CAMBRIDGE
UNIVERSITY PRESS

PAPER

Noncommutative network models

Joe Moeller[iD]

Department of Mathematics, University of California, Riverside, Riverside, CA, USA
Email: moeller@math.ucr.edu

Abstract

Network models, which abstractly are given by lax symmetric monoidal functors, are used to construct operads for modeling and designing complex networks. Many common types of networks can be modeled with simple graphs with edges weighted by a monoid. A feature of the ordinary construction of network models is that it imposes commutativity relations between all edge components. Because of this, it cannot be used to model networks with bounded degree. In this paper, we construct the free network model on a given monoid, which can model networks with bounded degree. To do this, we generalize Green's graph products of groups to pointed categories which are finitely complete and cocomplete.

Keywords: graph product; universal algebra; Kneser graph; monoidal category; network

1. Network Models

One way to combine two graphs is to identify the vertices of one with some of the vertices of the other in a one-to-one way, then gluing the two graphs together at the identified vertices. We can decompose such a combination into a sequence of simpler operations of three types: disjoint union of any two graphs, gluing two graphs together which have the same vertex set, which we call *overlay*, and permutation of vertices. In previous work, *network models* were introduced to formally encode these operations (Baez et al. 2018). The algebras of a network operad can serve as tools for designing complex multi-agent networks. Network operads are constructed from network models, which are certain symmetric lax monoidal functors. There is a functorial construction of a network model from a monoid, which we call the *ordinary network model for weighted graphs*. In this paper, we provide a different construction in order to realize a larger class of networks as algebras of network operads, which we call the *free varietal network model for weighted graphs*. In Section 4, we give an example of a family of networks which cannot form an algebra for any ordinary network model for weighted graphs, but does for a varietal one.

The reader is assumed to be familiar with basic notions from category theory (Mac Lane 1998), especially symmetric monoidal categories and lax symmetric monoidal functors (Joyal and Street 1993). Let S be the *symmetric groupoid*, that is, the category with objects $\mathbf{n} = \{1, \ldots, n\}$ (including the empty set for $\mathbf{0}$) and bijections for morphisms. Let Mon denote the category of monoids. A *one-colored network model* is a symmetric lax monoidal functor

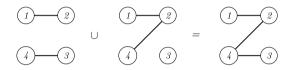$$(F, \Phi) \colon (\mathsf{S}, +) \to (\mathsf{Mon}, \times),$$

where Φ is the *laxator* of F, that is, a natural transformation with components

$$\Phi_{x,y} \colon F\mathbf{n} \times F\mathbf{m} \to F(\mathbf{n} + \mathbf{m}).$$

We call the monoids $F(\mathbf{n})$ the *constituent monoids* of the network model F. There is a more general notion of network model which replaces the category S with a free symmetric monoidal category. We do not consider this generalization here, so we always mean a one-colored network model when we say network model. Let NetMod denote the category of network models with monoidal natural transformations as morphisms.

Essentially, a network model is a family of monoids $\{M_n\}_{n \in \mathbb{N}}$ each with a group action of the corresponding symmetric group S_n, such that the product of any two embed into the one indexed by the sum of their indices equivariantly, that is, in a way which respects the group action: $M_m \times M_n \hookrightarrow M_{m+n}$.

Example 1. For a set X and $k \in \mathbb{N}$, let $\binom{X}{k}$ denote the set of k-element subsets of X. A simple graph on $\mathbf{n}$ is a subset of $\binom{\mathbf{n}}{2}$. The set $\mathbf{n}$ is the set of vertices, and each two-element subset is an edge. Let $SG(\mathbf{n})$ be the monoid whose underlying set is $2^{\binom{\mathbf{n}}{2}}$, the set of all simple graphs with vertex set $\mathbf{n}$, and whose monoid operation is union.

The group S_n acts on $SG(\mathbf{n})$ by permuting the vertices, so we have a functor $SG\colon \mathsf{S} \to \mathsf{Mon}$. Disjoint union of graphs defines a family of monoid homomorphisms $\sqcup\colon SG(\mathbf{m}) \times SG(\mathbf{n}) \to SG(\mathbf{m}+\mathbf{n})$. These maps form a natural transformation which acts as the laxator, making $(SG, \sqcup)\colon (\mathsf{S}, +) \to (\mathsf{Mon}, \times)$ a lax symmetric monoidal functor, thus a network model.

The simple graph network model is the motivating example for network models. Since the operation in the constituent monoids of SG is defined by union, we use $\cup$ to denote the operation in all the constituent monoids for all network models. Since the elements of the monoids $SG(\mathbf{n})$ are graphs, for a general network model F, the elements of the monoids $F(\mathbf{n})$ are called F-*networks*. Since a network model is a lax symmetric monoidal functor, we can apply a symmetric monoidal variant of the Grothendieck construction to obtain a symmetric monoidal category. From this, we obtain the *underlying operad*. The main result on network models is that given any network model $F\colon \mathsf{S} \to \mathsf{Mon}$, there is an $\mathbb{N}$-typed operad O_F with F-networks as its operations (Baez et al. 2018).

This gives us a method of obtaining operads that are useful for modeling complex networks of various sorts. We start with a suitable network model and then apply this theorem to obtain an operad. This leads to the question: which kinds of network can be described by network models?

One large class of network models comes from monoids (Baez et al. 2018). Indeed, for any monoid M, there is a network model for which the networks are simple graphs weighted by M. In the simple graphs example, $SG(\mathbf{n})$ is the monoid whose underlying set was $2^{\binom{\mathbf{n}}{2}}$ and whose monoid operation was union. This could be said more succinctly by letting $\mathbb{B}$ be the boolean monoid, $(\{T, F\}, \vee)$ and then defining $SG(\mathbf{n}) = \mathbb{B}^{\binom{\mathbf{n}}{2}}$. We interpret an element of $SG(\mathbf{n})$, which is a function $g\colon \binom{\mathbf{n}}{2} \to \mathbb{B}$, as a graph on $\mathbf{n}$ with an edge between $u \in \mathbf{n}$ and $v \in \mathbf{n}$ if and only if $g(\{u, v\}) = T$. For a given monoid, let Γ_M be the network model defined by $\Gamma_M(\mathbf{n}) = M^{\binom{\mathbf{n}}{2}}$, and everything else defined as with simple graphs. As with $SG(\mathbf{n})$, an element of $\Gamma_M(\mathbf{n})$ should be thought of as a simple graph, but with edges weighted by values in M. This construction is functorial:

Theorem 2 (c.f. Baez et al. 2018, Theorem 22). *There is a functor $\Gamma\colon \mathsf{Mon} \to \mathsf{NetMod}$ sending any monoid M to the network model Γ_M and any homomorphism of monoids $f\colon M \to M'$ to the morphism of network models $\Gamma_f\colon \Gamma_M \to \Gamma_{M'}$. The network model Γ_M is called the* ordinary network model for M-weighted graphs *or the* ordinary M network model.

This construction is designed to model networks which carry information on the edges. For example, with $\mathbb{N}$ a monoid under addition, $\Gamma_{\mathbb{N}}$ is a network model for loopless undirected multigraphs where overlaying is given by adding the number of edges. A similar example is $\Gamma_{\mathbb{B}} = \text{SG}$. There is a monoid homomorphism $\mathbb{N} \to \mathbb{B}$ which sends all but 0 to T. This induces a map of network models $\Gamma_{\mathbb{N}} \to \Gamma_{\mathbb{B}}$. Essentially, this map reduces the information of a graph from the number of connections between each pair of vertices to just the existence of any connection.

Example 3 (**Algebra for range-limited communication**). Consider a communication network where each node represents a boat and an edge between two nodes represents a working communication channel between the corresponding boats. Some forms of communication are restricted by the distance between those communicating. Assume that there is a known maximal distance over which our boats can communicate. Networks of this sort form an algebra of the simple graphs operad in the following way.

Let (X, d) be a metric space, and $0 \le L \in \mathbb{R}$. Our boats will be located at points in this space. The operad O_{SG} has an algebra $(A_{d,L}, \alpha)$ defined as follows. The set $A_{d,L}(\mathbf{n})$ is the set of pairs (h, f) where $h \in \text{SG}(\mathbf{n})$ is a simple graph and $f : \mathbf{n} \to X$ is a function such that if $\{v_1, v_2\}$ is an edge in g then $d(f(v_1), f(v_2)) \le L$. The number L represents the maximal distance over which the boat's communication channels operate. Notice that this condition does not demand that all connections within range must be made. An operation $(\sigma, g) \in O_{\text{SG}}(\mathbf{n}_1, \ldots, \mathbf{n}_k; \mathbf{n})$ acts on a k-tuple $(h_i, f_i) \in A_{d,L}(\mathbf{n}_i)$ by

$$\alpha(\sigma, g)((h_1, f_1), \ldots, (h_k, f_k)) = (g \cup \sigma(h_1 \sqcup \cdots \sqcup h_k), f_1 \sqcup \cdots \sqcup f_k).$$

Elements of this algebra are simple graphs in the space X with an upper limit on edge lengths. When an operation acts on one of these, it tries to put new edges into the graph, but fails to when the range limit is exceeded (Baez et al. 2018).

A characteristic of the construction given in Theorem 2 is that elements of the resulting monoids that correspond to different edges automatically commute with each other. For example, for a monoid M, the fourth constituent monoid of the ordinary M network model is $\Gamma_M(4) = M^6$. Then, the element $(m_1, 0, 0, 0, 0, 0)$ represents a graph with one edge with weight $m_1 \in M$, the element $(0, m_2, 0, 0, 0, 0)$ represents a graph with a different edge with weight $m_2 \in M$, and

$$(m_1, 0, 0, 0, 0, 0) \cup (0, m_2, 0, 0, 0, 0) = (m_1, m_2, 0, 0, 0, 0)$$
$$= (0, m_2, 0, 0, 0, 0) \cup (m_1, 0, 0, 0, 0, 0).$$

This commutativity between edges means that networks given by ordinary network models cannot record information about the order in which edges were added to it. The ability to record such information about a network is desirable, for example, if one wishes to model networks which have a limit on the number of connections, each agent can make to other agents.

The *degree* of a vertex in a simple graph is the number of edges which include that vertex. The *degree* of a graph is the maximum degree of its vertices. A graph is said to have *degree bounded by* k, or simply *bounded degree*, if the degree of each vertex is less than or equal to k. Let $B_k(\mathbf{n})$ denote the set of networks with $\mathbf{n}$ vertices and degree bound k. One might guess that the family of such networks could form an algebra for the simple graphs operad.

Question 4. Does the collection of networks of bounded degree form an algebra of a network operad? If so, is there such an algebra which is useful in applications?

Specifically, can networks of bounded degree form an algebra of O_{SG}, the simple graph operad? Setting two graphs next to each other will not change the degree of any of the vertices. Overlaying them almost definitely will, which makes defining an action of $\text{SG}(\mathbf{n})$ on $B_k(\mathbf{n})$ less obvious.

Ordinary network models are not sufficient to model this type of network because the graph monoids it produced could not remember the order that edges were added into a network. Even if M is a noncommutative monoid, since Γ_M is a product of several copies of M, one for each pair of vertices, it cannot distinguish the order that two different edges touching v_1 were added to a network if their other endpoints are different.

Instead of taking the product of $\binom{n}{2}$ copies of M, we consider taking the coproduct, so as not to impose any commutativity relations between the edges. Since the lax structure map $\sqcup\colon F(\mathbf{m}) \times F(\mathbf{n}) \to F(\mathbf{m}+\mathbf{n})$ associated with a network model $F\colon \mathsf{S} \to \mathsf{Mon}$ must be a monoid homomorphism, then

$$(a \sqcup b) \cup (c \sqcup d) = (a \cup c) \sqcup (b \cup d).$$

In particular, if we let $\emptyset$ denote the the identity of $F(\mathbf{n})$ for any $\mathbf{n}$, then

$$\begin{aligned}(a \sqcup \emptyset) \cup (\emptyset \sqcup b) &= (a \cup \emptyset) \sqcup (\emptyset \cup b) \\ &= (\emptyset \cup a) \sqcup (b \cup \emptyset) \\ &= (\emptyset \sqcup b) \cup (a \sqcup \emptyset).\end{aligned}$$

This is reminiscent of the Eckmann–Hilton argument, but notice that the domains of the operations $\cup$ and $\sqcup$ are not the same. This equation says that elements which correspond to disjoint edges must commute with each other. Simply taking the coproduct of $\binom{n}{2}$ copies of M cannot give the constituent monoids of a network model.

For a collection of monoids $\{M_i\}_{i \in I}$, elements of the product monoid which come from different components always commute with each other. In the coproduct, they never do. A *graph product* (in the sense of Green (1990)) of such a collection allows one to impose commutativity between certain components and not others by indicating such relations via a simple graph. The calculation above shows that the constituent monoids of a network model must satisfy certain partial commutativity relations. We use graph products to construct a family of monoids with the right amount of commutativity to both answer Question 4 and satisfy the conditions of being a network model. The following theorems are proven in Section 3.

Theorem. *The functor* $\mathsf{NetMod} \to \mathsf{Mon}$ *defined by* $F \mapsto F(\mathbf{2})$ *has a left adjoint* $\Gamma_{-,\mathsf{Mon}}\colon \mathsf{Mon} \to \mathsf{NetMod}$.

The fact that this construction is a left adjoint tells us that the network models constructed are ones in which the only relations that hold are those that follow from the defining axioms of network models.

A *variety* of monoids is the class of all monoids satisfying a given set of identities. For example, Mon has subcategories CMon of commutative monoids and GMon of graphic monoids which are varieties of monoids satisfying the equations $ab = ba$ and $aba = ab$, respectively. Given a variety of monoids $\mathcal{V}$, let $\mathsf{NetMod}_{\mathcal{V}}$ be the subcategory of NetMod consisting of $\mathcal{V}$-valued network models. We recreate graph products in varieties of monoids to obtain a more general result.

Theorem. *The functor* $\mathsf{NetMod}_V \to \mathcal{V}$ *defined by* $F \mapsto F(\mathbf{2})$ *has a left adjoint* $\Gamma_{-,\mathcal{V}}\colon \mathcal{V} \to \mathsf{NetMod}_V$.

In particular, if $\mathcal{V} = \mathsf{CMon}$, since products and coproducts are the same in CMon, the ordinary M network model and the CMon varietal M network model are also the same. Note that this does not indicate that $\Gamma_{-,\mathcal{V}}$ is a complete generalization of Γ_- from Theorem 2, since Γ_M is not an example of $\Gamma_{-,\mathcal{V}}$ when M is not commutative.

The ordinary construction for a network model given a monoid M has constituent monoids given by finite cartesian powers of M. To include the networks described in Question 4 into the

theory of network models, we must construct a network model from a given monoid which does not impose as much commutativity as the ordinary construction does, specifically among elements corresponding to different edges. The first attempt at a solution is to use coproducts instead of products. However, in this section, we saw that we cannot create the constituent monoids of a network model simply by taking them to be coproducts of M instead of products. There must be some commutativity between different edges, specifically between edges which do not share a vertex.

Given a monoid M, we want to create a family of monoids indexed by $\mathbb{N}$, the nth of which looks like a copy of M for each edge in the complete graph on $\mathbf{n}$, has minimal commutativity relations between these edge components, but does have commutativity relations between disjoint edges. Partial commutativity like this can be described with Green's graph products, which we describe in Section 2.1. The type of graph which describes disjointness of edges in a graph as we need is called a *Kneser graph*, which we describe in Section 2.2. Besides concerning ourselves with relations between edge components, sometimes we also want the constituent monoids in a network model to obey certain relations which M obeys. In Section 2.3, we describe *varieties of monoids* and a construction which produces monoids in a chosen variety. In Section 3, we prove this construction is functorial, and in Section 4, we use this construction to give a positive answer to Question 4.

2. Graphs

This section is dedicated to constructing the constituent monoids for the network models we want. In this section, there are two different ways that graphs are being used. It is important that the reader does not get these confused. One way is the graphs which are elements of the constituent monoids of the network models we are constructing. The other way we use graphs is to index the *Green product* (which we define in Section 2.1) to describe commutativity relations in the constituent monoids of the network models we are constructing.

A network model is essentially a family of monoids with properties similar to the simple graphs example, so we think of the elements of these monoids as graphs, and we think of the operation as overlaying the graphs. These monoids have partial commutativity relations they must satisfy, as we see in Section 1. The graphs we use in the Green product, the Kneser graphs, are there to describe the partial commutativity in the constituent monoids.

2.1 Green products

Given a family of monoids $\{M_v\}_{v\in V}$ indexed by a set V, there are two obvious ways to combine them to get a new monoid, the product and the coproduct. From an algebraic perspective, a significant difference between these two is whether or not elements that came from different components commute with each other. In the product, they do. In the coproduct, they do not. *Green products*, or commonly *graph products*, of groups were introduced in 1990 by Green (1990) and later generalized to monoids by Veloso da Costa (2001). The idea provides something of a sliding scale of relative commutativity between components. We follow Fountain and Kambites (2009) in the following definitions.

By a *simple graph* $G = (V, E)$, we mean a set V which we call the set of vertices, and a set $E \subseteq \binom{V}{2}$, which we call the set of edges. A *map of simple graphs* $f : (V, E) \to (V', E')$ is a function $f : V \to V'$ such that if $\{u, v\} \in E$, then $\{f(u), f(v)\} \in E'$. Let $\mathsf{SimpGph}$ denote the category of simple graphs and maps of simple graphs.

For a set V, a family of monoids $\{M_v\}_{v\in V}$, and a simple graph $G = (V, E)$, the *G Green product* (or simply *Green product* when unambiguous) of $\{M_v\}_{v\in V}$, denoted $G(M_v)$, is

$$G(M_v) = \left(\coprod_{v\in V} M_v \right) / R_G,$$

where R_G is the congruence generated by the relation

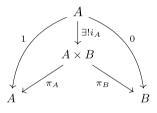$$\{(mn, nm)\mid m \in M_v, n \in M_u, u, v \text{ are adjacent in } G\},$$

where the operation in the free product is denoted by concatenation. If G is the complete graph on n vertices, then $G(M_v) \cong \prod M_v$. If G is the n-vertex graph with no edges, then $G(M_v) \cong \coprod M_v$.

We call each M_v a *component* of the Green product. Elements of $G(M_v)$ are written as *expressions* as in the free product, $m_1^{v_1} \ldots m_k^{v_k} \in G(M_v)$ where the superscript indicates that $m_i \in M_{v_i}$. We often consider Green products of several copies of the same monoid, so this notation allows one to distguish elements coming from different components of the product, even if they happen to come from the same monoid. The intention and result of the imposed relations are that for an expression $m_1^{v_1} \ldots m_k^{v_k}$ of an element, if there is an i such that $\{v_i, v_{i+1}\} \in E$, then we can rewrite the expression by replacing $m_i^{v_i} m_{i+1}^{v_{i+1}}$ with $m_{i+1}^{v_{i+1}} m_i^{v_i}$. This move is called a *shuffle*, and two expressions are called *shuffle equivalent* if one can be obtained from the other by a sequence of shuffles. An expression $m_1^{v_1} \ldots m_k^{v_k}$ is *reduced* if whenever $i < j$ and $v_i = v_j$, there exists l with $i < l < j$ and $\{v_i, v_l\} \notin E$. If two reduced expressions are shuffle equivalent, they are clearly expressions of the same element. The converse is also true.
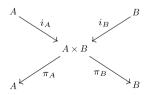
Theorem 5 (Fountain and Kambites 2009, Theorem 1.1). *Every element of M is represented by a reduced expression. Two reduced expressions represent the same element of M if and only if they are shuffle equivalent.*

In this section, we use a categorical description of Green products to define a similar construction in a more general context. The relevant property of Mon that we need for this generalization is that Mon is a *pointed category*.

Let $\mathcal{C}$ be a category. An object of $\mathcal{C}$ which is both initial and terminal is called a *zero object*. If $\mathcal{C}$ has such an object, $\mathcal{C}$ is called a *pointed category* (Quillen 1967). For any two objects A, B of a pointed category, there is a unique map $0\colon A \to B$ which is the composite of the unique map from A to the zero object, and the unique map from the zero object to B. If $\mathcal{C}$ is a pointed category with finite products, then for two objects A, B of $\mathcal{C}$, the objects admit canonical maps $A \to A \times B$.

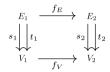So, we have the following maps

satisfying the following properties.

$$\pi_A i_A = 1_A \qquad\qquad \pi_B i_B = 1_B$$
$$\pi_B i_A = 0 \qquad\qquad \pi_A i_B = 0.$$

This is suggestive of a biproduct, but in a general pointed category $A \times B$ is not necessarily isomorphic to $A + B$.

In Section 3, we use a generalized Green product to construct network models. A generalized Green product is a colimit of a diagram whose shape is derived from a given graph. We describe the shapes of the diagrams here with *quivers*.

We assume the reader is familiar with directed multi-graphs. We refer to them here as *quivers* to help distinguish them from other variants of graphs and the role they play in this work. A *quiver* is a pair of sets E, V, respectively, called the *set of edges* and *set of vertices*, and a pair of functions $s, t : E \to V$ assigning to each edge its *starting* vertex and its *terminating* vertex, respectively. A *map of quivers* is a pair of functions

$$
\begin{array}{ccc}
E_1 & \xrightarrow{\ f_E\ } & E_2 \\
s_1 \downarrow\downarrow t_1 & & s_2 \downarrow\downarrow t_2 \\
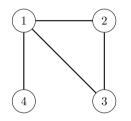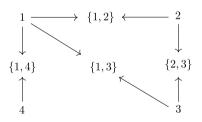V_1 & \xrightarrow{\ f_V\ } & V_2
\end{array}
$$

such that the s-square and the t-square both commute.

We will use the word *cospan* to refer to the quiver with the following shape.

$$\bullet \to \bullet \leftarrow \bullet$$

Define a functor $IC : \mathsf{SimpGph} \to \mathsf{Quiv}$ which replaces every edge with a cospan (IC stands for "insert cospan"). Specifically, given a simple graph (V, E) where $E \subseteq \binom{V}{2}$, define the quiver $Q_1 \rightrightarrows Q_0$ where $Q_0 = V \sqcup E$ and $Q_1 = \{(v, e) \in V \times E \mid v \in e\}$, then define the source map $s : Q_1 \to Q_0$ by projection onto the first component, and the target map $t : Q_1 \to Q_0$ by projection onto the second component. For example, the simple graph

gives the quiver

Let $G = (V, E)$ and $G' = (V', E')$ be simple graphs, and $f : G \to G'$ a map of simple graphs. Define a map of quivers $ICf : IC(G) \to IC(G')$ by $ICf_0 = f_V \sqcup f_E$ and $ICf_1(v, e) = (f_V(v), f_E(e))$.

$$
\begin{array}{ccc}
IC(G)_1 & \xrightarrow{\ ICf_1\ } & IC(G')_1 \\
s_G \downarrow t_G & & s_{G'} \downarrow t_{G'} \\
IC(G)_0 & \xrightarrow{\ ICf_0\ } & IC(G')_0
\end{array}
$$

This construction gives a coproduct preserving functor $IC : \mathsf{SimpGph} \to \mathsf{Quiv}$.

Let $F : \mathsf{Quiv} \to \mathsf{Cat}$ denote the free category (or path category) functor (Mac Lane 1998). Since F is a left adjoint, it preserves colimits. Notice that any quiver of the form $IC(G)$ would never have a path of length greater than 1. Thus, the free path category on $IC(G)$ simply has identity morphisms adjoined.
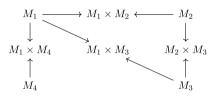
The objects in the category $F(IC(G))$ come from two places. There is an object for each vertex of G, and there is an object at the apex of the cospan for each edge in G. We call these two subsets of objects *vertex objects* and *edge objects*. We abuse notation and refer to the object given by the vertex u by the same name, and similar for edge objects.

If $\{M_v\}_{v \in V}$ is a family of monoids indexed by the set V, that means that there is a functor $M \colon V \to \mathsf{Mon}$ from the set V thought of as a discrete category. Notice that if G is a simple graph with vertex set V, then the discrete category V is a subcategory of $F(IC(G))$. We can then extend the functor M to

$$D \colon F(IC(G)) \to \mathsf{Mon}$$

in the following way. Obviously, we let $D(u) = M_u$ for a vertex object u. If $\{u, v\}$ is an edge in G, then $D(\{u, v\}) = M_u \times M_v$. The morphism $(u, \{u, v\})$ is sent to the canonical map $M_u \to M_u \times M_v$. For example, for a family of monoids $\{M_1, \ldots, M_4\}$, we have the following diagram.

$$
\begin{array}{ccccc}
M_1 & \longrightarrow & M_1 \times M_2 & \longleftarrow & M_2 \\
\downarrow & \searrow & & & \downarrow \\
M_1 \times M_4 & & M_1 \times M_3 & & M_2 \times M_3 \\
\uparrow & & \nwarrow & & \uparrow \\
M_4 & & & & M_3
\end{array}
$$

Since there are no non-trivial pairs of composable morphisms in categories of the form $F(IC(G))$, nothing further needs to be checked to confirm D is a functor.

Despite the way we are denoting these products, we are not considering them to be ordered products. Alternatively, we could have used a more cumbersome notation that does not suggest any order on the factors.

Theorem 6. *Let V be a set, $\{M_v\}_{v \in V}$ be a family of monoids indexed by V, and $G = (V, E)$ be a simple graph with vertex set V. The G Green product of M_v is the colimit of the diagram $D \colon F(IC(G)) \to \mathsf{Mon}$ defined as above*

$$G(M_v) \cong \operatorname{colim} D.$$

Proof. We show that $G(M_v)$ satisfies the necessary universal property. The vertex objects in the diagram have inclusion maps into the edge objects $i_{u,v} \colon M_u \to M_u \times M_v$, and all the objects have inclusion maps into $G(M_v)$, $j_u \colon M_u \to G(M_v)$ and $j_{u,v} \colon M_u \times M_v \to G(M_v)$ such that $j_{u,v} \circ i_{u,v} = j_u$. Note that due to the fact that we have unordered products for objects, there is some redundancy in our notation, namely, $j_{u,v} = j_{v,u}$. If we have a monoid Q and maps $f_u \colon M_u \to Q$ and $f_{u,v} \colon M_u \times M_v \to Q$ such that

$$f_{u,v} = f_{v,u}$$
$$f_{u,v} \circ i_{u,v} = f_u,$$

then we define a map $\phi \colon G(M_v) \to Q$ by $\phi(m_1^{v_1} \ldots m_k^{v_k}) = f_{v_1}(m_1) \ldots f_{v_k}(m_k)$. Since this map is defined via expressions of elements, Theorem 5 tells us that to check this map is well-defined, we only need to check that the values of two expressions that differ by a shuffle are the same. Let $m_1^{v_1} \ldots m_k^{v_k}$ be an expression, and i such that $\{v_i, v_{i+1}\} \in E$.

$$
\begin{aligned}
\phi(m_i^{v_i} m_{i+1}^{v_{i+1}}) &= f_{v_i}(m_i) f_{v_{i+1}}(m_{i+1}) \\
&= f_{v_i, v_{i+1}}(m_i, m_{i+1}) \\
&= f_{v_{i+1}}(m_{i+1}) f_{v_i}(m_i) \\
&= \phi(m_{i+1}^{v_{i+1}} m_i^{v_i})
\end{aligned}
$$

It is clear that

$$\phi(m_1^{v_1} \ldots m_k^{v_k}) = \phi(m_1^{v_1} \ldots m_{i-1}^{v_{i-1}})\phi(m_i^{v_i} m_{i+1}^{v_{i+1}})\phi(m_{i+2}^{v_{i+2}} \ldots m_k^{v_k}),$$

so two shuffle equivalent expressions have the same value under ϕ, and ϕ is well-defined. It is clearly a monoid homomorphism, and has the property $\phi \circ j_u = f_u$ and $\phi \circ j_{u,v} = f_{u,v}$. To show this map is unique, assume there is another such map $\psi : G(M_v) \to Q$. Since $\psi \circ j_u = f_u$, then $\psi(m_u) = f(u)$, and

$$\psi(m_1^{v_1} \ldots m_k^{v_k}) = \psi(m_1^{v_1}) \ldots \psi(m_k^{v_k})$$
$$= f_{v_1}(m_1) \ldots f_{v_k}(m_k)$$
$$= \phi(m_1^{v_1} \ldots m_k^{v_k}). \qquad \square$$

This result makes it reasonable to generalize Green products in the following way.

Definition 7. *Let $\mathcal{C}$ be a pointed category with finite products and finite colimits, V a set, $\{A_v\}_{v \in V}$ a family of objects of $\mathcal{C}$ indexed by V, and G a simple graph with vertex set V. Let $D : F(IC(G)) \to \mathcal{C}$ be the diagram defined by $v \mapsto A_v$, $\{u, v\} \mapsto A_u \times A_v$, and the morphism $(u, \{u, v\})$ is mapped to the inclusion $A_u \to A_u \times A_v$ as above. The G Green product of $\{A_v\}_{v \in V}$ is the colimit of D in $\mathcal{C}$,*
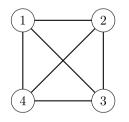
$$G^{\mathcal{C}}(A_v) = \operatorname{colim} D.$$

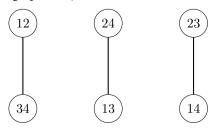*If $\mathcal{C} = \mathsf{Mon}$, we denote the Green product simply as $G(A_v)$.*

In Section 3, we use this general notion of graph products in *varieties of monoids* to construct network models whose constituent monoids are in those varieties. Note that since $F \circ IC$ is a functor, the group $\operatorname{Aut}(G)$ of graph automorphisms of G naturally acts on $G^{\mathcal{C}}(A_v)$.

2.2 Kneser graphs

We focus here on a special family of simple graphs known as the *Kneser graphs* (Lovász 1978). The *Kneser graph* $KG_{n,m}$ has vertex set $\binom{n}{m}$, the set of m-element subsets of an n-element set, and an edge between two vertices if they are disjoint subsets. Since a simple graph is defined as a collection of two-element subsets of an n-element set, the Kneser graph $KG_{n,2}$ has a vertex for each edge in the complete graph on $\mathbf{n}$, and has an edge between every pair of vertices which correspond to disjoint edges. So, the Kneser graph $KG_{n,2}$ can be thought of as describing the disjointness of edges in the complete graph on $\mathbf{n}$. For instance, the complete graph on $\mathbf{4}$ is

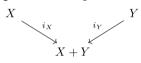and the corresponding Kneser graph $KG_{4,2}$ is

where we label the node with uv if it corresponds to the edge $\{u, v\}$. One can see that an edge in the second graph corresponds exactly to a pair of disjoint edges in the first graph. For another example, $KG_{5,2}$ is the Petersen graph. For sets X, Y and a function $f \colon X \to Y$, let $f[U] = \{f(x) \mid x \in U\}$ for $U \subseteq X$. Let FinInj denotes the category of finite sets and injective functions.

Lemma 8. *For* $k \in \mathbb{N}$, *there is a functor* $\binom{-}{k} \colon$ FinInj $\to$ FinInj *which sends* X *to* $\binom{X}{k}$ *the set of* k-*element subsets of* X, *and injections* $f \colon X \to Y$ *to the functions* $\binom{f}{k} \colon \binom{X}{k} \to \binom{Y}{k}$ *defined by* $\binom{f}{k}(U) = f[U]$.
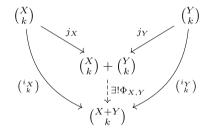
Note that this result holds for Inj the category of sets and injective functions, but we only require FinInj for our purposes.

Proof. If $f \colon X \to Y$ is an injection, then $|f[U]| = |U|$ for $U \subseteq X$. It then makes sense to restrict the induced map on power sets to subsets of a fixed cardinality. The map $\binom{f}{k} \colon \binom{m}{k} \to \binom{n}{k}$ defined by $\binom{f}{k}(U) = f[U]$ is then well-defined. If $f[U] = f[V]$ and $x \in U$, then $f(x) \in f[U] = f[V]$, which implies there is a $y \in V$ such that $f(y) = f(x)$. Since f is injective, then $x = y \in V$. Thus, $U = V$ by symmetry. $\square$

Let i_X and i_Y denote the following inclusion maps.

Since these maps are injective, they induce maps $\binom{i_X}{k}, \binom{i_Y}{k}$ and we get a map $\Phi_{X,Y} \colon \binom{X}{k} + \binom{Y}{k} \to \binom{X+Y}{k}$ by the universal property in the following way.

Lemma 9. *The functor* $\binom{-}{k}$ *is made lax symmetric monoidal*

$$\left(\binom{-}{k}, \Phi, \phi \right) \colon (\mathsf{FinInj}, +, \emptyset) \to (\mathsf{FinInj}, +, \emptyset)$$

where the components of Φ *are defined as above.*

Proof. The family of maps $\{\Phi_{X,Y}\}$ is clearly a natural transformation. There is no choice for the map $\phi \colon \emptyset \to \binom{\emptyset}{k}$. The left and right unitor laws hold trivially. Checking the coherence conditions for the associator and the symmetry are straightforward computations. $\square$

For $n, k \in \mathbb{N}$, the simple graph $KG_{n,k}$ has vertex set $V = \binom{\mathbf{n}}{k}$ and edge set $\{\{u, v\} \subseteq \binom{V}{2} \mid u \cap v = \emptyset\}$. If $f \colon \mathbf{m} \to \mathbf{n}$ is injective, then we get a map $\binom{f}{k}$ between the vertex sets of $KG_{m,k}$ and $KG_{n,k}$. Let $\{u, v\} \in \binom{V}{2}$ be an edge in $KG_{m,k}$. Then $f[u] \cap f[v] = \emptyset$ by injectivity, so $\{f[u], f[v]\}$ is an edge of $KG_{n,k}$. An injection f then induces a map of graphs, denoted $KG_{f,k} \colon KG_{m,k} \to KG_{n,k}$. Since $\binom{f}{k}$ is injective, $KG_{f,k}$ is an embedding. Nothing about

this construction requires finiteness of the sets involved, but our applications only call for finite graphs.

Proposition 10. *For $k \in \mathbb{N}$, there is functor $KG_{-,k} \colon \mathsf{FinInj} \to \mathsf{SimpGph}$ which sends $\mathbf{n}$ to $KG_{n,k}$ and $f \colon \mathbf{m} \to \mathbf{n}$ to $KG_{f,k}$.*

Not only does $KG_{m,k}$ embed into $KG_{n,k}$ when $m < n$ but also $KG_{m,k} + KG_{n,k}$ embeds into $KG_{m+n,k}$. We construct the embedding $KG_{m,k} + KG_{n,k} \to KG_{m+n,k}$ by using the lax structure map from Lemma 9 for the vertex map, $\Phi_{\mathbf{m},\mathbf{n}} \colon \binom{\mathbf{m}}{k} + \binom{\mathbf{n}}{k} \to \binom{\mathbf{m}+\mathbf{n}}{k}$. Restricting this map to either $\binom{\mathbf{m}}{k}$ (resp. $\binom{\mathbf{n}}{k}$) gives the map $\binom{i\mathbf{m}}{k}$ (resp. $\binom{i\mathbf{n}}{k}$) which we already know induces a map of graphs. Thus, $\Phi_{\mathbf{m},\mathbf{n}}$ induces a map of graphs, which we call $\Psi_{\mathbf{m},\mathbf{n}}$.

Proposition 11. *The functor $KG_{-,k}$ is made lax (symmetric) monoidal*

$$(KG_{-,k}, \Psi) \colon (\mathsf{Inj}, +) \to (\mathsf{SimpGph}, +),$$

where the components of Ψ are defined as above.

Proof. All the necessary properties for Ψ are inherited immediately from Φ. $\qquad\square$

Let $(L, \Lambda) \colon (\mathsf{Inj}, +) \to (\mathsf{Cat}, +)$ be the composite $L = F \circ IC \circ KG_{-,2}$ with the obvious laxator. Let M be a monoid. Then from the construction given in the previous subsection, for each $\mathbf{n}$, we get a diagram $D_n \colon L(\mathbf{n}) \to \mathsf{Mon}$ which sends all vertex objects to M, all edge objects to $M \times M$, and all nontrivial morphisms to inclusions $M \to M \times M$. Taking the colimit of D_n then gives the Green product $KG_{n,2}(M)$.
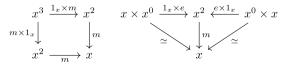
Note that we identify constituent monoids with the corresponding submonoid of the graph product when this can be done without confusion.

Proposition 12. *Let $M_{p,q}$ be a $\binom{m+n}{2}$ family of monoids, and G_1 and G_2 be graphs with m and n vertices, respectively. Let $a_1 \in M_{p_1,q_1}$ with $p_1, q_1 \leq m$ and $a_2 \in M_{p_2,q_2}$ with $p_2, q_2 > m$, and let $\bar{a}_1, \bar{a}_2$ be their values under the canonical inclusions $M_{p,q} \hookrightarrow (G_1 \sqcup G_2)(M_{p,q})$. Then $\bar{a}_1 \bar{a}_2 = \bar{a}_2 \bar{a}_1$ in $(G_1 \sqcup G_2)(M_{p,q})$.*

Proof. By assumption, there is an edge in $KG_{m+n,2}$ between the vertices p_1, q_1 and p_2, q_2. $\qquad\square$

2.3 Varieties of monoids

A *finitary algebraic theory* or *Lawvere theory* is a category T with finite products in which every object is isomorphic to a finite cartesian power $x^n = \prod^n x$ of a distinguished object x (Adámek et al. 2003; Lawvere 1963). An *algebra* of a theory T, or T-*algebra*, is a product preserving functor $T \to \mathsf{Set}$. Let $T\mathsf{Alg}$ denotes the category of T-algebras with natural transformations for morphisms. We are primarily concerned with monoids in this paper. The theory of monoids T_{Mon} has morphisms $m \colon x \times x \to x$ and $e \colon x^0 \to x$, which makes the following diagrams commute.

$$
\begin{array}{ccc}
x^3 & \xrightarrow{1_x \times m} & x^2 \\
{\scriptstyle m \times 1_x}\downarrow & & \downarrow{\scriptstyle m} \\
x^2 & \xrightarrow{m} & x
\end{array}
\qquad\qquad
\begin{array}{ccccc}
x \times x^0 & \xrightarrow{1_x \times e} & x^2 & \xleftarrow{e \times 1_x} & x^0 \times x \\
& {\scriptstyle \simeq}\searrow & \downarrow{\scriptstyle m} & \swarrow{\scriptstyle \simeq} & \\
& & x & &
\end{array}
$$

A *variety* of T-algebras is a full subcategory of $T\mathsf{Alg}$ which is closed under products, subobjects, and homomorphic images. Birkhoff's theorem implies that this is equivalent to the category $T'\mathsf{Alg}$ of algebras of another theory T' which has the same morphisms, but satisfies more commutative diagrams (Burris and Sankappanavar 1981). For example, commutative monoids are given by
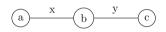
algebras of the theory of commutative monoids T_{CMon}, which has morphisms m, e as in T_{Mon}, not only satisfies the same commutative diagrams as T_{Mon} but also satisfies the following commutative diagram

$$
\begin{array}{ccc}
x^2 & \xrightarrow{\;b\;} & x^2 \\
& \searrow{\scriptstyle m} & \downarrow{\scriptstyle m} \\
& & x
\end{array}
$$

where $b : x^2 \to x^2$ is the braid isomorphism. We only use varieties of monoids in this paper, so we give these "extra" conditions by equations, for example, commutative monoids are those which satisfy the equation $ab = ba$ for all elements a, b. We call the extra equations the *defining equations* of the variety.

A *graphic monoid* is a monoid which satisfies the *graphic identity*: $aba = ab$ for all elements a, b. Graphic monoids are algebras of a theory T_{GMon}. A semigroup obeying this relation is known as a *left regular band* (Margolis et al. 2015). The term *graphic monoid* was introduced by Lawvere (Lawvere 1989). Let M be a graphic monoid. If we let b be the unit of M, then the graphic relation says that $a^2 = a$. Every element of M is idempotent. If $a, c \in M$, then $ca = c$ if c already has a as a factor.

Graphic monoids are present when talking about types of information where a piece of information cannot contain the same piece of information twice. A simple example can be seen in the powerset of a given set X, given the structure of a monoid by union. Of course, this example is overly simple because the operation is commutative idempotent, which is stronger than graphic. A more interesting example can be seen by considering the following simple graph.

We will define a monoid structure on the set $M = \{1, a, b, c, x, y\}$ in the following way. First, 1 is a freely adjoined identity element. For $p, q \in M \setminus \{1\}$, define pq as follows. Pick a generic point f in p and a generic point g in q. Then, move a small distance along a straight line path from f to g. We define the product pq to be the component of the graph you land in. Here are some example computations:

$$
\begin{aligned}
ab &= x & aa &= a \\
bc &= y & xb &= x \\
ac &= x & ca &= y
\end{aligned}
$$

The last two demonstrate that this monoid is not commutative. More complicated examples can be constructed by using the same idea for the operation, but applying it to different spaces.

Our motivation for using graphic monoids is that we use the graphic relation to model "commitment" in the following way. Let M be a graphic monoid, where we think of an element of M as a task or list of tasks. If we first commit to doing task x, and then commit to doing task y, then we have the element xy as our task list, indicating that we committed to x before y. If we then try to commit to doing x, the graphic relation saves us from recording this information twice. The relation also preserves the order in which we committed to x and y: if x is a task list of the form $x = ab$, and we have committed to xy, and then try to commit to bc, we get $(xy)(bc) = (aby)(bc) = a(byb)c = a(by)c = abyc = xyc$.

We want to construct a network model from a monoid in a variety $\mathcal{V}$ which has constituent monoids that are also in $\mathcal{V}$. If M is a monoid in a variety $\mathcal{V}$, then each constituent monoid $\Gamma_M(n)$ is a product of several copies of M, and so is also in $\mathcal{V}$ by definition. Thus, the ordinary network model (given in Theorem 2) restricted to a variety gives a functor $\mathcal{V} \to \mathsf{NetMod}_{\mathcal{V}}$, where $\mathsf{NetMod}_{\mathcal{V}}$ denotes the category of $\mathcal{V}$-valued network models.

The free product of two monoids is a monoid, $M + N$ an element of which is given by a list with entries in the set $M \sqcup N$ such that if two consecutive entries of a list are either both elements of M or both elements of N, then the list is identified with the list that is the same everywhere except that those two entries are reduced to one entry occupied by their product. Note that the empty list is identified with both the singleton list consisting of the identity element of M, and the singleton list consisting of the identity element of N. Free products of monoids gives the coproduct in the category of monoids Mon. Free products of monoids are very similar to free products of groups, which can be found in most books introducing group theory (Hungerford 1974).

If two monoids M and N are in a variety $\mathcal{V}$, taking their free product will not necessarily produce a monoid in $\mathcal{V}$, that is, varieties are not necessarily closed under the coproduct of Mon. It is easy to find an example demonstrating this. Consider IMon, the variety of idempotent monoids, that is, monoids satisfying the equation $x^2 = x$ for all elements x. The boolean monoid $\mathbb{B}$ is an object in IMon. The free product of $\mathbb{B}$ with itself $\mathbb{B} + \mathbb{B}$ can be generated by elements a and b which correspond to the element 1 in each copy of $\mathbb{B}$. The element $ab \in \mathbb{B} + \mathbb{B}$ is not idempotent, as $abab \neq ab$. However, every variety $\mathcal{V}$ does have coproducts. The coproduct in a variety of monoids is the quotient of the free product by the congruence relation generated by the variety's defining equations. In Section 3, we give a construction $\mathcal{V} \to \mathsf{NetMod}_\mathcal{V}$ which uses colimits in order to impose minimal relations.

We need the following fact for the main construction of this paper. It follows immediately from the definitions.

Lemma 13. *Every variety of monoids is a pointed category and has finite colimits.*

This lemma tells us that it makes sense to talk about Green products in a variety, which we call *varietal Green products*. In the next section, we use varietal Green products with Kneser graphs to construct network models.
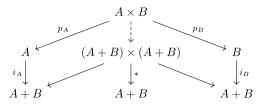
3. Functorial Network Models

In this section, we state and prove the main result of this paper. It says that given a monoid M in a variety $\mathcal{V}$, we can construct a network model whose constituent monoids are also in $\mathcal{V}$, while avoiding to impose commutativity relations when possible. In the following section, we see how this construction resolves the dilemma presented in Question 4.

Let M be a monoid in a variety $\mathcal{V}$. Define $\Gamma_{M,\mathcal{V}}(n)$ to be the $KG_{n,2}$ Green product of $\binom{n}{2}$ copies of M.

Theorem 14. *For $\mathcal{V}$ a variety of monoids, $\Gamma_{-,\mathcal{V}} \colon \mathcal{V} \to \mathsf{NetMod}_\mathcal{V}$ is a functor, as given above. The network model $\Gamma_{M,\mathcal{V}}$ is called the $\mathcal{V}$-varietal network model for M-weighted graphs, or just the varietal M network model.*

In order to prove this, we must first show that a monoid M gives a network model, that is, a lax symmetric monoidal functor. The laxator for $\Gamma_{M,\mathcal{V}}$ is canonically defined, but perhaps it is not as immediate as the one for the ordinary M network model. We treat this first before returning to the proof of the main theorem.
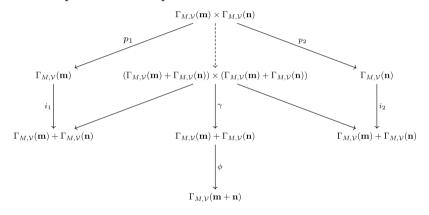
Let A and B be objects in a pointed category with finite products and coproducts. Let $p_A \colon A \times B \to A$ and $p_B \colon A \times B \to B$ denote the canonical projections, and $i_A \colon A \to A + B$ and $i_B \colon B \to A + B$ the canonical inclusions. The category CMon of commutative monoids is such a category. Recall that the operation of a monoid is a monoid homomorphism if and only if the monoid is commutative. We have

$$A \times B$$

$$p_A \qquad \qquad p_B$$

$$A \qquad (A+B) \times (A+B) \qquad B$$

$$i_A \qquad \qquad * \qquad \qquad i_B$$

$$A+B \qquad \qquad A+B \qquad \qquad A+B$$

where $*$ denotes the operation in the commutative monoid $A+B$ and the dashed arrow is $< i_A p_A, i_B p_B >$ given by universal property. The composite of the two maps going down the middle is the inverse to the canonical map $A+B \to A \times B$. The operation in a noncommutative monoid is not a monoid homomorphism, but all the above maps still exist *as functions*. Recall that we let $\cup$ denote the operation in the monoids $\Gamma_{M,\mathcal{V}}(\mathbf{n})$. There is always a homomorphism $\phi_{\mathbf{m},\mathbf{n}} \colon \Gamma_{M,\mathcal{V}}(\mathbf{m}) + \Gamma_{M,\mathcal{V}}(\mathbf{n}) \to \Gamma_{M,\mathcal{V}}(\mathbf{m}+\mathbf{n})$ by universal property of coproducts. Let

$$\gamma \colon (\Gamma_{M,\mathcal{V}}(\mathbf{m}) + \Gamma_{M,\mathcal{V}}(\mathbf{n})) \times (\Gamma_{M,\mathcal{V}}(\mathbf{m}) + \Gamma_{M,\mathcal{V}}(\mathbf{n})) \to \Gamma_{M,\mathcal{V}}(\mathbf{m}) + \Gamma_{M,\mathcal{V}}(\mathbf{n})$$

denote the monoid operation of the coproduct.

$$\Gamma_{M,\mathcal{V}}(\mathbf{m}) \times \Gamma_{M,\mathcal{V}}(\mathbf{n})$$

$$p_1 \qquad \qquad p_2$$

$$\Gamma_{M,\mathcal{V}}(\mathbf{m}) \qquad (\Gamma_{M,\mathcal{V}}(\mathbf{m}) + \Gamma_{M,\mathcal{V}}(\mathbf{n})) \times (\Gamma_{M,\mathcal{V}}(\mathbf{m}) + \Gamma_{M,\mathcal{V}}(\mathbf{n})) \qquad \Gamma_{M,\mathcal{V}}(\mathbf{n})$$

$$i_1 \qquad \qquad \gamma \qquad \qquad i_2$$

$$\Gamma_{M,\mathcal{V}}(\mathbf{m}) + \Gamma_{M,\mathcal{V}}(\mathbf{n}) \qquad \Gamma_{M,\mathcal{V}}(\mathbf{m}) + \Gamma_{M,\mathcal{V}}(\mathbf{n}) \qquad \Gamma_{M,\mathcal{V}}(\mathbf{m}) + \Gamma_{M,\mathcal{V}}(\mathbf{n})$$

$$\phi$$

$$\Gamma_{M,\mathcal{V}}(\mathbf{m}+\mathbf{n})$$

The monoids $\Gamma_{M,\mathcal{V}}(n)$ are constructed specifically so that $\phi \circ \gamma \circ < i_1 \circ p_1, i_2 \circ p_2 >$ is a monoid homomorphism despite the fact that γ is not.

In the proof of the following theorem, we utilize a string diagrammatic calculus suited for reasoning in a symmetric monoidal category. We refer the reader to Selinger's thorough exposition of such string diagramatic languages and their use in category theory (Selinger 2011).

Lemma 15. *The function* $\Gamma_{M,\mathcal{V}}(\mathbf{m}) \times \Gamma_{M,\mathcal{V}}(\mathbf{n}) \to \Gamma_{M,\mathcal{V}}(\mathbf{m}+\mathbf{n})$ *given by* $\phi \circ (i_1 \circ p_1 \cup i_2 \circ p_2)$ *is a monoid homomorphism. Moreover, the family of maps of this form gives a natural transformation, denoted* $\sqcup$.

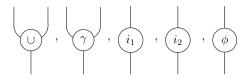*Proof.* We have the following actors in play:

- the monoid operations $\cup_k \colon \Gamma_{M,\mathcal{V}}(\mathbf{k})$ for $k = m, n, m+n$ (we leave off the subscripts below);
- the monoid operation of the coproduct

$$\gamma \colon (\Gamma_{M,\mathcal{V}}(\mathbf{m}) + \Gamma_{M,\mathcal{V}}(\mathbf{n})) \times (\Gamma_{M,\mathcal{V}}(\mathbf{m}) + \Gamma_{M,\mathcal{V}}(\mathbf{n})) \to \Gamma_{M,\mathcal{V}}(\mathbf{m}) + \Gamma_{M,\mathcal{V}}(\mathbf{n});$$
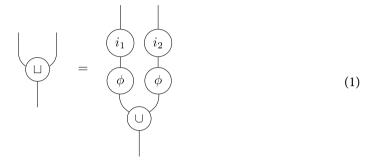
- the canonical inclusion maps $i_1 \colon \Gamma_{M,\mathcal{V}}(\mathbf{m}) \to \Gamma_{M,\mathcal{V}}(\mathbf{m}) \to \Gamma_{M,\mathcal{V}}(\mathbf{n})$ and $i_2 \colon \Gamma_{M,\mathcal{V}}(\mathbf{n}) \to \Gamma_{M,\mathcal{V}}(\mathbf{m}) \to \Gamma_{M,\mathcal{V}}(\mathbf{n})$;
- the canonical map $\phi \colon \Gamma_{M,\mathcal{V}}(\mathbf{m}) + \Gamma_{M,\mathcal{V}}(\mathbf{n}) \to \Gamma_{M,\mathcal{V}}(\mathbf{m}+\mathbf{n})$.

We represent these string diagramatically (read from top to bottom) as follows. Note that these are digrams in Set with its cartesian monoidal structure because the monoid operations $\cup_k$ and γ
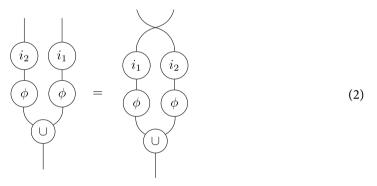
are not necessarily monoid homomorphisms.

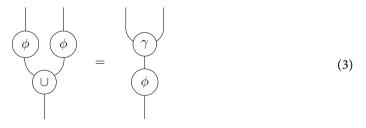$$\sqcup \quad , \quad \gamma \quad , \quad i_1 \quad , \quad i_2 \quad , \quad \phi$$

We define $\sqcup \colon \Gamma_{M,\mathcal{V}}(\mathbf{m}) \times \Gamma_{M,\mathcal{V}}(\mathbf{n}) \to \Gamma_{M,\mathcal{V}}(\mathbf{m}+\mathbf{n})$ as follows.
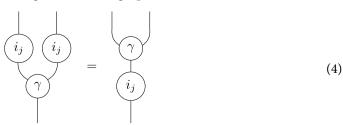
$$\tag{1}$$

Proposition 12 gives the following equation.

$$\tag{2}$$

Since ϕ is a homomorphism, we get the following equation.

$$\tag{3}$$

Since i_1 and i_2 are homomorphisms, we get the following equations.

$$\tag{4}$$

We want to show that $(g \sqcup h) \cup (g' \sqcup h') = (g \cup g') \sqcup (h \cup h')$. We compute:

$\overset{(1)}{=}$

$=$ $\overset{(2)}{=}$ $=$

$\overset{(3)}{=}$ $\overset{(4)}{=}$ $\overset{(1)}{=}$

Let $\sigma \in S_m$ and $\tau \in S_n$. Then

$$\Gamma_{M,\mathcal{V}}(\sigma + \tau)(g \sqcup h) = \Gamma_{M,\mathcal{V}}(\sigma + \tau)\phi(i_1(g) \cup i_2(h))$$
$$= \Gamma_{M,\mathcal{V}}(\sigma)\phi(i_1(g)) \cup \Gamma_{M,\mathcal{V}}(\tau)\phi(i_2(h))$$
$$= \Gamma_{M,\mathcal{V}}(\sigma(g)) \sqcup \Gamma_{M,\mathcal{V}}(\tau(h)),$$

so the following diagram commutes.

$$
\begin{array}{ccc}
\Gamma_{M,\mathcal{V}}(m) \times \Gamma_{M,\mathcal{V}}(n) & \xrightarrow{\sqcup} & \Gamma_{M,\mathcal{V}}(m + n) \\
{\scriptstyle \Gamma_{M,\mathcal{V}}(\sigma) \times \Gamma_{M,\mathcal{V}}(\tau)} \downarrow & & \downarrow {\scriptstyle \Gamma_{M,\mathcal{V}}(\sigma + \tau)} \\
\Gamma_{M,\mathcal{V}}(m) \times \Gamma_{M,\mathcal{V}}(n) & \xrightarrow[\sqcup]{} & \Gamma_{M,\mathcal{V}}(m + n)
\end{array}
$$

Thus, $\sqcup$ is a natural transformation. $\qquad\square$

Proof of Theorem 14. Checking the coherence conditions for ⊔ to be a laxator is a straightforward computation. Let $f\colon M \to N$. Then define the natural transformation $f_{\mathcal{V}}\colon \Gamma_{M,\mathcal{V}} \to \Gamma_{N,\mathcal{V}}$ with components $(f_{\mathcal{V}})_n\colon \Gamma_{M,\mathcal{V}}(n) \to \Gamma_{N,\mathcal{V}}(n)$ given by the universal property. Composition is clearly preserved. □

Theorem 16. *The functor* $\Gamma_{-,\mathcal{V}}$ *is left adjoint to* $E\colon \mathsf{NetMod}_{\mathcal{V}} \to \mathcal{V}$ *where* $E(F) = F(\mathbf{2})$ *for* $(F, \Phi)\colon (\mathsf{S}, +) \to (\mathcal{V}, \times)$ *a* $\mathcal{V}$-*network model.*

Because of this, we call $\Gamma_{M,\mathcal{V}}$ the *free* $\mathcal{V}$-*valued network model on the monoid* M or the *free* $\mathcal{V}$ *network model on* M.

Proof. By construction, $\Gamma_{M,\mathcal{V}}(\mathbf{2}) = M$, so let the unit $\eta = 1_{1_{\mathcal{V}}}\colon 1_{\mathcal{V}} \to \Gamma_{-,\mathcal{V}}(\mathbf{2})$.

We use the universal property of $\Gamma_{M,\mathcal{V}}$ to construct the counit. We define a map $F(\mathbf{2}) \to F(\mathbf{n})$ for each vertex in $\mathsf{KG}_{n,2}$, and a map $F(\mathbf{2}) \times F(\mathbf{2}) \to F(\mathbf{n})$ for each edge in $\mathsf{KG}_{n,2}$.

If $i,j \le n$, then $F((1\,i)(2\,j))\colon F(n) \to F(n)$. If e is the unit of the monoid $F(\mathbf{n}-\mathbf{2})$, and $m \in F(\mathbf{2})$, then $\Phi_{\mathbf{2},\mathbf{n}-\mathbf{2}}(m, e) \in F(n)$. Define maps $c_{i,j}\colon F(\mathbf{2}) \to F(\mathbf{n})$ by

$$c_{i,j} = F((1\,i)(2\,j))(\Phi_{\mathbf{2},\mathbf{n}-\mathbf{2}}(m, e)).$$

The intuition here is that m is a value on one edge of the graph, and e is a graph with $n-2$ vertices and no edges. Then, $\Phi(m, e)$ is the graph with n vertices, and just one m-valued edge between vertices 1 and 2. Then, the permutation $(1\,i)(2\,j)$ permutes this one-edge graph to put m between vertex i and vertex j. So, the map $c_{i,j}$ places the one-edge monoid M at the i,j-position in the n-vertex monoid.

Define maps $c_{i,j,p,q}\colon F(\mathbf{2}) \times F(\mathbf{2}) \to F(\mathbf{n})$ by $c_{i,j,p,q}(\mathbf{m}, \mathbf{m}') = c_{i,j}(\mathbf{m})c_{p,q}(\mathbf{m}')$. The second gives a monoid homomorphism precisely because (F, Φ) is a network model.

Then, we get a map $(\epsilon_F)_{\mathbf{n}}\colon \Gamma_{F(2),\mathcal{V}}(\mathbf{n}) \to F(\mathbf{n})$ by universal property, which gives a monoidal natural transformation automatically. That these maps form the components of a natural transformation can be seen by a routine computation.

Notice that

$$(\epsilon\Gamma_{-,\mathcal{V}})_M = \epsilon_{\Gamma_{M,\mathcal{V}}} = 1_{\Gamma_{M,\mathcal{V}}},$$
$$(\Gamma_{-,\mathcal{V}}\eta)_M = \Gamma_{1_M,\mathcal{V}} = 1_{\Gamma_{M,\mathcal{V}}},$$
$$(E\epsilon)_F = E(\epsilon_F) = (\epsilon_F)_2 = 1_{F(2)},$$
$$(\eta E)_F = \eta_{F(2)} = 1_{F(2)}.$$

Thus, checking that the snake equations hold is routine. □

Example 17. In CMon, products and coproducts are isomorphic. In particular, for a commutative monoid M, $\Gamma_{M,\mathsf{CMon}} \cong \Gamma_M$.

Note that this does not indicate that varietal network models completely encompass ordinary network models. If M is a noncommutative monoid, then $\Gamma_{M,\mathsf{CMon}}$ is not defined, but Γ_M is.

4. Commitment Networks

The motivating example of network models in general is SG, the network model of simple graphs. By Example 17, this network model is an example of the main construction of this paper, $\mathsf{SG} = \Gamma_{\mathbb{B},\mathsf{CMon}}$. The boolean monoid is not only an object in CMon but is also an object in GMon, the variety of graphic monoids. Then, we can consider the network models $\Gamma_{\mathbb{B},\mathsf{Mon}}$ and $\Gamma_{\mathbb{B},\mathsf{GMon}}$.

Example 18. Elements of the monoid $\Gamma_{\mathbb{B},\text{Mon}}(n)$ are words $e_{p_1,q_1} \ldots e_{p_k,q_k}$. These words are interpreted as graphs with edges that look like they were built with popsicle sticks, and if two edges lie directly on top of each other, they are identified. Besides that relation, you can stack edges as high as you want by placing them between different pairs of vertices, but sharing one vertex.

There are networks one could imagine building with this popsicle stick intuition which are not allowed by this formalism. For instance, consider a network with three nodes and an edge for each pair of nodes, each overlapping exactly one of its neighbors, forming an Escher-esque ever-ascending staircase. This sort of network is not allowed by the formalism, since networks are actually equivalence classes of words, where letters have a definite position relative to each other. This is an important feature for this network model as it is necessary to guarantee that the procedure in the following example is well-defined, giving an algebra of the related network operad. What this means in terms of popsicle stick intuition is that allowed networks are built by placing popsicle sticks one at a time.

Example 19. Elements of the $\Gamma_{\mathbb{B},\text{GMon}}(n)$ are similar to those in the previous example, except that they must obey the graphic identity, $xyx = xy$ for all $x, y \in \Gamma_{\mathbb{B},\text{GMon}}(n)$. What this means in the graphical interpretation is that all edges can be identified with the lowest occurring instance of an edge on the same vertex pair. This means that these networks in reduced form have at most as many edges as the complete simple graph with the same number of edges. Essentially these networks are simple graphs with a partial order on the edges which respects disjointness of edges.

The networks in the previous example have exactly what we need in a network model to realize networks of bounded degree as an algebra of a network operad.

Example 20 (**Networks of bounded degree, revisited**). The *degree* of a vertex in a simple graph is the number of edges in the graph which contain that vertex. For $k \in \mathbb{N}$, we say that a simple graph is *k-bounded* if all vertices have degree less than or equal to k. Then, we can consider the set $B_k(n)$ of k-bounded simple graphs. We can define an action of $\Gamma_{\mathbb{B},\text{GMon}}(n)$ on $B_k(n)$ in the following way. Let $g = e_1 \ldots e_l \in \Gamma_{\mathbb{B},\text{GMon}}(n)$ and $h \in B_k(n)$. Choose a graph $h' \in \Gamma_{\mathbb{B},\text{GMon}}(n)$ which has the same edges as h. Define $h_0 = h'$, then define $h_i = h_{i-1}e_i$ if that is k-bounded, else $h_i = h_{i-1}$. Let hg denote h_l, which is a k-bounded element of $\Gamma_{\mathbb{B},\text{GMon}}(n)$. Let $\Gamma^k_{\mathbb{B},\text{GMon}}(n)$ denote the set of k-bounded elements of $\Gamma_{\mathbb{B},\text{GMon}}(n)$. There is a function $s\colon \Gamma^k_{\mathbb{B},\text{GMon}}(n) \to B_k(n)$. So, we define hg to be $s(h_l)$. This is independent of the choice of h' and defines an action of $\Gamma_{\mathbb{B},\text{GMon}}$ on $B_k(n)$.

The networks in Question 4 can be represented by simple graphs with vertex degrees bounded by k. Then, $B_k(n)$ gives an algebra of the operad $\text{O}_{\mathbb{B},\text{GMon}}$.

This resolves the conflict encountered in Question 4. Ordinary network models could not record the order in which edges were added to a network, which was necessary to define a systematic way of attempting to add new connections to a network which has degree limitations on each vertex.

References

Adámek, J., Lawvere, F.W. and Rosický, J. (2003). On the duality between varieties and algebraic theories. *Algebra Universalis* **49** (1) 35–49.

Baez, J.C., Foley, J., Moeller, J. and Pollard, B.S. (2018). Network models. arXiv:1711.00037.

Burris, S.N. and Sankappanavar, H.P. (1981). *A Course in Universal Algebra*, New York, Springer-Verlag.

Fountain, J. and Kambites, M. (2009). Graph products of right cancellative monoids. *Journal of the Australian Mathematical Society* **87** (2) 227–252.

Green, E.R. (1990). *Graph Products of Groups*. Phd thesis, University of Leeds.

Hungerford, T.W. (1974). *Algebra*, Graduate Texts in Mathematics, vol. 73, New York, Springer-Verlag.

Joyal, A. and Street, R. (1993). Braided tensor categories. *Advances in Mathematics* **102** (1) 20–78.

Lawvere, F.W. (1963). *Functorial Semantics of Algebraic Theories*. Phd thesis, Columbia University.

Lawvere, F.W. (1989). Display of graphics and their applications, as exemplified by 2-categories and the hegelian "taco". In: *Proceedings of the First International Conference on Algebraic Methodology and Software Technology, University of Iowa*, 51–74.

Lovász, L. (1978). Kneser's conjecture, chromatic number, and homotopy. *Journal of Combinatorial Theory, Series A*, **25** (3) 319–324.

Mac Lane, S. (1998). *Categories for the Working Mathematician*. Berlin, Springer.

Margolis, S., Saliola, F. and Steinberg, B. (to appear). Cell complexes, poset topology and the representation theory of algebras arising in algebraic combinatorics and discrete geometry. *Memoirs of the American Mathematical Society*.

Quillen, D. (1967). *Homotopical Algebra*, Lecture Notes in Mathematics, vol. 43, Berlin, Heidelberg, Springer-Verlag.

Selinger, P. (2011). A survey of graphical languages for monoidal categories. In: Coecke, B. (ed.) *New Structures for Physics*, Berlin, Heidelberg, Springer-Verlag, 289–355.

Veloso da Costa, A. (2001). Graph products of monoids. *Semigroup Forum* **63** 247–277.