Non-Markov Behaviors

Mendel Keller

September 26, 2018

0 Introduction

Processes occur over intervals of time, and we propose that what's interesting about a process is exactly that which occurs over a longer interval, and cannot be found in the local properties which can be determined point-wise. We propose that this viewpoint can be of value in the sciences, where for example it may be more interesting for a neuroscientist to image the behavior of a neuron over time, and not just instants of time. An individual neuron spike is analogous to a note in a song, which means little on its own, and draws meaning from the wider context in which it appears.

We are able to talk about compositionality by accessing hidden information contained in a graph, using a higher order graph theory. Doing so we manage to capture subtle features of a graph. We accomplish this by embedding the category of graphs into a category of length presheaves, this provides access to new types of possible behaviors that can be exhibited by a graph, where not all forms of composition are always allowed.

By dealing with graphs on the one hand and with temporal type theory on the other, the present work can be seen as bridging the gap between [1] and [2]. It can also serve as a starting point for understanding how to think about temporal type theory. This paper is a simplified form of temporal type theory since we are working in a presheaf topos, so that categorical computations are much simplified.

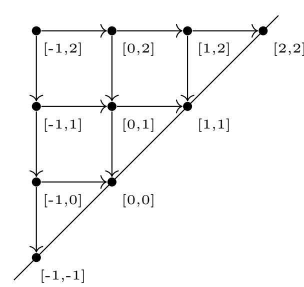
1 Introducing our presheaves

1.1 Interval presheaves

Our first object of study is a poset of integer intervals.

Definition 1. The *interval domain* $\mathbb{IZ}$ has for its objects pairs of integers $d, u \in \mathbb{Z}$ with $d \leq u$, which we denote as $[d, u]$ and there is a morphism $[d, u] \to [d', u']$ whenever $d \leq d' \leq u' \leq u$.

Although we think of the objects of this category as intervals, we reference them as pairs of integers $[d, u]$, which we think of as the endpoints: d being the lower boundary of the interval, and u being the upper endpoint. The objects of this category can be thought of as integer points in the plane that lie on or above the diagonal, with arrows being exactly those going in the downwards or rightwards direction. A picture of a part of this category around the origin would look like (1).



(1)

1.1.1 Presheaf categories

A presheaf[1] on a category $\mathcal{C}$ is a functor $F : \mathcal{C} \to \mathsf{Set}$, which assigns to each object $X \in \mathcal{C}$ a set $F(X)$, and to each morphism $f : X \to Y$ a function $F(f) : F(X) \to F(Y)$. In our case, we will be looking at presheaves on $\mathbb{IZ}$, and we think of a presheaf as a type of behavior. We then get a presheaf category:

[1] Technically these are called copresheaves, and presheaves are actually contravarient functors from $\mathcal{C}$ to Set, but we'll just be calling these presheaves, and they can be thought of as presheaves on $\mathcal{C}^{op}$.

Definition 2. The category of presheaves $\widehat{\mathcal{C}}$ on a category $\mathcal{C}$ has for its objects presheaves $\mathcal{C} \to \mathsf{Set}$ and for its morphisms natural transformations between presheaves.

1.1.2 Example Presheaves

One simple type of presheaf that can always be considered is a constant presheaf. A constant presheaf C_S sends everything in the base category, in our case $\mathbb{IZ}$, to the same set S, and sends every morphism to the identity of S. In fact, the constant presheaves give rise to a functor $C : \mathsf{Set} \to \widehat{\mathbb{IZ}}$ which acts by sending a set S to the constant presheaf C_S.

Theorem 3. *The functor $C : \mathsf{Set} \to \widehat{\mathbb{IZ}}$ mapping a set S to the constant presheaf at S is fully faithful.*

Proof. A natural transformation $C_S \to C_T$ must act by the same morphism $S \to T$ for every $[d, u]$, since C_S and C_T both send every morphism to the respective identity. Therefore a natural transformation is given by a function $S \to T$, and every function $S \to T$ determines a natural transformation. ■

Example 4. Another type of presheaf to consider on $\mathbb{IZ}$ is one which sends an interval $[d, u]$ to the set of functions from the set containing integer points of $[d, u]$ to some specified set S, we denote this sort of presheaf by F_S. More precisely, we make use of a function $I : \mathrm{Ob}(\mathbb{IZ}) \to \mathsf{Set}$ which sends an interval $[d, u] \in \mathbb{IZ}$ to its realization as a set $\{x \in \mathbb{Z} : d \leq x \leq u\}$. Then we define $F_S[d, u] := Hom(I[d, u], S) = \{f : I[d, u] \to S\}$, here a morphism $i : [d, u] \to [d', u']$ gets sent to the restrictions of functions, that is to the function $F_S(i) : F_S[d, u] \to F_S[d', u']$ which sends a function $(f : I[d, u] \to S) \in F_S[d, u]$ to the function $(f|_{I[d',u']} : I[d', u'] \to S) \in F_S[d', u']$.

prove'm The natural transformations from some F_S to a F_T are allowed to be determined pointwise on $\mathbb{Z}$. Namely a function $f : S \to T$ is chosen for every $z \in \mathbb{Z}$, so that a natural transformation is given by a function $\alpha : \mathbb{Z} \times S \to T$. The natural transformation given by α sends a function $f : I \to S$ to the function $\alpha(x, f(x)) : I \to T$. We leave it to the reader to find the natural transformations between a C_S and some F_T, and similarly from a F_S to some C_T.

1.1.3 Limits and colimits

In $\widehat{\mathbb{IZ}}$, and in fact in any presheaf category, limits and comlimits are computed pointwise. Formally, $(\lim_{i \in I} F_i)(X) = \lim_{i \in I} F_i(X)$, so that we can simply compute limits in Set, which makes their computation straightforward. For example, the product $F \times G$ of two presheaves is given by their pointwise product $(F \times G)(X) = F(X) \times G(X)$, and similarly for coproducts $(F \coprod G)(X) = F(X) \coprod G(X)$ the disjoint union. This also means that the terminal object 1 of a preasheaf category maps everything in the base category to the terminal object of Set, so that the terminal object in $\mathbb{IZ}$ is the constant presheaf C_1 mapping everything to the one element set, and dually the initial object in a preasheaf category is $C_\emptyset$ which maps every object to the empty set.

1.2 Length presheaves

Another category to consider is one we call Int, which is like a translation invariant version of $\mathbb{IZ}$. Int can be constructed as the twisted arrow category of $\mathbb{N}$, when $\mathbb{N}$ is viewed as free monoid on one generator, represented as a category with a single object.

1.2.1 Other Presentations

Definition 5. A more explicit description of Int is that it has as its objects the set of natural numbers $\{0, 1, 2, \ldots\}$, and as its morphisms $n \to m$ pairs of natural numbers (r, s) such that $r + m + s = n$.

We can think of the morphisms from n to m as the ways in which m can be inserted into n, or equivalently as the ways in which a length n interval can be restricted to a length m interval. In light of $\mathbb{IZ}$, we can think of $m \in \mathrm{Ob}(\mathsf{Int})$ as an integer interval of length m, and it can fit into a longer interval of length n at several different integer distances from the endpoints.

Another useful way to think of Int is as the free category obtained from the diagram:

$$0 \underset{(1,0)}{\overset{(0,1)}{\leftleftarrows}} 1 \underset{(1,0)}{\overset{(0,1)}{\leftleftarrows}} 2 \underset{(1,0)}{\overset{(0,1)}{\leftleftarrows}} 3 \underset{(1,0)}{\overset{(0,1)}{\leftleftarrows}} \cdots \tag{2}$$

where we impose the relations $(0,1) \circ (1,0) = (1,0) \circ (0,1)$ at every stage. In this view, any morphism (r, s) can be given by following (1,0) r many times, and then (0,1) s many times.

1.2.2 Example Length Presheaves

We now can consider the presheaf category $\widehat{\mathsf{Int}}$, which we will be focusing our attention on. One example presheaf to consider is the one which sends n to the set $\{n, n+1, n+2, \ldots\}$ and a morphism (r, s) sends each x to $x - r$. Another possible preheaf will send each n to the set $\{0, 1, 2, 3, 4\}$ and (r, s) will send x to $x + r \mod 5$. We can also construct an Int analogue to example 4:

Example 6. A presheaf F_S sends n to the set of functions $\{\{0, 1, \ldots, n\} \to S\} \cong S^{n+1}$ and the morphisms $(r, s) : n \to m$ send a function $f : \{0, \ldots, n\} \to S$ to the function $g : \{0, \ldots, m\} \to S$ defined by $g(x) = f(x + r)$.

The fact that we've defined these presheaves in a way that the morphisms only need pay attention to r and can ignore s is no accident, since from any three of n, m, r, s we can find the fourth, using the equation $r + m + s = n$.

1.3 Time

Of particular interest in temporal type theory is the presheaf $\mathtt{Time} : \mathsf{Int} \to \mathsf{Set}$ which sends n to the set of pairs $\{(a, a + n) : a \in \mathbb{Z}\}$ and morphisms (r, s) send $(a, a + n)$ to $(a + r, a + n - s)$.

One thing we can do with $\mathtt{Time}$ is demonstrate a relationship between $\widehat{\mathbb{IZ}}$ and $\widehat{\mathsf{Int}}$. Recall that for a category $\mathcal{C}$ and object X in $\mathcal{C}$ the *slice category* $\mathcal{C}/X$ has its objects morphisms $Y \to X$ into X in $\mathcal{C}$, and as its morphisms $h : (f : Y \to X) \to (g : Z \to X)$ it has morphisms $h : Y \to Z$ in $\mathcal{C}$ such that the diagram

$$\begin{array}{ccc} Y & \xrightarrow{\;h\;} & Z \\ & {\scriptstyle f}\searrow \quad \swarrow{\scriptstyle g} & \\ & X & \end{array} \tag{3}$$

commutes. In our case, we consider the slice category $\widehat{\mathsf{Int}}/\mathtt{Time}$. In particular, there is a pair of inverse functors $\Phi : \widehat{\mathbb{IZ}} \to \widehat{\mathsf{Int}}/\mathtt{Time}$ and $\Psi : \widehat{\mathsf{Int}}/\mathtt{Time} \to \widehat{\mathbb{IZ}}$.

clunky notation needs editing, watch out for ops The functor $\Phi : \widehat{\mathbb{IZ}} \to \widehat{\mathsf{Int}}/\mathtt{Time}$ sends a presheaf $(X : \mathbb{IZ} \to \mathsf{Set}) \in \widehat{\mathbb{IZ}}$ to the Int presheaf $\Phi(X)(n) := \coprod_{a \in \mathbb{Z}} X([a, a+n])$ sending n to the disjoint union of all the sets which X sends length n intervals to. $\Phi(X)$ sends morphisms $(r, s) : n \to m$ to morphisms $\Phi(X)((r, s)) : \coprod_{a \in \mathbb{Z}} X([a, a+n]) \to \coprod_{a \in \mathbb{Z}} X([a, a+m])$ which send $x \in X([a, a+n])$ to $y \in X([a+r, a+n-s])$ where $y = X([a, a+n] \to [a+r, a+n-s])(x)$, thus $\Phi(X)((r, s))$ acts component wise as X would on the maps $[d, u] \to [d + r, u - s]$. We also obtain a morphism $\Phi(X) \to \mathtt{Time}$ which sends $x \in \Phi(X)(n)$ to the $(a, a + n)$ from which we obtained it. Now $\Psi : \widehat{\mathsf{Int}}/\mathtt{Time} \to \widehat{\mathbb{IZ}}$ taken an Int presheaf X, and a morphism f from X to $\mathtt{Time}$, and returns the $\mathbb{IZ}$ presheaf $\Psi(X, f)$ which takes $[d, u]$ to $f^{-1}([d, u])$ and takes a morphism $[d, u] \to [d', u']$ to $X((d - d', u' - u))$ restricted to $[d, u]$.**watch pharentheese for f^{-1}, watch out for ops**

2 Subobject classifier

A *subobject* of an object X in a category $\mathcal{C}$ is an object Y along with a *monomorphism* $i : Y \to X$. A monomorphism is the categorical generalization of an injective function, which in the case of a presheaf topos $\mathcal{C} = \widehat{\mathcal{D}}$ is the condition that the natural transformation $i : Y \to X$ provides an injective function $i(d) : Y(d) \to X(d)$ for every $d \in \mathrm{Ob}(\mathcal{D})$.

Every topos has an object Ω, called the *subobject classifier*, which classifies subobjects of any object in the topos. The subobject classifier comes equipped with a morphism $\mathsf{True} : 1 \to \Omega$, such that for every subobject Y of an object X there is a unique characteristic morphism $\theta_Y : X \to \Omega$ such that

$$\begin{array}{ccc} Y & \xrightarrow{!} & 1 \\ {\scriptstyle i}\downarrow & \lrcorner & \downarrow{\scriptstyle \mathsf{True}} \\ X & \xrightarrow{\theta_Y} & \Omega \end{array}$$

is a pullback diagram. In this way, Ω classifies the subobjects of X. In particular we have that $\mathrm{Hom}(X, \Omega) \cong \mathrm{Sub}(X)$, the subobjects of X correspond exactly to morphisms $X \to \Omega$.

This is particularly interesting in type theory, where maps $X \to \Omega$ into the subobject classifier Ω can be understood as propositions about the domain X. This can be understood in the context of Set, where subobjects are subsets, and a proposition p produces a subset $Y = \{x|p(x)\}$. In our case, the subobject classifier will allow us to discuss the composition of processes into larger "superprocesses". We now set out to find the subobject classifier in $\widehat{\mathbb{IZ}}$ and $\widehat{\mathsf{Int}}$, and to determine that we will introduce representable functors and Yoneda's lemma.

2.1 Representable Functors

For any presheaf category, one natural and important type of presheaf to consider is the representable presheaves. A representable presheaf yX is determined by some object X of the base category $\mathcal{C}$, and sends an object Y to the set of morphisms from X into Y.

Definition 7. For an object $X \in Ob(\mathcal{C})$ the *representable presheaf* of X is the functor $yX = Hom(X, -) : \mathcal{C} \to \mathsf{Set}$ which maps any $Y \in Ob(\mathcal{C})$ to $yX(Y) = Hom(X, Y)$, the set of morphisms from X to Y.

Note that given a morphism $f : Y \to Z$, we obtain a function $yX(f) : Hom(X, Y) \to Hom(X, Z)$ which maps $g \in Hom(X, Y)$ to $f \circ g \in Hom(X, Z)$. Representable presheaves are the star players in the Yoneda lemma.

The Yoneda lemma tells us that the value a presheaf F takes on an object X is isomorphic to the hom object from the representable yX to F. Simply stated, $F(X) \cong \mathrm{Hom}(yX, F)$. Thus, in order to determine the value the subobject classifier Ω takes on some object X, we need only find $\mathrm{Hom}(yX, \Omega)$, but we've also said that $\mathrm{Hom}(yX, \Omega) \cong \mathrm{Sub}(yX)$. So we can find Ω by finding the subobjects of representable functors, and to do so we introduce the category of elements.

2.2 Category of elements

For a presheaf F, the category of elements of F is a way of representing F itself as a category.

Definition 8. The *category of elements* $\mathrm{el}(F)$ of a functor $F : \mathcal{C} \to \mathsf{Set}$ has for its objects pairs (x, X) where $X \in Ob(\mathcal{C})$ and $x \in F(X)$ and has morphisms $(x, X) \to (y, Y)$ induced by morphisms $f : X \to Y$ which satisfy $F(f)(x) = y$.

The category of elements can be used to give us a neat description for subobjects of a presheaf.

The category 2 is the interval category which has two objects $0, 1$ and a single nontrivial morphism $0 \to 1$.

Theorem 9. *For any presheaf* $F : \mathcal{C} \to \mathsf{Set}$ *there is an equivalence of categories* $[\mathrm{el}(F), 2] \cong \mathrm{Sub}(F)$

Proof. As explained above, a subobject of a presheaf $F \in \widehat{\mathcal{C}}$ is another presheaf G with a natural transformation $\eta : G \to F$ such that the map $\eta(X) : G(X) \hookrightarrow F(X)$ is an injection for every $X \in \mathcal{C}$.

Given a functor $\alpha : \mathrm{el}(F) \to 2$ we obtain a subobject G of F given by $G(X) = \{x \in F(X)|\alpha(x) = 1\} \subseteq F(x)$ with the natural transformation $i : G \hookrightarrow F$ simply being given by the inclusion maps. This map is natural (which in this case is verification that $G(Y)$ contains the images $y \in F(Y)$ of elements $x \in G(X)$ under function $F(f) : F(X) \to F(Y)$ arising from maps $f : Y \to X$ in $\mathcal{C}$), since for any object $X \in \mathcal{C}$, element $x \in G(X)$ and map $f_x : (x, X) \to (y, Y)$ inside $\mathrm{el}(F)$ we have that $\alpha(f_x) = \mathrm{id}_1$ since this is the only arrow in 2 with domain 1, so then we have that $\alpha(y, Y) = 1$ and this gives that $F(f)(x) \in G(Y)$ and so $F(f) \circ i = i \circ G(f)$ which is the desired naturality.

For the other direction, a subobject G can be thought of as some full subcategory of $\mathrm{el}(F)$, namely the subcategory which is $\mathrm{el}(G)$. The corresponding morphism $\alpha : \mathrm{el}(F) \to 2$ is simply given by taking things in $\mathrm{el}(G)$ to 1 and things which aren't to 0. This will work i.e. we never need a morphism $1 \to 0$, because if some $x \in F(X)$ is in $G(X)$ then $F(f)(x)$ must be in $G(Y)$ for any morphism $f : X \to Y$, be naturality.

Lastly, a natural transformation $\eta : \alpha \to \beta$ between functors $\alpha, \beta : \mathrm{el}(F) \to 2$ has 3 possible values for some (x, X), namely $\eta(x, X) : \alpha(x, X) \to \beta(x, X) = \mathrm{id}_0, \mathrm{id}_1, 0 \to 1$ as these are the morphisms of 2. These give three cases: (1) $x \notin S_\alpha(X)$ and $x \notin S_\beta(X)$ (2) $x \in S_\alpha(X)$ and $x \in S_\beta(X)$, or (3) $x \notin S_\alpha(X)$ but $x \in S_\beta(X)$. This tells us that $S_\alpha(X) \subset S_\beta(X)$ for every X so that S_α is a subobject of S_β. ∎

We now know that the subobject classifier is defined by $\Omega(X) \cong \mathrm{Ob}([yX, 2])$, and a restriction map $\Omega(f)$. **finish sentence**

We now look at the relevant categories of elements.

2.2.1 Interval Representables

The category of elements of a representable presheaf in $\mathbb{IZ}$ looks like a pyramid. For example, $y[0, 2]$ can be represented by its category of elements like so:

$$\begin{array}{ccccc} [0,2] & \to & [1,2] & \to & [2,2] \\ \downarrow & & \downarrow & & \\ [0,1] & \to & [1,1] & & \\ \downarrow & & & & \\ [0,0] & & & & \end{array} \tag{4}$$

Where we denote an object of $\mathrm{el}(y[0, 2])$ simply by the object of $\mathbb{IZ}$ it is associated with since $\mathrm{Hom}([d, u], [d', u'])$ contains a single element at most. Some example subobjects of $y[0, 2]$ can be represented as so:

$$\begin{array}{ccccc} \color{gray}{[0,2]} & \color{gray}{\to} & \color{gray}{[1,2]} & \color{gray}{\to} & [2,2] \\ \color{gray}{\downarrow} & & \color{gray}{\downarrow} & & \\ \color{gray}{[0,1]} & \color{gray}{\to} & [1,1] & & \\ \color{gray}{\downarrow} & & & & \\ [0,0] & & & & \end{array} \qquad \begin{array}{ccccc} \color{gray}{[0,2]} & \color{gray}{\to} & [1,2] & \to & [2,2] \\ \color{gray}{\downarrow} & & \downarrow & & \\ [0,1] & \to & [1,1] & & \\ \downarrow & & & & \\ [0,0] & & & & \end{array} \qquad \begin{array}{ccccc} \color{gray}{[0,2]} & \color{gray}{\to} & \color{gray}{[1,2]} & \color{gray}{\to} & [2,2] \\ \color{gray}{\downarrow} & & \color{gray}{\downarrow} & & \\ [0,1] & \to & [1,1] & & \\ \downarrow & & & & \\ [0,0] & & & & \end{array}$$

Where we display the objects of $\mathrm{el}(y[0, 2])$ which are mapped to 1 in black, and the ones which are mapped to 0 in gray.

2.2.2 Length Representables

The situation for Int is analogous, for example we can find a picture for $y2$ which mirrors (4) like so:

$$\begin{array}{ccccc} <\mathrm{id}_2, 2> & \xrightarrow{(1,0)} & <(1,0), 1> & \xrightarrow{(1,0)} & <(2,0), 0> \\ {\scriptstyle (0,1)}\downarrow & & {\scriptstyle (0,1)}\downarrow & & \\ <(0,1), 1> & \xrightarrow{(1,0)} & <(1,1), 0> & & \\ {\scriptstyle (0,1)}\downarrow & & & & \\ <(0,2), 0> & & & & \end{array} \tag{5}$$

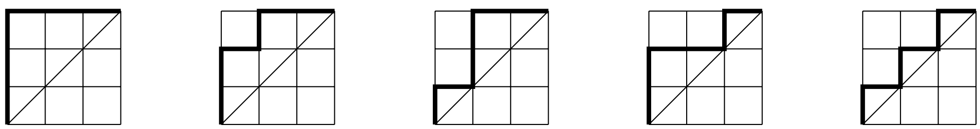
This category of elements analysis shows that the subobject classifier Ω of $\widehat{\mathbb{IZ}}$ is essentially identical to that of $\widehat{\mathsf{Int}}$, indicating that their logic systems are quite similar. This shouldn't be too surprising, as Int is just a translation invariant version of $\mathbb{IZ}$.

There is however some a little bit of complication with Int, since in this case there can be a few morphisms $n \to m$. So while in the $\mathbb{IZ}$ case it suffices to simply look at the objects themselves, here a subobject is understood as looking at objects together with their morphisms, as we have shown in diagram (5). It is therefore worth keeping in mind and then immediately forgetting that for Int we will always have a specific morphism in mind, while in $\mathbb{IZ}$ we don't have to, since there's always at most one choice of morphism.

While it may be possible to find the subobjects of such a representable explicitly, we're still presented with the issue of finding a concrete description of the subobject classifier in $\widehat{\mathbb{IZ}}$ and $\widehat{\mathsf{Int}}$. It turns out that we find such a description in Dyck paths.

2.3 Dyck paths

Dyck paths exactly describe the sort of thing we're looking for. A Dyck path is a path of edges in a $n \times n$ square grid which climbs from the lower left corner to the upper right corner, while only ever moving upwards or to the right, and lies entirely above the diagonal. For example the five Dyck paths on a 3×3 grid are:



(6)

and these exactly correspond to subobjects of $y[0,1]$, this can be seen by thinking of the three cells of the grid in the upper left corner which lie entirely above the diagonal as representing the category of elements of $y[0,1]$. So that the above Dyck paths mirror the subobjects of $y[0,1]$ like so:

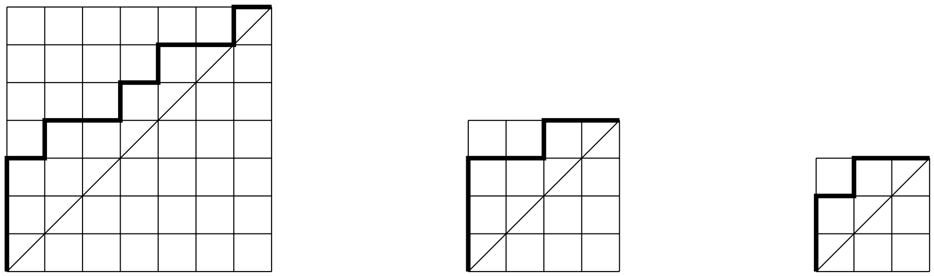
$$\begin{array}{ccccc} \mathbf{[0,1]} \to \mathbf{[1,1]} & [0,1] \to \mathbf{[1,1]} & [0,1] \to \mathbf{[1,1]} & [0,1] \to [1,1] & [0,1] \to [1,1] \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \mathbf{[0,0]} & \mathbf{[0,0]} & [0,0] & \mathbf{[0,0]} & [0,0] \end{array}$$

From these pictures we can see a correspondence between the Dyck path p on a 3×3 grid to the morphism $\mathrm{el}(y[0,1]) \to 2$ that sends the objects of $\mathrm{el}(y[0,1])$ corresponding to the cells under the path p to 1, and everything else to 0. So that in the order appearing in (6), the corresponding subobjects are: $y[0,1]$ itself, the subobject which acts like $y[0,1]$ with the exception of mapping $[0,1]$ to the empty set, the subobject sending only $[1,1]$ to a nonempty set, the subobject sending only $[0,0]$ to a nonempty set, and the subobject sending everything to the empty set (this is the initial presheaf $C_\emptyset$).

The idea then is that the section above the path is sent to 0, and the section below is sent to 1. So that the conditions for being a Dyck path on an $n \times n$ grid are exactly analogous to that of being a subobject of $y[d,u]$ where $n = u - d + 2$, with the connection between the two given via a morphism $\mathrm{el}(y[d,u]) \to 2$.

To find the action of a restriction map $\Omega([d,u] \to [d',u']) : \Omega([d,u]) \to \Omega([d',u'])$, we focus the appropriate subgrid of the grid for $[d,u]$ of size $u' - d' + 2$ lying on the diagonal and containing the cell corresponding to $[d',u']$, and assign to a Dyck path p on the $[d,u]$ grid the Dyck path q on the $[d',u']$ grid which has below it exactly those cells which p has below it when restricted to this subgrid. This induces the behavior of sending a $y[d,u]$ subobject F to the $y[d',u']$ subobject G which sends intervals $[a,b]$ with $d' \leq a \leq b \leq u'$ to a one element set exactly if F does, which is exactly what we might expect.

We have an exact analogue for $\widehat{\mathsf{Int}}$. The subobjects of yn are the Dyck paths on a grid of size $(n+2)\times(n+2)$, and the restriction maps behave as described above. For example, the morphism $(1,2) : 5 \to 2$ will send the dyck path on the left to the one in the middle, and then further applying $(1,0) : 2 \to 1$ will send it to the one on the right:

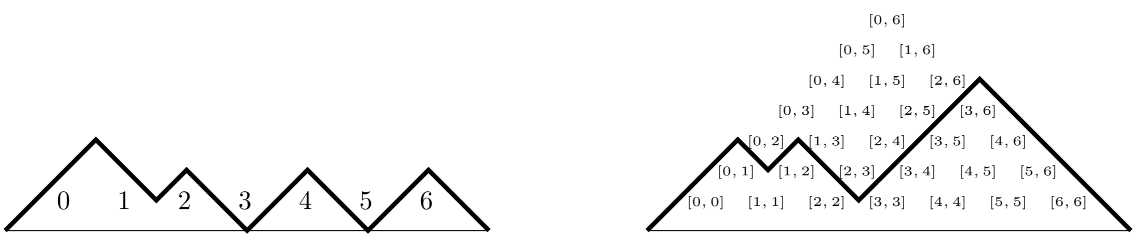


(7)

We will from here on focus on the Int side of things. We will also generally not distinguish between a subobject and the corresponding morphism into Ω, or between a Dyck path, a functor from the category of elements to the interval category, and a subobject of a representable presheaf.

2.4 Understanding Dyck Paths

A much more enlightening perspective on Dyck paths as classifying interval and length presheaf subobjects is obtained by rotating our picture by 45 degrees. So that for example we might get a Dyck path of size 8 that looks like one of the pictures in fig 8, the one on the left is presented as an Int Dyck path, and that on the right as an $\mathbb{IZ}$ Dyck path.



(8)

In this view we can place the paths above a numberline representing a start and end point, and then have height correspond to length. This is a far more natural and intuitive way to interpret and present a Dyck path in the context of studying presheaves on a category of intervals. In this way a point (x, y) corresponds to the interval $[x - y, x + y]$ it should be noted that this presentations makes us include half integer points to represent intervals of odd length. The $\mathbb{IZ}$ picture is rooted in a more absolute context than the Int one, as objects of $\mathbb{IZ}$ have a definite location on the numberline. It is therefore generally easier to imagine Dyck paths as lying over $\mathbb{Z}$ and think of them as if they were classifying $\mathbb{IZ}$ presheaves even when working with Int, so that one can think of an element of $\Omega(n)$ as being an element of $\Omega([0, n])$.

We can also imagine global $\mathbb{IZ}$ Dyck paths, defining a morphism $1 \to \Omega$. Such a global path will be given by some infinite zigzag lying above the numberline $\mathbb{Z}$. On $[d, u]$ the morphism $1 \to \Omega$ is then given by sending $* \in 1[d, u]$ to the highest possible Dyck path over $[d, u]$ which lies under the global Dyck path.

2.5 Classifying subobjects

Now that we've found Ω, we are left with the task of interpreting its function. First of all, the morphism $\mathsf{True} : 1 \to \Omega$ simply sends the one element $* \in 1(n)$ to $yn \in \Omega(n)$. Correspondingly, the morphism $X \to \Omega$ sending every $x \in X(n)$ to $yn \in \Omega(n)$ is the one which produces X as a subobject of itself.

Now, some proposition p may be false about some $x \in X(n)$, but still true on some restriction of x. For example, if we take the presheaf F_S, we might have a morphism $p : F_s \to \Omega$ which in effect states that a function $[f : \{0, \dots, n\} \to S] \in F_S(n)$ doesn't take on the value $s \in S$ for any input. Now while this may be false for a function f itself, unless f is identically s some restriction map (r, s) applied to f will produce a function $(r, s)f = g \in F_S(m)$ for which the statement is true. In this case, $p(f)$ will not be the subobject of yn which sends everything to the empty set, but rather it will be such that when we apply (r, s) we will obtain ym. This also means that the corresponding subobject of F_S will not include f, but will include g.

Notice that statements like "there is an input for which f takes on the value s" aren't valid, because even if f hits s, restrictions of f needn't. So if we cannot allow that sort of statement because if a subobject of F_S includes a function f, it also needs to include where f lands under a restriction map. So we can only allow statements which remain true under restrictions. We will elaborate on this and provide some more examples in section 4.2. Before we really get into the wealth provided by the subobject classifier, it will serve us well to introduce graph presheaves.

3 Graph-paths

First, to avoid any ambiguity, we define what we'll mean by a graph. We have in mind directed graphs, and we allow self loops, as well as multiple edges between a pair of vertices.

Definition 10. A graph G is a pair of sets (E, V) equipped with a pair of functions $s, t : E \to V$.

We can also view a graph as a presheaf on the category G which consists of the two objects E, V and two nontrivial morphisms $s, t : E \to V$, so a graph G is a functor $G : \mathsf{G} \to \mathsf{Set}$. The interpretation of this definition to keep in mind is that E is a set of edges, V is a set of vertices, s takes an edge to its source, and t takes an edge to its target. So a graph G is then simply a quadruple (E, V, s, t). Accordingly, we can define Grph, the category of graphs, in the natural way.

Definition 11. The category Grph has graphs (E, V, s, t) as its objects, and has its morphisms $f : (E, V, s, t) \to (E', V', s', t')$ given by pairs of functions $\alpha_f : E \to E'$ and $\beta_f : V \to V'$ such that the two diagrams:

$$\begin{array}{ccc} E & \xrightarrow{s} & V \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\beta} \\ E' & \xrightarrow{s'} & V' \end{array} \qquad \begin{array}{ccc} E & \xrightarrow{t} & V \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\beta} \\ E' & \xrightarrow{t'} & V' \end{array} \tag{9}$$

commute.

As can be easily seen, Grph is equivalent to the presheaf category on G. This then tells us that a graph homomorphism maps vertices to vertices and edges to edges in such a way that an edge follows the vertices adjacent to it, which is the sort of natural construction that would be expected, this naturality coming from the naturality of the transformation between presheaves.

3.1 Enriching graphs

An important type of Int-presheaf, and the focus of the remainder of this paper, is that induced by a graph. Namely, we choose a graph G, and send a natural number n to the paths of length n in G. This will give us an Int presheaf which is the right kan extension of the graph $G : \mathsf{G} \to \mathsf{Set}$ along the inclusion $\mathsf{G} \to \mathsf{Int}$.

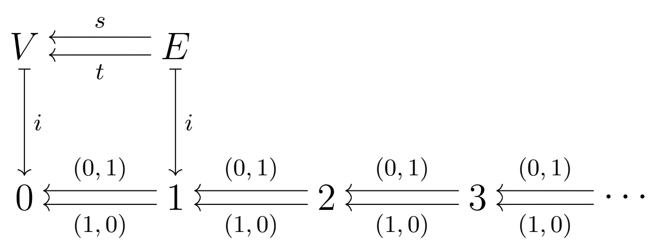
3.1.1 Kan extensions

A *kan extension* is a way of extending a functor $p : \mathcal{C} \to \mathcal{E}$ to a functor $\tilde{p} : \mathcal{D} \to \mathcal{E}$ along a functor $F : \mathcal{C} \to \mathcal{D}$. This should be imagined as extending the domain of p, this will be especially true in our case, where the F is an inclusion of categories. We will be considering a right kan extension, but left kan extensions also exist in general, and are dual to the right extension. In our case the left kan extension would yield something trivial, so we ignore it.

The general notion is to find a functor $X : \mathcal{D} \to \mathcal{E}$ such that there exists a natural transformation $X \circ F \Rightarrow p$. However, there are many such functors, so we take the one which is terminal, which is the one that hugs p most tightly. That is, we take the functor $\tilde{p} : \mathcal{D} \to \mathcal{E}$ which is terminal among functors $X : \mathcal{D} \to \mathcal{E}$ such that there exists a natural transformation $X \circ F \Rightarrow p$. This is a way of extending p, while doing as little as possible to alter it.

3.1.2 Graph extensions

In particular, the right kan extension we will be considering is extending a graph $G : \mathsf{G} \to \mathsf{Set}$ along the inclusion $i : \mathsf{G} \to \mathsf{Int}$ defined by $i(E) = 1, i(V) = 0, i(s) = (0,1)$ and $i(t) = (1,0)$ like so:



(10)

Theorem 12. *The right kan extension of a graph G along $i : \mathsf{G} \to \mathsf{Int}$ is the presheaf $\mathsf{Path}(G)$ which maps a natural number n to the set $\{< (v_0, \ldots, v_n), (e_1, \ldots, e_n) > | v_i \in V_G, e_i \in E_G, s(e_i) = v_{i-1}$ and $t(e_i) = v_i\}$.*

The restriction maps here are the obvious ones, namely for $(r, s) : n \to m$ we send $< (v_0, \ldots, v_n), (e_1, \ldots, e_n) >$ to $< (v_r, \ldots, v_{n-s}), (e_{r+1}, \ldots, e_{n-s}) >$. In this way, we can interpret $< (v_0, \ldots, v_n), (e_1, \ldots, e_n) >$ as the element of $\mathsf{Path}(G)(n)$ which restricts to v_i under $(i, n-i)$ and to e_i under $(i-1, n-i)$.

Proof. To show that $\mathsf{Path}(G)$ is indeed terminal, we need to show that for any presheaf $X \in \widehat{\mathsf{Int}}$ such that there is a natural transformation $\eta : X \circ F \Rightarrow G$ in Grph, we obtain a unique natural transformation $\zeta : X \to \mathsf{Path}(G)$ in $\widehat{\mathsf{Int}}$ such that η is the same as $\zeta \circ F : X \circ F \Rightarrow \mathsf{Path}(G) \circ F$ (note that $\mathsf{Path}(G) \circ F \cong G$).

The existence of such a ζ is guaranteed by the two functions $\eta_0 : X(0) \to V_G$ and $\eta_1 : X(1) \to E_G$. These will similarly ensure uniqueness. Given any $x \in X(n)$ we can send this over to the element of $\mathsf{Path}(G)(n)$ which has $\eta_0((i, n-i)x)$ for each v_i and $\eta_1((i-1, n-i)x)$ for each e_i. This is in fact forced, since once we know that ζ agrees with η on 0,1 the naturality of a transformation $\zeta : X \to \mathsf{Path}(G)$ ensures that $\eta_0((i, n-i)x) = \zeta_0((i, n-i)x) = (i, n-i)\zeta_n(x)$. That is, an element $x \in X(n)$ is mapped by ζ to the unique element of $\mathsf{Path}(G)(n)$ which describes where the restrictions of x to 0,1 land under η. ∎

The above becomes quite intuitive upon reflection. A natural higher order extension of a graph is the sets of paths of given length inside the graph, which are possible length n behaviors that a graph can be host to. So we then obtain $\mathsf{Path}(G)$ which has exactly one copy of each edge-permissible concatenation of vertices. As we will see later in section 4.1.1, the examples we already met in 1 were in fact all obtained as the kan extension of an appropriate graph.

One way in which these path presheaves are well behaved is that a path is entirely determined by its length 0 and 1 subpaths, we will later see that this in fact characterizes those presheaves which are obtained from graphs.

Lemma 13. *Two paths are identical exactly if they agree on all their length 0 and 1 restrictions.*

This follows directly from the way we defined paths as n-tuples.

3.1.3 The path functor

We now obtain a functor $\mathsf{Path} : \mathsf{Grph} \to \widehat{\mathsf{Int}}$, which is simply the map $G \mapsto \mathsf{Path}(G)$. This will define for us a subcategory of $\widehat{\mathsf{Int}}$: the image of the Path functor. We will study the ways in which an Int presheaf can fail to be in this subcategory, we will also find an automorphism of Ω which characterizes these Path objects in $\widehat{\mathsf{Int}}$. But first, we will examine how Path treats graph homomorphisms. A graph homomorphism $f : G \to H$ determines the obvious natural transformation $\mathsf{Path}(G) \to \mathsf{Path}(H)$ given by $< (v_0, \ldots, v_n), (e_1, \ldots, e_n) > \mapsto < (f(v_0), \ldots, f(v_n)), (f(e_1), \ldots, f(e_n)) >$. As we're about to see, this manages to capture behavior inherent to $\widehat{\mathsf{Int}}$ itself.

3.2 The graph embedding

We have that Grph is equivalent to a subcategory of $\widehat{\mathsf{Int}}$.

Theorem 14. Path *is a fully faithful functor.*

Proof. It is faithful because if we have two graph homomorphisms $f, g : G \to H$ such that $\mathsf{Path}(f) = \mathsf{Path}(g) : \mathsf{Path}(G) \to \mathsf{Path}(H)$ then in particular, we have that $\mathsf{Path}(f)(0) = \mathsf{Path}(g)(0) : V_G \to V_H$ and $\mathsf{Path}(f)(1) = \mathsf{Path}(g)(1) : E_G \to E_H$, so that f and g are the same graph homomorphism.

To see that it is full, we assume we're given a morphism $\tilde{f}$ between graph presheaves and use it to define a graph homomorphism f, we then show that the image of a path of length > 1 is entirely determined by shorter paths, so that in fact $\tilde{f} = \mathsf{Path}(f)$.

So assume we're given a morphism $\tilde{f} : \mathsf{Path}(G) \to \mathsf{Path}(H)$, restricting to $\tilde{f}(0) : V_G \to V_H$ and $\tilde{f}(1) : E_G \to E_H$ we obtain a graph homomorphism, the desired commutativity from figure (9) is assured by the naturality of a presheaf morphism, which gives the commutativity the diagrams:

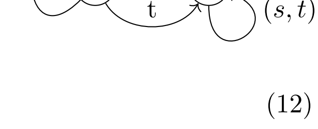
$$\begin{array}{ccc} E_G & \xrightarrow{(0,1)} & V_G \\ {\scriptstyle \tilde{f}(1)}\downarrow & & \downarrow{\scriptstyle \tilde{f}(0)} \\ E_H & \xrightarrow{(0,1)} & V_H \end{array} \qquad \begin{array}{ccc} E_G & \xrightarrow{(1,0)} & V_G \\ {\scriptstyle \tilde{f}(1)}\downarrow & & \downarrow{\scriptstyle \tilde{f}(0)} \\ E_H & \xrightarrow{(1,0)} & V_H \end{array} \tag{11}$$

thus, we obtain a graph homomorphism $f : G \to H$, such that $\mathsf{Path}(f)$ agrees with $\tilde{f}$ on 0 and 1. However, by an inductive process, $\tilde{f}$ is entirely determined by $\tilde{f}(0)$ and $\tilde{f}(1)$. Since, given some path $p \in \mathrm{Path}(G)(n)$ with $p = ((v_0, \ldots, v_n), (e_1, \ldots, e_n))$ we have by the naturality of $\tilde{f}$ that $\tilde{f}(p)$ maps under $(i, n-i)$ to $\tilde{f}((i, n-i)p)$ and to $\tilde{f}((i-1, n-i)p)$ under $i-1, n-i$. This then has length 0 and 1 restrictions agreeing with $< (f(v_0), \ldots, f(v_n)), (f(e_1), \ldots, f(e_n)) >$ and so by lemma 13 we get that $\tilde{f} = \mathsf{Path}(f) : \mathsf{Path}(G) \to \mathsf{Path}(H)$. ■

It is quite nice to be able to work with graphs, as this allows for a much more visual and intuitive analysis, since paths can actually be seen and imagined directly. This provides a visual description for at least an important subcategory of $\widehat{\mathsf{Int}}$. In this way, Grph can add a new understanding to $\widehat{\mathsf{Int}}$, but the converse is also true. Embedding Grph into $\widehat{\mathsf{Int}}$ also provides a lot more structure to study graphs with, in particular through adding extra subobjects which aren't necessarily graphs themselves. This is made possible by exploring that paths that a graph admits, which allows us to consider a graph that keeps its nodes and edges intact, but ignores or loses some of these longer paths. We will explore this more when discussing what sorts of statements we can make about graphs in $\widehat{\mathsf{Int}}$ later in section 4.2.

3.3 Subobject classifiers

So now we have in hand our most interesting tool of analysis, and that is looking at morphisms from some $\mathsf{Path}(G)$ to the subobject classifier Ω. To get a clearer picture of what that will look like, we will take a moment to talk about the subobject classifier represented as a graph, which is the same as the subobject classifier of Grph.



(12)

In general, given an Int presheaf X, we can find a graph related to it by taking the graph $(X(1), X(0), (0,1), (1,0))$ to be the pair of sets with a pair of maps between them that define a graph. We then apply this to Ω. We have that $\Omega(0)$ is the subobjects of $y0$ of which there are two: $y0$ and $C_\emptyset$. We have then $\Omega(0) = \{0, 1\}$. So given a morphism $p : X \to \Omega$ with corresponding subobject $S \hookrightarrow X$, then for the function $p_0 : X(0) \to \Omega(0)$ we get that 0 is where an element of $X(0)$ not included in the corresponding subobject $S(0)$ lands, and 1 is where an element of $X(0)$ included in $S(0)$ lands. Now, $\Omega(1)$ has 5 elements, and our labeling of them in (12) somewhat follows [1]. Important to note is that the question of where an element $x \in X(1)$ is taken to under p_1 depends on what p_0 does to $(0,1)x \in X(0)$ and $(1,0)x \in X(0)$. Is one of those is not an element of $S(0)$, then x has no chance of being in $S(1)$. If for example neither of these are in $S(0)$, then x gets mapped to $\emptyset$, a self loop at 0, consistent with the fact that both its restrictions to $X(0)$ get sent to

0 under p_0. Then it is also possible for one restriction of x to be in $S(0)$ but not the other, in this case we get s or t, depending on whether it's $(0,1)x$ or $(1,0)x$ respectively that's in $S(0)$. But if both of these are in $S(0)$, then there's still the question of whether x itself is in $S(1)$ to contend with. If it isn't, p_1 will map x to (s,t) (we can imagine that s (or t) is the statement asserting that the source (or target) node of x is included in S, and then (s,t) communicates that both these statements hold, but nothing more). And finally if $x \in S(1)$, then $p_1(x) = E$, the highest length 3 dyck path.

The graph in (12) is also the subobject classifier of Grph. The interpertation here is exactly as given above, only that in the case of Grph is also ends there, while for $\widehat{\mathsf{Int}}$ there is also $\Omega(n)$ for $n > 1$ to consider. In Grph however, a subgraph is just a subset of vertices and a subset of edges. There is however still the same condition, that for an edge to be included in a subgraph, both its source and target must be in that subgraph. We have then that a subgraph S of a graph G corresponds to a morphism $G \to \Omega_{\mathsf{Grph}}$ sending elements of V_G that are also in V_S to 1 and the others to 0. Edges in E_G will then follow the vertices adjacent to them, and in the case of a self loop at 1, that is an edge e with both its adjacent vertices included in V_S, there is also an option to include e in E_S, which means mapping it to E, and then mapping e to (s,t) otherwise.

If we try extending the above analysis and viewing $\Omega(2)$ as the length 2 paths in (12), so that to give Ω as in the image of the Path functor, we will find only 13 of the 14 elements we are seeking. That is because for the repeated self loop E there are two length 2 options, because even if both restrictions of an element $x \in X(2)$ are in $S(1)$, there is still the question of whether to include x itself in $S(2)$, but the presheaves in the subcategory cut out by Path do not allow for the possibility of two distinct elements agreeing on their restrictions (lemma 13). Thus we know at least that Ω itself is not the path presheaf of any graph.

3.4 Wild presheaves

At this point it might be natural to wonder what would make a wild presheaf in $\widehat{\mathsf{Int}}$ i.e. a presheaf that, like Ω, isn't given by a graph, and isn't a subobject of a graph presheaf. As mentioned above, for any presheaf $F : \mathsf{Int} \to \mathsf{Set}$ we automatically have a graph determined by the sets and functions:

$$F(1) \underset{(1,0)}{\overset{(0,1)}{\rightrightarrows}} F(0) \tag{13}$$

we can, by abuse of notation, call the presheaf of the above graph $\mathsf{Path}(F)$. This tells us that the only way a presheaf F might fail to be a subobject of a graph is if for some n we have that $F(n) \not\subset \mathsf{Path}(F)(n)$. But the fact that F is a presheaf is enough to enforce path-like behavior, so that every $p \in F(n)$ must actually be a length n path in the graph determined by F, as determined by its length 0 and 1 restrictions. It must therefore be the case that if F is not a subobject of a graph object, that it has too many elements, in particular there are some $p, q \in F(n)$ for some n such that p and q agree on their restrictions, but are distinct in $F(n)$ or in other words, F fails to satisfy Lemma 13.

Once we have this characterization of wild presheaves, we can easily see that many wild presheaves do in fact exist. In fact given any graph-presheaf, we need only pick a length > 1 element and duplicate it, and this will give us a wild presheaf.

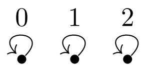
4 Graph presheaves and subobjects

Now that we have an expressive language for higher order graphs, let's look at some of the things we can do.

4.1 Visualizing presheaves

4.1.1 Revisiting our examples

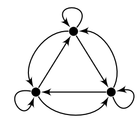
Given a presheaf F, we can use the construction of (13) to find the graph that F is based upon, and then check whether F is in fact simply the path presheaf for some graph. In fact, doing this we find that all the presheaves we've introduced in 1 can be given as the path presheaf for some graph, and the graph can be found in the manner suggested by (13). You can try this as an exercise on your own, or simply read how it's done here.



(14)

In the case of a constant presheaf C_S, this is the path presheaf of the graph with vertex set S and only self loops. So is we set $S = \{0, 1, 2\}$ the resulting C_S is the functor defined by taking paths in the graph in (14). Each vertex supplies exactly one length n path: repeating the loop at that vertex n times. Any restriction maps yield an identical path, it doesn't matter whether we drop loops from the beginning of our path or the end of it, we always get the same length m path consisting of repeating the loop at a vertex m times. In the case of the terminal object of $\widehat{\mathsf{Int}}$, the constant presheaf C_1, we are looking at the paths in a graph containing a single vertex with a self loop. Unsurprisingly, this is also the terminal object of Grph.

Similarly, a presheaf F_S, which maps lengths to sets of functions on intervals of that length, can be obtained from the complete graph on a vertex set S. That is, the graph with one copy of every possible arrow on S. For example, if S is again a three element set, the F_S it determines is given by the Path functor acting on the graph in (15). A function $f : [0, n] \to S$ is now represented as a path of length n following the vertices $v_i = f(i)$. Since this is a complete graph so that there is exactly one edge between each pair of vertices, all such paths are possible and unique. We can also see how restriction maps work nicely, since the restriction of a function to a subset of its domain corresponds to looking at a subpath.



(15)

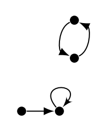
As for the presheaf sending a natural number to the natural numbers greater than or equal to it, this can be given as the paths in the infinite linear graph $\bullet \leftarrow \bullet \leftarrow \bullet \leftarrow \cdots$ where a path is written as its initial vertex, and vertices are numbered such that the terminal vertex of the graph is 0 and increasing by 1 moving to the right. As you can check, if we apply (r, s) to some path in this graph, s will just hack off some of the leading end, but r will give it a new initial vertex with value decreased by r. Note that the representables yn can similarly be given, except now with a finite length n path.

As for the preasheaf sending each number to the set $\{0, 1, 2, 3, 4\}$, this can be given as the graph with five vertices $\{0, 1, 2, 3, 4\}$, and an arrow going from each i to $i + 1$, and an arrow from 4 to 0. This is a circular graph with five vertices, and a path is identified with its starting vertex.

4.1.2 Constructing new presheaves

We can also use the Path functor to create new Int prsheaves, and investiagte the relationship between these, since it will generally be easier to find a graph homomorphism than a natural transformation between graph presheaves.

A demonstration of the power granted to us by graphs is provided by examining presheaves which map every $n \in \mathsf{Int}$ to a two element set, which is more or less as simple as nontrivial presheaves can get. In (16) we can see two variations on the constant presheaf C_2. Both of these graphs have exactly two paths of length n for every n, but differ in their behavior under restriction maps. For the graph on top, the maps $(0, 1)$ and $(1, 0)$ are both isomorphisms, but they have the opposite effect of one another, in this way the top graph determines a presheaf which is like C_2 with a twist in it. For the graph on the bottom however, only $(0, 1)$ is an isomorphism. Are there any other Int presheaves which can be given as the Path of some graph which yield a two element set for every n? You can also here use graph homomorphisms to find all the natural transformations between these presheaves.



(16)

Needless to say, with larger graphs we can invent much more subtle presheves, in accordance with whatever phenomena is being modeled. With larger graphs we can also start to consider more complex behavior types, in contrast to this paper which is concerned merely with presenting the basic building blocks used in those constructions.

4.2 Propositions about graphs

A graph subobject $S \hookrightarrow \mathsf{Path}(G)$ should be thought of as paths in G which satisfy some proposition given by the characteristic map θ_S. For example we might choose paths satisfying the proposition P which states "this path does not visit the vertex v". The vital thing which makes P allowable, so that this condition can be expressed in $\widehat{\mathsf{Int}}$, is that it satisfies downward consistency, but not necessarily upward consistency. That is, if a length n path p never visits a specified vertex, then certainly any subpath of p of length $m < n$ will never visit that vertex either. However, a path which satisfies P can definitely be a part of a longer path which doesn't. This condition on propositions arises from the fact that under $(r, s) : n \to m$ the highest dyck path in $\Omega(n)$ gets sent to the highest dyck path in $\Omega(m)$, but the converse doesn't hold because there are dyck paths in $\Omega(n)$ that aren'tas high as possible that still restrict to the highest dyck path in $\Omega(m)$.

An example of a proposition that isn't allowable is something like "this path visits the vertex v". Here, the statement could be true on some path p, but not true on its restrictions. In fact, if p visits any vertex besides v, then the restriction to that vertex will not preserve the truth of the proposition. Therefore, this is the sort of thing that cannot be expressed in $\widehat{\mathsf{Int}}$. We should therefore think of propositions in $\widehat{\mathsf{Int}}$ as being made over the entirety of an interval n. These are also exactly the sorts of statements we should allow, those that apply to a whole interval all at once, and therefore retain truth under restriction to subintervals. This is the sort of statement that is meant when someone says "it didn't rain today", or if they were to say "it rained from 2 to 5", which are typically statements made about the entirety of that interval of time. That sort of statement is to exclude a statement like "it rained today" which doesn't necessitate that it was raining all day, but rather that an event of rain occurred sometime during the day.

A more interesting sort of proposition P on a graph is one of the sort "this path visits the vertex v at least once per interval of length 5". This is allowable of course, since the statement remains true under restriction. Note that if we look at any path of length less than 5, it satisfies this condition trivially, even if it is so far way from v that it couldn't possibly get there in five steps. This sort of length-dependent statement opens the door to all sorts of interesting statements, such as "always visits v_1 within 3 ticks of visiting v_0" or "switches vertices every 6 ticks or less". We can even admit statements like "avoids v for at least twice as long as it last did" and "leaves v through a different edge than it did last", along with countless other sorts of statements.

4.3 more examples

In addition to the presheaves introduced by graphs themselves, we get other presheaves, which are subobjects of these presheaves given by graphs, but cannot themselves necessarily be produced as the presheaf given by some graph. A subobject F of a graph presheaf $\mathsf{Path}(G)$ corresponds to a morphism $\theta_F : \mathsf{Path}(G) \to \Omega$. If some path $p \in \mathsf{Path}(G)(n)$ is also in $F(n)$, then $\theta_{F,n}(p) = yn \in \Omega(n)$, and otherwise $\theta_{F,n}(p)$ will be some Dyck path subobject of yn, depending on which subpaths of p, if any, are in some $F(m)$.

The general notion of what can be a subobject F of a graph presheaf $\mathsf{Path}(G)$ is that a subobject may discard some possible paths in G, so long as it also discards longer paths which contain that path. Conversely, for every path p that $F(n)$ contains, every subpath of p must be contained in the appropriate $F(m)$. We may, for example, choose a vertex $v \in V_G$, and have F be the subobject of $\mathsf{Path}(G)$ that only contains paths which visit v at least once in every subpath of length 4. Notice that if a path p satisfies this property, a subpath of p will as well, while a superpath of p needn't.

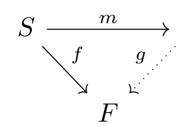
5 The g-modality

In a topos, of which any presheaf category is an example, there is often a notion of a *modality*. A modality is given by a automorphism $j : \Omega \to \Omega$ of the subobject classifier. Broadly speaking, a modality makes statements "more true". This is formalized by the three properties a modality must satisfy: (1) truth preservation $j \circ \mathsf{True} = \mathsf{True}$ (2) idempotence $j \circ j = j$, and (3) distributivity $j \circ \wedge = \wedge \circ j \times j : \Omega \times \Omega \to \Omega$. Two simple modalities are the identity and the morphism sending everything to true (i.e. the highest dyck path of size n). These are in fact the extremes of a modality, and a typical modality will find itself somewhere between these two: increasing the truth of some propositions, but leaving some others untouched. Relatedly, we have the j-closure $\overline{S}$ of a subobject $S \hookrightarrow X$ which is the subobject classified by $j \circ \theta_S : X \to \Omega$. The properties of j translate to properties of the closure, namely: (1) S is a subobject of $\overline{S}$ (2) $\overline{\overline{S}} = \overline{S}$, and (3) $\overline{S \cap T} = \overline{S} \cap \overline{T}$.

We will be considering a modality we call g which acts on a Dyck path as if it were the statement "this path sends its length 1 restrictions to 1" (that is, as a functor $\mathrm{el}(yn) \to 2$). So g sends a Dyck path (or a part of it) to true (the highest possible path) whenever the path doesn't dip below height 1. We can think of this modality as checking for truth on edges. Verification that g satisfies the required properties is straightforward.

5.1 g-sheafs

A subobject $S \hookrightarrow X$ is called *j-dense* if $\overline{S} = X$. So a $\widehat{\mathsf{Int}}$ sub-presheaf $S \hookrightarrow F$ is g-dense if S defines the same graph as F. A presheaf F is called *g-separated* if for every dense subobject $m : S \hookrightarrow Y$ and morphism $f : S \to F$ there is at most one morphism $g : Y \to F$ making the triangle in (17) commute, if there is always at least one such morphism, we call F an effective g-presheaf. If there is always exactly one such morphism, we call F a g-sheaf.



(17)

Theorem 15. *The separated g-presheaves are presheaves F such that F is a subobject of a graph object* $\mathsf{Path}(G)$*, and graph objects are also effective g-presheaves while their proper subobjects are not, so that g-sheaves are exactly graph objects.*

Proof. First, we have that a graph object $\mathsf{Path}(G)$ is separated, since a dense subobject S defines the same graph as Y and by the process in the proof of theorem 14 a morphism into a graph object $\mathsf{Path}(G)$, if it exists, is entirely defined by the maps $Y(0) \to \mathsf{Path}(G)(0)$ and $Y(1) \to \mathsf{Path}(G)(1)$, so that the extension of $f : S \to \mathsf{Path}(G)$ to $g : Y \to \mathsf{Path}(G)$ is unique if it exists. Now if F is a subobject of a graph object $\mathsf{Path}(G)$ it too must be separated. This follows from the defining property of a subobject $i : F \hookrightarrow \mathsf{Path}(G)$ that if two morphisms $a, b : H \to F$ have the same composition with i so that $i \circ a = i \circ b$ then $a = b$. So now, if the extension $g : Y \to F$ wasn't unique, then this could be made into a non-unique extension $g : Y \to F \to \mathsf{Path}(G)$, which we have shown impossible, thus subobjects of graph objects are also separated. Conversely, if a presheaf F is not a subobject of any graph, then there must be some $p, q \in F(n)$ for some n such that p and q are not equal, but agree on all their restrictions. We can then take the dense subobject $S \hookrightarrow yn$ defined by $S(0) = yn(0), S(1) = yn(1), S(m) = \emptyset : \forall m > 1$, so that S is the graph of yn without any of the paths, and then we can map S to the restrictions of p and q and extend this in two ways to yn, one for p and another for q, so that F is not separated.

It is also clear that a graph object is an effective g-presheaf. Given a morphism $S \to \mathsf{Path}(G)$, as above we have a graph homomorphism from the graph of Y to G. This is sufficient to take any element $y \in Y(n)$ and map it to the corresponding element of $\mathsf{Path}(G)$. However, for a presheaf F which is a strict subobject of a graph, more precisely F is a dense subobject of a graph object $\mathsf{Path}(G)$, but is not isomorphic to $\mathsf{Path}(G)$, then we will have some $p \in \mathsf{Path}(G)(n)$ such that $p \notin F(n)$. So that the subobject of p which is its length 1 restrictions is dense in p, and there is a morphism from it to F, but the morphism cannot be extended to p itself, and thus F is not an effective g-presheaf. ■

Some wild type presheaves are effective g-presheaves, but not all. Being an effective g-presheaf is the condition that every possible superpath exists. More precisely, it is the condition on a presheaf F that for every $x \in F(n)$ and $y \in F(m)$ if $(n,0)x = (0,m)y$ then there is $z \in F(m+n)$ such that $(0,m)z = x$ and $(n,0)z = y$, so that z is the composition of x and y. This is necessary for effectiveness, since otherwise we could take S to be x and y and their restrictions, and take Y to be z and its restrictions, which would contradict effectiveness.

6 Conclusion

7 Acknowledgements

This research was supported by the Paul E. Gray UROP Fund. The author would like to thank our colleagues from the MIT category theory group who provided insight and expertise that greatly assisted the research, although they may not agree with all of the interpretations/conclusions of this paper.

The author thanks David Spivak for supervising and guiding this research project, who greatly improved the present work with his patience and many insightful comment.

References

[1] S. Vigna, "A Guided Tour in the Topos of Graphs," *ArXiv Mathematics e-prints*, June 2003.

[2] P. Schultz and D. I. Spivak, "Temporal Type Theory: A topos-theoretic approach to systems and behavior," *ArXiv e-prints*, Oct. 2017.