The Algebra of Open and Interconnected Systems

Brendan Fong

Hertford College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy in Computer Science

Trinity 2016

Contents

Part I

Mathematical Foundations

Chapter 1

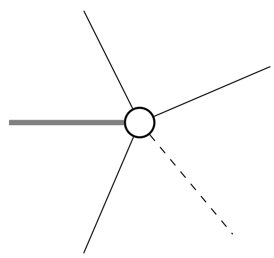
Hypergraph categories: the algebra of interconnection

In this chapter we introduce hypergraph categories, giving a definition, coherence theorem, and graphical language. We then explore a fundamental example of hypergraph categories: categories of cospans.

We assume basic familiarity with category theory and symmetric monoidal categories; although we give a sparse overview of the latter for reference. A proper introduction to both can be found in Mac Lane [?].

1.1 The algebra of interconnection

Our aim is to algebraicise network diagrams. A network diagram is built from pieces like so:



These represent open systems, concrete or abstract; for example a resistor, a chemical reaction, or a linear transformation. The essential feature, for openness and for networking, is that the system may have terminals, perhaps of different 'types', each one depicted by a line radiating from the central body. In the case of a resistor each terminal might represent a wire, for chemical reactions a chemical species, for linear transformations a variable in the domain or codomain. Network diagrams are formed by connecting terminals of systems to build larger systems.

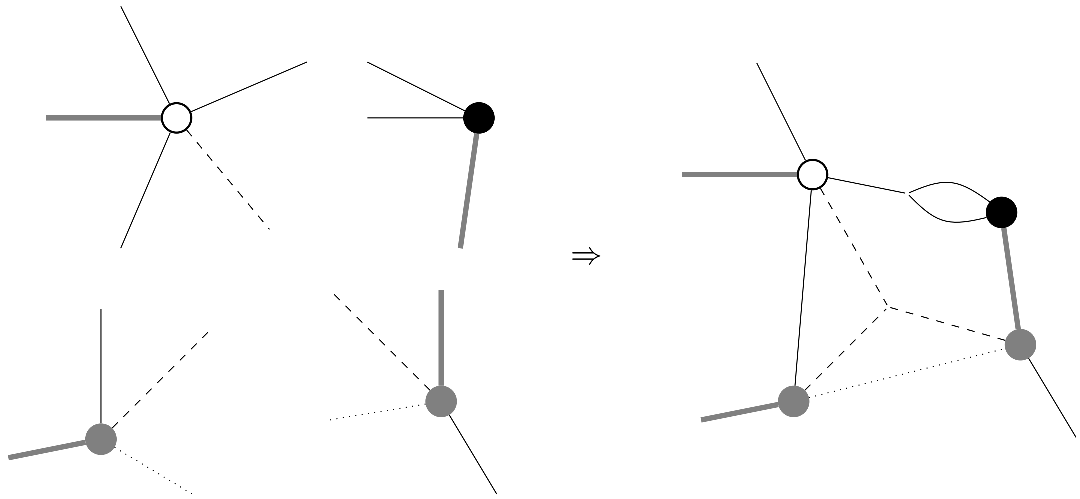


Figure 1.1: Interconnection of network diagrams. Note that we only connect terminals of the same type, but we can connect as many as we like.

A network-style diagrammatic language is a collection of network diagrams together with the stipulation that if we take some of these network diagrams, and connect terminals of the same type in any way we like, then we form another diagram in the collection. The point of this chapter is that hypergraph categories provide a precise formalisation of network-style diagrammatic languages.

In jargon, a hypergraph category is a symmetric monoidal category in which every object is equipped with a special commutative Frobenius monoid in a way compatible with the monoidal product. We will walk through these terms in detail, illustrating them with examples and a few theorems.

The key data comprising a hypergraph category are its objects, morphisms, composition rule, monoidal product, and Frobenius maps. Each of these model a feature of network diagrams and their interconnection. The objects model the terminal types, while the morphisms model the network diagrams themselves. The composition, monoidal product, and Frobenius maps model different aspects of interconnection: composition models the interconnection of two terminals of the same type, the monoidal product models the network formed by taking two networks without interconnecting any terminals, while the Frobenius maps model multi-terminal interconnection.

These Frobenius maps are the distinguishing feature of hypergraph categories as compared to other structured monoidal categories, and are crucial for formalising the intuitive concept of network languages detailed above. In the case of electric circuits

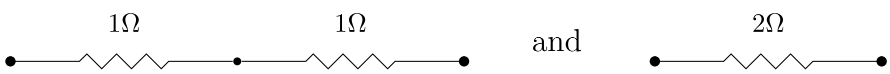
Networks	Hypergraph categories
list of terminal types	object
network diagram	morphism
series connection	composition
juxtaposition	monoidal product
branching	Frobenius maps

Figure 1.2: Corresponding features of networks and hypergraph categories.

the Frobenius maps model the 'branching' of wires; in the case when diagrams simply model an abstract system of equations and terminals variables in these equations, the Frobenius maps allow variables to be shared between many systems of equations.

Examining these correspondences, it is worthwhile to ask whether hypergraph categories permit too much structure to be specified, given that the interconnection rule is now divided into three different aspects, and features such as domains and codomains of network diagrams, rather than just a collection of terminals, exist. The answer is given by examining the additional coherence laws that these data must obey. For example, in the case of the domain and codomain, we shall see that hypergraph categories are all compact closed categories, and so there is ultimately only a formal distinction between domain and codomain objects. One way to think of these data is as scaffolding. We could compare it to the use of matrices and bases to provide language for talking about linear transformations and vector spaces. They are not part of the target structure, but nonetheless useful paraphenalia for constructing it.

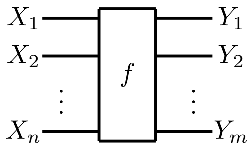
Network languages are not only syntactic entities: as befitting the descriptor 'language', they typically have some associated semantics. Circuits diagrams, for instance, not only depict wire circuits that may be constructed, they also represent the electrical behaviour of that circuit. Such semantics considers the circuits



the same, even though as 'syntactic' diagrams they are distinct. A cornerstone of the utility of the hypergraph formalism is the ability to also realise the semantics of these diagrams as morphisms of another hypergraph category. This 'semantic' hypergraph category, as a hypergraph category, still permits the rich 'networking' interconnection structure, and a so-called hypergraph functor implies that the syntactic category provides a sound framework for depicting these morphisms. Network languages syntactically are often 'free' hypergraph categories, and much of the interesting structure lies in their functors to their semantic hypergraph categories.

1.2 Symmetric monoidal categories

Suppose we have some tiles with inputs and outputs of various types like so:



These tiles may vary in height and width. We can place these tiles above and below each other, and to the left and right, so long as the inputs on the right tile match the outputs on the left. Suppose also that some arrangements of tiles are equal to other arrangements of tiles. How do we formalise this structure algebraically? The theory of monoidal categories provides an answer.

Hypergraph categories are first monoidal categories, indeed symmetric monoidal categories. A monoidal category is a category with two notions of composition: ordinary categorical composition and monoidal composition, with the monoidal composition only associative and unital up to natural isomorphism. They are the algebra of processes that may occur simultaneously as well as sequentially. First defined by Bénabou and Mac Lane in the 1960s [**?**, **?**], their theory and their links with graphical representation are well explored. We bootstrap on this, using monoidal categories to define hypergraph categories, and so immediately arriving at an understanding of how hypergraph categories formalise our network languages.

Moreover, symmetric monoidal functors play a key role in our framework for defining and working with hypergraph categories: decorated cospans and corelations constructions. For this reason we provide, for quick reference, a definition of symmetric monoidal categories.

1.2.1 Monoidal categories

A **monoidal category** $(\mathcal{C}, \otimes)$ consists of a category $\mathcal{C}$, a functor $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$, a distinguished object I, and natural isomorphisms $\alpha_{A,B,C} : (A \otimes B) \otimes C \to A \otimes (B \otimes C)$, $\rho_A : A \otimes I \to A$, and $\lambda_A : I \otimes A \to A$ such that for all A, B, C, D in $\mathcal{C}$ the following two diagrams commute:

$$\begin{tikzcd}
\big((A\otimes B)\otimes C\big)\otimes D \arrow[rr, "\alpha_{(A\otimes B),C,D}"] \arrow[d, "\alpha_{A,B,C}\otimes \mathrm{id}_D"'] & & (A\otimes B)\otimes (C\otimes D) \arrow[d, "\alpha_{A,B,(C\otimes D)}"] \\
\big(A\otimes (B\otimes C)\big)\otimes D \arrow[r, "\alpha_{A,(B\otimes C),D}"'] & A\otimes\big((B\otimes C)\otimes D\big) \arrow[r, "\mathrm{id}_A\otimes \alpha_{B,C,D}"'] & A\otimes\big(B\otimes (C\otimes D)\big)
\end{tikzcd}$$

$$\begin{tikzcd}
(A\otimes I)\otimes B \arrow[rr, "\alpha_{A,I,B}"] \arrow[dr, "\rho_A\otimes \mathrm{id}_B"'] & & A\otimes (I\otimes B) \arrow[dl, "\mathrm{id}_A\otimes \lambda_B"] \\
& A\otimes B &
\end{tikzcd}$$

We call $\otimes$ the **monoidal product**, I the **monoidal unit**, α the **associator**, ρ and λ the **right** and **left unitor** respectively. The associator and unitors are known collectively as the **coherence maps**.

By Mac Lane's coherence theorem, these two axioms are equivalent to requiring that 'all formal diagrams'—that is, all diagrams in which the morphism are built from identity morphisms and the coherence maps using composition and the monoidal product—commute. Consequently, between any two products of the same ordered list of objects up to instances of the monoidal unit, such as $((A\otimes I)\otimes B)\otimes C$ and $A\otimes((B\otimes C)\otimes(I\otimes I))$, there is a unique so-called **canonical** map. See Mac Lane [**?**, Corollary of Theorem VII.2.1] for a precise statement and proof.

A **lax monoidal functor** $(F,\varphi):(\mathcal{C},\otimes)\to(\mathcal{C}',\boxtimes)$ between monoidal categories consists of a functor $F:\mathcal{C}\to\mathcal{C}'$, and natural transformations $\varphi_{A,B}: FA\boxtimes FB\to F(A\otimes B)$ and $\varphi_1: 1_{\mathcal{C}'}\to F1_{\mathcal{C}}$, such that for all $A,B,C\in\mathcal{C}$ the three diagrams

$$\begin{tikzcd}
(FA\otimes FB)\otimes FC \arrow[r, "\varphi_{A,B}\otimes \mathrm{id}_{FC}"] \arrow[d, "\alpha_{FA,FB,FC}"'] & F(A\otimes B)\otimes FC \arrow[r, "\varphi_{A\otimes B,C}"] & F((A\otimes B)\otimes C) \arrow[d, "F\alpha_{A,B,C}"] \\
FA\otimes (FB\otimes FC) \arrow[r, "\mathrm{id}_{FA}\otimes \varphi_{B,C}"'] & FA\otimes F(B\otimes C) \arrow[r, "\varphi_{A,B\otimes C}"'] & F(A\otimes (B\otimes C))
\end{tikzcd}$$

$$\begin{tikzcd}
F(A)\otimes I' \arrow[r, "\rho"] \arrow[d, "\mathrm{id}\otimes\varphi_1"'] & F(A) \\
F(A)\otimes F(I) \arrow[r, "\varphi_{A,I}"'] & F(A\otimes I) \arrow[u, "F\rho"']
\end{tikzcd}
\qquad
\begin{tikzcd}
I'\otimes F(A) \arrow[r, "\lambda"] \arrow[d, "\varphi_1\otimes\mathrm{id}"'] & F(A) \\
F(I)\otimes F(A) \arrow[r, "\varphi_{I,A}"'] & F(I\otimes A) \arrow[u, "F\lambda"']
\end{tikzcd}$$

commute. We further say a monoidal functor is a **strong monoidal functor** if the φ are isomorphisms, and a **strict monoidal functor** if the φ are identities.

A **monoidal natural transformation** $\theta : (F,\varphi) \Rightarrow (G,\gamma)$ between two monoidal functors F and G is a natural transformation $\theta : F \Rightarrow G$ such that

$$\begin{tikzcd} F1_{\mathcal{C}} \arrow[rr, "\theta_I"] & & G1_{\mathcal{C}} \\ & 1_{\mathcal{C}'} \arrow[ul, "\varphi_1"] \arrow[ur, "\gamma_1"'] & \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} FA \boxtimes FB \arrow[r, "\theta_A \otimes \theta_B"] \arrow[d, "\varphi_{A,B}"'] & GA \boxtimes GB \arrow[d, "\gamma_{A,B}"] \\ F(A \otimes B) \arrow[r, "\theta_{A\otimes B}"'] & G(A \otimes B) \end{tikzcd}$$

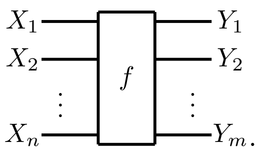
commute for all objects A, B.

Two monoidal categories $\mathcal{C}, \mathcal{D}$ are **monoidally equivalent** if there exist strong monoidal functors $F \colon \mathcal{C} \to \mathcal{D}$ and $G \colon \mathcal{D} \to \mathcal{C}$ such that the composites FG and GF are monoidally naturally isomorphic to the identity functors. (Note that identity functors are immediately strict monoidal functors.)

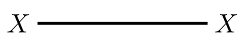
1.2.2 String diagrams

A **strict monoidal category** category is a monoidal category in which the associators and unitors are all identity maps. In this case then any two objects that can be related by associators and unitors are equal, and so we may write objects without parentheses and units without ambiguity. An equivalent statement of Mac Lane's coherence theorem is that every monoidal category is monoidally equivalent to strict monoidal category.

Yet another equivalent statement of the coherence theorem is the existence of a graphical calculus for monoidal categories. As discussed above, monoidal categories figure strongly in our current investigations precisely because of this. We leave the details to discussions elsewhere. The main point is that we shall be free to assume our monoidal categories are strict, writing $X_1 \otimes X_2 \otimes \cdots \otimes X_n$ for objects in $(\mathcal{C}, \otimes)$ without a care for parentheses. We then depict a morphism $f \colon X_1 \otimes X_2 \otimes \cdots \otimes X_n \to Y_1 \otimes Y_2 \otimes \cdots \otimes Y_n$ with the diagram:

$f \quad =$ 

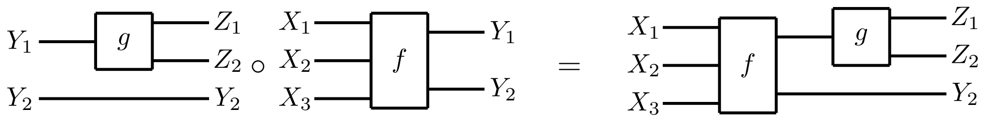
Identity morphisms are depicted by 'wires':

$\mathrm{id}_X \quad =$ 

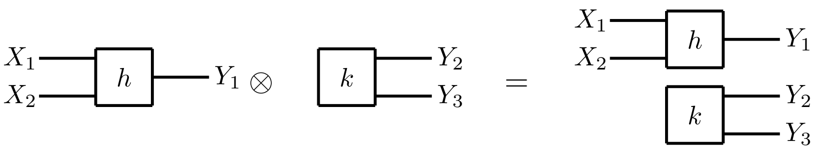
and the monoidal unit is not depicted at all:

$\mathrm{id}_I \quad =$

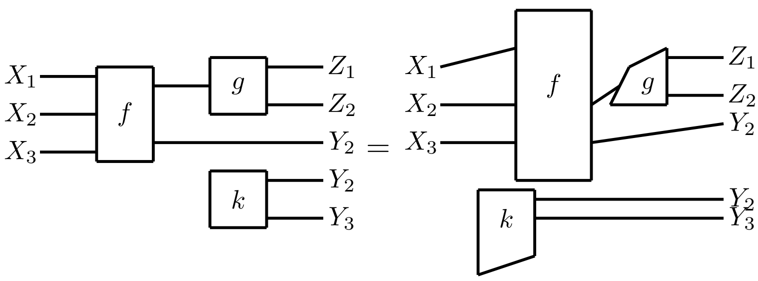
Composition of morphisms is depicted by connecting the relevant 'wires':



while monoidal composition is just juxtaposition:



Only the 'topology' of the diagrams matters: if two diagrams are homotopy equivalent, they represent the same algebraic expression. On the other hand, two algebraic expressions might have the same diagrammatic representation. For example, the equivalent diagrams



read as all of the equivalent algebraic expressions

$$((g \otimes \mathrm{id}_{Y_2}) \otimes k) \circ f = (g \otimes ((\mathrm{id}_{Y_2} \otimes k)) \circ \rho \circ (f \otimes \mathrm{id}_I) = (g \otimes \mathrm{id}_{Y_2}) \circ f \circ (\mathrm{id}_{X_1 \otimes (X_2 \otimes X_3)} \otimes k)$$

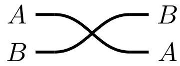
and so on. The coherence theorem says that this does not matter: if two algebraic expressions have the same diagrammatic representation, then the algebraic expressions are equal. In more formal language, the graphical calculus is sound and complete for the axioms of monoidal categories.

The coherence theorem thus implies that the graphical calculi goes beyond visualisations of morphisms: it can provide provide bona-fide proofs of equalities of morphisms. As a general principle, string diagrams are more intuitive than the conventional algebraic language for understanding monoidal categories.

1.2.3 Symmetry

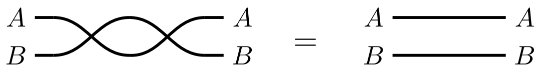
A symmetric braiding in a monoidal category provides the ability to permute objects or, equivalently, cross wires. We define symmetric monoidal categories making use of the graphical notation outlined above, but introducing a new, special symbol $\times$.

A **symmetric monoidal category** is a monoidal category $(\mathcal{C}, \otimes)$ together with natural isomorphisms

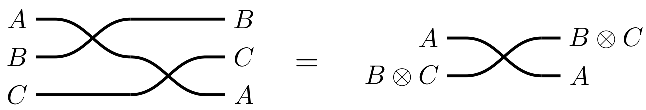


$$\sigma_{A,B} : A \otimes B \to B \otimes A$$

such that



and



for all A, B, C in $\mathcal{C}$. We call σ the **braiding**.

A **(lax/strong) symmetric monoidal functor** is a (lax/strong) monoidal functor that further obeys

$$\begin{tikzcd}
FA \otimes FB \arrow[r, "\varphi_{A,B}"] \arrow[d, "\sigma'_{FA,FB}"'] & F(A \otimes B) \arrow[d, "F\sigma_{A,B}"] \\
FB \otimes FA \arrow[r, "\varphi_{B,A}"'] & F(B \otimes A)
\end{tikzcd}$$

Morphisms between symmetric monoidal functors are simply monoidal natural transformations. Thus two symmetric monoidal categories are **symmetric monoidally equivalent** if they are monoidally equivalent by strong *symmetric* monoidal functors.

The coherence theorem for symmetric monoidal categories, with respect to string diagrams, states that two morphisms in a symmetric monoidal category are equal according to the axioms of symmetric monoidal categories if and only if their diagrams are equal up to homotopy equivalence and applications of the defining graphical identities above. See Joyal–Street [**?**, Theorem 2.3] for more precision and details.

1.3 Hypergraph categories

Just as symmetric monoidal categories equip monoidal categories with precisely enough extra structure to model crossing of strings in the graphical calculus, hypergraph categories equip symmetric monoidal categories with precisely enough extra structure to model multi-input multi-output interconnections of strings of the same type. For this, we require each object to be equipped with a so-called special commutative Frobenius monoid, which provides chosen maps to model this interaction. These have a coherence result, known as the 'spider theorem', that says exactly how we use the maps to describe the connection of strings does not matter: all that matters is that the strings are connected.

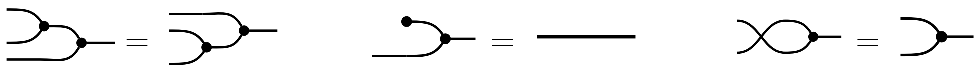
1.3.1 Frobenius monoids

A Frobenius monoid comprises a monoid and comonoid on the same object that interact according to the so-called Frobenius law.

Definition 1.1. A **special commutative Frobenius monoid** $(X, \mu, \eta, \delta, \epsilon)$ in a symmetric monoidal category $(\mathcal{C}, \otimes)$ is an object X of $\mathcal{C}$ together with maps

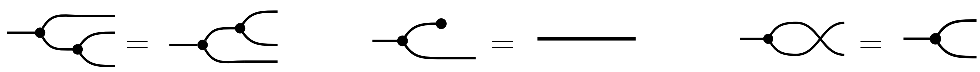
$\mu\colon X \otimes X \to X$ $\quad$ $\eta\colon I \to X$ $\quad$ $\delta\colon X \to X \otimes X$ $\quad$ $\epsilon\colon X \to I$

obeying the commutative monoid axioms



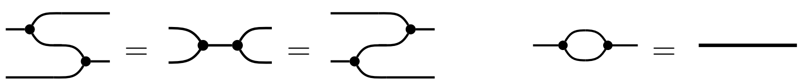
(associativity) (unitality) (commutativity)

the cocommutative comonoid axioms



(coassociativity) (counitality) (cocommutativity)

and the Frobenius and special axioms



(Frobenius) (special)

We call μ the **multiplication**, η the **unit**, δ the **comultiplication**, and ϵ the **counit**.

Special commutative Frobenius monoids were first formulated by Carboni and Walters, under the name commutative separable algebras. The Frobenius law and the special law were termed the S=X law and the diamond=1 law respectively [**?**, **?**].

Alternate axiomatisations are possible. In addition to the 'upper' unitality law above, the mirror image 'lower' unitality law also holds, due to commutativity and the naturality of the braiding. While we write two equations for the Frobenius law, this is redundant: given the other axioms, the equality of any two of the diagrams implies the equality of all three. Further, note that a monoid and comonoid obeying the Frobenius law is commutative if and only if it is cocommutative. Thus while a commutative and cocommutative Frobenius monoid might more properly be called a bicommutative Frobenius monoid, there is no ambiguity if we only say commutative.

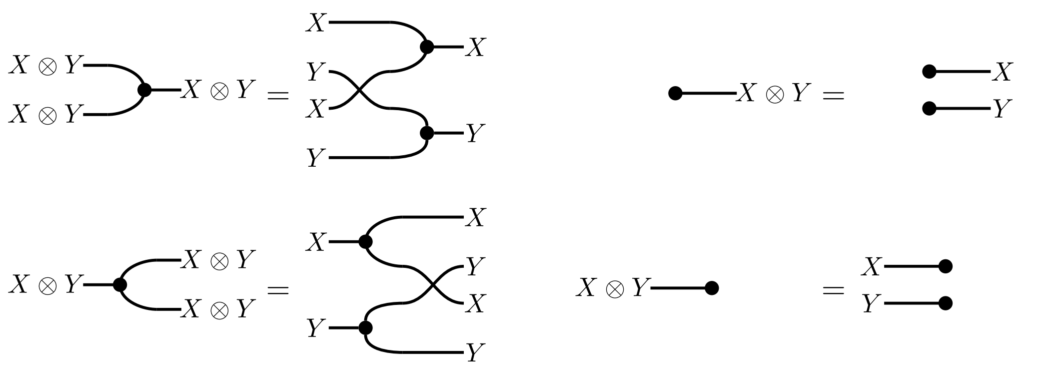
The common feature to these equations is that each side describes a different way of using the generators to connect some chosen set of inputs to some chosen set of outputs. This observation provides a 'coherence' type result for special commutative Frobenius monoids, known as the 'spider theorem'.

Theorem 1.2. *Let $(X, \mu, \eta, \delta, \epsilon)$ be a special commutative Frobenius monoid, and let $f, g\colon X^{\otimes n} \to X^{\otimes m}$ be map constructed, using composition and the monoidal product, from μ, η, δ, ϵ, the coherence maps and braiding, and the identity map on X. Then f and g are equal if and only if given their string diagrams in the above notation, there exists a bijection between the connected components of the two diagrams such that corresponding connected components connect the exact same sets of inputs and outputs.*

See [?, ?] for further details.

1.3.2 Hypergraph categories

Definition 1.3. A **hypergraph category** is a symmetric monoidal category in which each object X is equipped with a special commutative Frobenius structure $(X, \mu_X, \delta_X, \eta_X, \epsilon_X)$ such that



Note that we do *not* require these Frobenius morphisms to be natural in X. While morphisms in a hypergraph category need not interact with the Frobenius structure in any particular way, we do require functors between hypergraph categories to preserve it.

Definition 1.4. A functor (F, φ) of hypergraph categories, or **hypergraph functor**, is a strong symmetric monoidal functor (F, φ) such that for each object X the following diagrams commute:

$$\begin{tikzcd} FX \boxtimes FX \arrow[rr, "\mu_{FX}"] \arrow[dr, "\varphi"'] & & FX \\ & F(X \otimes X) \arrow[ur, "F\mu_X"'] & \end{tikzcd} \qquad \begin{tikzcd} 1_{\mathcal{D}} \arrow[rr, "\eta_{FX}"] \arrow[dr, "\varphi_1"'] & & FX \\ & F1_{\mathcal{C}} \arrow[ur, "F\eta_X"'] & \end{tikzcd}$$

$$\begin{tikzcd} FX \arrow[rr, "\delta_{FX}"] \arrow[dr, "F\delta_X"'] & & FX \boxtimes FX \\ & F(X \otimes X) \arrow[ur, "\varphi^{-1}"'] & \end{tikzcd} \qquad \begin{tikzcd} FX \arrow[rr, "\epsilon_{FX}"] \arrow[dr, "F\epsilon_X"'] & & 1_{\mathcal{D}} \\ & F1_{\mathcal{C}} \arrow[ur, "\varphi^{-1}"'] & \end{tikzcd}$$

Equivalently, a strong symmetric monoidal functor F is a hypergraph functor if for every X the special commutative Frobenius structure on FX is

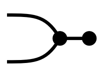
$$(FX,\ \ F\mu_X \circ \varphi_{X,X},\ \ \varphi^{-1}_{X,X} \circ F\delta_X,\ \ F\eta_X \circ \varphi_1,\ \ \varphi_1^{-1} \circ F\epsilon_X).$$

Just as monoidal natural transformations themselves are enough as morphisms between symmetric monoidal functors, so too they suffice as morphisms between hypergraph functors. Two hypergraph categories are **hypergraph equivalent** if there exist hypergraph functors with monoidal natural transformations to the identity functors.

The term hypergraph category was introduced recently [**?**, **?**], in reference to the fact that these special commutative Frobenius monoids provide precisely the structure required to draw graphs with 'hyperedges': edges connecting any number of inputs to any number of outputs. Again first defined by Walters and Carboni [**?**], under the name well-supported compact closed categories, in recent years hypergraph categories have been rediscovered a number of times, also appearing under the names dungeon categories [**?**] and dgs-monoidal categories [**?**].

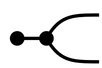
1.3.3 Hypergraph categories are self-dual compact closed

Note that if an object X is equipped with a Frobenius monoid structure then the maps



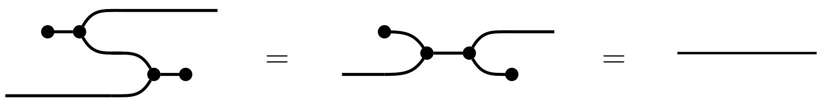
$\epsilon \circ \mu \colon X \otimes X \to 1$

and



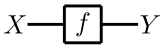
$\delta \circ \eta \colon 1 \to X \otimes X$

obey both

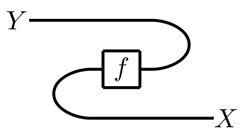


and the reflected equations. Thus if an object carries a Frobenius monoid it is also self-dual, and any hypergraph category is a fortiori self-dual compact closed.

We introduce the notation

As in any self-dual compact closed category, mapping each morphism 

to its dual morphism



further equips each hypergraph category with a so-called dagger functor—an involutive contravariant endofunctor that is the identity on objects—such that the category is a dagger compact category. Dagger compact categories were first introduced in the context of categorical quantum mechanics [**?**], under the name strongly compact closed category, and have been demonstrated to be a key structure in diagrammatic reasoning and the logic of quantum mechanics.

1.3.4 Coherence

The lack of naturality of the Frobenius maps in hypergraph categories affects some common properties of structured categories. For example, it is not always possible to construct a skeletal hypergraph category hypergraph equivalent to a given hypergraph category: isomorphic objects may be equipped with 'different' Frobenius monoids. Similarly, a fully faithful, essentially surjective hypergraph functor does not necessarily define a hypergraph equivalence of categories.

Nonetheless, in this section we prove that every hypergraph category is hypergraph equivalent to a strict hypergraph category. This coherence result will be important in proving that every hypergraph category can be constructed using decorated corelations.

Theorem 1.5. *Every hypergraph category is hypergraph equivalent to a strict hypergraph category. Moreover, the objects of this strict hypergraph category form a free monoid.*

Proof. Let $(\mathcal{H}, \otimes)$ be a hypergraph category. As $\mathcal{H}$ is a fortiori a symmetric monoidal category, a standard construction (see Mac Lane [?, Theorem]) gives an equivalent objectwise-free strict symmetric monoidal category $(\mathcal{H}_{\mathrm{str}}, \cdot)$ with objects finite lists $[x_1, \dots, x_n]$ of objects of $\mathcal{H}$ and morphisms $[x_1, \dots, x_n] \to [y_1, \dots, y_m]$ those morphisms from $((x_1 \otimes x_2) \otimes \dots) \otimes x_n \to ((y_1 \otimes y_2) \otimes \dots) \otimes y_m$ in $\mathcal{H}$. Composition is given by composition in $\mathcal{H}$.

The monoidal structure is given as follows. Given a list X of objects in $\mathcal{H}$, write PX for the corresponding monoidal product in $\mathcal{H}$ with all open parathesis at the front. The monoidal product of two objects is given by concatenation $\cdot$ of lists; the monoidal unit is the empty list. The monoidal product of two morphisms is given by their monoidal product in $\mathcal{H}$ pre- and post-composed with the necessary canonical maps: given $f\colon X \to Y$ and $g\colon Z \to W$, their product $f \cdot g\colon X \cdot Y \to Z \cdot W$ is

$$P(X \cdot Y) \longrightarrow PX \otimes PY \xrightarrow{f \otimes g} PZ \otimes PW \longrightarrow P(Z \cdot W).$$

By design, the associators and unitors are simply identity maps. The braiding $X \cdot Y \to Y \cdot X$ is given by the braiding $PX \otimes PY \to PY \otimes PX$ in $\mathcal{H}$, similarly pre- and post-composed with the necessary canonical maps. This defines a strict symmetric monoidal category [?].

To make $\mathcal{H}_{\mathrm{str}}$ into a hypergraph category, we equip each object $[x_1, \dots, x_n]$ with a special commutative Frobenius monoid in a similar way. For example, the multiplication on $[x_1, \dots, x_n]$ is given by

$$\begin{array}{c}
P([x_1, \dots, x_n] \cdot [x_1, \dots, x_n]) = ((((((x_1 \otimes x_2) \otimes \dots) \otimes x_n) \otimes x_1) \otimes x_2) \otimes \dots) \otimes x_n \\
\big\downarrow \\
(((x_1 \otimes x_1) \otimes (x_2 \otimes x_2)) \otimes \dots) \otimes (x_n \otimes x_n) \\
\big\downarrow {\scriptstyle ((\mu_{x_1} \otimes \mu_{x_2}) \otimes \dots) \otimes \mu_{x_n}} \\
P([x_1, \dots, x_n]) = ((x_1 \otimes x_2) \otimes \dots) \otimes x_n
\end{array}$$

where the first map is the canonical map such that each pair of x_i's remains in the same order. It is straightforward to check that this defines a hypergraph category.

The strict hypergraph category $(\mathcal{H}_{\mathrm{str}}, \cdot)$	
objects	finite lists $[x_1, \dots, x_n]$ of objects of $\mathcal{H}$
morphisms	$\hom_{\mathcal{H}_{\mathrm{str}}}\big([x_1, \dots, x_n], [y_1, \dots, y_m]\big)$ $= \hom_{\mathcal{H}}\big(((x_1 \otimes x_2) \otimes \dots) \otimes x_n, ((y_1 \otimes y_2) \otimes \dots) \otimes y_m\big)$
composition	composition of corresponding maps in $\mathcal{H}$
monoidal product	concatenation of lists
coherence maps	associators and unitors are strict; braiding is inherited from $\mathcal{H}$
hypergraph maps	lists of hypergraph maps in $\mathcal{H}$

Our standard construction further gives strong symmetric monoidal functors $P\colon \mathcal{H}_{\mathrm{str}} \to \mathcal{H}$ extending the map P above, and $S\colon \mathcal{H} \to \mathcal{H}_{\mathrm{str}}$ sending $x \in \mathcal{H}$ to the string $[x]$ of length 1 in $\mathcal{H}_{\mathrm{str}}$. These extend to hypergraph functors.

In detail, the functor P is given on morphisms by taking a map in $\hom_{\mathcal{H}_{\mathrm{str}}}(X, Y)$ to the same map considered now as a map in $\hom_{\mathcal{H}}(PX, PY)$; its coherence maps are given by the canonical maps $PX \otimes PY \to P(X \cdot Y)$. The functor S is even easier to define: a morphism $x \to y$ in $\mathcal{H}$ is by definition a morphism $[x] \to [y]$ in $\mathcal{H}_{\mathrm{str}}$, so S is a monoidal embedding of $\mathcal{H}$ into $\mathcal{H}_{\mathrm{str}}$.

By Mac Lane's proof of the coherence theorem for monoidal categories these are both strong monoidal functors; by inspection they also preserve hypergraph structure, and so are hypergraph functors. As they already witness an equivalence of symmetric monoidal categories, thus $\mathcal{H}$ and $\mathcal{H}_{\mathrm{str}}$ are equivalent as hypergraph categories. $\square$

1.4 Example: cospan categories

A central example of a hypergraph category is the category $\mathrm{Cospan}(\mathcal{C})$ of cospans in any category $\mathcal{C}$ with finite colimits. We will later see that decorated cospan categories are a generalisation of such categories, and each inherits a hypergraph structure from such.

We first recall the basic definitions. Let $\mathcal{C}$ be a category with finite colimits. A **cospan** $X \xrightarrow{i} N \xleftarrow{o} Y$ from X to Y in $\mathcal{C}$ is a pair of morphisms with common codomain. We refer to X and Y as the **feet**, and N as the **apex**. Given two cospans $X \xrightarrow{i} N \xleftarrow{o} Y$ and $X \xrightarrow{i'} N' \xleftarrow{o'} Y$ with the same feet, a **map of cospans** is a morphism $n: N \to N'$ in $\mathcal{C}$ between the apices such that

$$\begin{tikzcd} & N \arrow[dd, "n"] & \\ X \arrow[ur, "i"] \arrow[dr, "i'"'] & & Y \arrow[ul, "o"'] \arrow[dl, "o'"] \\ & N' & \end{tikzcd}$$

commutes.

Cospans may be composed using the pushout from the common foot: given cospans $X \xrightarrow{i_X} N \xleftarrow{o_Y} Y$ and $Y \xrightarrow{i_Y} M \xleftarrow{o_Z} Z$, their composite cospan is $X \xrightarrow{j_N \circ i_X} N +_Y M \xleftarrow{j_M \circ i_Z} Z$, where

$$\begin{tikzcd} & & N +_Y M & & \\ & N \arrow[ur, "j_N"] & & M \arrow[ul, "j_M"'] & \\ X \arrow[ur, "i_X"] & & Y \arrow[ul, "o_Y"'] \arrow[ur, "i_Y"] & & Z \arrow[ul, "o_Z"'] \end{tikzcd}$$

is a pushout square. This composition rule is associative up to isomorphism, and so we may define a category, in fact a symmetric monoidal bicategory, Cospan($\mathcal{C}$) with objects the objects of $\mathcal{C}$ and morphisms isomorphism classes of cospans [**?**].

Write FinSet for the category of finite sets and functions. It is well-known, due to Lack [**?**], that special commutative Frobenius monoids in a monoidal category $\mathcal{C}$ are in one-to-one correspondence with monoidal functors Cospan(FinSet) $\to \mathcal{C}$. In the next few chapters we will further explore this deep link between cospans and special commutative Frobenius monoids and hypergraph categories.

To begin, we detail a natural hypergraph structure on Cospan($\mathcal{C}$).

Given maps $f: A \to C$, $g: B \to C$ with common codomain, the universal property of the coproduct gives a unique map : $A+B \to C$. We call this the **copairing** of f and g, and write it $[f, g]$. We consider any category $\mathcal{C}$ with finite colimits a symmetric monoidal category with monoidal product given by the coproduct, written $+$, and braiding given by the maps $A + B \to B + A$ by copairing identity maps.

Now Cospan($\mathcal{C}$) inherits a symmetric monoidal structure from $\mathcal{C}$. We call a subcategory $\mathcal{C}$ of a category $\mathcal{D}$ **wide** if $\mathcal{C}$ contains all objects of $\mathcal{D}$, and call a functor that is faithful and bijective-on-objects a **wide embedding**. Note then that we have a wide embedding

$$\mathcal{C} \hookrightarrow \mathrm{Cospan}(\mathcal{C})$$

that takes each object of $\mathcal{C}$ to itself as an object of Cospan($\mathcal{C}$), and each morphism $f\colon X \to Y$ in $\mathcal{C}$ to the cospan

$$\begin{tikzcd} & Y \arrow[dr, equal] & \\ X \arrow[ur, "f"] & & Y, \end{tikzcd}$$

where the extended 'equals' sign denotes an identity morphism. This allows us to view $\mathcal{C}$ as a wide subcategory of Cospan($\mathcal{C}$).

As $\mathcal{C}$ has finite colimits, it can be given a symmetric monoidal structure with the coproduct the monoidal product; we write this monoidal category $(\mathcal{C}, +)$, and write $\varnothing$ for the initial object, the monoidal unit of this category.

Then Cospan($\mathcal{C}$) inherits the same symmetric monoidal structure: since the monoidal product $+\colon \mathcal{C}\times\mathcal{C} \to \mathcal{C}$ is left adjoint to the diagram functor, it preserves colimits, and so extends to a functor $+\colon \mathrm{Cospan}(\mathcal{C}) \times \mathrm{Cospan}(\mathcal{C}) \to \mathrm{Cospan}(\mathcal{C})$. The remainder of the monoidal structure is inherited because $\mathcal{C}$ is a wide subcategory of Cospan($\mathcal{C}$).

Next, the Frobenius structure comes from copairings of identity morphisms. We call cospans

$$\begin{tikzcd} & N & \\ X \arrow[ur, "i"] & & Y \arrow[ul, "o"'] \end{tikzcd} \qquad \text{and} \qquad \begin{tikzcd} & N & \\ Y \arrow[ur, "o"] & & X \arrow[ul, "i"'] \end{tikzcd}$$

that are reflections of each other **opposite** cospans. Given any object X in $\mathcal{C}$, the copairing $[1_X, 1_X]\colon X + X \to X$ of two identity maps on X, together with the unique map $!\colon \varnothing \to X$ from the initial object to X, define a monoid structure on X. Considering these maps as morphisms in Cospan($\mathcal{C}$), we may take them together with their opposites to give a special commutative Frobenius structure on X. In this way we consider each category Cospan($\mathcal{C}$) a hypergraph category.

Given $f\colon X \to Y$ in $\mathcal{C}$, we also abuse notation by writing $f \in \mathrm{Cospan}(\mathcal{C})$ for the cospan $X \xrightarrow{f} Y \xleftarrow{1_Y} Y$, and f^{opp} for the cospan $Y \xrightarrow{1_Y} Y \xleftarrow{f} X$.

Definition 1.6. Let $\mathcal{C}$ be a category with finite colimits. We define the hypergraph category Cospan($\mathcal{C}$) to comprise:

The hypergraph category $(\mathrm{Cospan}(\mathcal{C}), +)$	
objects	the objects of $\mathcal{C}$
morphisms	isomorphism classes of cospans in $\mathcal{C}$
composition	given by pushout
tensor product	the coproduct in $\mathcal{C}$.
coherence maps	inherited from $(\mathcal{C}, +)$ $\sigma_{X,Y} = [\iota_Y, \iota_X]\colon X + Y \to Y + X$
hypergraph maps	$\mu_X = [1_X, 1_X]$, $\eta_X = !$, $\delta_X = \mu_X^{\mathrm{opp}}$, $\epsilon_X = \eta_X^{\mathrm{opp}}$.

We will often abuse our terminology and refer to cospans themselves as morphisms in some cospan category $\mathrm{Cospan}(\mathcal{C})$; we of course refer instead to the isomorphism class of the said cospan.

FinCocompleteCat faithfully embeds into HyperCat. ie any monoidal category of cospans has a hypergraph structure inherited from the identity morphisms.

Hypergraph categories are closely related to cospans. The free hypergraph category on a single object in the category of cospans in the category of finite sets. SpivakVagner?

Walters: cospan graph is the generic special commutative Frobenius monoid.

Later, also Vagner Spivak Schultz: hypergraph categories are algebras of cospan.

Part II

Applications

Bibliography

[Fon15] Brendan Fong. Decorated Cospans. *Theory and Applications of Categories*, 30(33):25, August 2015. arXiv: 1502.00872. (Referred to on page 17.)