Journal *of* Singularities
Volume 25 (2022), 1-29

Proc. of 16th International Workshop on Singularities, São Carlos, 2020
DOI: 10.5427/jsing.2022.25a

RIGHT NETWORK-PRESERVING DIFFEOMORPHISMS

FERNANDO ANTONELI AND IAN STEWART

Abstract. In the formal theory of networks of coupled dynamical systems, the topology of the network and a classification of nodes and arrows into specific types determines a class of 'admissible' ODEs that are compatible with the network structure. In dynamical systems theory and singularity theory, coordinate changes that preserve appropriate structures play key roles. Coordinate changes appropriate for network dynamics should, in particular, preserve admissibility. Such 'network-preserving diffeomorphisms' have been characterised completely for fully inhomogeneous networks, and for five types of action: right, left, contact, vector field, and conjugacy. Here we characterise right network-preserving diffeomorphisms for an arbitrary network. Such coordinate changes are, in particular, appropriate for the study of homeostasis, which occurs in a biological or chemical system when some output variable remains approximately constant as input parameters vary over some region.

1. Introduction

Changes of coordinates are widely used to analyse dynamical systems and their bifurcations. These coordinate changes satisfy different conditions, depending on the structure that must be preserved. Vector field coordinate changes preserve all solution trajectories. Right equivalence and contact equivalence in singularity theory preserve zero sets. In equivariant dynamics, symmetry conditions must also be imposed to preserve the symmetries of solutions.

There is an analogous theory of network dynamics, [10, 15, 28]. In this context a network is a directed graph whose nodes (vertices) and arrows (directed edges) are classified into distinct types (often called colours in the graph-theoretic literature). In the dynamic interpretation, nodes represent the variables of the system and arrows represent couplings between these variables. The network topology determines which variables affect any given variable, and the node- and arrow-types determine the form of the components of 'admissible' systems of ODEs, which respect the network structure. The crucial feature that distinguishes a network system from a general dynamical system is the presence of distinguished variables corresponding to the nodes. This allows the dynamics of nodes to be compared, so that two nodes can meaningfully be said to be synchronised or related by a phase shift. It also distinguishes variables that play different roles in the dynamics, which is common in applications.

In this paper we consider coordinate changes in network dynamics, and investigate when these changes preserve the network structure of admissible differential equations. This problem has been solved for fully inhomogeneous networks in [12], for five different types of coordinate change: left, right, contact, vector field, and conjugacy. Both vector field and conjugacy changes require diffeomorphisms, and the paper assumes throughout that the coordinate change is invertible. However, the results for left, right, and contact actions do not use invertibility. In this paper, we work throughout with maps, except in Section 9 where we prove that the inverse of a right network-preserving diffeomorphism is also right network-preserving.

2000 *Mathematics Subject Classification.* Primary 05C20; Secondary 05C25, 20B25.
Key words and phrases. Coupled Systems, Right equivalence, Homeostasis, Perfect Adaptation.

A network is fully inhomogeneous if all nodes and arrows have different types, and the corresponding admissible maps are determined by the topology of the network. Here we make a start on similar questions for general networks. Assume that the state space of the network is a real vector space V. Then a map $\Phi : V \to V$ is *right network-preserving* if, for any admissible map $F : V \to V$, the right composition $F \circ \Phi$ is admissible. We give necessary and sufficient conditions for Φ to be right network-preserving. We also make a few remarks about the analogous but often very different case of left network-preserving maps, for which $\Phi \circ F$ is admissible for any admissible F.

Motivation. This paper was motivated by a potential application, which led directly to a basic theoretical question.

This question concerns the composition of admissible maps for a network. In equivariant dynamics, the composition of two equivariant maps is always equivariant. However, the natural analog of this statement for networks is false. The next example shows that the composition of two admissible maps need not be admissible, even when they are linear. It also shows that the inverse of an invertible admissible map need not be admissible.

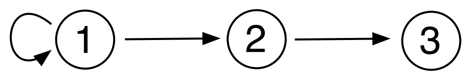

Figure 1. 3-node feedforward network.

Example 1.1. Consider the network of Figure 1. This is network number 3 in the classification of regular 3-node networks of valence ≤ 2 in [18]. The adjacency matrix is

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Assume node spaces are $\mathbb{R}$. The linear admissible maps are those of the form $aI + bA$ where I is the identity and $a, b, \in \mathbb{R}$. The square of A is

$$A^2 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

which is not admissible since its $(3, 1)$ entry is nonzero. Similarly

$$I + A = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

is admissible and invertible, but its inverse is

$$(I + A)^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -1 & 1 \end{bmatrix}$$

which is not admissible since its $(3, 1)$ entry is nonzero. $\diamond$

This failure of the composition property provides mathematical motivation for determining the right network-preserving coordinate changes for a general network — and more generally for other types of network-preserving map. In any area of mathematics, structure-preserving maps are important. So this issue is a basic problem in the formal theory of network dynamics.

The characterisation of network-preserving maps for fully inhomogeneous networks in [12] shows that this question can be answered, but the proofs are far from straightforward, involving some heavy mathematical machinery. The methods do not extend in any obvious manner to general networks. Moreover, the most obvious necessary conditions, analogous to the fully inhomogeneous case, turn out to be insufficient. We will see that the general case differs from the fully inhomogeneous case in subtle and interesting ways.

The potential application is the concept of homeostasis, which is important in biochemistry, gene regulatory networks, and many other areas. A system exhibits homeostasis if some output variable remains constant, or almost constant, when an input variable or parameter changes by a relatively large amount. Homeostasis is inherently a network concept, because it involves three distinguished types of variable: the input, the output, and everything else. Network dynamics has been applied to homeostasis [1, 11–13, 16, 27, 30].

Mathematical models of homeostasis are often constructed using a control-theoretic approach, requiring the output to be *constant* when the input lies in some interval R. Such models have *perfect homeostasis* or *perfect adaptation* [6, 29]. Here the derivative of the input-output function is identically zero on R. An alternative approach, [11–13], uses an 'infinitesimal' notion of homeostasis — namely, the derivative of the input-output function is zero at an isolated point. Near such a point, the variation of the output against input is stationary. This notion of homeostasis makes it possible to apply singularity theory. In [13] it is shown that infinitesimal homeostasis points of an input-output map are naturally classified by a slight extension of *right equivalence* of singularities [7, 8, 19]. In a network approach, it is then natural to consider coordinate changes that preserve admissibility. A characterisation of right network-preserving maps leads to a characterisation of allowable coordinate changes for input-output mappings.

An Example. Examples of fully inhomogeneous networks with nontrivial right and left network-preserving maps are given in [11]. We now give a simple example of a regular network with nontrivial right and left network-preserving maps.

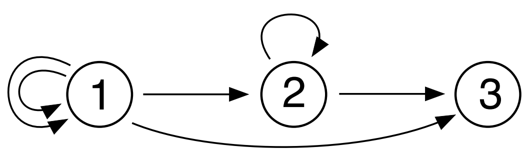

FIGURE 2. Regular 3-node feedforward network of valence 2.

Figure 2 is network 31 in the classification of regular 3-node networks of valence ≤ 2 in [18, Figure 5]. It is regular and feedforward.

Admissible maps for this example have the form

$$F(x) = \begin{bmatrix} f(x_1, \overline{x_1, x_1}) \\ f(x_2, \overline{x_1, x_2}) \\ f(x_3, \overline{x_1, x_2}) \end{bmatrix}$$

where the overlies indicate symmetry under a swap and f is an arbitrary smooth map $P \to P$ for a real vector space P.

We claim that for this network, if F and G are admissible, then so is $f \circ g$. Suppose that

$$G(x) = \begin{bmatrix} g(x_1, \overline{x_1, x_1}) \\ g(x_2, \overline{x_1, x_2}) \\ g(x_3, \overline{x_1, x_2}) \end{bmatrix}$$

Then

$$\begin{aligned}
F\circ G(x) &= F(G(x)) \\
&= F\left(\begin{bmatrix} g(x_1,\overline{x_1,x_1}) \\ g(x_2,\overline{x_1,x_2}) \\ g(x_3,\overline{x_1,x_2}) \end{bmatrix}\right) \\
&= \begin{bmatrix} f(g(x_1,\overline{x_1,x_1}),\overline{g(x_1,\overline{x_1,x_1}),g(x_1,\overline{x_1,x_1})}) \\ f(g(x_2,\overline{x_1,x_2}),\overline{g(x_1,\overline{x_1,x_1}),g(x_2,\overline{x_1,x_2})}) \\ f(g(x_3,\overline{x_1,x_2}),\overline{g(x_1,\overline{x_1,x_1}),g(x_2,\overline{x_1,x_2})}) \end{bmatrix}
\end{aligned}$$

This is admissible with

$$f(u,v,w) = f(g(w,\overline{u,v}),\overline{g(u,\overline{u,u}),g(v,\overline{u,v})})$$

For this example, any admissible map is left and right network-preserving.

This network is an example of what Rink and Sanders [26] call a *semigroup network*. That is, the admissible maps form a semigroup under composition. Although the composition of two admissible maps need not be admissible, it is in this case. This property is related to the feedforward structure, and in particular the arrow from node 1 to node 3.

Structure of the Paper. Section 2 recalls those features of the basic network formalism from [15, 28] that we need in this paper. As already remarked, we modify the definition to remove the consistency condition on head and tail nodes. We discuss domain and pullback conditions, the groupoid of the network, input sets, state type, input tuples (required by the multiarrow formalism), admissible maps, and diagonal and strongly admissible maps.

Section 3 defines network-preserving maps for the left and right actions. We discuss a complication concerning the inverse Φ^{-1} when the map Φ is a diffeomorphism. We define the left and right cores and skeletons of a network.

Section 4 states without proof the main results of the paper. These characterise right network-preserving maps Φ for any network via a set of combinatorial conditions on the components φ_c of Φ. Although we state these conditions in terms of node coordinates, they are independent of the choice of node state spaces. We also characterise right and left network-preserving diagonal maps.

Section 5 Shows how the main theorem can be used to determine right network-preserving maps for three examples. Each illustrates how general networks differ from fully inhomogeneous ones.

Section 6 introduces a simplification that makes pullback conditions more tractable, the use of 'standard order' on arrows. This allows us to reduce the study of admissible maps to pullback conditions defined by generators of the groupoid. Moreover, transition maps in the groupoid can be assumed to be the identity provided suitable identifications of state spaces are made.

Section 7 proves the main result on right network-preserving maps. Section 8 proves the main result on right network-preserving diagonal maps. Finally, Section 9 proves that if Φ is a right network-preserving map, then its inverse Φ^{-1} is also right network-preserving.

2. Network Formalism

Networks. In the literature, the networks that we consider are often referred to as 'coupled cell networks'. More recently the 'coupled cell' terminology has been dropped, in part because of potential confusion when the theory is applied to biology, and we do the same here. We therefore talk of 'nodes' rather than 'cells', but preserve some of the standard notation for consistency with the existing literature. We consider only finite networks, and employ a slight modification of the multiarrow formalism of [15]. This modification, discussed in Remark 2.2, removes the standard 'consistency condition' on head and tail nodes of equivalent arrows, replacing it by the notion of 'state equivalence', which is a consequence of the network architecture. This change has two useful effects. It resolves an ambiguity in the notion of cell-equivalence (henceforth node-equivalence), and it permits a wider class of admissible systems without affecting any of the standard theorems.

Definition 2.1. A network $\mathcal{G}$ comprises:

(a) A finite set $\mathcal{C}$ of *nodes*.
(b) An equivalence relation $\sim_C$ on $\mathcal{C}$, called *node equivalence*. The *node-type* of $c \in \mathcal{C}$ is its $\sim_C$-equivalence class.
(c) A finite set $\mathcal{A}$ of *arrows*.
(d) An equivalence relation $\sim_A$ on $\mathcal{A}$, called *arrow-equivalence*. The *arrow-type* of $e \in \mathcal{A}$ is its $\sim_A$-equivalence class.
(e) Two maps $\mathcal{H} : \mathcal{A} \to \mathcal{C}$ and $\mathcal{T} : \mathcal{A} \to \mathcal{C}$. For $e \in \mathcal{A}$ we call $\mathcal{H}(e)$ the *head* of e and $\mathcal{T}(e)$ the *tail* of e. $\diamond$

Remarks 2.2. (a) Distinct arrows can have the same heads and/or tails, giving rise to multiple arrows between the same pair of nodes. Similarly an arrow can have the same head and tail, giving a self-loop. There are several reasons for allowing these ingredients. The main one is that the theory of quotient networks is much better behaved if we do, see [15].

(b) We have omitted the traditional *consistency condition*: equivalent arrows have equivalent tails and heads. That is,

$$\mathcal{H}(e_1) \sim_C \mathcal{H}(e_2) \qquad \mathcal{T}(e_1) \sim_C \mathcal{T}(e_2)$$

for all $e_1, e_2 \in \mathcal{A}$ with $e_1 \sim_A e_2$. In it space, we introduce the notion of state equivalence, Definition 2.5. Example 2.9 below shows that removing the consistency condition permits a wider class of networks. It can be verified that this modification does not change any of the basic theorems of the subject: even the proofs remain the same [14].

(c) Informally, $c \sim_C d$ means that nodes c and d have the same 'internal dynamic', and $a \sim_A b$ means that arrows a and b represent the same type of coupling. In contrast, state equivalence $c \sim_S d$ indicates a weaker condition: nodes c and d *necessarily* have the same state space. By this we mean that these equalities are required for the admissible maps to make sense. $\diamond$

Networks in the sense of Definition 2.1 can be presented in the usual way as diagrams, using symbols such as circles and squares for nodes, and arrows with various types of shaft (solid, dashed) and arrowhead (open, closed) to distinguish node- and arrow-types.

Input Sets. Associated with each node $c \in \mathcal{C}$ is a canonical set of arrows, namely, those that represent couplings into node c.

Definition 2.3. Let $c \in \mathcal{C}$. The *input set* of c is the finite set of arrows directed to c, that is,

$$I(c) = \{e \in \mathcal{A} : \mathcal{H}(e) = c\} \tag{2.1}$$

The *extended input set* of c (of tail nodes of arrows) is

$$J(c) = \{c\} \cup I(c) \tag{2.2}$$

Here we consider node c to be an 'internal arrow' with head and tail c.

Definition 2.4. The relation $\sim_I$ of *input equivalence* on $\mathcal{C}$ is defined by $c \sim_I d$ if and only if $c \sim_C d$ and there exists a bijection $\beta : I(c) \to I(d)$ such that $i \sim_A \beta(i)$ for every $i \in I(c)$.

Any such bijection β is called an *input isomorphism* from node c to node d. The set $B(c,d)$ denotes the collection of all input isomorphisms from node c to node d.

The condition $c \sim_C d$ ensures that c and d have the same 'internal dynamic', so they respond to isomorphic inputs in the same manner.

A network is *homogeneous* if all input sets are isomorphic. $\diamond$

The union

$$\mathcal{B}_{\mathcal{G}} = \bigcup_{c,d \in \mathcal{C}} B(c,d) \tag{2.3}$$

has the structure of a groupoid [3, 17]. This is an algebraic structure obeying most axioms for a group, except that composition is not always defined. The groupoid operation on $\mathcal{B}_{\mathcal{G}}$ is composition of maps, and in general the composition $\beta\alpha$ is defined only when $\alpha \in B(a,b)$ and $\beta \in B(b,c)$ for nodes a,b,c. We call $\mathcal{B}_{\mathcal{G}}$ the *groupoid of the network* $\mathcal{G}$. When $\mathcal{G}$ is clear we write just $\mathcal{B}$.

State Type. In the formalism of [15, 28], the relation $\sim_C$ does double duty. It determines canonical identifications of node spaces, and it also implies that equivalent nodes have 'the same internal dynamic'. The consistency conditions on head and tail nodes of arrows combine these two roles, which is potentially confusing. It is also superfluous, because the required equalities of node spaces can be deduced from the network diagram or the admissible ODEs.

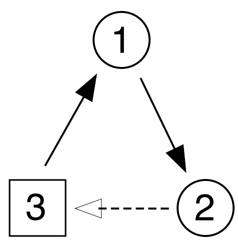

FIGURE 3. 3-node network with two node-types and two arrow-types.

For example, consider $\mathcal{G}$ with three nodes as in Figure 3. There are two node types (square, circle) and two arrow-types (solid, dashed). The $\mathcal{G}$-admissible maps are those of the form

$$F(x) = \begin{bmatrix} f_1(x_1, x_3) \\ f_1(x_2, x_1) \\ f_3(x_3, x_2) \end{bmatrix}$$

Here $f_1 \equiv f_2$ because nodes 1 and 2 are input-equivalent, but f_3 can be different.

Here f_1 appears twice; first as a map $f_1 : P_1 \times P_3 \to P_1$, and second as a map $f_1 : P_2 \times P_1 \to P_2$. For this to make sense, we need $P_1 = P_2$ and $P_3 = P_1$. That is, $P_1 = P_2 = P_3$. In this manner, the domain and pullback conditions require certain node spaces to be identical: that is, they have the same 'state type', which we define in Definition 2.5 after setting up some preliminary concepts.

In this example we could retain the consistency condition by requiring node 3 to have the same node type as nodes 1 and 2. However, we have deduced that all three nodes are state equivalent without invoking the consistency condition. Example 2.9 shows that removing the consistency condition permits a wider class of networks.

Input Tuples. The input set consists of arrows, not nodes. When defining admissible maps, we use input variables that run through the tail nodes of all input arrows to a given node. In the multiarrow formalism the same node may appear as the tail of several arrows. We must therefore consider not just input *sets* of arrows, but input *tuples* of these tail nodes. (Alternatively, we can use multisets [15].)

If $c \in \mathcal{C}$ and $I(c) = \{e_1, \ldots, e_k\}$ ordered in some manner, we write

$$T(c) = \mathcal{T}(I(c)) = (\mathcal{T}(e_1), \ldots, \mathcal{T}(e_k))$$

This is the k-tuple of tail nodes of the input arrows to node c, excluding the 'internal arrow' c. If $\mathcal{T}(e_j) = i_j \in \mathcal{C}$, then $T(c) = (i_1, \ldots, i_k)$. With this notation we also define:

$$\begin{array}{rcl} x_{T(c)} & = & (x_{i_1}, \ldots, x_{i_k}) \\ U(c) & = & (c, T(c)) = \mathcal{T}(J(c)) \\ x_{U(c)} & = & (x_c, x_{T(c)}) = (x_c, x_{i_1}, \ldots, x_{i_k}) \end{array}$$

We are now ready to define state equivalence:

Definition 2.5. Let $i, j \in \mathcal{C}$. We first define a relation $\dot{\sim}_S$ by $i \,\dot{\sim}_S\, j$ if there exists an input isomorphism $\beta \in \mathcal{B}_\mathcal{G}$ such that either

$$\exists a \in J(c) : i = \mathcal{H}(a), j = \mathcal{H}(\beta(a))$$

or

$$\exists a \in J(c) : i = \mathcal{T}(a), j = \mathcal{T}(\beta(a))$$

The relation $\dot{\sim}_S$ need not be an equivalence relation. We therefore define the *transitive closure* $\sim_S$ by

$$a \sim_S b \iff a = a_0 \,\dot{\sim}_S\, a_1 \,\dot{\sim}_S\, \ldots \dot{\sim}_S\, a_r = b$$

for suitable nodes $a_1 \ldots a_{r-1}$. If $i \sim_S j$, the nodes i, j are *state equivalent.* $\diamond$

The transitive closure combines all of the forced equalities of node spaces. Therefore nodes are state-equivalent if the conditions for admissibility force them to have the same node space.

Having the same state type must be distinguished from 'accidental' equalities of node spaces. For example, in any network digram we can choose all node spaces equal to $\mathbb{R}$. But only the equalities forced by domain and pullback conditions are necessary for *all* such choices.

The definition easily implies:

Proposition 2.6. *State equivalence $\sim_S$ refines node type $\sim_C$. Indeed, $\sim_S$ is the finest equivalence relation compatible with the class of admissible maps determined by the network.* $\square$

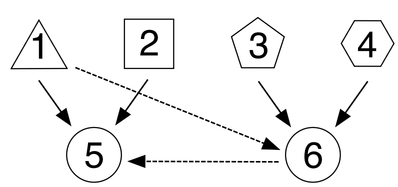

FIGURE 4. 6-node network with two arrow-types.

Example 2.7. We illustrate state type using the 6-node network of Figure 4. There are five node-types and two arrow-types. The input equivalence classes are

$$\{1\}, \{2\}, \{3\}, \{4\}, \{5, 6\}$$

We claim that all six nodes have the same state type.

There is an input isomorphism that swaps the two solid arrows inputting to node 5, so their tails are state equivalent: $1 \sim_S 2$. Similarly $3 \sim_S 4$. Since nodes 5 and 6 are input isomorphic, $5 \sim_S 6$. There is an input isomorphism sending the arrow from 1 to 5 to the arrow from 3 to 6, so $3 \sim_S 1$. Finally, any input isomorphism $I(5) \to I(6)$ maps the dotted arrow from 6 to 5 to the dotted arrow from 1 to 6, so $1 \sim_S 6$. Since $\sim_S$ is an equivalence relation, all six nodes are state equivalent. $\diamond$

Admissible Maps. We now introduce the class of maps (vector fields) that determine those ODEs that respect the architecture of a network $\mathcal{G}$.

For each node in $\mathcal{C}$ choose a *node space* P_c, the state space of the corresponding node variable. We assume that P_c is a nonzero finite-dimensional real vector space. (More generally it could be a manifold, but we do not pursue this generalisation.) As explained above, we require state equivalent nodes to have the same state space:

$$c \sim_S d \quad \Rightarrow \quad P_c = P_d$$

In this case we employ the same coordinate systems on P_c and P_d. The *(total) state space* is then

$$P = \prod_{c \in \mathcal{C}} P_c$$

with a node-based coordinate system

$$x = (x_c)_{c \in \mathcal{C}}$$

If $\mathcal{D} \subseteq \mathcal{C}$ is any finite set of nodes we write

$$P_{\mathcal{D}} = \prod_{d \in \mathcal{D}} P_d$$

and

$$x_{\mathcal{D}} = (x_{c_1}, \ldots, x_{c_\ell})$$

where $x_c \in P_c$.

For any $\beta \in B(c, d)$ we define the *pullback map*

$$\beta^* : P_{T(d)} \to P_{T(c)}$$

by

$$(\beta^* z)_{\mathcal{T}(i)} = z_{\mathcal{T}(\beta(i))} \tag{2.4}$$

for all $i \in I(c)$ and $z \in P_{T(d)}$.

We use pullback maps to relate different components of a map associated with a given network. Specifically, the class of maps that are encoded by a network is given by the following definition.

Definition 2.8. A map $F = (f_c) : P \to P$ is *$\mathcal{G}$-admissible*, or just *admissible* when $\mathcal{G}$ is clear, if:

(a) *Domain condition*: For all $c \in \mathcal{C}$ the component $f_c(x)$ depends only on the node variable x_c and the input variables $x_{\mathcal{T}(I(c))}$. That is, there exists $\hat{f}_c : P_c \times P_{\mathcal{T}(I(c))} \to P_c$ such that

$$f_c(x) \; = \; \hat{f}_c(x_c, x_{T(c)}) \tag{2.5}$$

(b) *Pullback condition*: For all $c, d \in \mathcal{C}$ and $\beta \in B(c,d)$

$$\hat{f}_d(x_d, x_{T(d)}) = \hat{f}_c(x_d, \beta^* x_{T(d)}) \tag{2.6}$$

for all $x \in P$. $\diamond$

In practice we often omit the hats from the maps $\hat{f}_c$.

In components, we write

$$F(x) = [f_1(x), \ldots, f_n(x)]^{\mathrm{T}}$$

where square brackets [ ] are used for clarity and $^{\mathrm{T}}$ appears to save space, because of the convention that components of admissible maps are shown as column vectors.

The set of smooth $\mathcal{G}$-admissible maps $F : P \to P$, denoted by $\mathcal{F}_{\mathcal{G}}(P)$, forms a real vector space.

Example 2.9. We return to example 2.7 to consider its admissible maps, and also to explain why replacing the consistency condition by state equivalence allows more networks and corresponding admissible maps.

The admissible maps for this example have the form:

$$\begin{array}{rcl}
\dot{x}_1 & = & f_1(x_1) \\
\dot{x}_2 & = & f_2(x_2) \\
\dot{x}_3 & = & f_3(x_3) \\
\dot{x}_4 & = & f_4(x_4) \\
\dot{x}_5 & = & f_5(x_5, \overline{x_1, x_2}, x_6) \\
\dot{x}_6 & = & f_5(x_6, \overline{x_3, x_4}, x_1)
\end{array}$$

Here the overline indicates symmetry in the corresponding variables (pullback from swapping the input arrows concerned). Because nodes 5 and 6 are input isomorphic they use the same function f_5. The other component functions are arbitrary, with the specified domains.

If we require the consistency condition, nodes $1, 2, 3, 4$ are all forced to have the same node type, hence also the same function f_c. That is, $f_1 = f_2 = f_3 = f_4$. However, the admissible ODE here is entirely reasonable, and there is no contradiction if we make the f_c different for $c = 1, 2, 3, 4$. We can also make some of them equal to others. The class of admissible ODEs changes for each such choice, but all of them make sense and could reasonably occur in a model.

The issue here is that the consistency condition does not just force nodes to have the same state space; it also forces them to have the same 'internal dynamic'. This confuses two distinct roles. $\diamond$

Diagonal and Strongly Admissible Maps. There is one general class of maps that compose both on the left and the right with any admissible map to yield an admissible map. These are the strongly admissible maps of [15]:

Definition 2.10. (a) A map Φ is *diagonal* if

$$\Phi(x) = [\varphi_1(x_1), \ldots, \varphi_n(x_n)]^{\mathrm{T}}$$

for suitable maps $\varphi_c : P_c \to P_c$.

(b) A map Φ is *strongly admissible* if it is diagonal, and

$\diamond$ $$c \sim_C d \Rightarrow \varphi_c = \varphi_d$$

If the φ_j are invertible, it is obvious that the set of all diagonal maps forms a group under composition.

Proposition 2.11. *For a given network $\mathcal{G}$ and choice of state space P, let $\Phi : P \to P$ be strongly admissible, and let $F : P \to P$ be any admissible map. Then both $F \circ \Phi$ and $\Phi \circ F$ are admissible.*

Proof. See [15]. The definition of strong admissibility in [15] uses $\sim_C$ where we have $\sim_S$. This is an instance of the 'dual role' previously played by node-equivalence. In this case the role of $\sim_S$ is the one that is used in the proof. □

In particular, this proposition shows that non-identity maps can be both left and right network-preserving. So the key question is whether there are any others. As shown in [11], the answer is affirmative for some networks.

3. Network-Preserving Maps

The precise characterization of network-preserving maps depends on the network, and also on the type of coordinate change concerned. The paper [12] considers only fully inhomogeneous networks, for five distinct actions (left, right, contact, conjugacy and vector field). Here we consider general networks, but only two types of coordinate change: left and right. We focus mainly on right network-preserving coordinate changes.

Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a smooth map, and let $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ be a map.

(a) The *right action* of Φ transforms F into $G(x) = F \circ \Phi(x)$.
(b) The *left action* of Φ transforms F into $G(x) = \Phi \circ F(x)$.

Definition 3.1. Let $\mathcal{G}$ be a network and P be a state space for $\mathcal{G}$. Recall that $\mathcal{F}_{\mathcal{G}}(P)$ is the space of smooth $\mathcal{G}$-admissible maps.

(a) A *right network-preserving map* is a map $\Phi : P \to P$ such that $F \circ \Phi$ is admissible for all $F \in \mathcal{F}_{\mathcal{G}}(P)$. We denote the set of all right network-preserving maps by $\mathcal{D}^R_{\mathcal{G}}(P)$.
(b) A *left network-preserving map* is a map $\Phi : P \to P$ such that $\Phi \circ F$ is admissible for all $F \in \mathcal{F}_{\mathcal{G}}(P)$. We denote the set of all left network-preserving maps by $\mathcal{D}^L_{\mathcal{G}}(P)$.

We often omit the subscript $\mathcal{G}$ when the network is clear. ◇

The next proposition states a basic property of network-preserving maps. Its proof is trivial, but it raises a difficult issue.

Proposition 3.2. *The sets $\mathcal{D}^R_{\mathcal{G}}(P)$ and $\mathcal{D}^L_{\mathcal{G}}(P)$ are semigroups under composition of maps.* □

Another obvious but useful result is:

Proposition 3.3. *Every right or left network-preserving map is admissible.*

Proof. Compose with the identity map, which is admissible. □

Invertibility. Since diffeomorphisms are invertible, it seems plausible that when Φ is a diffeomorphism, the semigroups $\mathcal{D}^R_{\mathcal{G}}(P)$ and $\mathcal{D}^L_{\mathcal{G}}(P)$ are actually groups. This would be the case if Φ network-preserving implies Φ^{-1} network-preserving. However, because the space of admissible maps is infinite-dimensional, this is not obvious.

There are several ways to get round this issue. In [12] it was proved that, for the five actions considered there, and for fully inhomogeneous networks, it is enough to assume only the conditions on Φ. The inverse Φ^{-1} automatically has the required properties. This was proved using G-structures, which 'linearise' the conditions defining the diffeomorphisms.

An alternative is to build invertibility into the definition of 'network-preserving', requiring both Φ and Φ^{-1} to be right network-preserving. However, this can cause difficulties because properties of Φ need not transfer automatically to Φ^{-1}. Also, it excludes non-invertible maps.

We prefer to avoid both of these approaches, because in the right network-preserving case we can (eventually) prove that when Φ is a diffeomorphism, $\mathcal{D}^R_{\mathcal{G}}(P)$ is a group, using groupoid properties and diagram-chasing. See Section 9. However, we do not currently know whether Φ being left network-preserving implies that Φ^{-1} is left network-preserving.

Cores, Skeletons and Domain Conditions. Recall [12, Definition 2.1] that a network $\mathcal{G}$ is *fully inhomogeneous* if distinct arrows and nodes are inequivalent. Three distinguished subnetworks of any fully inhomogeneous network $\mathcal{G}$ are defined in [12]. These determine certain classes of admissible maps.

Definition 3.4. Let $\mathcal{G}$ be a fully inhomogeneous network.

(a) The *left core* $\mathcal{G}^{\mathrm{L}}$ is the network whose nodes are the nodes of $\mathcal{G}$ and whose arrows are the arrows $j \Longrightarrow i$ in $\mathcal{G}$ that satisfy: for every diagram in $\mathcal{G}$ of the form

$$\begin{tikzcd} k \arrow[d] & \\ j \arrow[r, Rightarrow] & i \end{tikzcd} \quad \text{there exists an arrow such that} \quad \begin{tikzcd} k \arrow[d] \arrow[dr] & \\ j \arrow[r, Rightarrow] & i \end{tikzcd}$$

(b) The *right core* $\mathcal{G}^{\mathrm{R}}$ is the network whose nodes are the nodes of $\mathcal{G}$ and whose arrows are the arrows $j \Longrightarrow i$ in $\mathcal{G}$ that satisfy: for every diagram in $\mathcal{G}$ of the form

$$\begin{tikzcd} j \arrow[r, Rightarrow] & i \arrow[d] \\ & k \end{tikzcd} \quad \text{there exists an arrow such that} \quad \begin{tikzcd} j \arrow[r, Rightarrow] \arrow[dr] & i \arrow[d] \\ & k \end{tikzcd}$$

$\diamond$

Heuristically, the $\Longrightarrow$ arrow corresponds to Φ and the $\downarrow$ arrow corresponds to F. The $\searrow$ arrow corresponds to their composition in the order indicated in the diagram. For the right core, this is $F\circ\Phi$, and admissibility requires the $\searrow$ to exists to obtain the correct domain condition. For the left core the same goes for $\Phi\circ F$.

In the language of cores, we can restate the main result [12, Theorem 3.4] on right and left network-preserving maps as:

Theorem 3.5. *Let $\mathcal{G}$ be fully inhomogeneous. The left network-preserving maps for $\mathcal{G}$ are precisely the admissible maps for $\mathcal{G}^{\mathrm{L}}$. The right network-preserving maps for $\mathcal{G}$ are precisely the admissible maps for $\mathcal{G}^{\mathrm{R}}$.* $\square$

(The cited paper assumes that Φ is a diffeomorphism, but this property is not used in the proofs.)

We can characterise the domain conditions for both right and left network-preserving maps in terms of the corresponding cores for a related fully inhomogeneous network, which we now define.

Definition 3.6. Let $\mathcal{G}$ be any network.

(a) The *skeleton* $\mathcal{G}^\diamond$ of $\mathcal{G}$ is the network with the same nodes as $\mathcal{G}$, but with all self-loops deleted, the set of all arrows from i to j replaced by a single arrow (for each pair i, j of distinct nodes), and all nodes and arrows are deemed to have different types.

It is fully inhomogeneous. Therefore we can also define:

(b) The *right skeleton* $\mathcal{G}^{\diamond\mathrm{R}}$ is the right core of the skeleton of $\mathcal{G}$; that is, $(\mathcal{G}^\diamond)^{\mathrm{R}}$.

(c) The *left skeleton* $\mathcal{G}^{\diamond\mathrm{L}}$ is the left core of the skeleton of $\mathcal{G}$; that is, $(\mathcal{G}^\diamond)^{\mathrm{L}}$. $\diamond$

Figure 5 shows a 6-node network and its right skeleton.

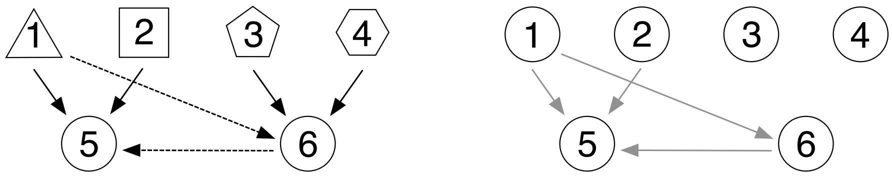

FIGURE 5. (*Left*) A 6-node network with two input types and nontrivial vertex symmetries. (*Right*) its right skeleton. All arrows are by definition distinct. Nodes 1, 2, 3, 4 are state-equivalent but not node-equivalent.

4. MAIN RESULTS

Now we are ready to state the main results of this paper. Some definitions are given later, as are all proofs. We consider only the right action of the map Φ. We start with a characterisation of the domain condition for right network-preserving maps.

Henceforth we omit the composition sign $\circ$ when composing maps.

Theorem 4.1. *Let $\mathcal{G}$ be any network and let $\Phi : P \to P$. Then the following are equivalent:*

(a) *For all $\mathcal{G}$-admissible F the map $F\Phi$ satisfies the domain conditions for $\mathcal{G}$-admissibility.*
(b) *Φ is $\mathcal{G}^{\diamond\mathrm{R}}$-admissible.*

Corollary 4.2. *Every right network-preserving map is both $\mathcal{G}$-admissible and $\mathcal{G}^{\diamond\mathrm{R}}$-admissible.*

Remark 4.3. The analogue of Theorem 4.1 for left network-preserving maps and the left core is also valid. The proof is quite different, and will not be given here. $\diamond$

We write $\asymp$ to indicate equality of $B^*(c,c)$-orbits, where the group $B^*(c,c)$ consists of all β^* for $\beta \in B(c,c)$ acting on $U(c)$. This group is anti-isomorphic to $B(c,c)$. We can use it to characterise right network-preserving maps precisely:

Theorem 4.4. *A map Φ is right network-preserving if and only if it is $\mathcal{G}^{\diamond\mathrm{R}}$-admissible and, for every pair of nodes $c \sim_I d$, every $\beta \in B(c,d)$, and every $x_{U(d)} \in P_{U(d)}$,*

$$\beta^* \hat{\Phi}_{U(d)}(x_{U(d)}) \asymp \hat{\Phi}_{U(c)} \beta^*(x_{U(d)}) \qquad \forall x_{U(d)} \in P_{U(d)} \tag{4.1}$$

Equivalently, if variables are in standard order, then for $c \neq d$

$$\hat{\Phi}_{U(d)}(x_{U(d)}) \asymp \hat{\Phi}_{U(c)}(x_{U(d)}) \qquad \forall x_{U(d)} \in P_{U(d)} \tag{4.2}$$

but when $d = c$ we also require

$$\alpha^* \hat{\Phi}_{U(c)}(x_{U(c)}) \asymp \hat{\Phi}_{U(c)} \alpha^*(x_{U(c)}) \qquad \forall x_{U(c)} \in P_{U(c)} \tag{4.3}$$

for every $\alpha^* \in B^*(c,c)$.

Remarks 4.5. (a) For each c, d, the above conditions impose conditions on the components φ_c, φ_d on possibly overlapping tuples of nodes $U(c), U(d)$. The necessary and sufficient condition for Φ to be right network-preserving is that *all* of these conditions on its components are simultaneously satisfied. Disentangling the combinatorial implications of those conditions is not entirely straightforward, because invariance under the appropriate vertex groups introduces equalities up to the vertex group action, denoted $\asymp$. Examples are given in Section 5 to show that the computations are routine in any specific case.

(b) Conditions (4.1), (4.2),and (4.2) are stated in terms of input tuples $x_{U(c)}, x_{U(d)}$. However, they are independent of the choice of node spaces and are thus intrinsic to the network.

(c) A natural question is whether a purely graph-theoretic characterisation of right network-preserving maps exists, analogous to the right core characterisation for fully inhomogeneous networks. Given a network $\mathcal{G}$, does there exist a related network $\mathcal{G}^*$ whose admissible maps are precisely the network-preserving maps for $\mathcal{G}$? In our examples, the answer is affirmative, but this question remains open in general. $\diamond$

Theorem 4.6. *If Φ is right network-preserving then Φ^{-1} is right network-preserving.*

A similar characterisation for the left action has not yet been investigated. Nevertheless, we can fully characterize the right and left network-preserving diagonal maps.

Theorem 4.7. *A diagonal map Φ is right network-preserving if and only if*

$$c \sim_S d \implies \varphi_c = \varphi_d \tag{4.4}$$

Equivalently, a diagonal map is right network-preserving if and only if it is strongly admissible.

The left network-preserving case is different:

Theorem 4.8. *A diagonal map Φ is left network-preserving if and only if*

$$c \sim_I d \implies \varphi_c = \varphi_d \tag{4.5}$$

Remark 4.9. One reason why these cases differ is the role of vertex groups. Suppose that F is invariant under a group action and Φ is diagonal. Then ΦF is also invariant under the group action, but $F\Phi$ need not be, unless the components of Φ are identical on group orbits. However, this is not the full story. $\diamond$

5. Three Examples

Before proceeding to the proofs of the above results, we use equation (4.1) in the equivalent form (4.2) + (4.3) to perform calculations in examples. These examples show how to use Theorem 4.4 to characterise right network-preserving maps for specific networks.

Three-Node Example. Consider $\mathcal{G}$ with three nodes as in Figure 3. There are two arrow-types. Every $\mathcal{G}$-admissible map takes the form

$$F(x) = \begin{bmatrix} f_1(x_1, x_3) \\ f_1(x_2, x_1) \\ f_3(x_3, x_2) \end{bmatrix}$$

Here $f_1 \equiv f_2$ because nodes 1 and 2 are input-equivalent.

The right skeleton $\mathcal{G}^{\diamond\mathrm{R}}$ is trivial, comprising only the nodes (and their tacit internal 'arrows'). Therefore the $\mathcal{G}^{\diamond\mathrm{R}}$-admissible maps have the form

$$\Phi(x) = \begin{bmatrix} \varphi_1(x_1) \\ \varphi_2(x_2) \\ \varphi_3(x_3) \end{bmatrix} \tag{5.1}$$

The intersection of these two spaces (sufficient condition as in Corollary 7.2) consists of all

$$\Phi(x) = \begin{bmatrix} \varphi_1(x_1) \\ \varphi_1(x_2) \\ \varphi_3(x_3) \end{bmatrix}$$

We now show these conditions are *not* sufficient. This can be seen by direct calculation, but we use the general characterisation above.

A set of representatives is $\mathcal{R} = \{1, 3\}$.

All vertex groups are trivial, so 'same orbit' $\asymp$ becomes equality for all values of the variables; that is, $\equiv$.

There is one nontrivial input isomorphism; namely

$$\beta : J(2) \to J(1)$$

with 'partial permutation' formula

$$\begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$$

and we use this to identify $P_{J(2)}$ with $P_{J(1)}$.

Condition (4.3) is trivial since $\alpha = \mathrm{id}$.

Condition (4.2) becomes:

$$\hat{\Phi}_{J(2)}(x_{J(2)}) = \hat{\Phi}_{J(1)}(x_{J(2)})$$

That is,

$$\begin{bmatrix} \varphi_2(x_2, x_1) \\ \varphi_1(x_2, x_1) \end{bmatrix} \equiv \begin{bmatrix} \varphi_1(x_2, x_1) \\ \varphi_3(x_2, x_1) \end{bmatrix}$$

Here we have not used the extra information on the domains of the φ_j stated in (5.1) in order to illustrate how the general formalism works here.

Since $\asymp$ has become $\equiv$ in this example it follows that

$$\varphi_1 = \varphi_2 = \varphi_3$$

and

$$\Phi(x) = \begin{bmatrix} \varphi_1(x_1) \\ \varphi_1(x_2) \\ \varphi_1(x_3) \end{bmatrix}$$

This example motivates a general theorem characterising diagonal maps right network-preserving maps. See Section 8. When $\mathcal{G}^{\diamond \mathrm{R}}$ is trivial (as is common) this characterises the right network-preserving diffeomorphisms.

Four-Node Example. The network of Figure 6 has 4 nodes, is all-to-all connected (so the right skeleton is also all-to-all connected but has distinct arrow types). It is homogeneous, with two arrow-types, and $B^*(c,c) = \mathbb{Z}_2 \times \mathbf{1}$. This complicates the calculations but introduces some features of the effect of vertex symmetries.

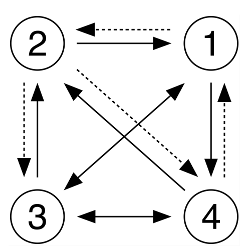

FIGURE 6. 4-node network.

The $\mathcal{G}$-admissible maps F and right skeleton maps Φ have the form

$$F(x) = \begin{bmatrix} f(x_1, \overline{x_2, x_3}, x_4) \\ f(x_2, \overline{x_3, x_4}, x_1) \\ f(x_3, \overline{x_1, x_4}, x_2) \\ f(x_4, \overline{x_1, x_3}, x_2) \end{bmatrix} \qquad \Phi(x) = \begin{bmatrix} \varphi_1(x_1, x_2, x_3, x_4) \\ \varphi_2(x_1, x_2, x_3, x_4) \\ \varphi_3(x_1, x_2, x_3, x_4) \\ \varphi_4(x_1, x_2, x_3, x_4) \end{bmatrix}$$

By Corollary 4.2, every right network-preserving map is both $\mathcal{G}$-admissible and $\mathcal{G}^{\diamond\mathrm{R}}$-admissible, so we can write

$$\Phi(x) = \begin{bmatrix} \varphi(x_1, \overline{x_2, x_3}, x_4) \\ \varphi(x_2, \overline{x_3, x_4}, x_1) \\ \varphi(x_3, \overline{x_1, x_4}, x_2) \\ \varphi(x_4, \overline{x_1, x_3}, x_2) \end{bmatrix}$$

for a single function φ.

In the standard order used in (5.2), the extended input tuples are:

$$\begin{array}{rcl} J(1) & = & (1,2,3,4) \\ J(2) & = & (2,3,4,1) \\ J(3) & = & (3,1,4,2) \\ J(4) & = & (4,1,3,2) \end{array}$$

There is one $\sim_I$ class, namely $\{1,2,3,4\}$ because the network is homogeneous.

The groups $B^*(c,c)$ are all $\mathbb{Z}_2$, interchanging the second and third nodes in the tuple. (They are all conjugate under the groupoid.)

We take $\mathcal{R} = \{1\}$. Now $B^*(1,1)$ is generated by

$$\alpha = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 2 & 4 \end{pmatrix}$$

acting via α^* on $P_J(1)$.

The identification maps are

$$\begin{array}{ll} \beta_2 = \beta : J(1) \to J(2) & \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix} \\ \beta_3 = \gamma : J(1) \to J(3) & \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix} \\ \beta_4 = \delta : J(1) \to J(4) & \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 3 & 2 \end{pmatrix} \end{array}$$

First we deal with α^*. The condition is:

$$\alpha^* \hat{\Phi}_{U(1)}(x_{U(1)}) \asymp \hat{\Phi}_{U(1)} \alpha^*(x_{U(1)})$$

We have

$$x_{J(1)} = (x_1, x_2, x_3, x_4) \qquad \alpha^* x_{J(1)} = (x_1, x_3, x_2, x_4)$$

Therefore

$$\begin{array}{rcl} \hat{\Phi}_{J(1)}(x_{J(1)}) & = & \begin{bmatrix} \varphi(x_1, x_2, x_3, x_4) \\ \varphi(x_1, x_2, x_3, x_4) \\ \varphi(x_1, x_2, x_3, x_4) \\ \varphi(x_1, x_2, x_3, x_4) \end{bmatrix} \\ \alpha^* \hat{\Phi}_{J(1)}(x_{J(1)}) & = & \begin{bmatrix} \varphi(x_1, x_2, x_3, x_4) \\ \varphi(x_1, x_2, x_3, x_4) \\ \varphi(x_1, x_2, x_3, x_4) \\ \varphi(x_1, x_2, x_3, x_4) \end{bmatrix} \\ \hat{\Phi}_{J(1)}(\alpha^* x_{J(1)}) & = & \begin{bmatrix} \varphi(x_1, x_3, x_2, x_4) \\ \varphi(x_1, x_3, x_2, x_4) \\ \varphi(x_1, x_3, x_2, x_4) \\ \varphi(x_1, x_3, x_2, x_4) \end{bmatrix} \end{array}$$

Therefore either

$$\varphi(x_1, x_2, x_3, x_4) = \varphi(x_1, x_3, x_2, x_4) \tag{5.2}$$

or, because of $\asymp$:

$$\varphi(x_1, x_2, x_3, x_4) = \varphi(x_1, x_3, x_2, x_4) \tag{5.3}$$

These equations just tell us that φ is symmetric in its second and third variables, which we already know.

Next we deal with β^*. The condition is:

$$\beta^* \hat{\Phi}_{U(2)}(x_{U(2)}) \asymp \hat{\Phi}_{U(1)} \beta^*(x_{U(1)})$$

In the same manner, this condition leads to

$$\varphi(x_2, x_3, x_4, x_1) \equiv \varphi(x_2, x_3, x_4, x_1)$$

which is trivially true.

Similar calculations can for γ^*, δ^* add no new information. We omit these.

The final result, putting all case and alternatives together, is that in this instance Φ is right network-preserving if and only if φ has the form (5.2). That is, Φ is both $\mathcal{G}$-admissible and $\mathcal{G}^{\diamond \mathrm{R}}$-admissible. So for this network, the necessary condition of Corollary 7.2 is also sufficient.

Remark 5.1. In the terminology of [20,25], this network of Figure 6 is a *semigroup network*. In such networks, composition of admissible maps always yields an admissible map. The context for semigroup networks is the work of Rink, Sanders, and Nijholt [20,21,24–26], who have developed an elegant approach to dimension reduction methods for network bifurcations, notably Liapunov–Schmidt, centre manifold, and Poincaré–Birkhoff normal form reduction. Their approach is based on graph fibrations (Boldi and Vigna [2], Deville and Lerman [5]) which are equivalent to forming quotient networks [15,28]. They focus on semigroup equivariance, in which there is a semigroup whose action preserves solutions of admissible ODEs.

The central idea is that instead of trying to preserve admissible maps, we can use more general coordinate changes while requiring them to preserve specific *features* of admissible maps. In some types of network, admissible maps are semigroup-equivariant, and the composition of semigroup-equivariant maps is always semigroup-equivariant. Nijholt et al. [23] have formulated a sweeping generalization using quivers. A quiver is a directed graph with multiple arrows and self-loops permitted, Derksen and Wayman [4]. A representation of a quiver associates a vector space to each node and a linear map between the corresponding vector spaces to each arrow. Quivers have led to major advances in representation theory. The key idea for network dynamics is 'quiver equivariance'. Admissible maps are quiver equivariant (though the converse is generally false), and quiver equivariant maps compose to give quiver equivariant maps. Thus, although composition does not preserve admissibility, it can preserve some useful features of admissible maps, which constrain the form of reduced bifurcation equations. $\diamond$

Six-Node Example. As a final example, we consider the 6-node network of Figure 5 (left). This example illustrates the effect of vertex symmetries. This has a nontrivial right skeleton. Its right network-preserving maps are nontrivial but have equalities of components not required by either $\mathcal{G}$- or $\mathcal{G}^{\diamond \mathrm{R}}$-admissibility: see (5.7).

The $\mathcal{G}$-admissible maps F and $\mathcal{G}^{\diamond\mathrm{R}}$-admissible maps Φ have the forms:

$$F(x) = \begin{bmatrix} f_1(x_1) \\ f_2(x_2) \\ f_3(x_3) \\ f_4(x_4) \\ f_5(x_5, \overline{x_1, x_2}, x_6) \\ f_5(x_6, \overline{x_3, x_4}, x_1) \end{bmatrix} \qquad \Phi(x) = \begin{bmatrix} \varphi_1(x_1) \\ \varphi_2(x_2) \\ \varphi_3(x_3) \\ \varphi_4(x_4) \\ \varphi_5(x_5, x_1, x_2, x_6) \\ \varphi_6(x_6, x_1) \end{bmatrix} \tag{5.4}$$

Any right network-preserving map must be both $\mathcal{G}$- and $\mathcal{G}^{\diamond\mathrm{R}}$-admissible, so

$$\Phi(x) = \begin{bmatrix} \varphi_1(x_1) \\ \varphi_2(x_2) \\ \varphi_3(x_3) \\ \varphi_4(x_4) \\ \varphi_5(x_5, \overline{x_1, x_2}, x_6) \\ \varphi_5(x_6, x_1) \end{bmatrix} \tag{5.5}$$

In the standard order used in (5.4), the extended input tuples are:

$$\begin{array}{l} J(1) = (1) \quad J(2) = (2) \quad J(3) = (3) \quad J(4) = (4) \\ J(5) = (5, 1, 2, 6) \quad J(6) = (6, 3, 4, 1) \end{array}$$

The ordering in $J(5), J(6)$ respects the action of $\beta \in B^*(5,6)$ and is standard.

Both $B^*(5,5)$ and $B^*(6,6) \cong \mathbb{Z}_2$, acting on the second and third nodes in the tuple. All other $B^*(c,c) = \mathbf{1}$.

We take $\sim_I$ representatives $\mathcal{R} = \{1,2,3,4,5\}$. The only nontrivial vertex group for these is $B^*(5,5)$ generated by the transposition

$$\alpha = \begin{pmatrix} 5 & 1 & 2 & 6 \\ 5 & 2 & 1 & 6 \end{pmatrix}$$

We also need $\beta = \beta_6 : J(5) \to J(6)$, for which

$$\beta = \begin{pmatrix} 5 & 1 & 2 & 6 \\ 6 & 3 & 4 & 1 \end{pmatrix}$$

First, consider $\alpha^* : P_{U(5)} \to P_{U(5)}$. We require

$$\alpha^* \hat{\Phi}_{U(5)} x_{U(5)} \asymp \hat{\Phi}_{U(5)} \alpha^* x_{U(5)}$$

That is,

$$\begin{bmatrix} \varphi_5(x_5, \overline{x_1, x_2}, x_6) \\ \varphi_2(x_5) \\ \varphi_1(x_5) \\ \varphi_6(x_5, x_6) \end{bmatrix} \asymp \begin{bmatrix} \varphi_5(x_5, \overline{x_2, x_1}, x_6) \\ \varphi_1(x_5) \\ \varphi_2(x_5) \\ \varphi_6(x_5, x_6) \end{bmatrix}$$

The first and last rows are redundant by $B^*(5,5)$-invariance, so we require

$$\varphi_1 = \varphi_2 \tag{5.6}$$

We must also consider the effect of β. Using (4.2):

$$\hat{\Phi}_6(x_{J(6)}) \asymp \hat{\Phi}_5(x_{J(6)})$$

That is:

$$\begin{bmatrix} \varphi_5(x_6, x_3, x_4, x_1) \\ \varphi_3(x_6) \\ \varphi_4(x_6) \\ \varphi_1(x_6) \end{bmatrix} \asymp \begin{bmatrix} \varphi_5(x_6, x_3, x_4, x_1) \\ \varphi_1(x_6) \\ \varphi_2(x_6) \\ \varphi_6(x_6) \end{bmatrix}$$

The first row is an identity. Since $B^*(5,5)$ swaps the second and third rows, the other rows tell us that *either*

$$\varphi_1 = \varphi_3 = \varphi_6 \qquad \varphi_2 = \varphi_4$$

or, considering $B^*(5,5)$-orbits (the relation $\asymp$):

$$\varphi_1 = \varphi_4 = \varphi_6 \qquad \varphi_2 = \varphi_3$$

But $\varphi_1 = \varphi_2$ by (5.6), so in either case all φ_i except φ_5 are equal, say

$$\varphi_1 = \varphi_2 = \varphi_3 = \varphi_4 = \varphi_6 = \varphi \qquad \varphi_5 = \psi$$

and

$$\Phi(x) = \begin{bmatrix} \varphi(x_1) \\ \varphi(x_2) \\ \varphi(x_3) \\ \varphi(x_4) \\ \psi(x_5, \overline{x_1, x_2}, x_6) \\ \varphi(x_6) \end{bmatrix} \tag{5.7}$$

6. Standard Order and Admissibility

Before moving on to the the proofs of the results stated in Section 4, we must establish some preparatory results that simplify computations and make the problem more tractable.

Setting $d = c$ in the pullback condition

$$f_d(x_d, x_{T(d)}) \equiv f_c(x_d, \beta_d^* x_{T(d)}) \tag{6.1}$$

implies that $f_c(x_c, x_{T(c)})$ is $B(c,c)$-invariant, where the *vertex group* $B(c,c)$ acts trivially on the node coordinate x_c and permutes the coordinates of $x_{T(c)}$ according to the pullback maps. The action of β is:

$$(x_c, x_{T(c)}) \mapsto (x_c, \beta^* x_{T(c)}) \tag{6.2}$$

It is convenient to consider β as an arrow-type preserving bijection on the *extended* input set: $\beta : J(c) \to J(d)$. The corresponding pullback map then takes the form

$$\beta^* : P_{U(d)} \to P_{U(c)} \tag{6.3}$$

where, $P_{U(d)} = P_{\mathcal{T}(J(d))}$, is defined by

$$(\beta^* x)_i = x_{\beta(i)} \qquad i \in J(c), x \in P_{U(d)}$$

In particular, if $c = d$ then $\beta \in B(c,c)$ and $\beta^* : P_{U(c)} \to P_{U(c)}$. Also $\beta(c) = d$.

The pullback condition (6.1) becomes:

$$f_d(x_{U(d)}) \equiv f_{d'}(\beta_d^* x_{U(d)}) \tag{6.4}$$

It is also useful to make a distinction:

Definition 6.1. If $c \in \mathcal{C}$ then the group

$$B^*(c,c) = \{\beta^* : \beta \in B(c,c)\}$$

◇

This is anti-isomorphic to $B(c,c)$ because $(\alpha\beta)^* = \beta^*\alpha^*$. Using inverses would make it an isomorphism, but we prefer not to do this.

We now state a well known characterisation of admissible maps, based on a specific ordering of coordinates in the domains of component maps f_c, which is more convenient for the purposes of this paper. See [28, Proposition 4.6].

Choose a total order on arrow-types, so that arrows of a given type occur as a consecutive block; then order arrows arbitrarily within each block. The internal arrow-types specified by $\sim_C$ are placed before all others in this order. (Each internal arrow-type occurs once for the corresponding component f_c.) Call this a *standard order*. Unless, for all $c \in \mathcal{C}$, the input set $I(c)$ has at most one arrow of any given type, standard order is not unique.

It is well known and easy to prove that the groupoid $\mathcal{B}$ is generated by a single *transition map* $\beta_{cd} : I(c) \to I(d)$ for each $c \neq d$ with $c \sim_I d$, together with a suitable subset of the vertex symmetry groups $B(c,c)$. It is enough to let c run through any set of representatives $\mathcal{R}$ of the $\sim_I$-equivalence classes. (In groupoid terminology, these classes determine the connected components of $\mathcal{B}$; see for example Higgins [17, Corollaries 1 and 2, page 47].) In standard order, if $c \sim_I d$ then we can identify $P_c \times P_{T(c)}$ with $P_d \times P_{T(d)}$, so that some transition map $\beta_{cd} \in B(c,d)$ satisfies $\beta^*_{cd} = \mathrm{id}$. This is why the usual way to represent symmetries of components using overlines on the relevant blocks of input variables (see [15, 28]) is possible. These overlines collect together all tail variables for a given arrow-type in a consecutive block. This gives the identity transition map $P_{T(d)} \to P_{T(c)}$ when these spaces are identified to preserve standard order.

The group $B(c,c)$ acts on the input set set $I(c)$ by permuting arrows and preserving arrow-type, so it preserves blocks of arrows in standard order. It acts trivially on the distinguished first coordinate x_c, so the action is as in (6.2). We can now characterise admissible maps in terms of $B(c,c)$-invariance, avoiding explicit reference to pullback maps. We now omit the hats in (2.5), so $\hat{f}_c$ is replaced by f_c.

Proposition 6.2. *A map $F = (f_c) : P \to P$ is admissible if and only if, in standard order, and with the appropriate identifications:*

(a) *The map f_c depends only on the coordinates x_c and $x_{T(c)}$, so we can assume that*

$$f_c : P_c \times P_{T(c)} \to P_c$$

(b) *For each $c \in \mathcal{R}$, the map f_c is invariant under the action (6.2) of $B(c,c)$.*

(c) *If $c \sim_I d$ then $f_c = f_d$.*

Proof. This follows from [28, Lemma 4.5 and Proposition 4.6], with the extra observation that when the inputs are in standard order, each transition map β_{cd} can be taken to be the appropriate identity map. □

The groupoid $\mathcal{B}_{\mathcal{G}}$ is generated by the $B(c,c)$ for $c \in \mathcal{R}$ and the β_d. Thus to prove admissibility we can consider only condition (b), namely $B(c,c)$-invariance, for each $c \in \mathcal{R}$, together with condition (c) for $c \neq d$ where both $c, d \in \mathcal{R}$.

Crucially, a similar remark holds for right network-preserving maps, as proved in Lemma 7.9. Therefore to check $F\Phi$ for (1) we can consider just those $c \in \mathcal{R}$. (Note that (1) is automatic if $B(c,c) = \mathbf{1}$ for all c — the vertex-trivial case.) To check (2) we must also consider the β_d. So we only need to check (4.1) for this set of generators. Each generator imposes conditions on Φ, and we have to work out the consequences of all of these conditions.

Finally, we can construct a coordinate system on $P_{U(c)}$ using a set of representatives $\mathcal{R} = \{c_1, \dots, c_r\}$ to $\sim_I$ and defining β_d as above. For each $c \in \mathcal{R}$ choose a coordinate system on $P_{U(c)}$ defined by the input nodes $J(c)$, and assume standard order. Now we can and do take β_d^* to be the identity. When $d \neq c$ (and only then) equation (4.1) therefore reduces to (4.2) and (4.3).

7. Proofs of the Main Results

We now prove the results in Section 4.

Throughout let $F = (f_1, \ldots, f_n)$ be any admissible map for a network $\mathcal{G}$ with nodes $\mathcal{C} = \{1, \ldots, n\}$ and let Φ be a right network-preserving map. Let $P = \oplus P_c$ be the state space. For simplicity assume $P_c = \mathbb{R}$ for all c.

For any $G : P \to P$ with $G(x) = [g_1(x), \ldots, g_n(x)]$, and any tuple $\mathbf{u} = (u_1, \ldots, u_k)$ where all $u_j \in \mathcal{C}$, we define

$$\begin{aligned} x_{\mathbf{u}} &= (x_{u_1}, \ldots, x_{u_k}) \\ G_{\mathbf{u}} &= (g_{u_1}, \ldots, g_{u_k}) \end{aligned}$$

Symmetry Lemma. We begin with a technical lemma that helps to deal with vertex symmetries.

Lemma 7.1. *Let $y_j = \varphi_j(x_{i_j})$ for $1 \leq j \leq k$ where the functions φ_j are not constant. Suppose that for every $\mathbb{S}_k$-invariant function $f(y_1, \ldots, y_k)$ the function*

$$g(x_{i_1}, \ldots, x_{i_k}) = f(\varphi_{i_1}(x_{i_1}), \ldots, \varphi_{i_k}(x_{i_k}))$$

is invariant under the action of $\mathbb{S}_k$ on the indices $(i_1, \ldots, i_k)$ of the x_{i_j}, but leaving the indices of the φ_{i_j} fixed. Then

$$\varphi_{i_1} = \varphi_{i_2} = \cdots = \varphi_{i_k}$$

Proof. If all indices i_j are equal there is nothing to prove. Otherwise let $1 \leq l < m \leq k$ be such that $i_l \neq i_k$, so that the corresponding variables x_{i_l} and x_{i_m} are independent.

Let $\tau = (lm) \in \mathbb{S}_k$ be the transposition that swaps l and m. Then for all $\mathbb{S}_k$-invariant f, we have

$$\begin{aligned} &f(\varphi_{i_1}(x_{i_1}), \ldots; \varphi_{i_l}(x_{i_l}); \ldots; \varphi_{i_m}(x_{i_m}); \ldots \varphi_{i_k}(x_{i_k})) \equiv \\ &\quad f(\varphi_{i_1}(x_{i_1}), \ldots; \varphi_{i_l}(x_{i_m}); \ldots; \varphi_{i_m}(x_{i_l}); \ldots \varphi_{i_k}(x_{i_k})) \end{aligned}$$

Let

$$f(y) = y_1 + \cdots + y_k$$

Cancelling common terms, we obtain:

$$\varphi_{i_l}(x_{i_l}) + \varphi_{i_m}(x_{i_m}) \equiv \varphi_{i_l}(x_{i_m}) + \varphi_{i_m}(x_{i_l})$$

To simplify notation set

$$x_{i_l} = u \qquad x_{i_m} = v \qquad \varphi_{i_l} = \varphi \qquad \varphi_{i_m} = \psi$$

so that

$$\varphi(u) + \psi(v) \equiv \varphi(v) + \psi(u)$$

Therefore

$$\varphi(u) - \psi(u) \equiv \varphi(v) - \psi(v)$$

Since u, v are independent variables, both of these expressions must be a constant $c \in \mathbb{R}$. That is,

$$\psi(u) \equiv \varphi(u) + c \qquad \psi(v) \equiv \varphi(v) + c \tag{7.1}$$

We claim that $c = 0$. To prove this, set

$$f(y) = y_1 y_2 + y_2 y_3 + \cdots + y_{k-1} y_k$$

and again consider the transposition τ. Substituting $y_j = \varphi_{i_j}(x_{i_j})$ in f and cancelling common terms we find that

$$c\varphi(u) \equiv c\varphi(v)$$

But u, v are independent variables, and φ is not constant, so $c = 0$. By (7.1), $\varphi = \psi$.

Since this equality holds for all pairs l, m such that $i_l \neq i_m$, the result follows. □

Domain Conditions for Right Network-Preserving Maps. Now we prove the main result about domain conditions.

Proof of Theorem 4.1.

First we prove that (b) implies (a). This follows from the analogous result for fully inhomogeneous networks, because every $\mathcal{G}$-admissible map is $\mathcal{G}^\diamond$-admissible. 'Satisfies the domain conditions for $\mathcal{G}$-admissibility' is the same as '$\mathcal{G}^\diamond$-admissible'. And $\mathcal{G}^\diamond$ is fully inhomogeneous.

We can also give a simple self-contained proof, as follows. For all nodes $c \in \mathcal{C}$ define

$$K(c) = \{d \in \mathcal{C} : \frac{\partial \varphi_c}{\partial x_d} \not\equiv 0\}$$

That is, 'those d for which φ_c depends on x_d.

If $\mathcal{W} \subseteq \mathcal{C}$, write

$$K(\mathcal{W}) = \cup_{c \in \mathcal{W}} K(c)$$

In the skeleton, $U(c)$ can be identified with the usual extended input set $J(c)$, which can be considered as a set of *nodes*.

The statement that Φ is $\mathcal{G}^{\diamond \mathrm{R}}$-admissible is equivalent to:

$$K(T(c)) \subseteq T(c) \quad \forall c \in \mathcal{C} \tag{7.2}$$

by the definition of the right core. Assume (7.2). Since F is $\mathcal{G}$-admissible,

$$(F\Phi)_c(x) = f_c(\Phi(x)) = \hat{f}_c(\Phi(x_{T(c)}))$$

for some $\hat{f}$. Let $T(c) = \{i_1, \ldots, i_m\}$. Then

$$(F\Phi)_c(x) = f_c(\Phi(x)) = \hat{f}_c(\varphi_c(x), \varphi_{i_1}(x), \ldots, \varphi_{i_m}(x)))$$

Now $\varphi_c(x) = \hat{\varphi}_c(x_{U(c)})$ by admissibility, and

$$\varphi_{i_k}(x) = \hat{\varphi}_{i_k}(x_{K(i_k)})$$

So $(F\Phi)_c(x)$ is a function of x_l where $l \in K(T(c)) \subseteq T(c)$. That is, $F\Phi$ satisfies the domain condition for $\mathcal{G}$-admissibility for component c. Since this holds for all $c \in \mathcal{C}$, we have proved (a).

Next we prove that (a) implies (b). Initially we assume node spaces are 1-dimensional, for simplicity. Then we explain how to modify the proof for higher-dimensional nodes.

By Proposition 6.2 we can define a $\mathcal{G}$-admissible map uniquely by choosing a representative c for each input-equivalence class, and defining f_c to have the correct domain and to be $B(c, c)$-invariant. All other f_d are then defined by pullback. We use this on one specific node c.

Suppose that (a) is true but (b) is false. Then there exist nodes c, d with

$$d \in K(T(c)) \qquad d \notin T(c)$$

Since $d \in K(T(c))$ there is some node i such that $i \in T(c)$ and $d \in K(i)$.

Since $i \in T(c)$ there exists some arrow e_1 with $\mathcal{H}(e_1) = c, \mathcal{T}(e_1) = i$. (That is, e_1 links $i \to c$.) Let $e_2, \ldots, e_p$ be all the other input arrows to c with the same arrow-type as e_1. Formally,

$$\mathcal{H}(e_q) = c \qquad e_q \sim_e e_1 \qquad 2 \leq q \leq p$$

By $\mathcal{G}$-admissibility, f_c is $\mathbb{S}_p$-invariant where the action of $\mathbb{S}_p$ is induced by permuting the input arrows $e_1, e_2, \ldots, e_p$ (and then taking tail nodes).

Let $\mathcal{T}(e_j) = i_j \in \mathcal{C}$. Observe that $i_1 = i$.

Choose f_c to depend only on the tail nodes of these arrows, via these arrows. That is, $\hat{f}_c(x) = g(x_{U(e_1)}, \ldots, x_{U(e_p)})$ where $g = g(y_1, \ldots, y_p)$ is $\mathbb{S}_p$-invariant. The other direct factors of the vertex group act trivially.

This g is one component of a special type of $\mathcal{G}$-admissible map. The other components are defined by pullback using Proposition 6.2 for all nodes input-equivalent to c. For the other nodes, we can use any arbitrary $\mathcal{G}$-admissible map, for example zero. Let

$$\theta_r = \mathcal{T}(e_r) \qquad 1 \leq r \leq p$$

The appropriate direct factor $\mathbb{S}_p$ of $B(c,c)$ acts on the e_r by permuting them arbitrarily. If we can find some $\mathbb{S}_p$-invariant g such that $\frac{\partial g}{\partial x_d} \neq 0$ we get a contradiction. So we may assume (for a contradiction) that $\frac{\partial g}{\partial x_d} \equiv 0$ for all $\mathbb{S}_p$-invariant g.

Let t be a new indeterminate and consider the polynomial

$$\mu(t) \quad = \quad (t - \varphi_{\theta_1}(x)) \ldots (t - \varphi_{\theta_p}(x)) \in \mathcal{C}^\infty(\mathbb{R})[t]$$

where $\mathcal{C}^\infty(\mathbb{R})$ is the ring of $\mathbb{R}$-valued smooth functions on $\mathbb{R}^n$. Clearly $\mu(t)$ is $\mathbb{S}_p$-invariant (for any $t \in \mathbb{R}$, or formally as a polynomial).

By assumption, the x_d-partial derivative of *any* $\mathbb{S}_p$-invariant function g of the φ_{θ_r} is identically zero. Since t is an indeterminate, $\frac{\partial t}{\partial x_d} = 0$. Therefore $\frac{\partial \mu(t)}{\partial x_d} \equiv 0$. Removing a minus sign,

$$\sum_{r=1}^{p} \prod_{s \neq r} (t - \varphi_{\theta_s}(x)) \equiv 0$$

Substitute $t = \varphi_{\theta_1}(x)$ to deduce that

$$\frac{\partial \varphi_{\theta_1}}{\partial x_d} (\varphi_{\theta_1}(x) - \varphi_{\theta_2}(x)) \cdots (\varphi_{\theta_1}(x) - \varphi_{\theta_p}(x)) \equiv 0$$

When $x \in U$ we have $\frac{\partial \varphi_{\theta_1}}{\partial x_d} \neq 0$, so

$$(\varphi_{\theta_1}(x) - \varphi_{\theta_2}(x)) \cdots (\varphi_{\theta_1}(x) - \varphi_{\theta_p}(x)) = 0 \qquad \forall x \in U$$

Therefore $\Phi(U)$ is contained in a nontrivial union of proper hyperplanes, so not an open set. Contradiction.

This prove the result when node spaces are 1-dimensional. For higher-dimensional node spaces we replace the φ_{θ_i} by their projections onto 1-dimensional subspaces of the P_i and use the same argument. □

A simple corollary to Theorem 4.1 is:

Corollary 7.2. *The following conditions are necessary for Φ to be right network-preserving for $\mathcal{G}$:*

(a) *Φ is $\mathcal{G}$-admissible.*
(b) *Φ is $\mathcal{G}^{\diamond \mathrm{R}}$-admissible.*

There are many examples for which (a) and (b) determine the right network-preserving maps. However, the first example in Section 5 shows that in general (a) and (b) are *not* sufficient for Φ to be right network-preserving.

Characterisation of Right Network-Preserving Maps. In this section we write down a formal statement of the conditions that must be satisfied by Φ in order for it to be right network-preserving.

It remains to understand what the pullback conditions imply about Φ. We now investigate these.

Notation. We introduce the notation $j \sqsubset \mathbf{j}$ to mean that 'j is a component of $\mathbf{j} = (j_1, \ldots, j_k)$. Also, we write

$$\Phi_{U(c)} = (\varphi_c, \varphi_{i_1}, \ldots, \varphi_{i_k})$$

when $T(c) = (i_1, \ldots, i_k)$. We also write points in P and maps $P \to P$ as column vectors when this is more convenient typographically.

Theorem 4.1 leads to a useful lemma:

Lemma 7.3. *For every node $c \in \mathcal{C}$ we have*

$$\Phi_{U(c)}(x) = \hat{\Phi}_{U(c)}(x_{U(c)})$$

where

$$\hat{\Phi}_{U(c)} : P_{U(c)} \to P_{U(c)} \tag{7.3}$$

Proof. First we show that if $j \sqsubset J(c)$ then $\hat{\varphi}_j$ depends only on the components $x_{U(c)}$. More precisely, it is independent of $x_{L(c)}$ where $L(c) = \mathcal{C} \setminus J(c)$. Therefore $\hat{\varphi}_j(x_{U(c)})$ is well defined, and $\hat{\varphi}_j(x_{U(c)}) \in P_j \subseteq P_{U(c)}$.

By Theorem 4.1:

$$\varphi_j(x) = \hat{\varphi}_j(x_{K(j)})$$

where $K(j)$ is the input set of j in $\mathcal{G}^{\diamond \mathrm{R}}$. But $K(j) \subseteq J(c)$ by the transitivity property defining the right core. Therefore $\hat{\varphi}_j(x)$ is independent of variables x_p with $p \in \mathcal{C} \setminus K(j) \supseteq \mathcal{C} \setminus J(c) = L(c)$. So $\hat{\varphi}_j(x_{U(c)})$ is well defined as claimed.

We can therefore naturally interpret $\hat{\varphi}_j$ as a map

$$\hat{\varphi}_j : P_{U(c)} \to P_j$$

(we avoid introducing new notation such as $\tilde{\varphi}_j$). But this is true for every $j \sqsubset J(c)$, which run through the indices of the components of $\Phi_{U(c)}$. So

$$\hat{\Phi}_{U(c)} : P_{U(c)} \to \bigoplus_{j \sqsubset J(c)} P_j = P_{U(c)} \qquad \square$$

Equation (7.3) shows that the group $B^*(c, c)$ acts on $U(c)$. When $d \sim_I c$, we have see the transition map β_{cd} to identify $U(d)$ with $U(c)$. Now $\beta_{cd} = \mathrm{id}$, so $B^*(c, c)$ acts on $U(d)$ and is identified with $B^*(d, d)$.

Lemma 7.3 is the key to the main result of this section, the basic characterisation of right network-preserving maps Theorem 4.4. The proof will be given after some remarks and a lemma.

Remark 7.4. The groups $B(c, c)$ are all finite, so each condition (4.1) is equivalent to a finite set of conditions

$$\beta^* \hat{\Phi}_{U(d)}(x_{U(d)}) = \gamma^* \hat{\Phi}_{U(c)} \beta^*(x_{U(d)}) \qquad \forall x_{U(d)} \in P_{U(d)} \text{ and some } \gamma \in B(c, c) \tag{7.4}$$

Remark 7.5. Equation (7.4) is a kind of groupoid-equivariance condition. Although we use nothing deep about groupoids, they play a useful role in organising the calculations.

Compositions of maps make sense because of (6.3) and Lemma 7.3. The proof implicitly shows that if Φ is network-preserving then those compositions have to make sense, which is what drives the proof of Theorem 4.1.

The equivalent conditions (4.1) and (4.2) do not involve F and express $\hat{\Phi}_{U(d)}$ in terms of $\hat{\Phi}_{U(c)}$ and $\mathcal{B}_{\mathcal{G}}$. They therefore characterise right network-preserving maps in terms of network architecture.

However, when $B(c, c) \neq \mathbf{1}$, equality of orbits introduces complications in examples since there are alternative possibilities: see Section 5. To obtain a more explicit characterisation of the components $\hat{\varphi}_c$ we have to sort out the effect of all the conditions on equation (4.2). The same component can be affected by several different conditions here, the $B^*(c,c)$ symmetries come into play in interpreting the effect of these conditions, and Φ being $\mathcal{G}^{\diamond\mathrm{R}}$-admissible imposes important and useful constraints. The key point is that the characterisation pulls apart into conditions on the subspaces $P_{U(c)}$. $\diamond$

Remark 7.6. From now on we omit the hats on the φ. We have established conditions on domains that makes this possible without ambiguity. In general $\varphi_c(x_{\mathbf{u}})$ makes sense as long as $\mathbf{u}$ includes the domain of $\hat{\varphi}_c$: just project $x_{\mathbf{u}}$ onto that domain. $\diamond$

Before giving the proof of the Theorem 4.4, we state a lemma that is useful whenever the vertex group $B(c,c) \neq \mathbf{1}$. It is well-known ('invariants separate orbits') but we give a short proof for completeness.

Lemma 7.7. *Suppose that a finite group Γ acts on $\mathbb{R}^n$. Then the following are equivalent:*
(a) *$x, y \in \mathbb{R}^n$ lie in the same Γ-orbit.*
(b) *$\psi(x) = \psi(y)$ for all Γ-invariant functions $\psi : \mathbb{R}^n \to \mathbb{R}$.*

Proof. (a) $\Rightarrow$ (b) is trivial.
For the converse, assume (b). Consider x as fixed and define

$$\psi(y) = \prod_{\gamma \neq \delta \in \Gamma} \|\gamma y - \delta x\|^2$$

This is Γ-invariant. By (b)

$$\psi(y) = \psi(x) = 0$$

Therefore $\gamma y = \delta x$ for some γ, δ. Thus $y = \gamma^{-1}\delta x$, in the same orbit as x. $\square$

Corollary 7.8. *Let $y, z \in P_{U(c)}$. The equation*

$$\hat{f}_c(y) = \hat{f}_c(z) \quad \forall\ \mathcal{G}\text{-admissible } \hat{f}_c$$

holds if and only if y and z are in the same $B^*(c,c)$-orbit.

Proof of Theorem 4.4. We chase diagrams. By (6.1), F is $\mathcal{G}$-admissible if and only if, for all $c, d, \beta \in B(c,d)$, the following diagram commutes:

$$\begin{array}{ccc} P_{U(c)} & \xrightarrow{f_c} & P_c \\ \uparrow{\scriptstyle \beta^*} & & \| \\ P_{U(d)} & \xrightarrow{f_d} & P_d \end{array} \tag{7.5}$$

Extend it to the following diagram to compose with the relevant components of Φ:

$$\begin{array}{ccccc} P_{U(c)} & \xrightarrow{\hat{\Phi}_{U(c)}} & P_{U(c)} & \xrightarrow{f_c} & P_c \\ \uparrow{\scriptstyle \beta^*} & & \uparrow{\scriptstyle \beta^*} & & \| \\ P_{U(d)} & \xrightarrow{\hat{\Phi}_{U(d)}} & P_{U(d)} & \xrightarrow{f_d} & P_d \end{array} \tag{7.6}$$

Now $(F\Phi)_c(x) = f_c(\Phi(x))$ so $(F\Phi)_c = f_c\Phi$. Therefore $F\Phi$ is $\mathcal{G}$-admissible if and only if the outer rectangle in the diagram (7.6) commutes for all f_c, f_d.

The right-hand square (7.5) commutes. Therefore the outer rectangle commutes if and only if the left-hand square commutes (once acted on by f_c, f_d). More precisely, the left-hand square is

$$\begin{tikzcd} P_{U(c)} \arrow[r, "\hat{\Phi}_{U(c)}"] & P_{U(c)} \\ P_{U(d)} \arrow[u, "\beta^*"] \arrow[r, "\hat{\Phi}_{U(d)}"] & P_{U(d)} \arrow[u, "\beta^*"] \end{tikzcd} \tag{7.7}$$

By Lemma 7.7, this must commute *modulo the action of* $B^*(c,c)$ at the top right-hand corner. That is, equality up to $B^*(c,c)$-orbits. This diagram is independent of f_c, f_d, so it commutes modulo $B^*(c,c)$ if and only if (4.1) is valid. □

For later use, we note:

Lemma 7.9. *If* (7.6) *holds for all* β *in a set of generators for* $\mathcal{B}_\mathcal{G}$*, then it holds for all* $\beta \in \mathcal{B}_\mathcal{G}$*.*

Proof. The main step is this. Let $\gamma \in B(d,e)$for a node e. Then

$$\begin{gathered} \gamma^* : P_{J(e)} \to P_{J(e)} \\ \beta^*\gamma^* = (\gamma\beta)^* \end{gathered} \tag{7.8}$$

The diagram

$$\begin{tikzcd} P_{U(c)} \arrow[r, "\hat{\Phi}_{U(c)}"] & P_{U(c)} \arrow[r, "f_c"] & P_c \\ P_{U(d)} \arrow[u, "\beta^*"] \arrow[r, "\hat{\Phi}_{U(d)}"] & P_{U(d)} \arrow[u, "\beta^*"] \arrow[r, "f_d"] & P_d \arrow[u, equal] \\ P_{J(e)} \arrow[u, "\gamma^*"] \arrow[r, "\hat{\Phi}_{J(e)}"] & P_{J(e)} \arrow[u, "\gamma^*"] \arrow[r, "f_e"] & P_e \arrow[u, equal] \end{tikzcd} \tag{7.9}$$

commutes. Composing vertical arrows and using (7.8), the diagram

$$\begin{tikzcd} P_{U(c)} \arrow[r, "\hat{\Phi}_{U(c)}"] & P_{U(c)} \arrow[r, "f_c"] & P_c \\ P_{J(e)} \arrow[u, "(\gamma\beta)^*"] \arrow[r, "\hat{\Phi}_{J(e)}"] & P_{J(e)} \arrow[u, "(\gamma\beta)^*"] \arrow[r, "f_e"] & P_e \arrow[u, equal] \end{tikzcd} \tag{7.10}$$

commutes. □

Now we show how to extract useful information from (4.1), so that we can calculate examples. Such calculations require choices of coordinates on the spaces $P_{U(c)}$. Everything becomes much simpler if this is done systematically using a suitable generating set for $\mathcal{B}_\mathcal{G}$ and appealing to Lemma 7.9. In standard order, we can and do take β_d^* to be the identity. Now equation (4.1) reduces to (4.2) and (4.3).

8. Diagonal Maps

In this section we prove Theorems 4.7 and 4.8, characterise the diagonal right network-preserving maps for any network $\mathcal{G}$ (without self-loops and multiple arrows). 'Diagonal' is defined in Definition 2.10.

Proof of Theorem 4.7. We must show that a diagonal map Φ is right network-preserving if and only if

$$i \sim_S j \implies \varphi_i = \varphi_j$$

We apply Theorem 4.4 in the version that uses (4.2) and (4.3). We restate them for convenience:

For all $\alpha \in B(c,c)$:

$$\alpha^*\hat{\Phi}_{U(c)}(x_{U(c)}) \asymp \hat{\Phi}_{U(c)}\alpha^*(x_{U(c)}) \qquad \forall x_{U(c)} \in P_{U(c)} \tag{8.1}$$

and when $d \neq c$, for all $\beta \in B(c,d)$:

$$\hat{\Phi}_{U(d)}(x_{U(d)}) \asymp \hat{\Phi}_{U(c)}(x_{U(d)}) \qquad \forall x_{U(d)} \in P_{U(d)} \tag{8.2}$$

Further, recall Lemma 7.9: it is necessary and sufficient to verify these conditions for generators of $\mathcal{B}_{\mathcal{G}}$.

When Φ is diagonal these conditions simplify. We can replace $x_{J(i)}$ by x_i, and $\hat{\Phi}_{J(j)}$ by φ_j. First consider (8.1). This becomes

$$\alpha^*\varphi_c(x_c) \asymp \varphi_c\alpha^*(x_c) \qquad \forall x_c \in P_c$$

However, $B(c,c)$ acts trivially on the distinguished node variable, so $\alpha^*(x_c) = x_c$. By definition,

$$\alpha^*\varphi_c(x_c) = \varphi_{\alpha(c)}(x_c)$$

So conditions (8.1) and (8.2) are equivalent to: For all $\alpha \in B(c,c)$ and $\beta = \beta_d$ (so $d \sim_I c$)

$$\varphi_{\alpha(c)}(x_c) = \varphi_c(x_c) \qquad \forall x_c \in P_c \tag{8.3}$$

$$\varphi_d(x_d) = \varphi_c(x_d) \qquad \forall x_d \in P_d \tag{8.4}$$

First, suppose that Φ is right network-preserving. Equations (8.4) and (8.4) are equivalent to the condition: if $d \dot{\sim}_S c$ then $\varphi_d = \varphi_c$. Because $\sim_S$ is the transitive extension of $\dot{\sim}_S$, it follows that if $d \sim_S c$ then $\varphi_d = \varphi_c$.

Conversely, suppose that if $d \sim_S c$ then $\varphi_d = \varphi_c$. In particular, if if $d \dot{\sim}_S c$ then $\varphi_d = \varphi_c$. But, as previously remarked, equation (8.4) is equivalent to the condition: if $d \dot{\sim}_S c$ then $\varphi_d = \varphi_c$. Therefore (8.4) is valid.

It remains to verify (8.3). Suppose that $i, j \sqsubset J(c)$ and $\alpha \in B(c,c)$. Then $i \dot{\sim}_S j$, so $\varphi(i) = \varphi(j)$. Now (8.3) is valid. □

Corollary 8.1. *Suppose that $\mathcal{G}^{\diamond\mathrm{R}}$ is trivial (that is, it has no arrows, only nodes). Then the right network-preserving maps are precisely the diagonal maps $\Phi = [\varphi_1, \ldots, \varphi_n]^{\mathrm{T}}$ such that $\varphi_i = \varphi_j$ whenever $i \sim_S j$.*

Proof of Theorem 4.8.

We must prove that a diagonal map

$$\Phi(x) = (\varphi_1(x_1), \ldots, \varphi_n(x_n))$$

is left network-preserving if and only if

$$c \sim_I d \implies \varphi_c = \varphi_d \tag{8.5}$$

Suppose Φ is left network-preserving. Then it is $\mathcal{G}$-admissible so (4.5) holds.

Conversely, suppose (4.5) holds. We must verify the pullback conditions

$$(\Phi F)_d(x_{J(d)}) = (\Phi F)_c(\beta^* x_{J(d)})$$

for all $\beta \in B(c,d)$. Since $B(c,d) \neq \emptyset$ we have $c \sim_I d$, so by (4.5) we have $\varphi_c = \varphi_d$. But now

$$\begin{aligned} (\Phi F)_d(x_{J(d)}) &= \varphi_d(f_d(x_{J(d)}) \\ &= \varphi_d(f_c(\beta^* x_{J(d)})) \\ &= \varphi_c(f_c(\beta^* x_{J(d)})) \\ &= (\Phi F)_c(\beta^* x_{J(d)}) \end{aligned}$$

as required. □

Remarks 8.2. (a) These are precisely the diagonal maps that are also $\mathcal{G}$-admissible.

(b) This differs from the right network-preserving case, where we have $\sim_S$ in place of $\sim_I$. In general $\sim_I$ refines $\sim_S$ ($\sim_S$ equivalence classes are unions of $\sim_I$ classes). ◇

9. Inverses for Diffeomorphisms

For the 3-node and 6-node examples in Section 5, it is easy to check that if Φ is right network-preserving and the inverse Φ^{-1} exists — that is, Φ is a diffeomorphism — then Φ^{-1} is also right network-preserving. However, it is not so obvious for the 4-node example, showing that the fact that $\mathcal{D}^R_{\mathcal{G}}(P)$ is a group is not trivial.

We end by proving this result in general. Theorem 4.4 implies that if $\Phi \in \mathcal{D}^R_{\mathcal{G}}(P)$ is a diffeomorphism, then its inverse $\Phi^{-1} \in \mathcal{D}^R_{\mathcal{G}}(P)$. That is, the semigroup $\mathcal{D}^R_{\mathcal{G}}(P)$ is actually a group. The proof is mostly straightforward diagram-chasing, but some care is needed.

Motivation: The basic idea is to invert the maps in (7.7), obtaining:

(9.1)
$$\begin{tikzcd} P_{U(c)} \arrow[d, "\beta^{*-1}"] & P_{U(c)} \arrow[l, "\hat{\Phi}^{-1}_{U(c)}"'] \arrow[d, "\beta^{*-1}"] \\ P_{U(d)} & P_{U(d)} \arrow[l, "\hat{\Phi}^{-1}_{U(d)}"'] \end{tikzcd}$$

and observe that this diagram must now commute modulo the group action at the bottom left corner. This is $B(d,d)$, not $B(c,c)$. (They are conjugate in $\mathcal{B}_{\mathcal{G}}$, and the conjugacy transfers the action from $P_{U(c)}$ to $P_{U(d)}$.) Swapping c and d we get the same condition for Φ^{-1} since β^{-1} runs through the groupoid when β does.

We also need Lemma 7.9.

Proof of Theorem 4.6. Suppose Φ is right network-preserving. Then Φ is $\mathcal{G}^{\mathrm{DR}}$-admissible. Therefore by [12], Φ^{-1} is $\mathcal{G}^{\mathrm{DR}}$-admissible. This deals with the domain condition.

For the pullback condition, we use Theorem 4.4. Equation (4.1) is valid. We deduce the corresponding equation(s) for Φ^{-1} in two steps: $c = d$ and $c \neq d$.

Step 1: Set $d = c$. Then (with all maps acting on $P_{J(c)}$ we have

$$\beta^* \hat{\Phi}_{U(c)} \asymp \hat{\Phi}_{U(c)} \beta^* \tag{9.2}$$

To simplify notation let $\Psi = \hat{\Phi}_{U(c)}$, so that (9.2) becomes $\beta^*\Psi \asymp \Psi\beta^*$, and let $B = B^*(c,c)$, which is (anti)-isomorphic to $B(c,c)$, hence is a finite group. Recall that $\asymp$ denotes 'same $B(c,c)^*$-orbit'. That is, there exists $\alpha^* \in B$ (*which may depend on* β^*) such that

$$\beta^*\Psi = \alpha^*\Psi\beta^* \tag{9.3}$$

Therefore $\Psi\beta^*\Psi^{-1} = \alpha^{*-1}\beta^* \in B$ so

$$\Psi\beta^*\Psi^{-1} \in B \tag{9.4}$$

Equation (9.4) is equivalent to (9.3). To show that a similar equation holds for $(\hat{\Phi}_{U(c)})^{-1}$, we must show that

$$\Psi^{-1}\beta^*\Psi \in B \tag{9.5}$$

Clearly $\Psi B\Psi^{-1} \subseteq B$. This is almost the definition of the normaliser of B, but that requires equality. We can obtain equality because B is finite. For fixed Ψ the map $B \to B$ defined by $\beta^* \mapsto \Psi\beta^*\Psi^{-1}$ is injective. Since B is finite, the map is surjective, so $\Psi B\Psi^{-1} = B$. Therefore $B = \Psi^{-1}B\Psi$, and this implies the required equation (9.5).

Step 2: Now let $d \neq c$.

By (4.1), for $\beta \in B(c,d)$, we have $\beta^*\hat{\Phi}_{U(d)} \asymp \hat{\Phi}_{U(c)}\beta^*$, so there exists $\alpha \in B(c,c)$ such that

$$\beta^*\hat{\Phi}_{U(d)} = \alpha^*\hat{\Phi}_{U(c)}\beta^* \tag{9.6}$$

Observe that $\beta^{-1} \in B(d,c)$. The conjugate $\beta^{*-1}\alpha^*\beta^*$ lies in $B^*(d,d)$, so by Step 1 of the proof applied at node d we get

$$\hat{\Phi}^{-1}_{U(d)}(\beta^{*-1}\alpha^*\beta^*) = \gamma^*\hat{\Phi}^{-1}_{U(d)} \tag{9.7}$$

for some $\gamma^* \in B(d,d)^*$. Therefore

$$\hat{\Phi}^{-1}_{U(d)}\beta^{*-1}\alpha^* = \gamma^*\hat{\Phi}^{-1}_{U(d)}\beta^{*-1} \tag{9.8}$$

Inverting (9.6) we obtain

$$\hat{\Phi}^{-1}_{U(d)}\beta^{*-1} = \beta^{*-1}\hat{\Phi}^{-1}_{U(c)}\alpha^{*-1} \tag{9.9}$$

Substitute from (9.8) and then (9.9) to get:

$$\gamma^*\hat{\Phi}^{-1}_{U(d)}\beta^{*-1} = \hat{\Phi}^{-1}_{U(d)}\beta^{*-1}\alpha^* = \beta^{*-1}\hat{\Phi}^{-1}_{U(c)}\alpha^{*-1}\alpha^* = \beta^{*-1}\hat{\Phi}^{-1}_{U(c)}$$

so

$$\hat{\Phi}^{-1}_{U(d)}\beta^{*-1} = \gamma^{*-1}\beta^{*-1}\hat{\Phi}^{-1}_{U(c)}$$

Therefore

$$\hat{\Phi}^{-1}_{U(d)}\beta^{*-1} \asymp \beta^{*-1}\hat{\Phi}^{-1}_{U(c)}$$

and the inverted diagram commutes up to $B^*(d,d)$-orbits.

There is a potential issue because α depends on β. However, by Lemma 7.9 a single element of $B^*(c,d)$ for each $c \neq d$ suffices, because $\mathcal{B}_{\mathcal{G}}$ is generated by the vertex groups $B(c,c)$ and a single element from each $B^*(c,d)$.

We have $\beta^{-1} \in B(d,c)$. Therefore, by Theorem 4.4, the inverse Φ^{-1} also satisfies the pullback conditions, so it is $\mathcal{G}$-admissible. □

Acknowledgements. We thank Marty Golubitsky for encouragement, inspiration, and helpful conversations. The work of FA is partially supported by FAPESP grant 2019/12247-7.

References

[1] F. Antoneli, M. Golubitsky and I. Stewart. Homeostasis in a feed forward loop gene regulatory network motif. *J. Theoretical Biology* **445** (2018) 103–109. DOI: 10.1016/j.jtbi.2018.02.026

[2] P. Boldi and S. Vigna. Fibrations of graphs. *Discrete Math.* **243** (2002) 21–66. DOI: 10.1016/s0012-365x(00)00455-6

[3] R. Brown. From groups to groupoids: a brief survey. *Bull. London Math. Soc.* **19** (1987) 113–134.

[4] H. Derksen and J. Weyman. Quiver representations. *Notices Amer. Math. Soc.* **52** (2005) 200–206.

[5] L. DeVille and E. Lerman. Modular dynamical systems on networks. *J. Eur. Math. Soc.* **17** (2013) 2977–3013. DOI: 10.4171/jems/577

[6] J.E. Ferrell. Perfect and near perfect adaptation in cell signaling. *Cell Systems* **2** (2016) 62–67. DOI: 10.1016/j.cels.2016.02.006

[7] C. Gibson. *Singular Points of Smooth Mappings.* Research Notes in Mathematics **25**, Pitman, London 1979.

[8] M. Golubitsky and V. Guillemin. *Stable Mappings and Their Singularities*. Graduate Texts in Mathematics **14**, Springer, New York 1973. DOI: 10.1007/978-1-4615-7904-5_3
[9] M. Golubitsky, M. Pivato, and I. Stewart. Interior symmetry and local bifurcation in coupled cell networks. *Dyn. Sys.* **19** (2004) 389–407. DOI: 10.1080/14689360512331318006
[10] M. Golubitsky and I. Stewart. Nonlinear dynamics of networks: the groupoid formalism. *Bull. Amer. Math. Soc.*, **43** (2006) 305–364. DOI: 10.1090/s0273-0979-06-01108-6
[11] M. Golubitsky and I. Stewart. Homeostasis, singularities, and networks. *J. Mathematical Biology* **74** (2017) 387–407. DOI: 10.1007/s00285-016-1024-2
[12] M. Golubitsky and I. Stewart. Coordinate changes for network dynamics. *Dynamical Systems* **32** (2017) 80–116. [*Equivariance and Beyond: M. Golubitsky's 70th Birthday*.] DOI: 10.1080/14689367.2016.1235136
[13] M. Golubitsky and I. Stewart. Homeostasis with multiple inputs. *SIAM J. Applied Dynamical Systems* **17** (2018) 1816–1832. DOI: 10.1137/17m115147x
[14] M. Golubitsky and I. Stewart. *Dynamics and Bifurcation in Networks*. In preparation 2021.
[15] M. Golubitsky, I. Stewart, and A. Török. Patterns of synchrony in coupled cell networks with multiple arrows. *SIAM J. Appl. Dynam. Sys.* **4** (2005) 78–100. DOI: 10.1137/040612634
[16] M. Golubitsky and Y. Wang. Infinitesimal homeostasis in three-node input-output networks. *J. Math. Biol.* **80** (2020) 1163–1185. DOI: 10.1007/s00285-019-01457-x
[17] P.J. Higgins. *Notes on Categories and Groupoids*. Van Nostrand Reinhold Mathematical Studies **32**, Van Nostrand Reinhold, London 1971.
[18] M.C.A. Leite and M. Golubitsky. Homogeneous three-cell networks. *Nonlinearity* **19** (2006) 2313–2363. DOI: 10.1088/0951-7715/19/10/004
[19] J. Martinet. *Singularities of Smooth Functions and Maps* (transl. C. Simon). London Math. Soc. Lect. Notes **58**, Cambridge University Press, Cambridge 1982.
[20] E. Nijholt. *Bifurcations in Network Dynamical Systems*. PhD thesis, Vrije Universiteit Amsterdam 2018.
[21] E. Nijholt, B. Rink, and J. Sanders. Graph fibrations and symmetries of network dynamics. *J. Diff. Eq.* **261** (2016) 4861–4896. DOI: 10.1016/j.jde.2016.07.013
[22] E. Nijholt, B. Rink, and J. Sanders. Center manifolds of coupled cell networks. *SIAM J. Math. Anal.* **49** (2017) 4117–4148. DOI: 10.1137/16m106861x
[23] E. Nijholt, B. Rink, and S.Schwenker. Quiver representations and dimension reduction in dynamical systems. *SIAM J. Appl. Dyn. Syst.* **19** (4) (2020) 2428–2468. DOI: 10.1137/20m1345670
[24] B. Rink and J. Sanders, Amplified Hopf bifurcations in feed-forward networks. *SIAM J. Appl. Dyn. Syst.* **12** (2) (2013) 1135–1157. DOI: 10.1137/120899649
[25] B. Rink and J. Sanders, Coupled cell networks and their hidden symmetries. *SIAM J. Math. Anal.* **46** (2) (2014) 1577–1609. DOI: 10.1137/130916242
[26] B. Rink and J. Sanders, Coupled cell networks: semigroups, Lie algebras and normal forms. *Trans. Amer. Math. Soc.* **367** (5) (2015) 3509–3548. DOI: 10.1090/s0002-9947-2014-06221-1
[27] M. Reed, J.Best, M. Golubitsky, I. Stewart, and F. Nijhout. Analysis of homeostatic mechanisms in biochemical networks. *Bull. Math. Biology* **79** (2017) 2534–2557. DOI: 10.1007/s11538-017-0340-z
[28] I. Stewart, M. Golubitsky, and M. Pivato. Symmetry groupoids and patterns of synchrony in coupled cell networks. *SIAM J. Appl. Dynam. Sys.* **2** (4) (2003) 609–646. DOI: 10.1137/s1111111103419896
[29] Z.F. Tang and D.R. McMillen. Design principles for the analysis and construction of robustly homeostatic biological networks. *J. Theor. Biol.* **408** (2016) 274–289. DOI: 10.1016/j.jtbi.2016.06.036
[30] Y. Wang, Z. Huang, F. Antoneli, and M. Golubitsky. The structure of infinitesimal homeostasis in input-output networks. *J. Math. Biol.* **82** (2021) 62. DOI: 10.1007/s00285-021-01614-1

Fernando Antoneli, Centro de Bioinformática Médica, Universidade Federal de São Paulo, São Paulo SP 04023-062, Brazil

Ian Stewart, Mathematics Institute, University of Warwick, Coventry CV4 7AL, UK