Polynomial functors, a degree of generality

tslil clingman

February 2019

1. Introduction

In their 2010 revision of the paper Polynomial Functors and Polynomial Monads, Gambino and Kock write:

> Notions of polynomial functor have proved useful in many areas of mathematics, ranging from algebra [41, 43] and topology [10, 50] to mathematical logic [17, 45] and theoretical computer science [24, 2, 20].

One of the primary differences between this document and theirs is that they have a bibliography. The other is that this document is an attempt at a motivating preamble for the early definitions and theorems of their one. This information might allow the reader to infer the content of the following two sections.

1.1. What this document is

1.2. What this document is not

1.3. Prerequisites

In this document we will take for granted the internal language supported by a locally cartesian-closed category – a category equipped with a choice of pullback functor f^* for each arrow f, such that f^* has a right adjoint Π_f, in addition to its automatic left adjoint Σ_f. Depending on the internet-time point of origin of this document, a supporting introduction to locally cartesian-closed categories may be related to my website `math.jhu.edu/~tclingm1` by at least one of the following

$$\{\text{will appear on, is available on}\}.$$

The existence or availability of that other document notwithstanding, we will briefly remind the reader of the important aspects of our notation.

Objects $X \xrightarrow{x} C$ of a slice category $\mathfrak{C}/C$ are to be regarded as indexed collections of objects $(X_c \mid c \in C)$. Of course this isn't literally true, but we have the following serially commutative diagram for sets as motivation, where the vertical isomorphisms are natural in the indicated manner.

$$
\begin{tikzcd}
\textsc{Set}/A \arrow[rr, bend left, "\Sigma_f"] \arrow[rr, bend right, "\Pi_f"'] \arrow[dd, "\cong"'] & & \textsc{Set}/B \arrow[ll, "f^*" description] \arrow[dd, "\cong"] \\
& & \\
{[A^{\mathrm{op}}, \textsc{Set}]} \arrow[rr, bend left, "\mathrm{Lan}_f"] \arrow[rr, bend right, "\mathrm{Ran}_f"'] & & {[B^{\mathrm{op}}, \textsc{Set}]} \arrow[ll, "{[f, \textsc{Set}]}" description]
\end{tikzcd}
$$

In fact, understanding the internal language amounts to understanding this diagram – and especially the nature of the vertical natural isomorphisms. The reader for whom this is new is recommended to fully explore the diagram before proceeding.

Extending the above situation to an arbitrary locally cartesian-closed category $\mathfrak{C}$ with morphism $f : A \to B$ and objects $(X_a \mid a \in A)$ of $\mathfrak{C}/A$ and $(Y_b \mid b \in B)$ of $\mathfrak{C}/B$ we derive the following definitions, writing Σ for coproduct:

$$
\begin{aligned}
\Sigma_f(X_a \mid a \in A) &:\equiv \left(\sum_{(a \in A_b)} X_a \mid b \in B \right) \\
f^*(Y_b \mid b \in B) &:\equiv \left(Y_{f\,a} \mid a \in A \right) \\
\Pi_f(X_a \mid a \in A) &:\equiv \left(\prod_{(a \in A_b)} X_a \mid b \in B \right)
\end{aligned}
$$

The reader is encouraged to make the connection between this and the "quantifiers as adjoints" prescription of locally cartesian-closed categories[1]. Extending this, we consider $\Sigma(a \in A_b), X_a$ to comprise pairs (a, x) where the type/fibre of x, that is, X_a depends on the first component a. Likewise we view $\Pi(a \in A_b), X_a$ as the collection of functions $\{f : A \to \cup X_a \mid \forall a \in A_b, [f\,a \in X_a]\}$ – dependent functions. Of specific import to us will be the counit of the adjunction $(-)^* \dashv \Pi_{(-)}$ which bears the interpretation of function evaluation.

The reader is encouraged to find a way to ensure that all of this means something before proceeding.

[1] See, for instance, §9.5 of "Category Theory" by S. Awodey for a lucid account.

2. The climb

2.1. On generalisation and new behaviour

A classical polynomial is something of the form $\Sigma(n \in N), a_n x^n$ where $N \subseteq \mathbb{N}$, and where perhaps we require some finiteness condition on the set $\{n \in N \mid a_n \neq 0\}$ or N. As x and a_n are really 'valued' in the same sort of things, say natural numbers, we see that regarded simply as a formal sum the data of such a polynomial is no more than that of the sequence of coefficients $(a_n)_N$.

There is a very limited theory to be enjoyed should we constrain the study of polynomials to those concepts naturally captured by their sequences of coefficients – that is, as opposed to polynomial *functions*. In order to clearly delineate the theory of polynomial functions from that of sequences of coefficients, in the coming sections we will be careful to distinguish a *polynomial* – a sequence of coefficients or a generalisation thereof – from the *extension of the polynomial* to a *function* – or a generalisation thereof. Classically these terms are either intentionally confused or the former notion is not dealt with. That is, we not wish to make a *category* error in our work, and moreover the two theories – that corresponding to polynomials and that corresponding to the extension of polynomials to functions – are independently interesting.

With that said, the idea that polynomials may be extended to functions leads to all sorts of new behaviour when compared to the idea that polynomials are merely sequences of numbers. For example, when extended to functions, we now have the tools of composition and roots, ideas which we understand to be at the heart of classical algebraic geometry and complex analysis. We wish to say: when extended to functions, polynomials have interesting behaviour.

Our goal then, broadly stated, is to liberate ourselves of the constraint that polynomials functions are *merely* functions, and determine whether new behaviour reveals itself should we consider them somehow as *functors*.

2.2. Categorical polynomials

A traditional first step in generalisation is to rephrase our current understanding in some equivalent, though seemingly less familiar way. We have no intention of breaking with this custom here.

In the case of polynomials whose coefficients are valued in the natural numbers, polynomials *in* $\mathbb{N}$, we may leverage the fact that

$$a_n x^n = \underbrace{x^n + \ldots + x^n}_{a_n \text{ times}}$$

to recapture the data of a polynomial $\Sigma(n \in N), a_n x^n$ in the slightly unfamiliar form of $\Sigma(n \in N), x^{b_n}$. For example, $2x^3$ may be encoded as $b_0 = b_1 = 3$ and $b_n = 0$ for $n > 1$. It is this form that we will find most amenable to immediate generalisation. A sample thought process of categorification is provided in the table below:

thing	boring	new
x	number	object
$(b_n)_N$	sequence	object $B \xrightarrow{f} N$ of slice
$x^{b_n} = \underbrace{x \cdots x}_{b_n}$	product	... product?

The idea here is that we can reshape the data of a polynomial into that captured by an object of a slice category, an indexed object. Should we do so, we may define the extension of our newly formed polynomial $f : B \to N$ to a *functor* Ext_f as the composite functor

$$\mathfrak{C}/1 \xrightarrow{(!_B)^*} \mathfrak{C}/B \xrightarrow{\Pi_f} \mathfrak{C}/N \xrightarrow{\Sigma_{!_N}} \mathfrak{C}/1$$

In the internal language, this functor acts on (X) of $\mathfrak{C}/1$ to produce the object

$$\left(\sum_{(n \in N)} \prod_{(b \in B_n)} X \right).$$

In Set, when viewed through the lens $b_n = |B_n|$, this really is the analogue of a polynomial function acting on sets. Encouraged by this we fix terminology in our first definition.

Def. 2.1. A *polynomial* in a category $\mathfrak{C}$ is an arrow $f : B \to N$. The *extension of a polynomial* f to a functor, $\mathrm{Ext}_f : \mathfrak{C}/1 \to \mathfrak{C}/1$, is defined to be composite $\Sigma_{!_N} \Pi_f (!_B)^*$. ⌟

Here again we carefully distinguish a polynomial from its extension to a functor, and we will see that in our generalisation both categories of objects may be of independent interest. In the later sections we will relate them, but for now we keep our distinction clear.

Remark 2.2. The categorically minded reader may find themself wondering whether Ext is the 'on objects' portion of a functor extending polynomials and their morphisms to functors and natural transformations. The answer is both satisfying and deep. But although it may have delighted the reader to see preliminary notions in this direction here, it would be unwise to present it at such a time as we have not yet reached our most general definition of polynomial. ◁

Remark 2.3. Although of no immediate interest, in fact we already have new behaviour. Ext_f is a *functor* and so now our polynomials understand how to operate on 'arrows between numbers'. ◁

2.3. Composition

A generalisation is only worthy of the name if it captures some of the richness of the original notion. In our case this richness stems from the notion that composites of polynomials are polynomials. Said more carefully, the classical theorem might be phrased as follows.

Thm. 2.4. *Given coefficient sequences $(a_n)_N$ and $(a'_n)_N$ and their extension to polynomial functions $p(x)$ and $p'(x)$, there exists a coefficient sequence $(a''_n)_N$ such that the composite function $p \circ p'$ is the extension of $(a''_n)_N$.*

Our task, therefore, is to prove the analogous theorem for categorical polynomials and their extensions to functors. We will prove this fact to the satisfaction of the reader critical of internal languages, and in greater generality, later. For now we content ourselves with the far more manageable task of working internally.

To that end, let $g : A \to M$ and $f : B \to N$ be polynomials in $\mathfrak{C}$. We will examine the action of the composite functor $\mathrm{Ext}_g \circ \mathrm{Ext}_f$ on an object (X) of $\mathfrak{C}/1$ and attempt to recast the result in the form $(\Sigma(?), \Pi(??), X)$ so that for some choice of arrow $??? : ? \to ??$, $\mathrm{Ext}_g \circ \mathrm{Ext}_f \cong \mathrm{Ext}_{???}$.

To begin we perform the straightforward expansion

$$\mathrm{Ext}_g \, \mathrm{Ext}_f(X) \equiv \sum_{(m \in M)} \prod_{(a \in A_m)} \sum_{(n \in N)} \prod_{(b \in B_n)} X = \sum_{(m \in M)} \left(\sum_{(n \in N)} X^{B_n} \right)^{A_m} .$$

Beyond the fact that this expression begins with a Σ, there is nothing else about it which matches our desired form. Our goal is thus to rewrite this so that we may deduce ?, ??, and ???. We do this in several phases.

2.3.1. Distributivity

For the moment let us consider the expression $\left(\Sigma(n \in N), X^{B_n}\right)^{A_m}$ as though we were working in a ring and introduce some finiteness constraints. That is, suppose that A_m and B_n were natural numbers, that $N = \{0, \ldots, k\}$, and that X was a ring element. That is to say, imagine that we were tasked with rewriting $(x^{b_0} + \ldots + x^{b_k})^{a_m}$ as a sum of products instead of as a product of sums.

We may be tempted to appeal to the peculiarities of combinatorics in our ring in attempting to write down a *reduced* form of the expression, but there is of course a general answer which does not depend on the nature of the underlying ring.

Should be begin to multiply out,

$$(x^{b_0} + \ldots + x^{b_k})^{a_m} = \underbrace{x^{b_0} \cdots x^{b_0}}_{a_m} + \underbrace{x^{b_1} \cdot x^{b_0} \cdots x^{b_0}}_{a_m} + \ldots + \underbrace{x^{b_k} \cdots x^{b_k}}_{a_m}$$

we see that the answer is precisely a sum over all possible a_m-length lists in $\{b_0, \ldots, b_k\}$ of a_m-length products of x's to the power of elements of the list. This clumsy natural language description is really the claim that $(x^{b_0} + \ldots + x^{b_k})^{a_m} = \Sigma(l \in N^{a_m}), x^{b_{l_1}} \cdots x^{b_{l_k}}$ – a basic fact about the distributivity of multiplication over addition.

This claim remains formally true in our category when translated back into the appropriate language – a fact we shall prove as lemma 3.3. That is, we assert that we have the equality

$$\left(\sum_{(m \in M)} \prod_{(a \in A_m)} \sum_{(n \in N)} \prod_{(b \in B_n)} X \right) = \left(\sum_{(m \in M)} \sum_{(l \in \Pi(a \in A_m), N)} \prod_{(a \in A_m)} \prod_{(b \in B_{l(a)})} X \right). \tag{2.3.1}$$

This property we will be able to recast as a consequence of a distributivity lemma about LCCs, or as we shall see later, the ~~Axiom~~ Theorem of Choice[2]. Either way, this is *almost* what we wanted, but we appear to have contracted a case of double-vision.

2.3.2. Associativity

The last tool we need is a generalisation of associativity for products. We do this through the means of the following obvious lemma and its corollaries.

Lem. 2.5 (Associativity). *Given a category $\mathfrak{C}$ define $\Sigma(C) :\equiv \mathfrak{C}/C$ on objects, and on morphisms $c : C \to C'$ let $\Sigma_c : \mathfrak{C}/C \to \mathfrak{C}/C'$ be post-composition. Then Σ is a functor.* ∎

Cor. 2.6. *Given $f : A \to B$ and $g : B \to C$ of $\mathfrak{C}$ we have in particular that $gf = \Sigma_g f$ and so by the above lemma the following diagram commutes.*

$$\begin{tikzcd} \mathfrak{C}/A \arrow[r, "\Sigma_f"] \arrow[rr, bend left, "\Sigma_{gf}"] \arrow[rr, bend right, "\Sigma_{\Sigma_g f}"'] & \mathfrak{C}/B \arrow[r, "\Sigma_g"] & \mathfrak{C}/C \end{tikzcd}$$

Thus for $(X_a \mid a \in A)$ we have

$$\left(\sum_{(b \in B_c)} \sum_{(a \in A_b)} X_a \,\middle|\, c \in C \right) = \left(\sum_{((b,a) \in \Sigma(b \in B_c), A_b)} X_a \,\middle|\, c \in C \right)$$

∎

In then non-dependent case where $X = X' \times A$, $A = A' \times B$, $B = B' \times C$, and all the maps are projections, $\Sigma(b \in B_c), A_b$ reduces to $B' \times A'$ and so on so that our corollary is the statement $B' \times (A' \times X') = (B' \times A') \times X'$.

We are now in the position to further reduce our work. By the above result we have the equality

$$\sum_{(m \in M)} \sum_{(l \in \Pi(a \in A_m), N)} \prod_{(a \in A_m)} \prod_{(b \in B_{l(a)})} X \quad = \quad \sum_{((m,l) \in \Sigma(m \in M), \Pi(a \in A_m), N)} \prod_{(a \in A_m)} \prod_{(b \in B_{l(a)})} X.$$

Of course there is a similar lemma for the functoriality of $\Pi_{(-)}$ and a similar associativity corollary for it, but beware: $\Pi_g \Pi_f = \Pi_{gf} = \Pi_{\Sigma_g f}$. Using this we obtain our final reduction:

$$\mathrm{Ext}_g \, \mathrm{Ext}_f(X) \; = \sum_{((m,l) \in \Sigma(m \in M), \Pi(a \in A_m), N)} \prod_{((a,b) \in \Sigma(a \in A_m), B_{l(a)})} X$$

[2] Of course there is a trick here, but we hope that this statement will be sufficiently jarring or tantalising so as to interest the reader in the coming matter.

2.3.3. A triumph of notation

At this point it is hoped that the reader will have found all our manipulations to be intuitive, if not justifiably correct. Assuming as much we may declare our project to be a success:

Given two polynomials $g : A \to M$ and $f : B \to N$, the composite of their extensions to functors $\mathrm{Ext}_g \circ \mathrm{Ext}_f$ is indeed the extension of a third polynomial to a functor. We have proven this fact by construction, the composite of the two polynomials is easily read off to be the family

$$\left(\sum_{(a \in A_m)} B_{l(a)} \,\middle|\, (m, l) \in \sum_{(m \in M)} \prod_{(a \in A_m)} N \right)$$

and the previous sections demonstrate that its extension to a functor is exactly as we desire. Of course, to the reader sceptical of the truth of our manipulations or the value of the use of the internal language of our LCCC, such a statement could be viewed as anywhere from un-enlightening to useless. It is to that reader to which we dedicate the next section.

2.4. Externalisation

In this section our goal is to describe the arrow $\theta : \beta \to \alpha$ of $\mathfrak{C}$ corresponding to the map we constructed internally, above. It is a fairly straightforward task to externalise the object $\alpha :\equiv \Sigma(m \in M), \Pi(a \in A_m), N$ – in fact its very name is essentially its recipe: view N as dependent on $a \in A$ and then take Π_g of this dependence. More concretely, we may define α to be the *projection* of the object $\Pi_g(!_A)^*(!_N)$ of $\mathfrak{C}/M$ – the 'total space'. That is, we set $\alpha :\equiv \Sigma_{!_M}\Pi_g(!_A)^*(!_N)$ under the isomorphism $\mathfrak{C}/1 \cong \mathfrak{C}$.

On the other hand β is somewhat more complicated for the appearance of $l(a)$ in the indexing of B. We know that function evaluation is given by the counit of the adjunction $(-)^* \dashv \Pi_{(-)}$ and that substitution is given by pullback.

The precise nature of this construction may be understood by contemplating the below diagram, whence we ultimately define $\theta :\equiv \Sigma_{(\Pi_g \pi_1)^* g}(\varepsilon^* {\pi_2}^* f)$ to be the top horizontal composite, from which we may read off β and α as its domain and codomain respectively.

$$
\begin{tikzcd}
 & (B_{l(a)} \mid ((m,l),a) \in \Sigma((m,l) \in \alpha), A_m) & (A_m \mid (m,l) \in \alpha) & & \Sigma(m \in M), \Pi(a \in A_m), N & \\
 & \varepsilon^* {\pi_2}^* B \arrow[r, "\varepsilon^* {\pi_2}^* f"] \arrow[dl] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & g^* \Pi_g (!_A)^* N \arrow[rr, "(\Pi_g \pi_1)^* g"] \arrow[d, "\varepsilon"] \arrow[dr] \arrow[dr, phantom, "\lrcorner", very near start] & & \alpha :\equiv \Pi_g (!_A)^* N \arrow[d, "\Pi_g \pi_1"] \arrow[dr] & \\
1 & {\pi_2}^* B \arrow[r, "{\pi_2}^* f"] \arrow[dd] \arrow[dr, phantom, "\lrcorner", very near start] & (!_A)^* N \arrow[r, "\pi_1"] \arrow[dd, "\pi_2"] \arrow[dr, phantom, "\lrcorner", very near start] & A \arrow[r, "g"'] \arrow[dd, "!_A"] & M \arrow[r] & 1 \\
(B_n \mid (a,n) \in A \times N) & & & & & (\Pi(a \in A_m), N \mid m \in M) \\
 & B \arrow[r, "f"'] \arrow[uul] & N \arrow[r, "!_N"'] & 1 & (A_m \mid m \in M) & \\
 & (B_n \mid n \in N) & (N \mid a \in A) & & &
\end{tikzcd}
$$

Once this diagram has been understood by the reader[3], we wish to draw attention to the top and bottom zig-zag boundaries beginning at the left 1 and ending at the right 1 – and in so doing address the arguably irrelevant appearance of the terminal at both extremes.

The bottom zig-zag, $1 \leftarrow B \rightarrow N \rightarrow 1 \leftarrow A \rightarrow M \rightarrow 1$, is awfully reminiscent of our two polynomials and their functor extensions. In particular, lining up the maps with $^* \ \Pi \ \Sigma \ ^* \ \Pi \ \Sigma$ gives precisely the composite $\mathrm{Ext}_g \circ \mathrm{Ext}_f$. Similarly, for the top zig-zag, $1 \leftarrow \varepsilon^* {\pi_2}^* B \rightarrow \alpha \rightarrow 1$ where we have composed the middle maps[4], applying $^* \ \Pi \ \Sigma$ gives us Ext_θ – a functor which, we have argued, is isomorphic to $\mathrm{Ext}_g \circ \mathrm{Ext}_f$. Thus although the commutativity of *this* diagram does not seem to be of immediate import, its *structure* does inform the way that various functors inter-mingle.

In the coming section we will distil from this situation two important components, and in section 4 we will demonstrate how composition of the extensions of polynomials (there in greater generality yet) comprises only these two components.

At this point we hope that the once-sceptical reader now feels that the validity of our construction depends only upon our claims of distributivity – a deficiency to which we will soon attend – and that the internal language is useful both as a guide for our external arguments and as an independent methodology all its own.

[3] there's no rush

[4] so that the second arrow in the zig-zag is θ

3. The ladder

This section is dedicated entirely to establishing the validity of lemmas 3.1 and 3.3 below, results which will find repeated and useful employ in our theory to come. The reader with only a passing interest in the details may content themself with the statements of the lemmas alone, the surrounding framework is precisely that.

3.1. The logician's Beck-Chevalley lemma

Lem. 3.1. *In a locally cartesian-closed category $\mathfrak{C}$, given the below pullback square we have isomorphisms of functors $\Sigma_q p^* \cong g^* \Sigma_f$ and $\Pi_q p^* \cong g^* \Pi_f$.*

$$\begin{tikzcd} D \arrow[r, "p"] \arrow[d, "q"'] \arrow[dr, phantom, "\lrcorner", very near start] & A \arrow[d, "f"] \\ B \arrow[r, "g"'] & C \end{tikzcd}$$

Proof. Take $X \xrightarrow{x} A$ of $\mathfrak{C}/A$ and consider that by the pullback composition lemma we have the situation in the below-left diagram. From this we may read off that $\Sigma_q p^* x \cong g^* \Sigma_f x$, and the extension to morphisms and naturality is clear. With this in hand we may deduce the natural isomorphisms of functors as indicated below-right.

$$\begin{tikzcd} \bullet \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d] \\ D \arrow[r, "p"] \arrow[d, "q"'] \arrow[dr, phantom, "\lrcorner", very near start] & A \arrow[d, "f"] \\ B \arrow[r, "g"'] & C \end{tikzcd}$$

$$\begin{aligned} \mathfrak{C}/B(-, \Pi_q p^* -) &\cong \mathfrak{C}/P(q^* -, p^* -) \\ {}^{\text{(adj.)}} &\cong \mathfrak{C}/A(\Sigma_p q^* -, -) \\ {}^{\dagger} &\cong \mathfrak{C}/A(f^* \Sigma_g -, -) \\ {}^{\text{(adj.)}} &\cong \mathfrak{C}/B(-, g^* \Pi_f -) \end{aligned}$$

In the isomorphism marked † we have made use of the fact that, if the square of f, p, q, g above is a pullback then so too is the square g, q, p, f, and we then applied our partial result to it. Yoneda concludes the proof. ■

Remark 3.2. Mates also provide an efficient means to derive the Π isomorphism from that of the Σ, but this proof is (in the author's opinion) more fun. ◁

Under the interpretation of adjoints as quantifiers the statement of lemma 3.1 may be understood to mean that substitution of variables commutes with quantification.

3.2. The distributivity lemma

Lem. 3.3. *(Distributivity) In a locally cartesian-closed category $\mathfrak{C}$, given the below diagram we have an isomorphism of functors $\Pi_f \Sigma_u \cong \Sigma_{\Pi_f u} \Pi_g {\varepsilon_u}^*$.*

$$\begin{tikzcd}
 & f^*\Pi_f C \arrow[r, "g"] \arrow[dl, "\varepsilon_u"'] \arrow[d, "f^*\Pi_f u"] \arrow[dr, phantom, "\lrcorner" very near start] & \Pi_f C \arrow[d, "\Pi_f u"] \\
C \arrow[r, "u"'] & B \arrow[r, "f"'] & A
\end{tikzcd}$$

In order to prove this result easily we will need some supporting theory, introduced in the coming section.

3.2.1. Cartesian functors, natural transformations and adjunctions

Def. 3.4. With no assumptions on the categories,

1. A functor is termed cartesian if it preserves pullbacks.

2. A natural transformation is termed cartesian if all of its naturality squares are pullbacks.

3. An adjunction $F \dashv G : \mathfrak{C} \to \mathfrak{D}$ is cartesian when both the unit and counit are cartesian natural transformations, and both functors are cartesian.

$\lrcorner$

Next we need some structural lemmas about cartesian transforms.

Lem. 3.5. *(Properties of cartesian natural transformations) Given functors $H, K : \mathfrak{C} \to \mathfrak{D}$ where $\mathfrak{C}$ has a terminal object 1, and a natural transformation $\alpha : H \Rightarrow K$,*

1. *if α is cartesian then α is an isomorphism iff $\alpha_1 : H1 \to K1$ is, as an arrow of $\mathfrak{D}$,*

2. *if $H', K' : \mathfrak{D} \to \mathfrak{E}$ are functors such that either H' is cartesian or K' is cartesian, and $\beta : H' \Rightarrow K'$ is cartesian and so too is α, then $\beta * \alpha$ is cartesian.*

$\blacksquare$

We also establish a connection to mates.

Lem. 3.6. *(Cartesian mates) Given a natural transformation α as below with $F \dashv G$ and $F' \dashv G'$ cartesian adjunctions and H and K cartesian functors, α is cartesian iff its mate is.*

$$\begin{tikzcd}
\mathfrak{C} \arrow[r, "F \dashv G"] \arrow[d, "H"'] \arrow[dr, phantom, "\overset{\alpha}{\Nearrow}"] & \mathfrak{D} \arrow[d, "K"] \\
\mathfrak{C}' \arrow[r, "F' \dashv G'"'] & \mathfrak{D}'
\end{tikzcd}$$

Proof. By direct computation, applying lemma 3.5 (2) and pullback lemmas. ∎

Our use-case demands a specific instance of cartesian adjunctions.

Lem. 3.7. *In a locally cartesian-closed category $\mathfrak{C}$, for $f : A \to B$ the adjunction $\Sigma_f \dashv f^*$ is cartesian.*

Proof. It is clear that both functors are cartesian for they are left adjoints. That the unit and counit are cartesian follows from the same pullback lemmas as before. ∎

Finally, while both $(-)^*$ and $\Pi_{(-)}$ are right adjoints and so are continuous, all is not lost for $\Sigma_{(-)}$.

Lem. 3.8. *Given $f : A \to B$ in $\mathfrak{C}$, $\Sigma_f : \mathfrak{C}/A \to \mathfrak{C}/B$ preserves connected limits.*

Proof. By direct computation. ∎

3.2.2. Proof of the distributivity lemma

Armed with the above lemmas, the proof of lemma 3.3 is now readily obtained.

Proof. (Distributivity lemma) We apply lemma 3.1 to our pullback square in lemma 3.3 noting that $u\varepsilon_u = \Pi_f u$, to construct an isomorphism as below-left. Note that $u^* \dashv \Sigma_u$ and $(\Pi_f u)^* \dashv \Sigma_{\Pi_f u}$ are cartesian adjunctions (lemma 3.7), isomorphisms are cartesian, and $\Pi_g {\varepsilon_u}^*$ and Π_f are (composites of) right adjoints and so are cartesian. Thus we may apply lemma 3.6 to deduce that the mate of the isomorphism, α below-right, is cartesian so that by lemma 3.5 (1) it is an isomorphism iff α_{id_C} is an isomorphism in $\mathfrak{C}/A$. Direct computation shows that this component is the identity.

$$\begin{tikzcd}
\mathfrak{C}/C \arrow[r, "\Pi_g {\varepsilon_u}^*"] & \mathfrak{C}/\Pi_f C \\
\mathfrak{C}/B \arrow[u, "u^*"] \arrow[r, "\Pi_f"'] \arrow[ur, phantom, "\cong\Nearrow"] & \mathfrak{C}/A \arrow[u, "(\Pi_f u)^*"']
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathfrak{C}/C \arrow[r, "\Pi_g {\varepsilon_u}^*"] \arrow[d, "\Sigma_u"'] \arrow[dr, phantom, "\alpha\Swarrow"] & \mathfrak{C}/\Pi_f C \arrow[d, "\Sigma_{\Pi_f u}"] \\
\mathfrak{C}/B \arrow[r, "\Pi_f"'] & \mathfrak{C}/A
\end{tikzcd}$$
∎

3.2.3. What does this mean for you?™

As before we explore the internal version of this theorem. In recalling the diagram at hand, below, we annotate it just as we did in section 2.4. Beware that the *object* $f^*\Pi_f C$ is naturally a *matrix*, it is indexed by two variables – those coming from B and those coming from $\Pi_f C$ – and so we have annotated it in two ways. We leave it as an exercise to determine the action of g and $f^*\Pi_f u$ on the respectively reversed presentations of $f^*\Pi_f C$.

$$
\begin{tikzcd}
\color{green}{\substack{\Sigma(b\in B),\Pi(b'\in B_{f(b)}),C_{b'}\\ \Sigma((a,l)\in\Sigma(a\in A),\Pi(b\in B_a),C_b),B_a}} \arrow[drr, dashed, green] & & & & \color{blue}{(B_a \mid (a,l)\in\Sigma(a\in A),\Pi(b\in B_a),C_b)} \arrow[dl, dashed, blue] \\
\color{blue}{(\Pi(b'\in B_{fb}),C_{b'} \mid b\in B)} \arrow[drr, dashed, blue] & & f^*\Pi_f C \arrow[r, "g"] \arrow[d, "f^*\Pi_f u"] \arrow[dl, "\varepsilon_u"'] \arrow[dr, phantom, "\lrcorner", very near start] & \Pi_f C \arrow[d, "\Pi_f u"] & \\
& C \arrow[r, "u"'] & B \arrow[r, "f"'] & A & \color{blue}{(\Pi(b\in B_a),C_b \mid a\in A)} \arrow[ul, dashed, blue] \\
& \color{blue}{(C_b \mid b\in B)} \arrow[ur, dashed, blue] & & \color{blue}{(B_a \mid a\in A)} \arrow[ul, dashed, blue] &
\end{tikzcd}
$$

Now let $(X_c \mid c\in C)$ be an object of $\mathfrak{C}/C$. On the one hand we have

$$\Pi_f\Sigma_u(X_c \mid c\in C) \equiv \left(\prod_{(b\in B_a)}\sum_{(c\in C_b)} X_c \,\middle|\, a\in A\right). \tag{3.2.1}$$

On the other hand, using our annotated diagram above, we may deduce that

$$\Sigma_{\Pi_f u}\Pi_g {\varepsilon_u}^*(X_c \mid c\in C) \equiv \left(\sum_{(l\in\Pi(b\in B_a),C_b)}\prod_{(b\in B_a)} X_{l(b)} \,\middle|\, a\in A\right). \tag{3.2.2}$$

It is the author's intention that this situation seem familiar. Indeed, the careful reader will be able to deduce that this is a generalisation and externalisation of the argument made in section 2.3.1 and that the equality asserted in equation (2.3.1) is a special case of the equality between equations (3.2.1) and (3.2.2) above.

Thus we are already familiar with the 'arithmetic' interpretation of this statement in the internal language: the distributive law for multiplication over addition. However, as we mentioned earlier, this may be read in another way.

Under the standard correspondence between quantifiers and adjoints, and suppressing the dependence upon A and setting $X = C$ so that $X_c = 1$, equation (3.2.1) may be read as "$\forall b\in B, \exists c\in C_b$". In this light equation (3.2.2) may be read as "$\exists l\in\Pi_{b\in B}C_b$" – where we have collapsed the product, $\Pi(b\in B), X_{l(b)} = \Pi(b\in B), 1 = 1$. The classical Axiom of Choice says that from an inhabited family of inhabited sets[5] we may derive a choice function. We have just seen that lemma 3.3 asserts not only that the antecedent implies the consequent, but that they are in fact *equal*.

Of course like any good magic this is fundamentally a trick, and we were conniving in our efforts to relegate the enabling equivocation to a footnote – a place no reader would ever look. Shame on us.

While it is true that classically the propositions "inhabited" and "non-empty" are logically equivalent, they are not *constructively* equivalent. Herein lies the trick. At the risk of angering the reader with opinions about the Axiom of Choice – opinions one way or another –, a reason that a choice principle might seem appealing is that

[5] We prefer the term "inhabited" over the doubly negative "non-empty". These are classically equivalent.

we intuitively view it in such a constructive manner: if a mathematician is positively able to demonstrate an inhabitant of every set in the family, then that mathematician *is* a choice function. In such cases where to prove inhabitance involves an explicit construction we have already assembled all the data of a choice function – this is the statement of lemma 3.3, though several times generalised. It is only when we begin to non-constructively demonstrate that the sets are inhabited – that their being empty would lead to a contradiction – that choice functions might seem strange.

With our magic systematically deconstructed and our guile laid bare, we leave ourselves no choice but to press on to the theory of polynomials in general, in the hope that we may yet reestablish the trust the reader had once so willingly placed in us.

4. The view

In the penultimate section we saw how, contingent upon a now proven claim, given two polynomials in a category we may find a third whose extension to a functor agrees with the composite of the extensions of the first two. Unfortunately, our rather direct approach to the matter did not seem to uncover any readily recognisable results which might aid us in further generalisations.

That is to say, we are not satisfied with our definition of polynomial just yet. A first unnecessary constraint is that of univariance – polynomials ought to be allowed to combine multiple input variables.

As before, in order to arrive at a form more readily amenable to generalisation we must restate the familiar in a slightly different manner. In this case we repeat our earlier translation of $\Sigma(n \in N), a_n x^n = \Sigma(n \in N), x^{b_n}$ but in the multivariate case. That is, we begin with the familiar form of

$$\sum_{(n_1,\dots,n_k,\in N)} a_{\vec{n}} \cdot {x_1}^{n_1} \cdots {x_k}^{n_k} = \sum_{(n_1,\dots,n_k,\in N)} a_{\vec{n}} \cdot \prod_{(i \in I)} {x_i}^{n_i},$$

and rewrite it instead as

$$\sum_{(n \in N)} \prod_{(i \in I)} {x_i}^{b_{n,i}}.$$

An instance of such a reshuffling of information might be the polynomial $2{x_1}^2{x_2}^1 + {x_2}^4$ – in the coefficient sense. To such a collection of coefficients we might associate $N :\equiv \{0, 1, 2\}$ and the matrix $b \in \{1, 2\}^{3\times 2}$ of the form

$$\begin{bmatrix} 2 & 1 \\ 2 & 1 \\ 0 & 4 \end{bmatrix}.$$

This is of course not the only representation, but under our formalism this does give rise to the same multivariate polynomial. While this works, this is not yet the most convenient form for us.

We had previously reasoned that x^k should be a repeated product of x with itself k times and so we recast $x^k = \Pi(b \in B), x$ where $|B| = k$. How then are we to interpret expressions of the form $\Pi(i \in I), {x_i}^{b_{n,i}}$?

A naïve attempt at the same formalism would leave i unbound in $\Pi(b \in B), x_i$, but all is not lost. The exponents $b_{n,i}$ "knew" the variable index to which they were associated, and so under the interchange of exponent and repeated product, the elements $b \in B$ must now "know" the same thing. That is, we can recover fully the data of the $b_{n,i}$ should we supply an additional function $s : B \to I$ and pose the form $\Pi(b \in B), x_{s(b)}$ instead.

Altogether, the *equivalent* form of polynomials we shall generalise is that of

$$s : B \to I, \qquad \sum_{(\, n \in N\,)} \prod_{(\, b \in B_n\,)} x_{s(b)} \quad .$$

To tie this in to the previous form, consider that B is now indexed by $N \times I$ and *behaves* as though it was a matrix, but we have lost the rigidity constraints that all rows are the same length and that all columns have entries in all indices!

As a concluding example of this rephrasing, we may recapture the same polynomial as before, $2{x_1}^2{x_2}^1 + {x_2}^4$, by defining

$$\begin{aligned} N &:\equiv \{0, 1, 2\} & I &:\equiv \{1, 2\} \\ B_0 :\equiv B_1 &:\equiv \{(0,1),(1,1),(2,2)\} & B_2 &:\equiv \{(0,2),(1,2),(2,2),(3,2)\} \\ B &:\equiv B_0 \sqcup B_1 \sqcup B_2 & &\text{and } s \text{ as the evident projection.} \end{aligned}$$

4.1. General polynomials

There is one final generalisation to which we must attend before stating the definition of a polynomial in a category at the level of detail we desire. While multivariate polynomials are common place in mathematics, the somehow dual notion of indexed families of polynomials is not often thought of as related. We wish to capture both of these aspects in our framework and so move to make the following definition.

Def. 4.1. A *polynomial P* in a locally cartesian-closed category $\mathfrak{C}$ is a diagram in $\mathfrak{C}$ of the form

$$I \xleftarrow{s} A \xrightarrow{f} B \xrightarrow{t} J.$$

The *extension of a polynomial P* to a functor $\mathrm{Ext}_P : \mathfrak{C}/I \to \mathfrak{C}/J$ is defined to be the composite $\Sigma_t \circ \Pi_f \circ s^*$. Given a functor we may sometimes be concerned only with the *property* of being the extension of some polynomial, and in these cases we may term such a functor a *polynomial functor*. ⌟

Let us elaborate the action of Ext_P on $(X_i \mid i \in I)$ of $\mathfrak{C}/I$. By unwinding definitions we see that the result is $(\Sigma(a \in A_j), \Pi(b \in B_a), X_{s(b)} \mid j \in J)$ – an indexed family of multivariate polynomial functions evaluated on an indexed family of input variables.

Specialising to the case of $J = 1$ recovers the context of the immediately prior discussion, and specialising further to $I = 1$ constrains our definition to that of section 2.

Remark 4.2. As a consequence of lemma 3.8, polynomial functors preserve connected limits and are, in particular, cartesian. ◃

Example 4.3

Polynomials of the form $I \xleftarrow{s} M = M \xrightarrow{t} J$ are termed *linear polynomials* and functors which are the extension of linear polynomials are termed *linear functors* or *matrices*.

To see that this terminology is justified, first consider that M may be thought of as an internal matrix itself, $(M_{j,i} \mid (j,i) \in J \times I)$, so that $M_j = \Sigma(i \in I), M_{j,i}$. Moreover $\Pi_{\mathrm{id}_M} = \mathrm{id}_{\mathfrak{C}/M}$ and so if L is a linear polynomial then we may recast $\mathrm{Ext}_L(X_i \mid i \in I) \equiv (\Sigma(m \in M_j), X_{s(m)} \mid j \in J) = (\Sigma(i \in I), \Sigma(m \in M_{j,i}), X_i \mid j \in J) = (\Sigma(i \in I), M_{j,i} \times X_i \mid j \in J)$. In the case that $I = M = J$ and $s = t = \mathrm{id}_M$ we recover the usual identity linear polynomial and its extension, the identity matrix.

An excellent example of linear polynomials is provided by internal category objects. If $C = (C_0, C_1, \partial_0, \partial_1, \mathrm{id}, \circ)$ is an internal category object then let FP_C be the linear polynomial $(\partial_0, \mathrm{id}_{C_1}, \partial_1)$. By our prescription we should view C_1 as the usual matrix $(C_1(a,b) \mid (b,a) \in C_0 \times C_0)$. With that, if $(X_a \mid a \in C_0)$ is an object of $\mathfrak{C}/C_0$ then $\mathrm{Ext}_{FP_C}(X_a \mid a \in C_0) = (\Sigma(a \in C_0), C_1(a,b) \times X_a \mid b \in C_0)$ which is nothing but the object underlying the free internal presheaf on $(X_a \mid a \in C_0)$.

Example 4.4

Another example which bears mentioning is the polynomial FM in Set defined by

$$1 \leftarrow N' \xrightarrow{f} \mathbb{N} \rightarrow 1$$

in which $|f^{\leftharpoonup}\{n\}| = n$. Should we expand the action of Ext_{FM} on (X) of $\mathfrak{C}/1$ we see that it is precisely the underlying functor of the free monoid monad on Set. That is, $\mathrm{Ext}_{FM}(X) = (\Sigma(n \in \mathbb{N}), X^n)$.

Non-example 4.5

Not every functor is polynomial, as we may see in cases where we have some appropriate notion of size. In particular, on Set, covariant powerset $\mathcal{P} : \mathrm{Set} \to \mathrm{Set}$ is not the extension of a polynomial. To see this, if $f : A \to B$ is an arrow of Set considered as a polynomial P of the form $1 \leftarrow A \to B \to 1$, then for suitably large X we have $|\mathrm{Ext}_P(X)| = |\Sigma(b \in B), X^{A_b}| \leq |B \times X^A| < |\mathcal{P}X|$.

Now that we have suitably generalised polynomials and their extension, we turn to the generalised version of our composition theorem of section 2.3.

4.2. Composition of polynomials

Def. 4.6. Given two polynomials, $P = \{I \xleftarrow{s} B \xrightarrow{f} A \xrightarrow{t} J\}$ and $Q = \{J \xleftarrow{u} D \xrightarrow{g} C \xrightarrow{v} K\}$ in a locally cartesian-closed category $\mathfrak{C}$, we define the composite polynomial to be

$$Q \circ P :\equiv \left\{ I \xleftarrow{spq} (h\varepsilon)^* B \xrightarrow{((\Pi_g k)^* g)((h\varepsilon)^* f)} \Pi_g u^* A \xrightarrow{v \Pi_g k} K \right\}$$

where the arrows and morphisms arise from the following diagram:

$$\begin{tikzcd}
 & & & (h\varepsilon)^*B \arrow[rr, "(h\varepsilon)^*f"] \arrow[dl, "q"'] & & g^*\Pi_g u^*A \arrow[r, "(\Pi_g k)^*g"] \arrow[dl, "\varepsilon"'] \arrow[dd] & \Pi_g u^*A \arrow[dd, "\Pi_g k"] & \\
 & & h^*B \arrow[rr, "h^*f"] \arrow[dl, "p"'] & & u^*A \arrow[dl, "h"'] \arrow[dr, "k"] & & & \\
 & B \arrow[rr, "f"'] \arrow[dl, "s"'] & & A \arrow[dr, "t"'] & & D \arrow[r, "g"'] \arrow[dl, "u"] & C \arrow[dr, "v"] & \\
I & & & & J & & & K
\end{tikzcd}$$

wherein h^*B, $(h\varepsilon)^*B$, u^*A, and $g^*\Pi_g u^*A$ are the vertices of pullbacks. ⌟

Thm. 4.7. *(Composition of polynomials) Given two polynomials P and Q as above in a locally cartesian-closed category $\mathfrak{C}$, $\mathrm{Ext}_Q \circ \mathrm{Ext}_P \cong \mathrm{Ext}_{P \circ Q}$ as functors $\mathfrak{C}/I \to \mathfrak{C}/K$.*

Proof. (External) With reference to the diagram of we derive the following isomorphisms of functors.

$$\begin{aligned}
\mathrm{Ext}_Q \circ \mathrm{Ext}_P :\equiv \Sigma_v \Pi_g u^* \Sigma_t \Pi_f s^* &\cong \Sigma_v \Pi_g \Sigma_k h^* \Pi_f s^* && \text{(lemma 3.1)} \\
&\cong \Sigma_v \Sigma_{\Pi_g k} \Pi_{(\Pi_g k)^* g} \varepsilon^* h^* \Pi_f s^* && \text{(lemma 3.3)} \\
&\cong \Sigma_v \Sigma_{\Pi_g k} \Pi_{(\Pi_g k)^* g} \Pi_{(h\varepsilon)^* f} (spq)^* && \text{(lemma 3.1)} \\
&\cong \Sigma_{v \Pi_g k} \Pi_{((\Pi_g k)^* g)((h\varepsilon)^* f)} (spq)^* && \text{(functoriality)}
\end{aligned}$$

■

It is a good exercise to unpack this external isomorphism into an internal equality, in a manner generalising that of section 2.3. In particular, this proof specialises precisely to our previous arguments in the case $I = J = 1$.

4.3. A Yoneda excursus

At this point we turn to attend to an unfortunate restriction of theory. So far we have assumed that all categories involved are locally cartesian-closed categories. This prohibits a naïve application of our theory to, among other categories, Cat. There are two immediate remedies to this, and we mention one while exploring the other.

While Cat is not locally cartesian-closed, it is 'mostly so'. There exist characterisations of those functors for which pullback does have a right adjoint – the Conduché functors[6] –, and this class is evidently closed under composition and pullback. With this knowledge we might instead redo our theory in general by requiring that the maps whose right adjoint to pullback we need belong to a class axiomatised in this manner. This approach is that of the so termed 'exponentiable' maps, and provides a weakening on the part of our assumptions.

Exponentiable maps, the somewhat obvious response to 'what do we really need?', certainly suffice to give a stronger theory – see "Polynomials in categories with pullbacks" by M. Weber for such an account. However, among the many canonical tools of the working category theorist in dealing with generalisations we find the indispensable utility of working representably in a suitable 'virtualisation' of our structure.

Instead of directly working with objects and requirements imposed thereupon, we may embed them in a broader context in which those requirements are always satisfied but are not necessarily closed under representability *in such a way* so as to guarantee that whenever a construction culminates in an object that is representable, the result is canonically the representation of the analogous construction in the first context. This verbiage is equivalently the following construction and lemma.

Prob. 4.8. *Let $y : \mathfrak{C} \to \widehat{\mathfrak{C}}$ be the Yoneda embedding, construct for each $c \in \mathfrak{C}$ an equivalence of categories $P^C : \widehat{\mathfrak{C}}/yC \to \widehat{\mathfrak{C}/C}$ such that the following diagram commutes.*

$$\begin{tikzcd}
\widehat{\mathfrak{C}}/C \arrow[r, "P^C"] & \widehat{\mathfrak{C}/C} \\
\mathfrak{C}/C \arrow[u, "y^{\mathfrak{C}}/yC"] \arrow[ur, "y^{\mathfrak{C}/C}"'] &
\end{tikzcd}$$

Constr. 4.9. We begin by defining P^C on objects. Let $\alpha : F \Rightarrow yC$ be an object of $\widehat{\mathfrak{C}}/yC$, $f : D \to C$ and $g : E \to C$ be objects of $\mathfrak{C}/C$, and $h : f \to g$ be a morphism of $\mathfrak{C}/C$.

Define $P^C(\alpha)(f) :\equiv \alpha_D{}^{\leftarrow}\{f\} \subseteq FD$ and $P^C(\alpha)(h) :\equiv Fh{\restriction}_{\alpha_E{}^{\leftarrow}\{g\}}$ as the restriction of Fh, which is well-typed by the naturality of α. It is immediate that $P^C(\alpha)$ is a functor $(\mathfrak{C}/C)^{\mathrm{op}} \to \mathrm{Set}$.

Now let $\beta : G \Rightarrow yC$ be an object and $\theta : \alpha \Rightarrow \beta$ be a morphism of $\widehat{\mathfrak{C}}/yC$, and define $P^C(\theta)_f :\equiv (\theta_D){\restriction}_{\alpha_D{}^{\leftarrow}\{f\}} : P^C(\alpha)(f) \to P^C(\beta)(f)$ – this is well-typed by virtue of the equation $\beta\theta = \alpha$. Moreover, by the definition of $P^C(\alpha)$ on morphisms as the restriction of F, mutatis mutandis for β, and the assumed naturality of θ we see that these components assemble into a natural transformation $P^C(\theta) : P^C(\alpha) \Rightarrow P^C(\beta)$. Finally functoriality of P^C is clear from the definition.

[6] Although this notion appears to have been originally developed by Giraud

Next we prove that the diagram above commutes. Let us expand on objects $g : E \to C$ of $\mathfrak{C}/C$ as $(P^C \circ (y^{\mathfrak{C}}/yC))(g) \equiv P^C(y^{\mathfrak{C}}g)$. To give the equality of this functor to $y^{\mathfrak{C}/C}(g)$ we must further expand on objects and morphisms of $(\mathfrak{C}/C)^{\mathrm{op}}$.

On objects $f : D \to C$ we have $P^C(y^{\mathfrak{C}}g)(f) \equiv ((y^{\mathfrak{C}}g)_D)^{\leftharpoonup}\{f\} \equiv \{h \in \mathfrak{C}(D,E) \mid gh = f\} \equiv (\mathfrak{C}/C)(f,g) \equiv y^{\mathfrak{C}/C}(g)(f)$ and on morphisms the equality is straightforward.

Thus $(P^C \circ (y^{\mathfrak{C}}/yC))$ agree on objects and a similar computation shows they agree on morphisms, so the diagram commutes.

By leveraging the idea that functions of sets are entirely determined by their values on points and elaborating definitions we may check that P^C as defined above is fully faithful. We will conclude the construction by *constructively* demonstrating that it is essentially surjective[7].

To this end, fix a functor $K : (\mathfrak{C}/C)^{\mathrm{op}} \to \textsc{Set}$. We will construct from this K the complete data of a natural transformation $\alpha : F \Rightarrow yC$ such that $P^C(\alpha) = K$. We begin with the functor F.

Let $F : (\mathfrak{C})^{\mathrm{op}} \to \textsc{Set}$ be defined on objects as $F(D) :\equiv \coprod_{f \in \mathfrak{C}(D,C)} K(f)$ and on morphisms $h : D \to E$ of $\mathfrak{C}$ via the diagram below. It is straightforward to check that this defines a functor, and from this we define $\alpha_D(f,k) :\equiv f$. Naturality follows easily, and we see that by definition $P^C(\alpha)(f : D \to C) :\equiv {\alpha_D}^{\leftharpoonup}\{f\} = K(f)$ and $P^C(\alpha)(h) = Kh$, thereby concluding the construction.

$$\begin{tikzcd}
\coprod_{g' \in \mathfrak{C}(E,C)} K(g') \arrow[r, dashed, "Fh"] & \coprod_{f' \in \mathfrak{C}(D,E)} K(f') \\
K(g) \arrow[u, "\iota_g"] \arrow[r, "K(h)"'] & K(gh) \arrow[u, "\iota_{gh}"']
\end{tikzcd}$$

■

In this sense the Yoneda embedding *commutes* with the formation of slice categories. While undoubtedly there are more sophisticated ways of saying this, for our purposes this direct construction is sufficient. More still is true of the Yoneda embedding, but we leave the precise formulation and proof to the reader and supply only the below sketch.

[7]In this way the inverse functor may be constructed and we avoid any choice principles.

Lem. 4.10. *(Sketch) Let $\mathfrak{C}$ be a category and $f : A \to B$ a morphism of $\mathfrak{C}$. Then the leftmost diagram commutes, where P^B is the equivalence of construction 4.9. Moreover, whenever the functors f^* or Π_f exist, there exist isomorphisms filling in the respective diagrams below such that these isomorphisms paste with the functoriality isomorphisms of $(-)^*$ and $\Pi_{(-)}$ coherently.*

$$\begin{tikzcd} & \mathfrak{C}/A \arrow[dl, "y^{\mathfrak{C}}/yA"'] \arrow[dr, "\Sigma_f"] & \\ \widehat{\mathfrak{C}}/y^{\mathfrak{C}}A \arrow[d, "\Sigma_{y^{\mathfrak{C}}f}"'] & & \mathfrak{C}/B \arrow[d, "y^{\mathfrak{C}/B}"] \\ \widehat{\mathfrak{C}}/y^{\mathfrak{C}}B \arrow[rr, "P^B"'] & & \widehat{\mathfrak{C}/B} \end{tikzcd} \qquad \begin{tikzcd} & \mathfrak{C}/B \arrow[dl, "y^{\mathfrak{C}}/yB"'] \arrow[dr, "f^*"] & \\ \widehat{\mathfrak{C}}/y^{\mathfrak{C}}B \arrow[d, "(y^{\mathfrak{C}}f)^*"'] & & \mathfrak{C}/A \arrow[d, "y^{\mathfrak{C}/A}"] \\ \widehat{\mathfrak{C}}/y^{\mathfrak{C}}A \arrow[rr, "P^A"'] & & \widehat{\mathfrak{C}/A} \end{tikzcd} \qquad \begin{tikzcd} & \mathfrak{C}/A \arrow[dl, "y^{\mathfrak{C}}/yA"'] \arrow[dr, "\Pi_f"] & \\ \widehat{\mathfrak{C}}/y^{\mathfrak{C}}A \arrow[d, "\Pi_{y^{\mathfrak{C}}f}"'] & & \mathfrak{C}/B \arrow[d, "y^{\mathfrak{C}/B}"] \\ \widehat{\mathfrak{C}}/y^{\mathfrak{C}}B \arrow[rr, "P^B"'] & & \widehat{\mathfrak{C}/B} \end{tikzcd}$$

Thus we may instead transfer all of theory of polynomials from $\mathfrak{C}$ to $\widehat{\mathfrak{C}}$ with the knowledge that whenever the underlying construction in $\mathfrak{C}$ may be performed, it represents the same construction but executed representably in $\widehat{\mathfrak{C}}$, and thereby require *nothing* of $\mathfrak{C}$ at all while still retaining the influence of its structure on our results.

5. This is the last section

The reader who has either understood the content to this point, or determined that it is in sore need of improvement, is now – it is hoped – fully equipped to read first-hand the theory of polynomials as developed by Gambino-Kock. From here any further explanations would come to little more than a restating of parts of the paper, indeed even the examples and theorem 4.7 are but lifted from the document.

Gambino-Kock go on to develop notions of morphism of polynomial, prove characterisation theorems on such, collect the totality of polynomials in a category into a variety of structures, and explore the intersection of monads and polynomial functors.

These notes arose as an attempt to write down the introductory portion of a talk on polynomials[8] so that others might enjoy a shallower road to understanding the theory. It is the author's hope that the reader who has read this far will continue reading into polynomials and develop and apply them further still.

[8] whose latter portion was not given in full because the author is terrible at making appropriate length talks and Emily if you're reading this i promise i'm trying, and to Daniel, David, Martina, Naruki, and Tomas i'm sorry.