$p_{ij} \, D' \, e_{ij}, \ \overline{p}_{ik} \, D' \, \overline{e}_{ik}, \ q_i \, D' \, f_i, \ q_{ij} \, D' \, f_{ij}, \ \overline{q}_{ik} \, D' \, \overline{f}_{ik} \ ,$$

$$I' \cap S^2 = \mathrm{id}_S, \quad I' \cap (\Xi \cup \{\star\})^2 = \mathrm{id}_{\Xi \cup \{\star\}} \ ,$$
$$I' \cap \Omega^2 = \mathrm{id}_\Omega \cup \mathrm{sym}(\{\alpha, \alpha_0\} \times \{\beta_1, \beta_2, \beta_3, \beta_4\}) \ ,$$

$$I'\cap(A\cup E\cup F\cup G)^2 = \mathrm{id}_{A\cup E\cup F\cup G}\cup\mathrm{sym}\{(e,a),(e_{ij},a_{ij}),(\overline{e}_{ik},\overline{a}_{ik}),$$
$$(f_i,a),(f_{ij},a_{ij}),(\overline{f}_{ik},\overline{a}_{ik}),(g_i,a),(g_i,\overline{a}_{i|\overline{w}_i|}),(g_{ij},a_{ij}),(g_{ij},a_{ij-1}),$$
$$(\overline{g}_{ik},\overline{a}_{ik}),(\overline{g}_{ik},\overline{a}_{ik-1}),(g_i,f_i),(g_{ij},f_{ij}),(\overline{g}_{ik},\overline{f}_{ik})\}\ ,$$

where $a_{i0}=a$ and $\overline{a}_{i0}=a_{i|w_i|}$.

Now let us construct mappings $\mu,\nu:\Sigma\to 2^{\Sigma'}$. For each element $\omega\in\Omega\setminus\{\alpha_0\}$, the images of $(\omega,1)$ and $(\omega,2)$ under μ and ν are together understood as one rule for a computation. In fact, we cannot fix contents of images completely since we have to satisfy all assumptions of Proposition 4.1.2 which should be applied to every instance produced here in order to prolong the reduction to the TCP. As the desired contents are given by ν, let us define ν first.

The following rules guarantee that a computation starts correctly and that we can remove control letters at the end of a successful computation:

$$\begin{aligned}
\nu((\alpha,1)) &= \{l_1,r_1,b,s_1,a_{11},\alpha\} & \nu((\alpha,2)) &= \{l_1,r_1,b,s_1,a_{11},\alpha\}\\
\nu((\beta_1,1)) &= \{l_2,r_1,\alpha_0,\beta_1\} & \nu((\beta_1,2)) &= \{l_2,r_2,\beta_1\}\\
\nu((\beta_2,1)) &= \{l_2,\alpha_0,\beta_2\} & \nu((\beta_2,2)) &= \{l_2,r_2,\alpha,\beta_2\}\\
\nu((\beta_3,1)) &= \{l_1,r_2,\alpha_0,\beta_3\} & \nu((\beta_3,2)) &= \{l_2,r_2,\beta_3\}\\
\nu((\beta_4,1)) &= \{r_2,\alpha_0,\beta_4\} & \nu((\beta_4,2)) &= \{l_2,r_2,\alpha,\beta_4\}\\
\nu((\beta_5,1)) &= \{l_2,s_{\mathrm{f}},\beta_5\} & \nu((\beta_5,2)) &= \{r_2,s_1,\beta_5\}\\
\nu((\beta_6,1)) &= \{l_2,s_1,\beta_6\} & \nu((\beta_6,2)) &= \{r_2,s_{\mathrm{f}},a,\beta_6\}\\
\nu((\beta_7,1)) &= \{l_2,c,\beta_7\} & \nu((\beta_7,2)) &= \{r_2,b,\beta_7\}\\
\nu((\beta_8,1)) &= \{l_2,b,\beta_8\} & \nu((\beta_8,2)) &= \{r_2,d,\beta_8\}\\
\nu((\gamma_1,1)) &= \{l_2,s_{\mathrm{f}},\gamma_1\} & \nu((\gamma_1,2)) &= \{l_1,s_{\mathrm{f}},\gamma_1\}\\
\nu((\gamma_2,1)) &= \{r_1,s_{\mathrm{f}},\gamma_2\} & \nu((\gamma_2,2)) &= \{r_2,s_{\mathrm{f}},\gamma_2\}\\
\nu((\gamma_3,1)) &= \{a_{11},s_{\mathrm{f}},\gamma_3\} & \nu((\gamma_3,2)) &= \{a,s_{\mathrm{f}},\gamma_3\}\ .
\end{aligned}$$

The next family of rules serves for the initial placement of p's and q's:

$$\begin{aligned}
\nu((\delta_{i1},1)) &= \{l_2,c,s_i,\delta_{i1}\} & \nu((\delta_{i1},2)) &= \{r_2,d,s_{i+1},q_i,\delta_{i1}\}\\
\nu((\delta_{i2},1)) &= \{l_2,c,s_i,q_i,\delta_{i2}\} & \nu((\delta_{i2},2)) &= \{r_2,d,s_{i+1},\delta_{i2}\}\\
\nu((\varepsilon_{ij1},1)) &= \{l_2,s_{ij},\varepsilon_{ij1}\} & \nu((\varepsilon_{ij1},2)) &= \{r_2,s_{ij+1},p_{ij},\varepsilon_{ij1}\}\\
\nu((\varepsilon_{ij2},1)) &= \{l_2,s_{ij},p_{ij},\varepsilon_{ij2}\} & \nu((\varepsilon_{ij2},2)) &= \{r_2,s_{ij+1},\varepsilon_{ij2}\}\\
\nu((\overline{\varepsilon}_{ik1},1)) &= \{l_2,\overline{s}_{ik},\overline{\varepsilon}_{ik1}\} & \nu((\overline{\varepsilon}_{ik1},2)) &= \{r_2,\overline{s}_{ik+1},\overline{p}_{ik},\overline{\varepsilon}_{ik1}\}\\
\nu((\overline{\varepsilon}_{ik2},1)) &= \{l_2,\overline{s}_{ik},\overline{p}_{ik},\overline{\varepsilon}_{ik2}\} & \nu((\overline{\varepsilon}_{ik2},2)) &= \{r_2,\overline{s}_{ik+1},\overline{\varepsilon}_{ik2}\}\\
\nu((\zeta_{ij1},1)) &= \{l_2,t_{ij},\zeta_{ij1}\} & \nu((\zeta_{ij1},2)) &= \{r_2,t_{ij+1},q_{ij},\zeta_{ij1}\}\\
\nu((\zeta_{ij2},1)) &= \{l_2,t_{ij},q_{ij},\zeta_{ij2}\} & \nu((\zeta_{ij2},2)) &= \{r_2,t_{ij+1},\zeta_{ij2}\}
\end{aligned}$$

$$\begin{array}{ll}
\nu((\overline{\zeta}_{ik1},1)) = \{l_2,\bar{t}_{ik},\overline{\zeta}_{ik1}\} & \nu((\overline{\zeta}_{ik1},2)) = \{r_2,\bar{t}_{ik+1},\overline{q}_{ik},\overline{\zeta}_{ik1}\} \\
\nu((\overline{\zeta}_{ik2},1)) = \{l_2,\bar{t}_{ik},\overline{q}_{ik},\overline{\zeta}_{ik2}\} & \nu((\overline{\zeta}_{ik2},2)) = \{r_2,\bar{t}_{ik+1},\overline{\zeta}_{ik2}\} \ ,
\end{array}$$

where we use the notation

$$s_{n+1} = s_{11},\ s_{i|w_i|+1} = s_{i+11},\ s_{n+11} = \bar{s}_{11},\ \bar{s}_{i|\overline{w}_i|+1} = \bar{s}_{i+11},\ \bar{s}_{n+11} = t_{11},$$
$$t_{i|w_i|+1} = t_{i+11},\ t_{n+11} = \bar{t}_{11},\ \bar{t}_{i|\overline{w}_i|+1} = \bar{t}_{i+11},\ \bar{t}_{n+11} = s\ .$$

The main cycle inserting letters from Ξ is performed by the rules:

$$\begin{array}{ll}
\nu((\eta_{ij1},1)) = \{l_2,s,p_{ij},x_{ij},\eta_{ij1}\} & \nu((\eta_{ij1},2)) = \{r_2,s,e_{ij},\eta_{ij1}\} \\
\nu((\eta_{ij2},1)) = \{l_2,s,e_{ij},\eta_{ij2}\} & \nu((\eta_{ij2},2)) = \{r_2,s,p_{ij},\eta_{ij2}\} \\
\nu((\eta_{ij3},1)) = \{l_2,s,p_{ij},\eta_{ij3}\} & \nu((\eta_{ij3},2)) = \{r_2,s,e_{ij},\eta_{ij3}\} \\
\nu((\overline{\eta}_{ik1},1)) = \{l_2,s,\overline{p}_{ik},\overline{\eta}_{ik1}\} & \nu((\overline{\eta}_{ik1},2)) = \{r_2,s,\overline{e}_{ik},\overline{x}_{ik},\overline{\eta}_{ik1}\} \\
\nu((\overline{\eta}_{ik2},1)) = \{l_2,s,\overline{e}_{ik},\overline{\eta}_{ik2}\} & \nu((\overline{\eta}_{ik2},2)) = \{r_2,s,\overline{p}_{ik},\overline{\eta}_{ik2}\} \\
\nu((\overline{\eta}_{ik3},1)) = \{l_2,s,\overline{p}_{ik},\overline{\eta}_{ik3}\} & \nu((\overline{\eta}_{ik3},2)) = \{r_2,s,\overline{e}_{ik},\overline{\eta}_{ik3}\} \\
\nu((\vartheta_{ij},1)) = \{l_2,s,a_{ij+1},x_{ij},\vartheta_{ij}\} & \nu((\vartheta_{ij},2)) = \{r_2,s,a_{ij},g_{ij+1},\vartheta_{ij}\} \\
\nu((\overline{\vartheta}_{ik},1)) = \{l_2,s,\overline{a}_{ik+1},\overline{\vartheta}_{ik}\} & \nu((\overline{\vartheta}_{ik},2)) = \{r_2,s,\overline{a}_{ik},\overline{g}_{ik+1},\overline{x}_{ik},\overline{\vartheta}_{ik}\} \\
\nu((\iota_i,1)) = \{l_2,s,a_{i1},\iota_i\} & \nu((\iota_i,2)) = \{r_2,s,a,g_{i1},\iota_i\} \ ,
\end{array}$$

where $a_{i|w_i|+1} = \overline{a}_{i1}$, $\overline{a}_{i|\overline{w}_i|+1} = a$, $g_{i|w_i|+1} = \overline{g}_{i1}$ and $\overline{g}_{i|\overline{w}_i|+1} = g_i$, which corresponds to the desired order of introducing letters from Ξ. Later we will also employ the notation $f_{i|w_i|+1} = \overline{f}_{i1}$ and $\overline{f}_{i|\overline{w}_i|+1} = f_i$. A computation of a solution of $\mathscr{P}$ is successful if we can eventually use one of the following rules:

$$\begin{array}{ll}
\nu((\kappa_1,1)) = \{l_2,s,\star,\kappa_1\} & \nu((\kappa_1,2)) = \{r_2,s_{\mathrm{f}},e,\star,\kappa_1\} \\
\nu((\kappa_2,1)) = \{l_2,s_{\mathrm{f}},\star,\kappa_2\} & \nu((\kappa_2,2)) = \{r_2,s,e,\star,\kappa_2\} \ .
\end{array}$$

Finally, for manipulating letters from G we need the rules:

$$\begin{array}{ll}
\nu((\lambda_{ij1},1)) = \{l_2,s,g_{ij},q_{ij},\lambda_{ij1}\} & \nu((\lambda_{ij1},2)) = \{r_2,s,f_{ij},\lambda_{ij1}\} \\
\nu((\lambda_{ij2},1)) = \{l_2,s,f_{ij},\lambda_{ij2}\}, & \nu((\lambda_{ij2},2)) = \{r_2,s,q_{ij},\lambda_{ij2}\} \\
\nu((\lambda_{ij3},1)) = \{l_2,s,q_{ij},\lambda_{ij3}\} & \nu((\lambda_{ij3},2)) = \{r_2,s,f_{ij},\lambda_{ij3}\} \\
\nu((\overline{\lambda}_{ik1},1)) = \{l_2,s,\overline{g}_{ik},\overline{q}_{ik},\overline{\lambda}_{ik1}\} & \nu((\overline{\lambda}_{ik1},2)) = \{r_2,s,\overline{f}_{ik},\overline{\lambda}_{ik1}\} \\
\nu((\overline{\lambda}_{ik2},1)) = \{l_2,s,\overline{f}_{ik},\overline{\lambda}_{ik2}\} & \nu((\overline{\lambda}_{ik2},2)) = \{r_2,s,\overline{q}_{ik},\overline{\lambda}_{ik2}\} \\
\nu((\overline{\lambda}_{ik3},1)) = \{l_2,s,\overline{q}_{ik},\overline{\lambda}_{ik3}\} & \nu((\overline{\lambda}_{ik3},2)) = \{r_2,s,\overline{f}_{ik},\overline{\lambda}_{ik3}\}
\end{array}$$

$$\begin{aligned}
\nu((\xi_{i1},1)) &= \{l_2,s,g_i,q_i,\xi_{i1}\} & \nu((\xi_{i1},2)) &= \{r_2,s,f_i,\xi_{i1}\} \\
\nu((\xi_{i2},1)) &= \{l_2,s,f_i,\xi_{i2}\} & \nu((\xi_{i2},2)) &= \{r_2,s,q_i,\xi_{i2}\} \\
\nu((\xi_{i3},1)) &= \{l_2,s,q_i,\xi_{i3}\} & \nu((\xi_{i3},2)) &= \{r_2,s,f_i,\xi_{i3}\} \,.
\end{aligned}$$

The mapping $\mu : \Sigma \to 2^{\Sigma'}$ is defined by the same rules as the mapping ν except for $\mu((\alpha,1)) = \mu((\alpha,2)) = \emptyset$.

Let φ be the (μ,ν)-weak morphism from $\mathbb{M}(\Sigma,I)$ to $\mathbb{M}(\Sigma',I')$ determined by the conditions $\varphi((\alpha,1)) = l_1 b a_{11}\alpha$, $\varphi((\alpha,2)) = r_1 b s_1 \alpha$ and for all $\omega \in \Omega \setminus \{\alpha,\alpha_0\}$, $z \in \Sigma'$ and $h \in \{1,2\}$:

$$|\varphi((\omega,h))|_z = \begin{cases} 1 & \text{if } z \in \nu((\omega,h)) \,, \\ 0 & \text{otherwise} \,. \end{cases}$$

Now we are ready to formulate the result, which will be proved in Section 4.4.

Proposition 4.2.1. *Let $\mathscr{P}$, (Σ,I), (Σ',I'), μ, ν and φ be as defined above. Then 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 $\mathbb{M}(\Sigma',I')$.

This proposition is the last item we need to construct a reduction of the coPCP to the TCP, thereby proving that the TCP is undecidable.

Proposition 4.2.2. *There exists an effective reduction of the coPCP to the TCP.*

Proof. As the first step, we use Proposition 4.2.1 to reduce the coPCP to the problem of deciding the existence of (μ,ν)-weak codings. Since every instance (consisting of alphabets (Σ,I) and (Σ',I') and mappings μ and ν) constructed there satisfies all assumptions of Proposition 4.1.2, we can employ this claim to prolong the reduction to the problem of deciding the existence of ν-weak codings. Finally, Proposition 4.1.3 can be applied to all instances of this problem obtained in Proposition 4.1.2 and we get a reduction to the TCP. □

Observe that Proposition 4.2.1 immediately implies that injectivity is not decidable for cp-morphisms because for every instance $\mathscr{P}$ of the PCP we have constructed the cp-morphism φ being a coding if and only if $\mathscr{P}$ possesses no solution. This result was first announced in [17] and proved in [16].

Corollary 4.2.3 ([16]). *The restriction of the trace code problem to cp-morphisms is undecidable.*

4.3 Example of the Encoding

Before proceeding to prove Proposition 4.2.1, let us demonstrate its basic principles by an example.

Consider the alphabet $\Xi = \{x, y, z, \#\}$ and the instance $\mathscr{P}$ of the PCP consisting of the pairs $(x\#y\#, x\#)$ and $(z\#, y\#z\#)$. Further, let $\psi : \mathbb{M}(\Sigma, I) \to \mathbb{M}(\Sigma', I')$ be the (μ, ν)-weak morphism defined by the same rules as φ except for:

$$\begin{aligned} \psi((\alpha,1)) &= l_1^3 b s_1 a_{11}^5 \alpha^2 & \psi((\alpha,2)) &= l_1 r_1 b^2 s_1 a_{11} \alpha \\ \psi((\gamma_1,1)) &= l_2 s_{\mathrm{f}}^3 \gamma_1 & \psi((\kappa_1,2)) &= r_2 s_{\mathrm{f}} e \star \kappa_1^2 \\ \psi((\overline{\eta}_{241},1)) &= l_2 s \overline{p}_{24}^3 \overline{\eta}_{241} & \psi((\overline{\eta}_{242},2)) &= r_2 s \overline{p}_{24}^2 \overline{\eta}_{242} \\ \psi((\vartheta_{22},1)) &= l_2 s \overline{a}_{21} \#^2 \vartheta_{22} \end{aligned}$$

We will use the solution $(x\#y\#)(z\#) = (x\#)(y\#z\#)$ of the instance $\mathscr{P}$ to show that the morphism ψ is not a coding. Let us take the semi-equality $\big((\alpha,1),(\alpha,2)^2\big)$, whose state is $(l_1 a_{11}^3, r_1^2 b^3 s_1)$. We are going to successively append elements of Σ from the right to this semi-equality preserving the property that no letter from Ω occurs in its state until we reach a semi-equality satisfying the assumptions of Lemma 2.1.22. During the whole construction we will be interested only in the state of the current semi-equality, which contains enough information to verify that the next addition does not violate the definition of semi-equalities and to calculate the new state.

First we add the pair $\big((\beta_8,1)^3,(\beta_8,2)^3\big)$ to remove b's from the state (in such a case let us simply say that we add the pair β_8^3); the resulting state is $(l_1 l_2^3 a_{11}^3, r_1^2 r_2^3 d^3 s_1)$. Then we use the pair δ_{11} to replace s_1 with s_2. We get the state $(l_1 l_2^4 c a_{11}^3, r_1^2 r_2^4 d^4 s_2 q_1)$. Similarly, appending δ_{21} now produces $(l_1 l_2^5 c^2 a_{11}^3, r_1^2 r_2^5 d^5 s_{11} q_1 q_2)$ and we continue in this way with ε_{111}, ε_{121}, ε_{131}, ε_{141}, ε_{211}, ε_{221}, $\overline{\varepsilon}_{111}$, $\overline{\varepsilon}_{121}$, $\overline{\varepsilon}_{211}$, $\overline{\varepsilon}_{221}$ and $\overline{\varepsilon}_{231}$, thus obtaining the state

$$(l_1 l_2^{16} c^2 a_{11}^3, r_1^2 r_2^{16} d^5 \overline{s}_{24} p_{11} p_{12} p_{13} p_{14} p_{21} p_{22} \overline{p}_{11} \overline{p}_{12} \overline{p}_{21} \overline{p}_{22} \overline{p}_{23} q_1 q_2)\ .$$

Since the images of $(\overline{\eta}_{241},1)$ and $(\overline{\eta}_{242},2)$ under ψ differ from the images under φ, we have to be more careful with introducing the letter $\overline{p}_{24}$ to the state. Later we will see that the right choice is to append the pair $\overline{\varepsilon}_{242}$, which leads to the state

$$(l_1 l_2^{17} c^2 a_{11}^3 \overline{p}_{24}, r_1^2 r_2^{17} d^5 t_{11} p_{11} \cdots \overline{p}_{23} q_1 q_2)\ .$$

Now we deal with ζ's in the same way as with ε's to reach the state

$$(l_1 l_2^{29} c^2 a_{11}^3 \overline{p}_{24}, r_1^2 r_2^{29} d^5 s p_{11} \cdots \overline{p}_{23} q_1 q_2 q_{11} \cdots q_{22} \overline{q}_{11} \cdots \overline{q}_{24})\ .$$

As the letter from S in the current state is s, we can start composing the solution of $\mathscr{P}$ using the rules containing s. First we replace the letter a_{11} with a_{12} by adding the pair ϑ_{11}^3. We obtain the state

$$(l_1 l_2^{32} c^2 a_{12}^3 \overline{p}_{24} x^3, r_1^2 r_2^{32} d^5 s g_{12}^3 p_{11} \cdots \overline{p}_{23} q_1 \cdots \overline{q}_{24}) \ .$$

Now we have to remove all occurrences of g_{12}. In order to do this, we employ the pair λ_{121} exchanging q_{12} for f_{12} :

$$(l_1 l_2^{33} c^2 a_{12}^3 \overline{p}_{24} x^3, r_1^2 r_2^{33} d^5 s f_{12} g_{12}^2 p_{11} \cdots \overline{p}_{23} q_1 q_2 q_{11} q_{13} \cdots \overline{q}_{24}) \ .$$

Then we perform this exchange in the reverse direction using λ_{122} :

$$(l_1 l_2^{34} c^2 a_{12}^3 \overline{p}_{24} x^3, r_1^2 r_2^{34} d^5 s g_{12}^2 p_{11} \cdots \overline{p}_{23} q_1 \cdots \overline{q}_{24}) \ .$$

Repeating the last two steps twice more, we eliminate also the remaining occurrences of g_{12} and we get the state

$$(l_1 l_2^{38} c^2 a_{12}^3 \overline{p}_{24} x^3, r_1^2 r_2^{38} d^5 s p_{11} \cdots \overline{p}_{23} q_1 \cdots \overline{q}_{24}) \ .$$

In the same way we now append the pairs ϑ_{12}^3, λ_{131}, λ_{132}, ϑ_{13}^3, λ_{141}, λ_{142}, ϑ_{14}^3, $\overline{\lambda}_{111}$ and $\overline{\lambda}_{112}$ to reach the state

$$(l_1 l_2^{65} c^2 \overline{a}_{11}^3 \overline{p}_{24} x^3 \#^3 y^3 \#^3, r_1^2 r_2^{65} d^5 s p_{11} \cdots \overline{p}_{23} q_1 \cdots \overline{q}_{24}) \ ,$$

where the reduct of the projection of the first component to Ξ is $x\#y\#$, which is just the word w_1.

Similarly as before, we add $\overline{\vartheta}_{11}^3$, $\overline{\lambda}_{121}$, $\overline{\lambda}_{122}$ and $\overline{\vartheta}_{12}^3$ to remove from the state the letters corresponding to the word $\overline{w}_1$:

$$(l_1 l_2^{77} c^2 a^3 \overline{p}_{24} y^3 \#^3, r_1^2 r_2^{77} d^5 s g_1^3 p_{11} \cdots \overline{p}_{23} q_1 \cdots \overline{q}_{24}) \ .$$

Then we eliminate g_1 using the pairs ξ_{11} and ξ_{12} and start introducing the second pair of words of $\mathscr{P}$, as suggested by the chosen solution, by appending ι_2^3. The resulting state is

$$(l_1 l_2^{86} c^2 a_{21}^3 \overline{p}_{24} y^3 \#^3, r_1^2 r_2^{86} d^5 s g_{21}^3 p_{11} \cdots \overline{p}_{23} q_1 \cdots \overline{q}_{24}) \ .$$

After employing the pairs λ_{211}, λ_{212}, ϑ_{21}^3, λ_{221}, λ_{222}, ϑ_{22}^3, $\overline{\lambda}_{211}$ and $\overline{\lambda}_{212}$, we end up with the state

$$(l_1 l_2^{110} c^2 \overline{a}_{21}^3 \overline{p}_{24} y^3 \#^3 z^3 \#^6, r_1^2 r_2^{110} d^5 s p_{11} \cdots \overline{p}_{23} q_1 \cdots \overline{q}_{24})$$

and the subsequent addition of $\overline{\vartheta}_{21}^3$ $\overline{\lambda}_{221}$, $\overline{\lambda}_{222}$, $\overline{\vartheta}_{22}^3$, $\overline{\lambda}_{231}$, $\overline{\lambda}_{232}$, $\overline{\vartheta}_{23}^3$, $\overline{\lambda}_{241}$ and $\overline{\lambda}_{242}$ transforms it into

$$(l_1 l_2^{137} c^2 \overline{a}_{24}^3 \overline{p}_{24} \#^6, r_1^2 r_2^{137} d^5 s p_{11} \cdots \overline{p}_{23} q_1 \cdots \overline{q}_{24}) \ .$$

In the next step of our construction we have to replace three occurrences of $\overline{a}_{24}$ in the state with a and simultaneously remove six occurrences of the letter #. This cannot be achieved by simply appending $\overline{\vartheta}_{24}^3$, therefore we have to modify the state before proceeding to use the pair $\overline{\vartheta}_{24}$. First we add the pair $\overline{\eta}_{242}$ by means of Lemma 2.1.24, where we take $\overline{p}_{24}$ as the distinguished letter in (2.4). Since the lengths of all blocks in the current state are multiplied by 2 due to Lemma 2.1.23, we reach the state

$$(l_1^2 l_2^{275} c^4 \overline{a}_{24}^6 \overline{e}_{24} \#^{12}, r_1^4 r_2^{275} d^{10} s^2 p_{11}^2 \cdots \overline{p}_{23}^2 q_1^2 \cdots \overline{q}_{24}^2) \ .$$

Then we append $\overline{\eta}_{241}$ in order to exchange $\overline{e}_{24}$ for $\overline{p}_{24}$:

$$(l_1^2 l_2^{276} c^4 \overline{a}_{24}^6 \overline{p}_{24}^3 \#^{11}, r_1^4 r_2^{276} d^{10} s^2 p_{11}^2 \cdots \overline{p}_{23}^2 q_1^2 \cdots \overline{q}_{24}^2) \ .$$

When performing this replacement of letters $\overline{p}_{24}$ and $\overline{e}_{24}$, the ratio of the number of occurrences of $\overline{a}_{24}$ in the state to the number of occurrences of # increases, so we repeat it until these numbers are equal. More precisely, in each step of the iteration the lengths of all blocks in the state are multiplied by 2 and the appropriate number of the pairs $\overline{\eta}_{242}$ and $\overline{\eta}_{241}$ is added, producing in sequence the following states:

$$\begin{gathered}
(l_1^4 l_2^{555} c^8 \overline{a}_{24}^{12} \overline{e}_{24}^3 \#^{22}, r_1^8 r_2^{555} d^{20} s^4 p_{11}^4 \cdots \overline{p}_{23}^4 q_1^4 \cdots \overline{q}_{24}^4) \\
(l_1^4 l_2^{558} c^8 \overline{a}_{24}^{12} \overline{p}_{24}^9 \#^{19}, r_1^8 r_2^{558} d^{20} s^4 p_{11}^4 \cdots \overline{p}_{23}^4 q_1^4 \cdots \overline{q}_{24}^4) \\
(l_1^8 l_2^{1125} c^{16} \overline{a}_{24}^{24} \overline{e}_{24}^9 \#^{38}, r_1^{16} r_2^{1125} d^{40} s^8 p_{11}^8 \cdots \overline{p}_{23}^8 q_1^8 \cdots \overline{q}_{24}^8) \\
(l_1^8 l_2^{1134} c^{16} \overline{a}_{24}^{24} \overline{p}_{24}^{27} \#^{29}, r_1^{16} r_2^{1134} d^{40} s^8 p_{11}^8 \cdots \overline{p}_{23}^8 q_1^8 \cdots \overline{q}_{24}^8) \\
(l_1^{16} l_2^{2295} c^{32} \overline{a}_{24}^{48} \overline{e}_{24}^{27} \#^{58}, r_1^{32} r_2^{2295} d^{80} s^{16} p_{11}^{16} \cdots \overline{p}_{23}^{16} q_1^{16} \cdots \overline{q}_{24}^{16})
\end{gathered}$$

Now there are exactly 10 redundant occurrences of # in the state, which is already less than the current number of occurrences of $\overline{e}_{24}$. Therefore we append only 10 copies of the pair $\overline{\eta}_{241}$ and finish the replacement using the pair $\overline{\eta}_{243}^{17}$ instead. This leads to the state

$$(l_1^{16} l_2^{2322} c^{32} \overline{a}_{24}^{48} \overline{p}_{24}^{47} \#^{48}, r_1^{32} r_2^{2322} d^{80} s^{16} p_{11}^{16} \cdots \overline{p}_{23}^{16} q_1^{16} \cdots \overline{q}_{24}^{16})$$

and we finally apply the pair $\overline{\vartheta}_{24}^{48}$ and the pairs ξ_{21}^{16} and ξ_{22}^{16} three times to obtain

$$(l_1^{16} l_2^{2466} c^{32} a^{48} \overline{p}_{24}^{47}, r_1^{32} r_2^{2466} d^{80} s^{16} p_{11}^{16} \cdots \overline{p}_{23}^{16} q_1^{16} \cdots \overline{q}_{24}^{16}) \ .$$

Notice that the construction we have just employed would not work if the letter $\overline{p}_{24}$ were placed on the other side of the state. In fact, in such a case the pairs $\overline{\eta}_{241}$ and $\overline{\eta}_{242}$ would have to be applied in the reverse order, thus if the replacement were performed iteratively, the ratio of the number of occurrences of $\overline{a}_{24}$ to the number of occurrences of # would converge to $3/5$.

Up to now, it has never been possible to use the rules containing κ's — either because the letter from A in the state was dependent on e or because some letters from Ξ, which are dependent on $\star$, were present in the state. But the current state allows us to append the pair $\big((\kappa_1,1)^{16},(\kappa_1,2)^8\big)$ to replace s with s_{f} :

$$(l_1^{16}l_2^{2482}c^{32}a^{48}\overline{p}_{24}^{47}\star^8, r_1^{32}r_2^{2474}d^{80}s_{\mathrm{f}}^8e^8p_{11}^{16}\cdots\overline{p}_{23}^{16}q_1^{16}\cdots\overline{q}_{24}^{16}) .$$

Finally, we employ the pairs γ_1^{16} and γ_2^{32} to reach a semi-equality with the state

$$(l_2^{2498}c^{32}s_{\mathrm{f}}^{24}a^{48}\overline{p}_{24}^{47}\star^8, r_2^{2506}d^{80}e^8p_{11}^{16}\cdots\overline{p}_{23}^{16}q_1^{16}\cdots\overline{q}_{24}^{16}) ,$$

which satisfies the assumptions of Lemma 2.1.22. This shows that ψ is not a coding.

4.4 Proof of Proposition 4.2.1

Let us start with one useful observation. In this proof we deal solely with weak morphisms $\psi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ satisfying

$$\forall\omega \in \Omega \setminus \{\alpha,\alpha_0\},\ h \in \{1,2\} : \left|\pi_{\Omega\setminus\{\alpha,\alpha_0\}}\big(\psi((\omega,h))\big)\right| = 1 . \tag{4.8}$$

Because all letters in the set $\Omega \setminus \{\alpha,\alpha_0\}$ are mutually dependent, if words $u,v \in \Sigma^*$ satisfy $\psi(u) \sim_{I'} \psi(v)$ for such a morphism ψ, letters from Σ on the same position in the words $\pi_{\Sigma\setminus\{(\alpha,1),(\alpha,2)\}}(u)$ and $\pi_{\Sigma\setminus\{(\alpha,1),(\alpha,2)\}}(v)$ have the same element of Ω on their first coordinate. This allows us to consider only those semi-equalities which respect these pairs. More precisely, the word morphism $\pi_{\Omega\setminus\{\alpha,\alpha_0\}} \circ \psi : \Sigma^* \to \Omega^*$ will be always equal to the morphism τ defined by the rule $\tau|_\Sigma = \pi_{\Omega\setminus\{\alpha\}} \circ \mathrm{p}_1$ and we call a semi-equality (u,v) for ψ *balanced* whenever $\tau(u) = \tau(v)$. Conversely, the state of any balanced semi-equality for a morphism satisfying (4.8) contains no letter from the set $\Omega \setminus \{\alpha,\alpha_0\}$.

Since the implication (ii) $\Longrightarrow$ (iii) is trivial, it is enough to prove the validity of the implications (i) $\Longrightarrow$ (ii) and (iii) $\Longrightarrow$ (i).

4.4.1 Implication (i) $\Longrightarrow$ (ii)

Suppose that the morphism φ is not a coding and let $u,v \in \Sigma^*$, $u \not\sim_I v$, $\varphi(u) \sim_{I'} \varphi(v)$, be a counter-example with minimal $|u|+|v|$. We have to construct a solution to $\mathscr{P}$. In order to do this, we are going to study balanced semi-equalities for φ arising from the pair (u,v) by Lemma 2.1.20, starting from the shortest ones.

First, one has to verify that φ satisfies (2.3). Since the φ-images of (ω_1,h_1) and (ω_2,h_2) for $h_1,h_2 \in \{1,2\}$, $\omega_1,\omega_2 \in \Omega \setminus \{\alpha_0\}$, $\omega_1 \neq \omega_2$, contain different letters from Ω, most of which are dependent, it consists of checking the condition for every pair $(\omega,1)$, $(\omega,2)$ with $\omega \in \Omega \setminus \{\alpha,\alpha_0\}$ and a few pairs of α's and β's.

By Lemma 2.1.1 we have, up to symmetry, $u = (\alpha,1)^{h_1}u_1$ and $v = (\alpha,2)^{h_2}v_1$ for some $h_1,h_2 \in \mathbb{N}$ and $u_1,v_1 \in \Sigma^*$ such that

$$\mathsf{p}_1(\mathrm{first}(u_1)) = \mathsf{p}_1(\mathrm{first}(v_1)) \neq \alpha\ .$$

If h_1 and h_2 differ, then α occurs in the state of the semi-equality $\big((\alpha,1)^{h_1},(\alpha,2)^{h_2}\big)$. Thus $\mathsf{p}_1(\mathrm{first}(u_1)) \in \{\beta_1,\beta_2,\beta_3,\beta_4\}$ because the other elements of $\Omega \setminus \{\alpha,\alpha_0\}$ are dependent on α, so they would violate (1.1) for prefixes of the words $\varphi(u)$ and $\varphi(v)$. But this is also impossible since l_1 and r_1 occur in the state and either l_2 or r_2 is present in both $\varphi(\mathrm{first}(u_1))$ and $\varphi(\mathrm{first}(v_1))$. Therefore $h_1 = h_2$ holds and the reduced state corresponding to the initial blocks of α's is (l_1a_{11},r_1s_1).

Observe that after the initial α's, letters from Σ using r_2 (l_2 respectively) in their images cannot occur in the word u (v) until the letter r_1 (l_1) is removed from the right (left) side of the state. As $\varphi(\mathrm{first}(u_1))$ contains either s_1 or no letter from S, it is just a matter of verification (to deal with β_6, β_7 and β_8 one has to employ also the letters a, b, c, d in the reasoning) to see that the pair $(\mathrm{first}(u_1),\mathrm{first}(v_1))$ is either $\big((\delta_{11},1),(\delta_{11},2)\big)$ or $\big((\delta_{12},1),(\delta_{12},2)\big)$. And after at least one of these pairs either the letter s_1 no longer occurs in the state or only other pairs of δ_1's follow by the same arguments and using the fact $c,d \in D'(b)$ to exclude α's. Actually, the occurrences of the letters c and d in the state prevent us from α's occurring in u and v also further on.

In the proof of this implication, none of the arguments makes use of elements of the set $P \cup Q$; therefore when writing states we will omit them even though they are often present and distributed somehow to both sides of the state.

The reduced state after the pairs of δ_1's is $(l_1l_2ca_{11},r_1r_2ds_2)$. Similarly as before, one can argue that the next pairs in (u,v) are in sequence

$$\delta_2\text{'s},\ldots,\delta_n\text{'s},\varepsilon_{11}\text{'s},\ldots,\overline{\varepsilon}_{n|\overline{w}_n|}\text{'s},\zeta_{11}\text{'s},\ldots,\overline{\zeta}_{n|\overline{w}_n|}\text{'s}$$

due to the changes of the first letter from S in the state. After these pairs, the reduced state is $(l_1l_2ca_{11},r_1r_2ds)$. It is easy to see that only pairs of letters using s can follow in (u,v) until the first occurrence of κ's because in the image of each of these letters there is exactly one occurrence of s, so the number of s's in the state does not change. Moreover, eventually some pair of κ's has to follow; otherwise l_1 and r_1 would never be removed from the state.

Now we want to describe balanced semi-equalities appearing before κ's are used. We define three special forms of these semi-equalities based on which letter from A occurs in their state and we show that after every semi-equality of one of these forms another one can be found. We also determine which semi-equalities allow the addition of a pair of κ's. The projection of the image of such a semi-equality to Ξ is a key to a solution of $\mathscr{P}$.

Let us consider balanced semi-equalities possessing the following projections of their reduced states to $\Sigma' \setminus \Xi$ and projections of their φ-images to Ξ :

$$(l_1 l_2 c a_{i_l j}[f_{i_l j}], r_1 r_2 d s[g_{i_l j}]) \qquad (w'_0 \cdots w'_{l-1} \cdot w'_l\langle j-1\rangle, \overline{w}'_0 \cdots \overline{w}'_{l-1}) \,, \tag{4.9}$$

$$(l_1 l_2 c \overline{a}_{i_l k}[\overline{f}_{i_l k}], r_1 r_2 d s \overline{g}_{i_l k}) \qquad (w'_0 \cdots w'_l, \overline{w}'_0 \cdots \overline{w}'_{l-1} \cdot \overline{w}'_l\langle k-1\rangle) \,, \tag{4.10}$$

$$(l_1 l_2 c a[f_{i_l}], r_1 r_2 d s g_{i_l}) \qquad (w'_0 \cdots w'_l, \overline{w}'_0 \cdots \overline{w}'_l) \,, \tag{4.11}$$

for some $l \in \mathbb{N}$ and some words $w'_{l'}, \overline{w}'_{l'} \in \Xi^*$ satisfying $\operatorname{red}(w'_{l'}) = w_{i_{l'}}$, $\operatorname{red}(\overline{w}'_{l'}) = \overline{w}_{i_{l'}}$ for $l' \in \{0, \ldots, l\}$, where $i_{l'} \in \{1, \ldots, n\}$, $i_0 = 1$, and letters in square brackets do not have to occur. Notice that the semi-equality obtained above is of the form (4.9).

Let us have an arbitrary balanced semi-equality for φ of the form (4.9) arising from the pair (u,v) and consider the first balanced semi-equality after this one such that its state (u',v') satisfies $a_{i_l j} \notin \operatorname{init}(u')$. The only pair which employs elements of A and which can occur when passing between these semi-equalities is $\big((\vartheta_{i_l j},1),(\vartheta_{i_l j},2)\big)$ and it is also the last one since it removes $a_{i_l j}$. Therefore the letter $a_{i_l j}$ is replaced with $a_{i_l j+1}$ in the reduced state. As long as $a_{i_l j}$ is an initial letter on the left side of the state, due to the definition of the relation I' no other elements of $E \cup F \cup G$ than $e_{i_l j}$, $f_{i_l j}$, $g_{i_l j}$ and $g_{i_l j+1}$ can be used in the images of letters of v.

From the above considerations we deduce that several occurrences of the letter $x_{i_l j}$ are introduced into the image of the left side of the semi-equality by occurrences of the pair $\big((\vartheta_{i_l j},1),(\vartheta_{i_l j},2)\big)$ and possibly also of the pair $\big((\eta_{i_l j1},1),(\eta_{i_l j1},2)\big)$ and no more changes in the projection to Ξ are done. These considerations further imply that $g_{i_l j+1}$ cannot be inserted by letters of u and thus it is never present on the left side of the state in the course of passing between these semi-equalities. Because the last pair inserts $g_{i_l j+1}$ into v', there can be only letters independent on $g_{i_l j+1}$ on the left side of the state at that moment; observe that $f_{i_l j+1}$ is the only letter in $E \cup F \cup G$ which satisfies this. And since the last pair inserts also $a_{i_l j+1}$ into u', none of the letters $e_{i_l j}$, $f_{i_l j}$ and $g_{i_l j}$ can occur in v'. Altogether, the new semi-equality has the desired form

$$(l_1 l_2 c a_{i_l j+1}[f_{i_l j+1}], r_1 r_2 d s g_{i_l j+1}) \qquad (w'_0 \cdots w'_{l-1} \cdot w'_l\langle j\rangle, \overline{w}'_0 \cdots \overline{w}'_{l-1}) \,.$$

For a semi-equality of the form (4.10), everything is similar to the previous case and we reach a semi-equality of the form

$$(l_1 l_2 c \overline{a}_{i_l k+1}[\overline{f}_{i_l k+1}], r_1 r_2 d s \overline{g}_{i_l k+1}) \qquad (w'_0 \cdots w'_l, \overline{w}'_0 \cdots \overline{w}'_{l-1} \cdot \overline{w}'_l\langle k\rangle) \,.$$

It remains to deal with the case (4.11). First notice that, unlike in the previous cases, a pair of κ's may be appended here since a is the only letter in A independent on e. Suppose it does not happen. Again, we consider the next balanced semi-equality such that its state (u',v') satisfies $a \notin \operatorname{init}(u')$. But this time one can use any pair

$\big((\iota_i,1),(\iota_i,2)\big)$, where $i \in \{1,\ldots,n\}$, for manipulating letters from A. All letters from the set $E \cup F \cup G$ except e, f_i, g_i and g_{i1} are prohibited to occur in the state on the right due to the occurrences of a on the left. This means that the projection of the image of the semi-equality to Ξ is not modified and no g_{i1} is inserted to the left. Let $\big((\iota_{i_{l+1}},1),(\iota_{i_{l+1}},2)\big)$ be the first pair of ι's used. Once this pair is added, there are some occurrences of $g_{i_{l+1}1}$ on the right side of the state and they do not vanish by the previous considerations. Because $(g_{i_{l+1}1},a_{i1}) \in D'$ for $i \neq i_{l+1}$, only this pair of ι's is allowed also further on. Hence we can finish the reasoning in the same way as in the first case to deduce that the new semi-equality is of the form (4.9):

$$(l_1 l_2 c a_{i_{l+1}1}[f_{i_{l+1}1}], r_1 r_2 d s g_{i_{l+1}1}) \qquad (w'_0 \cdots w'_l, \overline{w}'_0 \cdots \overline{w}'_l)\,.$$

As we have seen, the state of every balanced semi-equality without κ's contains a letter from A on the left. Since the first pair of κ's introduces the first occurrences of e and puts them on the right, it can be used just when the only letter from A in the state is a. From the preceding arguments we can conclude that in such a case the projection of the image of the semi-equality to Ξ is $(w'_0 w'_1 w'_2 \cdots w'_m, \overline{w}'_0 \overline{w}'_1 \overline{w}'_2 \cdots \overline{w}'_m)$ for some $m \in \mathbb{N}_0$. Because the pair of κ's introduces $\star$ for the first time to both sides of the semi-equality, its state contains no letters from Ξ. Hence the sequence $i_1,\ldots,i_m$ is a solution of $\mathscr{P}$ since we have

$$\begin{aligned} w_1 w_{i_1} w_{i_2} \cdots w_{i_m} &= \mathrm{red}(w'_0 w'_1 w'_2 \cdots w'_m) \\ &= \mathrm{red}(\overline{w}'_0 \overline{w}'_1 \overline{w}'_2 \cdots \overline{w}'_m) \\ &= \overline{w}_1 \overline{w}_{i_1} \overline{w}_{i_2} \cdots \overline{w}_{i_m} \end{aligned}$$

due to our assumption on instances of the PCP.

4.4.2 Implication (iii) $\Longrightarrow$ (i)

Assume that $\psi : \mathbb{M}(\Sigma,I) \to \mathbb{M}(\Sigma',I')$ is a (μ,ν)-weak morphism and the sequence $i_1,\ldots,i_m$ is a solution of $\mathscr{P}$. We have to find a counter-example to the injectivity of ψ. By Lemma 2.1.17 it is sufficient to do this for ψ satisfying (4.8). Under this assumption, it makes sense to consider only balanced semi-equalities for ψ as in the first part of the proof. The main course of the proof is a gradual construction of two words $u,v \in \Sigma^*$ satisfying $u \not\sim_I v$ and $\psi(u) \sim_{I'} \psi(v)$. In every step we extend an already constructed balanced semi-equality by appending new pairs from Σ using Lemma 2.1.24.

In order to get an initial semi-equality, we have to consider the numbers

$$\begin{aligned} N_{1l} &= |\psi((\alpha,1))|_{l_1} & N_{1r} &= |\psi((\alpha,1))|_{r_1} & N_{1\alpha} &= |\psi((\alpha,1))|_{\alpha} \\ N_{2l} &= |\psi((\alpha,2))|_{l_1} & N_{2r} &= |\psi((\alpha,2))|_{r_1} & N_{2\alpha} &= |\psi((\alpha,2))|_{\alpha} \end{aligned}$$

and their ratios

$$N_l = \frac{N_{1l}}{N_{2l}} \qquad N_r = \frac{N_{1r}}{N_{2r}} \qquad N_\alpha = \frac{N_{1\alpha}}{N_{2\alpha}} \,.$$

If more than one numerator or more than one denominator are equal to zero, then $\psi((\alpha,1))$, $\psi((\alpha,2))$ respectively, commutes with either $\psi((\beta_1,2))$ or $\psi((\beta_2,1))$ or $\psi((\beta_4,1))$ in $\mathbb{M}(\Sigma',I')$ and ψ is not injective. Otherwise if $N_l = N_r$ or $N_{1r} = N_{2r} = 0$, then we have

$$\psi\big((\alpha,1)^{N_{2l}}(\beta_1,2)(\alpha,2)^{N_{1l}}\big) \sim_{I'} \psi\big((\alpha,2)^{N_{1l}}(\beta_1,2)(\alpha,1)^{N_{2l}}\big) \,.$$

In the same way we get a counter-example for the two symmetric cases using $(\beta_2,1)$ and $(\beta_4,1)$. So the above ratios are pairwise different numbers in $\mathbb{Q}_0^+ \cup \{\infty\}$. Suppose that $N_r < N_l < N_\alpha$ holds. If we consider the semi-equality

$$\big((\alpha,1)^{N_{2l}},(\alpha,2)^{N_{1l}}\big) \,,$$

we can employ Lemma 2.1.24 to add some occurrences of the pair $\big((\beta_1,1),(\beta_1,2)\big)$ in order to replace r_1 in the state of this semi-equality with r_2. Then we do the same with the pair $\big((\beta_2,1),(\beta_2,2)\big)$ to remove α from the state and we obtain a semi-equality which satisfies the assumptions of Lemma 2.1.22, thus showing that ψ is not a coding. Since the case $N_\alpha < N_l < N_r$ can be handled similarly and the same can be shown also if the medium ratio is N_r using $\big((\beta_3,1),(\beta_3,2)\big)$ and $\big((\beta_4,1),(\beta_4,2)\big)$, the medium one must be N_α. In particular, $0 < N_\alpha < \infty$ holds and the letter α is contained in both images $\psi((\alpha,1))$ and $\psi((\alpha,2))$.

Let us start with the semi-equality

$$\big((\alpha,1)^{N_{2\alpha}},(\alpha,2)^{N_{1\alpha}}\big) \tag{4.12}$$

and denote its state by (u',v'). Due to the inequalities between the ratios, the letters l_1 and r_1 occur in the state on different sides. Without loss of generality, let us assume that $l_1 \in \mathrm{alph}(u')$. If $s_1 \notin \mathrm{alph}(u'v')$ then

$$\psi\big((\alpha,1)^{N_{2\alpha}}(\gamma_1,2)(\alpha,2)^{N_{1\alpha}}\big) \sim_{I'} \psi\big((\alpha,2)^{N_{1\alpha}}(\gamma_1,2)(\alpha,1)^{N_{2\alpha}}\big) \,.$$

In the case $s_1 \in \mathrm{alph}(u')$, we have to use Lemma 2.1.24 three times. First, we append the pair $\big((\beta_5,1),(\beta_5,2)\big)$ to (4.12) to reach a state without occurrences of s_1. Then we remove the letters l_1 and r_1 from the state using $\big((\gamma_1,1),(\gamma_1,2)\big)$ and $\big((\gamma_2,1),(\gamma_2,2)\big)$ and finally we apply Lemma 2.1.22 to get a counter-example to the injectivity of ψ. Thus $s_1 \in \mathrm{alph}(v')$. If $a_{11} \notin \mathrm{alph}(u')$, we perform the same construction with the pair $\big((\beta_6,1),(\beta_6,2)\big)$ and all pairs of γ's. Therefore $a_{11} \in \mathrm{alph}(u')$. If either $b \in \mathrm{alph}(u')$ or $b \in \mathrm{alph}(v')$, we remove it from the state using $\big((\beta_7,1),(\beta_7,2)\big)$ or $\big((\beta_8,1),(\beta_8,2)\big)$

respectively. Altogether, we obtain a balanced semi-equality for the morphism ψ with the reduced state $(l_1[l_2][c]a_{11}, r_1[r_2][d]s_1)$.

Now we add to this semi-equality in sequence some pairs of

$$\delta_1\text{'s}, \ldots, \delta_n\text{'s}, \varepsilon_{11}\text{'s}, \ldots, \overline{\varepsilon}_{n|\overline{w}_n|}\text{'s}, \zeta_{11}\text{'s}, \ldots, \overline{\zeta}_{n|\overline{w}_n|}\text{'s}$$

using Lemma 2.1.24 to replace the letter from S in the current state with the next one. In every step of this sequence we have to choose one of two pairs of letters which can be used; in fact, each of them inserts some occurrences of a letter from $P \cup Q$ to a different side of the state. In the case of ε's, we denote

$$J_{ij} = |\psi((\eta_{ij1}, 2))|_{e_{ij}} \cdot |\psi((\eta_{ij2}, 2))|_{p_{ij}}$$
$$K_{ij} = |\psi((\eta_{ij2}, 1))|_{e_{ij}} \cdot |\psi((\eta_{ij1}, 1))|_{p_{ij}}$$

and we append the pair $\big((\varepsilon_{ij1}, 1), (\varepsilon_{ij1}, 2)\big)$ if $J_{ij} \geq K_{ij}$ and the pair $\big((\varepsilon_{ij2}, 1), (\varepsilon_{ij2}, 2)\big)$ otherwise. The reason for this choice is the following. In the next part of the proof, we are going to employ an iterative addition of the pairs $\big((\eta_{ij1}, 1), (\eta_{ij1}, 2)\big)$ and $\big((\eta_{ij2}, 1), (\eta_{ij2}, 2)\big)$ by means of Lemma 2.1.24, replacing the letter p_{ij} in the state with e_{ij} and vice versa. But this construction will function properly only if during the iteration the ratios of the number of occurrences of the letter p_{ij} in the state to the numbers of occurrences of other letters do not decrease. This is achieved by inserting the letter p_{ij} to the appropriate side of the state at the beginning of the construction, thus determining the order in which the pairs of η's will be appended. The choices for δ's, $\overline{\varepsilon}$'s, ζ's and $\overline{\zeta}$'s should be decided in the same way by comparing the numbers of occurrences of the corresponding f's and q's ($\overline{e}$'s and $\overline{p}$'s, f's and q's, $\overline{f}$'s and $\overline{q}$'s respectively) in the images of the corresponding ξ's ($\overline{\eta}$'s, λ's and $\overline{\lambda}$'s respectively).

The reduced state of the non-trivial balanced semi-equality obtained in this way is $(l_1 l_2 c a_{11}, r_1 r_2 d s)$ with letters from $P \cup Q$ distributed to both sides as chosen above. After every step of our construction, letters from $P \cup Q$ will occur on the same sides as in this state and they will be independent on all letters in the state. Again, we omit these letters when describing states, but this time we have to be conscious of the fact just mentioned. Actually, each of these letters is dependent on exactly one letter in Σ', so we do not have to take care of its occurrences in the current state unless the corresponding dependent letter from $E \cup F$ is also in the state.

We are going to gradually construct the following balanced semi-equalities (using the same notation as in the first part of the proof):

$$(l_1 l_2 c[s] a_{i_l j}, r_1 r_2 d[s]) \qquad (w'_0 \cdots w'_{l-1} \cdot w'_l \langle j-1 \rangle, \overline{w}'_0 \cdots \overline{w}'_{l-1}), \tag{4.13}$$
$$(l_1 l_2 c[s] \overline{a}_{i_l k}, r_1 r_2 d[s]) \qquad (w'_0 \cdots w'_l, \overline{w}'_0 \cdots \overline{w}'_{l-1} \cdot \overline{w}'_l \langle k-1 \rangle), \tag{4.14}$$
$$(l_1 l_2 c[s] a, r_1 r_2 d[s]) \qquad (w'_0 \cdots w'_l, \overline{w}'_0 \cdots \overline{w}'_l), \tag{4.15}$$

for every $l \in \{0,\dots,m\}$, $j \in \{1,\dots,|w_{i_l}|\}$ and $k \in \{1,\dots,|\overline{w}_{i_l}|\}$, and for some words $w'_{l'}, \overline{w}'_{l'} \in \Xi^*$, for $l' \in \{0,\dots,l\}$, such that $\mathrm{red}(w'_{l'}) = w_{i_{l'}}$ and $\mathrm{red}(\overline{w}'_{l'}) = \overline{w}_{i_{l'}}$, where $i_0 = 1$ and if we denote by $(w,\overline{w})$ the projection of the semi-equality to Ξ, it satisfies either $w = \overline{w}$, or $w \prec \overline{w}$ together with $\mathrm{first}(w \backslash \overline{w}) \neq \mathrm{last}(w)$, or dually $\overline{w} \prec w$ and $\mathrm{first}(\overline{w} \backslash w) \neq \mathrm{last}(\overline{w})$. Observe that each block in the projection to Ξ corresponds to a unique letter of the word $w_1 w_{i_1} w_{i_2} \cdots w_{i_m}$ due to our assumption on instances of the PCP and the last condition actually states that the corresponding blocks on the opposite sides of the projection to Ξ have the same length. Let us remark that we do not require the words $w'_{l'}$ and $\overline{w}'_{l'}$ for distinct semi-equalities to be equal.

Since for $l = m$ any state of the form (4.15) contains no letters from Ξ, we can use one of the pairs of κ's to exchange s for s_{f} if there is some in the state, then the pairs of γ's to remove l_1 and r_1 and finally Lemma 2.1.22 to get a counter-example.

To finish the proof, it remains to construct the desired semi-equalities. All of the following manipulations are based purely on ratios between the lengths of blocks in the state, which allows us to modify the current semi-equality arbitrarily provided we do not affect the ratios having impact on some step of the construction. We show how to proceed from the state (4.13); the cases of (4.14) and (4.15) are similar.

Our first task is to introduce $x_{i_l j}$. If $w'_0 \cdots w'_{l-1} \cdot w'_l \langle j-1 \rangle \succeq \overline{w}'_0 \cdots \overline{w}'_{l-1}$ then we just add the pair $\big((\vartheta_{i_l j},1),(\vartheta_{i_l j},2)\big)$ by means of Lemma 2.1.24 to replace occurrences of $a_{i_l j}$ in the state with $a_{i_l j+1}$. If it is not the case, the $x_{i_l j}$-block corresponding in the image of the semi-equality to the block we are going to build is already present in the word $\overline{w}'_0 \cdots \overline{w}'_{l-1}$ and we have to insert exactly the number of $x_{i_l j}$'s to match this existing block. Let L_x be the length of this block and L_a be the number of occurrences of $a_{i_l j}$ in the state.

First suppose that

$$\frac{L_x}{L_a} \geq \frac{|\psi((\vartheta_{i_l j},1))|_{x_{i_l j}}}{|\psi((\vartheta_{i_l j},2))|_{a_{i_l j}}} \,. \tag{4.16}$$

The following arguments employ the construction of Lemma 2.1.24 although the assumption (2.5) is not satisfied. As $x_{i_l j}$ will be the only letter violating (2.5), it will suffice to take care of this single letter in order to ensure that the construction really produces a semi-equality. We would like to add the pair $\big((\vartheta_{i_l j},1),(\vartheta_{i_l j},2)\big)$ to replace $a_{i_l j}$ with $a_{i_l j+1}$ and at the same time match the corresponding $x_{i_l j}$-blocks. This can be done directly if (4.16) is satisfied as an equality. In the case the inequality is strict, we insert new occurrences of $x_{i_l j}$ to the left prior to appending ϑ's as follows. If $J_{i_l j} \geq K_{i_l j}$ then $p_{i_l j}$ occurs in the state on the right thanks to the choice of ε's and we can repeatedly append the pairs $\big((\eta_{i_l j1},1),(\eta_{i_l j1},2)\big)$ and $\big((\eta_{i_l j2},1),(\eta_{i_l j2},2)\big)$ to replace $p_{i_l j}$ with $e_{i_l j}$ and vice versa. (If $J_{i_l j} < K_{i_l j}$, these pairs have to be appended in the reverse order.) Let us calculate the state resulting from performing this exchange

M times.

First of all, we can assume that $L_a = |\psi((\vartheta_{i_l j}, 2))|_{a_{i_l j}}$ and that $|\psi((\eta_{i_l j1}, 1))|_{x_{i_l j}}$ divides $L_x - |\psi((\vartheta_{i_l j}, 1))|_{x_{i_l j}}$; in order to satisfy these two conditions it is enough to replace each letter in the semi-equality with its

$$|\psi((\eta_{i_l j1}, 1))|_{x_{i_l j}} \cdot |\psi((\vartheta_{i_l j}, 2))|_{a_{i_l j}} \text{ copies}$$

using Lemma 2.1.23 and to consider instead of the pair $\big((\vartheta_{i_l j}, 1), (\vartheta_{i_l j}, 2)\big)$ its

$$\big(|\psi((\eta_{i_l j1}, 1))|_{x_{i_l j}} \cdot L_a\big)\text{-th power.}$$

For every letter, the number of its occurrences coming from the state of the original semi-equality is multiplied by $(K_{i_l j})^M$ due to the modification of the semi-equality performed in Lemma 2.1.24. On the other hand, the number of occurrences of $p_{i_l j}$ in the resulting state is $L_p \cdot (J_{i_l j})^M$, where L_p is their number in the original state. We are interested mainly in the first $x_{i_l j}$-block on the right side of the state. Every iteration inserts new occurrences of the letter $x_{i_l j}$ to the left. Their number is determined by the current number of $p_{i_l j}$'s in the state and subsequently it is multiplied by the same amount as for all letters in the semi-equality. Thus the resulting length of this block is

$$L'_x = L_x \cdot (K_{i_l j})^M - \\ - L_p \cdot |\psi((\eta_{i_l j1}, 1))|_{x_{i_l j}} \cdot |\psi((\eta_{i_l j2}, 1))|_{e_{i_l j}} \cdot \sum_{N=0}^{M-1} (J_{i_l j})^N \cdot (K_{i_l j})^{M-N-1} .$$

Our goal is to make (4.16) an equality in the new semi-equality. Due to our initial assumption on L_a, this can be written as

$$L'_x = (K_{i_l j})^M \cdot |\psi((\vartheta_{i_l j}, 1))|_{x_{i_l j}} . \tag{4.17}$$

Since the inequality $J_{i_l j} \geq K_{i_l j}$ implies

$$\sum_{N=0}^{M-1} (J_{i_l j})^N \cdot (K_{i_l j})^{M-N-1} \geq M \cdot (K_{i_l j})^{M-1} ,$$

one can see that if M is sufficiently large, then L'_x is even smaller than the number requested in (4.17). By our assumption on L_x, the difference between the length of the first $x_{i_l j}$-block in the state and the number in (4.17) is after every step of the iteration divisible by $|\psi((\eta_{i_l j1}, 1))|_{x_{i_l j}}$, which is exactly the amount inserted to the image of a semi-equality by a single pair $\big((\eta_{i_l j1}, 1), (\eta_{i_l j1}, 2)\big)$. Therefore, if we consider the step of the iteration during which L'_x lowers under the desired value, we can construct a balanced semi-equality where the ratio of the length of the first $x_{i_l j}$-block in the state to the number of occurrences of $a_{i_l j}$ satisfies (4.16) as an equality by appending appropriately less number of the pairs $\big((\eta_{i_l j1},1),(\eta_{i_l j1},2)\big)$ in this step. Then we add to the resulting semi-equality the pairs $\big((\eta_{i_l j3},1),(\eta_{i_l j3},2)\big)$ and $\big((\eta_{i_l j2},1),(\eta_{i_l j2},2)\big)$ using Lemma 2.1.24 to replace all remaining occurrences of $p_{i_l j}$ in the state with $e_{i_l j}$ and back without further changing this ratio. This makes the semi-equality suitable for the addition of ϑ's replacing $a_{i_l j}$ with $a_{i_l j+1}$ and simultaneously matching the $x_{i_l j}$-blocks.

Now suppose that (4.16) is not valid. Then the $x_{i_l j}$-block we want to match is not long enough to allow appending of the required number of ϑ's. Let us consider the initial part of the current semi-equality where building of this block started, which is a balanced semi-equality of the form (4.14):

$$\langle l_1 l_2 c[s]\overline{a}_{i_{\hat{l}}k}, r_1 r_2 d[s]\rangle \qquad \langle w'_0 \cdots w'_{\hat{l}}, \overline{w}'_0 \cdots \overline{w}'_{\hat{l}-1} \cdot \overline{w}'_{\hat{l}}\langle k-1\rangle\rangle\ , \tag{4.18}$$

for certain $\hat{l} \in \{0,\ldots,l-1\}$ and $k \in \{1,\ldots,|\overline{w}_{i_{\hat{l}}}|\}$ satisfying in particular $\overline{x}_{i_{\hat{l}}k} = x_{i_l j}$. In order to achieve the validity of the condition (4.16), we have to modify the step of the construction corresponding to the semi-equality (4.18) by the same means as above using $\overline{\eta}$'s instead of η's to increase the length of the $\overline{x}_{i_{\hat{l}}k}$-block in proportion to the number of $\overline{a}_{i_{\hat{l}}k}$'s in the state. Let us verify that such a modification produces the desired outcome.

First observe that during the addition of $\overline{\eta}$'s the lengths of all blocks in the state are multiplied by the same positive integer K except for the blocks of l_2, r_2, s, $\overline{p}_{i_{\hat{l}}k}$ and the block of $\overline{x}_{i_{\hat{l}}k}$ under consideration. The pairs appended to the semi-equality (4.18) when building the semi-equality (4.13) do not employ in their images letters dependent on l_2, r_2, s and $\overline{p}_{i_{\hat{l}}k}$; as for the letter $\overline{p}_{i_{\hat{l}}k}$, it is due to the fact that $\overline{e}$'s are introduced only when we adjust the number of occurrences of some letter from Ξ on the right to match the corresponding block on the left, which was not yet carried out for elements of Ξ on the right inserted after the semi-equality (4.18). The length of the modified block of $\overline{x}_{i_{\hat{l}}k}$ has also no impact on these additions since the corresponding letters on the left are not inserted. Therefore it suffices to append to the modified semi-equality (4.18) the same pairs as we did for the original one, taking each of them K-times.

From the calculations performed previously for a similar situation, we can see that by iterating the addition of $\overline{\eta}$'s the ratio of the length of the $\overline{x}_{i_{\hat{l}}k}$-block to the number of occurrences of $\overline{a}_{i_{\hat{l}}k}$ converges to infinity. Further, the number of occurrences of $a_{i_l j}$ in the state of the modified semi-equality (4.13) is obtained as a constant multiple of the number of occurrences of $\overline{a}_{i_{\hat{l}}k}$ in the state of the modified semi-equality (4.18), where the constant is determined by the ratios between the numbers of occurrences of letters from A in the images of ϑ's, $\overline{\vartheta}$'s and ι's. Altogether, since the length of the $\overline{x}_{i_l k}$-block in the two modified semi-equalities is the same, if we perform sufficiently many iterations, the modified semi-equality (4.13) satisfies (4.16).

After introducing $x_{i_l j}$, the semi-equality state is

$$(l_1 l_2 c[s] a_{i_l j+1}, r_1 r_2 d[s] g_{i_l j+1}) \qquad (w'_0 \cdots w'_{l-1} \cdot w'_l \langle j \rangle, \overline{w}'_0 \cdots \overline{w}'_{l-1}) \ .$$

For removing occurrences of $g_{i_l j+1}$ from the state we can employ the same method as for the previous matching of corresponding $x_{i_l j}$-blocks using $\lambda_{i_l j+1}$'s instead of $\eta_{i_l j}$'s, where $\lambda_{i_l |w_{i_l}|+1} = \overline{\lambda}_{i_l 1}$. We obtain either the next state of the form (4.13) or a state of the form (4.14) if $j = |w_{i_l}|$.

Obviously, in this way some semi-equality of each of the forms (4.13), (4.14) and (4.15) for every admissible j, k and l will be eventually constructed. $\square$

Chapter 5

Conclusions

In this thesis we have introduced the notion of weak morphisms of trace monoids which appears to be a useful tool for exploring decidability issues of trace codings. This is already suggested by the following claim obtained due to Corollary 2.2.7.

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.* $\square$

The next result, which immediately follows from Proposition 2.2.9, was proved by shifting calculations to the case of weak codings.

Theorem 5.2. *The TCP is effectively reducible to instances with domain monoids defined by connected dependence alphabets.* $\square$

By examining all subtrees of domain dependence graphs we have achieved several positive results for the $\mathscr{W}$-TCP.

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.*

Proof. Since there are only finitely many candidates for wlt-mappings for every pair of independence alphabets and all the conditions of Definition 3.1.2 can be easily verified, this claim is a direct consequence of Propositions 3.2.3 and 3.3.4. $\square$

Because the reduction described in Proposition 2.2.5 preserves the domain monoid, we immediately deduce the following statements about general codings.

Corollary 5.4 ([7]). *The restriction of the TCP to instances whose domain monoids are direct products of free monoids is decidable.*

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

Using Proposition 3.3.4 we can partially answer the question of Diekert [14] asking for the number of free monoids needed for encoding a given trace monoid into their direct product.

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 $m \geq |\Sigma| - M$, where M is the number of non-trivial connected components of the graph (Σ, D).* □

On the other hand, as we have shown, our methods become insufficient for domain dependence alphabets containing cycles. This contrasts with Proposition 3.3.4 which asserts that in the case of codings into direct products of free monoids this limit moves as far as one could expect in view of the fact that subtrees of the domain dependence alphabet cannot capture enough properties of a morphism when each of two dependent letters occurs in the images of at least two generators of the domain monoid.

As for the general case of the TCP, Proposition 4.2.2 shows that it is not decidable.

Theorem 5.7. *The TCP is not recursively enumerable.* □

The same assertion for certain classes of trace morphisms immediately follows due to Theorem 5.1.

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.* □

Notice that the above results tell us nothing about the recursive enumerability of the coTCP, which remains a challenging open question.

An important special case of the TCP where the methods presented in this thesis fail to produce positive results is the restriction to domain monoids which are free products of free commutative monoids. Moreover, as all domain monoids resulting from applying Propositions 4.2.1 and 4.1.2 are of this form, we can conclude that the existence problem for ν-weak codings is undecidable in this case.

Bibliography

[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] IJ.J. Aalbersberg, E. Welzl, Trace languages defined by regular string languages, *RAIRO — Inform. Théor. Appl.* **20**(2) (1986) 103–119.

[3] J.A. Bergstra, J.W. Klop, Process algebra for synchronous communication, *Inform. and Control* **60** (1984) 109–137.

[4] J. Berstel, *Transductions and Context-Free Languages*, Teubner, Stuttgart, 1979.

[5] J. Berstel, D. Perrin, *Theory of Codes*, Pure and Applied Mathematics, vol. 117, Academic Press, Orlando, 1985.

[6] A. Bertoni, G. Mauri, N. Sabadini, Unambiguous regular trace languages, in: J. Demetrovics, G. Katona, A. Salomaa (Eds.), *Algebra, Combinatorics and Logic in Computer Science I*, Colloq. Math. Soc. János Bolyai, vol. 42, North Holland, Amsterdam, 1986, pp. 113–123.

[7] 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.

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

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

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

[11] P. Cartier, D. Foata, *Problèmes combinatoires de commutation et réarrangements*, Lecture Notes in Math., vol. 85, Springer, Berlin, 1969.

[12] M. Chrobak, W. Rytter, Unique decipherability for partially commutative alphabets, *Fund. Inform.* **10** (1987) 323–336.

[13] R. Cori, D. Perrin, Automates et commutations partielles, *RAIRO — Inform. Théor. Appl.* **19**(1) (1985) 21–32.

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

[15] V. Diekert, Y. Métivier, Partial commutation and traces, in: G. Rozenberg, A. Salomaa (Eds.), *Handbook of Formal Languages, vol. 3*, Springer, Berlin, 1997, pp. 457–533.

[16] 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.

[17] 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.

[18] V. Diekert, G. Rozenberg (Eds.), *The Book of Traces*, World Scientific, Singapore, 1995.

[19] C. Duboc, On some equations in free partially commutative monoids, *Theoret. Comput. Sci.* **46** (1986) 159–174.

[20] C.C. Elgot, J.E. Mezei, On relations defined by generalized finite automata, *IBM J. Res. Dev.* **9**(1) (1965) 47–68.

[21] A. Gibbons, W. Rytter, On the decidability of some problems about rational subsets of free partially commutative monoids, *Theoret. Comput. Sci.* **48**(2–3) (1986) 329–337.

[22] C.A.R. Hoare, *Communicating Sequential Processes*, Prentice Hall Internat. Ser. Comput. Sci., Prentice Hall, Englewood Cliffs, New Jersey, 1985.

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

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

[25] O.H. Ibarra, Reversal-bounded multicounter machines and their decision problems, *J. ACM* **25**(1) (1978) 116–133.

[26] 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.

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

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

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

[30] R. Milner, *A Calculus of Communicating Systems*, Lecture Notes in Comput. Sci., vol. 92, Springer, Berlin, 1980.

[31] R. Morin, On regular Message Sequence Chart languages and relationships to Mazurkiewicz trace theory, in: F. Honsell, M. Miculan (Eds.), *Foundations of Software Science and Computation Structures, Proc. 4th Internat. Conf., FOSSACS'01,* Lecture Notes in Comput. Sci., vol. 2030, Springer, Berlin, 2001, pp. 332–346.

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

[33] 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.

[34] M.O. Rabin, D. Scott, Finite automata and their decision problems, *IBM J. Res. Dev.* **3**(2) (1959) 114–125.

[35] J. Sakarovitch, On regular trace languages, *Theoret. Comput. Sci.* **52** (1987) 59–75.

[36] J. Sakarovitch, The “last” decision problem for rational trace languages, in: I. Simon (Ed.), *Proc. 1st Latin American Sympos. on Theoretical Informatics (LATIN’92)*, Lecture Notes in Comput. Sci., vol. 583, Springer, Berlin, 1992, pp. 460–473.

[37] A.A. Sardinas, G.W. Patterson, A necessary and sufficient condition for the unique decomposition of coded messages, in: *IRE Convention Record*, part 8, 1953, pp. 104–108.

Index