ELSEVIER Theoretical Computer Science 179 (1997) 61–102

Theoretical Computer Science

Fundamental Study

Restrictions and representations of vector controlled concurrent system behaviours

N.W. Keesmaat[a], H.C.M. Kleijn[b,*]

[a] KPN Research, St. Paulusstraat 4, P.O. Box 421, 2260 AK Leidschendam, Netherlands
[b] Department of Computer Science, Leiden University, P.O. Box 9512, 2300 RA Leiden, Netherlands

Received June 1995; revised February 1996
Communicated by G. Rozenberg

Abstract

Within the framework of Vector Controlled Concurrent Systems a concurrent system consists of a fixed number of sequential processes together with a vector synchronization mechanism controlling their mutual synchronization. The behaviour of a VCCS is described by a vector language consisting of those combinations of individual sequential computations that observe the synchronization constraints.

In this paper VCCS submodels are studied that are obtained by putting certain restrictions on the sequential components or on the control mechanism. First, the inclusion diagram relating the resulting families of vector languages is established. Next, the effect of certain operations on these families is investigated. This leads to representation results characterizing differences between the combinations of restrictions.

Contents

* Corresponding author. Tel.: +31 71277064; fax: 31 71276985.

0. Introduction

In a *Vector Controlled Concurrent System* (VCCS) a fixed number of sequential processes operate concurrently subject to the control of a vector synchronization mechanism, which imposes constraints on their mutual synchronization. The behaviours of the sequential processes are specified as languages (over alphabets of actions). The synchronization constraints are given in the form of a language over an alphabet of vectors; these vectors express synchronization of actions from the sequential components, while the language gives all permitted sequences of such synchronizations.

The behaviour of the VCCS is described by a *vector language* consisting of those combinations of computations (i.e. sequences of actions) of the components of the system that satisfy the synchronization constraints.

Vector Controlled Concurrent Systems have been introduced in [15] and were further investigated in [16–18]. The original idea underlying the VCCS model comes from the theory of path expressions (the COSY variant), and its vector firing sequence semantics. Path expressions were introduced in [7], while the COSY approach has its starting point in [19]. Vector firing sequences were introduced in [22] as a semantics of COSY systems. See also the book [13] and its list of references. Also in [3, 20, 1] models closely related to the VCCS model have been studied.

Vector Controlled Concurrent Systems have been introduced with the aim of providing a general uniform framework for the investigation of systems in which a fixed number of sequential processes work concurrently but synchronize on certain events. This framework is flexible in the sense that it allows to specify separately the component processes and the synchronization mechanism. By imposing restrictions on the components and/or the control of the synchronization, one obtains different VCCS submodels. For instance, the path expression model itself and the concurrent systems studied in [1] can directly be interpreted as instances of the VCCS model. Various submodels have also been defined in [15] and have been further investigated in [16]. In [17], for a whole range of VCCS submodels, the effect of vector synchronization on the behaviour of the sequential components has been investigated.

The aim of this paper is to compare various natural restrictions both on the behaviour of the sequential processes and on the control languages used. In particular, *regularity* and *prefix-closedness* are properties that arise in the context of concurrent systems. Regularity corresponds to having an underlying finite state device, while prefix-closedness reflects the idea of having ongoing computations. In COSY both restrictions are assumed most of the time (see, e.g., [13]) and they also frequently occur in the theory presented by Nivat and Arnold.

In addition, we consider the use of *monoids*, both as component languages and as control languages. These monoids are languages containing all possible sequences of letters (vectors respectively) from their alphabets. They are interesting as they are permissive in the sense that they provide all opportunities and cannot exclude certain events (synchronizations) from happening. Thus, when used as component languages they allow to focus on the study of the control mechanism, and when used as a control language they lead to more insight in the interplay between the components and pure vector synchronization.

Finally, we consider control languages satisfying the *completeness* property. This property originates from a net-based synchronization mechanism (the *Individual Token Net Controller*, or ITNC for short) introduced in [15] which formalizes the idea of a distributed finite state control for the synchronization of the components. In [18] this mechanism is compared with the finite state control from [1] and given a characterization in language theoretic terms.

For all combinations of these restrictions on sequential components and control mechanisms, we investigate the differences and similarities of the resulting VCCS submodels.

An obvious first step is to compare the expressive power of these VCCS submodels in terms of the vector languages they define. This leads to an inclusion diagram describing equalities, strict inclusions, and incomparabilities between the various families of vector languages.

Next, we investigate the effect of certain operations when applied to vector languages from the families of the diagram. These operations transform vector languages into vector languages. Thus, for each family of vector languages application of an operation of a certain type leads to a new family of vector languages. The operations we consider are such that each new family extends the original family. If the new family coincides with a family from the diagram which properly includes the original family, then the behavioural difference between the two underlying VCCS models is characterized by this type of operation.

This approach of relating different systems by representing the larger class of behaviours in terms of the smaller one using a perferably simple type of operation is a well-known technique and has for specific VCCS submodels already been applied in [1, 16].

For instance, in [16] it has been shown that the application of *multi codings*, i.e., products of letter-to-letter homomorphisms, to the vector languages defined by VCCSs with regular components and a monoid control yields precisely the vector languages of the VCCSs with regular components and ITNCs as control mechanism. Such multi codings have also been used in [1] to relate different families of vector languages.

In [1] it is also shown that the behaviour of a VCCS with prefix-closed regular components and a prefix-closed regular control language can be obtained as the vector language of a VCCS with prefix-closed regular components and a monoid control language from which one (auxiliary) component has been hidden.

Two of the operations we consider are again the *multi codings* and the *hiding* operator of [1]. Multi codings can be used to erase state information from actions: different symbols may be mapped to the same symbol, thus they can be viewed as being different incarnations of one action. Hiding serves to get rid of an extra component used in the computations of the systems to enhance the effect of the control mechanism.

The third operation we involve in our considerations has features in common with both hiding and multi codings: a *multi weak coding* allows to map different symbols to the same symbol, but it can also hide symbols by mapping them to the empty string.

The paper is organized as follows. After the preliminaries of Section 1, where we introduce the VCCS model formally as well as our notational conventions with respect to vectors and vector languages, Section 2 discusses the VCCS submodels considered in the paper, together with a first overview of their mutual relationships in terms of the vector languages they define. Then, in Section 3, the complete inclusion diagram of the various VCCS vector language families is presented. In Section 4, the investigation of the relationships expressed in terms of operations is started by formally introducing the three chosen operations with background and motivations. In Section 5, the main part of the paper, we present the relations obtained between our families by using those operations. The results here include results from [1, 16], and answer a number of open questions from [1]. Finally, in a concluding section we briefly discuss the general ideas emerging from the systematic set-up of our representation results as well as a number of topics for further research.

1. Preliminaries

Throughout this paper the reader is assumed to be familiar with the basic concepts and terminology of formal language theory as presented in, e.g, [12]. From the theory of rational relations some basic results are used. For these results we use [4] as a general reference.

In this preliminary section we fix some terminology and notation that may not be familiar to all readers. In particular, in Section 1.1, the terminology concerning vectors, vector languages, etc., is explained. Basic terminology concerning Vector Controlled Concurrent Systems is recalled in Section 1.2.

1.1. Basic terminology and notations

For each positive integer n, $[n]$ denotes the set $\{1,\dots,n\}$. Function composition is denoted by $\circ$. The set difference between two sets V and W is denoted $V-W$.

Let f be a function and let A and B be sets. Then $f[B] = \{f(b) \mid b \in B\}$ and $f^{-1}[A] = \{a \mid f(a) \in A\}$. Observe that $f[f^{-1}[A] \cap B] = A \cap f[B]$. This observation, to which we refer as the function-intersection rule, will be used often. If no confusion arises, we write $f(B)$ instead of $f[B]$ and $f^{-1}(A)$ instead of $f^{-1}[A]$.

Alphabets are finite. The empty word is denoted by Λ.

Let Σ and Δ be two alphabets. A homomorphism $\rho : \Sigma^* \to \Delta^*$ is called a weak coding if $\rho(\Sigma) \subseteq \Delta \cup \{\Lambda\}$. It is called a coding if $\rho(\Sigma) \subseteq \Delta$.

The reverse of a word $a_1 \dots a_m$, where $m \geqslant 0$, and $a_1, \dots, a_m$ are letters, is the word $a_m \dots a_1$.

A word v is a prefix of a word w if $w = vu$ for some word u. If $u \neq \Lambda$, then v is called a proper prefix of w. For a language K, $\mathbf{pref}(K) = \{v \mid v \text{ is a prefix of a word from } K\}$. Clearly, for all languages K, $K \subseteq \mathbf{pref}(K)$. If $K = \mathbf{pref}(K)$, then K is called prefix-closed. A family of languages $\mathbb{K}$ is closed under **pref** if $\mathbf{pref}(K) \in \mathbb{K}$ for all $K \in \mathbb{K}$. For any family $\mathbb{K}$ of languages, $p\mathbb{K} \subseteq \mathbb{K}$ will denote the subfamily of all prefix-closed languages from $\mathbb{K}$.

An element of a cartesian product of sets is called a vector. Vectors are denoted either horizontally or vertically. For a set V, the set of 1-dimensional vectors $\{(x) \mid x \in V\}$ is denoted by XV. For an n-dimensional vector $w = (w_1, \dots, w_n)$, where $n \geqslant 1$, and for $i \in [n]$, $\mathbf{proj}_i(w)$ denotes the projection of w on its ith component, i.e., $\mathbf{proj}_i(w) = w_i$.

Let $f_i : A_i \to B_i$ be functions for $i \in [n]$, where $n \geqslant 1$. Then $f_1 \times \cdots \times f_n : A_1 \times \cdots \times A_n \to B_1 \times \cdots \times B_n$ is the function defined by $(f_1 \times \cdots \times f_n)((a_1, \dots, a_n)) = (f_1(a_1), \dots, f_n(a_n))$, for all $(a_1, \dots, a_n) \in A_1 \times \cdots \times A_n$. Thus, $f_1 \times \cdots \times f_n$ may be viewed as the product of $f_1, \dots, f_n$.

Let $n \geqslant 1$.

An (n-dimensional) vector v having words as components is called an (n-dimensional) word vector. A component of v consisting of the empty word Λ is called a Λ-component or empty component of v. The position of the Λ-components of v determines its Λ-structure; formally the Λ-structure of v is identified with the set $\{i \in [n] \mid \mathbf{proj}_i(v) = \Lambda\}$.

The (n-dimensional) empty word vector is the n-dimensional word vector $(\Lambda, \dots, \Lambda)$, which is denoted by $\bar{\Lambda}$ if its dimension is clear from the context.

Two n-dimensional word vectors v and w are dependent, if they have a common non-empty component, i.e., there is an $i \in [n]$, such that both $\mathbf{proj}_i(v) \neq \Lambda$ and $\mathbf{proj}_i(w) \neq \Lambda$. Word vectors that are not dependent, are called independent. Note that the Λ-structure of v and w determines whether v and w are independent or dependent. The component-wise concatenation of v and w is denoted by $v \odot w$. If v and w are independent, then $v \odot w = w \odot v$.

As an illustration of some of the notions introduced above, consider the word vectors $u = \binom{ab}{c}$, $v = \binom{ab}{\Lambda}$, and $w = \binom{\Lambda}{ba}$. Then v and w are independent and $v \odot w = \binom{ab}{ba} = w \odot v$. In contrast to this u and v are dependent, as well as u and w. Note that $u \odot w \neq w \odot u$, but $u \odot v = v \odot u = \binom{abab}{c}$.

A set of (n-dimensional) word vectors is called an (n-dimensional) vector language. For n-dimensional vector languages V and W, we extend the operation $\odot$ in the usual way: $V \odot W = \{v \odot w \mid v \in V, w \in W\}$. The iterated component-wise concatenation of a vector language V is the set $V^{\circledast} = \{w_1 \odot \cdots \odot w_m \mid m \geqslant 1, w_1, \dots, w_m \in V\} \cup \{\bar{\Lambda}\}$.

Let $\Sigma_1, \dots, \Sigma_n$ be alphabets. Any vector $\alpha \in ((\Sigma_1 \cup \{\Lambda\}) \times \cdots \times (\Sigma_n \cup \{\Lambda\})) - \{\bar{\Lambda}\}$ is called an (n-dimensional) vector letter (over $\Sigma_1, \dots, \Sigma_n$). A finite and possibly empty set of (n-dimensional) vector letters is called an (n-dimensional) vector alphabet. The set of all vector letters over $\Sigma_1, \dots, \Sigma_n$ is called the total vector alphabet over $\Sigma_1, \dots, \Sigma_n$. It is denoted by $\mathbf{Tot}(\Sigma_1, \dots, \Sigma_n)$. Vector alphabets are alphabets. Therefore, all terminology and notations for alphabets, words and languages apply. Sometimes a language over an n-dimensional vector alphabet θ is called shortly an n-language (over θ). Note that, for vector letters $\alpha, \beta \in \theta$, the word $\alpha\beta \in \theta^*$ differs from the word vector $\alpha \odot \beta \in \theta^{\circledast}$.

Thus, e.g. given the vector letters $\alpha = \binom{\Lambda}{c}$, and $\beta = \binom{a}{b}$, the word $\alpha\beta = \binom{\Lambda}{c}\binom{a}{b}$ comes from a 2-language over $\{\alpha,\beta\}$, and the word vector $\alpha \odot \beta = \binom{a}{cb}$ from a 2-dimensional vector language over $\{\alpha,\beta\}$.

Let θ be an n-dimensional vector alphabet. The canonical homomorphism from θ^* to $\theta^{\circledast}$ collapses words over θ to word vectors in $\theta^{\circledast}$ and is denoted $\mathbf{coll}_\theta$ or **coll** if no confusion arises.

Thus, e.g.,

$$\mathbf{coll}\left(\begin{pmatrix}\Lambda\\c\end{pmatrix}\begin{pmatrix}a\\b\end{pmatrix}\begin{pmatrix}a\\\Lambda\end{pmatrix}\right) = \begin{pmatrix}\Lambda\\c\end{pmatrix} \odot \begin{pmatrix}a\\b\end{pmatrix} \odot \begin{pmatrix}a\\\Lambda\end{pmatrix} = \begin{pmatrix}aa\\cb\end{pmatrix} \quad \text{and} \quad \mathbf{coll}(\Lambda) = \bar{\Lambda}.$$

For $i \in [n]$, we denote by $\mathbf{proj}_{\theta^*,i}$ the homomorphic extension of $\mathbf{proj}_i$ to words over θ, i.e., $\mathbf{proj}_{\theta^*,i} : \theta^* \to \mathbf{proj}_i(\theta)^*$. Note that $\mathbf{proj}_{\theta^*,i}$ is a weak coding. We will also write $\mathbf{proj}_i$ instead of $\mathbf{proj}_{\theta^*,i}$ if no confusion arises.

Thus, e.g., given the vector letters α and β as above, we have that $\mathbf{proj}_{\{\alpha,\beta\}^*,1}(\alpha\beta) = \mathbf{proj}_1(\alpha)\mathbf{proj}_1(\beta) = \Lambda a = a$. Note that $\mathbf{coll}(\alpha\beta) = (\mathbf{proj}_{\{\alpha,\beta\}^*,1}(\alpha\beta), \mathbf{proj}_{\{\alpha,\beta\}^*,2}(\alpha\beta)) = \binom{a}{cb}$.

The following fact is used occasionally. For all languages $K_1,\ldots,K_n$, $\mathbf{coll}_\theta^{-1}(\times_{i=1}^n K_i) = \bigcap_{i=1}^n \mathbf{proj}_{\theta^*,i}{}^{-1}(K_i)$. Using this fact and the function-intersection rule we obtain the following observation.

Observation 1.1.1. For all n-languages M and all languages $K_1,\ldots,K_n$ we have that $\times_{i=1}^n K_i \cap \mathbf{coll}_\theta(M) = \mathbf{coll}_\theta(\bigcap_{i=1}^n \mathbf{proj}_{\theta^*,i}{}^{-1}(K_i) \cap M)$.

For a family $\mathbb{M}$ of languages and an $n \geqslant 1$, we denote by $\mathbb{L}(n\text{-}\mathbb{M})$ the subfamily of n-languages from $\mathbb{M}$ and by $\mathbb{V}(n\text{-}\mathbb{M}) = \{\mathbf{coll}(M) \mid M \in \mathbb{L}(n\text{-}\mathbb{M})\}$ the associated family of vector languages.

1.2. Vector controlled concurrent systems

In this subsection we briefly recall the notion of a Vector Controlled Concurrent System and its vector language as introduced in [15] and studied in [16–18]. Here we give an algebraic definition which is equivalent to the more operational definition from [15].

Definition 1.2.1. A Vector Controlled Concurrent System, or VCCS for short, is a construct $\mathscr{V} = (K_1,\ldots,K_n;M)$, where $n \geqslant 1$, $K_1,\ldots,K_n$ are languages and M is an n-language. The languages $K_1,\ldots,K_n$ are called the component languages of $\mathscr{V}$, and the n-language M is called the control language of $\mathscr{V}$.

The vector language of $\mathscr{V}$, is the vector language $V(\mathscr{V}) = \times_{i=1}^n K_i \cap \mathbf{coll}(M)$.

Thus, a VCCS $\mathscr{V} = (K_1, \ldots, K_n; M)$ models a concurrent system consisting of n sequential subsystems, represented by the n languages $K_1, \ldots, K_n$, that are synchronized by a single control mechanism, represented by the n-language M. The behaviour of $\mathscr{V}$ is represented by its vector language $V(\mathscr{V})$; it consists of those word vectors $(w_1, \ldots, w_n)$ from $K_1 \times \cdots \times K_n$ that have a decomposition into synchronization vectors forming a word from M. Observe that we use the vector language $\mathbf{coll}(M)$ rather than its specification, the language M, to determine the behaviour of the system $\mathscr{V}$.

2. The vector language families

In this section we introduce the various VCCS submodels by discussing the restrictions imposed on component and control languages. This leads to the families of vector languages considered in the rest of this paper. Our approach is the following.

In Section 2.1, first the control mechanisms and their vector languages are discussed. Next we turn to the component languages. In Section 2.2 we discuss the relationships between the families of VCCS vector languages obtained by combining the various types of control languages and component languages. Here, for the sake of reference, we also add some independently defined families of vector languages.

2.1. Families

In all our VCCS submodels we assume that the control languages are defined by some finite state device. Thus, each family of control languages (of dimension n) will be a subset of $\mathbb{L}(n\text{-}\mathbb{R}\text{eg})$, the family of regular n-languages, i.e., regular languages over n-dimensional vector alphabets. The family $\mathbb{L}(n\text{-}\mathbb{R}\text{eg})$ itself is the largest family of n-languages we use for defining n-dimensional control languages. Its associated family of vector languages $\mathbb{V}(n\text{-}\mathbb{R}\text{eg}) = \{\mathbf{coll}(L) \mid L \in \mathbb{L}(n\text{-}\mathbb{R}\text{eg})\}$ is in fact the family of *rational relations* or *rational vector languages* of dimension n (see [18]). In [1] this family (or rather the union $\bigcup_{n \geqslant 1} \mathbb{V}(n\text{-}\mathbb{R}\text{eg})$) occurs as *Rat*, the family of all rational relations.

In [18] it is shown that the family $\mathbb{V}(n\text{-}\mathbb{R}\text{eg})$ coincides with the family of vector languages defined by a generalized version of Individual Token Net Controllers. An *Individual Token Net Controller* (ITNC for short) is a particular type of Petri Net used as a control mechanism with distributed states and transitions. See [15] for a formal definition of ITNCs and see, e.g., [21] for an introduction of Petri Nets – including nets with individual tokens. In fact, Individual Token Net Controllers can be seen as a particular version of so-called *state-machine decomposable nets*. Such nets have occurred in Petri Net theory since [10]. A recent survey of such nets has been given in [5], and in [13] they have been used to give a semantics of COSY. Our approach differs slightly from the nets presented in the literature, because in an ITNC the separate state-machines are represented by individual tokens. Moreover, the transitions are labelled with synchronization vectors.

In [18] the rational vector languages defined by ITNCs are shown to be characterized by a property called *completeness*:

An n-dimensional vector language V, where $n \geqslant 1$, is called complete, if

(1) whenever $v, w \in V$ are such that v and w are independent, then $v \odot w \in V$ as well, and

(2) if $\Lambda \in \mathbf{proj}_i(\mathbb{V})$ for all $i \in [n]$, then $\bar{\Lambda} \in V$ as well.

For a family $\mathbb{V}(n\text{-}\mathbb{M})$ of n-dimensional vector languages, we let $\mathbb{V}(n\text{-c}\mathbb{M}) = \{V \in \mathbb{V}(n\text{-}\mathbb{M}) \mid V \text{ is complete}\}$ be the subfamily of complete vector languages, and we let $\mathbb{L}(n\text{-c}\mathbb{M})$ be the associated family of n-languages, $\mathbb{L}(n\text{-c}\mathbb{M}) = \{L \mid \mathbf{coll}(L) \in \mathbb{V}(n\text{-c}\mathbb{M})\}$.

The first condition of completeness expresses that a concatenation of independent computation vectors is also a computation vector, i.e. is in the vector language. The second condition expresses that, if for each position i there is a computation vector having an empty ith component, then also the empty word vector is a computation vector.

Note that the intersection of two complete vector languages is complete again, and that any cartesian product of languages is a complete vector language. (*)

In addition to completeness we also use Λ-completeness:

An n-dimensional vector language V is called Λ-complete, if condition (2) above is satisfied, i.e., if $\Lambda \in \mathbf{proj}_i(V)$ for all $i \in [n]$, then $\bar{\Lambda} \in V$.

For a family $\mathbb{V}(n\text{-}\mathbb{M})$ of n-dimensional vector languages, we write $\mathbb{V}(n\text{-}\lambda\mathbb{M}) = \{V \in \mathbb{V}(n\text{-}\mathbb{M}) \mid V \text{ is } \Lambda\text{-complete})$ and we let $\mathbb{L}(n\text{-}\lambda\mathbb{M})$ be the associated family of n-languages, that is $\mathbb{L}(n\text{-}\lambda\mathbb{M}) = \{L \mid \mathbf{coll}(L) \in \mathbb{V}(n\text{-}\lambda\mathbb{M})\}$.

Note that, as for completeness, the intersection of two Λ-complete vector languages is Λ-complete again, and that any cartesian product of languages is a Λ-complete vector language. (**)

The property of Λ-completeness is an almost trivial property, because any vector language can be made Λ-complete by adding $\bar{\Lambda}$. It turns out to be technically useful, because it appears as a weak version of prefix-closedness, our next restriction.

Prefix-closed languages often occur as descriptions of the ongoing behaviours of (concurrent) systems. In the context of concurrency, systems without acceptance conditions or final states frequently occur, giving rise to prefix-closed behaviour descriptions (see, e.g., [21] or [14] for examples from the theory of Petri Nets).

Prefixes may also be used as approximations of infinite computations, e.g., through adherences (see, e.g. [6]).

In case of accepting systems prefixes may represent all possible observations of (successful) computations. This approach is followed in, e.g., [2]. Also the theory of COSY may be seen as a representative (see, e.g., [13]), while its original motivation comes from systems without acceptance conditions.

In this paper we do not distinguish between the various uses of prefix-closedness, but simply use it as a restriction on languages in the context of concurrent and sequential systems.

Let $\mathbb{L}(n\text{-p}\mathbb{M}) = \{L \in \mathbb{L}(n\text{-}\mathbb{M}) \mid L \text{ is prefix-closed}\}$, for a family $\mathbb{L}(n\text{-}\mathbb{M})$ of n-languages, and let $\mathbb{V}(n\text{-p}\mathbb{M})$ be its associated family of vector languages, $\mathbb{V}(n\text{-p}\mathbb{M}) = \{\mathbf{coll}(L) \mid L \in \mathbb{L}(n\text{-p}\mathbb{M})\}$.

Since prefix-closedness implies Λ-completeness, we have that $\mathbb{V}(n\text{-p}\lambda\mathbb{M})=\mathbb{V}(n\text{-p}\mathbb{M})$, and $\mathbb{L}(n\text{-p}\lambda\mathbb{M})=\mathbb{L}(n\text{-p}\mathbb{M})$, for any family of vector languages $\mathbb{V}(n\text{-}\mathbb{M})$ and family of n-languages $\mathbb{L}(n\text{-}\mathbb{M})$. Hence, using all combinations of the three types of restrictions on $\mathbb{V}(n\text{-}\mathbb{R}\text{eg})$, we obtain the following five families of vector languages $\mathbb{V}(n\text{-}\mathbb{R}\text{eg})$, $\mathbb{V}(n\text{-}\lambda\mathbb{R}\text{eg})$, $\mathbb{V}(n\text{-c}\mathbb{R}\text{eg})$, $\mathbb{V}(n\text{-p}\mathbb{R}\text{eg})$, and $\mathbb{V}(n\text{-cp}\mathbb{R}\text{eg})$.

In addition to the above five families, we also investigate the family of control languages based upon *monoids*: $\mathbb{L}(n\text{-}\mathbb{M}\text{on})=\{\theta^* \mid \theta$ is an n-dimensional vector alphabet$\}$, and its associated family of vector languages, $\mathbb{V}(n\text{-}\mathbb{M}\text{on})$.

The control mechanism represented by $\mathbb{L}(n\text{-}\mathbb{M}\text{on})$ has been used in [15] in the VCCS submodel of *Vector Synchronized Systems*. Also in [1] systems with this type of monoid control have been studied. In the theory of COSY control is implicitly also of this type, though there it is even more restricted. The control represented by $\mathbb{L}(n\text{-}\mathbb{M}\text{on})$ can be termed *static*: at each moment during the history of such a VCCS the allowed synchronizations, represented by a set of vector letters, is the same. In contrast with this the control mechanisms represented by $\mathbb{V}(n\text{-}\mathbb{R}\text{eg})$ are *dynamic*: here the allowed set of synchronization vectors may change during computations, i.e., with the state of the system.

Note that all languages in $\mathbb{L}(n\text{-}\mathbb{M}\text{on})$ are prefix-closed, and that the vector languages in $\mathbb{V}(n\text{-}\mathbb{M}\text{on})$ are complete and hence Λ-complete as well. Hence, applying the three restrictions of prefix-closedness, Λ-completeness, and completeness does not lead to new families.

Now we turn to families of component languages.

Starting with $\mathbb{A}$ll, the family of all languages, and applying the restrictions of regularity and prefix-closedness, we obtain the families $\mathbb{R}$eg, the family of all regular languages, p$\mathbb{A}$ll, the family of all prefix closed languages, and p$\mathbb{R}$eg, the family of all prefix closed regular languages.

Regularity and prefix-closedness are again considered, because they stand for respectively, finiteness of the underlying state model and ongoing computations. In [15] and [16], VCCSs were studied having component languages without restrictions, i.e., from $\mathbb{A}$ll, and with the regularity restriction, i.e., from $\mathbb{R}$eg. In [1] prefix-closedness is assumed throughout.

For component languages the restrictions of completeness or Λ-completeness are void. These restrictions are only non-trivial in case of vector languages of dimension greater than one.

To the above four families of component languages, we add the family of (finitely generated) monoids, $\mathbb{M}\text{on}=\{\Sigma^* \mid \Sigma$ is an alphabet$\}$. Vector Controlled Concurrent Systems with component languages from $\mathbb{M}$on exhibit the behaviour of their control mechanism. For this reason we include this type of component languages into our investigations.

Combining the families of control languages and the families of component languages introduced above, a considerable number of VCCS models are obtained. In this paper we investigate the resulting families of vector languages of these VCCSs, that is, we investigate the families $\mathbb{V}(\mathbb{K};n\text{-}\mathbb{M})=\{V \mid V$ is the vector language of a VCCS with component languages from $\mathbb{K}$ and a control language from $\mathbb{L}(n\text{-}\mathbb{M})\}$, i.e.

$$\mathbb{V}(\mathbb{K};n\text{-}\mathbb{M}) = \left\{ \left(\bigtimes_{i=1}^{n} K_i \right) \cap \mathbf{coll}(M) \mid K_1,\dots,K_n \in \mathbb{K},\ M \in \mathbb{L}(n\text{-}\mathbb{M}) \right\},$$

where

$\mathbb{K} = \mathbb{M}\text{on}$, $\text{p}\mathbb{R}\text{eg}$, $\mathbb{R}\text{eg}$, $\text{p}\mathbb{A}\text{ll}$, or $\mathbb{A}\text{ll}$, and

$\mathbb{M} = \mathbb{M}\text{on}$, $\text{cp}\mathbb{R}\text{eg}$, $\text{p}\mathbb{R}\text{eg}$, $\text{c}\mathbb{R}\text{eg}$, $\lambda\mathbb{R}\text{eg}$, or $\mathbb{R}\text{eg}$.

2.2. *Inclusions and equalities*

The families of VCCS vector languages introduced in the previous subsection can be directly related using the following observation.

Let n denote an arbitrary but fixed positive integer.

Observation 2.2.1. $\mathbb{V}(\mathbb{K}_1;n\text{-}\mathbb{M}_1) \subseteq \mathbb{V}(\mathbb{K}_2;n\text{-}\mathbb{M}_2)$ whenever $\mathbb{K}_1 \subseteq \mathbb{K}_2$ and $\mathbb{V}(n\text{-}\mathbb{M}_1) \subseteq \mathbb{V}(n\text{-}\mathbb{M}_2)$.

Since, for component languages, by definition,
$\mathbb{M}\text{on} \subseteq \text{p}\mathbb{R}\text{eg} \subseteq \mathbb{R}\text{eg} \subseteq \mathbb{A}\text{ll}$, and
$\text{p}\mathbb{R}\text{eg} \subseteq \text{p}\mathbb{A}\text{ll} \subseteq \mathbb{A}\text{ll}$,
and since similarly, for control languages,

$$\mathbb{V}(n\text{-}\mathbb{M}\text{on}) \subseteq \mathbb{V}(n\text{-cp}\mathbb{R}\text{eg}) \subseteq \mathbb{V}(n\text{-p}\mathbb{R}\text{eg}) \subseteq \mathbb{V}(n\text{-}\lambda\mathbb{R}\text{eg}) \subseteq \mathbb{V}(n\text{-}\mathbb{R}\text{eg}), \text{ and}$$

$$\mathbb{V}(n\text{-cp}\mathbb{R}\text{eg}) \subseteq \mathbb{V}(n\text{-c}\mathbb{R}\text{eg}) \subseteq \mathbb{V}(n\text{-}\lambda\mathbb{R}\text{eg}),$$

this yields an initial inclusion diagram, Fig. 1, given in the form of a matrix. A particular family $\mathbb{V}(\mathbb{K};n\text{-}\mathbb{M})$ is the entry in the row marked by $\mathbb{K}$ to the left and the column marked by $\mathbb{M}$ at the bottom. In the matrix each entry $\mathbb{V}(\mathbb{K};n\text{-}\mathbb{M})$ is represented by a dot and an arrow from a family $\mathbb{V}_1$ to a family $\mathbb{V}_2$ denotes that $\mathbb{V}_1 \subseteq \mathbb{V}_2$.

Not all inclusions in the diagram are strict as will become clear below.

In addition to the VCCS vector languages introduced in Section 2.1, also a number of other families of vector languages is included in our study. They are included because they turn out to characterize a number of VCCS vector language families or arise as the result of application of one of the operations and as such form reference points for these families. Moreover, they are defined directly and thus are simpler than VCCS vector languages and easier to handle technically.

They are, first of all, the families $\mathbb{V}(n\text{-}\mathbb{A}\text{ll})$, $\mathbb{V}(n\text{-}\mathbb{R}\text{eg})$, and $\mathbb{V}(n\text{-}\mathbb{M}\text{on})$, where $\mathbb{V}(n\text{-}\mathbb{A}\text{ll})$ is the family of all n-dimensional vector languages. By adding the restrictions of Λ-completeness, completeness, and prefix-closedness, we arrive at the following

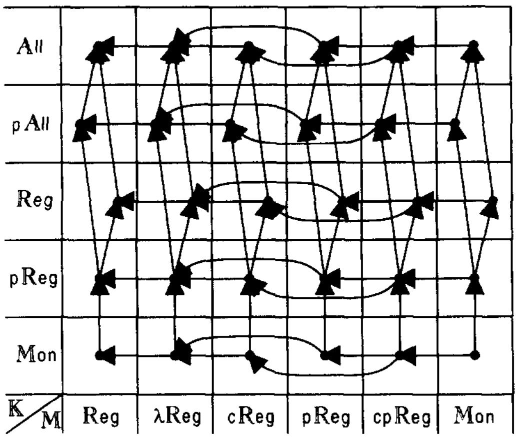

Fig. 1. (Non-strict) inclusion diagram of the families of the form $\mathbb{V}(\mathbb{K}; n\text{-}\mathbb{M})$ for $n \geqslant 1$.

families:

$$\mathbb{V}(n\text{-}\mathbb{A}\text{ll}),\ \mathbb{V}(n\text{-}\lambda\mathbb{A}\text{ll}),\ \mathbb{V}(n\text{-c}\mathbb{A}\text{ll}),\ \mathbb{V}(n\text{-p}\mathbb{A}\text{ll}),\ \mathbb{V}(n\text{-cp}\mathbb{A}\text{ll}),$$

$$\mathbb{V}(n\text{-}\mathbb{R}\text{eg}),\ \mathbb{V}(n\text{-}\lambda\mathbb{R}\text{eg}),\ \mathbb{V}(n\text{-c}\mathbb{R}\text{eg}),\ \mathbb{V}(n\text{-p}\mathbb{R}\text{eg}),\ \mathbb{V}(n\text{-cp}\mathbb{R}\text{eg}),\ \text{and}\ \mathbb{V}(n\text{-}\mathbb{M}\text{on}).$$

The last six families have already been introduced in Section 2.1, as families of control vector languages. The first five are new. They can be seen as counterparts of the five regular families. They capture precisely the combinations of restrictions of Λ-completeness, completeness, and prefix-closedness.

The following lemma gives a number of inclusions and equalities that relate these families to Fig. 1

Lemma 2.2.2. (1) $\mathbb{V}(n\text{-cp}\mathbb{A}\text{ll}) \subseteq \mathbb{V}(n\text{-c}\mathbb{A}\text{ll}) \subseteq \mathbb{V}(n\text{-}\lambda\mathbb{A}\text{ll}) \subseteq \mathbb{V}(n\text{-}\mathbb{A}\text{ll})$, *and* $\mathbb{V}(n\text{-cp}\mathbb{A}\text{ll}) \subseteq \mathbb{V}(n\text{-p}\mathbb{A}\text{ll}) \subseteq \mathbb{V}(n\text{-}\lambda\mathbb{A}\text{ll})$.

(2) $\mathbb{V}(\mathbb{A}\text{ll}; n\text{-}\mathbb{R}\text{eg}) \subseteq \mathbb{V}(n\text{-}\mathbb{A}\text{ll})$, $\mathbb{V}(\mathbb{A}\text{ll}; n\text{-}\lambda\mathbb{R}\text{eg}) \subseteq \mathbb{V}(n\text{-}\lambda\mathbb{A}\text{ll})$, $\mathbb{V}(\mathbb{A}\text{ll}; n\text{-c}\mathbb{R}\text{eg}) \subseteq \mathbb{V}(n\text{-c}\mathbb{A}\text{ll})$, $\mathbb{V}(\text{p}\mathbb{A}\text{ll}; n\text{-p}\mathbb{R}\text{eg}) \subseteq \mathbb{V}(n\text{-p}\mathbb{A}\text{ll})$, *and* $\mathbb{V}(\text{p}\mathbb{A}\text{ll}; n\text{-cp}\mathbb{R}\text{eg}) \subseteq \mathbb{V}(n\text{-cp}\mathbb{A}\text{ll})$.

(3) $\mathbb{V}(\mathbb{R}\text{eg}; n\text{-}\mathbb{M}) = \mathbb{V}(\text{p}\mathbb{R}\text{eg}; n\text{-}\mathbb{M}) = \mathbb{V}(\mathbb{M}\text{on}; n\text{-}\mathbb{M}) = \mathbb{V}(n\text{-}\mathbb{M})$, for $\mathbb{M} = \mathbb{R}\text{eg}$, $\lambda\mathbb{R}\text{eg}$, $\text{c}\mathbb{R}\text{eg}$, $\mathbb{V}(\text{p}\mathbb{R}\text{eg}; n\text{-}\mathbb{M}) = \mathbb{V}(\mathbb{M}\text{on}; n\text{-}\mathbb{M}) = \mathbb{V}(n\text{-}\mathbb{M})$, for $\mathbb{M} = \text{p}\mathbb{R}\text{eg}$, $\text{cp}\mathbb{R}\text{eg}$, *and* $\mathbb{V}(\mathbb{M}\text{on}; n\text{-}\mathbb{M}\text{on}) = \mathbb{V}(n\text{-}\mathbb{M}\text{on})$.

Proof.

(1) Obvious.

(2) The first inclusion is obvious. The other inclusions follow from Observation 1.1.1, the preservation of prefix-closedness by inverse weak codings and intersections, and the basic properties (*) and (**) of completeness and Λ-completeness (see Section 2.1), stating that the intersection of two (Λ-)complete vector languages is again (Λ-)complete and that a cartesian product of languages is complete and hence Λ-complete.

(3) The inclusions $\supseteq$ are obvious. So it remains to prove that $\mathbb{V}(\mathbb{R}\text{eg}; n\text{-}\mathbb{M}) \subseteq \mathbb{V}(n\text{-}\mathbb{M})$ for $\mathbb{M} = \mathbb{R}\text{eg}$, $\lambda\mathbb{R}\text{eg}$, $\text{c}\mathbb{R}\text{eg}$, $\mathbb{V}(\text{p}\mathbb{R}\text{eg}; n\text{-}\mathbb{M}) \subseteq \mathbb{V}(n\text{-}\mathbb{M})$ for $\mathbb{M} = \text{p}\mathbb{R}\text{eg}$, $\text{cp}\mathbb{R}\text{eg}$, and $\mathbb{V}(\mathbb{M}\text{on}; n\text{-}\mathbb{M}) \subseteq \mathbb{V}(n\text{-}\mathbb{M})$ for $\mathbb{M} = \mathbb{M}\text{on}$. As in (2) above these inclusions follow from Observation 1.1.1, the preservation properties of prefix-closedness, and the basic properties (*) and (**) from Section 2.1. In addition, the preservation of regularity by inverse weak codings and intersection is used. □

Combining the results proved in Lemma 2.2.2 we obtain an extended inclusion diagram, Fig. 2. In this diagram the families introduced in Section 2.1 are depicted together with the five new families $\mathbb{V}(n\text{-}\mathbb{M})$ for $\mathbb{M} = \mathbb{A}\text{ll}$, $\lambda\mathbb{A}\text{ll}$, $\text{c}\mathbb{A}\text{ll}$, $\text{p}\mathbb{A}\text{ll}$, $\text{cp}\mathbb{A}\text{ll}$. Again the families of the form $\mathbb{V}(\mathbb{K}; n\text{-}\mathbb{M})$ are depicted in a matrix form, where the $\mathbb{K}$s are denoted at the left of the matrix and the $\mathbb{M}$s are depicted at the bottom of the matrix. Each family of the form $\mathbb{V}(\mathbb{K}; n\text{-}\mathbb{M})$ is represented by a dot (as in Fig. 1).

The families proved equal in Lemma 2.2.2(3) are adjacent and their corresponding dots have been merged into one another, with dashed lines showing the equalities between different entries.

Each family of the form $\mathbb{V}(n\text{-}\mathbb{M})$ is denoted by a small circle. They have been marked by their corresponding $\mathbb{M}$.

Since the families $\mathbb{V}(n\text{-}\mathbb{M})$, where $\mathbb{M} = \mathbb{R}\text{eg}$, $\lambda\mathbb{R}\text{eg}$, $\text{c}\mathbb{R}\text{eg}$, $\text{p}\mathbb{R}\text{eg}$, $\text{cp}\mathbb{R}\text{eg}$, or $\mathbb{M}\text{on}$, coincide with some of the families of the form $\mathbb{V}(\mathbb{K}; n\text{-}\mathbb{M})$ circled dots have been used here.

The arrows again denote inclusion, but as will be shown in the next section these inclusions are strict now (at least for $n \geqslant 2$).

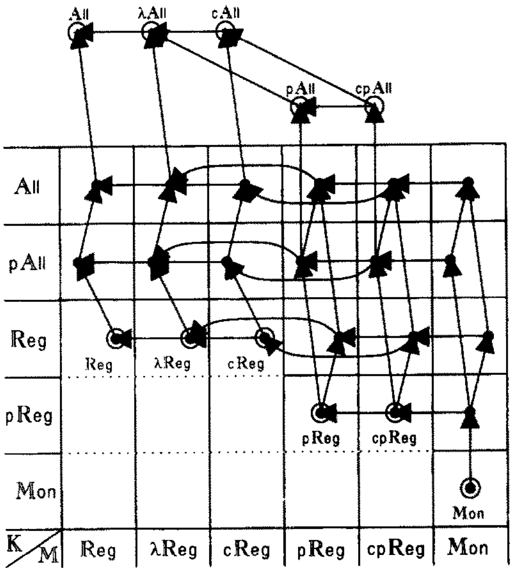

Fig. 2. Inclusion diagram of the families of the form $\mathbb{V}(\mathbb{K}; n\text{-}\mathbb{M})$ and $\mathbb{V}(n\text{-}\mathbb{M})$ for $n \geqslant 2$.

3. The inclusion diagram

Now we prove that the inclusion diagram (Fig. 2) is complete, i.e, any inclusion holding between any of the families presented can be derived from the diagram as a (directed) path between these families. This is done, for the case $n\geqslant 2$, in Section 3.1 by giving particular example vector languages contained in one family but not another. The family containing such an example vector language is chosen as small as possible, while the family that does not contain the example vector language is chosen as large as possible. In this way we establish with a small number of example vector languages all non-inclusions needed, and thus the strictness of the inclusions in the diagram and the incomparability of the families not connected by a directed path.

As shown in Section 3.2, however, for $n = 1$ additional equalities and inclusions hold, because, e.g., completeness and Λ-completeness are trivial for one-dimensional vector languages. The resulting complete diagram is presented in that subsection.

3.1. The more-dimensional case

In the proof of the following lemma we present the first set of example vector languages. The resulting non-inclusions also hold for the case $n = 1$.

Lemma 3.1.1. (1) $\mathbb{V}(\mathrm{p}\mathbb{R}\mathrm{eg}; n\text{-}\mathbb{M}\mathrm{on}) - \mathbb{V}(n\text{-}\mathbb{M}\mathrm{on}) \neq \emptyset$ *for all* $n\geqslant 1$.
(2) $\mathbb{V}(\mathbb{R}\mathrm{eg}; n\text{-}\mathbb{M}\mathrm{on}) - \mathbb{V}(n\text{-}\mathrm{p}\mathbb{A}\mathrm{ll}) \neq \emptyset$ *for all* $n\geqslant 1$.
(3) $\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathbb{M}\mathrm{on}) - \mathbb{V}(n\text{-}\mathbb{R}\mathrm{eg}) \neq \emptyset$ *for all* $n\geqslant 1$.
(4) $\mathbb{V}(\mathbb{A}\mathrm{ll}; n\text{-}\mathbb{M}\mathrm{on}) - \mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathbb{R}\mathrm{eg}) \neq \emptyset$ *for all* $n\geqslant 1$.

Proof. Each of the four non-inclusions can be proved using 1-dimensional vector languages. For $n > 1$, the lemma follows by extending these vector languages with Λ-components, i.e. replacing 1-dimensional vectors of the form (w) with n-dimensional vectors of the form $(w, \Lambda, \ldots, \Lambda)$.

The first 3 non-inclusions are easy. The fourth needs a more careful consideration.
(1) $V_1 = \{\bar{\Lambda}, (a)\} \in \mathbb{V}(\mathrm{p}\mathbb{R}\mathrm{eg}; 1\text{-}\mathbb{M}\mathrm{on}) - \mathbb{V}(1\text{-}\mathbb{M}\mathrm{on})$.
(2) $V_2 = \{(ab)\} \in \mathbb{V}(\mathbb{R}\mathrm{eg}; 1\text{-}\mathbb{M}\mathrm{on}) - \mathbb{V}(1\text{-}\mathrm{p}\mathbb{A}\mathrm{ll})$.
(3) $V_3 = \{(a^k b^l) \mid k\geqslant l\geqslant 0\} \in \mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; 1\text{-}\mathbb{M}\mathrm{on}) - \mathbb{V}(1\text{-}\mathbb{R}\mathrm{eg})$.
(4) $V_4 = \{(a^k b^k) \mid k\geqslant 0\} \in \mathbb{V}(\mathbb{A}\mathrm{ll}; 1\text{-}\mathbb{M}\mathrm{on}) - \mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; 1\text{-}\mathbb{R}\mathrm{eg})$.
That $V_4 \in \mathbb{V}(\mathbb{A}\mathrm{ll}; 1\text{-}\mathbb{M}\mathrm{on})$ is easy to see. That $\mathbb{V}_4 \notin \mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; 1\text{-}\mathbb{R}\mathrm{eg})$ is shown in the following way. If $V_4 = X(K \cap R)$ where K is a prefix-closed language and R is a regular language, then, by the pumping lemma for regular languages, $a^k b^k \in R$ for a large enough k, implies that $a^m b^k \in R$ for an $m > k$. Since $a^m b^k \notin V_4$, we then have that $a^m b^k \notin K$, and since K is prefix-closed also $a^m b^m \notin K$. This is in contradiction with $a^m b^m \in V_4$. □

In the following lemma we present the remaining non-inclusions. These only hold for the case $n\geqslant 2$.

Lemma 3.1.2. (1) $\mathbb{V}(n\text{-cp}\mathbb{R}\text{eg}) - \mathbb{V}(\mathbb{A}\text{ll}; n\text{-}\mathbb{M}\text{on}) \neq \emptyset$ *for all* $n \geqslant 2$.
(2) $\mathbb{V}(n\text{-p}\mathbb{R}\text{eg}) - \mathbb{V}(n\text{-c}\mathbb{A}\text{ll}) \neq \emptyset$ *for all* $n \geqslant 2$.
(3) $\mathbb{V}(n\text{-c}\mathbb{R}\text{eg}) - \mathbb{V}(\mathbb{A}\text{ll}; n\text{-p}\mathbb{R}\text{eg}) \neq \emptyset$ *for all* $n \geqslant 2$.
(4) $\mathbb{V}(n\text{-}\mathbb{R}\text{eg}) - \mathbb{V}(n\text{-}\lambda\mathbb{A}\text{ll}) \neq \emptyset$ *for all* $n \geqslant 2$.
(5) $\mathbb{V}(n\text{-cp}\mathbb{A}\text{ll}) - \mathbb{V}(\mathbb{A}\text{ll}; n\text{-}\mathbb{R}\text{eg}) \neq \emptyset$ *for all* $n \geqslant 2$.

Proof. We use 2-dimensional vector languages to prove the lemma. For the higher dimensions these vector languages are again extended with Λ-components.

(1) $V_5 = \binom{a}{b}^{\circledast} \cup \binom{a}{c}^{\circledast} \cup \{\binom{\Lambda}{b}, \binom{\Lambda}{c}\}^{\circledast} \in \mathbb{V}(2\text{-cp}\mathbb{R}\text{eg}) - \mathbb{V}(\mathbb{A}\text{ll}; 2\text{-}\mathbb{M}\text{on})$.

It is easy to see that V_5 is rational, prefix-closed and complete, hence $V_5 \in \mathbb{V}(2\text{-cp}\mathbb{R}\text{eg})$.

Now, suppose that $V_5 = (K_1 \times K_2) \cap W$, for some $K_1, K_2 \in \mathbb{A}\text{ll}$ and a $W \in \mathbb{V}(2\text{-}\mathbb{M}\text{on})$. Then $\binom{a}{b} \odot \binom{a}{c} \in W$, because $\binom{a}{b}, \binom{a}{c} \in V_5 \subseteq W$. Furthermore, $aa \in \mathbf{proj}_1(V_5) \subseteq K_1$ and $bc \in \mathbf{proj}_2(V_5) \subseteq K_2$. This implies that $\binom{a}{b} \odot \binom{a}{c} = \binom{aa}{bc} \in V_5$, a contradiction. Thus it follows that $V_5 \notin \mathbb{V}(\mathbb{A}\text{ll}; 2\text{-}\mathbb{M}\text{on})$.

(2) $V_6 = \{\bar{\Lambda}, \binom{a}{\Lambda}, \binom{\Lambda}{b}\} \in \mathbb{V}(2\text{-p}\mathbb{R}\text{eg}) - \mathbb{V}(2\text{-c}\mathbb{A}\text{ll})$. Follows directly from the definitions, because clearly V_6 is prefix closed and rational, but not complete.

(3) $V_7 = \{\binom{aa}{cc}, \binom{a}{d}, \binom{b}{c}\} = \mathbb{V}(2\text{-c}\mathbb{R}\text{eg}) - \mathbb{V}(\mathbb{A}\text{ll}; 2\text{-p}\mathbb{R}\text{eg})$.

It is easy to see that $V_7 \in \mathbb{V}(2\text{-c}\mathbb{R}\text{eg})$.

Now, suppose that $V_7 = (K_1 \times K_2) \cap \mathbf{coll}(M)$, for some $K_1, K_2 \in \mathbb{A}\text{ll}$ and an $M \in \mathbb{L}(2\text{-p}\mathbb{R}\text{eg})$. Since M is prefix closed and $\binom{aa}{cc} \in \mathbf{coll}(M)$, it follows that $\binom{a}{cc} \in \mathbf{coll}(M), \binom{aa}{c} \in \mathbf{coll}(M)$, or $\binom{a}{c} \in \mathbf{coll}(M)$. Since furthermore $\{a, aa\} \subseteq K_1$ and $\{c, cc\} \subseteq K_2$, at least one of these three word vectors must be in V_7 as well, which is a contradiction. Thus, it follows that $V_7 \notin \mathbb{V}(\mathbb{A}\text{ll}; 2\text{-p}\mathbb{R}\text{eg})$.

(4) $V_8 = \{\binom{a}{\Lambda}, \binom{\Lambda}{b}\} \in \mathbb{V}(2\text{-}\mathbb{R}\text{eg}) - \mathbb{V}(2\text{-}\lambda\mathbb{A}\text{ll})$. Follows directly from the definitions, because clearly V_8 is rational and not Λ-complete.

(5) $V_9 = \{\binom{a^{h+k}}{c^h d^k} \mid 0 \leqslant k \leqslant h\} \cup \{\binom{e^p}{w} \mid p \geqslant 1, w \in \{c,d\}^*\} \in \mathbb{V}(2\text{-cp}\mathbb{A}\text{ll}) - \mathbb{V}(\mathbb{A}\text{ll}; 2\text{-}\mathbb{R}\text{eg})$.

It is easy to see that $V_9 \in \mathbb{V}(2\text{-cp}\mathbb{A}\text{ll})$, because in the first place $V_9 = \mathbf{coll}(L)$, where $L = \{\binom{a}{c}^h \binom{a}{d}^k \mid 0 \leqslant k \leqslant h\} \cup \{\binom{e}{\Lambda}\}\{\binom{e}{\Lambda}\}^*\{\binom{\Lambda}{c}, \binom{\Lambda}{d}\}^*$ is prefix-closed, and in the second place V_9 is complete – the only word vector having an empty first component is $\bar{\Lambda}$.

Suppose that $V_9 = (K_1 \times K_2) \cap \mathbf{coll}(M)$, for some $K_1, K_2 \in \mathbb{A}\text{ll}$ and an $M \in \mathbb{L}(2\text{-}\mathbb{R}\text{eg})$. Then $a^* \cup e^* \subseteq K_1$ and $\{c,d\}^* \subseteq K_2$. Set $V' = (\{a\}^* \times \{c,d\}^*) \cap V_9$. Then $V' = ((K_1 \cap \{a\}^*) \times (K_2 \cap \{c,d\}^*)) \cap \mathbf{coll}(M) = (\{a\}^* \times \{c,d\}^*) \cap \mathbf{coll}(M) \in \mathbb{V}(\mathbb{M}\text{on}; 2\text{-}\mathbb{R}\text{eg})$. By Lemma 2.2.2 (3) $\mathbb{V}(\mathbb{M}\text{on}; 2\text{-}\mathbb{R}\text{eg}) = \mathbb{V}(2\text{-}\mathbb{R}\text{eg})$ and hence there exists a regular M' over a vector alphabet θ, such that $\mathbf{coll}(M') = V'$. Then $\mathbf{proj}_{\theta^*,2}(M')$ is regular, because $\mathbf{proj}_{\theta^*,2}$ is a homomorphism. A contradiction, because $\mathbf{proj}_{\theta^*,2}(M') = \mathbf{proj}_2(\mathbf{coll}(M')) = \{c^h d^k \mid 0 \leqslant k \leqslant h\}$ which clearly is *not* regular. Thus it follows that $V_9 \notin \mathbb{V}(\mathbb{A}\text{ll}; 2\text{-}\mathbb{R}\text{eg})$. □

The following lemma shows that the 9 non-inclusion results of Lemma 3.1.1 and Lemma 3.1.2 are sufficient for proving the completeness of Fig. 2 (as stated in Corollary 3.1.4). Moreover, it shows how this completeness can be proved: for each pair of families for which non-inclusion is to be shown, an auxiliary non-inclusion result given in Lemma 3.1.1 or Lemma 3.1.2 can be found. The straightforward checking of each pair of families, thus amounts to a simple inspection of Fig. 2, but due to the large number of pairs this is a tedious process. The proof is therefore left to the reader.

Lemma 3.1.3. *Let $\mathbb{V}_1$ and $\mathbb{V}_2$ be two families occurring in Fig. 2. Assume $n \geqslant 2$. If there is no directed path from $\mathbb{V}_1$ to $\mathbb{V}_2$ in the diagram, then there exist families $\mathbb{V}_1' \subseteq \mathbb{V}_1$ and $\mathbb{V}_2' \supseteq \mathbb{V}_2$, such that $\mathbb{V}_1' - \mathbb{V}_2' \neq \emptyset$ has been proved in Lemma* 3.1.1 *or in Lemma* 3.1.2.

Hence, there is a path from a family $\mathbb{V}_1$ to a family $\mathbb{V}_2$ *if and only if* $\mathbb{V}_1 \subseteq \mathbb{V}_2$. In other words:

Corollary 3.1.4. *Fig.* 2 *is complete, for $n \geqslant 2$.*

3.2. The one-dimensional case

In the previous subsection some of the non-inclusions were shown to hold for all $n \geqslant 1$, whereas others were shown to hold only if $n \geqslant 2$. The latter non-inclusion results cannot be extended to the case $n = 1$, as is shown next.

Several families that are distinct for $n \geqslant 2$, coincide for $n = 1$.

First of all, every 1-dimensional vector language is complete and hence also Λ-complete. Thus, for families of 1-dimensional vector languages completeness and Λ-completeness form no restriction. Hence, e.g., $\mathbb{V}(1\text{-c}\mathbb{R}\text{eg}) = \mathbb{V}(1\text{-}\lambda\mathbb{R}\text{eg}) = \mathbb{V}(1\text{-}\mathbb{R}\text{eg})$ and $\mathbb{V}(\mathbb{A}\text{ll}; 1\text{-c}\mathbb{R}\text{eg}) = \mathbb{V}(\mathbb{A}\text{ll}; 1\text{-}\lambda\mathbb{R}\text{eg}) = \mathbb{V}(\mathbb{A}\text{ll}; 1\text{-}\mathbb{R}\text{eg})$.

Secondly, we observe that the vector languages of one-dimensional VCCSs are nothing but the vector counterparts of intersections of ordinary languages. This means that we can use the intersection closure results of ordinary languages.

Lemma 3.2.1. *Let $\mathbb{K}$ and $\mathbb{M}$ be such that $\mathbb{V}(\mathbb{K}; n\text{-}\mathbb{M})$ is an entry in Fig. 2. Then*
- *If $\mathbb{L}(1\text{-}\mathbb{K}) \subseteq \mathbb{L}(1\text{-}\mathbb{M})$, then $\mathbb{V}(\mathbb{K}; 1\text{-}\mathbb{M}) = \mathbb{V}(1\text{-}\mathbb{M})$.*
- *If $\mathbb{L}(1\text{-}\mathbb{M}) \subseteq \mathbb{L}(1\text{-}\mathbb{K})$, then $\mathbb{V}(\mathbb{K}; 1\text{-}\mathbb{M}) = \mathbb{V}(1\text{-}\mathbb{K})$.*

Proof. The statement follows directly from the definition of $\mathbb{V}(\mathbb{K}; 1\text{-}\mathbb{M}) = \{XK \cap \mathbf{coll}(M) \mid K \in \mathbb{K},\ M \in \mathbb{L}(1\text{-}\mathbb{M})\}$, since each of the families $\mathbb{K}$ and each of the families $\mathbb{L}(1\text{-}\mathbb{M})$ is closed under intersection and contains $\mathbb{M}\text{on}$ or $\mathbb{L}(1\text{-}\mathbb{M}\text{on})$, respectively. □

As a consequence, we have, e.g., $\mathbb{V}(\mathbb{A}\text{ll}; 1\text{-p}\mathbb{R}\text{eg}) = \mathbb{V}(1\text{-}\mathbb{A}\text{ll})$, and $\mathbb{V}(\text{p}\mathbb{R}\text{eg}; 1\text{-}\mathbb{R}\text{eg}) = \mathbb{V}(1\text{-}\mathbb{R}\text{eg})$.

By the above observations, it follows that almost all families of the form $\mathbb{V}(\mathbb{K}; 1\text{-}\mathbb{M})$ coincide with one of the families of the form $\mathbb{V}(1\text{-}\mathbb{M})$, where $\mathbb{M} = \mathbb{A}\text{ll}$, $\text{p}\mathbb{A}\text{ll}$, $\mathbb{R}\text{eg}$, $\text{p}\mathbb{R}\text{eg}$, or $\mathbb{M}\text{on}$. The only exception is the family $\mathbb{V}(\text{p}\mathbb{A}\text{ll}; 1\text{-}\mathbb{R}\text{eg})$.

Summarizing the above, we obtain Fig. 3, a simplified inclusion diagram for the case $n = 1$. We use the drawing conventions of Fig. 2. Thus, again the families $\mathbb{V}(\mathbb{K}; 1\text{-}\mathbb{M})$

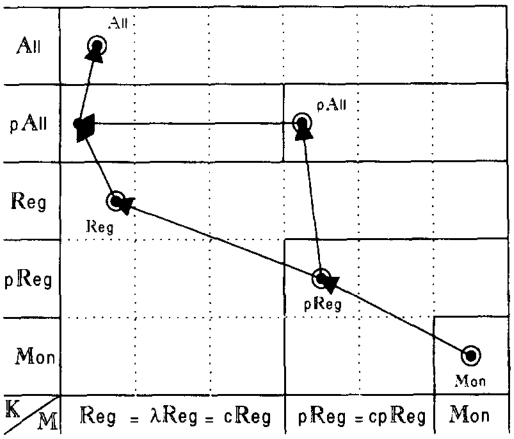

Fig. 3. Inclusion diagram of the families of the form $\mathbb{V}(\mathbb{K}; 1\text{-}\mathbb{M})$ and $\mathbb{V}(1\text{-}\mathbb{M})$.

are represented as entries in the row marked by $\mathbb{K}$ and the column marked by $\mathbb{M}$, while dashed lines between adjacent entries show that the corresponding families coincide. For each group of coinciding families only a single dot is drawn.

From Lemma 3.1.1, we obtain:

Corollary 3.2.2. *Fig. 3 is complete.*

4. The operations

Having completed the direct comparison of the VCCS submodels, we now turn to an indirect approach. Rather than considering inclusions of families of vector languages, we use operations as a means for comparison. That is, we look at relations of the form $\mathbb{V} = \mathbb{O}(\mathbb{V}')$, where $\mathbb{O}$ is a class of operations and $\mathbb{O}(\mathbb{V}') = \{O(V) \mid O \in \mathbb{O},\ V \in \mathbb{V}',\ V$ in the domain of $O\}$.

Note that the relation obtained between families using an operation is rather strong. Not only is it proved that the larger family can be obtained from the smaller one with the help of the operation, i.e. $\mathbb{V} \subseteq \mathbb{O}(\mathbb{V}')$, but also the converse is shown: the combination does not *exceed* the larger family, i.e. $\mathbb{O}(\mathbb{V}') \subseteq \mathbb{V}$.

The use of operations for relating families of languages is well established in the theory of formal languages. A classical example is the representation of the family of recursively enumerable languages: each recursively enumerable language can be obtained from a context-sensitive language using a weak coding. There is a range of representation results, where a family of languages is shown to be generated from a single generator language using one or more operations like homomorphisms, inverse homomorphisms, intersection with regular languages, etc. Famous results for the family of context-free languages are the theorems of Chomsky–Schützenberger [8], and Greibach [9]. In [11] a number of representation results have been collected concerning homomorphisms and rational transductions. Operations on families of languages have also been studied in the framework of *Abstract Families of Languages*, see, e.g. [4].

Here we consider the families of vector languages from Sections 2 and 3 and we focus on the operations of *multi coding*, *hiding*, and *multi weak coding*. These are vector operations and as such well suited for relating families of vector languages. Moreover, they are relatively simple and, as we have seen from [16, 1] they can be used to relate at least some of the VCCS families of vector languages leading to representations of one family in terms of a smaller one.

In this section we formally introduce these three operations and provide some more background and motivation for their use. In the next section we consider each of these operations in turn and investigate its effects.

A *multi coding* (*of dimension* n) is a mapping $\Phi = \varphi_1 \times \cdots \times \varphi_n$, where $\varphi_1, \ldots, \varphi_n$ are codings. Thus, if $\varphi_i : \Sigma_i^* \to \Delta_i^*$ for alphabets Σ_i and Δ_i, $i = 1, \ldots, n$, then $\Phi : \Sigma_1^* \times \cdots \times \Sigma_n^* \to \Delta_1^* \times \cdots \times \Delta_n^*$ is defined by $\Phi((w_1, \ldots, w_n)) = (\varphi_1(w_1), \ldots, \varphi_n(w_n))$.

As an illustration let $\varphi_1(a) = b$, $\varphi_1(b) = b$, $\varphi_2(a) = d$, and $\varphi_2(b) = a$, then $\Phi = \varphi_1 \times \varphi_2$ satisfies $\Phi(\binom{aba}{ba}) = \binom{bbb}{ad}$.

Note that multi codings are *not* special instances of the *multi-morphisms* of Nivat (see, e.g., [20]), as the latter are mappings from languages to vector languages: $\langle \varphi_1, \ldots, \varphi_n \rangle : \Sigma^* \to \Delta_1^* \times \cdots \times \Delta_n^*$ is a multi-morphism according to Nivat, if $\varphi_i : \Sigma^* \to \Delta_i^*$ is a homomorphism for all $i \in [n]$.

Thus, e.g., with φ_1 and φ_2 as above, the multi-morphism $\langle \varphi_1, \varphi_2 \rangle$ satisfies $\langle \varphi_1, \varphi_2 \rangle (aba) = \binom{bbb}{dad}$.

In many respects multi codings resemble ordinary codings. They map (vector) letters onto (vector) letters. They are, however, more refined in the sense that they act on the components of vector letters and not simply on the vector letters themselves. In general, an (ordinary) coding – defined on a vector alphabet – is not a multi coding, because, e.g., they can arbitrarily change the dimension of vector letters. Thus, e.g. a coding ψ may satisfy $\psi(\binom{a}{b}) = [a], \psi(\binom{a}{\Lambda}) = \begin{pmatrix} a \\ \Lambda \\ b \end{pmatrix}$, and $\psi(\binom{\Lambda}{b}) = \binom{a}{\Lambda}$.

The families of vector languages (and their associated n-languages) studied in this paper are closed under *multi injective codings*, i.e. products of injective codings. In general, families of n-languages are *not* closed under general injective codings, as those may change dimensions. A family like $\mathbb{L}(n\text{-c}\mathbb{R}\text{eg})$ is not even closed under dimension-preserving injective codings, because such codings do not necessarily preserve the Λ-structure of word vectors. Note that multi codings *do* preserve Λ-structure.

Note that the multi coding Φ defined above is *not* a multi injective coding, because φ_1 is not injective. However, Φ restricted to the domain $\{\binom{a}{b}, \binom{b}{a}, \binom{\Lambda}{a}\}$ is injective.

Multi codings may be seen as labellings leading to an identification of originally different symbols (actions). Note, however, that the labelling is applied to the behaviour of a system, i.e. its vector language, and *not* to the underlying model itself, i.e. to the specification by component languages and control language. Since the component languages and the control languages are ordinary languages, the latter way of applying the labelling would make it a simple matter of ordinary formal language theory to determine the type of the resulting VCCS.

In [16] multi codings have been used to relate the families $\mathbb{V}(\mathbb{R}\mathrm{eg}; n\text{-}\mathbb{M}\mathrm{on})$ and $\mathbb{V}(\mathbb{R}\mathrm{eg}; n\text{-c}\mathbb{R}\mathrm{eg})$.

The class of all n-dimensional multi codings will be denoted n-$\mathbb{MC}$od.

For a family $\mathbb{V}$ of n-dimensional vector languages, we let $n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}) = \{\Phi(V) \mid \Phi \in n\text{-}\mathbb{MC}\mathrm{od},\ V \in \mathbb{V},\ V \text{ in the domain of } \Phi\}$.

It is easy to see that n-$\mathbb{MC}$od is an operator (on families of vector languages) that is

monotonic: $\mathbb{V} \subseteq \mathbb{V}'$ implies $n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}) \subseteq n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}')$,

extensive: $\mathbb{V} \subseteq n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V})$, and

idempotent: $n\text{-}\mathbb{MC}\mathrm{od}(n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V})) = n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V})$,

where $\mathbb{V}$ and $\mathbb{V}'$ are families of n-dimensional vector languages. Since it is extensive and idempotent, it is a *closure* operator.

By its idempotency, a repeated application of multi codings has the same effect as a single application: the family has been closed with respect to the operator. Families that are not enlarged by a closure operator are already closed; such families turn out to be important in our investigations.

A *hiding* applied to a word vector removes the last component. Thus, it is a mapping from word vectors to word vectors. In fact, it is a particular type of projection. Denoting hiding by **hid**, we have $\mathbf{hid}((w_1, \ldots, w_{n+1})) = (w_1, \ldots, w_n)$.

If $\mathbb{V}$ is a family of vector languages, then $\mathbf{hid}(\mathbb{V}) = \{\mathbf{hid}(V) \mid V \in \mathbb{V},\ V \text{ has dimension at least } 2\}$. Note that we consider here families of, possibly, mixed dimensions and that **hid** is not defined for 1-dimensional vector languages.

Hiding is an operator that is monotonic. On families of n-dimensional vector languages, $n \geqslant 2$, it is neither extensive, nor idempotent, due to the change in dimension. It is, however, extensive for the families $\mathbb{V}(\mathbb{K}; \mathbb{M}) = \bigcup_{n\geqslant 1} \mathbb{V}(\mathbb{K}; n\text{-}\mathbb{M})$ and $\mathbb{V}(\mathbb{M}) = \bigcup_{n\geqslant 1} \mathbb{V}(n\text{-}\mathbb{M})$, where $\mathbb{V}(\mathbb{K}; n\text{-}\mathbb{M})$ and $\mathbb{V}(n\text{-}\mathbb{M})$ are the families from Fig. 2. This can be seen by noting that for each vector language $\mathbb{V}$ from such a family $\mathbb{V}$, the Λ-extension $V_\Lambda = \{(w_1, \ldots, w_n, \Lambda) \mid (w_1, \ldots, w_n) \in \mathbb{V}\}$ belongs to $\mathbb{V}$ as well, while $V = \mathbf{hid}(V_\Lambda)$. Hence $\mathbb{V} \subseteq \mathbf{hid}(\mathbb{V})$.

By repeated application of **hid**, in combination with permutation of components more general hiding operators can be obtained. Since the vector language families studied in this paper are invariant under permutation of components, it is sufficient for our purposes to consider the simple hiding operator defined above.

The hiding operation **hid** has been used in [1] for relating families of vector languages. For instance, it has been proved in [1] that $\mathbf{hid}(\mathbb{V}(\mathrm{p}\mathbb{R}\mathrm{eg}; n+1\text{-}\mathbb{M}\mathrm{on})) = \mathbb{V}(\mathrm{p}\mathbb{R}\mathrm{eg}; n\text{-p}\mathbb{R}\mathrm{eg})$, see also Section 5.2 in this paper.

As this example shows, also the hiding operation is able to enlarge (modulo a change of dimension) the families it operates on. The ‘extra’ component that is hidden is crucial for this enlargement power. To prove the example above it is shown how the simple control languages from $\mathbb{L}(n\text{-}\mathbb{M}\text{on})$ can be used in combination with the hidden component to simulate the larger family $\mathbb{L}(n\text{-p}\mathbb{R}\text{eg})$ of control languages.

In contrast to multi codings the hiding operation is capable of changing dependent word vectors (like $\begin{pmatrix} a \\ \Lambda \\ c \end{pmatrix}$ and $\begin{pmatrix} \Lambda \\ b \\ c \end{pmatrix}$) into independent word vectors (like $\binom{a}{\Lambda}$ and $\binom{\Lambda}{b}$). Thus, properties like completeness and Λ-completeness, that are based on the notion of dependency, can be changed by hiding.

When considered as operation on n-languages the operation **hid** acts as a weak coding: it can map different vector letters onto the same vector letter, and it can remove vector letters. For instance, $\begin{pmatrix} a \\ b \\ c \end{pmatrix}$ and $\begin{pmatrix} a \\ b \\ d \end{pmatrix}$ are both mapped to $\binom{a}{b}$, and $\begin{pmatrix} \Lambda \\ \Lambda \\ c \end{pmatrix}$ is mapped to $\bar{\Lambda}$.

We now formally introduce the third and last operation.

A *multi weak coding (of dimension n)* is a mapping $\Phi = \varphi_1 \times \cdots \times \varphi_n$, where $\varphi_1, \ldots, \varphi_n$ are weak codings. Let n-$\mathbb{MWC}$od denote the class of all n-dimensional multi weak codings.

For any family $\mathbb{V}$ of n-dimensional vector languages, $n\text{-}\mathbb{MWC}\text{od}(\mathbb{V})$ will be the family $n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}) = \{\Phi(V) \mid \Phi \in n\text{-}\mathbb{MWC}\text{od},\ V \in \mathbb{V},\ V \text{ in the domain of } \Phi\}$.

The operator n-$\mathbb{MWC}$od is monotonic, extensive, and idempotent (and hence a closure operator). Moreover, $n\text{-}\mathbb{MWC}\text{od}(n\text{-}\mathbb{MC}\text{od}(\mathbb{V})) = n\text{-}\mathbb{MC}\text{od}(n\text{-}\mathbb{MWC}\text{od}(\mathbb{V})) = n\text{-}\mathbb{MWC}\text{od}(\mathbb{V})$ for any family $\mathbb{V}$ of n-dimensional vector languages.

In the terminology of [4] $n\text{-}\mathbb{MC}\text{od} \leqslant n\text{-}\mathbb{MWC}\text{od}$, i.e. $n\text{-}\mathbb{MC}\text{od}(\mathbb{V}) \subseteq n\text{-}\mathbb{MWC}\text{od}(\mathbb{V})$ for all families $\mathbb{V}$ of n-dimensional vector languages. Thus, any family enlarged by the application of n-$\mathbb{MC}$od may be even further enlarged by the application of n-$\mathbb{MWC}$od.

In contrast to multi codings, multi weak codings are capable of changing the Λ-structure of vectors, namely by replacing non-empty components by empty ones. Thus, properties like completeness and Λ-completeness, that are dependent on the Λ-structure of vectors, can be changed by the application of multi weak codings. In this paper this is expressed by a result like $n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-c}\mathbb{R}\text{eg})) = \mathbb{V}(n\text{-}\mathbb{R}\text{eg})$ (see Section 5.3). As said before, multi codings keep the Λ-structure of vectors and hence the last result cannot be obtained for multi codings.

In [1] multi weak codings have been used to relate families of vector languages. For instance, in that paper – and here again, it has been proved that $n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(\text{p}\mathbb{R}\text{eg}; n\text{-}\mathbb{M}\text{on})) = \mathbb{V}(\text{p}\mathbb{R}\text{eg}; n\text{-p}\mathbb{R}\text{eg})$. Using multi weak codings is essential: $n\text{-}\mathbb{MC}\text{od}(\mathbb{V}(\text{p}\mathbb{R}\text{eg}; n\text{-}\mathbb{M}\text{on})) \subsetneq \mathbb{V}(\text{p}\mathbb{R}\text{eg}; n\text{-p}\mathbb{R}\text{eg})$ as is shown in Section 5.1.

The crucial property of multi weak codings used in the above equality result, is the fact that *each* vector alphabet can be obtained as the image of a multi weak coding applied to a set of vector letters each of the form $(b, b, \ldots, b)$, i.e. vector letters containing one and the same letter in each component. Thus, every vector language can be obtained from an ordinary language by first coding each letter b as a vector letter $(b, b, \ldots, b)$ and then applying a multi weak coding.

Multi weak codings applied to n-languages act as weak codings: vector letters are mapped to vector letters or the empty word vector. However, as for multi codings, not every weak coding on vector letters can be extended to a multi weak coding on word vectors.

5. The representations

In this section we study the effect of multi codings, hidings, and multi weak codings, i.e. the relations of the form $\mathbb{V} = n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}')$, $\mathbb{V} = \mathbf{hid}(\mathbb{V}')$, and $\mathbb{V} = n\text{-}\mathbb{MWC}\mathrm{od}(\mathbb{V}')$, where $\mathbb{V}$ and $\mathbb{V}'$ are families of vector languages. The families $\mathbb{V}'$ considered are those represented in Fig. 2, i.e. the families $\mathbb{V}(\mathbb{K}; n\text{-}\mathbb{M})$ and $\mathbb{V}(n\text{-}\mathbb{M})$, where $\mathbb{K} \in \{\mathbb{A}\mathrm{ll}, \mathrm{p}\mathbb{A}\mathrm{ll}, \mathbb{R}\mathrm{eg}, \mathrm{p}\mathbb{R}\mathrm{eg}, \mathbb{M}\mathrm{on}\}$ and $\mathbb{M} \in \{\mathbb{A}\mathrm{ll}, \lambda\mathbb{A}\mathrm{ll}, \mathrm{c}\mathbb{A}\mathrm{ll}, \mathrm{p}\mathbb{A}\mathrm{ll}, \mathrm{cp}\mathbb{A}\mathrm{ll}, \mathbb{R}\mathrm{eg}, \lambda\mathbb{R}\mathrm{eg}, \mathrm{c}\mathbb{R}\mathrm{eg}, \mathrm{p}\mathbb{R}\mathrm{eg}, \mathrm{cp}\mathbb{R}\mathrm{eg}, \mathbb{M}\mathrm{on}\}$.

Emphasis is on multi codings. They are the most elementary of the three types of operations we consider, as they do not affect the Λ-structure of the vectors to which they are applied. Because of the relatively little effect of multi codings, the application of n-$\mathbb{MC}$od to the families of Fig. 2 leads to more different families than the applications of n-$\mathbb{MWC}$od and **hid**. These latter operations have more effect on the structure of vectors leading to more identifications of families of vector languages. For our technical presentation this means that more proofs and more involved proofs are needed to establish the representation results based on multi codings. The observations on multi codings also form a basis for the results concerning n-$\mathbb{MWC}$od: each multi coding is also a multi weak coding.

The results are presented in graphical form for each of the three operations by the use of so-called *operation diagrams*. They are simply the graphs representing the relations above: in the operation diagram for $\mathbb{O}$, an arrow will be drawn from $\mathbb{V}'$ to $\mathbb{V}$ whenever $\mathbb{O}(\mathbb{V}') = V$, where $\mathbb{O} = n\text{-}\mathbb{MC}\mathrm{od}$, **hid**, or $n\text{-}\mathbb{MWC}\mathrm{od}$. For the families the same drawing convention will be used as in Fig. 2: dots represent families of the form $\mathbb{V}(\mathbb{K}; n\text{-}\mathbb{M})$, circles represent families of the form $\mathbb{V}(n\text{-}\mathbb{M})$, and circled dots represent families for which such forms coincide.

5.1. Multi codings

In order to facilitate the understanding of our approach to establish the results needed to complete the operation diagram for n-$\mathbb{MC}$od, we first present this diagram as Fig. 4. Each relation, i.e. each arrow, is labelled with the corresponding result. This allows for a quick lookup of results. The diagram only gives the results for $n \geqslant 2$. For $n = 1$, the results can be easily derived from Fig. 4 by using the additional equalities holding for the 1-dimensional families (see Fig. 3).

Since n-$\mathbb{MC}$od is a closure operator, the families resulting after a single application of n-$\mathbb{MC}$od are not affected by another application. Hence, it is sufficient to consider only the effect of a single application.

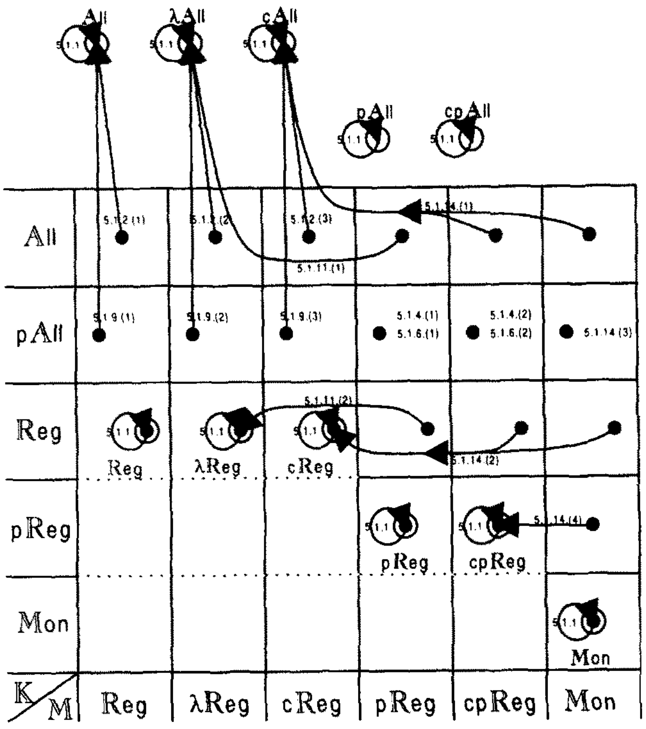

Fig. 4. Operation diagram of n-$\mathbb{M}$$\mathbb{C}$od for the families $\mathbb{V}(\mathbb{K}; n\text{-}\mathbb{M})$ and $\mathbb{V}(n\text{-}\mathbb{M})$ for $n \geqslant 2$.

Preceding the technical details needed to prove the results summarized in this diagram, an overview of our approach and the ordering of the results is given.

Six main steps can be distinguished.

(I) First we argue that all families $\mathbb{V}(n\text{-}\mathbb{M})$ are closed with respect to n-$\mathbb{M}\mathbb{C}$od. This proves the self-loops in the diagram. Because of the equalities established in Section 2, this implies that we are also done for almost half of the entries $\mathbb{V}(\mathbb{K}; n\text{-}\mathbb{M})$ in the diagram.

(II) For the families $\mathbb{V}(\mathbb{A}\text{ll}; n\text{-}\mathbb{M})$ with $\mathbb{M} = \mathbb{R}\text{eg}$, $\lambda\mathbb{R}\text{eg}$, or c$\mathbb{R}$eg, we prove in Lemma 5.1.2 that the application of multi codings leads to the families $\mathbb{V}(n\text{-}\mathbb{A}\text{ll})$, $\mathbb{V}(n\text{-}\lambda\mathbb{A}\text{ll})$, and $\mathbb{V}(n\text{-c}\mathbb{A}\text{ll})$, respectively. By Fig. 2 each of these resulting families strictly includes the original family and hence the original families are not closed with respect to n-$\mathbb{M}\mathbb{C}$od. Still we may compare this result with the effect of multi codings in case the underlying VCCSs have regular component languages: since $\mathbb{V}(\mathbb{R}\text{eg}; n\text{-}\mathbb{M}) = \mathbb{V}(n\text{-}\mathbb{M})$ for $\mathbb{M} = \mathbb{R}\text{eg}$, $\lambda\mathbb{R}\text{eg}$, or c$\mathbb{R}$eg, we have n-$\mathbb{M}\mathbb{C}od(\mathbb{V}(\mathbb{R}\text{eg}; n\text{-}\mathbb{R}\text{eg})) = \mathbb{V}(n\text{-}\mathbb{R}\text{eg})$, n-$\mathbb{M}\mathbb{C}od(\mathbb{V}(\mathbb{R}\text{eg}; n\text{-}\lambda\mathbb{R}\text{eg})) = \mathbb{V}(n\text{-}\lambda\mathbb{R}\text{eg})$, and n-$\mathbb{M}\mathbb{C}od(\mathbb{V}(\mathbb{R}\text{eg}; n\text{-c}\mathbb{R}\text{eg})) = \mathbb{V}(n\text{-c}\mathbb{R}\text{eg})$. Thus, replacing in Lemma 5.1.2 each occurrence of $\mathbb{A}$ll by an occurrence of $\mathbb{R}$eg leads to valid equalities.

(III) At this stage one is tempted to believe that also the representation of the families $\mathbb{V}(n\text{-p}\mathbb{R}\text{eg})$ and $\mathbb{V}(n\text{-cp}\mathbb{R}\text{eg})$ in terms of multi codings and the families $\mathbb{V}$(p$\mathbb{R}$eg;

n-p$\mathbb{R}$eg) and $\mathbb{V}$(p$\mathbb{R}$eg; n-cp$\mathbb{R}$eg), respectively, can be lifted to the level of arbitrary languages.

However, as we show next $\mathbb{V}(n$-p$\mathbb{A}$ll) and $\mathbb{V}(n$-cp$\mathbb{A}$ll) do not have such representations. By Lemma 5.1.4, the families n-$\mathbb{MC}$od($\mathbb{V}$(p$\mathbb{A}$ll; n-p$\mathbb{R}$eg)) and n-$\mathbb{MC}$od($\mathbb{V}$(p$\mathbb{A}$ll; n-cp$\mathbb{R}$eg)) are strictly included in $\mathbb{V}(n$-p$\mathbb{A}$ll) and $\mathbb{V}(n$-cp$\mathbb{A}$ll), respectively. In fact, as further supported by Lemma 5.1.6, none of the families of Fig. 2 is representable as n-$\mathbb{MC}$od($\mathbb{V}$(p$\mathbb{A}$ll; n-p$\mathbb{R}$eg)) or n-$\mathbb{MC}$od($\mathbb{V}$(p$\mathbb{A}$ll; n-cp$\mathbb{R}$eg)).

(IV) Imposing the restriction of prefix-closedness on the component languages only, does not diminish the effect of multi codings. Thus, as stated in Corollary 5.1.9, n-$\mathbb{MC}$od($\mathbb{V}$(p$\mathbb{A}$ll; n-$\mathbb{M}$)) = n-$\mathbb{MC}$od($\mathbb{V}$($\mathbb{A}$ll; n-$\mathbb{M}$)) for $\mathbb{M}$ = $\mathbb{R}$eg, $\lambda\mathbb{R}$eg, or c$\mathbb{R}$eg which leads to new representations of $\mathbb{V}(n$-$\mathbb{A}$ll), $\mathbb{V}(n$-$\lambda\mathbb{A}$ll), and $\mathbb{V}(n$-c$\mathbb{A}$ll), respectively.

(V) In Lemma 5.1.10 conditions are given under which multi codings reduce the effect of prefix-closed control languages. As a consequence, we have in Corollary 5.1.11 the observations that n-$\mathbb{MC}$od($\mathbb{V}$($\mathbb{K}$; n-p$\mathbb{R}$eg)) = n-$\mathbb{MC}$od($\mathbb{V}$($\mathbb{K}$; n-$\lambda\mathbb{R}$eg)) for $\mathbb{K}$ = $\mathbb{A}$ll or $\mathbb{R}$eg. Thus, n-$\mathbb{MC}$od($\mathbb{V}$($\mathbb{A}$ll; n-p$\mathbb{R}$eg)) = $\mathbb{V}(n$-$\lambda\mathbb{A}$ll) and n-$\mathbb{MC}$od($\mathbb{V}$($\mathbb{R}$eg; n-p$\mathbb{R}$eg)) = $\mathbb{V}(n$-$\lambda\mathbb{R}$eg).

(VI) This leaves us to consider the entries $\mathbb{V}$($\mathbb{K}$; n-$\mathbb{M}$on), $\mathbb{V}$(p$\mathbb{K}$; n-$\mathbb{M}$on), and $\mathbb{V}$($\mathbb{K}$; n-cp$\mathbb{R}$eg) with $\mathbb{K}$=$\mathbb{A}$ll or $\mathbb{R}$eg. Using results from [16, 18], we relate vector monoids and complete vector languages. This leads to the results mentioned in Corollary 5.1.14. For $\mathbb{K}$ = $\mathbb{A}$ll or $\mathbb{R}$eg, n-$\mathbb{MC}$od($\mathbb{V}$($\mathbb{K}$; n-$\mathbb{M}$on)) = n-$\mathbb{MC}$od($\mathbb{V}$($\mathbb{K}$; n-cp$\mathbb{R}$eg)) = n-$\mathbb{MC}$od($\mathbb{V}$($\mathbb{K}$; n-c$\mathbb{R}$eg)) which by Lemma 5.1.2 and Fig. 2, respectively, equals $\mathbb{V}(n$-c$\mathbb{K}$); similarly, n-$\mathbb{MC}$od($\mathbb{V}$(p$\mathbb{K}$; n-$\mathbb{M}$on)) = n-$\mathbb{MC}$od($\mathbb{V}$(p$\mathbb{K}$; n-cp$\mathbb{R}$eg)). Since $\mathbb{V}$(p$\mathbb{R}$eg; n-cp$\mathbb{R}$eg) = $\mathbb{V}(n$-cp$\mathbb{R}$eg) is closed with respect to n-$\mathbb{MC}$od, we conclude n-$\mathbb{MC}$od($\mathbb{V}$(p$\mathbb{R}$eg; n-$\mathbb{M}$on)) = $\mathbb{V}(n$-cp$\mathbb{R}$eg).

Hence, except for the families $\mathbb{V}$(p$\mathbb{A}$ll; n-p$\mathbb{R}$eg), $\mathbb{V}$(p$\mathbb{A}$ll; n-cp$\mathbb{R}$eg), and $\mathbb{V}$(p$\mathbb{A}$ll; n-$\mathbb{M}$on), we find for all families that multi codings lead to one of the families $\mathbb{V}(n$-$\mathbb{M}$). Now we turn to the full technical proofs of the results claimed above.

The first step is an easy one.

Lemma 5.1.1. *$\mathbb{V}(n$-$\mathbb{M}) = n$-$\mathbb{MC}$od($\mathbb{V}(n$-$\mathbb{M}$)) for* $\mathbb{M}$ = $\mathbb{A}$ll, x$\mathbb{A}$ll, $\mathbb{R}$eg, x$\mathbb{R}$eg, *or* $\mathbb{M}$on, *where* $x = \lambda$, c, p, *or* cp.

Proof. $\mathbb{V} \subseteq n$-$\mathbb{MC}$od($\mathbb{V}$) for *any* family $\mathbb{V}$ of n-dimensional vector languages.

The inclusions "$\supseteq$" follow directly from the fact that multi codings preserve completeness and Λ-completeness and the fact that the families $\mathbb{A}$ll, p$\mathbb{A}$ll, $\mathbb{R}$eg, p$\mathbb{R}$eg, $\mathbb{M}$on are closed under codings. □

The second step is quite involved.

Lemma 5.1.2. (1) $\mathbb{V}(n$-$\mathbb{A}$ll) = n-$\mathbb{MC}$od($\mathbb{V}$($\mathbb{A}$ll; n-$\mathbb{R}$eg)).

(2) $\mathbb{V}(n$-$\lambda\mathbb{A}$ll) = n-$\mathbb{MC}$od($\mathbb{V}$($\mathbb{A}$ll; n-$\lambda\mathbb{R}$eg)).

(3) $\mathbb{V}(n$-c$\mathbb{A}$ll) = n-$\mathbb{MC}$od($\mathbb{V}$($\mathbb{A}$ll; n-c$\mathbb{R}$eg)).

The proof shows how to construct a vector language V in $\mathbb{V}(n\text{-}\mathbb{A}\text{ll})$ from a vector language in $\mathbb{V}(\mathbb{A}\text{ll}; n\text{-}\mathbb{R}\text{eg})$ using multi codings. It is based on the following idea.

First, the information contained in a word vector from V is put into one of its components; for different word vectors different components may be used. After having synchronized the other components using the relative weak control languages from $\mathbb{L}(n\text{-}\mathbb{R}\text{eg})$, a multi coding is used to remove the extra information from the special components.

Before giving the proof with all formal details we discuss an example.

Example 5.1.3. Let $\mathbb{V} = \{\binom{uv}{wx} \mid u,v,w,x \in \{a,b\}^*$, v is the reverse of w, and $(u = \Lambda$ or $x = \Lambda)\}$.

Then $V \notin \mathbb{V}(\mathbb{A}\text{ll}; 2\text{-}\mathbb{R}\text{eg})$, which can be seen as follows. Suppose $V = (K_1' \times K_2') \cap \textbf{coll}(M)$ for some $K_1', K_2' \in \mathbb{A}\text{ll}$ and an $M \in \mathbb{R}\text{eg}$. Then, since $\{a,b\}^* = \textbf{proj}_1(V) = \textbf{proj}_2(V)$, we may assume without loss of generality that $K_1' = K_2' = \{a,b\}^*$. Thus, it follows that $V = (\{a,b\}^* \times \{a,b\}^*) \cap \textbf{coll}(M) \in \mathbb{V}(\mathbb{R}\text{eg}; 2\text{-}\mathbb{R}\text{eg}) = \mathbb{V}(2\text{-}\mathbb{R}\text{eg})$. Using the pumping lemma for regular languages (see [12]), it can easily be shown that $V \notin \mathbb{V}(2\text{-}\mathbb{R}\text{eg})$, which gives a contradiction. Hence $V \notin \mathbb{V}(\mathbb{A}\text{ll}; 2\text{-}\mathbb{R}\text{eg})$.

We now show that $V \in 2\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{V}(\mathbb{A}\text{ll}; 2\text{-}\mathbb{R}\text{eg}))$ by constructing $K_1, K_2 \in \mathbb{A}\text{ll}$, all $M \in \mathbb{R}\text{eg}$ and a $\Phi \in 2\text{-}\mathbb{M}\mathbb{C}\text{od}$, such that $V = \Phi[(K_1 \times K_2) \cap \textbf{coll}(M)]$.

Let $\theta = \textbf{Tot}(\{a,b\},\{a,b\})$, let $\theta_1 = \{\binom{\sigma}{\tau} \in \theta \mid \sigma \neq \Lambda\}$, and let $\theta_2 = \{\binom{\sigma}{\tau} \in \theta \mid \tau \neq \Lambda\}$, i.e.

$$\theta_1 = \left\{\begin{pmatrix} a \\ \Lambda \end{pmatrix}, \begin{pmatrix} b \\ \Lambda \end{pmatrix}, \begin{pmatrix} a \\ a \end{pmatrix}, \begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} b \\ a \end{pmatrix}, \begin{pmatrix} b \\ b \end{pmatrix}\right\},$$

and

$$\theta_2 = \left\{\begin{pmatrix} \Lambda \\ a \end{pmatrix}, \begin{pmatrix} \Lambda \\ b \end{pmatrix}, \begin{pmatrix} a \\ a \end{pmatrix}, \begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} b \\ a \end{pmatrix}, \begin{pmatrix} b \\ b \end{pmatrix}\right\}.$$

Next let $\Gamma_1 = \{\begin{pmatrix} \binom{\sigma}{\tau} \\ \tau \end{pmatrix} \mid \binom{\sigma}{\tau} \in \theta_1\}$ and $\Gamma_2 = \{\begin{pmatrix} \sigma \\ \binom{\sigma}{\tau} \end{pmatrix} \mid \binom{\sigma}{\tau} \in \theta_1\}$, i.e.

$$\Gamma_1 = \left\{\begin{pmatrix} \binom{a}{\Lambda} \\ \Lambda \end{pmatrix}, \begin{pmatrix} \binom{b}{\Lambda} \\ \Lambda \end{pmatrix}, \begin{pmatrix} \binom{a}{a} \\ a \end{pmatrix}, \begin{pmatrix} \binom{a}{b} \\ b \end{pmatrix}, \begin{pmatrix} \binom{b}{a} \\ a \end{pmatrix}, \begin{pmatrix} \binom{b}{b} \\ b \end{pmatrix}\right\},$$

and

$$\Gamma_2 = \left\{\begin{pmatrix} \Lambda \\ \binom{\Lambda}{a} \end{pmatrix}, \begin{pmatrix} \Lambda \\ \binom{\Lambda}{b} \end{pmatrix}, \begin{pmatrix} a \\ \binom{a}{a} \end{pmatrix}, \begin{pmatrix} a \\ \binom{a}{b} \end{pmatrix}, \begin{pmatrix} b \\ \binom{b}{a} \end{pmatrix}, \begin{pmatrix} b \\ \binom{b}{b} \end{pmatrix}\right\}.$$

Set $M = \Gamma_1^+ \cup \Gamma_2^+ \cup \{\Lambda\}$.

Let $\Phi : (\Gamma_1 \cup \Gamma_2)^{\circledast} \to \theta^{\circledast}$ be the 2-dimensional coding $\Phi = \varphi_1 \times \varphi_2$, where φ_1 and φ_2 are the codings defined by $\varphi_i(a) = a$, $\varphi_i(b) = b$, and $\varphi_i(\vartheta) = \mathbf{proj}_i(\vartheta)$ for all $\vartheta \in \theta$, $i = 1,2$. Thus,

$$\Phi\left(\left(\begin{pmatrix}\binom{b}{a}\\ a\end{pmatrix}\begin{pmatrix}\binom{b}{b}\\ b\end{pmatrix}\begin{pmatrix}a\\ \Lambda\end{pmatrix}\right)\right) = \begin{pmatrix}bba\\ ab\end{pmatrix}$$

and

$$\Phi\left(\left(\begin{pmatrix}b\\ \binom{b}{a}\end{pmatrix}\begin{pmatrix}b\\ \binom{b}{b}\end{pmatrix}\begin{pmatrix}a\\ \binom{a}{b}\end{pmatrix}\right)\right) = \begin{pmatrix}bba\\ abb\end{pmatrix}.$$

Let $L = L_1 \cup L_2$, where $L_1 = \{v \in \{\binom{a}{a}, \binom{a}{b}, \binom{b}{a}, \binom{b}{b}\}^*\{\binom{a}{\Lambda}, \binom{b}{\Lambda}\}^* \mid \mathbf{proj}_{\theta^*,2}(v)$ is a prefix of the reverse of $\mathbf{proj}_{\theta^*,1}(v)\}$ and $L_2 = \{v \in \{\binom{a}{a}, \binom{a}{b}, \binom{b}{a}, \binom{b}{b}\}^*\{\binom{\Lambda}{a}, \binom{\Lambda}{b}\}^* \mid$ $\mathbf{proj}_{\theta^*,1}(v)$ is a prefix of the reverse of $\mathbf{proj}_{\theta^*,2}(v)\}$. Here L_i, $i = 1,2$, is chosen such that $\mathbf{coll}(L_i)$ is the set of all word vectors from V having an ith component longer than or at least as long as the other component. Moreover, L_i is built from vector letters all having a non-empty ith component.

Clearly, $V = \mathbf{coll}(L)$. Note that $\mathbf{coll}(L_1)$ and $\mathbf{coll}(L_2)$ are incomparable, but not disjoint. For instance, $\binom{bba}{ab} \in \mathbf{coll}(L_1) - \mathbf{coll}(L_2)$, whereas $\binom{bba}{abb} \in \mathbf{coll}(L_1) \cap \mathbf{coll}(L_2)$.

Let $K_1 = L_1 \cup \mathbf{proj}_{\theta^*,1}(L) = L_1 \cup \{a,b\}^*$ and let $K_2 = L_2 \cup \mathbf{proj}_{\theta^*,2}(L) = L_2 \cup \{a,b\}^*$. Thus, e.g., $\binom{b}{a}\binom{b}{b}\binom{a}{\Lambda} \in K_1$ and $bba \in K_1$.

It is not very difficult to see that $V = \Phi[(K_1 \times K_2) \cap \mathbf{coll}(M)]$. All formalities can be found in the proof of Lemma 5.1.2. □

Proof of Lemma 5.1.2 From Lemma 5.1.1 we know that $\mathbb{V}(n\text{-}\mathbb{A}\mathrm{ll}) = n\text{-}\mathbb{M}\mathbb{C}\mathrm{od}(\mathbb{V}(n\text{-}\mathbb{A}\mathrm{ll}))$ and by the inclusion diagram Fig. 2 we have $\mathbb{V}(\mathbb{A}\mathrm{ll}; n\text{-}\mathbb{R}\mathrm{eg}) \subseteq \mathbb{V}(n\text{-}\mathbb{A}\mathrm{ll})$. This proves that $\mathbb{V}(n\text{-}\mathbb{A}\mathrm{ll}) \supseteq n\text{-}\mathbb{M}\mathbb{C}\mathrm{od}(\mathbb{V}(\mathbb{A}\mathrm{ll}; n\text{-}\mathbb{R}\mathrm{eg}))$. Similar reasoning, again using Lemma 5.1.1 and Fig. 2 shows that $\mathbb{V}(n\text{-}\lambda\mathbb{A}\mathrm{ll}) \supseteq n\text{-}\mathbb{M}\mathbb{C}\mathrm{od}(\mathbb{V}(\mathbb{A}\mathrm{ll}; n\text{-}\lambda\mathbb{R}\mathrm{eg}))$ and $\mathbb{V}(n\text{-c}\mathbb{A}\mathrm{ll}) \supseteq n\text{-}\mathbb{M}\mathbb{C}\mathrm{od}(\mathbb{V}(\mathbb{A}\mathrm{ll}; n\text{-c}\mathbb{R}\mathrm{eg}))$. This leaves us to prove the converse inclusions.

With this aim we start out with an arbitrary language $V \in \mathbb{V}(n\text{-}\mathbb{A}\mathrm{ll})$. After some consideration about the alphabets involved, we define an n-dimensional coding Φ, n languages $K_1, \ldots, K_n$, and a regular language M such that $\Phi[\bigtimes_{i=1}^n K_i \cap \mathbf{coll}(M)] = V - \{\bar{\Lambda}\}$. Finally, we show how to extend M to a regular M' such that $\Phi[\mathsf{X}_{i=1}^n K_i \cap \mathbf{coll}(M')] = V$, and such that $\mathbf{coll}(M')$ is Λ-complete or complete if V is Λ-complete or complete, respectively.

So let $V \in \mathbb{V}(n\text{-}\mathbb{A}\mathrm{ll})$. Let θ be a total vector alphabet such that $V \subseteq \theta^{\circledast}$ and let $\theta_i = \{\vartheta \in \theta \mid \mathbf{proj}_i(\vartheta) \neq \Lambda\}$, for $i = 1, \ldots, n$. Without loss of generality, we may assume that $\theta \cap \mathbf{proj}_i(\theta) = \emptyset$ for all $i \in [n]$, i.e. vector letters of θ do *not* occur as components in (other) vector letters from θ. For each $i \in [n]$ and each $\vartheta = (\vartheta_1, \ldots, \vartheta_n) \in \theta_i$, the vector letter $\gamma_{\vartheta,i}$ is defined by $\gamma_{\vartheta,i} = (\vartheta_1, \ldots, \vartheta_{i-1}, \vartheta, \vartheta_{i+1}, \ldots, \vartheta_n)$. So $\gamma_{\vartheta,i}$ is equal to ϑ except for the ith component, where ϑ itself is occurring as a component. Note that $\gamma_{\vartheta,i}$ and ϑ have the same Λ-structure due to the definition of θ_i. Let $\Gamma_i = \{\gamma_{\vartheta,i} \mid \vartheta \in \theta_i\}$ for $i = 1, \ldots, n$.

Let $\Phi : (\bigcup_{i=1}^{n} \Gamma_i)^{\circledast} \to \theta^{\circledast}$ be the homomorphism defined by $\Phi(\gamma_{\vartheta,i}) = \vartheta$, for all $i \in [n]$ and all $\vartheta \in \theta_i$. Then it is easy to see that $\Phi = \varphi_1 \times \cdots \times \varphi_n$, where, for all $i \in [n]$, $\varphi_i : (\theta_i \cup \mathbf{proj}_i(\theta))^* \to \mathbf{proj}_i(\theta)^*$ is the coding defined by: $\varphi_i(\vartheta) = \mathbf{proj}_i(\vartheta)$ if $\vartheta \in \theta_i$ and $\varphi_i(\vartheta) = \vartheta$ if $\vartheta \in \mathbf{proj}_i(\theta)$. Hence, $\Phi \in n$-$\mathbb{M}\mathbb{C}$od. Note that $\Phi(\mathbf{coll}(v)) = \mathbf{coll} \circ \mathbf{proj}_i \circ \mathbf{coll}(v)$ for all $i \in [n]$ and all $v \in \Gamma_i^*$.

Next we define the component languages $K_1, \ldots, K_n$ and the control language M.

For this, we first choose a language L with $\mathbf{coll}(L) = V$, such that $L = L_1 \cup \cdots \cup L_n$ where $L_i \subseteq \theta_i^*$ for all $i \in [n]$. In other words, L is chosen in such a way that, for each word $v \in L$, there is an $i \in [n]$, such that each vector letter from v has a non-empty ith component. Such an L can be chosen because θ is a total vector alphabet.

Next, set $K_i = \mathbf{proj}_{\theta^*,i}(L) \cup L_i$, for all $i \in [n]$, and set $M = \bigcup_{i=1}^{n} \Gamma_i^+$. Note that M is regular.

It is now easy to see that $w \in \mathsf{X}_{i=1}^{n} K_i \cap \mathbf{coll}(M)$ if and only if there is a $j \in [n]$ such that $w \in \Gamma_j^{\circledast}$ and $\mathbf{proj}_j(w) \in \theta_j^+ \cap L_j$. Consequently, whenever $w \in \mathsf{X}_{i=1}^{n} K_i \cap \mathbf{coll}(M)$, then $\Phi(w) \in \mathbf{coll}(L_j) - \{\bar{\Lambda}\} \subseteq V - \{\bar{\Lambda}\}$. Conversely, whenever $w \in V - \{\bar{\Lambda}\}$, then $w = \mathbf{coll}(v)$ for some $v \in L_j - \{\Lambda\}$ and $j \in [n]$. Thus, $v = \vartheta_1 \ldots \vartheta_m$ for some $m \geqslant 1$ and $\vartheta_l \in \theta_j$, for all $l \in [m]$. Then $w = \Phi(\mathbf{coll}(\gamma_{\vartheta_1,j} \cdots \gamma_{\vartheta_m,j}))$ and $\mathbf{coll}(\gamma_{\vartheta_1,j} \cdots \gamma_{\vartheta_m,j}) \in (\mathsf{X}_{i=1}^{n} K_i) \cap \mathbf{coll}(M)$. From this we may now conclude that $V - \{\bar{\Lambda}\} = \Phi[\mathsf{X}_{i=1}^{n} K_i \cap \mathbf{coll}(M)]$.

The last part of the proof concentrates on the extension of M in order to take care of the case that $\bar{\Lambda} \in V$ and of Λ-completeness and completeness.

If $\bar{\Lambda} \notin V$ and V is not Λ-complete, we leave M as it is. In that case we already have $V = \Phi[\mathsf{X}_{i=1}^{n} K_i \cap \mathbf{coll}(M)]$.

If $\bar{\Lambda} \in V$ or V is Λ-complete, we add Λ to M. In this way we obtain $M' = \bigcup_{i=1}^{n} \Gamma_i^+ \cup \{\Lambda\}$ which is a regular language, and $\mathbf{coll}(M')$ is Λ-complete. If $\bar{\Lambda} \in V$, then $\Lambda \in L$, and hence $\Lambda \in K_i$ for all $i \in [n]$. (#)
Thus, it follows that $\Phi[\mathsf{X}_{i=1}^{n} K_i \cap \mathbf{coll}(M')] = \Phi[(\mathsf{X}_{i=1}^{n} K_i \cap \mathbf{coll}(M)) \cup \{\bar{\Lambda}\}] = (V - \{\bar{\Lambda}\}) \cup \{\bar{\Lambda}\} = V$.
If $\bar{\Lambda} \notin V$ and V is Λ-complete, then $\Lambda \notin L$ and $\Lambda \notin \mathbf{proj}_j(V)$ for some $j \in [n]$. Hence, for this $j, \Lambda \notin \mathbf{proj}_{\theta^*,j}(L)$ and $\Lambda \notin L_j$. Hence, $\Lambda \notin K_j$. (##)
Consequently, $\bar{\Lambda} \notin \mathsf{X}_{i=1}^{n} K_i$. Thus, it follows that $\Phi[\mathsf{X}_{i=1}^{n} K_i \cap \mathbf{coll}(M')] = \Phi[\mathsf{X}_{i=1}^{n} K_i \cap \mathbf{coll}(M)] = V - \{\bar{\Lambda}\} = V$.

This leaves us with the case that V is complete. Now we extend M to $M' = \{v_1 \ldots v_m \mid m \geqslant 1, v_i \in M, \mathbf{coll}(v_j) \text{ and } \mathbf{coll}(v_k) \text{ are independent if } j \neq k, j,k \in [n]\} \cup \{\bar{\Lambda}\}$. Clearly, M' is complete and it is not difficult to see that M' is regular. Using the reasoning given at (#) above it follows easily that $\bar{\Lambda} \in V$ implies $\bar{\Lambda} \in \mathsf{X}_{i=1}^{n} K_i \cap \mathbf{coll}(M')$, and hence $V \subseteq \Phi[\mathsf{X}_{i=1}^{n} K_i \cap \mathbf{coll}(M')]$.

The last thing to prove is that $\Phi[\mathsf{X}_{i=1}^{n} K_i \cap \mathbf{coll}(M')] \subseteq V$, or equivalently that, for all $v \in M' - M$, $\Phi[\mathsf{X}_{i=1}^{n} K_i \cap \mathbf{coll}(\{v\})] \subseteq V$.
If $v = \Lambda$, then we can apply the reasonings given at (#) and (##) above which tell us that, since V is Λ-complete, $\bar{\Lambda} \in \mathsf{X}_{i=1}^{n} K_i$ if and only if $\bar{\Lambda} \in V$. Hence, $\Phi[\mathsf{X}_{i=1}^{n} K_i \cap \mathbf{coll}(\{\Lambda\})] \subseteq V$.

So assume that $v \neq \Lambda$ and that $\mathbf{coll}(v) \in \times_{i=1}^{n} K_i$. (If $\mathbf{coll}(v) \notin \times_{i=1}^{n} K_i$, there is nothing to prove.) Since $v \neq \Lambda$ and $v \in M' - M$, there is an $m \geqslant 2$ such that $v = v_1 \ldots v_m$ with $v_j \in M$ for all $j \in [m]$ and $\mathbf{coll}(v_i)$ and $\mathbf{coll}(v_k)$ are independent if $i \neq k$. By the definition of M, for each $j \in [m]$ there is an $i_j \in [n]$ such that $v_j \in \Gamma_{i_j}^{+}$. By the independence of the $\mathbf{coll}(v_j)$, we know that $i_j \neq i_k$ for all $j, k \in [m]$ with $j \neq k$. Since $K_i = \mathbf{proj}_{\theta^*, i}(L) \cup L_i$, for all $i \in [n]$ and $\mathbf{coll}(v_1 \ldots v_m) \in \times_{i=1}^{n} K_i$, we have that $\mathbf{proj}_{i_j}(\mathbf{coll}(v_1 \ldots v_m)) = \mathbf{proj}_{i_j}(\mathbf{coll}(v_j)) \in L_{i_j}$. Thus, $\Phi(\mathbf{coll}(v_j)) = \mathbf{coll}(\mathbf{proj}_{i_j}(\mathbf{coll}(v_j)) \in \mathbf{coll}(L_{i_j}) \subseteq V$. Moreover, $\{i \in [n] \mid \Phi(\mathbf{coll}(v_j)) \neq \Lambda\} = \{i \in [n] \mid \mathbf{coll}(v_j) \neq \Lambda\}$ and $\Phi(\mathbf{coll}(v_1 \ldots v_m)) = \Phi(\mathbf{coll}(v_1)) \odot \cdots \odot \Phi(\mathbf{coll}(v_m))$. Combining all this leads to $\Phi(\times_{i=1}^{n} K_i \cap \mathbf{coll}(v_1 \ldots v_m)) = w_1 \odot \cdots \odot w_m$, where $w_j \in V$ for $j \in [m]$ and w_i is independent from w_k if $i \neq k$. Since V is complete, this leads to the desired conclusion that $w_1 \odot \cdots \odot w_m \in V$. □

As already pointed out under (III) above, in case of prefix-closedness the situation changes.

Lemma 5.1.4. *Let $n \geqslant 2$.*

(1) $\mathbb{V}(n\text{-p}\mathbb{A}\text{ll}) \supsetneq n\text{-}\mathbb{MC}\text{od}(\mathbb{V}(\text{p}\mathbb{A}\text{ll}; n\text{-p}\mathbb{R}\text{eg}))$.
(2) $\mathbb{V}(n\text{-cp}\mathbb{A}\text{ll}) \supsetneq n\text{-}\mathbb{MC}\text{od}(\mathbb{V}(\text{p}\mathbb{A}\text{ll}; n\text{-cp}\mathbb{R}\text{eg}))$.

In an auxiliary lemma we first consider the example vector language to be used in the proof of Lemma 5.1.4.

Lemma 5.1.5. *Let $V = \{\binom{w_1}{w_2} \in \{a,b\}^* \times \{a,b\}^* \mid w_2$ is a prefix of the reverse of $w_1)\}$.*

Then $V \in \mathbb{V}(n\text{-cp}\mathbb{A}\text{ll}) - 2\text{-}\mathbb{MC}\text{od}(\mathbb{V}(\text{p}\mathbb{A}\text{ll}; 2\text{-p}\mathbb{R}\text{eg}))$.

Proof of Lemma 5.1.5 Let $K = \{v_1 v_2 \mid v_1 \in \{\binom{a}{\Lambda}, \binom{b}{\Lambda}\}^*, v_2 \in \{\binom{\Lambda}{a}, \binom{\Lambda}{b}\}^*$, $\mathbf{proj}_2(v_2)$ is a prefix of the reverse of $\mathbf{proj}_1(v_1)\}$. Then $V = \mathbf{coll}(K)$. Since K is prefix-closed and V is complete, this shows that $V \in \mathbb{V}(2\text{-cp}\mathbb{A}\text{ll})$.

In order to prove that $V \notin 2\text{-}\mathbb{MC}\text{od}(\mathbb{V}(\text{p}\mathbb{A}\text{ll}; 2\text{-p}\mathbb{R}\text{eg}))$, we assume to the contrary that $V = \Phi[W]$ where $\Phi = \varphi_1 \times \varphi_2 \in 2\text{-}\mathbb{MC}\text{od}$ and $W = (K_1 \times K_2) \cap \mathbf{coll}(M)$ with $K_1, K_2 \in \text{p}\mathbb{A}\text{ll}$ and $M \in \mathbb{L}(2\text{-p}\mathbb{R}\text{eg})$.

Then we can make the following technical observation: for all word vectors $w = \binom{vdc}{efx} \in W$ where $\varphi_1(c) = \varphi_2(e) = a$, $\varphi_1(d) = \varphi_2(f) = b$, and where v and x are words such that $\varphi_2(x)$ is a prefix of the reverse of $\varphi_1(v)$, there exist words r and s such that $\mathbf{coll}(r) = \binom{vdc}{e}$, $\mathbf{coll}(s) = \binom{\Lambda}{fx}$ and $rs \in M$. (&)

This can be seen as follows. As $w \in \mathbf{coll}(M)$, it follows that there exists a word $\hat{t} \in M$ such that $w = \mathbf{coll}(\hat{t})$. Clearly, there exist a prefix $\hat{r}$ of $\hat{t}$ such that $\mathbf{coll}(\hat{r}) = \binom{u}{e}$ where u is a prefix of vdc. If u is a prefix of vd, then clearly $\hat{r}$ can be extended to the prefix $\hat{s}$ of $\hat{t}$ satisfying $\mathbf{coll}(\hat{s}) = \binom{vd}{ey}$ where y is a prefix of fx. Since K_1, K_2 and M are prefix-closed this leads to the conclusion that $\binom{vd}{ey} \in W$ and hence $\binom{\varphi_1(v)b}{a\varphi_2(y)} \in V$. The latter is a contradiction, because $a\varphi_2(y)$ is *not* a prefix of the reverse of $\varphi_1(v)b$.

Consequently, $u = vdc$ which proves (&).

Now consider the sequence of word vectors $\binom{ba^iba}{aba^ib} \in V$, $i \geqslant 0$. Hence, by our assumption, there is a sequence of word vectors $w_i = \binom{v_id_ic_i}{e_if_ix_i} \in W$, $i \geqslant 0$, such that $\varphi_1(v_i) = ba^i$, $\varphi_1(c_i) = \varphi_2(e_i) = a$, $\varphi_1(d_i) = \varphi_2(f_i) = b$, and $\varphi_2(x_i) = a^ib$, for all $i \geqslant 0$. Since $\varphi_2^{-1}(a)$ is finite, it follows that there exists an $e \in \varphi_2^{-1}(a)$ such that $e_i = e$ infinitely often.

Hence, by our observation (&) above, there are infinitely many r_j, s_j with $\mathbf{coll}(r_j) = \binom{v_jd_jc_j}{e}$ and $\mathbf{coll}(s_j) = \binom{\Lambda}{f_jx_j}$ such that $r_js_j \in M$ and $e_j = e$. Since M is regular, there exist $k \neq l$, such that $r_ks_l \in M$. Now $\mathbf{coll}(r_ks_l) \in W$, because $\mathbf{proj}_1(\mathbf{coll}(r_ks_l)) = v_kd_kc_k = \mathbf{proj}_1(w_k) \in K_1$ and $\mathbf{proj}_2(\mathbf{coll}(r_ks_l)) = ef_lx_l = \mathbf{proj}_2(w_l) \in K_2$. However, $\Phi(\mathbf{coll}(r_ks_l)) = \binom{ba^kba}{aba^lb} \notin V$, because a^lb is *not* a prefix of a a^kb since $k \neq l$.

This contradiction proves that $V \notin 2\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; 2\text{-}\mathrm{p}\mathbb{R}\mathrm{eg}))$. □

Proof of Lemma 5.1.4 From Lemma 5.1.1 we know that $\mathbb{V}(n\text{-}\mathrm{p}\mathbb{A}\mathrm{ll}) = n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(n\text{-}\mathrm{p}\mathbb{A}\mathrm{ll}))$ and by the inclusion diagram (Fig. 2) we have $\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathrm{p}\mathbb{R}\mathrm{eg}) \subseteq \mathbb{V}(n\text{-}\mathrm{p}\mathbb{A}\mathrm{ll})$. This proves that $\mathbb{V}(n\text{-}\mathrm{p}\mathbb{A}\mathrm{ll}) \supseteq n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathrm{p}\mathbb{R}\mathrm{eg}))$. Similar reasoning, again using Lemma 5.1.1 and Fig. 2, shows that $\mathbb{V}(n\text{-}\mathrm{cp}\mathbb{A}\mathrm{ll}) \supseteq n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathrm{cp}\mathbb{R}\mathrm{eg}))$.

In order to prove both inclusions to be strict, we show that $\mathbb{V}(n\text{-}\mathrm{cp}\mathbb{A}\mathrm{ll}) - n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathrm{p}\mathbb{R}\mathrm{eg})) \neq \emptyset$ if $n \geqslant 2$. This is sufficient, because $\mathbb{V}(n\text{-}\mathrm{cp}\mathbb{A}\mathrm{ll}) \subseteq \mathbb{V}(n\text{-}\mathrm{p}\mathbb{A}\mathrm{ll})$ and $n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathrm{p}\mathbb{R}\mathrm{eg}) \supseteq n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathrm{cp}\mathbb{R}\mathrm{eg})$.

For $n = 2$, the above follows from Lemma 5.1.5. For dimensions greater than 2, the example can be obtained from Lemma 5.1.5 by adding Λ-components to the vectors involved. □

The following lemma shows that the two families $n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathrm{y}\mathbb{R}\mathrm{eg}))$, $y \in \{\mathrm{p}, \mathrm{cp}\}$, are not closed with respect to n-$\mathbb{MC}$od. Combining this with Lemma 5.1.4 and Fig. 2 shows that they do not correspond to any of the families of the diagram.

Lemma 5.1.6. *Let* $n \geqslant 2$.

(1) $n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathrm{p}\mathbb{R}\mathrm{eg})) \supsetneq \mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathrm{p}\mathbb{R}\mathrm{eg})$.

(2) $n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathrm{cp}\mathbb{R}\mathrm{eg})) \supsetneq \mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathrm{cp}\mathbb{R}\mathrm{eg})$.

Proof. The lemma is proved by proving the single non-inclusion result $n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathrm{cp}\mathbb{R}\mathrm{eg})) - \mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathrm{p}\mathbb{R}\mathrm{eg}) \neq \emptyset$ for $n \geqslant 2$. From the proof of Lemma 3.1.2 (5) we know that $V = \{\binom{a^{h+k}}{c^hd^k} \mid 0 \leqslant k \leqslant h\} \cup \{\binom{e^p}{w} \mid p \geqslant 1, w \in \{c,d\}^*\} \notin \mathbb{V}(\mathbb{A}\mathrm{ll}; 2\text{-}\mathbb{R}\mathrm{eg})$ and hence $V \notin \mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; 2\text{-}\mathrm{p}\mathbb{R}\mathrm{eg})$. After proving that $V \in 2\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; 2\text{-}\mathrm{cp}\mathbb{R}\mathrm{eg}))$, we are done, because the same vector language with additional Λ-components can be used for the case that $n > 2$.

Let $K_1 \in \mathrm{p}\mathbb{A}\mathrm{ll}$, $K_2 \in \mathrm{p}\mathbb{A}\mathrm{ll}$, and $M \in \mathbb{L}(2\text{-}\mathrm{cp}\mathbb{R}\mathrm{eg})$ be given by: $K_1 = \{a^hb^k \mid 0 \leqslant k \leqslant h\} \cup \{e\}^*$, $K_2 = \{c,d\}^*$, and $M = \binom{a}{c}^*\binom{b}{d}^* \cup \binom{e}{\Lambda}\{\binom{e}{\Lambda}, \binom{\Lambda}{c}, \binom{\Lambda}{d}\}^*$. Set $\Phi = \varphi_1 \times \varphi_2 \in$ 2-$\mathbb{MC}$od, where $\varphi_1(a) = \varphi_1(b) = a$, $\varphi_1(e) = e$, $\varphi_2(c) = c$, and $\varphi_2(d) = d$. Then $V = \Phi[(K_1 \times K_2) \cap \mathbf{coll}(M)]$ and hence $V \in 2\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; 2\text{-}\mathrm{cp}\mathbb{R}\mathrm{eg}))$. □

In the next step we turn to a general lemma showing when prefix-closedness of component languages can be "overcome" with the help of multi codings. The construction in the proof makes use of an endmarking technique.

Lemma 5.1.7. $\mathbb{V}(\mathbb{K}; n\text{-}\mathbb{M}) \subseteq n\text{-}\mathbb{M}\mathbb{C}\mathrm{od}(\mathbb{V}(\mathrm{p}\mathbb{K}; n\text{-}\mathbb{M}))$ *provided the families* $\mathbb{K}$ *and* $\mathbb{M}$ *of languages satisfy the following properties*:

– $\mathbb{K}$ *is closed under inverse codings, under intersection with regular languages, and under* **pref**,

– $\mathbb{V}(n\text{-}\mathbb{M})$ *is closed under inverse n-dimensional codings and under intersection with n-fold cartesian products of regular languages.*

Proof. Let $K_i \in \mathbb{K}$ for all $i \in [n]$ and let $V \in \mathbb{V}(n\text{-}\mathbb{M})$. We construct, for all $i \in [n]$, a language $K_i' \in \mathrm{p}\mathbb{K}$, a vector language $V' \in \mathbb{V}(n\text{-}\mathbb{M})$, and an n-dimensional coding $\Phi \in n\text{-}\mathbb{M}\mathbb{C}\mathrm{od}$, such that $(\mathsf{X}_{i=1}^{n} K_i) \cap V = \Phi[(\mathsf{X}_{i=1}^{n} K_i') \cap V']$.

Let Σ be an alphabet such that $K_i \subseteq \Sigma^*$, for all $i \in [n]$, and such that $V \subseteq \Sigma^* \times \cdots \times \Sigma^*$. Let $\hat{\Sigma}$ be a disjoint copy of Σ and define the endmarking $\mu : \Sigma^* \to \Sigma^*\hat{\Sigma} \cup \{\Lambda\}$ by $\mu(\Lambda) = \Lambda$ and $\mu(wa) = w\hat{a}$ for all $w \in \Sigma^*$ and $a \in \Sigma$. Then we set, for all $i \in [n]$, $K_i' = \mathbf{pref}\{\mu(w) \mid w \in K_i\}$, and $V' = \{(\mu(w_1), \ldots, \mu(w_n)) \mid (w_1, \ldots, w_n) \in V \cap \mathsf{X}_{i=1}^{n} L_i\}$, where, for all $i \in [n]$, $L_i = \Sigma^*$ if $\Lambda \in K_i$ and $L_i = \Sigma^+$ if $\Lambda \notin K_i$. Since in the K_i' all and only the original non-empty words from the K_i have an endmarking and all non-empty components of words from V have an endmarking, it is easy to see that $\mathsf{X}_{i=1}^{n} K_i' \cap V' = \{(\mu(w_1), \ldots, \mu(w_n)) \mid (w_1, \ldots, w_n) \in \mathsf{X}_{i=1}^{n} K_i \cap V\}$. Hence, $\mathsf{X}_{i=1}^{n} K_i \cap V = \Phi[\mathsf{X}_{i=1}^{n} K_i' \cap V']$, where $\Phi = \varphi \times \cdots \times \varphi \in n\text{-}\mathbb{M}\mathbb{C}\mathrm{od}$ removes the endmarkings through the coding φ defined by $\varphi(\hat{b}) = \varphi(b) = b$ for all $b \in \Sigma$. Note that the intersection by $\mathsf{X}_{i=1}^{n} L_i$ is necessary to handle the empty words: if, for an $i \in [n]$, $\Lambda \in K_i' - K_i$, then $\Lambda \notin \mathbf{proj}_i(V')$, because of this intersection.

The only things left to prove are that $K_i' \in \mathrm{p}\mathbb{K}$, for all $i \in [n]$, and that $V' \in \mathbb{V}(n\text{-}\mathbb{M})$. Clearly, $\{\mu(w) \mid w \in K\} = \varphi^{-1}(K) \cap (\Sigma^*\hat{\Sigma} \cup \{\Lambda\})$, for $K \subseteq \Sigma^*$. The properties of $\mathbb{K}$ now guarantee that $\mathbf{pref}(\varphi^{-1}(K) \cap (\Sigma^*\hat{\Sigma} \cup \{\Lambda\})) \in \mathbb{K}$ whenever $K \subseteq \Sigma^*$ is in $\mathbb{K}$. Since, for each $i \in [n]$, $K_i' = \mathbf{pref}(\varphi^{-1}(K_i) \cap (\Sigma^*\hat{\Sigma} \cup \{\Lambda\})$ is prefix-closed and $K_i \in \mathbb{K}$, we have $K_i' \in \mathrm{p}\mathbb{K}$ for all $i \in [n]$. Similarly, V' can be written as $V' = \Phi^{-1}(V) \cap \mathsf{X}_{i=1}^{n}(\Sigma^*\hat{\Sigma} \cup \{\Lambda\}) \cap \mathsf{X}_{i=1}^{n} L_i$, which – due to the closure properties of $\mathbb{V}(n\text{-}\mathbb{M})$ – shows that $V' \in \mathbb{V}(n\text{-}\mathbb{M})$. □

In order to apply the above lemma we need to verify that the conditions are satisfied. In particular, we want to apply the lemma for the cases that $\mathbb{M} = \mathbb{R}\mathrm{eg}, \lambda\mathbb{R}\mathrm{eg}, \mathrm{c}\mathbb{R}\mathrm{eg}$. From Proposition 5.1.8 below, it follows that $\mathbb{V}(n\text{-}\mathbb{R}\mathrm{eg})$ satisfies the required closure properties.

Proposition 5.1.8. (1) $\mathbb{V}(n\text{-}\mathbb{R}\mathrm{eg})$ *is closed under inverse n-dimensional multi codings.*

(2) $\mathbb{V}(n\text{-}\mathbb{R}\mathrm{eg})$ *is closed under intersection with n-fold cartesian products of regular languages.*

Proof. (1) Let $V = \mathbf{coll}(L)$ where L is a regular n-language. Let $\Phi = \varphi_1 \times \cdots \times \varphi_n$ where $\varphi_1, \ldots, \varphi_n$ are codings. Then $\varphi_1^{-1}, \ldots, \varphi_n^{-1}$ are finite substitutions mapping letters to finite sets of letters. If $(b_1, \ldots, b_n)$ is a vector letter, then clearly $\Phi^{-1}((b_1, \ldots, b_n)) = \{(c_1, \ldots, c_n) \mid \varphi_i(c_i) = b_i,\ i \in [n]\}$ is a finite set of vector letters. For vector letters $\beta = (b_1, \ldots, b_n)$ and $\gamma = (c_1, \ldots, c_n)$, we have that $\Phi^{-1}(\beta \odot \gamma) = \{(w_1, \ldots, w_n) \mid \varphi_i(w_i) = b_i c_i,\ i \in [n]\} = \Phi^{-1}(\beta) \odot \Phi^{-1}(\gamma)$. Thus, it follows that $\Phi^{-1}(\mathbf{coll}(L)) = \mathbf{coll}(\Phi^{-1}(L))$. Since the family of regular languages is closed under finite substitutions, it follows that $\Phi^{-1}(V) \in \mathbb{V}(n\text{-}\mathbb{R}\mathrm{eg})$.

(2) Follows from $\mathbb{V}(\mathbb{R}\mathrm{eg}; n\text{-}\mathbb{R}\mathrm{eg}) = \mathbb{V}(n\text{-}\mathbb{R}\mathrm{eg})$ (see Lemma 2.2.2(3)). □

Corollary 5.1.9. (1) $n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathbb{A}\mathrm{ll}; n\text{-}\mathbb{R}\mathrm{eg})) = n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathbb{R}\mathrm{eg}))$.

(2) $n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathbb{A}\mathrm{ll}; n\text{-}\lambda\mathbb{R}\mathrm{eg})) = n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\lambda\mathbb{R}\mathrm{eg}))$.

(3) $n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathbb{A}\mathrm{ll}; n\text{-}\mathrm{c}\mathbb{R}\mathrm{eg})) = n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathrm{c}\mathbb{R}\mathrm{eg}))$.

Proof. The inclusions "$\supseteq$" follow directly from the results presented in Fig. 2.

The converse inclusions "$\subseteq$" can be deduced from Lemma 5.1.7 and Proposition 5.1.8 in the following way. Clearly, $\mathbb{A}\mathrm{ll}$ is closed under inverse codings, under intersection with regular languages and under **pref**. Proposition 5.1.8 shows that $\mathbb{V}(n\text{-}\mathbb{R}\mathrm{eg})$ satisfies the required closure properties. From the basic properties of completeness and Λ-completeness – marked ($*$) and ($**$) in Section 2.1 – it follows that $\mathbb{V}(n\text{-}\lambda\mathbb{R}\mathrm{eg})$ and $\mathbb{V}(n\text{-}\mathrm{c}\mathbb{R}\mathrm{eg})$ are closed under intersection with n-fold cartesian product of regular languages. Finally, since inverse multi codings do not alter the Λ-structure of word vectors, Proposition 5.1.8 also proves that $\mathbb{V}(n\text{-}\lambda\mathbb{R}\mathrm{eg})$ and $\mathbb{V}(n\text{-}\mathrm{c}\mathbb{R}\mathrm{eg})$ are closed under inverse n-dimensional multi codings.

The conditions of Lemma 5.1.7 are satisfied and hence the inclusions "$\subseteq$" can be inferred from this lemma, using the idempotency of multi codings. □

In step V, it is shown that prefix-closedness of control languages can also be "overcome" with the help of multi codings. In contrast to Lemma 5.1.7, we end up in this case with a Λ-complete vector language, which is caused by the fact that prefix-closedness implies Λ-completeness and the fact that multi codings preserve Λ-completeness.

Lemma 5.1.10. $\mathbb{V}(\mathbb{K}; n\text{-}\lambda\mathbb{R}\mathrm{eg}) \subseteq n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathbb{K}; n\text{-}\mathrm{p}\mathbb{R}\mathrm{eg}))$ *provided the family* $\mathbb{K}$ *is closed under inverse codings and under intersection with regular languages.*

Proof. Let $K_i \in \mathbb{K}$ for all $i \in [n]$ and let $M \in \mathbb{L}(n\text{-}\lambda\mathbb{R}\mathrm{eg})$. We construct, for all $i \in [n]$, a language $K_i' \in \mathbb{K}$, a language $M' \in \mathbb{L}(n\text{-}\mathrm{p}\mathbb{R}\mathrm{eg})$, and a multi coding $\Phi \in n\text{-}\mathbb{MC}\mathrm{od}$, such that $(\bigtimes_{i=1}^{n} K_i) \cap \mathbf{coll}(M) = \Phi[(\bigtimes_{i=1}^{n} K_i') \cap \mathbf{coll}(M')]$.

As in the proof of Lemma 5.1.7 we use here endmarkings to distinguish original words from prefixes. Now, however, we have to be more careful, because we are dealing with more dimensions. We concentrate first on the construction of M' from M and its properties.

Let Σ be an alphabet such that $K_i \subseteq \Sigma^*$, for all $i \in [n]$, and such that $M \subseteq \mathbf{Tot}(\Sigma, \ldots, \Sigma)^*$. Let $\hat{\Sigma} = \{\hat{\sigma} \mid \sigma \in \Sigma\}$ be a disjoint copy of Σ. Let $\varphi_i : (\Sigma \cup \hat{\Sigma})^* \to \Sigma^*$ be the coding defined by $\varphi_i(\hat{b}) = \varphi_i(b) = b$, for all $i \in [n]$ and all $b \in \Sigma$, and let $\Phi = \varphi_1 \times \cdots \times \varphi_n$. Consider the vector language $V = \Phi^{-1}(\mathbf{coll}(M)) \cap \mathsf{X}_{i=1}^{n}(\Sigma^* \hat{\Sigma} \cup \{\Lambda\})$, i.e. V corresponds to the vector language $\mathbf{coll}(M)$ in which each non-empty component of each word vector has been endmarked. It is not difficult to see that $V \in \mathbb{V}(n\text{-}\mathbb{R}\mathrm{eg})$, because $\mathbb{V}(n\text{-}\mathbb{R}\mathrm{eg})$ is closed under inverse multi codings and under intersection with cartesian product of regular languages (see Proposition 5.1.8).

From Lemma 2.6 of [18] and its proof it follows that we can find an $M'' = \mathbb{L}(n\text{-}\mathbb{R}\mathrm{eg})$ with $\mathbf{coll}(M'') = V$, which satisfies the property that

$$ubcw \in M'' \text{ implies } b \text{ and } c \text{ are dependent,} \qquad (\dagger)$$

for any words u, w and any vector letters b and c. Thus, M'' is a language corresponding to the endmarked V and satisfying the special property ($\dagger$).

Let $M' = \mathbf{pref}(M'')$. The $M' \in \mathbb{L}(n\text{-p}\mathbb{R}\mathrm{eg})$ and – by property ($\dagger$) – M' satisfies, as we prove next, the property

$$\mathop{\mathsf{X}}_{i=1}^{n}(\Sigma^* \hat{\Sigma} \cup L_i) \cap \mathbf{coll}(M') = \mathbf{coll}(M''), \qquad (\ddagger)$$

where $L_i = \{\Lambda\}$ if $\Lambda \in \mathbf{proj}_i(\mathbf{coll}(M))$ and $L_i = \emptyset$ otherwise, for all $i \in [n]$. Thus non-empty proper prefixes of M'' can be distinguished from original words from M'' as intended. Property ($\dagger$) is necessary, since in general a proper 'prefix' of an endmarked word vector from $\mathbf{coll}(M'')$ can have the same form as an original endmarked word vector from $\mathbf{coll}(M'')$ in which some components are empty.

The proof of property ($\ddagger$) is as follows:

"$\supseteq$" follows directly from $M'' \subseteq M'$ and $\mathbf{coll}(M'') \subseteq (\mathsf{X}_{i=1}^{n}(\Sigma^* \hat{\Sigma} \cup L_i))$.

"$\subseteq$" Let $w \in \mathsf{X}_{i=1}^{n}(\Sigma^* \hat{\Sigma} \cup L_i) \cap \mathbf{coll}(M')$. Then there exist words v, v' such that $\mathbf{coll}(v) = w$ and $vv' \in M''$. Now v and v' must be independent, because if, for any $i \in [n]$, $\mathbf{proj}_i(v) \neq \Lambda$, then $\mathbf{proj}_i(v) \in \Sigma^* \hat{\Sigma}$ – because $v \in \mathsf{X}_{i=1}^{n}(\Sigma^* \hat{\Sigma} \cup L_i)$. Since also $\mathbf{proj}_i(vv') \in \Sigma^* \hat{\Sigma}$ – because $vv' \in M''$ – it follows that $\mathbf{proj}_i(v') = \Lambda$ for this i. By property ($\dagger$), the independence of v and v' implies that either $v = \Lambda$ or $v' = \Lambda$.
If $v = \Lambda$ and hence $w = \bar{\Lambda}$, then, for all $i \in [n]$, $L_i = \{\Lambda\}$. This implies, by the Λ-completeness of $\mathbf{coll}(M)$, that $\bar{\Lambda} \in \mathbf{coll}(M)$ and hence $w = \bar{\Lambda} \in \mathbf{coll}(M'')$. If $v' = \Lambda$, then $v = vv' \in M''$ and thus $w \in \mathbf{coll}(M'')$.

This proves the relation between M' and M'' formulated above.

Finally, we define, for all $i \in [n]$, $K_i' = \{w\hat{b} \mid wb \in K_i,\ w \in \Sigma^*,\ b \in \Sigma\} \cup (L_i \cap K_i) = \varphi_i^{-1}(K_i) \cap (\Sigma^* \hat{\Sigma} \cup L_i)$. By the closure properties of $\mathbb{K}$, $K_1', \ldots, K_n' \in \mathbb{K}$.

Now,

$$\begin{aligned}\mathop{\mathsf{X}}_{i=1}^{n} K_i \cap \mathbf{coll}(M) &= \mathop{\mathsf{X}}_{i=1}^{n} K_i \cap \Phi[\mathbf{coll}(M'')] \quad \text{(by the function-intersection rule)}\\ &= \mathop{\mathsf{X}}_{i=1}^{n} K_i \cap \Phi\left[\mathop{\mathsf{X}}_{i=1}^{n}(\Sigma^* \hat{\Sigma} \cup L_i) \cap \mathbf{coll}(M')\right]\end{aligned}$$

$$= \Phi\left[\Phi^{-1}\left(\bigtimes_{i=1}^{n} K_i\right) \cap \left(\bigtimes_{i=1}^{n}(\Sigma^*\hat{\Sigma} \cup L_i)\right) \cap \mathbf{coll}(M')\right]$$

(again by the function-intersection rule)

$$= \Phi\left[\bigtimes_{i=1}^{n} K_i' \cap \mathbf{coll}(M')\right],$$

and we are done. □

Corollary 5.1.11. (1) n-$\mathbb{MC}$od$(\mathbb{V}(\mathbb{A}$ll$; n$-$\lambda\mathbb{R}$eg$))$ = n-$\mathbb{MC}$od$(\mathbb{V}(\mathbb{A}$ll$; n$-p$\mathbb{R}$eg$))$.

(2) n-$\mathbb{MC}$od$(\mathbb{V}(\mathbb{R}$eg$; n$-$\lambda\mathbb{R}$eg$))$ = n-$\mathbb{MC}$od$(\mathbb{V}(\mathbb{R}$eg$; n$-p$\mathbb{R}$eg$))$.

Proof. The inclusions "$\supseteq$" follow directly from the results presented in Fig. 2.

The families $\mathbb{A}$ll and $\mathbb{R}$eg satisfy the closure properties of Lemma 5.1.10 and hence the inclusions "$\subseteq$" follow from this lemma, using the idempotency of multi codings. □

Our last main step focusses on monoid control languages.

The following lemma is a generalization of a result from [16], mentioned in Section 4. It is based on the theory of Individual Token Net Controllers developed in that paper, and on the characterization of the vector languages of ITNCs given in [18].

Lemma 5.1.12. (1) $\mathbb{V}(\mathbb{K}; n$-c$\mathbb{R}$eg$) \subseteq n$-$\mathbb{MC}$od$(\mathbb{V}(\mathbb{K}; n$-$\mathbb{M}on))$ *provided the family $\mathbb{K}$ of languages is closed under inverse codings and under intersection with regular languages.*

(2) $\mathbb{V}(\mathbb{K}; n$-cp$\mathbb{R}$eg$) \subseteq n$-$\mathbb{MC}$od$(\mathbb{V}(\mathbb{K}; n$-$\mathbb{M}on))$ *provided the family $\mathbb{K}$ of languages is closed under inverse codings and under intersection with prefix-closed regular languages.*

The proof of Lemma 5.1.12 makes use of the following auxiliary lemma.

Lemma 5.1.13. (1) *Let $V \in \mathbb{V}(n$-c$\mathbb{R}$eg$)$. Then there exist regular languages $R_1, \ldots, R_n$, a vector alphabet Γ, and a multi-coding Φ, such that $V = \Phi[\bigtimes_{i=1}^{n} R_i \cap \Gamma^{\circledast}]$.*

(2) *Let $V = \mathbb{V}(n$-cp$\mathbb{R}$eg$)$. Then there exist prefix-closed regular languages $R_1, \ldots, R_n$, a vector alphabet Γ, and a multi-coding Φ, such that $V = \Phi[\bigtimes_{i=1}^{n} R_i \cap \Gamma^{\circledast}]$.*

The proof of this lemma uses some results from [18]. The notations used there however relate to different underlying concepts. In order to make that paper more accessible as a reference, we briefly explain the relations between the different notations.

In [18] the family of n-dimensional rational relations is denoted by n-$\mathbb{R}$at. As already stated in Section 2 of this paper and as also observed in Section 2 of [18] we have n-$\mathbb{R}$at $= \mathbb{V}(n$-$\mathbb{R}$eg$)$.

The notation n-$\mathbb{CP}$ in [18] is used for the family of all n-dimensional complete vector languages. Thus, $\mathbb{V}(n$-c$\mathbb{R}$eg$) = \mathbb{V}(n$-$\mathbb{R}$eg$) \cap n$-$\mathbb{CP}$, and $\mathbb{V}(n$-cp$\mathbb{R}$eg$) = \mathbb{V}(n$-p$\mathbb{R}$eg$) \cap n$-$\mathbb{CP}$.

In [18], the families $\mathbb{V}(n\text{-GITNC})$ and $\mathbb{V}(n\text{-ITNC})$ denote the families of n-dimensional vector languages of, respectively, Generalized ITNCs and (ordinary) ITNCs. Similarly, the families $\mathbb{V}(n\text{-pGITNC})$ and $\mathbb{V}(n\text{-pITNC})$ denote the families of n-dimensional vector languages of, respectively, Generalized ITNCs with prefix-closed languages and ITNCs with prefix-closed languages. (For the purpose of this paper it is not necessary to precisely describe the ITNC and generalized ITNC models.)

Proof of Lemma 5.1.13. (1) Let $V \in \mathbb{V}(n\text{-}\mathbb{R}\text{eg})$. From Theorems 2.10 and 3.29 of [18], it follows that $V = \mathbf{coll}(L)$ for an ITNC language L (see also the remark in Section 2). Let θ be a vector alphabet such that $L \subseteq \theta^*$.

From Lemmas A.3 and A.7 of [16], it follows that there exist a regular language F over an alphabet T – i.e. $F \subseteq T^*$-, a coding $\varphi : T^* \to \theta^*$, and regular languages $F_1,\dots,F_n$ over T, such that $L = \varphi(F)$ and such that $F = \{w \in T^* \mid \mathbf{pres}_{T_i}(w) \in F_i$ for all $i \in [n]\}$, where $T_i = \{t \in T \mid \mathbf{proj}_i(\varphi(t)) \neq \Lambda\}$. Here $\mathbf{pres}_{T_i}$ is the weak coding defined by $\mathbf{pres}_{T_i}(t) = t$ if $t \in T_i$ and $\mathbf{pres}_{T_i}(t) = \Lambda$ otherwise. In terms of [18] F is a set of firing sequences of an ITNC and the associated L is the set of labelled firing sequences of this ITNC, i.e. the ITNC language.

Let

$$V' = \left\{ \begin{pmatrix} \mathbf{pres}_{T_1}(w) \\ \vdots \\ \mathbf{pres}_{T_n}(w) \end{pmatrix} \middle| \, w \in F \right\}, \qquad \Gamma = \left\{ \begin{pmatrix} \mathbf{pres}_{T_1}(t) \\ \vdots \\ \mathbf{pres}_{T_n}(t) \end{pmatrix} \middle| \, t \in T \right\},$$

and $R_i = \mathbf{pres}_{T_i}(F)$ for all $i \in [n]$. We prove that $V' = \mathsf{X}_{i=1}^n R_i \cap \Gamma^{\circledast}$ as follows.

The inclusion $V' \subseteq \mathsf{X}_{i=1}^n R_i \cap \Gamma^{\circledast}$ is easily shown, so we only need to prove the reverse inclusion. To that aim, let $(v_1,\dots,v_n) \in \mathsf{X}_{i=1}^n R_i \cap \Gamma^{\circledast}$. Then, there is an $m \geqslant 0$, and $t_1 \dots t_m \in T$, such that $v_i = \mathbf{pres}_{T_i}(t_1 \dots t_m)$ for all $i \in [n]$. Furthermore, since $\mathbf{pres}_{T_i}(F) \subseteq F_i$, for all $i \in [n]$, we have that $v_i \in R_i$ implies that $v_i \in F_i$ for all $i \in [n]$. From the relation between F and the F_i given above it follows that $t_1 \dots t_m \in F$, and hence $(v_1,\dots,v_n) \in V'$. This proves the inclusion and hence the equality $V' = \mathsf{X}_{i=1}^n R_i \cap \Gamma^{\circledast}$.

Finally, let $\Phi = \varphi_1 \times \cdots \times \varphi_n$, where $\varphi_i = \mathbf{proj}_i \circ \varphi$, for all $i \in [n]$. Then $V = \mathbf{coll}(L) = \mathbf{coll}(\varphi(F)) = \Phi[V'] = \Phi[\mathsf{X}_{i=1}^n R_i \cap \Gamma^{\circledast}]$, because

$$\begin{aligned} \mathbf{coll}(\varphi(w)) &= \begin{pmatrix} \mathbf{proj}_1 \circ \varphi(w) \\ \vdots \\ \mathbf{proj}_n \circ \varphi(w) \end{pmatrix} = \begin{pmatrix} \varphi_1(\mathbf{pres}_{T_1}(w)) \\ \vdots \\ \varphi_n(\mathbf{pres}_{T_n}(w)) \end{pmatrix} \\ &= \Phi\left(\begin{pmatrix} \mathbf{pres}_{T_1}(w) \\ \vdots \\ \mathbf{pres}_{T_n}(w) \end{pmatrix} \right) \quad \text{for all } w \in F. \end{aligned}$$

Note that the definition of T_i implies that $\varphi_i = \varphi_i \circ \mathbf{pres}_{T_i}$, for all $i \in [n]$.

(2) is proved analogously to (1). We only check the role of prefix-closedness here.

Firstly, by Theorems 4.7 and 4.10 of [18], it follows that any $V \in \mathbb{V}(n\text{-cp}\mathbb{R}\text{eg})$ is equal to $\mathbf{coll}(L)$ for a prefix-closed ITNC language L.

Secondly, by Remark 4.9(3) of [18], it follows that the associated regular language F can be assumed to be prefix–closed as well.

Finally, since the mapping $\mathbf{pres}_{T_i}$ are homomorphisms, also the languages $R_i = \mathbf{pres}_{T_i}(F)$ are prefix- closed. □

Proof of Lemma 5.1.12. (1) Let $V = (\mathsf{X}_{i=1}^{n} K_i) \cap U$, where $K_i \in \mathbb{K}$, for all $i \in [n]$, and $U \in \mathbb{V}(n\text{-c}\mathbb{R}\text{eg})$. By Lemma 5.1.13(1), there exist regular languages $R_1, \ldots, R_n$, a vector alphabet Γ, and a multi-coding Φ, such that $U = \Phi[\mathsf{X}_{i=1}^{n} R_i \cap \Gamma^{\circledast}]$. Set $\Phi = \varphi_1 \times \cdots \times \varphi_n$. Then $V = \Phi[\Phi^{-1}(\mathsf{X}_{i=1}^{n} K_i) \cap \mathsf{X}_{i=1}^{n} R_i \cap \Gamma^{\circledast}] = \Phi[\mathsf{X}_{i=1}^{n}(\varphi_i^{-1}(K_i) \cap R_i) \cap \Gamma^{\circledast}]$. From the closure properties of $\mathbb{K}$ it now follows directly that $V \in n\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{K}; n\text{-}\mathbb{M}\text{on})$, which concludes the proof.

(2) is proved analogously to (1). In this case the regular languages $R_1, \ldots, R_n$ are also prefix-closed and the closure properties of $\mathbb{K}$ take this into account. □

Corollary 5.1.14. (1) $n\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{V}(\mathbb{A}\text{ll}; n\text{-c}\mathbb{R}\text{eg})) = n\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{V}(\mathbb{A}\text{ll}; n\text{-cp}\mathbb{R}\text{eg})) = n\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{V}(\mathbb{A}\text{ll}; n\text{-}\mathbb{M}\text{on}))$.

(2) $n\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{V}(\mathbb{R}\text{eg}; n\text{-c}\mathbb{R}\text{eg})) = n\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{V}(\mathbb{R}\text{eg}; n\text{-cp}\mathbb{R}\text{eg})) = n\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{V}(\mathbb{R}\text{eg}; n\text{-}\mathbb{M}\text{on}))$.

(3) $n\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{V}(\text{p}\mathbb{A}\text{ll}; n\text{-cp}\mathbb{R}\text{eg})) = n\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{V}(\text{p}\mathbb{A}\text{ll}; n\text{-}\mathbb{M}\text{on}))$.

(4) $n\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{V}(\text{p}\mathbb{R}\text{eg}; n\text{-cp}\mathbb{R}\text{eg})) = n\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{V}(\text{p}\mathbb{R}\text{eg}; n\text{-}\mathbb{M}\text{on}))$.

Proof. The inclusions "$\supseteq$" follow directly from the results presented in Fig. 2.

The converse inclusions "$\subseteq$" – between the leftmost and rightmost parts – follow from Lemma 5.1.12, using the idempotency of multi codings.

(1) and (2) follow from Lemma 5.1.12(1), because $\mathbb{A}$ll and $\mathbb{R}$eg satisfy the required closure properties.

(3) and (4) follow from Lemma 5.1.12(2), because p$\mathbb{A}$ll and p$\mathbb{R}$eg satisfy the required closure properties. □

5.2. *Hiding*

Again we first present the operation diagram. In Fig. 5 the results are summarized for hid for $n \geqslant 2$, i.e. the graph of the relation $\mathbb{V} = \mathbf{hid}(\mathbb{V}')$ for $(n+1)$-dimensional $\mathbb{V}'$ is depicted. As before for $n = 1$, the results can be derived from the diagram for $n \geqslant 2$ by using the additional equalities holding for $n = 1$.

Again the arrows are marked with the numbers of the corresponding results. Note that we have been a little sloppy as we have ignored in the diagram the change in dimension caused by **hid**.

In spite of the fact that **hid** is not a closure operator for families of vector languages of a fixed dimension, the following lemma shows we have closure results when disregarding dimensions. The five self-loops in Fig. 5 reflect this idea.

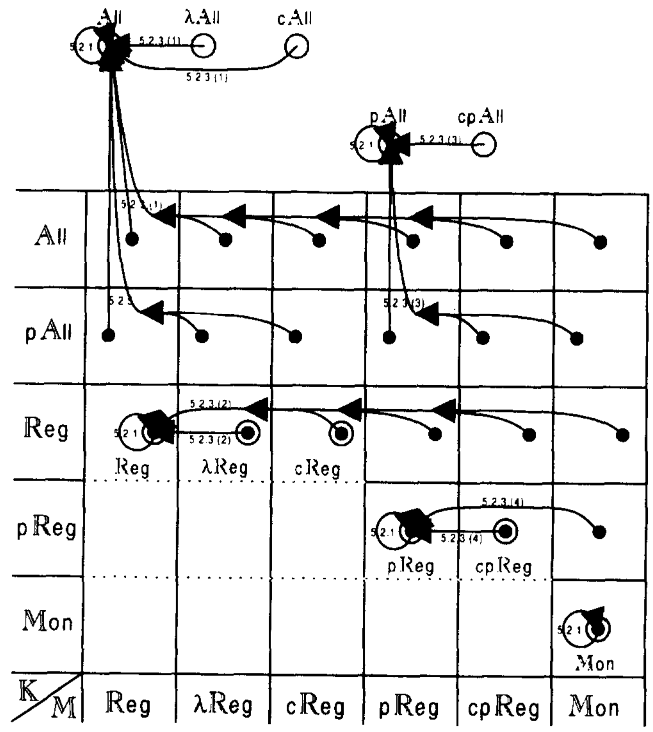

Fig. 5. Operation diagram of **hid** for the families $\mathbb{V}(\mathbb{K};n\text{-}\mathbb{M})$ and $\mathbb{V}(n\text{-}\mathbb{M})$ for $n \geqslant 3$.

Lemma 5.2.1. $\mathbb{V}(n\text{-}\mathbb{M}) = \mathbf{hid}(\mathbb{V}((n+1)-\mathbb{M}))$ *for all* $\mathbb{M} \in \{\mathbb{A}\text{ll}, \text{p}\mathbb{A}\text{ll}, \mathbb{R}\text{eg}, \text{p}\mathbb{R}\text{eg}, \mathbb{M}\text{on}\}$.

Proof. "$\subseteq$": Follows from the fact that the *Λ-extension* mapping (mapping n-dimensional vector letters $(a_1,\ldots,a_n)$ to $(n+1)$-dimensional vector letters $(a_1,\ldots,a_n,\Lambda)$) is an injective coding, and the observation that $\mathbb{A}$ll, p$\mathbb{A}$ll, $\mathbb{R}$eg, p$\mathbb{R}$eg, and $\mathbb{M}$on are closed under injective codings,

"$\supseteq$": Follows from the fact that **hid** is in essence a weak coding when restricted to vector letters. Thus the inclusion results follow, from the closure under weak codings of the families $\mathbb{A}$ll, p$\mathbb{A}$ll, $\mathbb{R}$eg, p$\mathbb{R}$eg, and $\mathbb{M}$on. □

The following general lemma is the basis of most of the results concerning hiding. Its proof uses "transfer" of control from the control language to the hidden last component.

Lemma 5.2.2. $\mathbb{V}(n\text{-}\mathbb{M}) \subseteq \mathbf{hid}(\mathbb{V}(\mathbb{M};(n+1)\text{-}\mathbb{M}\text{on}))$ *provided the family of languages* $\mathbb{M}$ *contains* $\mathbb{M}$on.

Proof. Let $V = \mathbf{coll}(L)$ where $L \in \mathbb{L}(n\text{-}\mathbb{M})$ and let θ be a vector alphabet such that $L \subseteq \theta^*$. Let Ξ be the $(n+1)$-dimensional vector alphabet $\{(\mathbf{proj}_1(\vartheta),\ldots,\mathbf{proj}_n(\vartheta),\vartheta) \mid \vartheta \in \theta\}$. Then, with $K_{n+1} = L$ and $K_i = \mathbf{proj}_1(\theta)^*$ for all $i \in [n]$, we have that

$V = \mathbf{hid}((\mathsf{X}_{i=1}^{n+1}K_i) \cap \Xi^{\circledast})$. Since $L \in \mathbb{M}$ and $\Xi^{\circledast} \in \mathbb{V}((n+1)\text{-}\mathbb{M}\mathrm{on})$, this proves the lemma. □

Corollary 5.2.3. *Let* $n \geqslant 2$.

$$(1)\quad \begin{aligned} \mathbf{hid}(\mathbb{V}(n\text{-}\mathbb{A}\mathrm{ll})) &= \mathbf{hid}(\mathbb{V}(n\text{-}\lambda\mathbb{A}\mathrm{ll})) = \mathbf{hid}(\mathbb{V}(n\text{-c}\mathbb{A}\mathrm{ll})) \\ &= \mathbf{hid}(\mathbb{V}(\mathbb{A}\mathrm{ll}; n\text{-}\mathbb{M})) \\ &\qquad \textit{for all } \mathbb{M} \in \{\mathbb{R}\mathrm{eg}, \lambda\mathbb{R}\mathrm{eg}, \mathrm{c}\mathbb{R}\mathrm{eg}, \mathrm{p}\mathbb{R}\mathrm{eg}, \mathrm{cp}\mathbb{R}\mathrm{eg}, \mathbb{M}\mathrm{on}\}. \end{aligned}$$

$$(2)\quad \begin{aligned} \mathbf{hid}(\mathbb{V}(n\text{-}\mathbb{R}\mathrm{eg})) &= \mathbf{hid}(\mathbb{V}(n\text{-}\lambda\mathbb{R}\mathrm{eg})) = \mathbf{hid}(\mathbb{V}(n\text{-c}\mathbb{R}\mathrm{eg})) \\ &= \mathbf{hid}(\mathbb{V}(\mathbb{R}\mathrm{eg}; n\text{-}\mathbb{M})) \\ &\qquad \textit{for all } \mathbb{M} \in \{\mathbb{R}\mathrm{eg}, \lambda\mathbb{R}\mathrm{eg}, \mathrm{c}\mathbb{R}\mathrm{eg}, \mathrm{p}\mathbb{R}\mathrm{eg}, \mathrm{cp}\mathbb{R}\mathrm{eg}, \mathbb{M}\mathrm{on}\}. \end{aligned}$$

$$(3)\quad \begin{aligned} \mathbf{hid}(\mathbb{V}(n\text{-p}\mathbb{A}\mathrm{ll})) &= \mathbf{hid}(\mathbb{V}(n\text{-cp}\mathbb{A}\mathrm{ll})) \\ &= \mathbf{hid}(\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathbb{M})) \\ &\qquad \textit{for all } \mathbb{M} \in \{\mathrm{p}\mathbb{R}\mathrm{eg}, \mathrm{cp}\mathbb{R}\mathrm{eg}, \mathbb{M}\mathrm{on}\}. \end{aligned}$$

$$(4)\quad \begin{aligned} \mathbf{hid}(\mathbb{V}(n\text{-p}\mathbb{R}\mathrm{eg})) &= \mathbf{hid}(\mathbb{V}(n\text{-cp}\mathbb{R}\mathrm{eg})) \\ &= \mathbf{hid}(\mathbb{V}(\mathrm{p}\mathbb{R}\mathrm{eg}; n\text{-}\mathbb{M})) \\ &\qquad \textit{for all } \mathbb{M} \in \{\mathrm{p}\mathbb{R}\mathrm{eg}, \mathrm{cp}\mathbb{R}\mathrm{eg}, \mathbb{M}\mathrm{on}\}. \end{aligned}$$

For the remaining cases we use the following lemma.

Lemma 5.2.4. $\mathbb{V}(n\text{-}\mathbb{M}) \subseteq \mathbf{hid}(\mathbb{V}(\mathrm{p}\mathbb{M}; (n+1)\text{-c}\mathbb{R}\mathrm{eg}))$ *provided the family of languages* $\mathbb{M}$ *is closed under inverse codings, under intersection with regular languages, under* **pref**, *and contains* $\mathbb{M}\mathrm{on}$.

Proof. Let $V = \mathbf{coll}(L)$ where $L \in \mathbb{L}(n\text{-}\mathbb{M})$ and let θ be a vector alphabet such that $L \subseteq \theta^*$. Let $\hat{\theta} = \{\hat{\vartheta} \mid \vartheta \in \theta\}$ be a disjoint copy of θ. Let $K_{n+1} = \mathbf{pref}(\{w\hat{b} \mid wb \in L\}) = \mathbf{pref}(\varphi^{-1}(L) \cap (\theta^*\hat{\theta} \cup \{\Lambda\}))$, where φ is the coding defined by $\varphi(\hat{\vartheta}) = \varphi(\vartheta) = \vartheta$, for all $\vartheta = \theta$. Set $K_i = \mathbf{proj}_i(\theta)^*$ for all $i \in [n]$. Let $\Xi = \{(\mathbf{proj}_1(\vartheta), \ldots, \mathbf{proj}_n(\vartheta), \vartheta) \mid \vartheta \in \theta\}$ and let $\hat{\Xi} = \{(\mathbf{proj}_1(\vartheta), \ldots, \mathbf{proj}_n(\vartheta), \hat{\vartheta}) \mid \vartheta \in \theta\}$. Set $M = \Xi^*\hat{\Xi} \cup N$ where $N = \{\Lambda\}$ if $\bar{\Lambda} \in V$ and $N = \emptyset$ otherwise. Then $V = \mathbf{hid}((\mathsf{X}_{i=1}^{n+1}K_i) \cap \mathbf{coll}(M))$, which follows directly from the observation that $\mathbf{proj}_{n+1}(M) \cap K_{n+1} = (\theta^*\hat{\theta} \cup N) \cap K_{n+1} = L$ and the definitions of Ξ and $\hat{\Xi}$. Now $\mathbf{coll}(M)$ is complete, because $\mathbf{proj}_{n+1}(w) \neq \Lambda$ for all non-empty $w \in M$. Hence, $\mathbf{coll}(M) \in \mathbb{V}((n+1)\text{-c}\mathbb{R}\mathrm{eg})$, because clearly $M \in \mathbb{L}((n+1)\text{-}\mathbb{R}\mathrm{eg})$. Combined with the fact that $K_i \in \mathrm{p}\mathbb{M}$, for all $i \in [n+1]$, this proves the lemma. □

Corollary 5.2.5. *Let* $n \geqslant 2$.

$$\mathbf{hid}(\mathbb{V}(n\text{-}\mathbb{A}\mathrm{ll})) = \mathbf{hid}(\mathbb{V}(p\mathbb{A}\mathrm{ll}; n\text{-}\mathbb{M})) \quad \textit{for all } \mathbb{M} \in \{\mathbb{R}\mathrm{eg}, \lambda\mathbb{R}\mathrm{eg}, \mathrm{c}\mathbb{R}\mathrm{eg}\}.$$

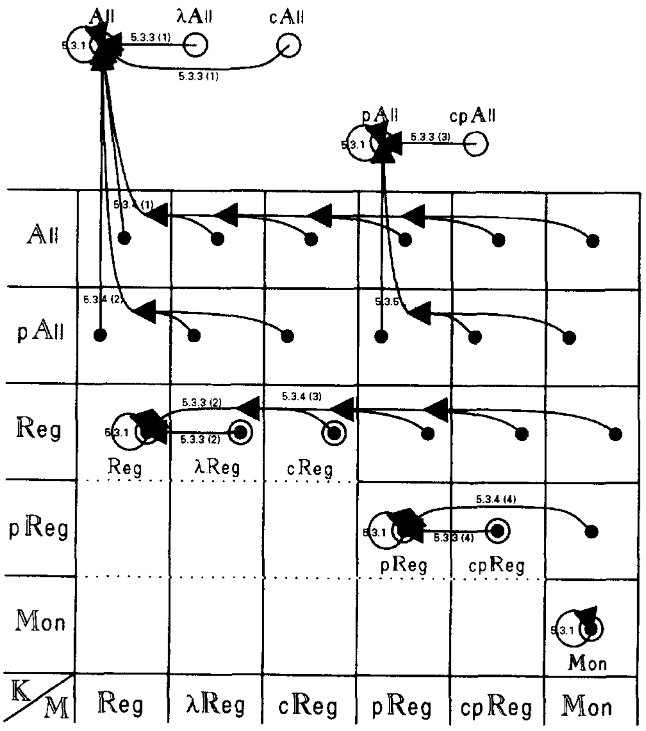

Fig. 6. Operation diagram of n-𝕄𝕎ℂod for the families 𝕍(𝕂; n-𝕄) and 𝕍(n-𝕄) for $n \geqslant 2$.

5.3. *Multi-weak codings*

In Fig. 6 we present an overview of the results of this section in the form of the operation diagram of n-𝕄𝕎ℂod for $n \geqslant 2$. Clearly, Figs. 5 and 6 are very similar. We come back to this in the discussion.

Again, the arrows are marked with the numbers of the corresponding results and for $n = 1$, the diagram can be obtained from the diagram for $n \geqslant 2$ using the additional equalities holding for $n = 1$ (see Fig. 3).

The operator n-𝕄𝕎ℂod is a closure operator and the families closed under n-𝕄𝕎ℂod correspond to the families closed for **hid**. This is shown in the following lemma.

Clearly, the families closed under multi weak codings are also closed under multi codings. However, not all of the families closed under multi codings are closed under multi weak codings.

Also in contrast to multi codings, *all* families of the form 𝕍(𝕂; n-𝕄)) from Fig. 2, yield under n-𝕄𝕎ℂod one of the families closed under multi weak codings.

Lemma 5.3.1. 𝕍(n-𝕄) = n-𝕄𝕎ℂod(𝕍(n-𝕄)) *for all* 𝕄 ∈ {𝔸ll, p𝔸ll, ℝeg, pℝeg, 𝕄on}.

Proof. "⊆": Obvious, since 𝕍 ⊆ n-𝕄𝕎ℂod(𝕍) for *any* family 𝕍 of n-dimensional vector languages.

"$\supseteq$": This follows directly from the fact that the families $\mathbb{A}$ll, p$\mathbb{A}$ll, $\mathbb{R}$eg, p$\mathbb{R}$eg, $\mathbb{M}$on are closed under weak codings. □

Together with the results concerned with multi codings, the following lemma forms the basis of most of the results concerned with multi weak codings.

Lemma 5.3.2. $\mathbb{V}(n\text{-}\mathbb{M}) \subseteq n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-c}\mathbb{M}))$ *provided the family of languages* $\mathbb{M}$ *is closed under injective codings.*

Proof. Let $\mathbb{V} = \mathbf{coll}(L)$ where $L \in \mathbb{L}(n\text{-}\mathbb{M})$. Let **eps** be the injective coding, mapping n-dimensional vector letters to n-dimensional vector letters, that replaces in every vector letter every Λ-component by the *new* letter ε (leaving the other components unchanged). Then $\mathbf{coll} \circ \mathbf{eps}(L) \in \mathbb{V}(n\text{-c}\mathbb{M})$, because $\bar{\Lambda}$ is the only possible word vector in $\mathbf{coll} \circ \mathbf{eps}(L)$ having an empty component. If Ψ is the n-dimensional weak coding that erases all occurrences of ε in word vectors, then $V = \Psi(\mathbf{coll} \circ \mathbf{eps}(L))$, which proves that $\mathbb{V}(n\text{-}\mathbb{M}) \subseteq n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-c}\mathbb{M}))$. □

Corollary 5.3.3.

(1) $n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-}\mathbb{A}\text{ll})) = n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-}\lambda\mathbb{A}\text{ll})) = n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-c}\mathbb{A}\text{ll}))$.

(2) $n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-}\mathbb{R}\text{eg})) = n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-}\lambda\mathbb{R}\text{eg})) = n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-c}\mathbb{R}\text{eg}))$.

(3) $n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-p}\mathbb{A}\text{ll})) = n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-cp}\mathbb{A}\text{ll}))$.

(4) $n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-p}\mathbb{R}\text{eg})) = n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-cp}\mathbb{R}\text{eg}))$.

The following corollary combines the previous one and some of the results of Section 5.1.

Corollary 5.3.4. (1) $n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-}\mathbb{A}\text{ll})) = n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(\mathbb{A}\text{ll}; n\text{-}\mathbb{M}))$ *for all* $\mathbb{M} \in \{\mathbb{R}\text{eg}, \lambda\mathbb{R}\text{eg}, \text{c}\mathbb{R}\text{eg}, \text{p}\mathbb{R}\text{eg}, \text{cp}\mathbb{R}\text{eg}, \mathbb{M}\text{on}\}$.

(2) $n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-}\mathbb{A}\text{ll})) = n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(\text{p}\mathbb{A}\text{ll}; n\text{-}\mathbb{M}))$ *for all* $\mathbb{M} \in \{\mathbb{R}\text{eg}, \lambda\mathbb{R}\text{eg}, \text{c}\mathbb{R}\text{eg}\}$.

(3) $n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-}\mathbb{R}\text{eg})) = n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(\mathbb{R}\text{eg}; n\text{-}\mathbb{M}))$ *for all* $\mathbb{M} \in \{\mathbb{R}\text{eg}, \lambda\mathbb{R}\text{eg}, \text{c}\mathbb{R}\text{eg}, \text{p}\mathbb{R}\text{eg}, \text{cp}\mathbb{R}\text{eg}, \mathbb{M}\text{on}\}$.

(4) $n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-p}\mathbb{R}\text{eg})) = n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(\text{p}\mathbb{R}\text{eg}; n\text{-}\mathbb{M}))$ *for all* $\mathbb{M} \in \{\text{p}\mathbb{R}\text{eg}, \text{cp}\mathbb{R}\text{eg}, \mathbb{M}\text{on}\}$.

The final results concerning multi weak codings are obtained using the following lemma.

Lemma 5.3.5. $n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-p}\mathbb{A}\text{ll})) = n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(\text{p}\mathbb{A}\text{ll}; n\text{-}\mathbb{M}))$ *for all* $\mathbb{M} \in \{\text{p}\mathbb{R}\text{eg}, \text{cp}\mathbb{R}\text{eg}, \mathbb{M}\text{on}\}$.

Proof. The inclusions "$\supseteq$" follow directly from the results presented in Fig. 2. The converse inclusions follow from the idempotency of multi weak codings and the fact that $\mathbb{V}(n\text{-p}\mathbb{A}\text{ll}) \subseteq n\text{-}\mathbb{MWC}\text{od}(\text{p}\mathbb{A}\text{ll}; n\text{-}\mathbb{M}\text{on}))$, which is proved as follows.

Let $V = \mathbf{coll}(L)$ where $L \in \mathbb{L}(n\text{-p}\mathbb{A}\mathrm{ll})$ and let θ be a vector alphabet such that $L \subseteq \theta^*$. Let Ξ be the n-dimensional vector alphabet $\{(\vartheta, \dots, \vartheta) \mid \vartheta \in \theta\}$ and let $M = \Xi^*$. Then, with $\Psi = \mathbf{proj}_{\Xi^*,1} \times \cdots \times \mathbf{proj}_{\Xi^*,n}$, we have that $V = \Psi((\mathrm{X}_{i=1}^n L) \cap \mathbf{coll}(M))$. Since $L \in \mathrm{p}\mathbb{A}\mathrm{ll}, \mathbf{coll}(M) \in \mathbb{V}(n\text{-}\mathbb{M}\mathrm{on})$, and $\Psi \in n\text{-}\mathbb{MWC}\mathrm{od}$, this proves the inclusion. $\square$

6. Discussion

Within the framework of Vector Controlled Concurrent Systems different models of concurrent systems based on vector synchronization can be studied in a uniform way. In [15, 16, 1] several such models have been investigated. Each of these models is formulated using regularity, prefix-closedness, or completeness as restrictions on the component languages or the control language. In this paper we have not singled out one or a few specific submodels as the focus of our interest. The aim here has been to investigate the effect of these restrictions and their combinations. This has led to a whole range of different VCCS submodels not all of which are individually interesting as a model for concurrent systems. Together, however, they lead to insight in the effect the restrictions have on the behaviour of the systems.

The effect of the restrictions has been measured in two different ways. Firstly, a direct comparison of the resulting behaviours in terms of families of vector languages has led to an inclusion diagram showing the equalities, strict inclusions and incomparabilities. Secondly, a more indirect approach has been followed. Each family of vector languages has been subjected to three types of operations with the aim of enlarging the family to one of the other families of the diagram. Each such representation of a larger family in terms of the smaller and one of the operations, corresponds to a characterization of the difference in effect between the combinations of restrictions involved.

In the inclusion diagram, the inclusions and equalities have been established on the basis of observations on the nature of the restrictions and on the interplay between component languages and control languages. For the non-inclusions and the incomparabilities a relatively small set of example vector languages has been used. Thus, here, it is efficient to consider the whole range of submodels rather than proving inequalities between certain specific submodels.

From the inclusion diagram we see that in a VCCS the control language has a strong influence on the resulting VCCS vector language. For VCCS families with a regular control language, regular, prefix-closed regular and monoidal component languages can be absorbed by the control: $\mathbb{V}(n\text{-}\mathbb{R}\mathrm{eg}) = \mathbb{V}(\mathbb{K}; n\text{-}\mathbb{R}\mathrm{eg})$ for $\mathbb{K} = \mathbb{R}\mathrm{eg}, \mathrm{p}\mathbb{R}\mathrm{eg}, \mathbb{M}\mathrm{on}$. On the other hand, even general component languages are *not* able to absorb the control language: $\mathbb{V}(\mathbb{A}\mathrm{ll}; n\text{-}\mathbb{R}\mathrm{eg}) \subsetneq \mathbb{V}(n\text{-}\mathbb{A}\mathrm{ll})$. Regular control languages cannot absorb arbitrary component languages: $\mathbb{V}(\mathbb{R}\mathrm{eg}; n\text{-}\mathbb{R}\mathrm{eg}) \subsetneq \mathbb{V}(\mathbb{A}\mathrm{ll}; n\text{-}\mathbb{R}\mathrm{eg})$. Similarly, prefix-closedness of the control language only guarantees a prefix-closed behaviour if also the component languages are prefix-closed: $\mathbb{V}(\mathrm{p}\mathbb{A}\mathrm{ll}; n\text{-}\mathbb{R}\mathrm{eg}) \subsetneq \mathbb{V}(n\text{-p}\mathbb{A}\mathrm{ll})$, but $\mathbb{V}(\mathbb{A}\mathrm{ll}; n\text{-p}\mathbb{R}\mathrm{eg})$ and $\mathbb{V}(n\text{-p}\mathbb{A}\mathrm{ll})$ are incomparable. Non-prefix-closed component languages again cannot fully overcome the prefix-closeness of the control language: $\mathbb{V}(\mathbb{R}\mathrm{eg}; n\text{-p}\mathbb{R}\mathrm{eg}) \subsetneq \mathbb{V}(n\text{-}\mathbb{R}\mathrm{eg})$.

Completeness is a property that is only applicable to control languages, though one could argue that the component languages are trivially complete. It leads to the completeness of the resulting vector languages: $\mathbb{V}(\mathbb{A}\text{ll}; n\text{-c}\mathbb{R}\text{eg}) \subseteq \mathbb{V}(n\text{-c}\mathbb{A}\text{ll})$.

To prove the representation results has required more technical effort. Of the three types of operations considered (multi codings, hiding, and multi weak codings) the multi codings are the weakest and, by the variety of results, also the most revealing. A number of interesting representation results have been obtained, using a variety of proof techniques. Still a number of generic results could be derived.

On the one hand, multi codings bridge the gap between $\mathbb{V}(\mathbb{A}\text{ll}; n\text{-}\mathbb{R}\text{eg})$ and $\mathbb{V}(n\text{-}\mathbb{A}\text{ll})$, thus enhancing the relative weakness of the component languages mentioned above. On the other hand, however, they are not sufficiently powerful to bridge a seemingly similar gap between the prefix-closed versions $\mathbb{V}(\text{p}\mathbb{A}\text{ll}; n\text{-p}\mathbb{R}\text{eg})$ and $\mathbb{V}(n\text{-p}\mathbb{A}\text{ll})$. A closer examination of the proofs of Lemmas 5.1.2 and 5.1.5 shows that here it is crucial that prefix-closedness basically is a *language* property instead of a *vector language* property.

In the cases that prefix-closedness is required only of the component languages or only of the control language, a multi coding can assist the absorption of this restriction. The basic techniques used to get these types of results are similar: an endmarking is used to distinguish proper prefixes from original words and a multi coding is used to remove the marking. However, the occurrence of the empty word in the control language in combination with the independent choice of component languages to include or exclude the empty word may prevent the full absorption of prefix-closedness. Then Λ-completeness, as a residual of prefix-closedness, is the best we can get.

Using a generic result (Lemma 5.1.12), the characterization $n\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{V}(\mathbb{R}\text{eg}; n\text{-}\mathbb{M}\text{on})) = \mathbb{V}(\mathbb{R}\text{eg}; n\text{-c}\mathbb{R}\text{eg}) = \mathbb{V}(n\text{-c}\mathbb{R}\text{eg})$ from [16] has been reproved, but now also its prefix-closed version could be shown to hold: $n\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{V}(\text{p}\mathbb{R}\text{eg}; n\text{-}\mathbb{M}\text{on})) = \mathbb{V}(\text{p}\mathbb{R}\text{eg}; n\text{-cp}\mathbb{R}\text{eg}) = \mathbb{V}(n\text{-cp}\mathbb{R}\text{eg})$. For the case of non-regular component languages, the same lemma has led to the equalities $n\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{V}(\mathbb{A}\text{ll}; n\text{-}\mathbb{M}\text{on})) = \mathbb{V}(n\text{-c}\mathbb{A}\text{ll})$ and $n\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{V}(\text{p}\mathbb{A}\text{ll}; n\text{-}\mathbb{M}\text{on})) = n\text{-}\mathbb{M}\mathbb{C}\text{od}(\mathbb{V}(\text{p}\mathbb{A}\text{ll}; n\text{-cp}\mathbb{R}\text{eg}))$. This confirms completeness as a characterizing property for the combination of monoidal control languages and multi codings.

Multi codings preserve completeness (and Λ-completeness), but both hiding and multi weak coding do *not* preserve these properties: for each family of vector languages the image under either multi weak codings or hidings is equal to the image of its complete (or Λ-complete) subfamily under this operation. Thus, for multi codings a larger number of closed families was obtained, than for the other two operations.

When comparing in Figs. 5 and 6 the effects of hidings and of multi weak codings, we see that – ignoring dimensions – they are essentially the same: in both situations the same five families are closed under the operation, and in both situations these families are precisely the families that we end up with when applying the operations to one of the other families. Moreover, similar representations are obtained in both situations.

For hiding, these representations results are obtained by the combination of a generic result (Lemma 5.2.2) applicable to most of the cases and a more specific result

(Lemma 5.2.4) for the remaining cases. The first result is based on a technique by which the role of the control languages is taken over by a component language. In fact, a similar situation exists for the multi weak codings, i.e. the role of the control language may be taken over by (one or more) component languages. We have not fully exploited this here however, because many of the multi weak coding results are already a consequence of multi coding results. Only for the remaining cases this technique was used (Lemma 5.3.5).

The gap left by multi codings when starting with a prefix closed control language – yielding Λ-completeness – is bridged by both hidings and multi weak codings: $\mathbf{hid}(\mathbb{V}(\mathbb{R}\mathrm{eg}; n\text{-p}\mathbb{R}\mathrm{eg})) = n\text{-}\mathbb{MWC}\mathrm{od}(\mathbb{V}(\mathbb{R}\mathrm{eg}; n\text{-p}\mathbb{R}\mathrm{eg})) = \mathbb{V}(n\text{-}\mathbb{R}\mathrm{eg})$, whereas $n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\mathbb{R}\mathrm{eg}; n\text{-p}\mathbb{R}\mathrm{eg})) = \mathbb{V}(n\text{-}\lambda\mathbb{R}\mathrm{eg}) \subsetneq \mathbb{V}(n\text{-}\mathbb{R}\mathrm{eg})$.

Hidings and multi weak codings also bridge the gaps between $\mathbb{V}(\text{p}\mathbb{A}\mathrm{ll}; n\text{-p}\mathbb{R}\mathrm{eg})$ and $\mathbb{V}(n\text{-p}\mathbb{A}\mathrm{ll})$ and between $\mathbb{V}(\text{p}\mathbb{A}\mathrm{ll}; n\text{-cp}\mathbb{R}\mathrm{eg})$ and $\mathbb{V}(n\text{-cp}\mathbb{A}\mathrm{ll})$ that could not be closed by multi codings.

When considering the three restrictions regularity, prefix-closedness, and completeness, we observe that both regularity and prefix-closedness are strong restrictions on the vector language level: all three operations preserve these. Completeness is weaker, because it is only preserved by multi codings.

When restrictions are used in only a part of a VCCS, i.e., only for component languages or only for control languages, then they can mostly be absorbed, though sometimes not fully. Using hidings and multi weak codings, the effect of these restrictions always disappears.

Completeness is a pure vector language property, regularity is a language property that can be extended to vector languages – in the form of rationality, whereas prefix-closedness is a pure language based property. For multi codings this distinction has turned out to be crucial.

The systematic approach of this paper has enabled us to not only repeat a number of results occurring in [1], it has also answered the open questions of [1] concerning the VCCS submodels included in our study. Thus, we have proved the strictness of the inclusions $\mathbb{V}(\text{p}\mathbb{R}\mathrm{eg}; n\text{-}\mathbb{M}\mathrm{on}) \subseteq \mathbb{V}(\text{p}\mathbb{R}\mathrm{eg}; n\text{-p}\mathbb{R}\mathrm{eg})$, and $n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\text{p}\mathbb{R}\mathrm{eg}; n\text{-}\mathbb{M}\mathrm{on})) \subseteq \mathbb{V}(\text{p}\mathbb{R}\mathrm{eg}; n\text{-p}\mathbb{R}\mathrm{eg})$. In fact, we have even obtained a characterization of the last strict inclusion, namely *completeness*: $n\text{-}\mathbb{MC}\mathrm{od}(\mathbb{V}(\text{p}\mathbb{R}\mathrm{eg}; n\text{-}\mathbb{M}\mathrm{on})) = \mathbb{V}(\text{p}\mathbb{R}\mathrm{eg}; n\text{-cp}\mathbb{R}\mathrm{eg}) \subsetneq \mathbb{V}(\text{p}\mathbb{R}\mathrm{eg}; n\text{-p}\mathbb{R}\mathrm{eg})$.

One of the models that has been studied in [1], but has not been considered here, is the COSY model. The main reason for this is that COSY does not fit in the set-up of this paper with its systematic combination of restrictions on independently defined component and control languages.

For component languages, COSY uses the additional and particular restriction of *cyclicity* in addition to regularity and prefix-closedness. Adding this extra restriction would have increased the set of models.

More importantly, COSY uses monoids as control languages and in addition imposes restrictions on the synchronization vectors themselves. This contrasts with the VCCS models studied in this paper: restrictions have only been used for component languages and control languages, never for their vector letters. A consequence of the additional vector letter restrictions in COSY systems is that behaviours of COSY systems are *not* closed under multi injective codings, whereas the VCCSs behaviours in this paper are. In other words, the component languages are mutually related in ways that go farther than in our VCCSs.

Next, the component languages are not defined independently of the control language of a COSY system. COSY systems demand that the set of letters occurring in component languages equals the set of letters occurring in the control languages.

Thus, COSY does not fit easily in the set-up of this paper. A preliminary study of the effect of the three operations on COSY systems given further proof of the special character of COSY vector languages. Let $\mathbb{V}(n\text{-COSY})$ denote the family of vector languages of n-dimensional COSY systems. It can be shown that $\mathbb{V}(n\text{-COSY}) \subsetneq \mathbf{hid}(\mathbb{V}(n+1\text{-COSY})) \subsetneq \mathbf{hid}(\mathbf{hid}(\mathbb{V}(n+2\text{-COSY}))) = \mathbf{hid}^3(\mathbb{V}(n+3\text{-COSY}))$. Thus, for COSY systems a single application of **hid** does *not* lead to a family closed under **hid**. For multi codings and multi weak codings the situation is also different: $n\text{-}\mathbb{MC}\text{od}(\mathbb{V}(n\text{-COSY})) = \mathbb{V}(n\text{-cp}\mathbb{R}\text{eg})$, and $n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(n\text{-COSY})) = \mathbb{V}(n\text{-p}\mathbb{R}\text{eg})$, for $n \geqslant 2$, but the family $\mathbb{V}(1\text{-COSY})$ is closed under multi codings, and multi weak codings: $n\text{-}\mathbb{MC}\text{od}(\mathbb{V}(1\text{-COSY})) = n\text{-}\mathbb{MWC}\text{od}(\mathbb{V}(1\text{-COSY})) = \mathbb{V}(1\text{-COSY})$, whereas $\mathbb{V}(1\text{-COSY}) \subsetneq \mathbb{V}(1\text{-p}\mathbb{R}\text{eg})$. Thus, for COSY systems, we get a decrease of the number of equalities for $n = 1$, instead of the usual increase.

We conclude this section by pointing out some topics for further research.

In the first place, as mentioned above, the COSY model has been been fully investigated, although we have obtained some results. A more thorough investigation of the COSY model within the VCCS framework may be worthwhile. The remaining open issues from [1] all concern the COSY model (and a variant). A deeper investigation of the COSY model along the lines of this paper may lead to answers to these open questions. Our preliminary investigations seem to confirm this idea.

As observed above, in a VCCS the control exercised by the control languages is rather strong. This had led us to exclude VCCSs with control languages from $\mathbb{L}(n\text{-}\mathbb{A}\text{ll}), \mathbb{L}(n\text{-c}\mathbb{A}\text{ll})$, or $\mathbb{L}(n\text{-p}\mathbb{A}\text{ll})$ from our investigations. Intuitively, VCCS submodels having such powerful families of control languages would hardly be interesting as the influence of additional component languages would probably be negligible.

In this paper our aim has been to study certain restrictions in the framework of VCCSs. Three specific operations have been used to investigate the effects of these restrictions. It is conceivable that other operations may also prove useful in these investigations. Dually, one could also argue that we have studied certain operations by applying these to different families of VCCS vector languages. From this point of view it may be worthwhile to consider more families of vector languages to which to apply the operations.

Both approaches may lead to further insight in the underlying fundamental properties of restrictions, operations, and their mutual relationships.

Acknowledgements

The authors are grateful to H.J. Hoogeboom for his useful comments on an earlier version of this paper. Also the comments of an anonymous referee which led to further improvements of the presentation are gratefully acknowledged.

References

[1] A. Arnold, Synchronized behaviours of processes and rational relations, *Acta Inform.* **17** (1982) 21–29.
[2] A. Arnold, Synchronization de processus, Université de Bordeaux I, Cours 83–84.
[3] J. Beauqier and M. Nivat, Application of formal language theory to problems of security and synchronization, in: R. Book, ed., *Formal Language Theory: Perspectives and Open Problems* (Academic Press, New York, 1980) 407–453.
[4] J. Berstel, *Transductions and Context-Free Lanauges* (Teubner, Stuttgart, 1979).
[5] L. Bernardinello and F. De Cindio, A survey of basic net models and modular net classes, Lecture Notes in Computer Science, Vol. 609 (Springer, Berlin, 1992) 304–351.
[6] L. Boasson and M. Nivat, Adherences of languages, *J. Comput. System Sci.* **20** (1980) 285–309.
[7] R.H. Campbell and A.N. Habermann, The specification of process synchronization by path expressions, Lecture Notes in Computer Science, Vol. 16 (Springer, Berlin, 1974) 89–102.
[8] N. Chomsky and M.P. Schützenberger, The algebraic theory of context-free languages, in: P. Bratfort and D. Hirschberg, eds., *Computer Programming and Formal Systems* (North-Holland, Amsterdam, 1963) 118–161.
[9] S. Greibach, The hardest CF language, *SIAM J. Comput.* **2** (1973) 304–310.
[10] M. Hack, Analysis of production schemata by Petri nets, TR-94, MIT, Boston, 1972.
[11] T. Harju and H.C.M. Kleijn, Morphisms and rational transducers, *EATCS Bull.* **51** (1993) 168–180.
[12] J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation* (Addison-Wesley, Reading, MA, 1979).
[13] R. Janicki and P.E. Lauer, *Specification and Analysis of Concurrent Systems, The COSY Approach,* EATCS Monographs on Theoretical Computer Science (Springer, Berlin, 1992).
[14] K. Jensen, *Coloured Petri Nets, Basic Concepts, Analysis Methods, and Practical Use, Vol. 1,* EATCS Monographs on Theoretical Computer Science (Springer, Berlin, 1992).
[15] N.W. Keesmaat, H.C.M. Kleijn and G. Rozenberg, Vector controlled concurrent systems, part I: basic classes, *Fund. Inform.* **13** (1990) 275–316.
[16] N.W. Keesmaat, H.C.M. Kleijn and G. Rozenberg, Vector controlled concurrent systems, part II: comparisons, *Fund. Inform.* **14** (1991) 1–38.
[17] N.W. Keesmaat and H.C.M. Kleijn, The effect of vector synchronization: residue and loss, Lecture Notes in Computer Science, Vol. 609 (Springer, Berlin, 1992) 215–250.
[18] N.W. Keesmaat and H.C.M. Kleijn, Net-based control versus rational control: the relation between ITNC vector languages and rational languages, *Acta Inform.*, to appear.
[19] P.E. Lauer and R.H. Campbell, Formal semantics for a class of high level primitives for coordinating processes, *Acta Inform.* **5** (1975) 297–332.
[20] M. Nivat, Behaviors of processes and synchronized systems of processes, in: M. Broy and G. Schmidt, eds., *Theoretical Foundations of Programming Methodology* (Reidel, Dordrecht, 1982) 473–551.
[21] W. Reisig, *Petri Nets, an Introduction,* EATCS Monographs on Theoretical Computer Science (Springer, Berlin, 1985).
[22] M.W. Shields, Adequate path expressions, Lecture Notes in Computer Science, Vol 70 (Springer, Berlin, 1979) 249–265.