AlgebraicJulia blog

Graphs and C-sets IV: The propositional logic of subgraphs and sub-C-sets

C-SETS GRAPHS LOGIC

Nonclassical logics are often thought to be abstruse and exotic, but they arise naturally as the logic of connected spaces. In this post, we introduce the propositional logic of subgraphs, and more generally of sub-C-sets, and illustrate it with computational examples. The nonclassicality of this logic is seen to be not just natural but inevitable, and also surprisingly useful.

AUTHOR
David Jaz Myers and Evan Patterson

PUBLISHED
September 23, 2021

Every proposition is either true or false. This famous tautology is called the *law of excluded middle*. As the words "tautology" and "law" suggest, the law of excluded middle is often taken to be obviously true and attempts to deny it as bizarre and esoteric. But it is easy enough to imagine everyday situations that challenge the obviousness of this principle. Suppose I loiter in the doorway of your office, with one foot in the room and the other in the hallway. Am I in your office, or not? The law of excluded middle says that we must regard exactly one of the two propositions as true, but the choice of which seems arbitrary. Such conundrums are typical of the "logic of space."

In this post, we will see that any graph, and more generally any C-set, can serve as the domain of discourse of a logical system. Unless the schema C satisfies certain special conditions, the logic will be intuitionistic, which means that the principle of excluded middle does not hold. Far from being paradoxical, such logics offer a natural and powerful language to reason about spaces whose parts are connected together.

To understand this post, it will helpful to know the definition of a C-set (Part I) and of a morphism of C-sets (Part III). The textbook by Reyes, Reyes, and Zolfaghari, now out of print but freely available online, is recommended for further reading (M. L. P. Reyes, Reyes, and Zolfaghari 2004). Complete code for similar examples is available in the Catlab documentation.

Propositions as subobjects

Logic supplies rules for arguing that certain propositions hold, given other propositions that are assumed to hold. But not all propositions concern all things; a person might be "good at baseball," but cannot be "prime", and a number can be "prime", but cannot be "good at baseball". Thus, propositions are understood relative to a *domain of discourse*, the set of things of which the proposition may be true or false. Each proposition determines a subset of the domain of discourse, namely the subset of things of which the proposition is true. Conversely, any subset of the domain determines a proposition, the proposition "is in the subset." So, once we have chosen a domain of discourse, propositions become interchangeable with subsets of the domain. Logical connectives then translate into set-theoretic operations; for example, conjunction ("and") and disjunction ("or") of propositions become intersection and union of subsets.

A point of departure for categorical logic is to replace "set" or "domain" with "object in a category" and replace "subset" with "subobject." What is a subobject? Given a category S, a **subobject** of an object $X \in \mathsf{S}$ is a monomorphism into X. A monomorphism in a category S is, loosely speaking, a morphism in S that behaves like an injection in the category of sets. In particular, a subobject of a set X is in S that behaves like an injection in the category of sets. In particular, a subobject of a set X is an injective function into X, whose image picks out a subset of X.[1]

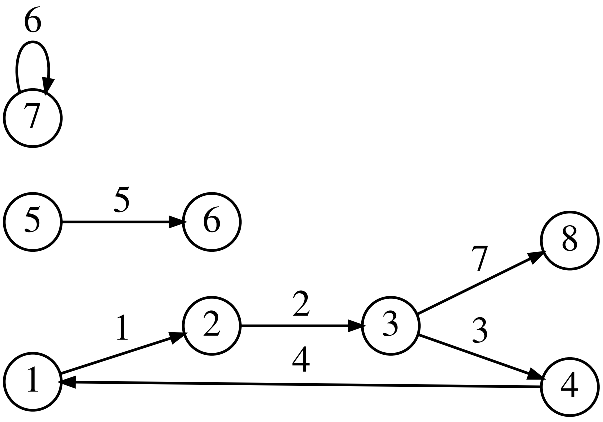
We can now begin to generalize propositional logic from sets to C-sets, for any schema C. Fix a C-set X, which will serve as our domain of discourse. A **sub-C-set** of X is a subobject of X, i.e., a monomorphism into X. As one might expect, a morphism $\phi : A \to X$ of C-sets is monic if and only if every component $\phi_c : A(c) \to X(c)$ is an injective function. Sub-C-sets of X define propositions relative to the domain of discourse X.

Taking C to be the terminal category $\mathbf{1} = \{*\}$ recovers the classical world of sets, but other choices lead to less familiar logics. As usual, our running example will be graphs, where $\mathsf{C} = \{E \rightrightarrows V\}$ is the schema for graphs. A monomorphism is then a graph homomorphism whose vertex and edge maps are injective, and a sub-C-set is a **subgraph**.

Throughout the post, we will take the following graph to be our domain of discourse.

▾ Code

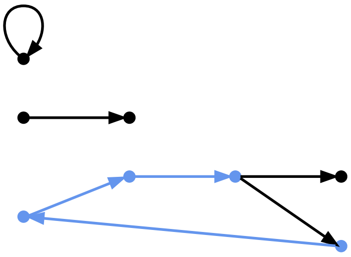
```
using Catlab.Theories, Catlab.CategoricalAlgebra, Catlab.Graphs
using Catlab.Graphics # hide

X = cycle_graph(Graph, 4) ⊕ path_graph(Graph, 2) ⊕ cycle_graph(Graph, 1)
add_edge!(X, 3, add_vertex!(X))
X
```



Here are two typical subgraphs, where membership in the subgraph is indicated by highlighting.

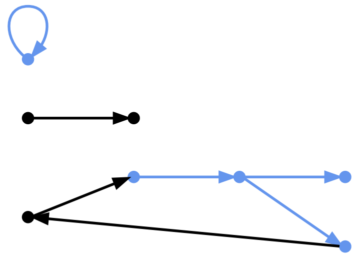
▾ Code

```
A = Subobject(X, V=1:4, E=[1,2,4])
```



▼ Code

```
B = Subobject(X, V=[2,3,4,7,8], E=[2,3,6,7])
```



Notice that an edge can belong to a subgraph only if both its source and target vertices do.

Logical connectives as adjoints

We have claimed that a C-set can serve as a domain of discourse and sub-C-sets as propositions about the domain, but we won't have a logical system until we have a few logical connectives, such as conjunction and disjunction. How should we go about defining them? In classical logic, the logical connectives are usually taken as primitive, given by something like pure intuition, but in our generalized logic, it is not as obvious how to proceed. It was the insight of Lawvere that logical operations can be derived systematically by taking adjoints of still more primitive operations. This procedure can be carried out in any category for which the needed adjoints exist.

To set this up, we need to explain how the subobjects of a given object form a preorder and how to compute adjoints in a preorder. Given an object X in a category S, let $\mathrm{Sub}(X)$ denote the set of all subobjects of X. A morphism from a subobject $A : \mathrm{dom}\, A \to X$ to a subobject $B : \mathrm{dom}\, B \to X$ is a morphism $f : \mathrm{dom}\, A \to \mathrm{dom}\, B$ in S such that $f \fatsemi B = A$. Since B is monic, there exists at most one such morphism f, and since A is monic, f is monic when it exists. With this definition, $\mathrm{Sub}(X)$ becomes a thin category or preorder.[2] We write $A \leq B$ when there exists a (unique) morphism $A \to B$ in $\mathrm{Sub}(X)$. The logical interpretation of $A \leq B$ is that the proposition A implies the proposition B.

Adjointness is a fundamental concept of category theory that is easily understood in the special case of preorders. An **adjunction** between two preorders P and Q consists of a pair of monotone maps $f : P \to Q$ and $g : Q \to P$ such that for any objects $x \in P$ and $y \in Q$, one has $f(x) \leq y$ if and only if $x \leq g(y)$. The "if and only if" statement is sometimes denoted by a vertical bar:

$$\begin{array}{c} f(x) \leq y \\ x \leq g(y) \end{array}$$

One says that f is **left adjoint** to g, or that g is **right adjoint** to f, and writes $f \dashv g$. Left and right adjoints are unique up to isomorphism when they exist; when P is a poset, they are unique without qualification. For more about adjunctions in preorders and posets, (see Fong and Spivak 2018, chap. 1; M. L. P. Reyes, Reyes, and Zolfaghari 2004, chap. 7).

Any preorder P whatsoever supports the basic operations of duplication and deletion. Let $P \times P$ be the cartesian product of P with itself, so that $(x, y) \leq (x', y')$ if and only if $x \leq x'$ and $y \leq y'$, and let $1 = \{*\}$ be the unique preorder with a single element $*$. Then the **diagonal** or **duplication** map let $1 = \{*\}$ be the unique preorder with a single element $*$. Then the **diagonal** or **duplication** map $\Delta_P : P \to P \times P$ is defined by $\Delta_P(x) := (x, x)$, and the **deletion** map $!_P : P \to 1$ is defined by $!_P(x) := *$. Starting from these seemingly trivial operations, we shall derive all the connectives of propositional logic.

Letting $P = \mathrm{Sub}(X)$ be the preorder of subobjects of X, **conjunction** of propositions, a binary operation $\wedge : P \times P \to P$, is defined to be the right adjoint of the duplication map $\Delta_P : P \to P \times P$. For any subobjects A, B, C, we have the equivalences:

$$\begin{gathered} C \leq A \wedge B \\ \Delta_P(C) \leq (A, B) \\ (C, C) \leq (A, B) \\ C \leq A \text{ and } C \leq B \end{gathered}$$

So $A \wedge B \geq C$ if and only if C is a lower bound of A and B, which means that $A \wedge B$ is a *greatest lower bound* or *meet* of A and B. In logical terms, the conjunction of A and B is weakest proposition that implies both A and B.

The **true** proposition, a constant $\top : 1 \to P$, is defined to be the right adjoint of the deletion map $!_P : P \to 1$. For any subobject C, we have:

$$\begin{gathered} C \leq \top \\ !_P(C) \leq * \\ * \leq * \\ \text{true} \end{gathered}$$

So any subobject C is a lower bound of $\top$, meaning that $\top$ is a *maximal* or *top* element. In logical terms, $\top$ is the proposition that is always true, i.e., the weakest of all propositions.

Dually, **disjunction** of propositions and the **false** proposition are defined to be left adjoint to the duplication and deletion maps, respectively, making them the *least upper bound* or *join* and a *minimal* or *bottom* element. In logical terms, the disjunction of A and B is the strongest proposition that is implied by both A and B, whereas $\bot$ is the proposition that is always false, i.e., the strongest proposition.

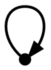
In the logic of sub-C-sets, all four operations have simple formulas. For any $c \in \mathsf{C}$ and $x \in X(c)$, conjunction and disjunction satisfy

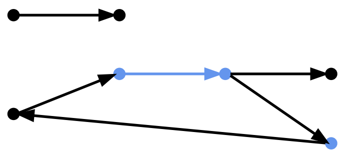
$$\begin{aligned} x \in (A \wedge B)(c) \qquad &\text{iff} \qquad x \in A(c) \text{ and } x \in B(c) \\ x \in (A \vee B)(c) \qquad &\text{iff} \qquad x \in A(c) \text{ or } x \in B(c) \end{aligned}$$

where, by a common abuse of notation, we are identifying a monomorphism into X with its image in X. Similarly, true and false are the full sub-C-set X and the empty sub-C-set, respectively. As a concrete example, the conjunction and disjunction of the two subgraphs A and B defined above are:

▼ Code

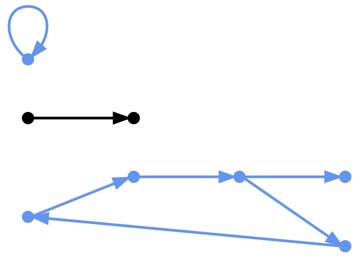
```
A ∧ B
```





▼ Code

```
A ∨ B
```



Implication and negation

The logical operations introduced so far have straightforward pointwise formulas reducing them to classical logic, but things become more interesting when we turn to implication and negation.

The left adjoint of conjunction is, by definition, the diagonal, but we can still ask whether conjunction has a right adjoint. To be more precise, fixing a subobject B, we ask whether conjunction with B, a map $(-) \wedge B : P \to P$, has a right adjoint. The answer is yes whenever P is the subobjects of a C-set. The adjoint operation is called **implication** and is denoted $B \Rightarrow (-) : P \to P$. By definition, implication is characterized by the equivalence

$$\frac{A \wedge B \leq C}{A \leq B \Rightarrow C}$$

for all A, B, C. Logically, $B \Rightarrow C$ is the weakest proposition whose conjunction with B implies C. But recall that $B \leq C$, or the existence of a morphism $B \to C$, already means that "B implies C." Thus, the proposition $B \Rightarrow C$ can be seen as "internalizing" implication within the logic itself. Indeed, implication is an example of an internal hom in a cartesian monoidal category, although that is slightly beyond the scope of this article.

As in classical logic, negation can defined using implication: the **negation** of A is $\neg A := (A \Rightarrow \bot)$. Using the adjunction, we have:

$$\begin{gathered} B \leq \neg A \\ B \leq A \Rightarrow \bot \\ B \wedge A \leq \bot \\ B \wedge A \cong \bot \end{gathered}$$

So $\neg A$ is the largest subobject whose meet with A is empty, or in logical terms, the weakest proposition that is inconsistent with A.

Unlike conjunction or disjunction, implication and negation of sub-C-sets generally cannot be

Unlike conjunction or disjunction, implication and negation of sub-C-sets generally cannot be computed by pointwise Boolean operations. Suppose we tried to define the negation of a subgraph A by including an element in $\neg A$ if and only if it is not included in A. In our running example of a graph X and subgraph A, we would include edge 3 of X in $\neg A$ but not include its source or target vertices—which does not define a valid subgraph.

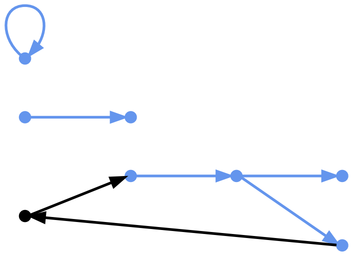
Instead, implication and negation of sub-C-sets satisfy the formulas

$$\begin{aligned} x \in (A \Rightarrow B)(c) \qquad &\text{iff} \qquad x \cdot f \in A(c') \Rightarrow x \cdot f \in B(c') \text{ for all } f : c \to c' \text{ in } \mathsf{C} \\ x \in (\neg A)(c) \qquad &\text{iff} \qquad x \cdot f \notin A(c') \text{ for all } f : c \to c' \text{ in } \mathsf{C} \end{aligned}$$

for every $c \in \mathsf{C}$ and $x \in X(c)$ (M. L. P. Reyes, Reyes, and Zolfaghari 2004, Proposition 9.1.5). The reader is encouraged to write out these formulas explicitly in the case of graphs. As concrete examples, here are $A \Rightarrow B$ and $\neg A$ for the subgraphs A and B defined above:

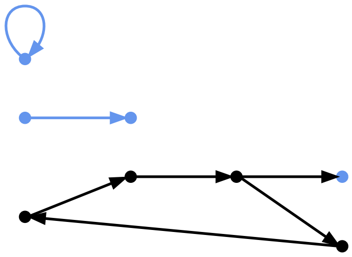
▼ Code

```
A ⇒ B
```



▼ Code

```
¬A
```



Subtraction and complement

Negation of sub-C-sets is conservative in the sense that an element x is included in $\neg A$ just when no element *reachable* from x is included in A. This suggests that there should be a more liberal notion of negation, say $\sim A$, such that an element x is included in $\sim A$ just when x is reachable from any element not in A. We can construct the “other negation” dually to negation: as the two-fold left adjoint to the diagonal map, rather than the two-fold right adjoint.

Fix a subobject B and consider disjunction with B, which is a map $B \vee (-) : P \to P$. When $P = \operatorname{Sub}(X)$ for a C-set X, this map has a left adjoint, called **subtraction** and denoted $(-) \setminus B : P \to P$

$\mathsf{Sub}(X)$ for a C-set X, this map has a left adjoint, called **subtraction** and denoted $(-) \setminus B : P \to P$. Subtraction is characterized by the equivalence

$$\begin{gathered} A \leq B \vee C \\ A \setminus B \leq C \end{gathered}$$

for all A, B, C. In logical terms, $A \setminus B$ is the strongest proposition whose disjunction with B is implied by A. The **complement** of A is then defined by $\sim A := (\top \setminus A)$. By the adjunction, we have:

$$\begin{gathered} \sim A \leq B \\ \top \setminus A \leq B \\ \top \leq A \vee B \\ \top \cong A \vee B \end{gathered}$$

So $\sim A$ is the smallest subobject whose join with A is all of X, or in logical terms, the strongest proposition whose disjunction with A is a tautology.

Following Lawvere, we read $\neg A$ as "not-A" and $\sim A$ as "non-A". For any subobject A, we have $\neg A \leq \sim A$, i.e., the complement of A is at least as large as the negation of A.

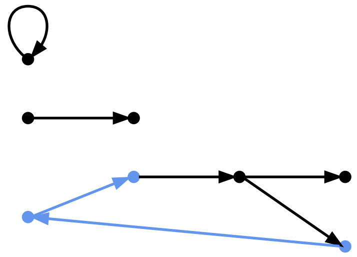
As hinted above, subtraction and complement of sub-C-sets obey the formulas

$$\begin{aligned} x \in (A \setminus B)(c) \qquad &\text{iff} \qquad x' \in A(c') \setminus B(c') \text{ for some } f : c' \to c \text{ in } \mathsf{C} \text{ and } x' \in f^{-1} \cdot x \\ x \in (\sim A)(c) \qquad &\text{iff} \qquad x' \notin A(c') \text{ for some } f : c' \to c \text{ in } \mathsf{C} \text{ and } x' \in f^{-1} \cdot x \end{aligned}$$

for all $c \in \mathsf{C}$ and $x \in X(c)$. In our running example, the subgraphs $A \setminus B$ and $\sim A$ are:

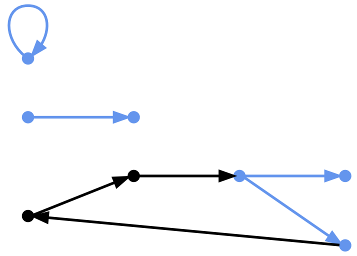
▾ Code

```
A \ B
```



▾ Code

```
~A
```



The principle of excluded middle

The principle of excluded middle says that every proposition is either true or false: that is, $P \vee \neg P$ for any proposition P. What else could it be? But as we saw at the beginning of this post, if there is connectivity between the parts of our domain of discourse—if I can loiter in the doorway of your office, not quite in but not quite out—then the law of excluded middle fails. We can now be more precise about how this happens. A graph satisfies the principle of excluded middle if and only if it is **discrete**, meaning that it has no edges whatsoever. A single subgraph A satisfies $A \vee \neg A = \top$ precisely when it is disconnected from the rest of the graph, meaning that there are no edges connecting A with $\neg A$. Since your office is connected to the rest of the building through the doorway, the proposition “I am in your office” will not satisfy the law of excluded middle.

The two negations of the logic of subgraphs lets us answer the question of what exactly the law of excluded middle is excluding. First, we can dualize the law of excluded middle to use the complement: instead of asking whether $A \vee \neg A = \top$, we could ask whether $A \wedge {\sim} A = \bot$. It turns out that one of these conditions holds for all A if and only the other does. The latter condition, $A \wedge {\sim} A = \bot$, is a nice formulation of the law of excluded middle because it excludes a proposition (asks it to be false). The “middle” being excluded here is the intersection of A and its complement ${\sim} A$. Lawvere calls this the **intrinsic boundary** of A:

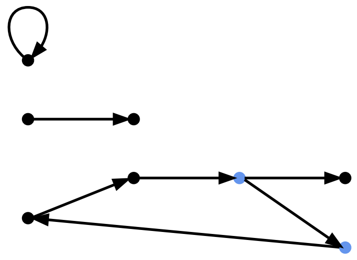
$$\partial A := A \wedge {\sim} A.$$

The law of excluded middle can then be reformulated as: every subobject has an empty boundary.

For graphs, the boundary of a subgraph A consists of all vertices in A which are connected to the outside of A.

▾ Code

```
∂(A) = A ∧ ~A
∂(A)
```



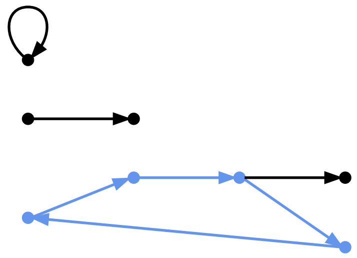
Using the logic of subgraphs

The logic of subgraphs offers a powerful language for constructing new subgraphs from old. The boundary operator is an example. Let us see some other examples.

If A is a subgraph, then $\neg\neg A$ is the subgraph **induced** by A: the subgraph having the same vertices as A but containing all the edges between those vertices.

▼ Code

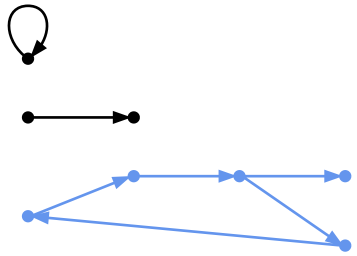
```
C = Subobject(X, V=1:4)
¬(¬C)
```



The expansion and contraction operators are also useful (G. E. Reyes and Zolfaghari 1996). The subgraph $\sim\neg A$ is A but expanded by one degree outward. It includes all the edges that are incident to a vertex in A.

▼ Code

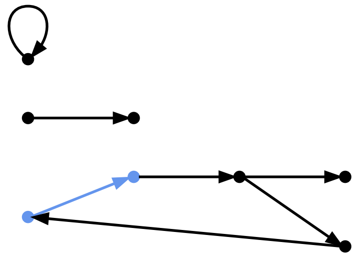
```
~(¬A)
```



Dually, $\neg\sim A$ is A but contracted by one degree, containing exactly the edges between vertices in A that are not connected to the outside of A.

▼ Code

```
¬(~A)
```



Iterating these two operations will expand or contract A until it satisfies $A \vee \neg A = \top$, so that there is nothing left to add or remove because A is disconnected from the rest of the graph.

Conclusion

This post has introduced the propositional logic of a bi-Heyting topos, whose primary example is the topos of graphs or any other topos of C-sets. We have seen that, while it may initially appear exotic, this generalized logic provides a natural language to reason about domains of discourse that exhibit connectivity. The topos of sets, a Boolean topos, exemplifies the very special case of a totally disconnected domain, where the two negations coincide and the principle of excluded middle holds.

The logic explored here is relative to a fixed domain of discourse. What if we wish to change the domain of discourse, say from one graph to another? Pursuing this line of thought ultimately leads to extending the propositional logic of a bi-Heyting topos to a full predicate logic, having existential and universal quantifiers. That will be the topic of a future post.

References

Fong, Brendan, and David I Spivak. 2018. "Seven Sketches in Compositionality: An Invitation to Applied Category Theory."

Johnstone, Peter T. 2002. *Sketches of an Elephant: A Topos Theory Compendium, Volume 1*. Oxford, England: Clarendon Press.

Reyes, Gonzalo E., and Houman Zolfaghari. 1996. "Bi-Heyting Algebras, Toposes and Modalities." *Journal of Philosophical Logic* 25 (1): 25–43. https://doi.org/10.1007/bf00357841.

Reyes, Marie La Palme, Gonzalo E. Reyes, and Houman Zolfaghari. 2004. "Generic Figures and Their Glueings: A Constructive Approach to Functor Categories." In. https://marieetgonzalo.files.wordpress.com/2004/06/generic-figures.pdf.

Footnotes

1. To make a stronger analogy between subsets and subobjects, many authors define a subobject of an object X to be an *isomorphism class* of monomorphisms into X. In practice, one tends to move freely between the two definitions as is convenient (Johnstone 2002, sec. A1.3). ↩
2. If one passes to isomorphism classes of subobjects, the preorder $\mathrm{Sub}(X)$ becomes a poset. ↩