

# Codes on Graphs: Observability, Controllability and Local Reducibility

G. David Forney, Jr. and Heide Gluesing-Luerssen

**Abstract**—This paper investigates properties of realizations of linear or group codes on general graphs that lead to local reducibility.

Trimness and properness are dual properties of constraint codes. A linear or group realization with a constraint code that is not both trim and proper is locally reducible. A linear or group realization on a finite cycle-free graph is minimal if and only if every local constraint code is trim and proper.

A realization is called observable if there is a one-to-one correspondence between codewords and configurations, and controllable if it has independent constraints. A linear or group realization is observable if and only if its dual is controllable. A simple counting test for controllability is given. An unobservable or uncontrollable realization is locally reducible. Parity-check realizations are controllable if and only if they have independent parity checks. In an uncontrollable tail-biting trellis realization, the behavior partitions into disconnected subbehaviors, but this property does not hold for non-trellis realizations. On a general graph, the support of an unobservable configuration is a generalized cycle.

## I. INTRODUCTION

The theory of minimal realizations of linear codes and systems on a conventional discrete time axis has long been well understood. In classical minimal realization theory, the concepts of observability and controllability are central; indeed, the classical mantra is “minimal = controllable + observable.”

Moreover, as is well known, classical minimal realization theory extends straightforwardly to realizations of linear (or abelian group) codes or systems on cycle-free graphs [4]. Given such a graph, the State Space Theorem defines a unique minimal “state space” for every internal (“state”) variable.

On the other hand, powerful codes that can approach capacity with iterative decoding (e.g., low-density parity-check (LDPC) or turbo codes) are necessarily defined on graphs that have cycles. Progress in extending minimal realization theory to general graphs with cycles has been slow. (Also, it is not clear that minimal realizations are necessarily the best ones for iterative decoding; see Section IV-H.)

A fundamental issue is that in general there is no unique minimal realization of a linear code or system on a graph that contains cycles. As was observed by Koetter and Vardy [18], even for the simplest such graph comprising a single cycle (called a “tail-biting trellis realization”), there are various

notions of minimality, and for any such notion there is only a partial ordering of tail-biting trellis realizations. However, Koetter and Vardy were able to specify a small, tractable set of possibly minimal tail-biting trellises, by showing that every linear tail-biting trellis realization factors into a product of one-dimensional “atomic” realizations [17] (as do conventional linear trellis realizations [19]).

On more general graphs, there is no unique minimal realization, and also no such product factorization (see Appendix A). How then can we pursue something like minimality?

The main idea of this paper is to look for possibilities for “locally” reducing the complexity of a realization—e.g., by reducing the size of one internal variable alphabet (“state space”)—without changing the code that is realized or the complexity of the rest of the realization. Such an approach follows a long tradition in the conventional trellis minimization literature (e.g., [20], [25]), and was previously explored by Koetter [16]. In Appendix B, we give a brief summary of Koetter’s pioneering work in the language of this paper.

For instance, a realization in which some state variable value never occurs is obviously locally reducible, as the state alphabet may be trimmed to the set of values that actually occur. But as Koetter [16] observed, duality considerations using the “dual realizations” of [4] can reveal less obvious dual local reducibility criteria. For example, we give a simple proof that a constraint code is not (state-)trim if and only if the dual constraint code is not “proper,” in a sense that extends the traditional definition of “properness” for trellis realizations [25]. By duality, any realization that involves an “improper” constraint code must therefore be locally reducible.

We next show that a finite linear or group normal realization on a cycle-free graph is minimal if and only if every constraint code is both trim and proper. Although this simple result has long been known for conventional trellis realizations [25], it seems to be new for realizations on more general cycle-free graphs. This development yields a constructive proof of the fundamental State Space Theorem, and leads immediately to straightforward iterative minimization algorithms.

Next, we consider linear or group realizations on general graphs that are not one-to-one—i.e., every codeword in  $\mathcal{C}$  is realized by multiple configurations. Such realizations are called *unobservable*, since the values of the internal variables are not determined by those of the external variables. Such realizations would seem to be obviously undesirable for iterative decoding; see Section IV-H.

We show that any unobservable realization is locally reducible. Therefore we may always assume that any linear or group realization on a general graph is observable.

The authors are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 (email: forneyd@comcast.net) and Department of Mathematics, University of Kentucky, Lexington KY 40506-0027 (email: heide.gl@uky.edu), respectively. H. Gluesing-Luerssen was partially supported by the National Science Foundation Grant DMS-0908379. This paper was presented in part at the 2012 Information Theory and Applications Workshop, San Diego, CA, and at the 2012 IEEE International Symposium on Information Theory, Cambridge, MA.

Following the principle that the dual of a locally reducible normal realization of  $\mathcal{C}$  is locally reducible, we show that the dual of an unobservable linear realization is a realization with dependent constraints. For consistency with classical terminology, we call such a realization *uncontrollable*. By duality, an uncontrollable realization is locally reducible.

However, it is not so clear that an uncontrollable realization is unsuitable for iterative decoding; see Section IV-H. For example, we show that any parity-check realization with redundant parity checks is uncontrollable; nevertheless, LDPC codes with redundant checks have sometimes been preferred in practice.

We show that an uncontrollable tail-biting trellis realization consists of disjoint subrealizations, as with classical uncontrollable conventional trellis realizations. However, realizations on general graphs do not in general have this property.

Finally, given an unobservable realization on a general graph, the support of the unobservable configurations must be a cycle or generalized cycle (a subgraph whose vertices all have degree 2 or more). In the dual uncontrollable realization, there exist nonlocal constraints that are defined on the same support. In the binary case, this support must actually be an Eulerian cycle (a generalized cycle whose vertices all have even degree).

We may summarize our progress on establishing conditions for local reducibility as follows. We partition the universe of linear or group codes on finite graphs into four quadrants, as shown in the  $2 \times 2$  matrix below, where one axis distinguishes trellis from non-trellis graphs, and one cycle-free from cyclic.

	trellis	non-trellis
cycle-free	conventional trellises	cycle-free graphs
cyclic	tail-biting trellises	general graphs

For conventional trellis realizations, simple conditions for minimality have long been known. In this paper, we find a simple, conclusive, and locally testable condition for minimality for a general finite linear or group cycle-free realization:

minimal  $\Leftrightarrow$  every constraint code is trim and proper.

For linear tail-biting trellis realizations, which in general have no unique minimal realization, we make some progress in this paper toward finding conditions for local irreducibility. In a subsequent paper [10], we will give necessary and sufficient conditions for a linear tail-biting trellis realization to be irreducible, under a more refined definition of local reducibility. These results largely settle the tail-biting trellis case, although we consider that there are still some open issues in this case.

Finally, for the general case, beyond trimness and properness, the dual concepts of observability and controllability yield further local reducibility criteria; however, this case remains largely open.

## II. CODES, REALIZATIONS, AND DUALITY

In this paper we will focus on linear codes over a finite field  $\mathbb{F}$ . Everything generalizes to group codes over finite abelian groups, and to linear systems over the real or complex field. We will state some of our most basic results for both the linear and group cases, since the linear case is often more familiar, whereas the group-theoretic statements and proofs are often more transparent.

### A. Codes and realizations

A linear code  $\mathcal{C}$  over a finite field  $\mathbb{F}$  is a linear subspace of a *symbol configuration space*  $\mathcal{A} = \prod_{k \in \mathcal{I}_{\mathcal{A}}} \mathcal{A}_k$ , where each *symbol alphabet*  $\mathcal{A}_k$  is a finite-dimensional vector space over  $\mathbb{F}$ , and  $\mathcal{I}_{\mathcal{A}}$  is a discrete index set. We will usually assume that  $\mathcal{I}_{\mathcal{A}}$  is finite (*i.e.*,  $\mathcal{C}$  is a block code); then the symbol configuration space  $\mathcal{A}$  is finite. We may alternatively refer to a symbol variable as an *external variable*  $A_k$ , which takes values  $a_k$  in an external variable alphabet  $\mathcal{A}_k$ .

Similarly, a group code  $\mathcal{C}$  is a subgroup of a symbol configuration space  $\mathcal{A} = \prod_{k \in \mathcal{I}_{\mathcal{A}}} \mathcal{A}_k$ , where each symbol alphabet  $\mathcal{A}_k$  is a finite abelian group. If  $\mathcal{I}_{\mathcal{A}}$  is finite, then  $|\mathcal{A}| = \prod_{k \in \mathcal{I}_{\mathcal{A}}} |\mathcal{A}_k|$  is finite.

For a *state realization* of  $\mathcal{C}$ , we define also a set  $\{\mathcal{S}_j : j \in \mathcal{I}_{\mathcal{S}}\}$  of *state spaces*  $\mathcal{S}_j$  indexed by a state index set  $\mathcal{I}_{\mathcal{S}}$ , and a set  $\{\mathcal{C}_i : i \in \mathcal{I}_{\mathcal{C}}\}$  of *local constraint codes*  $\mathcal{C}_i$  indexed by a constraint index set  $\mathcal{I}_{\mathcal{C}}$ , where each constraint code  $\mathcal{C}_i$  involves subsets of the symbol and state variables indexed by  $\mathcal{I}_{\mathcal{A}}^{(i)} \subseteq \mathcal{I}_{\mathcal{A}}$  and  $\mathcal{I}_{\mathcal{S}}^{(i)} \subseteq \mathcal{I}_{\mathcal{S}}$ , so  $\mathcal{C}_i \subseteq \mathcal{A}^{(i)} \times \mathcal{S}^{(i)}$ , where  $\mathcal{A}^{(i)} = \prod_{k \in \mathcal{I}_{\mathcal{A}}^{(i)}} \mathcal{A}_k$  and  $\mathcal{S}^{(i)} = \prod_{j \in \mathcal{I}_{\mathcal{S}}^{(i)}} \mathcal{S}_j$ .

In a linear realization, each state space  $\mathcal{S}_j$  and each constraint code  $\mathcal{C}_i$  is a vector space over  $\mathbb{F}$ ; in a group realization, each state space and constraint code is a finite abelian group.

The *state configuration space* is  $\mathcal{S} = \prod_{j \in \mathcal{I}_{\mathcal{S}}} \mathcal{S}_j$ . We may alternatively refer to a state variable as an *internal variable*  $S_j$ , which takes values  $s_j$  in an internal variable alphabet  $\mathcal{S}_j$ .

The *degree* of a variable is the number of constraint codes in which it is involved; the *degree* of a constraint code is the number of variables that it involves.

A *configuration* (or *trajectory*) of the realization is a pair  $(\mathbf{a}, \mathbf{s}) \in \mathcal{A} \times \mathcal{S}$  for which all constraints are satisfied; *i.e.*,  $(\mathbf{a}^{(i)}, \mathbf{s}^{(i)}) \in \mathcal{C}_i, \forall i \in \mathcal{I}_{\mathcal{C}}$ . Its *full behavior* is the set  $\mathfrak{B}$  of all such valid configurations. The *code*  $\mathcal{C}$  realized by the realization is then the set of all  $\mathbf{a} \in \mathcal{A}$  that appear in some  $(\mathbf{a}, \mathbf{s}) \in \mathfrak{B}$ ; *i.e.*,  $\mathcal{C}$  is the projection of  $\mathfrak{B}$  onto  $\mathcal{A}$ . A linear realization realizes a linear code; a group realization realizes a group code.

### B. Observability

A state realization is called *one-to-one*, or *observable*, if there is precisely one pair  $(\mathbf{a}, \mathbf{s}) \in \mathfrak{B}$  corresponding to each  $\mathbf{a} \in \mathcal{C}$ .

For a linear or group realization, we define the *unobservable state configuration space*  $\mathcal{S}^u$  as the subspace of *unobservable state configurations*  $\mathbf{s} \in \mathcal{S}$  such that  $(\mathbf{0}, \mathbf{s}) \in \mathfrak{B}$ ; thus such a realization is observable if and only if  $\mathcal{S}^u$  is trivial. Indeed,  $\mathcal{S}^u$  is evidently isomorphic to the kernel  $\mathcal{K}^u = \{(\mathbf{0}, \mathbf{s}) \in$

$\mathfrak{B}$  of the projection of  $\mathfrak{B}$  onto  $\mathcal{A}$ , whose image is  $\mathcal{C}$ , so by the fundamental theorem of homomorphisms  $\mathcal{C} \cong \mathfrak{B}/\mathcal{K}^u$ . Therefore  $|\mathcal{C}| = |\mathfrak{B}|/|\mathcal{S}^u|$ , or, in the linear case,  $\dim \mathcal{C} = \dim \mathfrak{B} - \dim \mathcal{S}^u$ .

### C. Minimality and local reducibility

As in [18], a realization with state spaces  $\mathcal{S}_j$  will be called *minimal* if there exists no realization of the same code with state spaces  $\tilde{\mathcal{S}}_j$  that has the same graph topology (*i.e.*, the same index sets  $\mathcal{I}_{\mathcal{A}}, \mathcal{I}_{\mathcal{S}}, \mathcal{I}_{\mathcal{C}}, \mathcal{I}_{\mathcal{A}^{(i)}}, \mathcal{I}_{\mathcal{S}^{(i)}}$ ) such that  $|\tilde{\mathcal{S}}_j| \leq |\mathcal{S}_j|$  for all  $j$ , with at least one strict inequality.

A realization of a code  $\mathcal{C}$  with state spaces  $\mathcal{S}_j$  and constraint codes  $\mathcal{C}_i$  will be called *locally reducible* if it is possible to replace a single state space  $\mathcal{S}_j$  by a strictly smaller space, and the constraint codes involving  $\mathcal{S}_j$  by constraint codes that are no bigger, without changing the rest of the realization or the code that it realizes. Evidently a locally reducible realization is nonminimal.

Our basic tools for local reduction will be trimming (restricting) and merging (taking quotients) of state spaces, which we will see are dual operations.

### D. State variables of degree one

As a first exercise in local reducibility, we observe that any realization that includes a nontrivial state variable of degree 1 is locally reducible, since we may always reduce the alphabet of that variable to the trivial alphabet (*i.e.*, merge all states with the zero state) without changing the code realized by the realization, there being no other constraints on that state variable. Thus we will assume henceforth that state variables of degree 1 have a trivial state space.

We note further that, in any state realization, a state variable whose alphabet size is one may optionally be deleted from the realization without affecting the code that is realized. However, this is not the usual convention for end states in trellis realizations; see below.

### E. Normal realizations

A state realization is called *normal* if the degree of every symbol variable is 1, and the degree of every state variable is 2. (More generally, a state variable could have degree 1, provided that its alphabet is trivial; see the previous subsection.)

As shown in [4], any realization may be straightforwardly “normalized” by introducing replica internal variables and equality constraints, as follows. For every constraint code  $\mathcal{C}_i$  and every variable  $A_k$  or  $S_j$  involved in  $\mathcal{C}_i$ , create a replica internal variable  $A_{ki}$  or  $S_{ji}$ ; then, for every  $S_j$ , introduce an equality constraint on all the replicas  $S_{ji}$  of  $S_j$  and delete the original variable  $S_j$ , and for every  $A_k$ , introduce an equality constraint on all the replicas  $A_{ki}$  of  $A_k$ , plus the original external variable  $A_k$ . The resulting realization is normal, realizes the same code  $\mathcal{C}$ , and has essentially the same graph topology. Thus we will assume henceforth without essential loss of generality that all realizations are normal.

A normal realization has a natural graphical representation, called a *normal graph*, in which constraints are represented by

vertices, internal variables by ordinary edges (edges of degree 2), and external variables by half-edges (edges of degree 1). An edge or half-edge is incident on a vertex whenever the corresponding variable is involved in the corresponding constraint code.

If the graph of a realization is disconnected, then the behavior and the code that it realizes are the Cartesian products of the respective behaviors and codes realized by the components. We will therefore assume that the graphs of all realizations are connected.

### F. Trellis realizations

A *trellis realization* is a state realization in which every constraint code involves precisely two state variables, every state variable is involved in one or two constraint codes and every symbol variable is involved in one constraint code. Thus the graph of a trellis realization must be one of the following (if connected):

- a finite, semi-infinite or bi-infinite chain graph (called a “path” in graph theory), which we call a *conventional trellis realization*;
- a single-cycle graph, which we call a *tail-biting trellis realization*.

Thus bi-infinite and tail-biting trellis realizations are inherently normal. Finite or semi-infinite conventional trellis realizations are not strictly normal, because they have “end” state variables of degree 1, but they may be taken to be normal if the associated degree-1 state alphabets are trivial.

We note that a finite conventional trellis realization may be considered to be a special case of a tail-biting trellis realization in which one state variable has a trivial alphabet, and therefore could be deleted.

We will sometimes depict a trellis realization by a traditional *trellis diagram*, in which all branches, states and symbols are depicted explicitly.

### G. Duality

We now briefly recapitulate some of the basic duality principles for both the group and linear cases that will be used heavily in this paper, following [9].

If  $G$  is a finite abelian group, then its *dual group*  $\hat{G}$  is the character group of  $G$ , namely the set of all homomorphisms from  $G$  to the circle group  $\mathbb{R}/\mathbb{Z}$ . The dual group of  $\hat{G}$  may be taken as  $G$ . In the finite abelian case,  $G$  and  $\hat{G}$  are isomorphic.

For  $g \in G$  and  $\hat{g} \in \hat{G}$ , there is a well-defined *pairing*  $\langle \hat{g}, g \rangle = \hat{g}(g) \in \mathbb{R}/\mathbb{Z}$  that has the usual bihomomorphic properties: *e.g.*,  $\langle 0, g \rangle = \langle \hat{g}, 0 \rangle = 0$ ,  $\langle \hat{g}_1 + \hat{g}_2, g \rangle = \langle \hat{g}_1, g \rangle + \langle \hat{g}_2, g \rangle$ , and so forth.

Elements  $g \in G$  and  $\hat{g} \in \hat{G}$  are said to be *orthogonal* if  $\langle \hat{g}, g \rangle = 0$ . If  $H$  is a subgroup of  $G$ , then the orthogonal subgroup  $H^\perp$  is the set of all elements of  $\hat{G}$  that are orthogonal to all elements of  $H$ .  $H^\perp$  is indeed a subgroup of  $\hat{G}$ , and the orthogonal subgroup to  $H^\perp$  is  $H$ .

The dual group  $\hat{H}$  to  $H \subseteq G$  is isomorphic to the quotient group  $\hat{G}/H^\perp$ . In the finite abelian case, this implies  $|H||H^\perp| = |G| = |\hat{G}|$ .

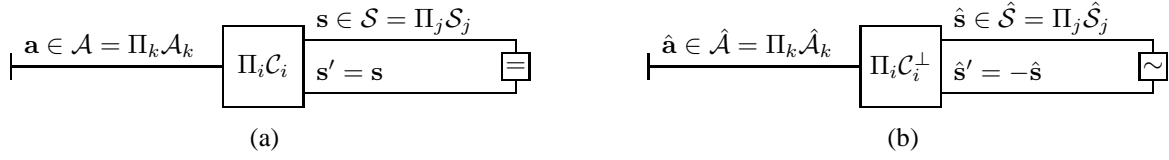


Fig. 1. Normal realization duality: Dual realizations realize orthogonal codes.

Similarly, if  $G$  is a finite-dimensional vector space over a finite field  $\mathbb{F}$ , then it has a *dual space*  $\hat{G}$ , a vector space over  $\mathbb{F}$  of the same dimension, which is furnished with an *inner product*  $\langle \hat{g}, g \rangle \in \mathbb{F}$  that is a bilinear form: *i.e.*,  $\langle 0, g \rangle = \langle \hat{g}, 0 \rangle = 0$ ,  $\langle \hat{g}_1 + \hat{g}_2, g \rangle = \langle \hat{g}_1, g \rangle + \langle \hat{g}_2, g \rangle$ , and so forth. It is well known that all dual spaces of  $G$  are isomorphic, where the isomorphism preserves the inner product. If  $G = \mathbb{F}^\ell$  for some  $\ell$ , then we may and will choose  $\hat{G} = \mathbb{F}^\ell$ , along with the standard inner (dot) product on  $\mathbb{F}^\ell$ .

Moreover, if  $\{b_1, \dots, b_n\}$  is a basis for an  $n$ -dimensional vector space  $G$ , then there exists a dual basis  $\{\hat{b}_1, \dots, \hat{b}_n\}$  for  $\hat{G}$  such that  $\langle \hat{b}_i, b_j \rangle = \delta_{ij}$ , the Kronecker delta function. Given any  $g \in G$ , there exists a unique *coordinate vector*  $\alpha \in \mathbb{F}^n$  such that  $g = \sum_j \alpha_j b_j$ ; this establishes an isomorphism between  $G$  and  $\mathbb{F}^n$ . Similarly, each  $\hat{g} \in \hat{G}$  has a unique coordinate vector  $\hat{\alpha} \in \mathbb{F}^n$  such that  $\hat{g} = \sum_i \hat{\alpha}_i \hat{b}_i$ . If  $g = \sum_j \alpha_j b_j$  and  $\hat{g} = \sum_i \hat{\alpha}_i \hat{b}_i$ , then

$$\langle \hat{g}, g \rangle = \sum_i \sum_j \hat{\alpha}_i \alpha_j \langle \hat{b}_i, b_j \rangle = \sum_i \hat{\alpha}_i \alpha_i = \hat{\alpha} \cdot \alpha;$$

*i.e.*, the inner product  $\langle \hat{g}, g \rangle$  is the dot product  $\hat{\alpha} \cdot \alpha$  of the coordinate vectors. In other words, dual bases define adjoint isomorphisms  $G \leftrightarrow \mathbb{F}^n$  and  $\hat{G} \leftrightarrow \mathbb{F}^n$  that preserve inner products.

As with groups, if  $H$  is a subspace of  $G$ , then the orthogonal subspace  $H^\perp$  is a subspace of  $\hat{G}$ , and the orthogonal subspace to  $H^\perp$  is  $H$ . Moreover,  $\dim H + \dim H^\perp = \dim G = \dim \hat{G}$ .

If  $\mathcal{G} = \Pi_k G_k$  is a finite direct product of a collection of groups or vector spaces  $G_k$ , then the dual group or space to  $\mathcal{G}$  is the finite direct product  $\hat{\mathcal{G}} = \Pi_k \hat{G}_k$ , and the pairing or inner product between  $\mathbf{g} \in \mathcal{G}$  and  $\hat{\mathbf{g}} \in \hat{\mathcal{G}}$  is given by the componentwise sum

$$\langle \hat{\mathbf{g}}, \mathbf{g} \rangle = \sum_k \langle \hat{\mathbf{g}}_k, \mathbf{g}_k \rangle.$$

If  $\mathcal{H} = \Pi_k H_k$  is a direct product of subgroups or subspaces  $H_k \subseteq G_k$ , then the orthogonal space is the direct product  $\mathcal{H}^\perp = \Pi_k H_k^\perp \subseteq \hat{\mathcal{G}}$ .

### H. Projection/cross-section duality

Projection/cross-section duality is one of the most fundamental and useful duality relationships for linear and group codes [4].

Given a linear or group code  $\mathcal{C} \subseteq \mathcal{A} = \prod_{k \in \mathcal{I}_A} \mathcal{A}_k$ , let  $\mathcal{J} \subseteq \mathcal{I}_A$  be any subset of the symbol index set, and  $\mathcal{K} = \mathcal{I}_A - \mathcal{J}$  its complement. Then  $\mathcal{A}$  may be written as the direct product  $\mathcal{A}^\mathcal{J} \times \mathcal{A}^\mathcal{K}$ , where  $\mathcal{A}^\mathcal{J} = \prod_{k \in \mathcal{J}} \mathcal{A}_k$  and  $\mathcal{A}^\mathcal{K} = \prod_{k \in \mathcal{K}} \mathcal{A}_k$ . Correspondingly we may write any  $\mathbf{a} \in \mathcal{A}$  as a pair  $(\mathbf{a}^\mathcal{J}, \mathbf{a}^\mathcal{K})$ , where  $\mathbf{a}^\mathcal{J} \in \mathcal{A}^\mathcal{J}$  and  $\mathbf{a}^\mathcal{K} \in \mathcal{A}^\mathcal{K}$  are the *projections* of  $\mathbf{a}$  on  $\mathcal{J}$  and  $\mathcal{K}$ , respectively.

The *projection map*  $P_\mathcal{J} : \mathcal{A} \rightarrow \mathcal{A}^\mathcal{J}$  defined by  $\mathbf{a} \mapsto \mathbf{a}^\mathcal{J}$  is evidently a homomorphism. Given a linear or group code  $\mathcal{C} \subseteq \mathcal{A}$ , the *projection* of  $\mathcal{C}$  onto  $\mathcal{A}^\mathcal{J}$  is  $\mathcal{C}_{|\mathcal{A}^\mathcal{J}} = P_\mathcal{J}(\mathcal{C})$ , a subgroup of  $\mathcal{A}^\mathcal{J}$ .

The *cross-section*  $\mathcal{C}_{:\mathcal{A}^\mathcal{J}}$  of  $\mathcal{C}$  on  $\mathcal{A}^\mathcal{J}$  is defined as  $\mathcal{C}_{:\mathcal{A}^\mathcal{J}} = \{\mathbf{a}^\mathcal{J} \in \mathcal{A}^\mathcal{J} \mid (\mathbf{a}^\mathcal{J}, \mathbf{0}^\mathcal{K}) \in \mathcal{C}\}$ , which is evidently a subgroup of  $\mathcal{C}_{|\mathcal{A}^\mathcal{J}}$ . We note that  $\mathcal{C}_{:\mathcal{A}^\mathcal{J}}$  is trivially isomorphic to the kernel of the projection map  $P_\mathcal{K} : \mathcal{C} \rightarrow \mathcal{A}^\mathcal{K}$ , a subcode of the  $\mathcal{C}$ ; therefore the following duality relationship is sometimes called projection/subcode duality.

**Lemma (Projection/cross-section duality).** If  $\mathcal{C} \subseteq \mathcal{A}^\mathcal{J} \times \mathcal{A}^\mathcal{K}$  and  $\mathcal{C}^\perp \subseteq \hat{\mathcal{A}}^\mathcal{J} \times \hat{\mathcal{A}}^\mathcal{K}$  are orthogonal linear or group block codes, then  $\mathcal{C}_{:\mathcal{A}^\mathcal{J}}$  and  $(\mathcal{C}^\perp)_{|\hat{\mathcal{A}}^\mathcal{J}}$  are orthogonal linear or group codes.

*Proof.* Because pairings or inner products are componentwise sums, we have  $\langle \hat{\mathbf{a}}, \mathbf{a} \rangle = \langle \hat{\mathbf{a}}^\mathcal{J}, \mathbf{a}^\mathcal{J} \rangle + \langle \hat{\mathbf{a}}^\mathcal{K}, \mathbf{a}^\mathcal{K} \rangle$ . Thus  $(\hat{\mathbf{a}}^\mathcal{J}, \hat{\mathbf{a}}^\mathcal{K}) \perp (\mathbf{a}^\mathcal{J}, \mathbf{0}^\mathcal{K}) \Leftrightarrow \hat{\mathbf{a}}^\mathcal{J} \perp \mathbf{a}^\mathcal{J}$ . We therefore have the following logical chain:

$$\mathbf{a}^\mathcal{J} \in \mathcal{C}_{:\mathcal{A}^\mathcal{J}} \Leftrightarrow (\mathbf{a}^\mathcal{J}, \mathbf{0}^\mathcal{K}) \in \mathcal{C} \Leftrightarrow (\mathbf{a}^\mathcal{J}, \mathbf{0}^\mathcal{K}) \perp \mathcal{C}^\perp \Leftrightarrow \mathbf{a}^\mathcal{J} \perp (\mathcal{C}^\perp)_{|\hat{\mathcal{A}}^\mathcal{J}}.$$

### I. Dual realizations

The dual realization of a normal linear or group realization is defined below. The normal graph duality theorem of [4] states that the dual realization realizes the orthogonal code  $\mathcal{C}^\perp$  to  $\mathcal{C}$ . By now there exist several proofs of this theorem, the most elegant of which involve Fourier transforms of code indicator functions [22], [1], [7]. We give here a new elementary proof (similar to that of Koetter [16]), partly in order to introduce concepts and notation that will be useful later in Section IV.

Consider the set  $\Pi_i \mathcal{C}_i$  of all possible combinations of codewords of the constraint codes  $\mathcal{C}_i$ . With a normal realization, every external variable  $A_k$  is involved in precisely one constraint code, and every internal variable  $S_j$  is involved in precisely two constraint codes. Let us replace  $S_j$  by a replica variable  $S'_j$  in one of its two appearances (it does not matter which), and add an equality constraint  $s'_j = s_j$ . Then each element of  $\Pi_i \mathcal{C}_i$  may be uniquely identified by a triple  $(\mathbf{a}, \mathbf{s}, \mathbf{s}') \in \mathcal{A} \times \mathcal{S} \times \mathcal{S}$ .

A valid configuration  $(\mathbf{a}, \mathbf{s}, \mathbf{s}') \in \Pi_i \mathcal{C}_i$  is one for which  $\mathbf{s}' = \mathbf{s}$ . Thus the behavior of the realization is  $\mathfrak{B} = \{(\mathbf{a}, \mathbf{s}) \in \mathcal{A} \times \mathcal{S} \mid (\mathbf{a}, \mathbf{s}, \mathbf{s}) \in \Pi_i \mathcal{C}_i\}$ . This is illustrated in Fig. 1(a).

The *dual realization* is defined as the realization with the same graph topology in which the external alphabets  $\mathcal{A}_k$  are replaced by their duals  $\hat{\mathcal{A}}_k$ , the internal alphabets  $\mathcal{S}_j$  are replaced by their duals  $\hat{\mathcal{S}}_j$ , the constraint codes  $\mathcal{C}_i$  are replaced by their orthogonal codes  $\mathcal{C}_i^\perp$ , and each equality constraint  $s'_j = s_j$  is replaced by the sign inversion constraint  $\hat{s}'_j = -\hat{s}_j$ , as illustrated in Fig. 1(b). Thus the dual behavior may be

written as  $\mathfrak{B}^\circ = \{(\hat{\mathbf{a}}, \hat{\mathbf{s}}) \in \hat{\mathcal{A}} \times \hat{\mathcal{S}} \mid (\hat{\mathbf{a}}, \hat{\mathbf{s}}, -\hat{\mathbf{s}}) \in \Pi_i \mathcal{C}_i^\perp\}$ . Note the slight sign asymmetry between the definitions of the primal and dual behaviors.

Alternatively, a dual way of defining the primal behavior is as follows. Following Koetter [16], we may write  $\mathfrak{B} = \{(\mathbf{a}, \mathbf{s}) \in \mathcal{A} \times \mathcal{S} \mid (\mathbf{a}, \mathbf{s}, \mathbf{s}) \perp \Pi_i \mathcal{C}_i^\perp\}$ ; that is,  $\mathfrak{B}$  is the set of all configurations  $(\mathbf{a}, \mathbf{s})$  such that  $(\mathbf{a}, \mathbf{s}, \mathbf{s}) \perp (\hat{\mathbf{a}}, \hat{\mathbf{s}}, \hat{\mathbf{s}}')$  for all  $(\hat{\mathbf{a}}, \hat{\mathbf{s}}, \hat{\mathbf{s}}') \in \Pi_i \mathcal{C}_i^\perp$ , or equivalently  $(\mathbf{a}, \mathbf{s}) \perp (\hat{\mathbf{a}}, \hat{\mathbf{s}} + \hat{\mathbf{s}}')$ . Therefore if we define  $\mathfrak{B}^\perp$  as the image of the homomorphism  $\Sigma : \prod_i \mathcal{C}_i^\perp \rightarrow \hat{\mathcal{A}} \times \hat{\mathcal{S}}$  that is defined by the sum map  $(\hat{\mathbf{a}}, \hat{\mathbf{s}}, \hat{\mathbf{s}}') \mapsto (\hat{\mathbf{a}}, \hat{\mathbf{s}} + \hat{\mathbf{s}}')$ , then  $\mathfrak{B}$  is the orthogonal code to  $\mathfrak{B}^\perp$ .

Notice now that  $(\hat{\mathbf{a}}, \hat{\mathbf{s}}, \hat{\mathbf{s}}') \in \prod_i \mathcal{C}_i^\perp$  corresponds to a valid configuration  $(\hat{\mathbf{a}}, \hat{\mathbf{s}}, -\hat{\mathbf{s}})$  in the dual behavior if and only if it is mapped by  $\Sigma$  to  $(\hat{\mathbf{a}}, \mathbf{0})$ . Therefore the code  $\mathcal{D}$  realized by the dual behavior is precisely the cross-section  $(\mathfrak{B}^\perp)_{\hat{\mathcal{A}}} = \{\hat{\mathbf{a}} \in \hat{\mathcal{A}} \mid (\hat{\mathbf{a}}, \mathbf{0}) \in \mathfrak{B}^\perp\}$ . But by projection/cross-section duality, the orthogonal code to  $\mathcal{D}$  must be the projection  $\mathfrak{B}|_{\mathcal{A}}$ , which is  $\mathcal{C}$ ; *i.e.*,  $\mathcal{D} = \mathcal{C}^\perp$ .

In summary, we have proved:

**Lemma (Normal realization duality).** If a normal realization as in Fig. 1(a) realizes a linear or group code  $\mathcal{C}$ , then its dual realization as in Fig. 1(b) realizes the orthogonal code  $\mathcal{C}^\perp$ .  $\square$

### III. TRIMNESS AND PROPERNESS

In this section we discuss trimness and properness of constraint codes. These are dual properties; *i.e.*,  $\mathcal{C}_i$  is trim if and only if  $\mathcal{C}_i^\perp$  is proper. We show that a realization is locally reducible if it has a constraint code that is not both trim and proper. Finally, we show that a linear finite cycle-free realization is minimal if and only if all constraint codes are trim and proper.

#### A. Trim-proper duality

A constraint code  $\mathcal{C}_i$  will be called *trim* if the projection of  $\mathcal{C}_i$  onto every state space  $\mathcal{S}_j$  that is involved in  $\mathcal{C}_i$  is  $\mathcal{S}_j$ ; *i.e.*, if every such projection is surjective.

A constraint code  $\mathcal{C}_i$  will be called *proper* if the value of any state variable  $\mathcal{S}_j$  involved in  $\mathcal{C}_i$  is determined by the values of all other variables; *i.e.*, if for any set of values of all variables involved in  $\mathcal{C}_i$  other than  $\mathcal{S}_j$ , at most one codeword in  $\mathcal{C}_i$  has those values. In the linear or group case,  $\mathcal{C}_i$  is proper if and only if there is no codeword of  $\mathcal{C}_i$  whose support is a single state variable  $\mathcal{S}_j$ ; *i.e.*, for all  $\mathcal{S}_j$  the cross-section  $(\mathcal{C}_i)_{\mathcal{S}_j}$  is trivial.

This definition of “proper” generalizes the traditional definition for a trellis realization, which is called proper [23] (or “biproper” [25]) if the symbol values associated with all transitions to or from a given state are all different.<sup>1</sup>

As in [12], we observe that trimness and properness are dual properties:

**Theorem 1 (Trim-proper duality).** A linear or group constraint code  $\mathcal{C}_i$  is trim if and only if its orthogonal code  $\mathcal{C}_i^\perp$  is proper.

<sup>1</sup>In system theory, for conventional state-space (trellis) realizations, properness is sometimes called “instantaneous observability,” because then and only then is the next state determined by the current state and symbol.

*Proof:* By projection/cross-section duality, the cross-section  $(\mathcal{C}_i^\perp)_{\hat{\mathcal{S}}_j}$  is the orthogonal code to the projection  $(\mathcal{C}_i)_{\mathcal{S}_j}$ . Thus  $\{0\} \subset (\mathcal{C}_i^\perp)_{\hat{\mathcal{S}}_j}$  if and only if  $(\mathcal{C}_i)_{\mathcal{S}_j} \subset \mathcal{S}_j$ .<sup>2</sup>  $\square$

This duality is illustrated in Fig. 2.  $\mathcal{C}_i$  is not trim at  $\mathcal{S}_j$  if and only if the projection  $(\mathcal{C}_i)_{\mathcal{S}_j}$  is a proper subspace or subgroup  $\mathcal{T} \subset \mathcal{S}_j$ . In this case  $\mathcal{C}_i$  may be represented as the concatenation of a trimmed constraint code  $\tilde{\mathcal{C}}_i$ , in which  $\mathcal{S}_j$  is replaced by  $\mathcal{T}$  without changing any codewords, and an *inclusion constraint code*  $\mathcal{C}_{\hookrightarrow} = \{(t, t) \in \mathcal{T} \times \mathcal{S}_j : t \in \mathcal{T}\}$ , as illustrated in Fig. 2(a).

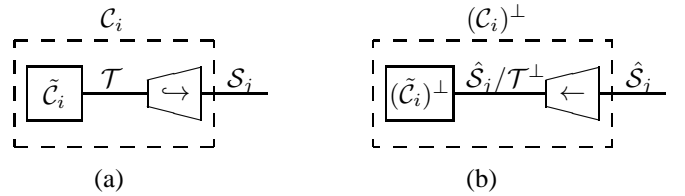


Fig. 2. Dual realizations: (a) nontrim  $\mathcal{C}_i$ ; (b) improper dual  $(\mathcal{C}_i)^\perp$ .

By definition, the dual realization of the improper code  $(\mathcal{C}_i)^\perp$  is obtained by concatenating the orthogonal code  $(\tilde{\mathcal{C}}_i)^\perp$ , a sign inverter, and the orthogonal code  $(\mathcal{C}_{\hookrightarrow})^\perp$ . It is easy to see that  $(\mathcal{C}_{\hookrightarrow})^\perp = \{(-\hat{s} + \mathcal{T}^\perp, \hat{s}) \in \hat{\mathcal{S}}_j/\mathcal{T}^\perp \times \hat{\mathcal{S}}_j : \hat{s} \in \hat{\mathcal{S}}_j\}$ , using the inner product  $\langle \hat{s} + \mathcal{T}^\perp, t \rangle = \langle \hat{s}, t \rangle$  for  $t \in \mathcal{T}$  and  $\hat{s} \in \hat{\mathcal{S}}_j$ .<sup>3</sup> Combining the sign inverter and  $(\mathcal{C}_{\hookrightarrow})^\perp$ , we obtain the *quotient constraint code*  $\mathcal{C}_{\leftarrow} = \{(\hat{s} + \mathcal{T}^\perp, \hat{s}) \in \hat{\mathcal{S}}_j/\mathcal{T}^\perp \times \hat{\mathcal{S}}_j : \hat{s} \in \hat{\mathcal{S}}_j\}$ , which involves the natural map from  $\hat{\mathcal{S}}_j$  to its quotient  $\hat{\mathcal{S}}_j/\mathcal{T}^\perp$ , as shown in Fig. 2(b).

In view of Theorem 1, every improper constraint code may be represented as in Fig. 2(b) as a code with a reduced state space  $\hat{\mathcal{S}}'_j = \hat{\mathcal{S}}_j/\mathcal{T}^\perp$ , obtained by merging all states in each coset  $\hat{s} + \mathcal{T}^\perp$ , plus a constraint code which allows every coset of  $\mathcal{T}^\perp$  to branch to all of its elements.

#### B. Local irreducibility requires trimness and properness

We now show that both “trim” and “proper” are necessary for local irreducibility.

**Theorem 2 (Local reducibility).** A normal linear or group realization is locally reducible if any constraint code  $\mathcal{C}_i$  is not both trim and proper.

*Proof:* We give a pictorial proof, following Fig. 2. Suppose  $\mathcal{C}_i$  is not trim at  $\mathcal{S}_j$ ; *i.e.*,  $\mathcal{T} = (\mathcal{C}_i)_{\mathcal{S}_j} \subset \mathcal{S}_j$ . Let  $\mathcal{C}_{i'}$  be the other constraint in which  $\mathcal{S}_j$  is involved. Then, as shown in Fig. 3(a), the combination of  $\mathcal{C}_i$  with  $\mathcal{C}_{i'}$  is equivalent to the combination of the trimmed code  $\tilde{\mathcal{C}}_i$  with a second trimmed code  $\tilde{\mathcal{C}}_{i'}$ , namely the combination of  $\mathcal{C}_{i'}$  with  $\mathcal{C}_{\hookrightarrow}$ , which restricts  $\mathcal{S}_j$  to  $\mathcal{T}$ . Replacing the combination of  $\mathcal{C}_i$  and  $\mathcal{C}_{i'}$  with that of  $\tilde{\mathcal{C}}_i$  and  $\tilde{\mathcal{C}}_{i'}$  thus strictly reduces the connecting state space from  $\mathcal{S}_j$  to  $\mathcal{T}$ , without changing the code  $\mathcal{C}$  realized by the realization.

Dually, suppose  $\mathcal{C}_i$  is not proper at  $\mathcal{S}_j$ ; *i.e.*,  $\{0\} \subset \mathcal{T} = (\mathcal{C}_i)_{\mathcal{S}_j}$ . Again, let  $\mathcal{C}_{i'}$  be the other constraint in which  $\mathcal{S}_j$  is involved. Then, as shown in Fig. 3(b), the combination of  $\mathcal{C}_i$  with  $\mathcal{C}_{i'}$  is equivalent to the combination of the merged code

<sup>2</sup>By “ $A \subset B$ ,” we mean that  $A$  is a proper subset of  $B$ .

<sup>3</sup>In other words, the inclusion map  $\mathcal{T}_j \hookrightarrow \mathcal{S}_j$  and the natural map  $\hat{\mathcal{S}}_j \rightarrow \hat{\mathcal{S}}_j/(\mathcal{T}_j)^\perp$  are adjoint homomorphisms.

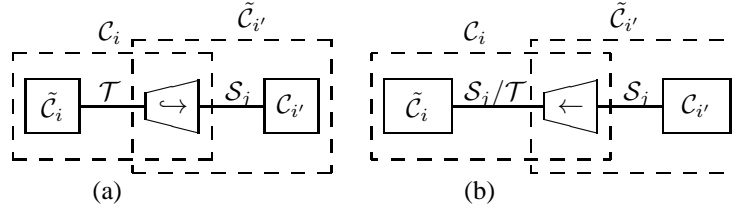


Fig. 3. Dual reductions: (a) nontrim  $C_i$ ; (b) improper  $C_i$ .

$\tilde{C}_i$  with a second merged code  $\tilde{C}_{i'}$ , namely the combination of  $C_{i'}$  with  $C_{\leftarrow}$ , which merges  $S_j$  to  $S_j/T$  via the natural map. Replacing the combination of  $C_i$  and  $C_{i'}$  with that of  $\tilde{C}_i$  and  $\tilde{C}_{i'}$  thus strictly reduces the connecting state space from  $S_j$  to  $S_j/T$ , without changing the code  $\mathcal{C}$  realized by the realization.  $\square$

Theorem 2 and its proof show that in any normal linear or group realization, if any constraint code is not trim or improper, then there is a strict reduction of the corresponding state space  $S_j$  via trimming or merging, respectively, such that only the two constraint codes in which  $S_j$  is involved are affected. It is therefore hard to imagine any application in which one would not begin by ensuring that all constraint codes have been made trim and proper.

This development shows that “merging” (quotient-taking) is the dual reduction to “trimming” (restricting). Quotient-taking preserves the linear or group property of the realization, and thus is the appropriate notion of “merging” for linear or group realizations.

For conventional and tail-biting trellis realizations, the fact that “improper” implies “mergeable” is well known; see, *e.g.*, [18].

**Example 1** (compare [25, Fig. 2]). Consider the binary linear block code  $\mathcal{C} = \{000, 110\}$ . This code may be realized by the linear trellis realization shown in Fig. 4(a), with three binary symbol alphabets, four state spaces  $S_0 = S_3 = \{0\}, S_1 = S_2 = \{0, 1\}$ , and three constraint codes  $C_0 = \{000, 011\}, C_1 = \{000, 111\}$ , and  $C_2 = \{000, 100\}$ . Since  $C_2$  has a nonzero codeword 100 supported by the single state space  $S_2$ ,  $C_2$  is improper, and therefore can be reduced. Indeed, the two states of  $S_2$  can be merged to a single state  $\{0\}$  without affecting  $\mathcal{C}$ ; this yields the minimal linear trellis realization of this code, shown in Fig. 4(b).

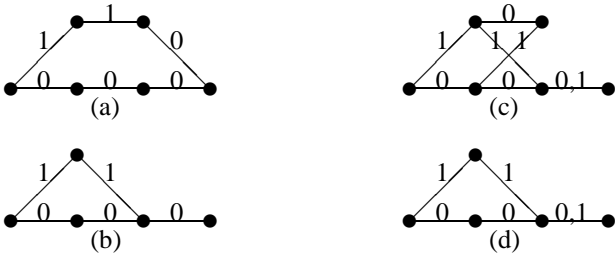


Fig. 4. (a) Improper, (b) merged, (c) non-trim and (d) trimmed realizations of dual codes.

The dual linear trellis realization to that of Fig. 4(a) uses the same variable alphabets and the orthogonal constraint codes  $C_0^\perp = \{000, 011\}, C_1^\perp = \{000, 011, 110, 101\}$ , and  $C_2^\perp = \{000, 010\}$ , as shown in Fig. 4(c). (Because the ground field is binary, no sign inverters are needed.) It can be seen

that this dual realization, although unusual, does indeed realize  $\mathcal{C}^\perp = \{000, 110, 001, 111\}$ . Since the projection of  $C_2^\perp$  onto  $S_2$  is  $\{0\}$ ,  $C_2^\perp$  is not trim, and therefore can be reduced. Indeed, restricting  $S_2$  to  $\{0\}$ , we obtain the minimal trellis realization of  $\mathcal{C}^\perp$  of Fig. 4(d).  $\square$

### C. Minimal cycle-free realizations

It is well known that a finite conventional linear trellis realization is minimal if and only if all end state spaces are trivial and every constraint code  $C_i$  is trim and proper [25]. We now show that this result holds for any finite cycle-free linear or group realization. As a byproduct, we obtain a constructive proof of the State Space Theorem, which is perhaps the most fundamental result of linear behavioral system theory [27].

The key graph-theoretic property of cycle-free graphs is that every edge is a cut set. In other words, if  $S_j$  is any state space in a normal cycle-free realization, then cutting the edge corresponding to  $S_j$  partitions the normal graph of the realization into two disconnected subgraphs, which we label arbitrarily as  $\mathcal{P}_j$  and  $\mathcal{F}_j$  (for “past” and “future”), as shown in Fig. 5.

Furthermore, if we define the *past depth*  $d_j^{\mathcal{P}}$  of the edge  $S_j$  as the maximum distance to any leaf node in  $\mathcal{P}_j$ , then the depths of all other states involved in the constraint code  $\mathcal{D}_j^{\mathcal{P}}$  in  $\mathcal{P}_j$  to which  $S_j$  is connected must be less than  $d_j^{\mathcal{P}}$  (see Fig. 5). (Again, we assume there are no degree-1 states.)

We correspondingly represent the symbol configuration space  $\mathcal{A}$  as the product  $\mathcal{A}^{\mathcal{P}_j} \times \mathcal{A}^{\mathcal{F}_j}$  of past and future components, and write a symbol configuration as a pair  $\mathbf{a} = (\mathbf{a}^{\mathcal{P}_j}, \mathbf{a}^{\mathcal{F}_j})$ . The projection and cross-section of the code  $\mathcal{C}$  on the past are written as  $\mathcal{C}_{|\mathcal{A}^{\mathcal{P}_j}}$  and  $\mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}}$ , respectively. We similarly define past and future state configurations  $\mathbf{s}^{\mathcal{P}_j} \in \mathcal{S}^{\mathcal{P}_j}$  and  $\mathbf{s}^{\mathcal{F}_j} \in \mathcal{S}^{\mathcal{F}_j}$ , where  $S_j$  is included in both  $\mathcal{S}^{\mathcal{P}_j}$  and  $\mathcal{S}^{\mathcal{F}_j}$ . Finally, we say that a past configuration  $\mathbf{a}^{\mathcal{P}_j}$  reaches a state  $s_j \in S_j$  if there is a past configuration  $(\mathbf{a}^{\mathcal{P}_j}, \mathbf{s}^{\mathcal{P}_j})$  that satisfies all past constraints and includes  $s_j$ .

We then have the following remarkably simple result:

**Theorem 3 (Minimal = trim + proper).** Given a normal linear or group realization of a code  $\mathcal{C}$  on a finite connected cycle-free graph  $\mathcal{G}$ , the following are equivalent:

- (1) The realization is minimal.
- (2) Every constraint code  $C_i$  is both trim and proper.
- (3) Every state space  $S_j$  is isomorphic to the “past-induced state space”  $\mathcal{C}_{|\mathcal{A}^{\mathcal{P}_j}} / \mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}}$ , and the set of  $\mathbf{a}^{\mathcal{P}_j}$  that reach a given state  $s_j \in S_j$  is the corresponding coset of  $\mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}}$  in  $\mathcal{C}_{|\mathcal{A}^{\mathcal{P}_j}}$ .

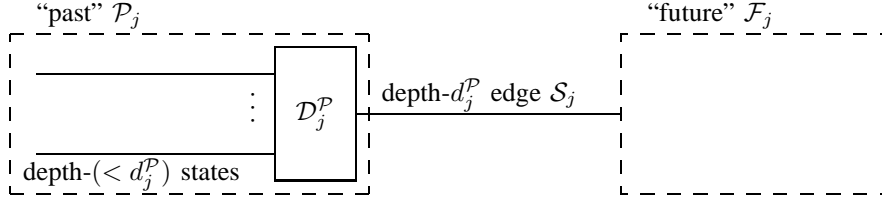


Fig. 5. Illustration of the past  $\mathcal{P}_j$  and future  $\mathcal{F}_j$  of an edge  $\mathcal{S}_j$ .

*Proof:*

(1  $\Rightarrow$  2) By Theorem 2, if any  $\mathcal{C}_i$  is not both trim and proper, then the realization is locally reducible, and thus not minimal.

(2  $\Rightarrow$  3) Let  $(\mathcal{C}^{\mathcal{P}_j})_{|\mathcal{A}^{\mathcal{P}_j}}$  denote the set of all past configurations  $\mathbf{a}^{\mathcal{P}_j} \in \mathcal{A}^{\mathcal{P}_j}$  that appear in some valid past configuration  $(\mathbf{a}^{\mathcal{P}_j}, \mathbf{s}^{\mathcal{P}_j})$ . We prove by finite induction on  $d_j^{\mathcal{P}}$  that

- (a) trim  $\Rightarrow$  every state  $s_j \in \mathcal{S}_j$  is reached by some symbol configuration  $\mathbf{a}^{\mathcal{P}_j} \in (\mathcal{C}^{\mathcal{P}_j})_{|\mathcal{A}^{\mathcal{P}_j}}$ ;
- (b) proper  $\Rightarrow$  every symbol configuration  $\mathbf{a}^{\mathcal{P}_j} \in (\mathcal{C}^{\mathcal{P}_j})_{|\mathcal{A}^{\mathcal{P}_j}}$  reaches a unique state  $s_j \in \mathcal{S}_j$ .

In other words, a trim and proper realization realizes a well-defined surjective “reaching map”  $R : (\mathcal{C}^{\mathcal{P}_j})_{|\mathcal{A}^{\mathcal{P}_j}} \rightarrow \mathcal{S}_j$ .

Clearly (a) and (b) hold if the depth of  $\mathcal{S}_j$  is 1; i.e., if  $\mathcal{P}_j$  comprises a single leaf node, representing a constraint that involves no state other than  $\mathcal{S}_j$ . Now if (a) and (b) hold for all depth- $(< d_j^{\mathcal{P}})$  edges in  $\mathcal{P}_j$ , then (a) and (b) hold for all depth- $d_j^{\mathcal{P}}$  edges  $\mathcal{S}_j$ , because, denoting the constraint code to which  $\mathcal{S}_j$  is connected in  $\mathcal{P}_j$  as  $\mathcal{D}_j^{\mathcal{P}}$  (see Fig. 5):

- (a) if  $\mathcal{D}_j^{\mathcal{P}}$  is trim, and all of its depth- $(< d_j^{\mathcal{P}})$  states are reachable, then all states  $s_j \in \mathcal{S}_j$  are reachable, since  $s_j$  occurs with some configuration of depth- $(< d_j^{\mathcal{P}})$  state values in some codeword of  $\mathcal{D}_j^{\mathcal{P}}$ ;
- (b) if  $\mathcal{D}_j^{\mathcal{P}}$  is proper, and every symbol configuration in the “past” of its depth- $(< d_j^{\mathcal{P}})$  states reaches a unique state value, then the same must be true for  $\mathcal{S}_j$ , since otherwise there would be two codewords of  $\mathcal{D}_j^{\mathcal{P}}$  that differ only on  $\mathcal{S}_j$ .

In particular, the all-zero symbol configuration  $\mathbf{0}^{\mathcal{P}_j}$  reaches only the zero state of  $\mathcal{S}_j$ .

Now, applying the same argument to  $\mathcal{F}_j$ , we conclude that all states of  $\mathcal{S}_j$  are reached by future configurations, so all valid past configurations are past projections of codewords; i.e.,  $(\mathcal{C}^{\mathcal{P}_j})_{|\mathcal{A}^{\mathcal{P}_j}} = \mathcal{C}_{|\mathcal{A}^{\mathcal{P}_j}}$ . Moreover, the unique state of  $\mathcal{S}_j$  that can be reached by the all-zero future symbol configuration  $\mathbf{0}^{\mathcal{F}_j}$  is the zero state. Hence the set of configurations in  $\mathcal{C}_{|\mathcal{A}^{\mathcal{P}_j}}$  that reach the zero state in  $\mathcal{S}_j$  is precisely the cross-section  $\mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}} = \{\mathbf{a}^{\mathcal{P}_j} \in \mathcal{C}_{|\mathcal{A}^{\mathcal{P}_j}} \mid (\mathbf{a}^{\mathcal{P}_j}, \mathbf{0}^{\mathcal{F}_j}) \in \mathcal{C}\}$ .

By linearity, the “reaching map”  $R : \mathcal{C}_{|\mathcal{A}^{\mathcal{P}_j}} \rightarrow \mathcal{S}_j$  is a homomorphism. We have shown that it is surjective and has kernel  $\mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}}$ . Thus, by the fundamental theorem of homomorphisms,  $\mathcal{S}_j \cong \mathcal{C}_{|\mathcal{A}^{\mathcal{P}_j}} / \mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}}$ . Moreover, since the subset of configurations in  $\mathcal{C}_{|\mathcal{A}^{\mathcal{P}_j}}$  that reach the zero state in  $\mathcal{S}_j$  is  $\mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}}$ , by linearity the subset of configurations in  $\mathcal{C}_{|\mathcal{A}^{\mathcal{P}_j}}$  that reach an arbitrary state in  $\mathcal{S}_j$  is the corresponding coset of  $\mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}}$ .

(3  $\Rightarrow$  1) Since (3) is equally valid for  $\mathcal{F}_j$  and  $\mathcal{P}_j$ , it follows that  $\mathcal{S}_j$  is also isomorphic to the “future-induced state space”  $\mathcal{C}_{|\mathcal{A}^{\mathcal{F}_j}} / \mathcal{C}_{:\mathcal{A}^{\mathcal{F}_j}}$ , and the set of  $\mathbf{a}^{\mathcal{F}_j} \in \mathcal{C}_{|\mathcal{A}^{\mathcal{F}_j}}$  that reach a given state  $s_j \in \mathcal{S}_j$  is the corresponding coset of  $\mathcal{C}_{:\mathcal{A}^{\mathcal{F}_j}}$  in  $\mathcal{C}_{|\mathcal{A}^{\mathcal{F}_j}}$ .

Thus if  $\{\mathbf{r}^{\mathcal{P}_j}(s_j) : s_j \in \mathcal{S}_j\}$  is a set of coset representatives for the cosets of  $\mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}}$  in  $\mathcal{C}_{|\mathcal{A}^{\mathcal{P}_j}}$ , and  $\{\mathbf{r}^{\mathcal{F}_j}(s_j) : s_j \in \mathcal{S}_j\}$  is a set of coset representatives for the cosets of  $\mathcal{C}_{:\mathcal{A}^{\mathcal{F}_j}}$  in  $\mathcal{C}_{|\mathcal{A}^{\mathcal{F}_j}}$ , then the set of all symbol configurations  $(\mathbf{a}^{\mathcal{P}_j}, \mathbf{a}^{\mathcal{F}_j}) \in \mathcal{A}^{\mathcal{P}_j} \times \mathcal{A}^{\mathcal{F}_j}$  that pass through a given state  $s_j \in \mathcal{S}_j$  is  $(\mathbf{r}^{\mathcal{P}_j}(s_j), \mathbf{r}^{\mathcal{F}_j}(s_j)) + \mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}} \times \mathcal{C}_{:\mathcal{A}^{\mathcal{F}_j}}$ , a coset of  $\mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}} \times \mathcal{C}_{:\mathcal{A}^{\mathcal{F}_j}}$ . The code  $\mathcal{C}$  is thus the union of these  $|\mathcal{S}_j|$  disjoint cosets. (An easy corollary is that  $\mathcal{S}_j$  is also isomorphic to  $\mathcal{C} / (\mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}} \times \mathcal{C}_{:\mathcal{A}^{\mathcal{F}_j}})$ .)

This shows that every realization of  $\mathcal{C}$  with the same graph topology must have at least  $|\mathcal{S}_j|$  states at time  $j$ , because  $(\mathbf{a}^{\mathcal{P}_j}, \mathbf{a}^{\mathcal{F}_j}) \in \mathcal{A}^{\mathcal{P}_j} \times \mathcal{A}^{\mathcal{F}_j}$  is in  $\mathcal{C}$  if and only if  $\mathbf{a}^{\mathcal{P}_j}$  and  $\mathbf{a}^{\mathcal{F}_j}$  are in corresponding cosets of  $\mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}}$  and  $\mathcal{C}_{:\mathcal{A}^{\mathcal{F}_j}}$ . Thus the realization is minimal.  $\square$

This proof shows constructively that in a minimal cycle-free realization there is a one-to-one match of the cosets of  $\mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}}$  in  $\mathcal{C}_{|\mathcal{A}^{\mathcal{P}_j}}$  with the cosets of  $\mathcal{C}_{:\mathcal{A}^{\mathcal{F}_j}}$  in  $\mathcal{C}_{|\mathcal{A}^{\mathcal{F}_j}}$  such that every  $\mathbf{a}^{\mathcal{P}_j}$  in a given coset can be followed by every  $\mathbf{a}^{\mathcal{F}_j}$  in its matching coset to form a codeword in  $\mathcal{C}$ , and no other pairs  $(\mathbf{a}^{\mathcal{P}_j}, \mathbf{a}^{\mathcal{F}_j})$  are codewords in  $\mathcal{C}$ . This is the essence of the State Space Theorem [27], [8], [4]. This theorem also yields the SST isomorphisms

$$\frac{\mathcal{C}_{|\mathcal{A}^{\mathcal{P}_j}}}{\mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}}} \cong \frac{\mathcal{C}_{|\mathcal{A}^{\mathcal{F}_j}}}{\mathcal{C}_{:\mathcal{A}^{\mathcal{F}_j}}} \cong \frac{\mathcal{C}}{\mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}} \times \mathcal{C}_{:\mathcal{A}^{\mathcal{F}_j}}},$$

and the fact that a minimal state space  $\mathcal{S}_j$  is isomorphic to any of these quotients.

In general, we will say that a realization is *trim* (resp. *proper*) if every constraint code is trim (resp. proper). We will further define a realization to be *state-trim* if the projection of the full behavior  $\mathfrak{B}$  onto every state space  $\mathcal{S}_j$  is surjective (i.e., equal to  $\mathcal{S}_j$ ), and *branch-trim* if the projection of  $\mathfrak{B}$  onto every constraint code  $\mathcal{C}_i$  is surjective. Thus “trim” and “proper” are local properties, since they involve only local constraint codes, whereas “state-trim” and “branch-trim” are global, since they involve the full behavior. Notice that the second part of this proof shows that for a finite cycle-free realization, “trim” implies “state-trim” and “branch-trim.”

Koetter [16, Lemma 6] shows that a state space  $\mathcal{S}_j$  may be reduced by merging if the dual behavior  $\mathfrak{B}^\circ$  is not state-trim at  $\mathcal{S}_j$ , a global condition that yields our local result “improper”  $\Rightarrow$  “locally reducible by merging” as a corollary.

Kashyap [13] has shown that a given tree realization may be minimized by first making it state-trim (“essential”), and then merging each state space  $\mathcal{S}_j$  into its quotient  $\mathcal{S}_j/W_j$ , where  $W_j$  is the subspace of  $\mathcal{S}_j$  that is reached by configurations in the zero-state subcode  $\mathcal{C}_{:\mathcal{A}^{\mathcal{P}_j}} \times \mathcal{C}_{:\mathcal{A}^{\mathcal{F}_j}}$ . However, because the

latter step requires computing the full behavior, this procedure is not very efficient.

From Theorem 3, it follows that if a linear or group realization on a finite cycle-free graph is nonminimal, then it may be made minimal by a finite sequence of local reductions. In other words, there exists a straightforward and efficient finite algorithm for minimizing any given linear or group realization on any finite cycle-free graph.

From Theorems 1 and 3, it follows that the dual to a finite cycle-free minimal realization is minimal. The dual minimal state spaces are thus the dual groups or spaces to the primal minimal state spaces; *i.e.*, we obtain the dual state space theorem [9] as a corollary.

Finally, we remark that the “shortest basis” approach to minimality that is often used for conventional linear trellis realizations (see [6] and references therein) cannot be extended to general cycle-free realizations, because it relies on the “product factorization,” which generally does not exist for general cycle-free graphs; see Appendix A. In this respect, the “trim and proper” approach to minimality may be regarded as more basic than the “shortest basis” approach.

#### IV. OBSERVABILITY AND CONTROLLABILITY

In this section, we define a realization to be controllable if its constraints are independent, and give a simple test for controllability. We show that a realization is uncontrollable if and only if the dual realization is unobservable, and that in either case such a realization is locally reducible.

We give conditions for controllability and observability for finite cycle-free realizations, generator realizations, parity-check realizations and tail-biting trellis realizations. A tail-biting trellis realization is uncontrollable if and only if its behavior consists of disconnected subbehaviors, as with classical uncontrollable conventional trellis realizations; however, on more general graphs, uncontrollability does not necessarily imply disconnectedness. We show that the support of an unobservable configuration must be a cycle or a generalized cycle.

We conclude by observing that for iterative decoding, unobservable realizations seem clearly undesirable; however, uncontrollability may not hurt, and may even be advantageous.

##### A. Independent constraints and controllability

The behavior  $\mathfrak{B}$  of a linear realization is defined by a system of linear homogeneous constraint equations. Specifically, each constraint code  $\mathcal{C}_i$ ,  $i \in \mathcal{I}_c$ , of the realization may be specified by  $\dim \mathcal{C}_i^\perp$  independent equations, so  $\mathfrak{B}$  is the solution space of  $\sum_{i \in \mathcal{I}_c} \dim \mathcal{C}_i^\perp$  homogeneous equations. We say that the realization has *independent constraints* if the system of all these equations is linearly independent.

We then have the following fundamental duality theorem:

**Theorem 4 (Observability/controllability duality).** A normal linear or group realization has independent constraints if and only if its dual realization is observable.

*Proof:* Suppose the system of constraints is linearly dependent; *i.e.*, there exists a nontrivial linear combination

$\sum_i (\hat{\mathbf{a}}^{(i)}, \hat{\mathbf{s}}^{(i)}) = (\mathbf{0}, \mathbf{0})$  with  $(\hat{\mathbf{a}}^{(i)}, \hat{\mathbf{s}}^{(i)}) \in (\mathcal{C}_i)^\perp$  for all  $i$ . Since each symbol variable is involved in exactly one constraint code, this implies  $\hat{a}_k = 0$  for all  $k$ . Moreover, since each state variable is involved in exactly two constraint codes, this implies that the two corresponding values  $\hat{s}_j$  and  $\hat{s}'_j$  satisfy  $\hat{s}_j = -\hat{s}'_j$ . But all this implies that  $(\mathbf{0}, \hat{\mathbf{s}})$  is a valid configuration in the dual realization (see Fig. 1(b)). Thus the dual realization is unobservable. The converse follows by reversing these arguments.  $\square$

For classical conventional state (trellis) realizations, the dual property to observability is called controllability. We will therefore call a linear or group realization that has independent constraints *controllable*. We use this term even though (a) the classical definition arose at a time when linear system theory was embedded in control theory, which is not our context here; (b) although the behavior of uncontrollable tail-biting trellis realizations is similar to that of uncontrollable conventional trellis realizations, as we will see in Theorem 10, such uncontrollability properties do not necessarily extend to realizations on general graphs, as we will see in Section IV-G. The reader who is not so interested in continuity with classical linear system theory may therefore prefer terms like “one-to-one” and “independent” to “observable” and “controllable.”

**Example 2** (*cf.* [18, Fig. 5]). The binary linear  $(3, 2, 2)$  block code  $\mathcal{C} = \{000, 110, 101, 011\}$  may be realized by the linear tail-biting trellis realization shown in Fig. 6(a), with three binary symbol alphabets, three binary state spaces  $\mathcal{S}_0 = \mathcal{S}_1 = \mathcal{S}_2 = \{0, 1\}$ , and three constraint codes  $\mathcal{C}_0 = \mathcal{C}_1 = \mathcal{C}_2 = \{000, 110, 101, 011\}$ , where  $\mathcal{C}_2$  involves  $\mathcal{S}_2$  and  $\mathcal{S}_0$ . Because the all-zero codeword  $\mathbf{0}$  is realized by two configurations, this realization is unobservable, with  $\dim \mathcal{S}^u = 1$ .



Fig. 6. (a) Unobservable and (b) uncontrollable tail-biting realizations of orthogonal codes.

The dual linear tail-biting trellis realization to that of Fig. 6(a) uses the same variable alphabets and the orthogonal constraint codes  $\mathcal{C}_0^\perp = \mathcal{C}_1^\perp = \mathcal{C}_2^\perp = \{000, 111\}$ , shown in Fig. 6(b). (Again, because the field is binary, no sign inverters are needed.) The dual realization realizes the orthogonal  $(3, 1, 3)$  code  $\mathcal{C}^\perp = \{000, 111\}$ ; however, since it is the dual to an unobservable realization, it is uncontrollable by Theorem 4. Explicitly, the three constraint equations corresponding to the branches 101 in the three primal constraint codes (*i.e.*, the constraints that form the unobservable configuration  $(\mathbf{a}, \mathbf{s}) = (\mathbf{0}, \mathbf{1})$  in the primal behavior) are dependent.  $\square$

##### B. Finite cycle-free realizations

For the finite cycle-free realizations that were considered in Section III-C, we have immediately:

**Theorem 5 (Finite cycle-free realizations).** A normal linear or group realization on a finite cycle-free graph is observable if it is proper, and controllable if it is trim.



*Proof.* From the second part of the proof of Theorem 3, part (b), if a realization on a finite cycle-free graph is proper, then every symbol configuration  $\mathbf{a} \in \mathcal{C}$  maps to a unique state configuration  $\mathbf{s} \in \mathcal{S}$ , so the realization is one-to-one, and thus observable. It then follows from the dualities of Theorems 1 and 4 that under the same conditions a trim realization is controllable.  $\square$

Thus a minimal (trim and proper) finite cycle-free linear or group realization is observable and controllable. However, the converse is not true:

**Example 1** (cont.). The trellis realization of Fig. 4(a) is observable, but improper. The dual realization of Fig. 4(c) is controllable, but not trim.  $\square$

Thus in this context properness is stronger than observability, and trimness than controllability. Properness implies invertibility not only globally, but also for fragments of a realization created by cuts. Similarly, trimness implies state reachability not only globally, but also for fragments.

### C. Controllability test

For a direct test of controllability, we regard  $\prod_i \mathcal{C}_i^\perp$  as a subspace of  $\hat{\mathcal{A}} \times \hat{\mathcal{S}} \times \hat{\mathcal{S}}$ , as in Fig. 1(b). We observe that the subspace  $\mathcal{K} = \{(\mathbf{0}, \hat{\mathbf{s}}, -\hat{\mathbf{s}}) \in \prod_i \mathcal{C}_i^\perp\}$  is isomorphic to the unobservable state configuration subspace  $\hat{\mathcal{S}}^u = \{\hat{\mathbf{s}} \in \hat{\mathcal{S}} \mid (\mathbf{0}, \hat{\mathbf{s}}, -\hat{\mathbf{s}}) \in \prod_i \mathcal{C}_i^\perp\}$  of the dual realization. Then we obtain:

**Theorem 6 (Controllability test).** The size of the behavior  $\mathfrak{B}$  of a linear or group realization with state configuration space  $\mathcal{S} = \prod_j \mathcal{S}_j$  and constraint codes  $\mathcal{C}_i$  is  $|\mathfrak{B}| = (\prod_i |\mathcal{C}_i|) |\hat{\mathcal{S}}^u| / |\mathcal{S}|$ ; or, in the linear case,  $\dim \mathfrak{B} = \sum_i \dim \mathcal{C}_i + \dim \hat{\mathcal{S}}^u - \dim \mathcal{S}$ . Thus the realization is controllable if and only if  $|\mathfrak{B}| = (\prod_i |\mathcal{C}_i|) / |\mathcal{S}| = (\prod_i |\mathcal{C}_i|) / (\prod_j |\mathcal{S}_j|)$ ; or equivalently, in the linear case, if and only if

$$\dim \mathfrak{B} = \sum_i \dim \mathcal{C}_i - \dim \mathcal{S} = \sum_i \dim \mathcal{C}_i - \sum_j \dim \mathcal{S}_j.$$

*Proof.* As in Section II-I, define  $\mathfrak{B}^\perp$  as the image of the homomorphism  $\Sigma : \prod_i \mathcal{C}_i^\perp \rightarrow \hat{\mathcal{A}} \times \hat{\mathcal{S}}$  such that  $(\hat{\mathbf{a}}, \hat{\mathbf{s}}, \hat{\mathbf{s}}') \mapsto (\hat{\mathbf{a}}, \hat{\mathbf{s}} + \hat{\mathbf{s}}')$ ; then  $\mathfrak{B}^\perp$  is the orthogonal code to  $\mathfrak{B}$ . Since  $\mathcal{K}$  is the kernel of  $\Sigma$ , we have  $\mathfrak{B}^\perp \cong (\prod_i \mathcal{C}_i^\perp) / \mathcal{K}$  by the fundamental theorem of homomorphisms, so  $|\mathfrak{B}^\perp| = (\prod_i |\mathcal{C}_i^\perp|) / |\mathcal{K}|$ . Since  $|\mathfrak{B}| = |\mathcal{A}| |\mathcal{S}| / |\mathfrak{B}^\perp|$ ,  $|\mathcal{C}_i| = |\mathcal{A}^{(i)}| |\mathcal{S}^{(i)}| / |\mathcal{C}_i^\perp|$  and  $|\mathcal{K}| = |\hat{\mathcal{S}}^u|$ , and moreover, as a consequence of the normal degree restrictions,  $\prod_i |\mathcal{A}^{(i)}| = |\mathcal{A}|$  and  $\prod_i |\mathcal{S}^{(i)}| = |\mathcal{S}|^2$ , we have  $|\mathfrak{B}| = (\prod_i |\mathcal{C}_i|) |\hat{\mathcal{S}}^u| / |\mathcal{S}|$ ; or, in the linear case,  $\dim \mathfrak{B} = \sum_i \dim \mathcal{C}_i + \dim \hat{\mathcal{S}}^u - \dim \mathcal{S}$ . Thus  $|\mathfrak{B}| \geq (\prod_i |\mathcal{C}_i|) / |\mathcal{S}|$ , with equality if and only if  $|\hat{\mathcal{S}}^u| = 1$ ; i.e., if and only if the realization is controllable.  $\square$

**Example 2** (cont.). In Fig. 6(b), we have  $\dim \mathfrak{B}^\circ = 1$ ,  $\sum_i \dim \mathcal{C}_i^\perp = 3$  and  $\sum_i \dim \hat{\mathcal{S}}_i = 3$ ; therefore by Theorem 6 this realization is uncontrollable. However, for Fig. 6(a), we have  $\dim \mathfrak{B} = 3$ ,  $\sum_i \dim \mathcal{C}_i = 6$  and  $\sum_i \dim \mathcal{S}_i = 3$ , so this realization is controllable.  $\square$

If a linear realization is both observable and controllable, then we have  $\dim \mathcal{C} = \sum_i \dim \mathcal{C}_i - \sum_j \dim \mathcal{S}_j$ , since

$\dim \mathfrak{B} = \dim \mathcal{C}$ . As Kashyap [14] observed, this implies that for minimal finite cycle-free linear realizations, the two complexity measures  $\sum_i \dim \mathcal{C}_i$  and  $\sum_j \dim \mathcal{S}_j$  are equivalent.

### D. Generator and parity-check realizations

We now investigate the observability and controllability of two standard types of realizations.

A *generator realization* of a linear code  $\mathcal{C} \subseteq \mathbb{F}^n$  is specified by a set of  $\ell$  generator  $n$ -tuples  $\mathbf{g}_i \in \mathbb{F}^n$ , such that  $\mathcal{C}$  is the set of all linear combinations  $\mathbf{a} = \sum_i \alpha_i \mathbf{g}_i$  as the coefficient  $\ell$ -tuple  $\alpha$  runs through  $\mathbb{F}^\ell$ . The generators are *linearly independent* if  $\dim \mathcal{C} = \ell$ . A corresponding generator realization has (up to)  $n$  internal replicas  $\alpha_{ik}$  of each of the  $\ell$  free coefficients  $\alpha_i$ , constrained to be equal by  $\ell$  equality constraints  $\alpha_{i1} = \dots = \alpha_{in}$ ;  $n$  external variables  $A_k$ , and  $n$  linear constraint codes that enforce the constraints  $a_k = \sum_i \alpha_{ik} g_{ik} = \sum_i \alpha_i g_{ik}$ . (If  $g_{ik} = 0$ , then the replica  $\alpha_{ik}$  may be omitted.)

A *parity-check realization* of a linear code  $\mathcal{C} \subseteq \mathbb{F}^n$  is specified by a set of  $r$  check  $n$ -tuples  $\mathbf{h}_j \in \mathbb{F}^n$ , such that  $\mathcal{C}$  is set of all  $n$ -tuples  $\mathbf{a} \in \mathbb{F}^n$  that are orthogonal to all check  $n$ -tuples  $\mathbf{h}_j$ . In other words,  $\mathcal{C}$  is the orthogonal code to the linear code  $\mathcal{C}^\perp$  generated by the  $r$  check  $n$ -tuples. The checks are linearly independent if  $\dim \mathcal{C}^\perp = r$ , or equivalently if  $\dim \mathcal{C} = n - r$ . In the corresponding parity-check realization, there are  $n$  external variables  $A_k$ ,  $n$  one-dimensional constraint codes that generate (up to)  $r$  multiples  $a_k h_{jk}$  of each of the external variables  $a_k$ , and  $r$  zero-sum constraint codes that enforce the parity checks  $0 = \sum_k a_k h_{jk}$ . (If  $h_{jk} = 0$ , then the multiple  $a_k h_{jk}$  may be omitted.)

As expected, generator and parity-check realizations are duals:

### Theorem 7 (Generator/parity-check realization duality).

Let a generator realization of a linear code  $\mathcal{C}$  be specified by  $\ell$  generator  $n$ -tuples  $\mathbf{g}_i$ ; then its dual realization is the parity-check realization specified by the same set of  $\ell$   $n$ -tuples as checks, which realizes the orthogonal code  $\mathcal{C}^\perp$ .

*Proof.* To prove that the realizations are duals, one need only check that:

- the orthogonal code to the code generated by the equality constraint  $\alpha_{i1} = \dots = \alpha_{in}$  is the code generated by the zero-sum constraint  $\sum_k \hat{\alpha}_{ik} = 0$ ;
- the constraint code defined by a check  $a_k = \sum_i \alpha_{ik} g_{ik}$  on the variables  $a_k, \alpha_{1k}, \dots, \alpha_{\ell k}$  is the dual code to the code  $\{(\hat{a}_k, -\hat{a}_k g_{1k}, \dots, -\hat{a}_k g_{\ell k}) : \hat{a}_k \in \hat{\mathcal{A}}_k\}$ .  $\square$

**Example 3** (binary Reed-Muller code). The five binary 8-tuples 11110000, 00111100, 00001111, 11000011, 01011010 form a set of five linearly dependent generators for a binary linear block code  $\mathcal{C}$  of length 8 and dimension 4, namely the  $(8, 4, 4)$  first-order Reed-Muller (RM) code. The corresponding generator realization is shown in Fig. 7(a).

Since  $\mathcal{C}$  is self-dual (i.e.,  $\mathcal{C}^\perp = \mathcal{C}$ ), the same five binary 8-tuples form a set of linearly dependent checks for  $\mathcal{C}$ . Fig. 7(b) shows the parity-check realization based on these five check 8-tuples, which is the dual to that of Fig. 7(a).  $\square$

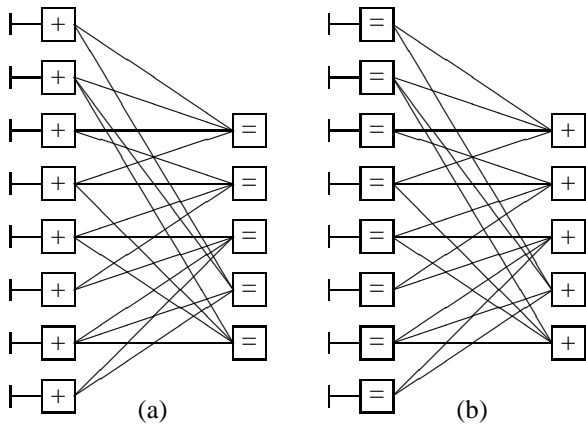


Fig. 7. (a) generator realization of  $(8, 4, 4)$  RM code; (b) parity-check realization of same code.

**Example 2 (cont.).** We note that the primal realization of Fig. 6(a) is a generator realization based on the three linearly dependent generators 110, 011, 101, whereas the dual realization of Fig. 6(b) is a parity-check realization based on the same three 3-tuples as checks.  $\square$

We now settle the observability and controllability properties of such realizations as follows:

**Theorem 8 (Observability/controllability of generator and parity-check realizations).** A parity-check realization is observable, and a generator realization is controllable. A generator realization is observable if and only if its generators are linearly independent, and a parity-check realization is controllable if and only if its checks are linearly independent.

*Proof:* In a parity-check realization, the internal variables are all multiples of external variables; therefore if all external variables are zero, then all internal variables must be zero. Thus a parity-check realization is necessarily observable. By observability/controllability duality, a generator realization must therefore be controllable.

In a generator realization, the internal variables are replicas of the  $\ell$  free coefficients, so  $\mathbf{s} \neq \mathbf{0}$  if the coefficients are nonzero. Thus there exists a nonzero configuration  $(\mathbf{0}, \mathbf{s})$ —i.e., the realization is unobservable— if and only if there exists some nontrivial linear combination of the generators that equals the zero codeword  $\mathbf{0} \in \mathcal{C}$ , which is true if and only if the generators are linearly dependent. By observability/controllability duality, a parity-check realization is controllable iff its checks are linearly independent.  $\square$

It is also interesting to determine controllability using the test of Theorem 6. In a generator realization, there are  $\ell$  equality constraints of dimension 1, and  $n$  constraint codes of total dimension  $e$ , where  $e = \dim \mathcal{S}$  is the number of one-dimensional internal variables. Thus  $\sum_i \dim \mathcal{C}_i = \ell + e$ . Since the internal variables are replicas of the  $\ell$  free coefficients, the dimension of the behavior  $\mathfrak{B}$  is  $\ell$ . Thus  $\dim \mathfrak{B} = \sum_i \dim \mathcal{C}_i - \dim \mathcal{S}$ , so a generator realization is controllable. In a parity-check realization, there are  $n$  equality constraints of dimension 1, and  $r$  single-parity-check constraints of total dimension  $e - r$ , where again  $e = \dim \mathcal{S}$  is the number of internal variables. We thus have  $\sum_i \dim \mathcal{C}_i = n + e - r$ . By

Theorem 8, the realization is observable, so  $\dim \mathfrak{B} = \dim \mathcal{C}$ . By Theorem 6, the realization is thus controllable if and only if  $\dim \mathcal{C} = n - r$ .

The fact that a parity-check realization is controllable iff its checks are linearly independent nicely illustrates our definition of “controllable” as “having independent constraints.”

#### E. Unobservable or uncontrollable $\Rightarrow$ locally reducible

We now show how, given an unobservable linear realization of a linear code  $\mathcal{C}$  on a finite graph  $\mathcal{G}$ , we may reduce the dimension of the behavior  $\mathfrak{B}$  by trimming one state space, without changing the realized code  $\mathcal{C}$ . Thus an unobservable realization is locally reducible. It follows that the dual uncontrollable realization may be locally reduced by merging the corresponding dual state space. A similar result was found by Koetter [16, Lemma 8].

For brevity and clarity, we assume that the unobservable realization is linear; the group case is similar. We select any state space  $\mathcal{S}_j$  such that  $s_j \neq 0$  in some nonzero configuration  $(\mathbf{0}, \mathbf{s}) \in \mathfrak{B}$ . We choose a basis  $\{g_{j\ell}\}$  for  $\mathcal{S}_j$  with  $g_{j1} = s_j$ . The coordinates of  $s_j$  in this basis are thus  $10\dots 0$ .

Define the subspace  $\mathcal{T}_j \subset \mathcal{S}_j$  as the set of all states in  $\mathcal{S}_j$  whose first coordinate in the chosen basis is 0, and trim the realization by restricting  $\mathfrak{B}$  to the subbehavior  $\mathfrak{B}'$  consisting of those configurations that pass through a state in  $\mathcal{T}_j$ . We then replace  $\mathcal{S}_j$  by  $\mathcal{T}_j$ , reducing the state space dimension by one.

The original unobservable configuration  $(\mathbf{0}, \mathbf{s})$  is then not in the trimmed behavior  $\mathfrak{B}'$ , since the first coordinate of  $s_j$  is 1. However, given any  $(\mathbf{a}, \mathbf{s}') \in \mathfrak{B}$ , the entire coset  $\{(\mathbf{a}, \mathbf{s}' + \alpha \mathbf{s}) : \alpha \in \mathbb{F}\}$  is in  $\mathfrak{B}$  and thus realizes  $\mathbf{a}$ . Since  $s_{j1} = 1$ , there is precisely one element in this coset whose state at time  $j$  has first coordinate zero. Thus the trimmed realization still realizes every  $\mathbf{a} \in \mathcal{C}$ .

In the dual realization, the corresponding local reduction is the merging of the states in  $\hat{\mathcal{S}}_j$  to their cosets in  $\hat{\mathcal{S}}_j / (\mathcal{T}_j)^\perp$  via the natural map. (In the coordinate representation, this amounts to deleting the first coordinate of  $\hat{\mathcal{S}}_j$ .) Since the trimmed primal realization still generates  $\mathcal{C}$ , the merged dual realization must still generate  $\mathcal{C}^\perp$ . Thus we have proved:

**Theorem 9 (Local reducibility of unobservable or uncontrollable realizations).** An unobservable linear realization on a finite graph  $\mathcal{G}$  with an unobservable configuration  $(\mathbf{0}, \mathbf{s}) \in \mathfrak{B}$  may be locally reduced by trimming any single state space in the support of  $\mathbf{s}$ . The dual uncontrollable realization may be locally reduced by the dual merging operation.  $\square$

**Example 4** (tail-biting trellis realization; from [11], [12] via [10]). The linear tail-biting trellis realization shown in Fig. 8(a) realizes the binary linear  $(5, 3)$  block code  $\mathcal{C} = \langle 01110, 00011, 10001 \rangle$ . Since it has a nonzero configuration  $(\mathbf{0}, \mathbf{s})$ , this realization is unobservable. It has five state spaces, which have been coordinatized so that the state values along the nonzero trajectory  $(\mathbf{0}, \mathbf{s})$  are either 10 or 1. It has five constraint codes  $\mathcal{C}_0 = \langle 01|1|0, 10|0|1 \rangle$ ,  $\mathcal{C}_1 = \langle 0|1|1, 1|1|0 \rangle$ ,  $\mathcal{C}_2 = \langle 0|1|01, 1|0|10 \rangle$ ,  $\mathcal{C}_3 = \langle 00|1|01, 01|1|10, 10|0|10 \rangle$ , and  $\mathcal{C}_4 = \langle 00|1|01, 01|1|00, 10|0|10 \rangle$ . This realization is the product of one-dimensional realizations of the four generators 01110, 00011, 10001, 01 110, with the indicated circular

spans, and correspondingly its behavior  $\mathfrak{B}$  has the following four generators:

$S_0$	$A_0$	$S_1$	$A_1$	$S_2$	$A_2$	$S_3$	$A_3$	$S_4$	$A_4$
00	0	0	1	1	1	11	1	00	0
00	0	0	0	0	0	00	1	01	1
01	1	0	0	0	0	00	0	00	1
10	0	1	1	0	1	01	1	10	0

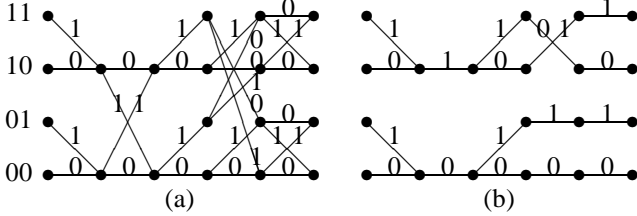


Fig. 8. (a) unobservable tail-biting trellis realization; (b) uncontrollable dual realization.

The dual realization to that of Fig. 8(a) uses the same variable alphabets and state spaces, and the orthogonal constraint codes  $\mathcal{C}_0^\perp = \langle 01|1|0, 10|0|1 \rangle$ ,  $\mathcal{C}_1^\perp = \langle 1|1|1 \rangle$ ,  $\mathcal{C}_2^\perp = \langle 0|1|01, 1|0|10 \rangle$ ,  $\mathcal{C}_3^\perp = \langle 11|0|10, 10|1|11 \rangle$ , and  $\mathcal{C}_4^\perp = \langle 10|0|10, 01|1|01 \rangle$ , as shown in Fig. 8(b). This dual realization realizes the orthogonal  $(5, 2)$  code  $\mathcal{C}^\perp = \langle 10111, 01100 \rangle$ . It is a product of realizations of generators  $\underline{10111}, \underline{01100}$  with the indicated circular spans, the second span being the entire time axis. Its behavior  $\mathfrak{B}^\circ$  is correspondingly generated by the following two generators:

$S_0$	$A_0$	$S_1$	$A_1$	$S_2$	$A_2$	$S_3$	$A_3$	$S_4$	$A_4$
01	1	0	0	0	1	01	1	01	1
10	0	1	1	1	1	11	0	10	0

By the test of Theorem 6, this realization is uncontrollable, since  $\dim \mathfrak{B}^\circ = 2$ ,  $\sum_i \dim \mathcal{C}_i^\perp = 9$  and  $\dim \hat{\mathcal{S}} = 8$ . (Notice the two disjoint subbehaviors.)

To reduce the unobservable realization of Fig. 8(a), let us trim the state space  $S_0$ . The resulting constraint codes are then  $\mathcal{C}_0 = \langle 1|1|0 \rangle$ ,  $\mathcal{C}_4 = \langle 01|1|0, 00|1|1 \rangle$ ; note that both are not trim.

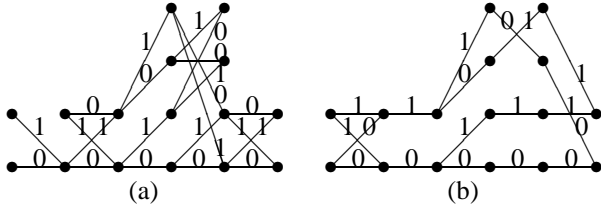


Fig. 9. State-trimmed realization, and state-merged dual realization.

The dual realization merges states in  $\hat{\mathcal{S}}_0$ , yielding the dual constraint codes  $\mathcal{C}_0^\perp = \langle 1|1|0, 0|0|1 \rangle$ ,  $\mathcal{C}_4^\perp = \langle 01|1|1, 10|0|0 \rangle$  (both improper). The resulting merged realization is shown in Fig. 9(b); it is controllable but improper.

If we trim the realization of Fig. 9(a), then we obtain a minimal (hence irreducible) realization with state space dimension profile  $(1, 0, 1, 1, 1)$ , shown in Fig. 10(a). This is a product realization with generators  $\underline{01110}, \underline{00011}, \underline{10001}$ , with the indicated circular spans.

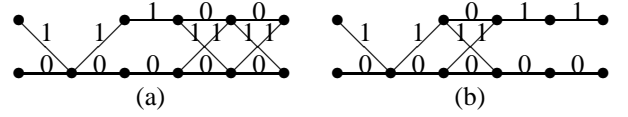


Fig. 10. Trimmed state-trimmed realization and its dual, both minimal.

The dual minimal (hence irreducible) realization is shown in Fig. 10(b). This is a product realization with generators  $\underline{01100}, \underline{10111}$ .  $\square$

#### F. Unobservable and uncontrollable tail-biting trellises

We now show that the disconnected subtrellises that can be observed in Figs. 6(b) and 8(b) are characteristic of uncontrollable linear tail-biting trellis realizations. This property is reminiscent of the similar classical uncontrollability property for conventional linear state (trellis) realizations.

**Theorem 10 (Uncontrollable tail-biting trellis realizations).** If a trim linear tail-biting trellis realization is uncontrollable, then its behavior consists of disconnected subbehaviors.

*Proof.* We first observe that a proper linear tail-biting trellis realization is unobservable if and only if it has a nonzero trajectory  $(\mathbf{0}, \mathbf{s})$  for which  $s_i \neq 0$  for all  $i$ , since properness implies that no constraint code can have a word of the form  $(0, 0, s_{i+1})$  or  $(s_i, 0, 0)$ . By trim/proper and observability/controllability duality, a trim linear tail-biting trellis realization is therefore uncontrollable if and only if its dual realization has such a nonzero trajectory  $(\mathbf{0}, \hat{\mathbf{s}})$ .

As in Section IV-E, we choose a basis  $\{\hat{g}_{i\ell}\}$  for each dual state space  $\hat{\mathcal{S}}_i$  with  $\hat{g}_{i1} = \hat{s}_i$ . The coordinates of each  $\hat{s}_i$  in this basis are then  $10\dots 0$ . Moreover, if we choose a dual basis for each primal state space  $\mathcal{S}_i$ , then the inner product of elements of  $\mathcal{S}_i$  and  $\hat{\mathcal{S}}_i$  is given by the dot product of their coordinate vectors over  $\mathbb{F}$ . Thus the set  $\hat{\mathcal{S}}_i^\perp \subseteq \mathcal{S}_i$  of primal states orthogonal to  $\hat{s}_i$  is the subspace  $\mathcal{T}_i$  of  $\mathcal{S}_i$  consisting of elements whose first coordinate is zero. Moreover, the  $|\mathbb{F}|$  cosets of  $\mathcal{T}_i$  in  $\mathcal{S}_i$  are the subsets of  $\mathcal{S}_i$  whose first coordinates are equal to a certain value of  $\mathbb{F}$ .

Now each dual constraint code  $(\mathcal{C}_i)^\perp$  contains an element  $(\hat{s}_i, 0, \hat{s}_{i+1})$  with  $\hat{s}_i$  and  $\hat{s}_{i+1}$  having coordinates  $10\dots 0$ . It follows that if  $(s_i, a_i, s_{i+1})$  is any element of  $\mathcal{C}_i$ , then (using the convention that the term  $\langle s_{i+1}, \hat{s}_{i+1} \rangle$  is negated in the inner product)

$$\begin{aligned} 0 &= \langle (s_i, a_i, s_{i+1}), (\hat{s}_i, \hat{a}_i, \hat{s}_{i+1}) \rangle \\ &= \langle s_i, \hat{s}_i \rangle + \langle a_i, \hat{a}_i \rangle - \langle s_{i+1}, \hat{s}_{i+1} \rangle \\ &= s_{i1} - s_{i+1,1}. \end{aligned}$$

It follows that in any trajectory  $(\mathbf{a}, \mathbf{s}) \in \mathfrak{B}$ , the first coordinates of the state variables must be equal:  $s_{i1} = s_{i+1,1}$ . Moreover, since this holds for all  $i$ , all first state coordinates are equal in all trajectories  $(\mathbf{a}, \mathbf{s}) \in \mathfrak{B}$ . Thus the state spaces  $\mathcal{S}_i$  are partitioned by their first coordinates into  $|\mathbb{F}|$  cosets of  $\mathcal{T}_i = \hat{\mathcal{S}}_i^\perp$ , such that state transitions are possible only between cosets with the same first state coordinates. It follows that the  $|\mathbb{F}|$  subbehaviors of  $\mathfrak{B}$  comprising trajectories that have the  $|\mathbb{F}|$  different possible values of the first state coordinate are disjoint.  $\square$

Using the coset partitioning of this proof, we define the *zero subbehavior*  $\mathfrak{B}_0$  as the zero coset of  $\mathfrak{B}$ , which contains the zero trajectory  $(\mathbf{0}, \mathbf{0})$ . Each of the  $|\mathbb{F}|$  cosets of  $\mathfrak{B}_0$  in  $\mathfrak{B}$  then corresponds to the trajectories in  $\mathfrak{B}$  that go through states with a particular value of the first coordinate.

**Example 2** (cont.). In the uncontrollable linear tail-biting trellis realization of Fig. 6(b), the zero subbehavior comprises the all-zero trajectory; the all-one trajectory is its coset.  $\square$

**Example 4** (cont.). In Fig. 8(b), the zero subbehavior comprises the two lower trajectories; the two upper trajectories comprise its coset.  $\square$

To better understand this result, and to sketch a proof of its converse, we recall the fundamental structure theorem of Koetter and Vardy [17], [18]: every linear state-trim and branch-trim tail-biting trellis realization is a product realization. In other words, the behavior  $\mathfrak{B}$  is the product of  $\dim \mathfrak{B}$  one-dimensional tail-biting trellis realizations. Proper realizations are the product of proper one-dimensional realizations. Unobservable realizations are those with  $\dim \mathfrak{B} > \dim \mathcal{C}$  generators.

One-dimensional realizations may have state support of size  $\ell < n$  or size  $n$  (called “degenerate support”). By the controllability test of Theorem 6, the behavior is controllable in the former case (since  $\sum_i \dim \mathcal{C}_i = \ell + 1$ , whereas  $\sum_i \dim \mathcal{S}_i = \ell$  and  $\dim \mathfrak{B} = 1$ ), but uncontrollable in the latter (since  $\sum_i \dim \mathcal{C}_i = \sum_i \dim \mathcal{S}_i = n$ ). It is easy to see that a product of one-dimensional realizations is uncontrollable and has disconnected subbehaviors if and only if a component has degenerate support.

**Example 4** (cont.). In Fig. 8(b), there is only one possible generator whose state support is not the entire state time axis, namely the generator of the zero subbehavior; the other generator must be taken from the nonzero coset, and necessarily has degenerate support.  $\square$

We remark that uncontrollable (but observable) tail-biting trellis realizations in general have too few “starts” and “stops” (transitions between a zero state and a nonzero state); e.g., Fig. 8(b) has only one “start” and “stop,” even though  $\dim \mathcal{C}^\perp = 2$ . Conversely, unobservable (but controllable) tail-biting trellis realizations have too many “starts” and “stops;” e.g., Fig. 8(a) has four “starts” and “stops,” even though  $\dim \mathcal{C} = 3$ . The difference is the dimension  $\dim \mathcal{S}^u$  of the dual unobservable subspace in the former case, and  $\dim \mathcal{S}^u = \dim \mathfrak{B} - \dim \mathcal{C}$  in the latter.<sup>4</sup>

### G. General unobservable and uncontrollable realizations

In this subsection, we consider a general unobservable linear realization on a finite graph  $\mathcal{G}$ , and its dual uncontrollable realization. We will see that uncontrollability does not manifest itself in such an obvious way as in the tail-biting case.

<sup>4</sup> Koetter and Vardy make the following related observations in [18, Theorem 4.6 ff.]. Define  $k = \sum_i \dim \mathcal{C}_i - \sum_i \dim \mathcal{S}_i$ . A linear tail-biting trellis realization that has no degenerate-support generating trajectories (i.e., that is controllable, so  $k = \dim \mathfrak{B}$  by Theorem 6) satisfies  $k \geq \dim \mathcal{C}$ , with equality if the realization is observable. If the realization is uncontrollable but observable, then  $k < \dim \mathcal{C}$  (i.e.,  $\sum_i \dim \mathcal{C}_i - \sum_i \dim \mathcal{S}_i < \dim \mathcal{C} = \dim \mathfrak{B}$ ).

We consider a proper unobservable realization with nonzero configuration  $(\mathbf{0}, \mathbf{s}) \in \mathfrak{B}$ . Then, by properness, the projection  $\mathbf{s}^{(i)}$  of the state configuration  $\mathbf{s}$  on the state alphabet  $\mathcal{S}^{(i)}$  of any local constraint code  $\mathcal{C}_i$  must have either 0 or 2 or more nonzero components.

We will define a *generalized cycle*<sup>5</sup> as a subgraph  $\mathcal{G}' \subseteq \mathcal{G}$ , not necessarily connected, in which all vertices have degree 2 or greater; i.e., there are no isolated (degree-0) or leaf (degree-1) vertices.<sup>6</sup> Thus the support of a nonzero configuration  $(\mathbf{0}, \mathbf{s}) \in \mathfrak{B}$  must be a generalized cycle.

Fig. 11 shows a simple generalized cycle. In general, a generalized cycle is a union of cycles.

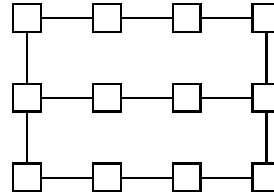


Fig. 11. Example of a generalized cycle.

Again, we may choose a basis  $\{g_{j\ell}\}$  for each state space  $\mathcal{S}_j$  in the generalized cycle with  $g_{j1} = s_j$ , so that every state variable  $s_j$  has first coordinate 1 and remaining coordinates equal to 0. Then the one-dimensional subspace of  $\mathfrak{B}$  generated by the nonzero configuration  $(\mathbf{0}, \mathbf{s})$  is the set of all  $(\mathbf{0}, \alpha \mathbf{s})$  in which the first coordinates of the state vector  $\mathbf{s}$  are all equal to  $\alpha \in \mathbb{F}$  on the generalized cycle, and all remaining state coordinates are zero. Hence the first coordinates are realized by a simple repetition realization  $\mathcal{C}$  over  $\mathbb{F}$  defined on the generalized cycle  $\mathcal{G}$  in which all constraints are equality constraints, as illustrated in Fig. 12(a).

For a realization of the dual uncontrollable code, we choose a dual basis for each dual state space  $\hat{\mathcal{S}}_j$ , so that the inner product of elements of  $\mathcal{S}_j$  and  $\hat{\mathcal{S}}_j$  is given by the dot product of their coordinate vectors over  $\mathbb{F}$ . Since all configurations  $(\hat{\mathbf{a}}, \hat{\mathbf{s}}) \in \mathfrak{B}^\circ$  in the dual realization must be orthogonal to the configuration  $(\mathbf{0}, \mathbf{s})$ , it follows that the first coordinates  $\mathbf{s}_1$  in any such configuration must lie in the orthogonal code  $\mathcal{C}^\perp$  to the repetition code  $\mathcal{C}$ . Thus  $\mathcal{C}^\perp$  is realized by the dual to the repetition realization, in which every equality constraint is replaced by a zero-sum constraint, and a sign inverter (represented by a dot) is inserted in the middle of every edge, as in Fig. 12(b).

Such a dual realization may be further simplified as follows. Since the concatenation of a degree-2 zero-sum constraint and a sign inverter is an equality constraint, and since a linear constraint surrounded by sign inverters is equivalent to the constraint without the sign inverters, we can replace a system such as that of Fig. 12(b) by an equivalent system such as that of Fig. 12(c).

<sup>5</sup> Interestingly, Chernyak and Chertkov introduce the same concept, which they call a “generalized loop,” in their loop calculus [2], [3]. They express the partition function of a finite graph as a finite series, whose leading term is a Bethe approximation, and whose higher-order terms correspond to the “generalized loops” of the graph.

<sup>6</sup> Such a graph is sometimes called a “2-core;” however, we avoid this term because a 2-core is implicitly a *maximal* connected subgraph with no leaf vertices.

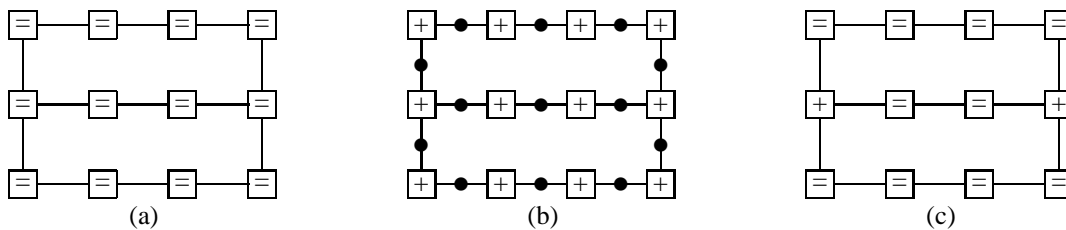


Fig. 12. Repetition realization defined on a generalized cycle, its dual, and equivalent dual.

In this example, the dual behavior consists of all configurations in which the values of the first state coordinates are constant across the top, middle and bottom chains of degree-2 vertices, say to  $\hat{a}_t, \hat{a}_m, \hat{a}_b$ , respectively, and furthermore  $\hat{a}_t + \hat{a}_m + \hat{a}_b = 0$ . Thus in general 0, 2 or 3 of the values  $\hat{a}_t, \hat{a}_m, \hat{a}_b$  are nonzero. The subgraph  $\mathcal{G}'' \subseteq \mathcal{G}'$  defined by the nonzero labels is therefore either a set of isolated vertices (*i.e.*, the graph comprising all vertices in  $\mathcal{G}'$ , but no edges), or a generalized cycle, which can be either one of the three cycles embedded in  $\mathcal{G}'$ , or the generalized cycle  $\mathcal{G}'$  itself.

In the binary case ( $\mathbb{F} = \mathbb{F}_2$ ), a zero-sum codeword must have an even number of ones. Therefore the support  $\mathcal{G}''$  of a nonzero configuration in a dual to a repetition realization must be a generalized cycle in which every vertex has even degree, which is called an *Eulerian graph*. For example, in the above example,  $\mathcal{G}''$  must be either a set of isolated vertices, or one of the three cycles embedded in  $\mathcal{G}'$ . It is easy to see that in general an Eulerian graph is a union of edge-disjoint cycles.

In summary:

**Theorem 11 (Supports of unobservable and uncontrollable behaviors).** Given a proper unobservable linear realization on a finite graph  $\mathcal{G}$  with an unobservable configuration  $(\mathbf{0}, \mathbf{s}) \in \mathfrak{B}$ , the support of  $(\mathbf{0}, \mathbf{s})$  must be a generalized cycle  $\mathcal{G}' \subseteq \mathcal{G}$ . A repetition realization as in Fig. 12(a) determines the possible values of the first state coordinates for the subspace generated by  $(\mathbf{0}, \mathbf{s})$ . Its dual realization as in Fig. 12(b) or (c) determines the possible values of the first dual state coordinates on  $\mathcal{G}'$  of all configurations in the behavior  $\mathfrak{B}^\circ$  of the dual uncontrollable realization. If  $\mathbb{F} = \mathbb{F}_2$ , then the support of any such dual configuration is an Eulerian graph  $\mathcal{G}'' \subseteq \mathcal{G}'$ .  $\square$

This theorem yields an alternative proof of Theorem 5, since it implies that a proper finite cycle-free linear realization is observable, and thus by duality a trim finite cycle-free linear realization is controllable.

If a dual constraint code  $\mathcal{C}_i^\perp$  has degree two in the generalized cycle, then it follows from this development that the first coordinates of the two dual state variables involved in  $\mathcal{C}_i^\perp$  must be equal. Thus, for any chain of degree-2 nodes in a generalized cycle, all first state coordinates must be equal throughout the chain. Thus when the generalized cycle is simply a single cycle, we retrieve the results of the previous subsection for tail-biting trellis realizations.

**Example 3 (cont.).** Consider again the unobservable generator realization of the  $(8, 4, 4)$  RM code  $\mathcal{C}$  shown in Fig. 7(a), and its uncontrollable dual parity-check realization shown in Fig. 7(b).

A repetition realization on the graph  $\mathcal{G}'$  supporting the nonzero configuration  $(\mathbf{0}, \mathbf{s})$  of Fig. 7(a) is shown in Fig. 13(a). In this case  $\mathcal{G}'$  is itself an Eulerian subgraph, with all vertices having even degree.

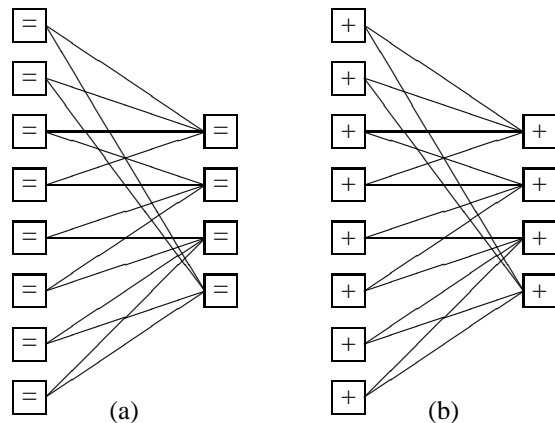


Fig. 13. (a) repetition realization on unobservable support graph  $\mathcal{G}'$ ; (b) dual realization.

The dual realization on  $\mathcal{G}'$  is shown in Fig. 13(b). Since all state variables in Fig. 7(b) are binary, all state configurations must satisfy the constraints of Fig. 13(b). There are 32 such state configurations, whose supports are the 32 possible Eulerian subgraphs of  $\mathcal{G}'$ . The 32 symbol 8-tuples corresponding to these 32 state configurations in Fig. 7(b) form a linear  $(8, 5)$  supercode  $\langle 11000000, 00110000, 00001100, 00000011, 01010101 \rangle = \{00, 11\}^4 + \{01, 10\}^4$  of the  $(8, 4, 4)$  RM code  $\mathcal{C}$ . The additional constraint imposed by the last equality constraint of Fig. 7(b) together with these constraints specifies  $\mathcal{C}$ .

We observe that in the uncontrollable realization of Fig. 13(b) there is no partition of the behavior  $\mathfrak{B}$ , the state spaces  $\mathcal{S}_j$ , or the constraint codes  $\mathcal{C}_i$  into disjoint, disconnected subsets as in the case of uncontrollable tail-biting trellis realizations (Theorem 10). In particular, all state variables are binary and are replicas of symbol variables, so for any pair of state variables that are replicas of different symbol variables, every possible pair of state values occurs in some valid configuration. Thus the effects of uncontrollability are not as easy to see as in the single-cycle case.  $\square$

#### H. Decoding unobservable or uncontrollable realizations

To decode a given linear code  $\mathcal{C}$ , we may use any realization of  $\mathcal{C}$  that we like, and any decoding algorithm based on that realization. In this subsection we will assume that the well-known sum-product (belief propagation) algorithm is used,

or any similar iterative message-passing algorithm. Would there ever be any advantage to using an unobservable or uncontrollable realization of  $\mathcal{C}$ ?

It is hard to imagine that there would be any advantage to using an unobservable realization, because in any such realization every codeword  $\mathbf{a} \in \mathcal{C}$  is realized by multiple configurations, and with the sum-product or any similar unbiased algorithm, the weights of all configurations realizing the codeword  $\mathbf{a}$  will be the same. Thus an iterative decoding algorithm will never converge to a single configuration.

On the other hand, suppose that we use an uncontrollable tail-biting trellis realization of  $\mathcal{C}$ . As we have seen, such a realization partitions into multiple disconnected subrealizations, each realizing a distinct coset of a subcode of  $\mathcal{C}$ . With the sum-product algorithm, each subrealization will be decoded independently of the others. It is possible that this could actually be helpful, because:

- decoding of the “correct” subrealization (the one containing the transmitted codeword) should be easier, because it realizes a subcode of lower rate than  $\mathcal{C}$  (*i.e.*, further from channel capacity);
- decoding of the “incorrect” subrealizations hopefully should fail in a detectable manner, because all incorrect codewords are far from the correct codeword, so none should be close to the received sequence.

So if we are able to detect and terminate nonconvergent decoding of the “incorrect” subrealizations, then we could come out ahead.

However, similar advantages could be obtained by deliberately choosing any realization of any linear subcode  $\mathcal{C}'$  of  $\mathcal{C}$ , using “translates” of this realization to realize the cosets of  $\mathcal{C}'$  in  $\mathcal{C}$ , independently decoding these translates to find the “best” codeword in each such coset, and then finally comparing these codewords to find the “best of the best.” The only advantage that we can see of using an uncontrollable tail-biting realization rather than this more general approach is that in the former case a standard sum-product algorithm may be used with little or no modification.

Furthermore, for more general uncontrollable realizations such as the redundant parity-check realization of Example 3, uncontrollability does not in general imply disconnected subrealizations. Thus we cannot straightforwardly generalize our above argument for uncontrollable tail-biting realizations.

In practice, realizations of LDPC codes with redundant parity checks have been considered for various purposes, and shown to be advantageous. In the original difference-set cyclic codes of Tanner [24], redundant checks were included for implementation symmetry; subsequently, MacKay and Davey [21] showed that redundant checks contributed to the good decoding performance of such codes. More recent theoretical and experimental work (*e.g.*, [26], [15], [28]) has shown that redundant checks can reduce the size of the fundamental polytope and therefore the number of pseudocodewords, thus improving performance.

In summary, there is theoretical and experimental evidence that redundant parity checks can be helpful. However, we are unaware of any comprehensive study of this question.

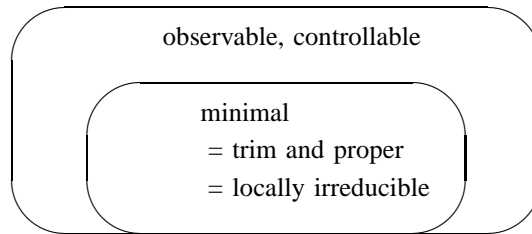
## V. CONCLUSION

For realizations on general finite graphs we have shown:

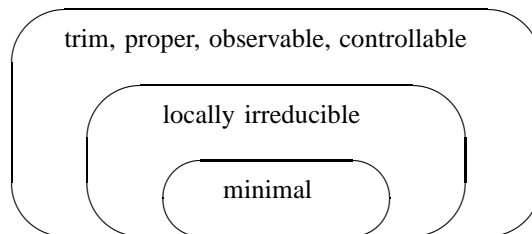
$$\text{proper} \xleftrightarrow{\text{dual}} \text{trim} \text{ and } \text{observable} \xleftrightarrow{\text{dual}} \text{controllable}.$$

Furthermore, we have shown the following relationships between the various classes of realizations. Each of the indicated containments is strict.

### Cycle-free graphs



### General finite graphs



We have shown that the behavior of uncontrollable tail-biting trellis realizations partitions into disconnected subbehaviors, as with bi-infinite conventional trellis realizations. However, this phenomenon does not necessarily occur with realizations on more general graphs. This may be related to the fact that trellis realizations always have product factorizations, whereas general realizations do not.

Finally, we have observed that whereas unobservability seems undesirable in practice, there is evidence that uncontrollability need not hurt, and may even be advantageous. We recommend further research into this question.

In a subsequent paper [10], we will focus on tail-biting trellis realizations. We find further criteria for local reducibility, and ultimately give a necessary and sufficient condition for local reducibility of tail-biting trellis realizations, under a more refined definition of local reducibility.

## ACKNOWLEDGMENTS

We are grateful to David Conti, Navin Kashyap, Yongyi Mao and Pascal Vontobel and the reviewers for comments on earlier versions of this paper.

APPENDIX A: MINIMAL CYCLE-FREE REALIZATIONS  
MAY HAVE NO PRODUCT FACTORIZATION

We now show by a simple counterexample that there is in general no “product factorization” for minimal cycle-free realizations, in contrast to trellis realizations. (A more complicated counterexample was given in [5, Section VII.A].)

**Example A** (binary SPC code). Fig. 14 shows a simple minimal cycle-free realization of the  $(3, 2, 2)$  binary linear single-parity-check (SPC) code. There are three binary-valued symbols, and three binary-valued state spaces. Three constraints are equality constraints of degree 2, and one constraint is a zero-sum constraint of degree 3.

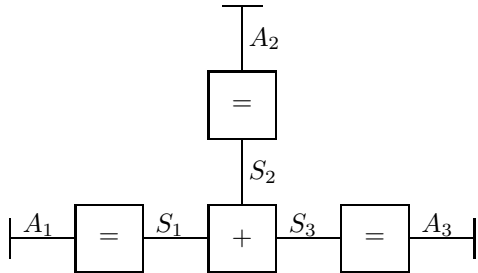


Fig. 14. Cycle-free realization of  $(3, 2, 2)$  binary linear block code.

The full behavior of this realization is generated by the two configurations below:

$$\begin{array}{c|cccccc} A_1 & S_1 & A_2 & S_2 & A_3 & S_3 \\ \hline 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{array}$$

Thus both generators are “active” for the state  $S_2$ , even though its state space dimension is 1.

By contrast, in a product factorization, the dimension of every state space is equal to the number of generators that are “active” at that time. For example, a product factorization would yield a behavior such as the one generated by the two following configurations:

$$\begin{array}{c|cccccc} A_1 & S_1 & A_2 & S_2 & A_3 & S_3 \\ \hline 1 & 1 & 1 & 10 & 0 & 0 \\ 0 & 0 & 1 & 01 & 1 & 1 \end{array}$$

But such a realization is nonminimal, as we see either by comparison with the realization above, or since its top constraint code includes the branch  $(a_2, s_2) = (0, 11)$ , and is thus improper.

By symmetry, it is clear that a similar situation occurs regardless of which two of the three nonzero configurations are selected as generators.  $\square$

APPENDIX B: SUMMARY OF KOETTER [16]

We give here a brief summary of some of the main results of Koetter [16], using the notation developed in this paper. In particular, we represent normal realizations by normal graphs, rather than by “trellis formations.”

As in Section II-I of this paper, Koetter (Lemma 1) characterizes the full behavior  $\mathfrak{B} \subseteq \mathcal{A} \times \mathcal{S}$  by its orthogonal code  $\mathfrak{B}^\perp \subseteq \hat{\mathcal{A}} \times \hat{\mathcal{S}}$ . (In our development,  $\mathfrak{B}^\perp$  is defined as the image of the homomorphism  $\Sigma : \prod_i \mathcal{C}_i^\perp \rightarrow \hat{\mathcal{A}} \times \hat{\mathcal{S}}$  that is defined by the sum map  $(\hat{a}, \hat{s}, \hat{s}') \mapsto (\hat{a}, \hat{s} + \hat{s}')$ ; this is equivalent to Koetter’s definition, because for any  $j$  at most one of  $\hat{s}_j$  or  $\hat{s}'_j$  can be involved in any  $(\mathcal{C}_i)^\perp$ .) The code  $\mathcal{C}$  realized by  $\mathfrak{B}$  is then the projection  $\mathfrak{B}|_{\mathcal{A}}$ .

Alternatively, Koetter (Theorem 2) shows that if the orthogonal code  $(\mathfrak{B}^\circ)^\perp \subseteq \mathcal{A} \times \mathcal{S}$  to the dual behavior  $\mathfrak{B}^\circ$  is constructed in the same way from  $\prod_i \mathcal{C}_i$ , then its cross-section  $((\mathfrak{B}^\circ)^\perp)|_{\mathcal{A}}$  is equal to  $\mathcal{C}$ . In our terms, a simple proof is that by projection/cross-section duality, the orthogonal code to  $((\mathfrak{B}^\circ)^\perp)|_{\mathcal{A}}$  is  $(\mathfrak{B}^\circ)|_{\hat{\mathcal{A}}} = \mathcal{C}^\perp$ . Koetter gives a direct proof, and then uses this result to prove the normal realization duality theorem (Theorem 3).

In his discussion of vertex merging, Koetter (Lemma 5) first shows that if a constraint code  $\mathcal{C}_i$  is not proper at  $S_j$ , so  $\mathcal{T} = (\mathcal{C}_i)_{:S_j} \neq \{0\}$ , and  $\mathcal{C}_{i'}$  is the other constraint code adjacent to  $S_j$ , then for any nonzero  $t \in \mathcal{T}$  we can add to  $\mathcal{C}_{i'}$  a word whose only nonzero element is  $s_j = t$  without changing the code  $\mathcal{C}$  that is realized. The proof is that  $(\mathfrak{B}^\circ)^\perp$  is unchanged, hence the dual behavior  $\mathfrak{B}^\circ$  is unchanged, hence the dual realization still realizes  $\mathcal{C}^\perp$ , hence the primal realization still realizes  $\mathcal{C}$ . This yields a merging procedure equivalent to that shown in our Fig. 2(b).

Koetter (Lemma 6) then shows that  $(\mathfrak{B}^\circ)^\perp$  contains a configuration supported by a single state space  $S_j$  if and only if the dual realization is not state-trim at  $\hat{S}_j$ , and that in this case  $S_j$  is mergeable.

In Lemmas 7 and 8 and Theorem 9, Koetter shows that the other condition for mergeability of  $S_j$  is that the dual realization is not one-to-one (*i.e.*, observable) and contains a nonzero configuration  $(0, \hat{s}) \in \mathfrak{B}^\circ$  such that  $\hat{s}_j$  is not orthogonal to  $S_j$ .

Finally, Koetter (Theorem 10) gives a polynomial-time algorithm for identifying mergeable state spaces.

## REFERENCES

- [1] A. Al-Bashabsheh and Y. Mao, "Normal factor graphs and holographic transformations," *IEEE Trans. Inf. Theory*, pp. 752–763, Feb. 2011.
- [2] M. Chertkov and V. Y. Chernyak, "Loop calculus in statistical physics and information science," *Phys. Rev. E*, vol. 73, no. 065102(R), 2006.
- [3] M. Chertkov and V. Y. Chernyak, "Loop series for discrete statistical models on graphs," *J. Stat. Mech.*, P06009, 2006.
- [4] G. D. Forney, Jr., "Codes on graphs: Normal realizations," *IEEE Trans. Inf. Theory*, vol. 47, pp. 520–548, Feb. 2001.
- [5] G. D. Forney, Jr., "Codes on graphs: Constraint complexity of cycle-free realizations of linear codes," *IEEE Trans. Inf. Theory*, vol. 49, pp. 1597–1610, Jul. 2003.
- [6] G. D. Forney, Jr., "Minimal realizations of linear systems: The 'shortest basis' approach," *IEEE Trans. Inf. Theory*, vol. 57, pp. 726–737, Feb. 2011.
- [7] G. D. Forney, Jr., "Codes on graphs: Duality and MacWilliams identities," *IEEE Trans. Inf. Theory*, vol. 57, pp. 1382–1397, Mar. 2011.
- [8] G. D. Forney, Jr. and M. D. Trott, "The dynamics of group codes: State spaces, trellis diagrams and canonical encoders," *IEEE Trans. Inf. Theory*, vol. 39, pp. 1491–1513, Sept. 1993.
- [9] G. D. Forney, Jr. and M. D. Trott, "The dynamics of group codes: Dual abelian group codes and systems," *IEEE Trans. Inf. Theory*, vol. 50, pp. 2935–2965, Dec. 2004.
- [10] H. Gluesing-Luerssen and G. D. Forney, Jr., "Local irreducibility of tail-biting trellis realizations," in preparation.
- [11] H. Gluesing-Luerssen and E. A. Weaver, "Linear tail-biting trellises: Characteristic generators and the BCJR construction," *IEEE Trans. Inf. Theory*, vol. 57, pp. 738–751, Feb. 2011.
- [12] H. Gluesing-Luerssen and E. A. Weaver, "Characteristic generators and dualization for tail-biting trellises," *IEEE Trans. Inf. Theory*, vol. 57, pp. 7418–7430, Nov. 2011.
- [13] N. Kashyap, "On minimal tree realizations of linear codes," *IEEE Trans. Inf. Theory*, vol. 55, pp. 3501–3519, Aug. 2009.
- [14] N. Kashyap, "Constraint complexity of realizations of linear codes on arbitrary graphs," *IEEE Trans. Inf. Theory*, vol. 55, pp. 4864–4877, Nov. 2009.
- [15] C. A. Kelley and D. Sridhara, "Pseudocodewords of Tanner graphs," *IEEE Trans. Inf. Theory*, vol. 53, pp. 4013–4038, Nov. 2007.
- [16] R. Koetter, "On the representation of codes in Forney graphs," in *Codes, Graphs, and Systems: A Celebration of the Life and Career of G. David Forney, Jr.* (R. E. Blahut, R. Koetter, eds.), pp. 425–450. Kluwer, 2002. On-line: [www.comm.csl.uiuc.edu/~koetter/publications/final\\_FG.ps](http://www.comm.csl.uiuc.edu/~koetter/publications/final_FG.ps).
- [17] R. Koetter and A. Vardy, "On the theory of linear trellises," in *Information, Coding and Mathematics: Festschrift on the 60th Birthday of R. J. McEliece* (M. Blaum, P. G. Farrell, H. C. A. van Tilborg, eds.), pp. 323–354. Kluwer, 2002.
- [18] R. Koetter and A. Vardy, "The structure of tail-biting trellises: Minimality and basic principles," *IEEE Trans. Inf. Theory*, vol. 49, pp. 2081–2105, Sept. 2003.
- [19] F. R. Kschischang and V. Sorokine, "On the trellis structure of block codes," *IEEE Trans. Inf. Theory*, vol. 41, pp. 1924–1937, Nov. 1995.
- [20] H.-A. Loeliger, G. D. Forney, Jr., T. Mittelholzer and M. D. Trott, "Minimality and observability of group systems," *Linear Algebra Appl.*, vol. 205–206, pp. 937–963, July 1994.
- [21] D. J. C. MacKay and M. D. Davey, "Evaluation of Gallager codes for short block length and high rate applications," in *Codes, Systems and Graphical Models* (B. Marcus and J. Rosenthal, eds.), pp. 113–130. New York: Springer, 2001.
- [22] Y. Mao and F. R. Kschischang, "On factor graphs and the Fourier transform," *IEEE Trans. Inf. Theory*, vol. 51, pp. 1635–1649, May 2005.
- [23] D. J. Muder, "Minimal trellises for block codes," *IEEE Trans. Inf. Theory*, vol. 34, pp. 1049–1053, Sept. 1988.
- [24] R. M. Tanner, "A recursive approach to low-complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, pp. 533–547, Sept. 1981.
- [25] A. Vardy, "Trellis structure of codes," in *Handbook of Coding Theory* (V. Pless and C. Huffman, eds.), pp. 1989–2118. New York: Elsevier, 1998.
- [26] P. O. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes," ArXiv: [cs/0512078](https://arxiv.org/abs/cs/0512078), Dec. 2005.
- [27] J. C. Willems, "Models for dynamics," in *Dynamics Reported* (U. Kirchgraber and H. O. Walther, eds.), vol. 2, pp. 171–269. New York: Wiley, 1989.
- [28] J. Zumbärgel, V. Skachek, and M. F. Flanagan, "On the pseudocodeword redundancy of binary linear codes," *IEEE Trans. Inf. Theory*, vol. 58, pp. 4848–4861, July 2012.

## AUTHORS' BIOGRAPHICAL SKETCHES

**G. David Forney, Jr.** received the B.S.E. degree in electrical engineering from Princeton University, Princeton, NJ, in 1961, and the M.S. and Sc.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, in 1963 and 1965, respectively.

From 1965-99 he was with the Codex Corporation, which was acquired by Motorola, Inc. in 1977, and its successor, the Motorola Information Systems Group, Mansfield, MA. Since 1996, he has been an Adjunct Professor at M.I.T.

Dr. Forney was Editor of the IEEE Transactions on Information Theory from 1970 to 1973. He has been a member of the Board of Governors of the IEEE Information Theory Society during 1970-76, 1986-94, and 2004-10, and was President in 1992 and 2008. He is currently a member of the IEEE Awards Board, and Chair of its Awards Review Committee. He has been awarded the 1970 IEEE Information Theory Group Prize Paper Award, the 1972 IEEE Browder J. Thompson Memorial Prize Paper Award, the 1990 and 2009 IEEE Donald G. Fink Prize Paper Awards, the 1992 IEEE Edison Medal, the 1995 IEEE Information Theory Society Claude E. Shannon Award, the 1996 Christopher Columbus International Communications Award, and the 1997 Marconi International Fellowship. In 1998 he received an IT Golden Jubilee Award for Technological Innovation, and two IT Golden Jubilee Paper Awards. He received an honorary doctorate from EPFL, Lausanne, Switzerland in 2007. He was elected a Fellow of the IEEE in 1973, a member of the National Academy of Engineering (U.S.A.) in 1983, a Fellow of the American Association for the Advancement of Science in 1993, an honorary member of the Popov Society (Russia) in 1994, a Fellow of the American Academy of Arts and Sciences in 1998, and a member of the National Academy of Sciences (U.S.A.) in 2003.

**Heide Gluesing-Luerssen** received the Ph.D. degree from the University of Bremen (Germany) in 1991, and the habilitation degree from the University of Oldenburg (Germany) in 2000, both in mathematics.

After a postdoctoral fellowship at the University of Bremen from 1991 to 1993, she joined the University of Oldenburg, where she served as faculty member in the Mathematics Department until 2004. From 2004 until 2006 she was a faculty member at the University of Groningen. In 2007 she joined the Department of Mathematics at the University of Kentucky. She has held visiting positions at the University of Notre Dame in 1997–99, at the University of Magdeburg in 2002, and at the University of Kentucky in 2003–04.

She currently serves as a Corresponding Editor for the *SIAM Journal on Control and Optimization*, and as an Associate Editor for *Advances in Mathematics of Communications*. Her research interests are in algebraic coding theory and codes on graphs.