Erkenntnis
https://doi.org/10.1007/s10670-019-00211-5

ORIGINAL RESEARCH

A Generic Figures Reconstruction of Peirce's Existential Graphs (Alpha)

Rocco Gangle[1] · **Gianluca Caterina**[1] · **Fernando Tohme**[2]

Received: 15 May 2018 / Accepted: 21 December 2019
© Springer Nature B.V. 2020

Abstract

We present a category-theoretical analysis, based on the concept of *generic figures*, of a diagrammatic system for propositional logic (Peirce's Existential Graphs α). The straightforward construction of a presheaf category $\mathcal{EG}_{\alpha^*}$ of cuts-only Existential Graphs (equivalent to the well-studied category of *finite forests*) provides a basis for the further construction of the category $\mathcal{EG}_{\alpha}$ which introduces variables in a reconstructedly generic, or label-free, mode. Morphisms in these categories represent syntactical embeddings or, equivalently but dually, extensions. Through the example of Peirce's system, it is shown how the generic figures approach facilitates the formal investigation of relations between syntax and semantics in such diagrammatic systems.

Introduction

A good diagram represents information by visually modeling relevant features of that information after its own fashion. Consider how a pie chart captures the relative distribution of some quantity into disjoint classes by dividing a disc pictorially into easily compared non-overlapping wedges. Similarly, a perspicuous logical notation organizes relations among the components of its syntax in such a way that relevant logical properties and relations under some intended interpretation are thereby readily apprehended and meaningfully manipulated according to simple and unambiguous protocols. When a diagram or a notational system directly instantiates or *presents* whatever properties or relations it also signifies or *represents*, and furthermore rep-

✉ Rocco Gangle
rgangle@endicott.edu

Gianluca Caterina
gianlucaterina@gmail.com

Fernando Tohme
ftohme@gmail.com

[1] Endicott College, Beverly, MA, USA

[2] Universidad Nacional del Sur, Buhía Blanca, Argentina

resents those properties or relations *by* instantiating or presenting them, it may be said to be *iconic*. The elaboration of how exactly such iconic representation works, especially in the case of iconic logical notation, may be greatly facilitated by the use of mathematical constructions within category theory. We aim to show how and why this is the case by means of a detailed application to a particular instance: C.S. Peirce's Existential Graphs.

The iconic nature of Peirce's logical system of Existential Graphs has been the object of extensive research over the last few decades [see Caterina and Gangle (2016), Pietarinen (2012), Pietarinen and Bellucci (2016), Ma and Pietarinen (2017), Stjernfelt (2007), Roberts (1973) and Zeman (1974), among others]. Scholars across multiple disciplines have focused on various aspects of his diagrammatic system, and Peirce's original approach to logic continues to inspire new research. Peirce himself introduced the term "diagrammatic syntax" to describe the distinctive character of his formal notation (MS 514 in Peirce and Eisele (1976)) since the diagrammatic or iconic nature of the syntactical elements of the notation (for instance, the use of a closed curve in the plane to distinguish the arguments of a commutative operator) and, more importantly, the relations between possible syntactical combinations of such elements, on the one hand, and the logical relations they are intended to represent, on the other, makes what Peirce called "diagrammatic reasoning" with this kind of formal notation possible. In his own words, such reasoning according to systematic iconic relations "constructs a diagram according to a precept expressed in general terms, performs experiments upon this diagram, notes their results, assures itself that similar experiments performed upon any diagram constructed according to the same precept would have the same results, and expresses this in general terms" (Peirce and Eisele (1976), 47–8). We aim to show in what follows how the mathematics of *category theory*, which abstracts away from all determinate features of some collection of objects save for those systematic relations that are relevant to the general type or aspect of these objects that is at issue in some given context, provides a suitable formal environment both for representing and investigating such iconic relations at an appropriately abstract or "generic" level and for explicating the forms of diagrammatic reasoning they enable.

More specifically, in this paper we propose a formal reconstruction of Peirce's α-level of Existential Graphs, which corresponds to the system of classical propositional logic. The reconstruction lifts Peirce's diagrammatic notation to a more abstract mathematical setting where certain features underlying the iconic character of its syntax may be more perspicuously represented and better understood. In particular, then, we aim to bring to the fore the iconic nature of Peirce's diagrammatic notation by characterizing it in a purely structural way using the contemporary mathematical tools of category theory that in the late nineteenth and early twentieth centuries remained unavailable to Peirce himself. Specifically, we recast Peirce's Existential Graphs-α notation in the category-theoretical framework of categories of presheaf functors using the "generic figures" approach pioneered by Reyes, Reyes and Zolfaghari in Reyes et al. (2004).

We will thereby show that the diagrammatic syntax of Peirce's α-graphs may be perspicuously captured by constructing functor categories (specifically, categories of presheaves) with increasingly rich structure. This method of lifting iconic relations present in a diagrammatic syntax to explicit constructions in category theory has been previously developed in Caterina and Gangle (2013, 2015, 2016). The present paper

continues that line of research and extends it to the treatment of propositional variables, which are characterized categorically in a label-free, or "generic" manner through the use of groupoids. We will also show how the standard logical semantics for Peirce's system is readily accommodated in the new framework.

Recasting Peirce's diagrammatic logic in the contemporary mathematical context of categories opens Peirce's work to productive communication with a variety of different diagrammatic notations as well as abstract logical and mathematical structures, and it contributes to the ongoing study of diagrammatic and visual reasoning. In general, categories of presheaves provide a wide-ranging and flexible mathematical setting for expressing the representational capacities of syntactical objects built up constructively from specifiable types of parts that may be linked together freely in accordance with determined rules. Such rule-governed yet freely constructive features characterize many different types of diagrammatic representation, and the relations that emerge naturally among all the possible instances of any given type of such construction (for instance those of embeddings, overlaps and extensions) are often found to reflect iconically at their own level of organization the original rules themselves. Categories of presheaves offer in this respect a formal representational framework for exhibiting and investigating the iconic character of diagrammatic syntax as such. Most importantly, they generate in one stroke, as it were, a system of relations among all such objects which may then be conceived and represented in turn in a variety of ways. In particular, each *type* of object is represented in the category of presheaves by a "generic figure", a concrete instance and standard representative of that type. Relations of the appropriate kind from each of these generic figures may then be used to pick out tokens of that type in the overall system of presheaves. Additionally, commutative diagrams formulable as standard constructions within category theory may in turn represent ways that distinct instances of diagrammatic syntax may be determined in terms of the patching together or overlap of their iconic elements.

Using categories of presheaves to encode, on the one hand, the structural constraints and, on the other, to instantiate all the combinatorial possibilities for instantiation of a diagrammatic system like Peirce's EG provides a natural and efficient way to integrate the relevant features of the construction and also generate the system together with all its dynamics at one stroke. The iconic structure of the notation is exhibited directly by the complex internal relations of the presheaf category taken as a whole.

The broader philosophical significance of the formal results detailed below thus resides in demonstrating through them how a mathematically natural way of coordinating abstract systems of general relations with particular systems of sets and functions instantiating those relations provides a setting for making the iconic aspects of diagrammatic notation explicit. Because a diagrammatic notation is one that represents properties and relations of whatever type by instantiating to some degree or in certain respects those properties and relations themselves, clarifying how such representational systems work requires a theoretical framework in which multiple levels of relational abstraction and instantiation may be both registered and coordinated. Categories of presheaf functors provide precisely this sort of framework.

The paper is organized as follows. In Sect. 1 we briefly recall the main features of the α level of Peirce's Existential Graphs. In Sect. 2 we introduce and explain the notion of generic figures as special objects in categories of presheaves, illustrating the abstract

notions involved with two concrete examples, directed graphs and forests, the latter of which will be central to our reconstruction of Peirce's diagrammatic syntax. Section 3 formalizes a presheaf category $\mathcal{EG}_{\alpha^*}$ that corresponds to a variable-free (cuts-only) fragment of Peirce's α-graphs. It also presents a representation within category theory of an algorithmic procedure reducing any such variable-free graph to either the empty sheet (true) or the empty cut (false) in accordance with the standard semantics for the graphs. Section 4 then introduces label-free or "generic" variables into this framework, thereby capturing the abstract relations of identity and difference that underlie the diagrammatic usage of propositional variables in Peirce's EG-α. This allows for the formal characterization of the category $\mathcal{EG}_{\alpha}$ which culminates the overall generic-figures reconstruction. Section 5 shows how the logical derivability of one graph from another may be characterized in the new context, and Sect. 6 summarizes the results and indicates directions for further research and exploration.

1 Peirce's Existential Graphs

In this section we introduce Peirce's diagrammatic syntax for formal logic that he named Existential Graphs. Peirce himself developed three increasingly sophisticated levels of the graphs (hereafter EG), which he denoted alpha, beta and gamma. The alpha level (EG-α) corresponds to classical propositional logic, the beta level (EG-β) to first-order predicate logic with identity, and the gamma level (EG-γ) to modal and higher-order logics. We summarize here the basic syntax and semantics of Peirce's EG-α, referring any readers unfamiliar with the graphs to some of the main literature that fills in the necessary details. We also provide a brief introduction to the formalism of category theory and the notion of presheaf and presheaf topos that will serve as the main constructions in our analysis. Here too, the discussion will be cursory for reasons of space and the interested reader is directed to standard literature for further details.

We now turn to the basic elements of Peirce's EG-α and present a schematic introduction for those unfamiliar with the graphs. Besides the rigorous mathematical presentation in Brady and Trimble (2000), detailed but more accessible treatments may be found in Roberts (1973) and Shin (2002).

Every EG-α graph is understood by Peirce to represent a propositional assertion according to a regimented diagrammatic syntax. The syntax consists of three elements: the Sheet of Assertion, characters and seps (or cuts). An EG-α graph is composed of characters and seps "scribed" on the Sheet of Assertion.

1. The blank sheet (called the Sheet of Assertion or SA) is both the site on which graphs are *scribed* and is itself a graph (called the empty graph).
2. A *character* is any reproducible image (an iconic symbol, typically letters of the Roman alphabet) scribed on part of the SA.
3. Characters may be enclosed, along with a local area surrounding them (a neighborhood on the SA) that may or may not include other characters, by a closed curve called a *sep* (or *cut*). It is convenient to construct these generally as ovals or circles. These seps may not intersect characters, nor may they intersect one another. They

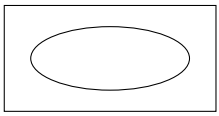
may, however, be nested with any number of characters and seps scribed in the areas or enclosures they distinguish.

The class of EG-α graphs may be characterized through the following recursive definition:

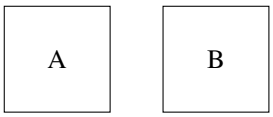
The sheet of assertion is a graph



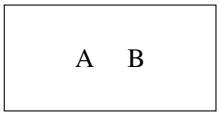
The sep is a graph



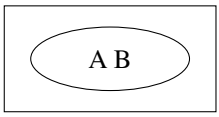
Every character is a graph



If α and β are graphs, then $\alpha\beta$ is a graph



If γ is a graph, then  *is a graph*



Let us first notice that the EG-α graphs constructed in this way separate regions in the sheet of assertion into evenly and oddly enclosed areas, as the following picture shows (Fig. 1).

We omit a detailed discussion of Peirce's transformation rules for EG-α. A full treatment of these rules may be found in Roberts (1973). As a reminder to the reader, examples of all of the rules are provided in an informal, iconic way below (Fig. 2).

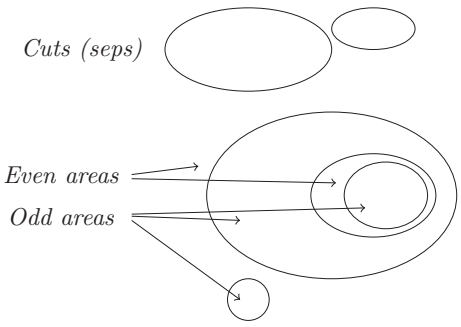


Fig. 1 Distinctions of even and odd areas

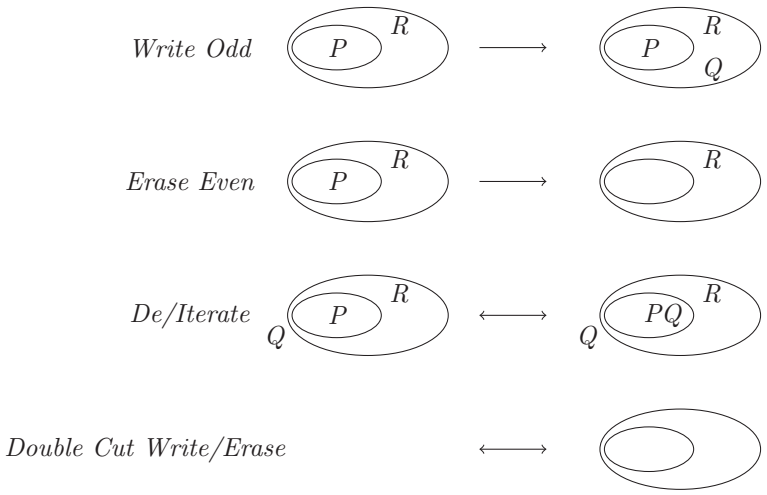


Fig. 2 EG-α transformation rules

The correspondence between EG-α and the classical propositional calculus (with seps interpreted as the *negation* of their contents and the inscription of multiple (sub)graphs on the same area as the logical *conjunction* of those (sub)graphs) has been exhaustively investigated by Zeman (1974) and more recently in a much broader context in Brady and Trimble (2000). The iconic quality of Peirce's calculus does indeed possess, as shown by several authors, including the knot theorist Louis Kauffman in Kauffman (2001), a truly topological nature. The derivation rules themselves may be characterized in a combinatorial-topological manner, such that the dual nature of a graph as possessing both syntactical and semantical features emerges naturally. In fact, two of the three basic elements of the EG-α syntax, the Sheet of Assertion and the seps, are defined topologically. This feature induces several of the aspects of the EG-α system that distinguish it from the more standard linear symbol-strings notation, such as the obviation of axioms of commutativity.

In the next sections we will show how the iconic character of Peirce's notation may be captured by representing it through the use of generic figures in suitable categories of presheaves.

2 Presheaf Categories and Their Generic Figures

The approach we follow in this paper was introduced in the book *Generic Figures and their Glueings* by Reyes et al. (2004). Its point of departure is to consider classes of structures, defined set-theoretically yet intuitively pictured as types of diagrams (sets, bisets, directed graphs, etc.) and, given that each element in the class is itself built up from determinate *kinds* of ingredients (set-elements represented as dots or points, pairs of such points, vertices and arrows, etc.), and to define a *category of generic*

figures $\mathcal{GF}$ in which those very ingredients are the objects.[1] Each of the objects of $\mathcal{GF}$ is called, by those authors, a *generic figure*.[2] The morphisms in $\mathcal{GF}$ then correspond to the structural relations among the ingredients in the original class (determinations of source and target vertices for each arrow in directed graphs, for instance).

Once the category $\mathcal{GF}$ of generic figures is defined, each element in the class of structures it is meant to represent can be seen as a particular (contravariant) *functor* from the category $\mathcal{GF}$ to the category **Set** of sets and functions. The class of these functors in turn constitutes a category of *presheaves* which induces a *topos* structure. The category of generic figures helps to provide a constructive characterization of whatever class of "diagrammatic" structures is at issue, and the required properties defining the associated topos can be explicitly constructed. We wish to emphasize the peculiar *level of abstraction* at which the presheaf category generated by $\mathcal{GF}$ resides. It may be understood both as a system of multiple concretizations by particular systems of sets and functions of the abstract system of relations encoded by $\mathcal{GF}$ itself *and* as a kind of general template or schema for multiple types of notational systems that may be used to express the particular systems of sets and functions in different empirical (but in the relevant sense representationally equivalent) modes.

The generic figures approach to diagrammatic notation thus essentially works by distinguishing and relating two mathematical structures, categories, via structure-preserving maps called functors. One is relatively abstract and variable (the source of the functor) and the other (the target) is the standard mathematical workhorse of sets and functions, fixed and relatively concrete. The distinct roles of the two categories may be apprehended by noting the different statuses accorded to the objects and morphisms in each case. In the source-category, the objects are conceived of as merely labeled types. The relevant features of the category are only those given by the arrows or morphisms, which effectively encode the rules of construction for syntactical components. The objects of the target category of sets and functions, on the other hand, are relevant due to their internal structure, namely the mathematically minimal structure of the cardinality of a collection, its mere "how many". Each element of a set is understood to represent a token of its instantiated type. The external relations between these elements are then variably specified in a determinate way in each case, but always as globally constrained by the invariant structure of the source-category. In other words, the syntactic structural constraints are given by the source category's system of relations and the category of presheaves instantiates this system in every way possible, given those constraints.

The key mathematical construction underlying the generic figures approach derives from the fact that to each object of $\mathcal{GF}$, as a result of the Yoneda lemma, is associated its *representable set* in the topos, and therefore each structure can be thought of as a

[1] A mathematical *category* consists of a class of *objects* together with *morphisms* or arrows between objects subject to axioms of *identity* (every object is equipped with an identity morphism that composes inertly), *composition* (head-to-tail morphisms compose to a unique morphism) and *associativity* (paths of morphisms compose uniquely). For a comprehensive introduction and details filling out this rough characterization, see Mac Lane (1998) and Spivak (2014).

[2] The text of Reyes et al. (2004) is at times ambiguous between referring to the objects of the base category as the generic figures and using the term to refer to the representable sets that represent those objects. Since these are in one-to-one correspondence and context usually determines which is meant, the ambiguity remains basically innocuous.

category of its figures. This new category is constructed as a gluing of several copies of the representable sets. This is particularly useful for characterizing constructively the *subobject classifier* of the topos as well as the truth values of its intrinsic logic (Reyes et al. (2004) ch. 6). In this paper we make use of only the elementary constructions of presheaf topoi involving representable sets, but we hope that readers familiar with category theory will recognize how the much more powerful resources of topoi and topos logic thereby become freely available by framing Peirce's system in this context. In what immediately follows, we work through a pair of elementary examples to clarify the notions of *presheaf category* and *representable set* before turning to the reconstruction of Peirce's EG-α in this framework.

2.1 Presheaf Categories

Given a (small) category $\mathcal{C}$, a functor F from $\mathcal{C}^{op}$ into **Set**, the category of sets and functions, is a $presheaf$. Here, $\mathcal{C}^{op}$ represents the category generated by reversing all the arrows in $\mathcal{C}$ while preserving objects, identity morphisms and composition of morphisms. Schematically, a morphism $f : A \to B$ in $\mathcal{C}$ becomes a morphism $f' : B \to A$ in $\mathcal{C}^{op}$. A natural transformation from one such functor F to another F' is then a family of functions in **Set** that maps one presheaf into another.

More generally, given any two categories $\mathcal{C}$ and $\mathcal{D}$, the *functor category* $\mathcal{D}^{\mathcal{C}}$ is defined as follows:

- The objects of $\mathcal{D}^{\mathcal{C}}$ are all *functors* $\mathcal{C} \to \mathcal{D}$.
- The arrows of $\mathcal{D}^{\mathcal{C}}$ are all *natural transformations* between functors $\mathcal{C} \to \mathcal{D}$.

Natural transformations are morphisms between functors: given two functors $F \in \mathcal{D}^{\mathcal{C}}$ and $G \in \mathcal{D}^{\mathcal{C}}$, a natural transformation between F and G is a family of morphisms η_O parametrized by the objects $O \in \mathcal{C}$ such that the following diagram commutes for any two objects A and B that are connected by a morphism f in $\mathcal{C}$:

$$\begin{array}{ccc} F(A) & \xrightarrow{F(f)} & F(B) \\ \downarrow{\scriptstyle \eta_A} & & \downarrow{\scriptstyle \eta_B} \\ G(A) & \xrightarrow{G(f)} & G(B) \end{array}$$

In the case of the functor category of all presheaves over $\mathcal{C}$, the collection of all functors $\mathcal{C}^{op} \to$ **Set** may be taken to be the objects of the new category $\mathbf{Set}^{\mathcal{C}^{op}}$, the morphisms of which are all the natural transformations between these functors. Each functor may be associated with an identity natural transformation. Also, natural transformations compose in the appropriate way and this composition is associative, so the axioms of a category are satisfied. Such a category is called a *presheaf category*. For details, see Reyes et al. (2004) and Mac Lane (1998) ch. 2.

2.2 Generic Figures and Representable Sets

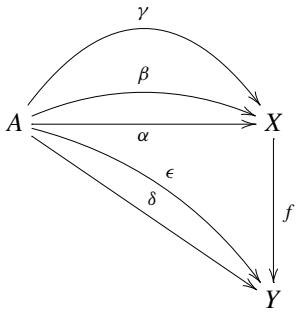
The structure of the objects of any presheaf category may be probed by way of the relations that obtain between certain special presheaves representing the objects of the base category, that is, the *generic figures*. These special presheaves are called *representable sets*. The relations that are used are natural transformations, as described above.

Each object in the original category $\mathcal{C}$ may be represented by a special presheaf called a representable set.[3] This representable set is unique up to isomorphism. Natural transformations from a representable set into any presheaf in the category may be understood as selections of objects of that type in the corresponding structure.

In order to formally define representable sets, we first define a canonical functor associated with every object of $\mathcal{C}$. From a philosophical point of view, this functor (which is closely related to Yoneda's lemma) may be understood to capture the base category *as relativized to just one of its constituent objects*, thus making concrete in terms of sets and functions exactly those relational features of the base category that are relevant for determining an object of that type in whatever representational mode.

Definition 1 Given any object $A \in O(\mathcal{C})$, the *hom-functor based at A* is denoted by $Hom(A, -) : \mathcal{C} \longrightarrow \mathbf{Sets}$ and takes any object $X \in O(\mathcal{C})$ into the set $Hom(A, X)$ of all the morphisms from A to X in $\mathcal{C}$. How arrows in $\mathcal{C}$ are treated by this functor is discussed in the example immediately below.

Example 1



Let us assume the following structural equation for this diagram:

$$f \circ \alpha = \epsilon,\ f \circ \beta = \epsilon,\ f \circ \gamma = \delta$$

We then have

$$Hom(A, X) = \{\alpha, \beta, \gamma\},\ Hom(A, Y) = \{\delta, \epsilon\}$$

[3] The terminology of "representable set" is somewhat unfortunate, as a representable set is, on the one hand, not a set but rather a presheaf and, on the other hand, more characteristically representative than representable. Nevertheless, we maintain the standard usage.

Notice that so far we have only expressed the action of $Hom(A, -)$ on objects. It is an easy exercise to show that $Hom(A, -)$ is an actual functor (that is, that it acts in a coherent and well-controlled way on morphisms by defining functions between Hom-sets in terms of pre-composition).

In the example above, for instance, $Hom(A, f)$ is a map between the sets $Hom(A, X)$ and $Hom(A, Y)$ defined as follows:

$$Hom(A, f)(\alpha) = f \circ \alpha = \epsilon$$
$$Hom(A, f)(\beta) = f \circ \beta = \epsilon$$
$$Hom(A, f)(\gamma) = f \circ \gamma = \delta$$

Now we can define a representable set:

Definition 2 A functor F for which there exists an object A such that $F(X) = Hom(A, X)$ for any object $X \in O(\mathcal{C})$ is said to be *representable*.

The following pair of examples should help to make the notions of presheaf category and representable set more vivid. The second example, that of forests, will be important in the reconstruction of Peirce's EG-α.

2.2.1 Example: The Category $\mathbf{Set}^{\mathcal{G}^{op}}$ of Directed Graphs

Consider the category $\mathcal{G}$ consisting of exactly two objects V and A and two non-identity arrows $s, t : V \to A$. This category is pictured here, with the identity arrows on V and A not shown:

$$V \underset{t}{\overset{s}{\rightrightarrows}} A$$

Contravariant functors from $\mathcal{G}$ into **Set** may be thought of as directed graphs. Given such a functor D: $\mathcal{G}^{Op} \to \mathbf{Set}$, where $D(V)$ is the set of graph vertices and $D(A)$ is the set of graph arrows, $D(s)$ and $D(t)$ are then two functions $D(A) \to D(V)$ assigning a source-vertex and a target-vertex, respectively, to each arrow, that is, each element of $D(A)$.

In this presheaf category, the representable set h_V is the directed graph pictured here:

•

Given any functor D: $\mathcal{G}^{Op} \to \mathbf{Set}$, a natural transformation $\eta : h_V \to D$ will pick out a single vertex of the directed graph corresponding to D. The collection of all such natural transformations will correspond to all the vertices of the graph.

The representable set h_A for this category is the directed graph pictured here:

• ⟶ •

Given any functor D: $\mathcal{G}^{Op} \to \mathbf{Set}$, a natural transformation $\mu : h_A \to D$ will pick out a single arrow of the corresponding directed graph, and each arrow in the directed graph may be associated with a unique natural transformation of this type.

In short, the structure of any directed graph G may be articulated exhaustively in terms of morphisms $h_V \rightarrow G$ and $h_A \rightarrow G$ together with incidence relations among these.

2.2.2 Example: the Category $\mathbf{Set}^{\mathbb{N}^{op}} = \mathcal{F}$ of Forests

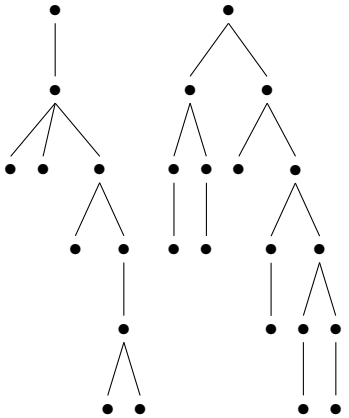
Consider the category $\overrightarrow{\mathbb{N}}$, the objects of which are the natural numbers 1, 2, 3,…and with morphisms between objects corresponding to the obvious $\leq$ ordering. Contravariant functors from this category into **Set** correspond to the highly useful mathematical structures called *forests*. For simplicity, we denote the category of all such functors with natural transformations between them as $\mathcal{F}$.

For $\mathcal{F}$, the presheaf category built over $\overrightarrow{\mathbb{N}}$, there are infinitely many representable sets, one for each natural number n (that is, each object of $\overrightarrow{\mathbb{N}}$). A representable set h_n corresponding to each such n is in this case the presheaf that takes each object $o \leq n \in \overrightarrow{\mathbb{N}}$ to a singleton set and all objects $o\prime \geq n \in \overrightarrow{\mathbb{N}}$ to the empty set, that is, a forest with the structure of a chain of n nodes. Natural transformations from representable sets h_n into other presheaves in this category correspond to paths of length n in the forest, or, equivalently, to nodes at depth n. For instance, the picture below corresponds to the representable set h_5:

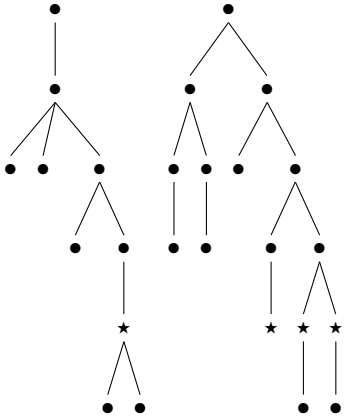


Thus, given a forest f (that is, a presheaf over $\overrightarrow{\mathbb{N}}$ or an object of $\mathcal{F}$) and a natural number n, the collection of natural transformations $h_n \rightarrow f$ will pick out all the nodes of that forest at depth n, or, equivalently, all paths in the forest of length n.

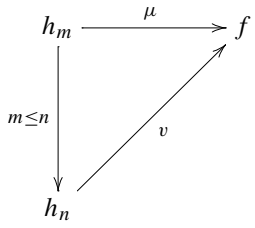
For example, the collection of all the natural transformations from h_5 into the forest pictured below



can be thought of as the action of picking out the star-shaped nodes below (or, equivalently, the paths ending at those nodes):



For two natural numbers m and n such that $m \leq n$, if two natural transformations $\mu : h_m \to f$ and $\nu : h_n \to f$ are such that the following diagram commutes:



then the path corresponding to μ in the forest f is an initial segment of the path corresponding to ν. Note that the morphism $m \leq n : h_m \to h_n$ as depicted in the diagram must exist and be unique since any representable set in the category of forests embeds in another representable set in at most one way and such embeddings naturally respect the order structure on $\mathbb{N}$.

In short, the structure of any forest f may be articulated exhaustively in terms of morphisms $h_n \rightarrow f$ for all $n \in \mathbb{N}$ together with incidence relations among these.

2.3 Topos Logic in Presheaf Categories

An important result in elementary category theory is that every presheaf category is a topos (Mac Lane 1998, pp. 106–7). For the purposes of the present paper, the only aspect of topos theory that will be needed is that within a topos, subobjects are represented by monic morphisms. The additional fact that monic morphisms themselves may be characterized in terms of morphisms from objects into the subobject classifier Ω of a topos and thereby induce the internal logic of the topos (see Mac Lane (1998), pp. 105–6) will be left tacit here for reasons of space, with detailed discussion of the relevance of this fact for the EG-α generic figures reconstruction reserved for future work. One important point worth noting, however, is that whereas the logic of EG-α is, under the standard interpretation, classical, the internal logic of the presheaf category detailed below is intuitionistic. This fact (which for reasons of space we can only state here) shows how the logical features of EG-α are indeed not directly encoded by this presheaf category but instead indirectly represented by effectively encoding constructive syntactical relations among all such possible graphs (in particular, embeddings and overlaps of graphs up to whatever depth), relations which are more finely-grained than can be captured by a classical two-valued semantics. Yet the fact that relevant logical semantic relations, as outlined in Sects. 3, 4 and 5 below, are indeed *also* encoded within the network of natural transformations among the presheaves in the category exhibits the complex relational syntactic-semantic connection that underwrites the iconic representational character of the graphs.

2.3.1 Monic Morphisms as Subobjects

Representable sets may be thought of as probes that pick out tokens of their corresponding figure types in the various objects of the relevant presheaf category. Since the figures thus selected are parts or "subobjects" of the presheaf probed by the representable set, the appropriate categorical notion for describing this relation is that of *monic morphism*.

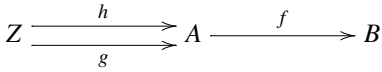
Given

$$A \xrightarrow{\quad f \quad} B$$

f is said to be *monic* if, for any two arrows

$$Z \underset{g}{\overset{h}{\rightrightarrows}} A$$

such that the pair of compositions represented by the diagram



are equal (that is, such that $f \circ g = f \circ h$), it must be the case that

$$h = g$$

We note in passing that statements about maps from representable sets into objects in a presheaf category may be conceived as maps from those objects into the topos subobject classifier Ω, which comes equipped with its own distinctive logical operations which are, in general, intuitionistic and not classical (see Reyes et al. (2004), chs. 6-9 for further details).

In what follows, we show how this general method of tracking structures in categories of presheaves using representable sets may be applied to the special case of Peirce's EG-α (as represented by presheaves), allowing for the analysis within that presheaf topos of logical properties of those graphs that are encoded diagrammatically, that is, iconically.

3 The Category $\mathcal{EG}_{\alpha^*}$ of Cuts-Only EG-α Graphs

In the previous section we gave a brief introduction to the notion of generic figures. The idea, in its most simplified version, is to define a formal representation of the building blocks of a certain class of structures within a categorical framework along with a coherent gluing scheme. In order to present our category-theory based approach to diagrammatic logic, in this section we will introduce a fragment of Peirce's EG-α diagrammatic systems for propositional logic which we will reconstruct as a presheaf category that we designate as $\mathcal{EG}_{\alpha^*}$. This category will represent the fragment of Peirce's system that consists only of graphs constructed without characters. That is to say, the objects of $\mathcal{EG}_{\alpha^*}$ will be "cuts-only" EG-α graphs, i.e. graphs consisting only of nested seps, or cuts, on the Sheet of Assertion.

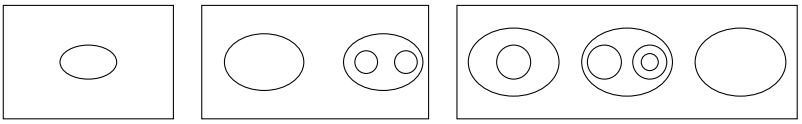
3.1 $\mathcal{EG}_{\alpha^*}$ Syntax

Every alpha graph may be conceived as a collection of non-overlapping closed curves (cuts) with labeled variable markers (characters) placed on the areas thereby determined. As said, we will here consider first the fragment of Peirce's EG-α system consisting only of cuts, that is, of graphs without characters. We name such graphs "cuts-only graphs".

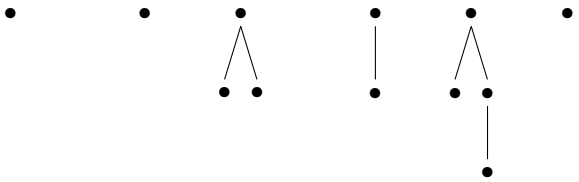
Peirce himself called the graph consisting of solely a cut the "pseudo-graph" and showed considerable hesitancy in even including it as a well-formed graph. We include it here as a fully-fledged element of our reconstruction with the justification that later developments in formal logic have led to the consensus of including the symbol $\perp$, read

"false" or "bottom" as an admissible element of the syntax. As we will show below, the single-cut graph or pseudo-graph corresponds semantically to the truth-value *false*.

So the class of cuts-only graphs includes representative instances such as the following:



We motivate the construction below by noting that every such cuts-only graph corresponds to a collection of trees (that is, a forest) in which each node is a cut. The graph of the SA corresponds obviously to the empty forest consisting of no nodes. For instance, each cuts-only graph pictured above may be translated into a corresponding forest depicted below:



3.1.1 Defining $\mathcal{EG}_{\alpha^*}$

We are now prepared to formally define our category $\mathcal{EG}_{\alpha^*}$ which is a subcategory of $\mathcal{F}$ the category of forests. The objects of $\mathcal{EG}_{\alpha^*}$ are functors $F : \overrightarrow{\mathbb{N}}^{op} \longrightarrow$ **FinSet** such that there exists some $n \in \overrightarrow{\mathbb{N}}$ where $F(n) = \varnothing$.[4] The morphisms of $\mathcal{EG}_{\alpha^*}$ are monic morphisms between the same objects in $\mathcal{F}$.

Proposition 1 *$\mathcal{EG}_{\alpha^*}$ is a category.*

Proof The composition of two monic arrows is a monic arrow. Composition is associative and identity arrows are monic. □

This formal characterization thus captures the relevant structure shared by finite forests and cuts-only graphs. The instantiations or concretizations by systems of sets and functions realized by the individual presheaves in $\mathcal{EG}_{\alpha^*}$ are thus in a philosophically interesting way neither as abstract as a "pure" or abstract category nor as concrete as a determinate expression in a given empirical notation. A presheaf category like $\mathcal{EG}_{\alpha^*}$ is in this respect not so much an iconic or diagrammatic notation of its own so much as a quite distinctive type of notation-template or notation-schema for producing and regulating a variety of possible iconic or diagrammatic notations sharing a common representational essence or propensity.

[4] **FinSet** is the category of finite sets and functions between them (a full subcategory of **Set**). We take presheaves into **FinSet** rather than **Set** in order to exclude the possibility of infinite conjunctions of cuts at any given depth.

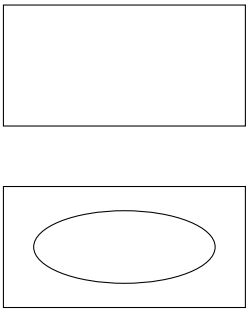
3.2 $\mathcal{EG}_{\alpha^*}$ Semantics

Having expressed the class of Peirce's cuts-only alpha graphs as the objects $O(\mathcal{EG}_{\alpha^*})$ of the category $\mathcal{EG}_{\alpha^*}$, we now aim to construct a function (that is, a morphism in the category **Set**)

$$eval : O(\mathcal{EG}_{\alpha^*}) \to \Omega$$

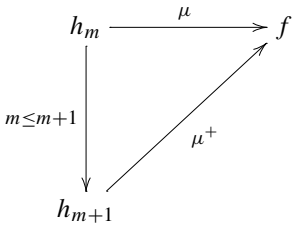
where Ω in this context represents the set of truth-values structuring our semantics (here, the two-element set $(\top, \bot)$).

In the category **Set** any two element set may serve as the subobject classifier Ω. In order to remain, as much as possible, "internal" to Peirce's system, we consider Ω here as the set consisting of the two cuts-only graphs shown here:



Intuitively, then, the function *eval* may be thought of as "reducing" each cuts-only graph (each object g of $\mathcal{EG}_{\alpha^*}$) to exactly one of the two simple cuts-only graphs shown above. In fact, this vague notion of "reduction" can be made precise, as we show below.

We define a *child* of a node n in a forest f as a node that is immediately below n in f. Formally, given a forest $f \in O(\mathcal{EG}_{\alpha^*})$ and a node n at some depth m selected by $\mu : h_m \longrightarrow f$, a child of n is any node selected by $\mu^+ : h_{m+1} \longrightarrow f$ such that this diagram commutes



A *terminal node* is a node that has no children. A *pre-terminal node* is a node with at least one child that is terminal.

Given an EG-α cuts-only graph g, we reduce it to either the empty sheet (**true**) or the empty cut (**false**) by applying the following iterative pruning procedure:

Procedure 1

(i) Consider the forest representation F of the graph g.
(ii) Take any pre-terminal node n in F.
(iii) Eliminate the node n and all nodes below it, yielding a new forest F'.
(iv) Replace F with F' and repeat (ii) and (iii) until the forest F' is either empty or consists of single nodes only (nodes of at most depth 1).
(v) If the forest F' is empty, replace it with the empty Sheet of Assertion. If it consists of single nodes only, replace it with a single empty cut.

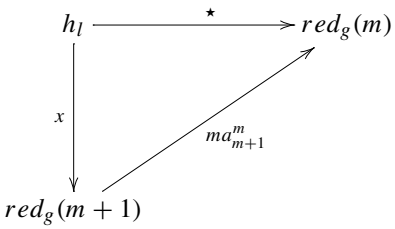
Note that this formulation of the pruning procedure uses terminology particular to the forest representation of $\mathcal{EG}_{\alpha^*}$. The same procedure as applicable to Peirce's areas-and-cuts notation would necessarily be formulated somewhat differently (though in a formally equivalent way). The right level of abstraction or formalization that captures the essence of the procedure common to both kinds of notation is given by the categorical generic figures representation.

The procedure may be formally defined in the presheaf categorical context solely in terms of generic figures in the following way. First, given an arbitrary object g of $\mathcal{EG}_{\alpha^*}$, we take the set $Sub(g)$ of all monic morphisms into g. This set may be associated with the object Ω^g. The elements of $Sub(g)$ form a lattice ordered by inclusion, the *subobject lattice* of g which we denote $\overrightarrow{Sub(g)}$.

The procedure outlined above may then be formalized as a path through $\overrightarrow{Sub(g)}$. Formally, a semantic reduction of an object g of $\mathcal{EG}_{\alpha^*}$ is a finite sequence of elements of $Sub(g)$, that is, a function $red_g : n \to Sub(g)$ for some $n \in \mathbb{N}$ such that

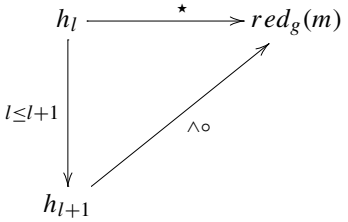
1. $red_g(1) = g$
2. for all $red_g(m)$ and $red_g(m+1)$, there is a monic arrow $ma^m_{m+1} : red_g(m+1) \rightarrowtail red_g(m)$
3. given $red_g(m)$ and $red_g(m+1)$ there exists a path of length l in $red_g(m)$, that is, a special morphism $\star : h_l \to red_g(m)$ such that

   (a) there are no morphisms x such that the following diagram commutes:



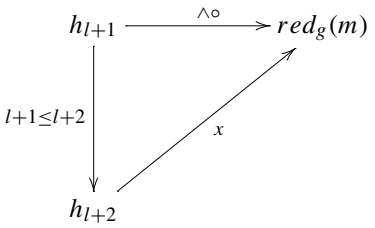
   (intuitively: path $\star$ is not in $red_g(m+1)$).

(b) there exists a path $\wedge\circ : h_{l+1} \rightarrow red_g(m)$ of length $l+1$ in $red_g(m)$ such that the following diagram commutes:



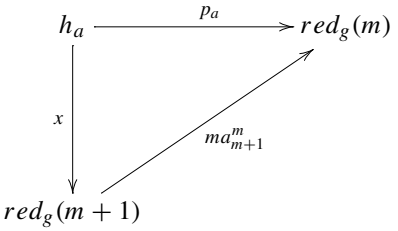
(intuitively: path $\star$ is extended by one node to $\wedge\circ$ in $red_g(m)$)

and there is no path x of length $l+2$ such that the following diagram commutes:

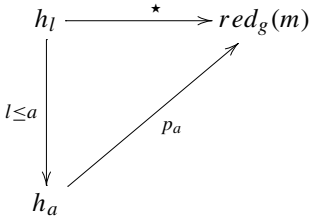


(intuitively: no path extends $\wedge\circ$ further; it is a terminating path).

(c) for all morphisms $p_a : h_a \rightarrow red_g(m)$ such that there is no morphism x that makes the diagram



commute (paths in $red_g(m)$ that are not in $red_g(m+1)$), the following diagram *does* commute:



($\star$ is a subpath of a path of length a in $red_g(m)$).

4. there are no morphisms $h_2 \rightarrow red_g(n)$ (intuitively: the last step of the sequence results in a graph of at most depth 1).

It should be clear, first of all, that at least one such sequence always exists, given an arbitrary object g of $\mathcal{EG}_{\alpha^*}$. Since each step in the sequence represents the "pruning" of at least two nodes from the previous step and every object of $\mathcal{EG}_{\alpha^*}$ is composed of a finite number of nodes, successive applications of the pruning procedure will eventually reduce a graph/forest to either the empty sheet or a collection of empty cuts at level one.

Perhaps less obvious is the fact that any such sequence for a given graph g necessarily concludes in the same final step. More precisely, notice that every time we apply steps (ii) and (iii), we need to make a choice among all the pre-terminal nodes that exist at that stage, and we need to show that any variant sequence of such choices leads nonetheless invariably to the same final result.

Theorem 1 *Given a finite forest F, the algorithm outlined in Procedure 1 will output either the empty sheet or the empty cut, independently of the choices made in step (ii).*

Proof We first label all the nodes of F either 0 or 1 according to the following rule:

- All nodes are labeled 0 if they have at least one child labeled 1 and they are labeled 1 otherwise. □

It should be clear that this labeling is well-defined and uniquely determined for any finite forest. We note first that prior to applying the pruning procedure every terminal node of the forest is necessarily labeled 1 since these (by definition) have no children. We then note that repeated applications of steps (ii) and (iii) in the pruning procedure above will always select and delete a node labeled 0. It is impossible for a node labeled 1 to have any children also labeled 1 according to the labeling rule. Since only a pre-terminal node can be chosen and deleted in steps (ii) and (iii), no such node can ever be labeled 1. Thus every node selected in step (ii) must be labeled 0.

Eventually, *all* nodes labeled 0 must be deleted. This will result in a final forest that has at most depth 1. *Q.E.D.*

Finally, it is straightforward to show that when the objects of $\mathcal{EG}_{\alpha^*}$ are interpreted logically in accordance with Peirce's original system, the steps in the sequence preserve logical value. To see this, it is necessary only to realize that step (iii) of the procedure above reflects two fundamental logical principles under the Peircean interpretation of the cut as the negation of its contents and the value of any area of a graph as the value of the conjunction of the contents scribed on that area: a) the conjunction of the truth value "false" ($\bot$) with anything returns the truth value $\bot$ and b) the negation of $\bot$ corresponds to the truth value "true" ($\top$). So the syntactical elimination of a node of a tree along with all nodes below it if it has at least one terminal child is in this way logically justified: $\neg(\bot \wedge X) = \top$.

Given these facts, we may define the global evaluation function $eval$ as indicated above, ranging over all EG cuts-only graphs, assigning to each such graph the value "empty sheet" (true) or "single empty cut" (false) based on the following criteria:

Given an element $g \in O(\mathcal{EG}_{\alpha^*})$,

1. $eval(g) = \top$ if $\exists n$ such that $red_g(n)$ is the empty sheet,
2. $eval(g) = \bot$ otherwise.

To remain fully internal to Peirce's diagrammatic syntax, we may identify $\top$ with the empty Sheet of Assertion and $\bot$ with a graph consisting of a single empty cut.

4 The Category $\mathcal{EG}_\alpha$ of Generic EG-α Graphs

We now extend our system of cuts-only graphs (equivalent to the category of finite forests and monic maps) to include variable tokens, that is, the placement of "characters" of distinct (possibly identical) types on the areas determined by the nested cuts of cuts-only graphs. In accordance with the spirit of the categorical generic figures approach, we will treat these variable tokens in a "generic" manner. That is, we will consider them only insofar as they are distinguished or identified with one another. In this way, we again use the categorical presheaf framework to capture a level of formal abstraction intermediate to the pure structuralism of abstract categories on the one hand (where objects are mere indexical placeholders or contentless "black boxes" for systems of morphisms) and on the other hand to specific notations which involve various superfluous (but also often intuitively suggestive) empirical features. Here too we recognize a kind of reduction to the diagrammatically notational *essential* in this particular level of abstraction. In this case, the approach will support a label-free determination of variables in our reconstruction of Peirce's system. We will call such label-free variables "generic variables".

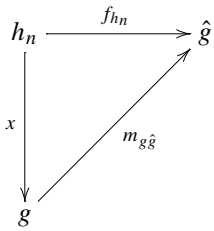
4.1 $\mathcal{EG}_\alpha$ Syntax

4.1.1 From EG_α Cuts-Only Graphs to Graphs-with-Marks

To recall, we define a cuts-only graph g as a functor $g : \overrightarrow{\mathbb{N}}^{op} \to \mathbf{FinSet}$ such that for some $n \in \mathbb{N}$ $g(n) = \varnothing$.

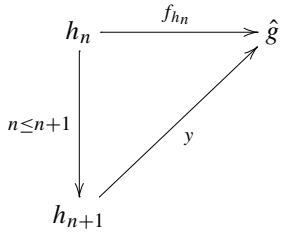
Now, given some such g that serves as the skeleton of cuts for our new graph with variables, we introduce the variables by adjoining them as possible empty cuts on the areas where the variables appear. Formally, we define a *graph-with-marks* g^+ as an ordered pair $(g, \hat{g})$ where $\hat{g}$ is also a cuts-only graph, that is, a functor $\hat{g} : \overrightarrow{\mathbb{N}}^{op} \to \mathbf{FinSet}$ such that for some $n \in \mathbb{N}$, $\hat{g}(n) = \varnothing$. Furthermore, g^+ satisfies the following additional conditions:

1. g is a subobject of $\hat{g}$. That is, there exists a monic morphism $m_{g\hat{g}} : g \rightarrowtail \hat{g}$, or, simply, a morphism $m_{g\hat{g}} : g \to \hat{g}$ in $\mathcal{EG}_{\alpha^*}$. (intuitively: $m_{g\hat{g}}$ embeds g in $\hat{g}$)
2. for all $n \in \mathbb{N}$ and for all n-figures $f_{h_n} : h_n \to \hat{g}$ if there does *not* exist a morphism x such that the following diagram commutes:



(intuitively: f_{h_n} embeds into $\hat{g}$ but *not* into g, extending g)

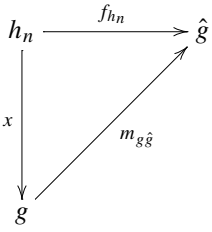
then there does not exist any h_{n+1} and morphism $y : h_{n+1} \to \hat{g}$ such that the following diagram *does* commutes:



(intuitively: f_{h_n} extends g by no more than just one node).

We denote G^+ the set of all such g^+. To each g^+ corresponds a single morphism in $\mathcal{EG}_{\alpha^*}$.[5]

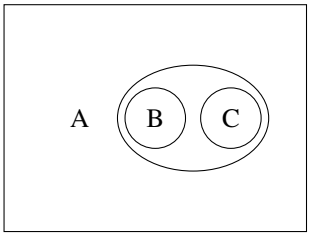
We call a g^+-*mark* any n-figure $f_{h_n} : h_n \to \hat{g}$ such that *no* morphism x makes the following diagram commutative:



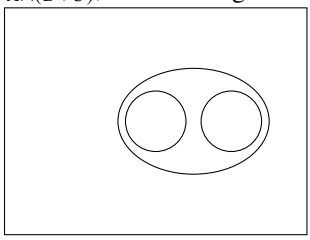
Note that this definition of *mark* is essentially the first part of the second condition in the definition of g^+ above. This fact should help to clarify the name "graph-with-marks" in the previous definition. For any graph-with-marks g^+ we designate the set of all g^+-marks as $Marks_{g^+}$.

Some diagrams should help clarify the construction. Take the well-formed EG-α graph $EG_{A\wedge(B\vee C)}$ with variables A and B and C below:

[5] Note, however, that the converse does not hold in general. It is nonetheless straightforward, if somewhat tedious, to define a function from morphisms in $\mathcal{EG}_{\alpha^*}$ into G^+ that replaces branching extensions of more than one node (cases in which $\hat{g}$ is too large) with single nodes. Details are left to the reader.

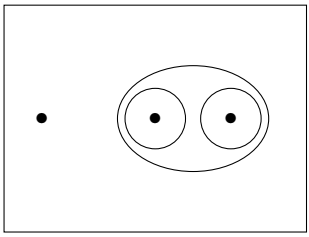


The diagram on the left below represents the cuts-only skeleton $g_{EG_{A\wedge(B\vee C)}}$ of the graph $EG_{A\wedge(B\vee C)}$, and the diagram on the right represents the same skeleton as a forest.

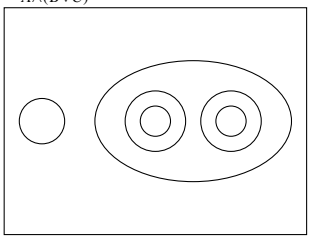


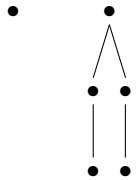


Now consider the generic version of $EG_{A\wedge(B\vee C)}$ in which all variables are replaced with dots (these are meant to represent the "marks" referred to above). This is pictured below:



Now we can construct $\hat{g}_{EG_{A\wedge(B\vee C)}}$ by turning each of the dots (marks) in the graph above into a new empty cut. The cuts-only graph below on the left shows $\hat{g}_{EG_{A\wedge(B\vee C)}}$ and the same graph represented as a forest appears on the right.



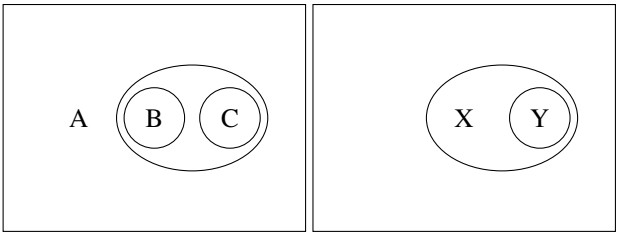


The reader should notice that the forest representing $g_{EG_{A\wedge(B\vee C)}}$ is a sub-forest of $\hat{g}_{EG_{A\wedge(B\vee C)}}$. Formally, there is an inclusion map (a monic morphism) $g_{EG_{A\wedge(B\vee C)}} \rightarrow \hat{g}_{EG_{A\wedge(B\vee C)}}$. This fact corresponds to condition 1 above. In addition, because each variable in the original graph was replaced by a single empty cut, the extension of $g_{EG_{A\wedge(B\vee C)}}$ into $\hat{g}_{EG_{A\wedge(B\vee C)}}$ lengthens no path in the forest by more than one node. This fact corresponds to condition 2. We thus have that $EG_{A\wedge(B\vee C)}$ in Peirce's EG-

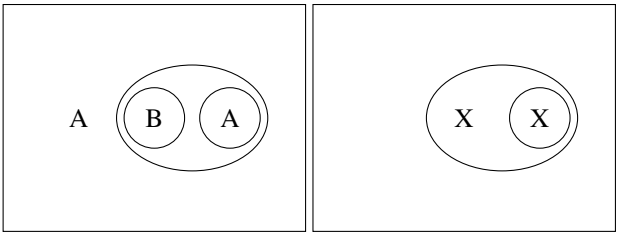
α system is translatable into $g^{+}_{EG_{A\wedge(B\vee C)}} = (g_{EG_{A\wedge(B\vee C)}}, \hat{g}_{EG_{A\wedge(B\vee C)}})$ in our generic figures reconstruction.

4.1.2 Variable-Typing Graphs-with-Marks

The generic figures formalization of the EG-α syntax described above relies upon the mutual independence of the truth-values of all the marks for any given graph g^{+}. Thus, it is capable of handling cases of EG-α graphs such as the following:



But this formalization of the EG-α syntax remains insufficient for evaluating graphs such as these:



To accommodate cases like the latter ones, in which variable-types (and not only variable-tokens, or marks) must be tracked, we must have recourse to formal means for "identifying" distinct marks, that is, for treating them as tokens of some common type.

What is required, essentially, is to partition the set of marks into equivalence classes, where two marks are considered equivalent if they are tokens of the same variable type.

In order to remain within a categorical framework, we formalize these equivalence classes as a *groupoid* defined on the set of marks. The concept of a groupoid is a useful generalization of the concept of a group with many applications in topology and other fields. For details, see, for instance, Brown (2006). The categorical definition of a groupoid is quite straightforward: it is simply a category all of whose morphisms are isomorphisms.

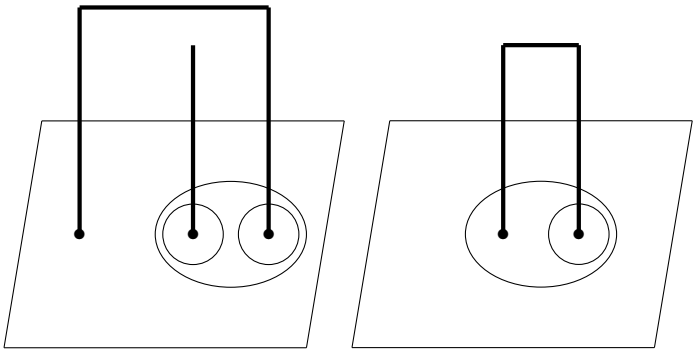
Definition 3 A *groupoid* is a category $\mathcal{C}$ such that for every morphism $f : X \longrightarrow Y$ in $\mathcal{C}$ there exists a morphism $g : Y \longrightarrow X$ such that $f \circ g$ is the identity morphism on Y and $g \circ f$ is the identity morphism on X.

For our purposes, the identification of collections of more than one mark as instances of some common variable type takes the form of a groupoid on the set of marks. Thus

a *variable-typing* of a graph g^+ is a category $\mathcal{VT}_{g^+}$ whose objects are the elements of $Marks_{g^+}$ and with morphisms defined as follows: Given two objects X and Y of $\mathcal{VT}_{g^+}$ there is a morphism $m : X \rightarrow Y$ if and only if X and Y are instances of the same variable type. The reader may easily check that this defines a category and, indeed, a groupoid.

A *graph with variable-typed marks* g^+_{var} is then defined as a pair $(g^+, \mathcal{VT}_{g^+})$. We call such a graph a *generic EGα graph* since the groupoid variable-typing operation abstracts away from the particular variable labels of EG-α graphs and preserves only the identities and differences of these labels. Thus, every standard EG-α graph corresponds to a unique graph with variable-typed marks (up to isomorphism), and every graph with variable-typed marks corresponds to a class of EG-α graphs that share the same formal structure.

Diagrammatically, the construction may be pictured as follows, where the two diagrams immediately below represent the generic reconstructions of the pair of EG-α graphs depicted above:



The fact that a morphism exists between two variable-marks in the appropriate category $\mathcal{VT}_{g^+}$ is here represented diagrammatically by connecting those marks with a thick, continuous line.[6] Of course, other representations are quite possible and might indeed be more appropriate in particular contexts. What matters is that whatever representational mode is chosen, it should respect and convey the appropriate categorical relations (here, of identity and difference) at stake.

4.1.3 Defining $\mathcal{EG}_\alpha$

We may now formally define the syntactic category $\mathcal{EG}_\alpha$ of generic EG-α graphs and generic EG-α graph morphisms.

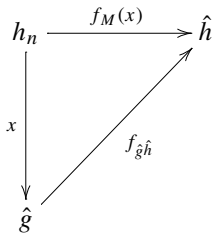
The objects of $\mathcal{EG}_\alpha$ are generic EG_α graphs g^+_{var}, that is, pairs $(g^+, \mathcal{VT}_{g^+})$.

A $\mathcal{EG}_\alpha$-morphism $f : (g^+, \mathcal{VT}_{g^+}) \longrightarrow (h^+, \mathcal{VT}_{h^+})$ from g^+_{var} to h^+_{var} consists of

- a pair of monic arrows $f_{gh} : g \longrightarrow h$, $f_{\hat{g}\hat{h}} : \hat{g} \longrightarrow \hat{h}$ in $\mathcal{EG}_{\alpha^*}$

[6] We represent these lines in the third dimension perpendicular to the plane of the sheet of assertion in order to preclude any possible ambiguity with the "lines of identity" used in Peirce's EG-β system.

- an injective function $f_M : Marks_{g^+} \longrightarrow Marks_{h^+}$ such that for all g^+-marks (that is, for every $x \in Marks_{g^+}$) the following diagram commutes:



- a coherent variable-typing inclusion, that is, a groupoid functor $f_{\mathcal{VT}_{g^+h^+}} : \mathcal{VT}_{g^+} \longrightarrow \mathcal{VT}_{h^+}$ that respects the function f_M. Formally, $f_{\mathcal{VT}_{g^+h^+}}(x) = f_M(x)$ for all objects x of $\mathcal{VT}_{g^+}$ = elements x of $Marks_{g^+}$.[7]

Proposition 2 *$\mathcal{EG}_\alpha$ is a category.*

Proof The pairs of arrows f_{gh} , $f_{\hat{g}\hat{h}}$ compose just as they do in $\mathcal{EG}_{\alpha^*}$. Composition of functions f_M works just as in the category **Set**. And the groupoid functors $f_{\mathcal{VT}_{g^+h^+}} : \mathcal{VT}_{g^+} \longrightarrow \mathcal{VT}_{h^+}$ compose as morphisms in the 2-category **Cat** of (small) categories. Identity morphisms are defined in each case in the obvious way and associativity of composition is inherited from the respective background categories. *Q.E.D.* □

As in the case of the category $\mathcal{EG}_{\alpha^*}$, we conceive of the morphisms of $\mathcal{EG}_\alpha$ as embeddings of one EG-α graph into another. Here, however, variables may be embedded as well, so long as they respect the respective variable-typings of the source and target graphs.

4.2 $\mathcal{EG}_\alpha$ Semantics

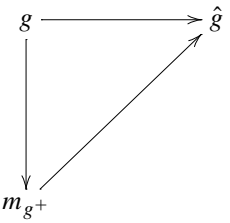
We first treat the simpler class of cases in which all variables in the graph are independent. That is, those cases in which there is no more than one token of any given variable type.

We then examine the more general class of cases in which more than one token of a given variable type may appear on the same or different areas of the graph.

[7] Note that the fact that $f_{\mathcal{VT}_{g^+h^+}}$ is a functor between categories (specifically groupoids) guarantees that the variable typing is respected across the mapping. Two variable-tokens that are identified as being of a common type in g^+_{var} *must* be mapped to variable-tokens that are of the same type in h^+_{var} (the existence of an arrow $X \longrightarrow Y$ in $\mathcal{VT}_{g^+}$ implies the existence of an arrow $f_{\mathcal{VT}_{g^+h^+}}(X) \longrightarrow f_{\mathcal{VT}_{g^+h^+}}(Y)$ in $\mathcal{VT}_{h^+}$).

4.2.1 Evaluating Graphs-with-Marks

A *model* of g^+, designated m_{g^+} is defined as a cuts-only graph such that the following diagram commutes, where all three morphisms are monic:



We formulate the condition in this way in order to emphasize the intermediary character of models of g^+ with respect to g and $\hat{g}$. Note, for instance, that both g and $\hat{g}$ are necessarily models of g^+.

The collection of all such m_{g^+} for some given g^+ is denoted M_{g^+}.

Proposition 3 *For any graph-with-marks g^+ the collection M_{g^+} of all models of g^+ induces a lattice ordered by inclusion.*

Proof Trivial: since each model m_{g^+} is defined via inclusion relations (monic arrows), there is a natural ordering of all such models as subobjects of $\hat{g}$. □

Intuitively, each model of g^+ corresponds to the assignment of an independent truth-value to each of the marks of $Marks_{g^+}$. The assignment of the value true (the graph consisting of solely the sheet of assertion) "erases" the mark, whereas the assignment of the value false (the single-cut graph) replaces the mark with a single empty cut. From a philosophical (and not merely logical) point of view, the semantic values of the variable marks are conceivable here as, in a sense, "just more syntax." They do not add new types of syntactical symbols, but simply pick out and exploit formal properties and relations that are already present and available at the specific level of abstraction realized by the presheaves. Their introduction and use is thus somewhat different in this respect from the otherwise similar addition of symbols $\bot$ and $\top$ as sentence constants to many standard logical notations.

We may identify the inclusion lattice with the subset lattice of $Marks_{g^+}$. We have the following proposition:

Proposition 4 *For any graph-with-marks g^+ representing an EG-alpha graph g, every possible assignment of values* **true** *or* **false** *to the variables of g specifies a unique element of the collection M_{g^+} of all models of g^+.*

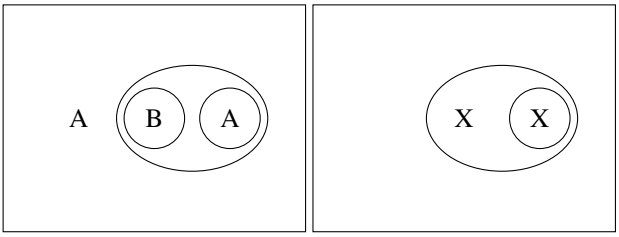
Proof It suffices to show that every function $marks \rightarrow \Omega$ specifies an element of M_{g^+}. There is then a well-defined function $\Omega^{marks} \rightarrow M_{g^+}$. Fix an assignment of values to the variables of the EG graph. Each variable of g takes either value **true** or value **false**. If the former is the case, the variable is replaced by the empty sheet. On the other hand, if the variable takes value **false**, it is replaced by an empty cut. The resulting cuts-only graph will thus be an element of M_{g^+}. □

There is thus a one-to-one correspondence between the elements of M_{g^+} and functions $marks_{g^+} \longrightarrow 2$. If an element $m_{g^+} \in M_{g^+}$ may in this way be conceived as an assignment of one of the values true or false to each variable (that is, each mark) in g^+, then $eval(m_{g^+})$ generates, as value of the function $eval$ the logical evaluation of the graph g^+ as a whole under that particular assignment to each of its variables. The logical payoff here is that what from a standard point of view may be understood as (roughly) the truth-tables for complex propositions in classical logic are neatly and efficiently encoded by tractable mathematical structures that are, on the one hand, already available and present in the presheaf category itself and, on the other hand, readily imaged as diagrams of various types. The philosophical payoff is that upon reflection we may perhaps gain a new perspective on truth-tables themselves, namely as regulated intervals between a lower bound of "all true" and an upper bound of "all false" variable semantic evaluations in which the evaluated variables themselves have the same formal status as the (sole syntactical) logical operator in the system, the cut.

4.2.2 Evaluating Variable-typed Graphs-with-Marks

The semantics described above relies upon the mutual independence of the truth-values of all the marks for any given graph g^+. Thus, just as in the case of the syntactical considerations addressed above in Sect. 4.1.2, here we must take into account the semantics for variable-typed graphs.

To accommodate cases like:



in which variable-types (and not only variable-tokens, or marks) must be tracked in the semantics, we must have recourse to the formal means established above for "identifying" distinct marks, that is, for treating them as tokens of some common type. Here, then, we must make use of the *variable-typing* construction of Sect. 4.1.3 as an essential component of the semantics.

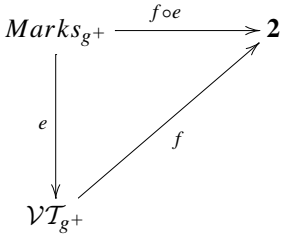
For a generic EG-α graph g^+_{var}, that is, a graph-with-marks g^+ equipped with a variable-typing $\mathcal{VT}_{g^+}$, consider the collection $\mathbf{2}^{\mathcal{VT}_{g^+}}$ of all the functors from $\mathcal{VT}_{g^+}$ into $\mathbf{2}$. Here, $\mathbf{2}$ is the discrete category with exactly two objects, which we interpret as the values $\top$ and $\bot$.[8] These functors are, essentially, all the possible ways we can "evaluate" the marks of the given graph such that they respect the given variable-typing.

The next step is to notice that each of these evaluations induces a model. That is, we can define a map

$$\omega : \mathbf{2}^{\mathcal{VT}_{g^+}} \longrightarrow M_{g^+}$$

[8] Note that any discrete category is, trivially, a groupoid.

that associates to each functor $f : \mathcal{VT}_{g^+} \longrightarrow \mathbf{2}$ (that is, each element of $\mathbf{2}^{\mathcal{VT}_{g^+}}$) its correspondening model $m_{g_f^+}$, which is just an element of M_{g^+}. Recall that the objects of $\mathcal{VT}_{g^+}$ are the elements of the set $Marks_{g^+}$. Thus a functor $f : \mathcal{VT}_{g^+} \longrightarrow \mathbf{2}$ may also be thought of as a function $Marks_{g^+} \longrightarrow \mathbf{2}$ where $\mathbf{2}$, in this latter context, is merely a set rather than a discrete category. Formally, there is an embedding of the discrete category on the set $Marks_{g^+}$ into $\mathcal{VT}_{g^+}$ and the composition $f \circ e$ in the diagram



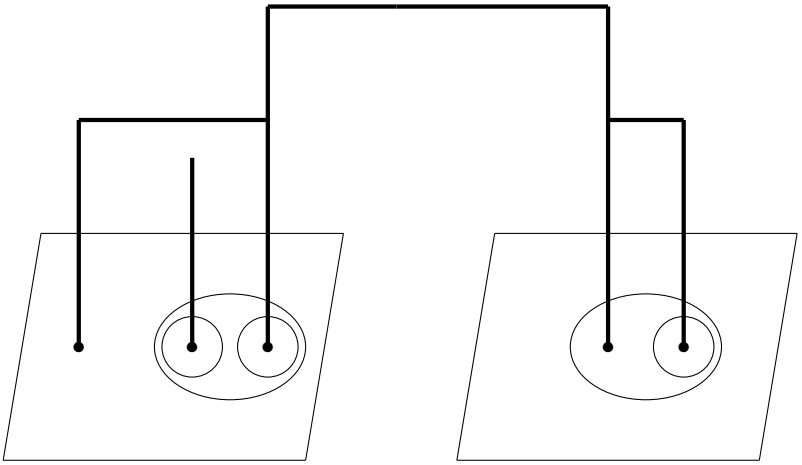
determines a functor from $Marks_{g^+}$ into $\mathbf{2}$. A functor between discrete categories is equivalent to a function between sets. Since functions $Marks_{g^+} \longrightarrow \mathbf{2}$ determine models of g^+, every such $f \circ e$ determines a model of g^+.

5 Truth-Preserving Graph Transformations

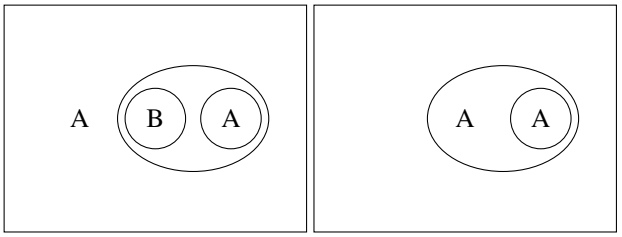
In Peirce's original system of EG-α, the transformation rules provide a sound and complete means for truth-preserving transformations of graphs. For details (including those for EG-β), see Shin (2002) (for alternate presentations, see Burch (1991) and Hammer (1995) ch. 8). For the purposes of the present reconstruction of the EG-α system via generic figures, we set aside the transformation rules provisionally and look to standard model-theoretic reasoning about the graphs to establish truth-preserving relations between generically reconstructed EG-α graphs (objects of the category $\mathcal{EG}_\alpha$).

Essentially, we extend the use of groupoids defined on sets of marks in individual graphs to groupoids defined on sets of marks gathered from collections of graphs. If marks on distinct graphs are understood to represent tokens of some common variable type, then they are connected by a morphism in the extended groupoid construction.

Formally, for some collection of graphs-with-marks $G^{+_i}_{i\in I}$, a variable-typing of $G^{+_i}_{i\in I}$ is a category $\mathcal{VT}_{\mathcal{G}^+}$ whose objects are the elements of $\bigcup Marks_{g_i^+}$ for all $g_i^+ \in G^+$ and with morphisms defined as follows: given two objects X and Y of $\mathcal{VT}_{\mathcal{G}^+}$ there is a morphism $m : X \to Y$ if and only if X and Y are instances of the same variable type. As in the more restricted case above, the reader may easily check that this defines a category and, indeed, a groupoid. Pictorially:

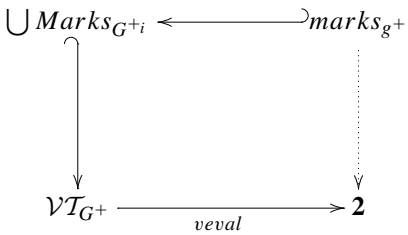


The above corresponds to the identification of A and X in the example of Sect. 4.2.2:



Given a variable typing $\mathcal{VT}_{g^+}$ for some individual graph g^+, a multi-graph variable typing $\mathcal{VT}_{\mathcal{G}^+}$ then respects the "local" typing $\mathcal{VT}_{g^+}$ if and only if there exist a pair of groupoid functors $F : \mathcal{VT}_{g^+} \longrightarrow \mathcal{VT}_{\mathcal{G}^+}$ and $G : \mathcal{VT}_{\mathcal{G}^+} \longrightarrow \mathcal{VT}_{g^+}$ such that GF is the identity on $\mathcal{VT}_{g^+}$. That is to say, if and only if F is a *section* for G, for some such F and G.

A variable evaluation for this extended case involving the marks of more than one graph follows closely the case of a single graph. For a collection of graphs G^+ with a variable-typing $\mathcal{VT}_{\mathcal{G}^+}$, a variable evaluation $veval$ is a functor $\mathcal{VT}_{\mathcal{G}^+} \rightarrow \Omega$. Each $veval$ of this kind generates a $veval$ as restricted to each individual graph, as illustrated by the following commutative diagram:

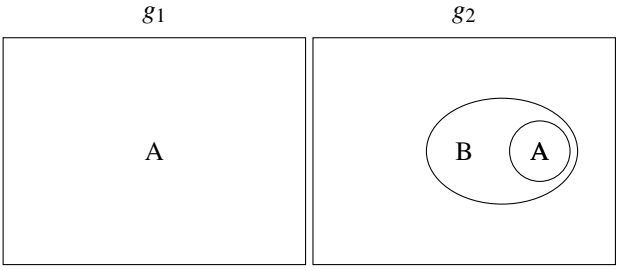


For some given collection of graphs G^+ and a variable-typing $\mathcal{VT}_{\mathcal{G}^+}$, we have the set of all such $veval$ functors.

Given two graphs g_i^+ and g_j^+, g_i^+ logically entails g_j^+ if and only if for all functors $veval$ such that $eval(veval(g_i^+))$ is $\top$ (the empty sheet), $eval(veval(g_j^+))$ is also $\top$ (the empty sheet). In other words, every model (every global evaluation of the variable typing $\mathcal{VT}_{\mathcal{G}^+}$) in which g_i^+ is true is also a model in which $eval(veval(g_j^+))$ is true. The reasoning involved here is model-theoretical: one graph entails another if and only if it is impossible for the former to be true and the latter to be false under any interpretation.

In the case of comparing just two graphs (the usual case for many applications) there is a nice diagrammatic representation of this operation. If we consider the two truth-values $\bot$ and $\top$ as the sole elements of a two-element set, then each evaluation of the variable-types of an ordered pair of graphs gives rise to an ordered pair of elements of $\{\bot, \top\}$. This ordered pair may be conceived as an "arrow" from the first to the second element of the pair. Since every variable-type evaluation induces such an arrow, it is possible to represent the set of all evaluations as a collection of arrows. This can be readily represented as a directed graph.

We can see how this approach allows us to represent the fact that the graph below on the right (g_2) may be derived from the graph on the left (g_1). This fact in Peirce's system corresponds to the truth of the conditional $B \rightarrow A$ (for any B) as derivable from the truth of A in classical logic. Using Peirce's transformation rules to derive g_2 from g_1 involves applying the "write double-cut" rule (around A) followed by "write odd" (scribing B on the one oddly enclosed area). In contrast, we use model-theoretic reasoning in conjunction with variable evaluation functors (each of which is essentially equivalent to a row in a traditional truth-table) to establish the same result.

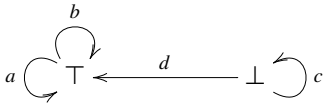


There are four possible evaluations of the variables A and B, which we may notate $A_\top B_\top$, $A_\top B_\bot$, $A_\bot B_\top$, $A_\bot B_\bot$. Each of these induces an evaluation of each of the two graphs, as displayed in the following table:

		g_1	g_2
a	$A_\top B_\top$	$\top$	$\top$
b	$A_\top B_\bot$	$\top$	$\top$
c	$A_\bot B_\top$	$\bot$	$\bot$
d	$A_\bot B_\bot$	$\bot$	$\top$

From the generic figures point of view, this table itself may be understood as a more or less arbitrary notational decision to represent certain objective properties of the cate-

gorical objects corresponding to g_1 and g_2 above in a particular way. The peculiar level of abstraction endemic to presheaf categories lends itself in general to a wide variety of notational expressions that may be chosen to pick out and emphasize whatever salient features are desired for various purposes. Protocols for the intertranslatibility across such different notational expressions, so long as they are appropriately faithful to the categorical structure at issue, are effectively guaranteed by the constraints built into the schematic character of the presheaf category itself. In the present case, the rows of the table may be reconfigured as the arrows of a directed graph with two vertices (labeled $\top$ and $\bot$) in which the tail of each arrow represents the evaluation of g_1 and the head represents the evaluation of g_2.



The fact that no arrows in the directed graph go from $\top$ to $\bot$ represents the fact that g_2 is a logical consequence of g_1.

The representation of the same information in a new and different context, for instance the representation of a truth table as a directed graph, may suggest new ways of handling and interpreting the data. This is one of the advantages and motivations for cultivating various forms of diagrammatic syntax in logic. The generic figures approach via functor categories unifies many otherwise quite different forms of diagrammatic structure. Although the elements of the constructions of Peirce's system of EG-α are distinct from those of directed graphs (cuts, areas, characters on the one hand versus nodes and arrows on the other), both are readily represented using categories of presheaves, and their respective structures may be probed mathematically through the use of representable sets. It is in this way that the use of category theory to investigate the interrelationships and layered generalizations of such diagrammatic environments promises to open up new areas of research in the production and analysis of diagrammatic logical syntax and to contribute to ongoing research programs in diagrammatic and visual reasoning such as those gathered in Moktefi and Shin (2013).

6 Conclusions and Further Work

These results shed light on Peirce's diagrammatic notation and also serve to indicate how categories of presheaves may be used to investigate such diagrammatic systems in general. On the one hand, our reconstruction of Peirce's EG-α system provides a new point of view on his diagrammatic notation by following the "generic figures" approach to presheaf categories as articulated in Reyes et al. (2004). Using this framework, we have defined two categories, $\mathcal{EG}_{\alpha^*}$ and $\mathcal{EG}_{\alpha}$, corresponding to cuts-only EG-α graphs and EG-α graphs with generic, or label-free, variables respectively. Morphisms in both these categories represent syntactical extensions in which new information is "scribed" on the areas of some given graph to produce another more detailed graph. That is, a morphism $g \longrightarrow g'$ in either category represents the "scribing" of (sub)graphs on areas of g so as to generate g'. Thus the structure of both categories corresponds to

the embedding of less detailed EG-α graphs into more detailed ones, in which the former are conceived as syntactic subobjects of the latter. A reduction procedure that transforms any cuts-only graph into either the empty sheet of assertion or a collection of empty cuts served as the basis for developing the semantics for the EG-α graphs in the generic figures context. Elementary groupoid constructions were used to gather otherwise independent variable "marks" (graphical tokens) into equivalence classes corresponding to distinct variable types. Groupoid functors then naturally correspond to semantic evaluations of the variables.

In this way, more generally, Peirce's EG-α provides an illuminating case study of how presheaf categories represent classes of structures by instantiating abstract categories of relations within the more concrete category of sets and functions in all ways possible, given the relational constraints, and how these relative concretizations or realizations by systems of sets and functions are at the same time relatively abstract or schematic with respect to their possible empirical expression in different types of diagrammatic syntax. This intermediate level of abstraction captures what is essential to any such syntax that would faithfully represent the structures at issue, thus providing an explanatory criterion and measure for the specifically iconic aspect of diagrammatic representation. In short, we have shown in the particular case of Peirce's EG-α system how category theory (and specifically the "generic-figures" approach to presheaf categories) can provide a suitable formal environment for representing precisely the iconic syntactical relations structuring a given formal notation that make diagrammatic reasoning with such notation possible. And we have shown thereby how and why the use of these mathematical techniques should apply in similar ways to other types of diagrammatic notation.

This new point of view opens up various lines of potential research. We conclude by listing four open topics for future investigation:

1. **Extending $\mathcal{EG}_\alpha$ to include infinite nestings of cuts** This extension is naturally motivated by noting that relaxing the requirement that each functor take some object in the base category to the empty set in fact simplifies the overall construction even as it enlarges the space of admissible graph-syntax beyond Peirce's own EG-α system. Deep connections with the analyses of "re-entry of the form" in Spencer-Brown (1969) and the application of this formal structure in work such as Luhmann (1995) would be anticipated in connection with this line of research.

2. **Formulating and justifying all of Peirce's original transformation rules in the generic figures context** Any perspicuous formalization of Peirce's rules for deduction requires a straightforward and tractable characterization of, roughly speaking, "subgraph G_s on area a of graph G". The structural nature of the generic figures approach not only facilitates such a characterization but also allows for the integration of this formal characterization with a higher-level representation of the intrinsic ordering of such subgraphs, which for any given graph are naturally ordered by a subobject lattice (indeed, a Heyting algebra) that is itself readily representable as a category. We hope to show in subsequent work how the emergent relations among such subgraph lattices, in coordination with the syntactical/semantical analysis of $\mathcal{EG}_\alpha$ given above, can be shown not only to be capable of *representing* but in fact of naturally *generating* the system of Peirce's

rules (or some generalization of them) as a consequence of inherent features of certain higher-order categorical constructions, something akin to a "transcendental deduction" of the transformation rules. New and perhaps more efficient soundness and completeness proofs for classical propositional logic may result.

3. **Extending the generic figures approach to EG-β and EG-γ** One of the main advantages of Peirce's three-level system of EG is that it provides a unified syntactical framework (with common but increasingly refined notational conventions and deduction rules) for propositional, first-order predicative, and modal and higher-order logics. In particular, the syntactical structure of EG-β preserves the nested cut structure of EG-α while introducing only the new conventions of the "line of identity" to formalize quantification and "spots" or "rhemata" to formalize n-ary relations (Roberts (1973), 47–48). Accordingly, the outline of a possible aligned reconstruction of EG-β becomes visible: the underlying category for EG-β using generic figures must introduce corresponding categorical objects that are suitably integrated into the base category $\overrightarrow{\mathbb{N}}$ and interconnected with one another to reflect the "wirings" of lines of identity with respect to rhemata on any given area and the passage of lines of identity across cuts. Once properly constructed, presheaves over this enhanced category (with suitable refinements) as probed and measured by the induced generic figures would then correspond to well-formed (generic) EG-β graphs just as the objects in $\mathcal{EG}_\alpha$ above correspond to well-formed (generic) EG-α graphs.

4. **Lifting the use of groupoids in the $\mathcal{EG}_\alpha$ semantics to the enriched context of higher-order groupoids and ∞- groupoids** This line of inquiry, especially in coordination with (2) and (3) immediately above, could open potentially fruitful connections between Peirce's graphs and contemporary developments in Homotopy Type Theory and related research programs.

References

Brady, G., & Trimble, T. H. (2000). A categorical interpretation of C.S. Peirce's propositional logic alpha. *Journal of Pure and Applied Algebra*, *49*, 213–239.

Brown, R. (2006). *Topology and groupoids* (3rd ed.). New York: McGraw-Hill. Reprint of *Elements of modern topology*.

Burch, R. (1991). *A Peircean reduction thesis and the foundations of topological logic*. Lubbock, Texas: Texas Tech University Press.

Caterina, G., & Gangle, R. (2013). Iconicity and abduction: a categorical approach to creative hypothesis-formation in Peirce's existential graphs. *Logic Journal of the IGPL*, *21*, 1028–1043.

Caterina, G., & Gangle, R. (2015). The sheet of indication: a diagrammatic semantics for Peirce's EG-alpha. *Synthese*, *192*, 923–940.

Caterina, G., & Gangle, R. (2016). *Iconicity and abduction*. Berlin: Springer.

Hammer, E. M. (1995). *Logic and visual information*. Stanford, CA: CSLI Publications.

Kauffman, L. (2001). Peirce's existential graphs. *Cybernetics and Human Knowing*, *18*, 49–81.

Luhmann, N. (1995). *Social systems*. Stanford, CA: Stanford University Press.

Ma, M., & Pietarinen, A. (2017). Proof analysis of Peirce's alpha system of graphs. *Studia Logica*, *105*(3), 625–647.

Mac Lane, S. (1998). *Categories for the working mathematician*. Berlin: Springer.

Moktefi, A., & Shin, S. (Eds.). (2013). *Visual reasoning with diagrams*. Basel: Springer.

Peirce, C. S., & Eisele, Ed C. (1976). *New elements of mathematics* (Vol. 4). The Hague: Mouton.
Pietarinen, A. (2012). Peirce and the logic of image. *iSemiotica*, *192*, 251–261.
Pietarinen, A., & Bellucci, F. (2016). Existential graphs as an instrument of logical analysis. Part 1: Alpha. *The Review of Symbolic Logic*, *9*(2), 209–237.
Reyes, M., Reyes, G., & Zolfaghari, H. (2004). *Generic figures and their glueings*. Milan (Italy): Polimetrica.
Roberts, D. D. (1973). *The existential graphs of C.S. Peirce*. The Hague: Mouton.
Shin, S. J. (2002). *The iconic logic of Peirce's graphs*. Cambridge MA: MIT Press.
Spencer-Brown, G. (1969). *Laws of form*. London: Allen & Unwin.
Stjernfelt, F. (2007). *Diagrammatology: An investigation on the borderlines of phenomenology, ontology, and semiotics*. Berlin: Springer.
Spivak, D. (2014). *Category theory for the sciences*. Cambridge MA: MIT Press.
Zeman, J. (1974). Peirce's logical graphs. *Semiotica*, *12*, 239–56.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.