Pattern runs on matter: The free monad monad as a module over the cofree comonad comonad

Sophie Libkind and David I. Spivak

Interviews run on people, programs run on operating systems, voting schemes run on voters, games run on players. Each of these is an example of the abstraction *pattern runs on matter*. Pattern determines the decision tree that governs how a situation can unfold, while matter responds with decisions at each juncture.

In this article, we will give a straightforward and concrete construction of the free monad monad for $(\mathbf{Poly}, \triangleleft, y)$, the category of polynomial functors with the substitution monoidal product. Although the free monad has been well-studied in other contexts, the construction we give is streamlined and explicitly illustrates how the free monad represents terminating decision trees. We will also explore the naturally arising interaction between the free monad and cofree comonad. Again, while the interaction itself is known, the perspective we take is the free monad as a module over the cofree comonad. Lastly, we will give four applications of the module action to interviews, computer programs, voting, and games. In each example, we will see how the free monad represents pattern, the cofree comonad represents matter, and the module action represents *runs on*.

1 Introduction

The etymology of matter and pattern are "mother" and "father"; this pair of terms offers a very basic sense in which to carve up the world. “Like two parents, matter and pattern represent a fundamental dichotomy: matter is the pure material, unconcerned with our ideas about it; pattern is pure structure, unconcerned with what substantiates it” [Spi]. And yet one may have an intuitive sense that pattern "runs on"—must be instantiated in—matter. In this paper, we show that this idea matches both our intuition and the mathematics of a module structure by which free monads are a module over cofree comonads.

Intuitively, interviews, programs, voting schemes, and games represent patterns. But what is an interview without a person to be interviewed; what is a program without a operating system to run it on; what is a voting scheme without voters to do the voting; what is a game without players to play it? In each case, the pattern runs on a material substrate.

In this paper we give an account for this intuition in terms of free monads $\mathfrak{m}$ and cofree comonads $\mathfrak{c}$ on polynomial functors. We show that for any $p, q : \mathbf{Poly}$, there is a natural map

$$\Xi_{p,q} \colon \mathfrak{m}_p \otimes \mathfrak{c}_q \to \mathfrak{m}_{p \otimes q}. \tag{1}$$

In fact, this map gives $\mathfrak{m}$ the structure of a $\mathfrak{c}$-module, in a precise sense; see Section 3.2.

Consider the behavior of a Moore machine, which transforms lists of A-inputs into lists of B-outputs for sets A, B. This behavior is determined by an element $b \colon y \to \mathfrak{c}_{By^A}$ of the terminal coalgebra on the polynomial $q := By^A$. But how does one apply this behavior to a list of A's? The latter is a map $\ell \colon y \to \mathfrak{m}_{Ay}$. To actually apply the behavior b to the list ℓ, we use an obvious map $\varphi \colon Ay \otimes By^A \to By$ and the module structure (1) to obtain a list of B's.

$$y \cong y \otimes y \xrightarrow{\ell \otimes b} \mathfrak{m}_{Ay} \otimes \mathfrak{c}_{By^A} \xrightarrow{\Xi_{Ay,By^A}} \mathfrak{m}_{Ay \otimes By^A} \xrightarrow{\mathfrak{m}_\varphi} \mathfrak{m}_{By} \tag{2}$$

Whereas patterns start and end, matter is never destroyed. One can think of $\mathfrak{m}_p$ as the type of terminating programs and of $\mathfrak{c}_q$ as the type of operating systems or online algorithms. This intuition is captured by the fact that elements of the free monad $\mathfrak{m}_p$ are "well-founded trees" [GK12; AGM01]—in the case p is finitary, a wellfounded tree is one with finite height—whereas elements of the cofree comonad $\mathfrak{c}_q$ are generally non-wellfounded, e.g. infinite in height even for finitary q. The module map $\Xi_{p,q}$ pairs the wellfounded tree with the non-wellfounded tree, following the shape of the wellfounded one; for example the list of B's in (2) will have exactly the same length as the list of A's.

Related work. A construction of the free monad for much more general (than polynomial) endofunctors was given by Kelly in [Kel80]. The treatment was greatly simplified by Shulman and others in [nLa24c]. The case of polynomial endofunctors is more restrictive, and as such has a far more straightforward construction.

A monad-comonad interaction law, again in more general settings, was described in [KRU20]. This paper structures interaction laws in an interesting and useful but somehow ad hoc way, as a category whose objects are triples (T, C, f), where T is a monad, C is a comonad, and $f : T \otimes C \to \mathrm{id}$ is a map satisfying certain natural conditions, and whose morphisms are maps $T \to T'$ and $C' \to C$ satisfying certain constraints. It should also be noted that this paper includes no proofs.

Contributions.
1. Simple concrete constructions of both free monads and cofree comonads in $(\mathbf{Poly}, \triangleleft, y)$.
2. A proof that the free monad is a module over the cofree comonad.
3. Four applications of this module, each having the form *pattern runs on matter*.

For these, see Definition 2.2 and Proposition 3.1; Theorem 3.4; and Section 4, respectively. We also give a precise definition of the notion of "dual" functor, defined in [KRU20], and generalize it to be functorial in a monad t; see (5).

Notation. We often denote the identity on an object x by the object name itself rather than id_x. We denote the cardinality of a set X by $\#X$. If $L \dashv R$ is an adjunction, we denote it

$$\mathcal{C} \underset{R}{\overset{L}{\underset{\longleftarrow}{\overset{\longrightarrow}{\Rightarrow}}}} \mathcal{D}$$

so that the 2-cell shown indicates the direction of both the unit $\mathcal{C} \to R \circ L$ and the counit $L \circ R \to \mathcal{D}$.

Acknowledgements. The authors thanks Harrison Grodin and Brandon Shapiro for many useful conversations. This material is based upon work supported by the Air Force Office of Scientific Research under award numbers FA9550-20-1-0348 and FA9550-23-1-0376.

1.1 Preliminaries

Although we will assume basic familiarity with the category **Poly** of univariate polynomial functors on **Set** and natural transformations between them, we will begin by clarifying notation. We write a polynomial functor $p : \mathbf{Poly}$ as $p = \sum_{P:p(1)} y^{p[P]}$ where $p(1)$ are the **positions** of p and $p[P]$ are the **directions** at the position P. With this notation, a map $\phi : p \to q$ in **Poly** consists of

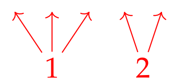
- A function on positions $\phi(1) : p(1) \to q(1)$.
- For each position $P : p(1)$, a function backwards on directions $\phi^\#_P : q[\phi_1(P)] \to p[P]$.

There are many monoidal products in **Poly** [NS22], however in this article we focus on the Dirichlet and substitution products. The substitution product has unit y and is defined by

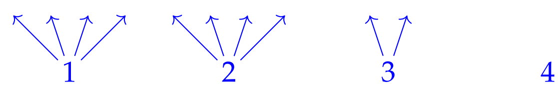
$$p \triangleleft q = \sum_{P:p(1)} \prod_{p[P]} \sum_{Q:q(1)} \prod_{q[Q]} y$$

Using the distributive law, a position in $p \triangleleft q$ is a position P in p and for every direction in $p[P]$ a position in q. It will be useful to consider the corolla forest view on polynomials and the $\triangleleft$-monoidal product, since these will be the building blocks of the decision trees represented by the free monad monad.

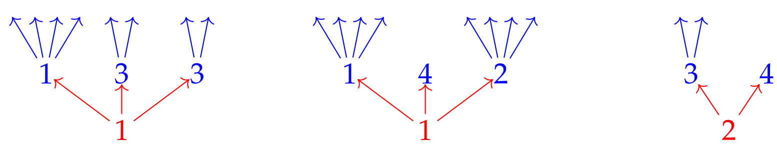
Consider $p = y^3 + y^2$. We can notate p using a corolla forest in which each position $P : p(1)$ is a tree with $p[P]$-many branches. The root of each tree is labeled by its corresponding position and branches correspond to directions.



Likewise, the polynomial $q = 2y^4 + y^2 + 1$ is notated by the corolla forest:



Then the positions of $p \triangleleft q$ can be represented by trees of height 2, whose building blocks are the trees corresponding to the positions of p and q. In particular, a position of $p \triangleleft q$ is a position of p and for each branch, a position of q. The directions are the number of dangling leaves. Here are three examples of positions in $p \triangleleft q$ with eight, eight, and two directions respectively.



2 The free monad monad

For a polynomial p, we will define the free monad $\mathfrak{m}_p$ through transfinite induction by defining polynomials $p_{(\alpha)}$ for each ordinal α and inclusions $p_{(\alpha)} \to p_{(\beta)}$ for ordinals $\alpha < \beta$.

- Base case: Define $p_{(0)} := y$.
- For successor ordinals $\alpha + 1$: Define $p_{(\alpha+1)} := y + p \triangleleft p_{(\alpha)}$. For the inclusions define $\iota_{(0)} : y \to y + p$ to be the inclusion; note that it is cartesian. Then, given $\iota_{(\alpha)}$, define $\iota_{(\alpha+1)}$ to be

$$p_{(\alpha+1)} = y + p \triangleleft p_{(\alpha)} \xrightarrow{y+p\triangleleft\iota_{(\alpha)}} y + p \triangleleft p_{(\alpha+1)} = p_{(\alpha+2)}.$$

  which is also cartesian.
- For limit ordinals α: Define $p_{(\alpha)} := \operatorname{colim}_{\alpha'<\alpha} p_{(\alpha')}$. For each $\alpha' < \alpha$, let $\iota_{(\alpha')} \colon p_{(\alpha')} \to p_{(\alpha)}$ be the natural inclusion. These are cartesian by Proposition A.7. Lastly, we define the inclusion $\iota_{(\alpha)} \colon p_{(\alpha)} \to p_{(\alpha+1)}$ to be induced by the cocone of maps which are defined for $\alpha' < \alpha$ by

$$p_{(\alpha')} \xrightarrow{\iota_{(\alpha')}} p_{(\alpha'+1)} = y + p \triangleleft p_{(\alpha')} \xrightarrow{y+p\triangleleft\iota_{(\alpha')}} y + p \triangleleft p_{(\alpha)} = p_{(\alpha+1)}.$$

Remark 2.1. By construction each $\iota_{(\alpha)}$ is cartesian. ◊

For ease of notation, we will also use $\iota_{(\alpha)}\colon p_{(\alpha)} \to p_{(\beta)}$ to represent the composite of inclusions for any pair of ordinals $\alpha < \beta$. We will also define $\lambda_{(\alpha)}\colon p \triangleleft p_{(\alpha)} \to p_{(\alpha+1)}$ to be the coproduct inclusion.

Definition 2.2. If p is κ-small, then define $\mathfrak{m}_p := p_{(\kappa)}$. ◊

We will show in Theorem 2.10 that $\mathfrak{m}_p$ is the free monad on p. Therefore, $\mathfrak{m}_p$ is unique and hence well-defined.

Think of the positions of $\mathfrak{m}_p$ as decision trees whose building blocks are the positions of p. For instance, suppose that a position $P : p(1)$ is a question and the directions $p[P]$ are possible answers to question P. Then y is a unique question with a unique answer, which we think of as "no further questions". Then, consider the positions of $\mathfrak{m}_p$ that factor through $p_{(\alpha)}$. For example, the positions of $p_{(2)} = y + p \triangleleft (y + p)$ is either (1) no further questions or (2) a question and for each possible answer either no further questions or another question. Therefore $p_{(2)}$ represents interviews with *at most two questions*, and in general for finite i, $p_{(i)}$ represents interviews with at most i questions.

Example 2.3. For finitary p, the positions of $\mathfrak{m}_p$ are p-trees with finite height and whose directions are the dangling leaves. ◊

Example 2.4. A finite y-tree is determined by its height and has one dangling leaf; thus, $\mathfrak{m}_y \cong \mathbb{N}y$. ◊

2.1 Monad structure on $\mathfrak{m}_p$

Let **Mod**(**Poly**) denote the category of monoids in (**Poly**, y, $\triangleleft$). Note that when viewed as endofunctors on **Set**, a $\triangleleft$-monoid is in fact a monad on **Set**. Therefore we refer to the objects of **Mod**(**Poly**) as **$\triangleleft$-monoids**, **polynomial monads**, or simply **monads**.

Next we will give a $\triangleleft$-monoid structure on $\mathfrak{m}_p$. The unit $\eta\colon y \to \mathfrak{m}_p$ is defined to be the inclusion $\iota_{(0)}\colon y = p_{(0)} \to \mathfrak{m}_p$. The multiplication is more complicated so we begin with the following Lemma.

Lemma 2.5. *For ordinals α, β there exist maps $\mu_{(\alpha,\beta)}\colon p_{(\alpha)} \triangleleft p_{(\beta)} \to p_{(\alpha+\beta)}$ such that for all $\alpha' < \alpha$ and $\beta' < \beta$ the following diagram commutes*

$$\begin{array}{ccc} p_{(\alpha')} \triangleleft p_{(\beta')} & \xrightarrow{\mu_{(\alpha',\beta')}} & p_{(\alpha'+\beta')} \\ {\scriptstyle \iota_{(\alpha')} \triangleleft \iota_{(\beta')}} \downarrow & & \downarrow {\scriptstyle \iota_{(\alpha'+\beta')}} \\ p_{(\alpha)} \triangleleft p_{(\beta)} & \xrightarrow[\mu_{(\alpha,\beta)}]{} & p_{(\alpha+\beta)} \end{array} \tag{3}$$

Proof. We define the maps $\mu_{(\alpha,\beta)}$ transinductively on α.

For $\alpha = 0$, $\mu_{(\alpha,\beta)}$ is the identity on $p_{(\beta)}$.

For successor ordinals $\alpha + 1$, suppose we have already defined $\mu_{(\alpha,\beta)}$. Note that

$$p_{(\alpha+1)} \triangleleft p_{(\beta)} = (y + p \triangleleft p_{(\alpha)}) \triangleleft p_{(\beta)} = p_{(\beta)} + p \triangleleft p_{(\alpha)} \triangleleft p_{(\beta)}.$$

So we define $\mu_{(\alpha+1,\beta)}$ to be the copairing of the inclusion $\iota_{(\beta)}\colon p_{(\beta)} \to p_{(\alpha+\beta)}$ with the map

$$p \triangleleft p_{(\alpha)} \triangleleft p_{(\beta)} \xrightarrow{p \triangleleft \mu_{(\alpha,\beta)}} p \triangleleft p_{(\alpha+\beta)} \xrightarrow{\lambda_{(\alpha+\beta)}} p_{(\alpha+\beta+1)}.$$

That the diagram in (3) commutes for $\alpha' = \alpha + 1$ can be shown inductively on α.

Next suppose that α is a limit ordinal and suppose we have defined $\mu_{(\alpha',\beta)}\colon p_{(\alpha')} \triangleleft p_{(\beta)} \to p_{(\alpha'+\beta)}$ for $\alpha' < \alpha$. Then we define $\mu_{(\alpha,\beta)}$ to be the composite

$$p_{(\alpha)} \triangleleft p_{(\beta)} = \left(\operatorname*{colim}_{\alpha'<\alpha} p_{(\alpha')}\right) \triangleleft p_{(\beta)} \cong \operatorname*{colim}_{\alpha'<\alpha}(p_{(\alpha')} \triangleleft p_{(\beta)}) \to \operatorname*{colim}_{\alpha'<\alpha} p_{(\alpha'+\beta)} \to p_{(\alpha+\beta)}.$$

The second isomorphism follows from Proposition A.8. That these maps make the diagram in (3) commute is immediate. □

This Lemma implies that for ordinals $\alpha, \beta < \kappa$ the maps

$$p_{(\alpha)} \triangleleft p_{(\beta)} \xrightarrow{\mu_{(\alpha,\beta)}} p_{(\alpha+\beta)} \to \mathfrak{m}_p$$

form a cocone into $\mathfrak{m}_p$. Define the multiplication $\mu \colon \mathfrak{m}_p \triangleleft \mathfrak{m}_p \to \mathfrak{m}_p$ using the universal property:

$$\mathfrak{m}_p \triangleleft \mathfrak{m}_p = \left(\operatorname*{colim}_{\alpha<\kappa} p_{(\alpha)}\right) \triangleleft \left(\operatorname*{colim}_{\beta<\kappa} p_{(\beta)}\right) \cong \operatorname*{colim}_{\alpha<\kappa} \operatorname*{colim}_{\beta<\kappa} (p_{(\alpha)} \triangleleft p_{(\beta)}) \to \mathfrak{m}_p.$$

The second isomorphism follows from Proposition A.3 and Proposition A.8.

From this definition and Lemma (2.5), the following Lemma is immediate.

Lemma 2.6. *Let p be κ-small. Then for all $\alpha, \beta < \kappa$, the following diagram commutes.*

$$\begin{array}{ccc}
p_{(\alpha)} \triangleleft p_{(\beta)} & \xrightarrow{\mu_{(\alpha,\beta)}} & p_{(\alpha+\beta)} \\
{\scriptstyle \iota_{(\alpha)} \triangleleft \iota_{(\beta)}} \big\downarrow & & \big\downarrow {\scriptstyle \iota_{(\alpha+\beta)}} \\
\mathfrak{m}_p \triangleleft \mathfrak{m}_p & \xrightarrow[\mu]{} & \mathfrak{m}_p
\end{array}$$

As a predecessor to proving the associativity law for $(\mathfrak{m}_p, \eta, \mu)$ as a $\triangleleft$-monoid, we present the following variant of Lemma 2.5 whose proof appears in Appendix B.

Lemma 2.7. *For ordinals α, β, γ, the following diagram commutes:*

$$\begin{array}{ccc}
p_{(\alpha)} \triangleleft p_{(\beta)} \triangleleft p_{(\gamma)} & \xrightarrow{p_{(\alpha)} \triangleleft \mu_{(\beta,\gamma)}} & p_{(\alpha)} \triangleleft p_{(\beta+\gamma)} \\
{\scriptstyle \mu_{(\alpha,\beta)} \triangleleft p_{(\gamma)}} \big\downarrow & & \big\downarrow {\scriptstyle \mu_{(\alpha,\beta+\gamma)}} \\
p_{(\alpha+\beta)} \triangleleft p_{(\gamma)} & \xrightarrow[\mu_{(\alpha+\beta,\gamma)}]{} & p_{(\alpha+\beta+\gamma)}
\end{array}$$

Proposition 2.8. *For every polynomial p :* **Poly***, there is a $\triangleleft$-monoid structure on $\mathfrak{m}_p$, for which the unit and multiplication $\eta \colon y \to \mathfrak{m}_p$ and $\mu \colon \mathfrak{m}_p \triangleleft \mathfrak{m}_p \to \mathfrak{m}_p$ are defined as above.*

Proof. First we show the left unit law. Since $y \triangleleft \mathfrak{m}_p$ is isomorphic to $\operatorname{colim}_{\beta<\kappa}(y \triangleleft p_{(\beta)})$, the left unit law follows from Lemma 2.6 with $\alpha = 0$. Likewise, for the right unit law.

Second we show associativity. Due to the isomorphism $\mathfrak{m}_p \triangleleft \mathfrak{m}_p \triangleleft \mathfrak{m}_p \cong \operatorname{colim}_{\alpha,\beta,\gamma<\kappa} \left(p_{(\alpha)} \triangleleft p_{(\beta)} \triangleleft p_{(\gamma)}\right)$, it suffices to show that for $\alpha, \beta, \gamma < \kappa$, the outer diagram in the following commutes.

$$\begin{array}{ccccc}
p_{(\alpha)} \triangleleft p_{(\beta)} \triangleleft p_{(\gamma)} & \xrightarrow{p_{(\alpha)} \triangleleft \mu_{(\beta,\gamma)}} & p_{(\alpha)} \triangleleft p_{(\beta+\gamma)} & \xrightarrow{\iota_{(\alpha)} \triangleleft \iota_{(\beta+\gamma)}} & \mathfrak{m}_p \triangleleft \mathfrak{m}_p \\
{\scriptstyle \mu_{(\alpha,\beta)} \triangleleft p_{(\gamma)}} \big\downarrow & & \big\downarrow {\scriptstyle \mu_{(\alpha,\beta+\gamma)}} & & \\
p_{(\alpha+\beta)} \triangleleft p_{(\gamma)} & \xrightarrow[\mu_{(\alpha+\beta,\gamma)}]{} & p_{(\alpha+\beta+\gamma)} & & \big\downarrow {\scriptstyle \mu} \\
{\scriptstyle \iota_{(\alpha+\beta)} \triangleleft \iota_{(\gamma)}} \big\downarrow & & & \searrow {\scriptstyle \iota_{(\alpha+\beta+\gamma)}} & \\
\mathfrak{m}_p \triangleleft \mathfrak{m}_p & & \xrightarrow[\mu]{} & & \mathfrak{m}_p
\end{array}$$

The upper left-hand square commutes by Lemma 2.7 and the other squares commute by Lemma 2.6. □

2.2 The monad $\mathfrak{m}_p$ is free

Now that we have verified that $\mathfrak{m}_p$ is indeed a monad, we want to justify calling it the **free monad** by giving a left adjoint $\mathfrak{m}_-\colon \mathbf{Poly} \to \mathbf{Mod}(\mathbf{Poly})$ to the forgetful functor $U\colon \mathbf{Mod}(\mathbf{Poly}) \to \mathbf{Poly}$ which takes a $\triangleleft$-monoid (q, η_q, μ_q) to its carrier q.

We begin by defining the action of $\mathfrak{m}_-$ on morphisms. Let $f\colon p \to q$ in **Poly**. We will define $\mathfrak{m}_f\colon \mathfrak{m}_p \to \mathfrak{m}_q$ by inductively defining morphisms $f_{(\alpha)}\colon p_{(\alpha)} \to q_{(\alpha)}$ such that for all $\alpha < \beta$ the following diagram commutes:

$$\begin{array}{ccc} p_{(\alpha)} & \xrightarrow{\iota_{(\alpha)}} & p_{(\beta)} \\ {\scriptstyle f_{(\alpha)}}\downarrow & & \downarrow{\scriptstyle f_{(\beta)}} \\ q_{(\alpha)} & \xrightarrow[\iota_{(\alpha)}]{} & q_{(\beta)} \end{array} \tag{4}$$

Define $f_{(0)}\colon p_{(0)} \to q_{(0)}$ to be the identity on y. For a successor ordinal $\alpha+1$, suppose that we have already defined $f_{(\alpha)}\colon p_{(\alpha)} \to q_{(\alpha)}$. Then we define $f_{(\alpha+1)}\colon p_{(\alpha+1)} \to q_{(\alpha+1)}$ to be

$$y + p \triangleleft p_{(\alpha)} \xrightarrow{y + f \triangleleft f_{(\alpha)}} y + q \triangleleft q_{(\alpha)}.$$

To show that the diagram in Equation (4) commutes for successor ordinals, it suffices to show that for all α, the diagram with $\beta = \alpha+1$ commutes. By induction this is immediate by the definitions of $\iota_{(\alpha+1)}$ and $f_{(\alpha+1)}$.

Suppose that α is a limit ordinal and that we have defined $f_{(\alpha')}\colon p_{(\alpha')} \to q_{(\alpha')}$ for all $\alpha' < \alpha$. For each $\alpha' < \alpha$ we have maps

$$p_{(\alpha')} \xrightarrow{f_{(\alpha')}} q_{(\alpha')} \xrightarrow{\iota_{(\alpha')}} q_{(\alpha)}.$$

Since the diagram in Equation (4) commutes for all pairs of ordinals less than α, these maps form a cocone into $q_{(\alpha)}$. So, by the universal property of the colimit, $p_{(\alpha)} = \operatorname{colim}_{\alpha' < \alpha} p_{(\alpha')}$, there is an induced map $f_{(\alpha)} : p_{(\alpha)} \to q_{(\alpha)}$. That the diagram in Equation (4) commutes when β is a limit ordinal follows from the uniqueness of the universal map.

Let κ be such that both p and q are κ-small. Then we define $\mathfrak{m}_f := f_{(\kappa)}$. We defer the proof that $\mathfrak{m}_f$ is a map of $\triangleleft$-monoids to Appendix B.

Proposition 2.9. *For $f : p \to q$ in* **Poly***, the polynomial map $\mathfrak{m}_f : \mathfrak{m}_p \to \mathfrak{m}_q$ is a map of $\triangleleft$-monoids.*

It is straightforward to show by transfinite induction that the action of $\mathfrak{m}_-$ on morphisms is functorial. Therefore, Proposition 2.8 and Proposition 2.9 define a functor $\mathfrak{m}_- : \mathbf{Poly} \to \mathbf{Mod}(\mathbf{Poly})$.

Theorem 2.10. *There is an adjunction*

$$\mathbf{Poly} \underset{U}{\overset{\mathfrak{m}_-}{\underset{\longleftarrow}{\overset{\longrightarrow}{\Rightarrow}}}} \mathbf{Mod}(\mathbf{Poly}) \ .$$

Therefore $\mathfrak{m}_p$ is the **free monad** on the polynomial p and $\mathfrak{m}_-$ is the **free monad monad**. We defer the proof of Theorem 2.10 and its requisite Lemmas to Appendix B.1.

3 Interactions between free monad and cofree comonad

3.1 The cofree comonad comonad

It is a beautiful fact that in **Poly** the $\triangleleft$-comonoids are categories and $\triangleleft$-comonoid maps are cofunctors. Thus, we use $\mathbf{Cat}^\sharp$ to denote the category of $\triangleleft$-comonoids and their maps. Dual to the construction of the free monad monad in Section 2, here we define the cofree comonad and show that it is right adjoint to the forgetful functor $U\colon \mathbf{Cat}^\sharp \to \mathbf{Poly}$ given by $U(c,\epsilon,\delta) \coloneqq c$. In Appendix C we prove the statements presented in this Section.

Proposition 3.1. *There is a functor $\mathfrak{c}_-\colon \mathbf{Poly} \to \mathbf{Poly}$ such that $\mathfrak{c}_p$ has the structure of a $\triangleleft$-comonoid for each $p : \mathbf{Poly}$,*

$$\mathfrak{c}_p \to y \qquad \text{and} \qquad \mathfrak{c}_p \to \mathfrak{c}_p \triangleleft \mathfrak{c}_p.$$

For a polynomial functor $q : \mathbf{Poly}$, the positions of the cofree comonad are q**-behavior trees**. These are defined coinductively as a position $Q : q(1)$ and a map from directions of $q[Q]$ to q-behavior trees.

Theorem 3.2. *There is an adjunction*

$$\mathbf{Cat}^\sharp \underset{\mathfrak{c}_-}{\overset{U}{\underset{\Leftarrow}{\rightleftarrows}}} \mathbf{Poly} \; .$$

Hence $\mathfrak{c}_q$ is the **cofree comonad** on the polynomial q and $\mathfrak{c}_-$ is the **cofree comonad comonad**.

3.2 The module structure $\mathfrak{m}_p \otimes \mathfrak{c}_q \to \mathfrak{m}_{p\otimes q}$

For polynomials p and q consider the map

$$p \otimes \mathfrak{c}_q \to p \otimes q \to \mathfrak{m}_{p\otimes q}$$

where the first map is induced by the counit of $\mathfrak{c}_q$ and the second map is the unit of $\mathfrak{m}_{p\otimes q}$. This composite induces a map of polynomials $p \to [\mathfrak{c}_q, \mathfrak{m}_{p\otimes q}]$. By duoidality of $\otimes$ and $\triangleleft$, the internal hom $[\mathfrak{c}_q, \mathfrak{m}_{p\otimes q}]$ is a $\triangleleft$-monoid as well. Therefore, by the adjunction in Theorem 2.10 the map of polynomials $p \to [\mathfrak{c}_q, \mathfrak{m}_{p\otimes q}]$ induces a map of $\triangleleft$-monoids $\mathfrak{m}_p \to [\mathfrak{c}_q, \mathfrak{m}_{p\otimes q}]$, which is equivalent to a polynomial map

$$\Xi_{p,q}\colon \mathfrak{m}_p \otimes \mathfrak{c}_q \to \mathfrak{m}_{p\otimes q}.$$

This map is the **free monad-comonad interaction law** described in [KRU20, Section 3.3]. In Appendix D we prove the statements presented in this Section.

Proposition 3.3. *The maps $\Xi_{p,q}$ are natural in p and q.*

Note that $\mathfrak{c}$ is lax monoidal (see Proposition C.1)

$$\mathfrak{c}_p \otimes \mathfrak{c}_q \to \mathfrak{c}_{p\otimes q} \qquad \text{and} \qquad y \to \mathfrak{c}_y$$

so in this sense we would say "matter can also take the place of pattern."

Recall from [nLa24b] the notion of a module over a monoidal functor.

Theorem 3.4. *There is a left-module over $\mathfrak{c}_- : (\mathbf{Poly}, \otimes, y) \to (\mathbf{Poly}, \otimes, y)$ consisting of:*

- **Poly** *as a left module category over $(\mathbf{Poly}, \otimes, y)$.*
- *The functor $\mathfrak{m}_-\colon \mathbf{Poly} \to \mathbf{Poly}$.*
- *The natural transformation $\Xi\colon \mathfrak{m}_- \otimes \mathfrak{c}_- \Rightarrow \mathfrak{m}_{-\otimes -}$.*

4 Applications

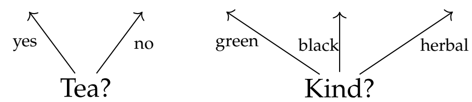
In this Section, we will give four applications of the module structure introduced in Section 3. Each consists of:

- A pattern of type p, represented by a map into $\mathfrak{m}_p$.
- Matter of type q, represented by a map into $\mathfrak{c}_q$.
- A *runs on* map $p \otimes q \to r$.

Then the composite $\mathfrak{m}_p \otimes \mathfrak{c}_q \to \mathfrak{m}_{p\otimes q} \to \mathfrak{m}_r$ represents the interaction *pattern runs on matter*.

4.1 Interviews run on people

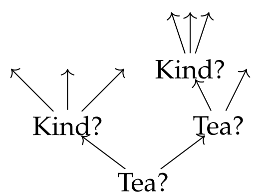
As in Section 2, suppose that the positions of p are questions and the directions are possible answers to each question. For example consider the polynomial, $p = \{\text{Tea?}\}y^{\{\text{yes,no}\}} + \{\text{Kind?}\}y^{\{\text{green,black,herbal}\}}$ which we can view as the corolla forest:



Essentially, p consists of the questions:

- "Do you want tea?", with possible answers "yes" or "no".
- "What kind of tea do you like?", with possible answers "green", "black", or "herbal".

Consider the pattern $y \to \mathfrak{m}_p$ which selects the following interview:



The polynomial $[p, y] = \left(\prod_{P:p(1)} p[P]\right) y^{p(1)}$ is the **universal answerer** for the polynomial p. Its positions are a choice of answer for each question in $p(1)$ and its directions are the questions $p(1)$. A person is a map $y \to \mathfrak{c}_{[p,y]}$. In other words, a person is a behavior tree in which a node is an answer for each question and a branch is one such question. Each answer may depend on the sequence of questions asked so far.

Note that $[p, y]$ is what is referred to in [KRU20, Section 2.5] as duals. We will see in Section 4.4 that it is also interesting to consider generalizations of this duality to $[p, t]$ for any polynomial monad t. Indeed, for any monad t, there is a map

$$\mathfrak{m}_p \otimes \mathfrak{c}_{[p,t]} \to \mathfrak{m}_t \to t \tag{5}$$

The evaluation map $p \otimes [p, y] \to y$ describes how to run a question on a universal answerer. Given an interview and a person we get the composite

$$y \to \mathfrak{m}_p \otimes \mathfrak{c}_{[p,y]} \to \mathfrak{m}_{p\otimes[p,y]} \to \mathfrak{m}_y = \mathbb{N}y,$$

which sends the single position of y to the number of questions asked when the interview is run on the person. For example, consider the person who would always respond that he does not want tea and he likes herbal tea. Running the tea interview on this person will cause two questions: "Tea?" then "Tea?". Conversely, consider the person who would at first respond that she does not want tea and that she likes black tea. Then after the first question, she would respond that she does want tea and she likes black tea. Running the tea interview on this person will cause three questions: "Tea?", "Tea?", then "Kind?".

4.2 Programs run on operating systems

Consider the following program.

```
def guessing_game(max_guesses, goal):
    if max_guesses==0:
        return False
    guess=read()
    if guess==goal:
        return True
    return guessing_game(max_guesses-1, goal)
```

We represent the argument/return type with the polynomial $r = \sum_{m:\mathbb{N},g:\mathbb{N}} y^{\mathrm{Bool}}$ where m represents the variable `max_guesses` and g represents the variable `goal`. We represent the effect type that reads in natural numbers with the polynomial $p = y^{\mathbb{N}}$. Finally, we will represent this program with a map $r \to \mathfrak{m}_p$ that we define inductively using the decomposition $r = \sum_{m:\mathbb{N}} r_m$ with $r_m = \sum_{g:\mathbb{N}} y^{\mathrm{Bool}}$. We start by defining the following maps:

- Define a map $r_0 \to y$. Each position of r_0 is sent to the single position of y. The single direction of y is sent to `False` : Bool.
- Define a map $r_{m+1} \to p \triangleleft (y + r_m)$. On positions, for each position $g : r_{m+1}(1)$, we need a function from $\mathbb{N}$ to the positions of $y + r_m$. In particular, we send g to the map sending $g' : \mathbb{N}$ to y if $g = g'$ and to the position $g : r_m(1)$ otherwise. Then on directions, if $g = g'$, the single direction of y is sent to `True` : Bool and if $g \neq g'$, then we use the identity on directions.

Now we will define maps $r_m \to \mathfrak{m}_p$ inductively using the isomorphism $\mathfrak{m}_p \cong y + p \triangleleft \mathfrak{m}_p$ and the inclusion $\iota_{(0)} \colon y \to \mathfrak{m}_p$. As a base case, we have the composite $r_0 \to y \xrightarrow{\iota_{(0)}} \mathfrak{m}_p$.Given $r_m \to \mathfrak{m}_p$, we define

$$r_{m+1} \to p \triangleleft (y + \mathfrak{m}_p) \xrightarrow{p \triangleleft (\iota_{(0)}, \mathfrak{m}_p)} p \triangleleft \mathfrak{m}_p \to \mathfrak{m}_p.$$

A operating system with effects in $[p, y] \cong \mathbb{N}y$ is a map $y \to \mathfrak{c}_{[p,y]} \cong (\mathbb{N}y)^{\mathbb{N}}$. It consists of a stream of natural numbers, which are the responses it will give to the `read()` effect.

Using the interaction $\Xi_{p,[p,y]}$ and the evaluation map $p \otimes [p, y] \to y$, we get the composite

$$r \cong r \otimes y \to \mathfrak{m}_p \otimes \mathfrak{c}_{[p,y]} \to \mathfrak{m}_{p \otimes [p,y]} \to \mathfrak{m}_y \cong \mathbb{N}y,$$

which expresses how the program runs on the chosen operating system. On positions it maps $(m : \mathbb{N}, g : \mathbb{N})$ to the minimum of m and the number of responses the operating system takes to guess the goal g. On directions, it maps the single direction to `True` if the goal was guessed in at most m guesses and to `False` otherwise.

4.3 Voting schemes run on voters

For a finite set of candidates X, consider the polynomial $p = \sum_{A \subseteq X} y^A$. The positions of p are ballots and the directions are winners. A voting scheme with M voters is a map $p \to \mathfrak{m}_{\bigotimes_M p}$ where $\bigotimes_0 p \coloneqq y$ and $\bigotimes_{M+1} p \coloneqq p \otimes \bigotimes_M p$. On positions, a subset of candidates $A \subseteq X$ maps to a terminating decision tree in which each node is a personalized ballot given to each of the M voters and each branch corresponds to the tuple of each voter's selection. On directions, each leaf of this decision tree maps to an overall winner.

Exhaustive run-off is a voting scheme in which each voter selects their preference from the A candidates, and then the candidates with the fewest number of votes are eliminated. If only a single candidate remains, then they are elected the winner. Otherwise, another round of voting proceeds with the remaining candidates. This voting scheme can be encoded into a polynomial map $p \to \mathfrak{m}_{\bigotimes_M p}$. As in Section 4.2 we define the map inductively. Consider the decomposition $p = \sum_{n=0}^{\#X} p_n$ where $p_n = \sum_{A \subseteq X, \#A = n} y^A$. Then consider the following maps:

- Note that $p_0 = 1$ is isomorphic to $\bigotimes_M p_0 = \bigotimes_M 1$. There is an inclusion of p_0 into p and hence a map $p_0 \cong \bigotimes_M p_0 \to \bigotimes_M p$.
- There is a unique map $p_1 \to y$.
- Define $p_{n+1} \to \left(\bigotimes_M p\right) \triangleleft \left(\sum_{k=0}^{n} p_k\right)$ as follows. On positions $A \subseteq X$ maps to the product $\prod_M A : \prod_M p(1)$ and the map $\prod_M A \to \sum_{k=0}^{n} p_k(1)$ defined by

$$(a_1, \cdots, a_M) \mapsto A' := A \setminus \arg\min \#_{(a_1,\cdots,a_n)}$$

where $\#_{(a_1,\cdots,a_n)} : A \to \mathbb{N}$ counts the number of votes for each candidate and so $\arg\min \#_{(a_1,\cdots,a_n)}$ is the *set* of candidates with the fewest votes. On directions, it is the inclusion $A' \to A$.

Now, we define $p \to \mathfrak{m}_{\bigotimes_M p}$ inductively as follows. As base cases we have the composites

$$p_0 = 1 = \bigotimes_M 1 \to \bigotimes_M p \to \mathfrak{m}_{\bigotimes_M p} \qquad \text{and} \qquad p_1 \to y \to \mathfrak{m}_{\bigotimes_M p}.$$

Given a maps $p_k \to \mathfrak{m}_{\bigotimes_M p}$ for $k < n+1$, we define

$$p_{n+1} \to \left(\bigotimes_M p\right) \triangleleft \left(\sum_{k=0}^{n} p_k\right) \to \mathfrak{m}_{\bigotimes_M p} \triangleleft \left(\sum_{k=0}^{n} \mathfrak{m}_{\bigotimes_M p}\right) \to \mathfrak{m}_{\bigotimes_M p} \triangleleft \mathfrak{m}_{\bigotimes_M p} \xrightarrow{\mu} \mathfrak{m}_{\bigotimes_M p}.$$

A voter selects a candidate from every subset of candidates. Such a voter is represented by a polynomial map $y \to [p, y]$. Running M elections on M voters is represented by the composite $\left(\bigotimes_M p\right) \otimes \left(\bigotimes_M [p, y]\right) \cong \bigotimes_M (p \otimes [p, y]) \to \bigotimes_M y = y$. Therefore, given M voters, we get a composite

$$p \to \mathfrak{m}_{\bigotimes_M p} \otimes \left(\bigotimes_M [p, y]\right) \to \mathfrak{m}_{\bigotimes_M p} \otimes \mathfrak{c}_{\bigotimes_M [p,y]} \to \mathfrak{m}_y = \mathbb{N}y.$$

On positions it maps a set of candidates $A \subseteq X$ to the number of run-offs required to elect a candidate. On directions, it maps the single direction of y to the winner.

It is tempting to expect that the maps $p \to \mathfrak{m}_{\bigotimes_M p}$ define an operad enriched in the Kleisli category $\mathbf{Poly}_{\mathfrak{m}}$ in the sense of [SS22]. However exhaustive run-off is gerrymander-able meaning that the division of voters into districts can affect the end-result of the election. This observation suggests the following definition of gerrymandering.

Definition 4.1. A voting scheme $p \to \mathfrak{m}_{\bigotimes_M p}$ can be **gerrymandered** if and only if it does not extend to an operad enriched in $\mathbf{Poly}_{\mathfrak{m}}$. ◊

4.4 Games run on players

For a game such as tic-tac-toe, let p be the polynomial whose positions are game states and whose directions are next possible moves. Then we can represent the game play as a position in $\mathfrak{m}_p$. In the game tic-tac-toe, a game state is a placement of $\times$'s and $\bigcirc$'s on a 3×3 grid. In other words, it is a map $b : 9 \to \{\times, \bigcirc, -\}$ where $-$ represents an open grid position. For $m = 1, \ldots, 9$, let B_m be the set of valid board states with m open positions. Assuming that $\times$ always plays first, these are

$$B_m := \{b : 9 \to \{\times, \bigcirc, -\} \mid \#(b^{-1}(-)) = m, \#(b^{-1}(\bigcirc)) \le \#(b^{-1}(\times)) \le \#(b^{-1}(\bigcirc)) + 1\}.$$

If there are an odd number of open positions, then it is $\times$'s turn. Otherwise, it is $\bigcirc$'s turn.

Given a board state $b : B_m$, the next possible moves are the open positions $b^{-1}(-)$. Therefore, the polynomial $p_\times = \sum_{b : B_1 + B_3 + \cdots + B_9} y^{b^{-1}(-)}$ represents the board states and next possible moves for $\times$.

Likewise the polynomial $p_{\bigcirc} = \sum_{b:B_2+B_4+\cdots+B_8} y^{b^{-1}(-)}$ represents the board states and next possible moves for $\bigcirc$.

Consider the polynomial $p = p_{\times} \triangleleft (y + p_{\bigcirc})$. This polynomial represents an $\times$ move followed by either game over or an $\bigcirc$ move. There exists a polynomial map $T\colon y^{\{\times,\bigcirc,-\}} \to \mathfrak{m}_p$ which selects the decision-tree in $\mathfrak{m}_p$ corresponding to the rules of tic-tac-toe, and maps directions to the winner of a completed game or $-$ if the game is tied. We will define the map T inductively. For $m = 1, \cdots, 9$, let $r_m = B_m y^{\{\times,\bigcirc,-\}}$.

- Define $r_1 \to y$, as follows. Given a board $b : B_1$ with a single open position, send the single direction in y to either the winner or to $-$ if the game is tied.
- Define $r_{m+1} \to p_{\bigcirc} \triangleleft (y + r_m)$ for odd m, and define $r_{m+1} \to p_{\times} \triangleleft (y + r_m)$ for even m, as follows. On positions send a board state $b : B_{m+1}$ to itself in $p_{\bigcirc}(1)$ (resp. $p_{\times}(1)$). Then for each valid move $m : b^{-1}(-)$ let $b' : B_m$ be the updated board state. If b' contains a winner, then send m to y and send the single direction of y to the winner. Otherwise, send m to $b' : r_m(1)$ and let the map on directions be the identity.

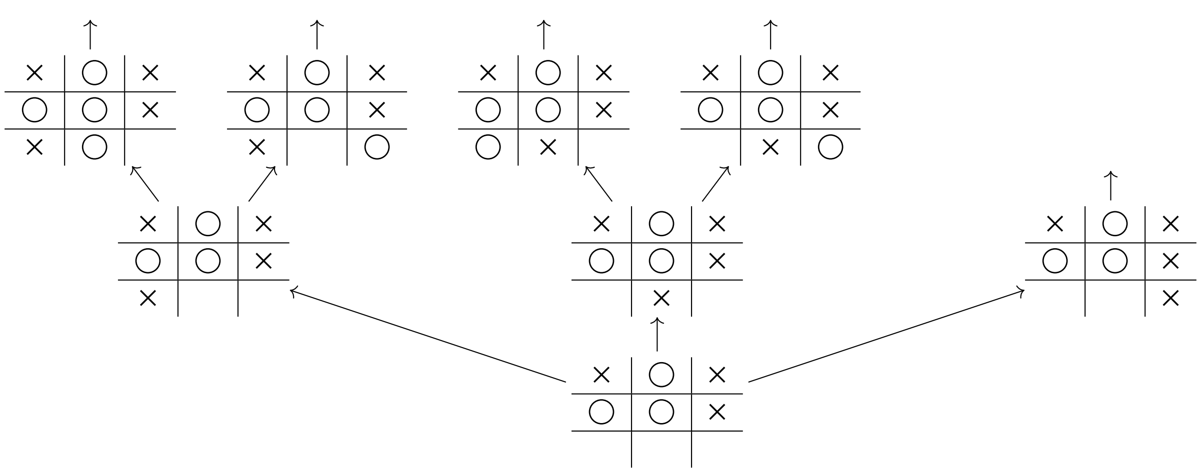
Using these maps, we can inductively define maps $r_m \to \mathfrak{m}_p$ for odd m as follows. As a base case we have the composite $r_1 \to y \to \mathfrak{m}_p$. Given a map $r_m \to \mathfrak{m}_p$, we have the composite

$$\begin{aligned} r_{m+2} \to p_{\times} \triangleleft (y + p_{\bigcirc} \triangleleft (y + r_m)) \to p_{\times} \triangleleft (y + p_{\bigcirc} \triangleleft (y + \mathfrak{m}_p)) &\to p_{\times} \triangleleft (y + p_{\bigcirc} \triangleleft \mathfrak{m}_p) \\ &\to p_{\times} \triangleleft (y + p_{\bigcirc}) \triangleleft \mathfrak{m}_p = p \triangleleft \mathfrak{m}_p \to \mathfrak{m}_p. \end{aligned}$$

The second to last map is induced by the composite

$$y + p_{\bigcirc} \triangleleft \mathfrak{m}_p = y \triangleleft y + p_{\bigcirc} \triangleleft \mathfrak{m}_p \to y \triangleleft \mathfrak{m}_p + p_{\bigcirc} \triangleleft \mathfrak{m}_p \to (y + p_{\bigcirc}) \triangleleft \mathfrak{m}_p.$$

Below is the image of a board state in $r_3(1)$ under the map $r_3 \to \mathfrak{m}_p$. From left to right the directions are sent to $\bigcirc$, $-$, $-$, $-$, and $\times$, as these are the win/loss/tie results of the games as shown.



Define T to be the map $r_9 = y^{\{\times,\bigcirc,-\}} \to \mathfrak{m}_p$.

Let (t, η, μ) be a $\triangleleft$-monad. A polynomial map $\varphi_{\times}\colon y \to [p_{\times}, t]$ represents an $\times$ player and a polynomial map $\varphi_{\bigcirc}\colon y \to [p_{\bigcirc}, t]$ represents an $\bigcirc$ player. If t is the trivial monad y, then for each board state, the player's next move is deterministic. If t is the lottery monad $\mathtt{lott} = \sum_{M:\mathbb{N}} \sum_{P:\Delta M} y^M$, then the players' next moves are stochastic.

By duoidality we have

$$p \otimes [p_{\times}, t] \otimes [p_{\bigcirc}, t] \to p \otimes [p_{\times}, t] \otimes [y + p_{\bigcirc}, t] \to (p_x \otimes [p_x, t]) \triangleleft ((y + p_{\bigcirc}) \otimes [y + p_{\bigcirc}, t]) \to t \triangleleft t \to t$$

where the first map is induced by the monoidal unit of t. Given an $\times$ player and an $\bigcirc$ player, we have a map $y^{\{\times,\bigcirc,-\}} \cong y^{\{\times,\bigcirc,-\}} \otimes y \otimes y \xrightarrow{T\otimes!\otimes!} \mathfrak{m}_p \otimes \mathfrak{c}_y \otimes \mathfrak{c}_y$. Then, the game play is represented by the composite

$$y^{\{\times,\bigcirc,-\}} \to \mathfrak{m}_p \otimes \mathfrak{c}_y \otimes \mathfrak{c}_y \xrightarrow{\mathfrak{m}_p\otimes\mathfrak{c}_{\varphi_\times}\otimes\mathfrak{c}_{\varphi_\bigcirc}} \mathfrak{m}_p \otimes \mathfrak{c}_{[p_\times,t]} \otimes \mathfrak{c}_{[p_\bigcirc,t]} \to \mathfrak{m}_{p\otimes[p_\times,t]\otimes[p_\bigcirc,t]} \to \mathfrak{m}_t \to t \tag{6}$$

which maps directions in t to winners of completed games.

We can promote this setup to players which *learn*, in other words players whose strategy dynamically changes after each completed game. A dynamic $\times$ player consists of a set of states $S_\times$ and a polynomial map $S_\times y^{S_\times} \to y^{\{\times,\bigcirc,-\}} \otimes \mathfrak{c}_{[p_\times,t]}$. Likewise for a dynamic $\bigcirc$ player. Such a player consists of the following:

- For each state, a behavior tree describing the player's strategy.
- For a winner or tied game and each finite path of the behavior tree, a new state.

Creating maps like φ is the subject of reinforcement learning [SB18]. As a typical such algorithm, take $S_\times$ to be the set of functions that assign a score in $\mathbb{N}$ to each move (direction in $p_\times$). Let $S_\times y^{S_\times} \to y^{\{\times,\bigcirc,-\}} \otimes \mathfrak{c}_{[p_\times,\texttt{lott}]}$ be defined as follows:

- On positions, it takes a score for each move and assigns a (stochastic) strategy which selects moves based on their relative scores.
- On directions, note that a direction of $\mathfrak{c}_{[p_\times,\texttt{lott}]}$ contains a finite number of moves played by $\times$ in the game. If the winner is $\times$, then we add 1 to the score for each played move. Otherwise, we keep the original scores.

Using the copy-on-directions map $y^{\{\times,\bigcirc,-\}} \otimes y^{\{\times,\bigcirc,-\}} \to y^{\{\times,\bigcirc,-\}}$, we have the composite

$$\begin{aligned} S_\times y^{S_\times} \otimes S_\bigcirc y^{S_\bigcirc} &\to y^{\{\times,\bigcirc,-\}} \otimes y^{\{\times,\bigcirc,-\}} \otimes \mathfrak{c}_{[p_\times,t]} \otimes \mathfrak{c}_{[p_\bigcirc,t]} \\ &\to y^{\{\times,\bigcirc,-\}} \otimes \mathfrak{c}_{[p_\times,t]} \otimes \mathfrak{c}_{[p_\bigcirc,t]} \xrightarrow{T} \mathfrak{m}_p \otimes \mathfrak{c}_{[p_\times,t]} \otimes \mathfrak{c}_{[p_\bigcirc,t]} \to t. \end{aligned}$$

This map takes a state for each player, runs the game using each player's strategy, and returns an updated state for each player based on the winner and the player's strategy.

5 Conclusion

In this paper, we constructed the free monad $\mathfrak{m}_p$ and cofree comonad $\mathfrak{c}_q$ on arbitrary polynomial functors $p, q : \mathbf{Poly}$, and defined a module structure $\mathfrak{m}_p \otimes \mathfrak{c}_q \to \mathfrak{c}_{p\otimes q}$. We also gave a series of examples to explain how this models the intuition "pattern runs on matter."

From here, it is not hard to show that $\mathfrak{m}$ and $\mathfrak{c}$ respectively extend to a monad and a comonad on $\mathbb{O}\mathbf{rg}$, the double category which serves as the base of enrichment for dynamic categorical structures for deep learning, prediction markets, etc., as defined in [SS22]. We (or others) may show in future work that for any polynomial monad t, there is a functor $[-,t]\colon \mathbb{O}\mathbf{rg}_{\mathfrak{m}}^{\mathrm{op}} \to \mathbb{O}\mathbf{rg}^{\mathfrak{c}}$ from opposite of the $\mathfrak{m}$-Kleisli category to the $\mathfrak{c}$-coKleisli category. The latter offers the ability for different machines to operate at different rates in wiring diagrams, and the former offers the ability to call multiple subprocesses before returning, though we have not found compelling examples of either; this again is future work.

The constructions in this paper should generalize straightforwardly to free monads and cofree comonads for familial functors between copresheaf categories, as in [Web07; LSS23]. One should check that there is again a module structure of the same form in that setting.

References

[AGM01] Samson Abramsky, Dov M Gabbay, and Thomas SE Maibaum. *Handbook of Logic in Computer Science: Volume 5. Algebraic and Logical Structures*. OUP Oxford, 2001 (cit. on p. 2).

[AR94] Jiri Adamek and Jiri Rosicky. *Locally presentable and accessible categories*. London Mathematical Society Lecture Note Series 189. Cambridge University Press, 1994 (cit. on p. 16).

[GK12] Nicola Gambino and Joachim Kock. "Polynomial functors and polynomial monads". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 154.1 (Sept. 2012), pp. 153–192 (cit. on p. 2).

[Kel80] G Max Kelly. "A unified treatment of transfinite constructions for free algebras, free monoids, colimits, associated sheaves, and so on". In: *Bulletin of the Australian Mathematical Society* 22.1 (1980), pp. 1–83 (cit. on p. 2).

[KRU20] Shin-ya Katsumata, Exequiel Rivas, and Tarmo Uustalu. "Interaction Laws of Monads and Comonads". In: *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS '20. Saarbrücken, Germany: Association for Computing Machinery, 2020, pp. 604–618 (cit. on pp. 2, 7, 8).

[LSS23] Owen Lynch, Brandon T. Shapiro, and David I. Spivak. *All Concepts are* $\mathbb{C}\mathbf{at}^{\#}$. 2023. arXiv: `2305.02571 [math.CT]` (cit. on p. 12).

[nLa24a] nLab authors. *Exact square*. `https://ncatlab.org/nlab/show/exact+square`. Revision 17. Mar. 2024 (cit. on p. 15).

[nLa24b] nLab authors. *Module over a monoidal functor*. `https://ncatlab.org/nlab/show/module+over+a+monoidal+functor`. Revision 4. Mar. 2024 (cit. on p. 7).

[nLa24c] nLab authors. *Transfinite construction of free algebras*. `https://ncatlab.org/nlab/show/transfinite+construction+of+free+algebras`. Revision 13. Mar. 2024 (cit. on p. 2).

[NS22] Nelson Niu and David I. Spivak. *Polynomial functors: a general theory of interaction*. *In preparation*. 2022 (cit. on p. 3).

[SB18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018 (cit. on p. 12).

[Spi] David I. Spivak. *Where matter & pattern meet*. URL: `https://topos.site/blog/2022-11-08-matter-and-pattern/` (cit. on p. 1).

[SS22] Brandon T. Shapiro and David I. Spivak. "Dynamic categories, dynamic operads: From deep learning to prediction markets". In: *Electronic Proceedings in Theoretical Computer Science* (2022) (cit. on pp. 10, 12).

[Web07] Mark Weber. "Familial 2-Functors and Parametric Right Adjoints". In: *Theory and Applications of Categories* 18 (2007), Paper No. 22, 665–732 (cit. on p. 12).

A Distributing colimits over $\triangleleft$

In general, colimits do not distribute over the subsitution product $\triangleleft$. However, in this section we give hypotheses under which the natural maps

$$\operatorname*{colim}_{j\in\mathcal{J}}(p\triangleleft q_j)\to p\triangleleft(\operatorname*{colim}_{j\in\mathcal{J}} q_j) \qquad\text{and}\qquad \operatorname*{colim}_{j\in\mathcal{J}}(p_j\triangleleft q)\to(\operatorname*{colim}_{j\in\mathcal{J}} p_j)\triangleleft q$$

are in fact isomorphisms.

A.1 Right distribution

We begin with hypotheses under which colimits right distribute over $\triangleleft$.

Definition A.1. For an ordinal κ, a polynomial p is called **κ-small** if for all κ-filtered categories $\mathcal{J}$ and all diagrams $Q\colon \mathcal{J} \to \mathbf{Set}$

$$p \triangleleft \left(\operatorname*{colim}_{j\colon \mathcal{J}} Q_j \right) = \operatorname*{colim}_{j\colon \mathcal{J}} \left(p \triangleleft Q_j \right). \qquad \Diamond$$

Remark A.2. A polynomial $p : \mathbf{Poly}$ is κ-small if and only if all of its direction-sets have cardinality less than κ. It is called **finitary** if and only if it is ω-small. $\Diamond$

Proposition A.3. *If a polynomial p is κ-small, then for all κ-filtered categories $\mathcal{J}$ and diagrams $q\colon \mathcal{J} \to \mathbf{Poly}$*

$$p \triangleleft \left(\operatorname*{colim}_{j\colon \mathcal{J}} q_j \right) = \operatorname*{colim}_{j\colon \mathcal{J}} \left(p \triangleleft q_j \right).$$

Proof. That this map is an isomorphism on positions follows directly from the definition of κ-small. That this map is an isomorphism on directions follows from the definition of colimits in **Poly** and the fact that connected limits preserve coproducts. $\square$

A.2 Left distribution

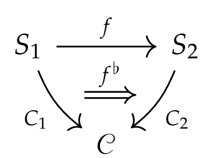
Next we consider hypotheses under which colimits left distribute over $\triangleleft$. A map $\phi\colon p \to q$ in **Poly** is **cartesian** if for every position $P : p(1)$, the map on directions $\phi^\#_P$ is an isomorphism (equivalently, when all ϕ's naturality squares are pullbacks). Let $\mathbf{Poly_{Cart}}$ be subcategory of **Poly** consisting of cartesian maps. If ϕ is cartesian, then so is $p \triangleleft \phi$ and $p + \phi$ for any $p : \mathbf{Poly}$.

First we give a series of Lemmas which allow us to prove Proposition A.8.

Lemma A.4. *A polynomial map $p \to q$ is an isomorphism if and only if for all sets X, the induced function $p \triangleleft X \to q \triangleleft X$ is a bijection.*

Proof. This is just a restatement of the fact that $\mathrm{Ext}\colon \mathbf{Poly} \to \mathbf{Set}^{\mathbf{Set}}$ is fully faithful; indeed, $\mathrm{Ext}(p)(X) \cong p \triangleleft X$. $\square$

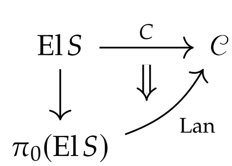
The coproduct completion $\Sigma\mathcal{C}$ of a category $\mathcal{C}$ has as objects pairs (S, C) where $S : \mathbf{Set}$ and $C\colon S \to \mathcal{C}$ is a discrete diagram, and it has as morphisms lax triangles



A diagram $\mathcal{J} \to \Sigma\mathcal{C}$ consists of a pair (S, C), where $S\colon \mathcal{J} \to \mathbf{Set}$ is a diagram and $C\colon \mathrm{El}\, S \to \mathcal{C}$ is a functor from the category of elements of S to $\mathcal{C}$.

Lemma A.5. *Let $\mathcal{C}$ be a category and let $\Sigma\mathcal{C}$ be its coproduct completion. If $\mathcal{C}$ has $\mathcal{J}$-shaped colimits, then so does $\Sigma\mathcal{C}$, and they are computed as follows.*

Given a diagram $(S, C)\colon \mathcal{J} \to \Sigma\mathcal{C}$, consider the Kan extension



where $\pi_0\colon \mathbf{Cat} \to \mathbf{Set}$ *is the connected components reflection. Then the colimit is given by*

$$\operatorname{colim}(S, C) \cong (\pi_0(\operatorname{El} S), \operatorname{Lan}). \tag{7}$$

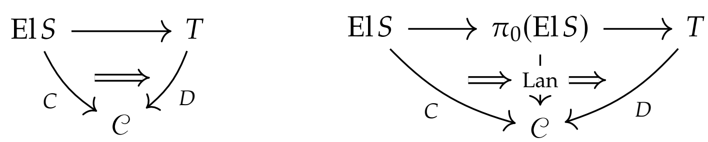
For each element $s_0 : \pi_0(\operatorname{El} S)$*, its image object* $\operatorname{Lan}(s_0) : \operatorname{Ob} \mathcal{C}$ *is given by the colimit over all* $(i, s_i) \mapsto s$ *of* $C(s_i)$*.*

Proof. First consider the case $\mathcal{C} = 1$, so that $\Sigma\mathcal{C} \cong \mathbf{Set}$. We need to check that for any $S\colon \mathcal{I} \to \mathbf{Set}$, there is a bijection

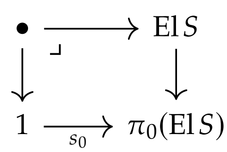
$$\operatorname{colim} S \cong \pi_0(\operatorname{El} S). \tag{8}$$

This follows from the fact that the inclusion of diagrams of sets into diagrams of categories is a left adjoint, the Grothendieck construction taking diagrams of categories to categories is a left adjoint, and π_0 taking categories to sets is a left adjoint.

Given $T : \mathbf{Set}$, a discrete diagram $D\colon T \to \mathcal{C}$, and diagrams $(S_i, C_i) \to (T, D)$, coherently over $i : \mathcal{I}$, one obtains a diagram as left, which factors uniquely as right:

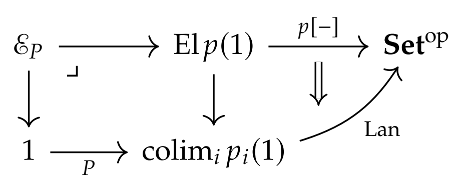


This gives (7). For the last statement, note that since $1 \to \pi_0(\operatorname{El} S)$ is a map of sets, the pullback of categories



is also a comma square. Hence it is exact in the sense of [nLa24a], completing the proof. □

Lemma A.5 provides a formula for computing colimits in **Poly**, since $\mathbf{Poly} \cong \Sigma\mathbf{Set}^{\mathrm{op}}$ is the coproduct completion of $\mathbf{Set}^{\mathrm{op}}$. Namely, the position-set of the colimit of $p\colon \mathcal{I} \to \mathbf{Poly}$ is given by the colimit $\operatorname{colim}_i p_i(1)$ of the position-sets, and the directions at a position $P : \operatorname{colim}_i p_i(1)$ are given by the limit (taken in **Set**, i.e. the colimit taken in $\mathbf{Set}^{\mathrm{op}}$) of the associated connected diagram $\mathcal{E}_P$ of direction sets.



Proposition A.6. *The inclusion* $\mathbf{Poly_{Cart}} \to \mathbf{Poly}$ *reflects all colimits.*

Proof. Suppose given a category I, a functor $p\colon I \to \mathbf{Poly_{Cart}}$, a polynomial $\overline{p} : \mathbf{Poly}$, and cartesian maps $\varphi_i\colon p_i \to \overline{p}$ forming a cone. Suppose further that in **Poly** it is a colimit cone. To see that it is a colimit cone in $\mathbf{Poly_{Cart}}$ we need only check that for any q and cone $\psi_i\colon p_i \to q$ in $\mathbf{Poly_{Cart}}$, the induced map $\psi\colon \overline{p} \to q$ in **Poly** is in fact in $\mathbf{Poly_{Cart}}$, i.e. for any position $P : \overline{p}(1)$, the induced function $\psi_P^\sharp\colon q[\varphi(P)] \to \overline{p}[P]$ is bijective.

By Lemma A.5, the position-set of a colimit is the colimit of the position-sets for any diagram of polynomials, meaning there exists some $i : I$ and $P_i : p_i(1)$ representing P, i.e. with $\varphi_i(P_i) = P$. Since $p_i \to \overline{p}$ is cartesian, we have a commuting triangle of functions $q[\varphi(P)] \to \overline{p}[P] \to p_i[P_i]$ of which two are bijections; it follows that the required one is as well. □

Proposition A.7. *The inclusion* $\mathbf{Poly}_{\mathbf{Cart}} \to \mathbf{Poly}$ *creates coproducts and filtered colimits. Moreover, the composite*

$$\mathbf{Poly}_{\mathbf{Cart}} \to \mathbf{Poly} \xrightarrow{\mathrm{Ext}} \mathbf{Set}^{\mathbf{Set}}$$

creates coproducts and filtered colimits.

Proof. The functor $\mathrm{Ext}\colon \mathbf{Poly} \to \mathbf{Set}^{\mathbf{Set}}$ is fully faithful and hence reflects all colimits; by Proposition A.6, $\mathbf{Poly}_{\mathbf{Cart}} \to \mathbf{Poly}$ does too. It suffices to show that $\mathbf{Poly}_{\mathbf{Cart}}$ has coproducts and filtered colimits, because and that the two functors $\mathbf{Poly}_{\mathbf{Cart}} \to \mathbf{Poly}$ and $\mathbf{Poly}_{\mathbf{Cart}} \to \mathbf{Set}^{\mathbf{Set}}$ preserve them.

Coproduct inclusions of polynomials are cartesian, so the coproduct cone exists in $\mathbf{Poly}_{\mathbf{Cart}}$, and hence it is a coproduct. Since Ext preserves coproducts, this concludes the case for coproducts.

By [AR94], a category has (resp. a functor preserves) all filtered colimits iff it has (resp. preserves) all directed colimits, so let $\mathcal{I}$ be a directed category and $p\colon \mathcal{I} \to \mathbf{Poly}_{\mathbf{Cart}}$ a directed sequence of polynomials and cartesian maps. Then the induced map $\mathcal{I} \to \mathbf{Poly}$ has a colimit, say $\overline{p}$, and it is easy to check that all the structure maps $p_i \to \overline{p}$ are cartesian. Hence $\overline{p} \cong \mathrm{colim}_i\, p_i$ is a colimit in $\mathbf{Poly}_{\mathbf{Cart}}$, and it is preserved by the inclusion.

It remains to check that $\overline{p}$ is a colimit in $\mathbf{Set}^{\mathbf{Set}}$. That is, we need to check that for any $X : \mathbf{Set}$, the function

$$\sum_{P:\pi_0(\mathrm{El}\,p)} \operatorname*{colim}_{e:\mathcal{E}_P} X^{p[e]} \longrightarrow \sum_{P:\pi_0(\mathrm{El}\,p)} X^{\lim_{e:\mathcal{E}_P} p[e]}$$

is a bijection. Choose any P; it suffices to show that $\mathrm{colim}_{e:\mathcal{E}_P} X^{p[e]} \to X^{\lim_{e:\mathcal{E}_P} p[e]}$ is a bijection. But $\mathcal{E}_P$ is filtered and for any $e \to e'$ in $\mathcal{E}_P$, the map $p[e'] \to p[e]$ is a bijection. Hence we can pick any object $e' : \mathcal{E}_P$, replace all p_e with $p_{e'}$, and have an isomorphic diagram. Now the colimit and limit are both constant, meaning that both sides of the desired map are isomorphic to $X^{p[e']}$, and hence to each other. □

Proposition A.8. *For any polynomial* q*, filtered category* $\mathcal{I}$*, and diagram* $p\colon \mathcal{I} \to \mathbf{Poly}_{\mathbf{Cart}}$*, the natural map*

$$\operatorname*{colim}_{i:I}(p_i \triangleleft q) \to (\operatorname*{colim}_{i:I} p_i) \triangleleft q$$

is an isomorphism in **Poly**.

Proof. By Lemma A.4 it suffices to show that the function $\mathrm{colim}_{i:I}(p_i \triangleleft q) \triangleleft X \to (\mathrm{colim}_{i:I}\, p_i) \triangleleft q \triangleleft X$ is a bijection for any $X : \mathbf{Set}$. This is just two uses of Proposition A.7:

$$\operatorname*{colim}_{i:I}(p_i \triangleleft q) \triangleleft X \cong \operatorname*{colim}_{i:I}(p_i \triangleleft q \triangleleft X) \cong (\operatorname*{colim}_{i:I} p_i) \triangleleft q \triangleleft X. \qquad \square$$

B Proofs for the free monad monad

Lemma 2.7. *For ordinals* α, β, γ*, the following diagram commutes:*

$$\begin{array}{ccc}
p_{(\alpha)} \triangleleft p_{(\beta)} \triangleleft p_{(\gamma)} & \xrightarrow{p_{(\alpha)} \triangleleft \mu_{(\beta,\gamma)}} & p_{(\alpha)} \triangleleft p_{(\beta+\gamma)} \\
{\scriptstyle \mu_{(\alpha,\beta)} \triangleleft p_{(\gamma)}} \downarrow & & \downarrow {\scriptstyle \mu_{(\alpha,\beta+\gamma)}} \\
p_{(\alpha+\beta)} \triangleleft p_{(\gamma)} & \xrightarrow[\mu_{(\alpha+\beta,\gamma)}]{} & p_{(\alpha+\beta+\gamma)}
\end{array}$$

Proof. We prove this lemma by transfinite induction on α. First, for $\alpha = 0$ the result is immediate.

Next, suppose that the result holds for α, and we will show that it holds for the successor ordinal $\alpha + 1$. By definition of $p_{(\alpha+1)}$, it suffices to show that the outer square in the following diagram commutes.

$$\begin{tikzcd}
p_{(\beta)} \triangleleft p_{(\gamma)} + p \triangleleft p_{(\alpha)} \triangleleft p_{(\beta)} \triangleleft p_{(\gamma)} \arrow[r, "\mu_{(\beta,\gamma)} + p \triangleleft p_{(\alpha)} \triangleleft \mu_{(\beta,\gamma)}"] \arrow[d, "p_{(\beta)} \triangleleft p_{(\gamma)} + p \triangleleft \mu_{(\alpha,\beta)} \triangleleft p_{(\gamma)}"'] & p_{(\beta+\gamma)} + p \triangleleft p_{(\alpha)} \triangleleft p_{(\beta+\gamma)} \arrow[d, "p_{(\beta+\gamma)} + p \triangleleft \mu_{(\alpha,\beta+\gamma)}"] \\
p_{(\beta)} \triangleleft p_{(\gamma)} + p \triangleleft p_{(\alpha+\beta)} \triangleleft p_{(\gamma)} \arrow[r, "\mu_{(\beta,\gamma)} + p_{(\alpha)} \mu_{(\alpha+\beta,\gamma)}"] \arrow[d, "(\iota_{(\beta)} \triangleleft p_{(\gamma)}, \lambda_{(\alpha+\beta)} \triangleleft p_{(\gamma)})"'] & p_{(\beta+\gamma)} + p_{(\alpha+\beta+\gamma)} \arrow[d, "(\iota_{(\beta+\gamma)}, \lambda_{(\alpha+\beta+\gamma)})"] \\
p_{(\alpha+\beta+1)} \triangleleft p_{(\gamma)} \arrow[r, "\mu_{(\alpha+\beta+1,\gamma)}"'] & p_{(\alpha+\beta+\gamma+1)}
\end{tikzcd}$$

The top square commutes by the induction hypothesis. To check that the bottom squares commute, it suffices to show that the following diagrams commute as well.

$$\begin{tikzcd}
p_{(\beta)} \triangleleft p_{(\gamma)} \arrow[r, "\mu_{(\beta,\gamma)}"] \arrow[d, "\iota_{(\beta)} \triangleleft p_{(\gamma)}"'] & p_{(\beta+\gamma)} \arrow[d, "\iota_{(\beta+\gamma)}"] \\
p_{(\alpha+\beta+1)} \triangleleft p_{(\gamma)} \arrow[r, "\mu_{(\alpha+\beta+1,\gamma)}"'] & p_{(\alpha+\beta+\gamma+1)}
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
p \triangleleft p_{(\alpha+\beta)} \triangleleft p_{(\gamma)} \arrow[r, "p \triangleleft \mu_{(\alpha+\beta,\gamma)}"] \arrow[d, "\lambda_{(\alpha+\beta)} \triangleleft p_{(\gamma)}"'] & p \triangleleft p_{(\alpha+\beta+\gamma)} \arrow[d, "\lambda_{(\alpha+\beta+\gamma)}"] \\
p_{(\alpha+\beta+1)} \triangleleft p_{(\gamma)} \arrow[r, "\mu_{(\alpha+\beta+1,\gamma)}"'] & p_{(\alpha+\beta+\gamma+1)}
\end{tikzcd}$$

The diagram on the left commutes by Equation (3). The diagram on the right commutes by the inductive definition of $\mu_{(\alpha+\beta+1,\gamma)}$.

Finally, suppose that α is a limit ordinal and suppose that the diagram commutes for all $\alpha' < \alpha$. Recall that $p_{(\alpha)} = \operatorname{colim}_{\alpha'<\alpha} p_{(\alpha')}$. So it suffices to show that the outer diagram in the following diagram commutes.

$$\begin{tikzcd}
p_{(\alpha')} \triangleleft p_{(\beta)} \triangleleft p_{(\gamma)} \arrow[r, "p_{(\alpha')} \triangleleft \mu_{(\beta,\gamma)}"] \arrow[d, "\mu_{(\alpha',\beta)} \triangleleft p_{(\gamma)}"'] & p_{(\alpha')} \triangleleft p_{(\beta+\gamma)} \arrow[r] \arrow[d, "\mu_{(\alpha',\beta+\gamma)}"] & \left(\operatorname{colim}_{\alpha'<\alpha} p_{(\alpha')}\right) \triangleleft p_{(\beta+\gamma)} \arrow[dd, "\mu_{(\alpha,\beta+\gamma)}"] \\
p_{(\alpha'+\beta)} \triangleleft p_{(\gamma)} \arrow[r, "\mu_{(\alpha'+\beta,\gamma)}"'] \arrow[d] & p_{(\alpha'+\beta+\gamma)} \arrow[dr] & \\
\left(\operatorname{colim}_{\alpha'<\alpha} p_{(\alpha'+\beta)}\right) \triangleleft p_{(\gamma)} \arrow[rr, "\mu_{(\alpha+\beta,\gamma)}"'] & & \operatorname{colim}_{\alpha'<\alpha} p_{(\alpha'+\beta+\gamma)}
\end{tikzcd}$$

The upper left square commutes by the induction hypothesis and the remaining diagrams commute by the definition of μ for limit ordinals. □

Proposition 2.9. *For $f\colon p \to q$ in* **Poly***, the polynomial map $\mathfrak{m}_f : \mathfrak{m}_p \to \mathfrak{m}_q$ is a map of $\triangleleft$-monoids.*

Proof. To show that $\mathfrak{m}_f$ respects the identity it suffices to show that the outer diagram in the following commutes:

$$\begin{array}{ccccc} y & \xrightarrow{y} & p_{(0)} & \xrightarrow{\iota_{(0)}} & \mathfrak{m}_p \\ & \searrow^{y} & \downarrow{\scriptstyle f_{(0)}} & & \downarrow{\scriptstyle \mathfrak{m}_f} \\ & & q_{(0)} & \xrightarrow[\iota_{(0)}]{} & \mathfrak{m}_q \end{array}$$

The left triangle commutes because each map is the identity on y. The square commutes as it is the diagram in Equation (4).

To show that $\mathfrak{m}_f$ respects multiplication, it suffices to show that the following diagram commutes for all α and β.

$$\begin{array}{ccc} p_{(\alpha)} \triangleleft p_{(\beta)} & \xrightarrow{\mu_{(\alpha,\beta)}} & p_{(\alpha+\beta)} \\ {\scriptstyle f_{(\alpha)} \triangleleft f_{(\beta)}}\downarrow & & \downarrow{\scriptstyle f_{(\alpha+\beta)}} \\ q_{(\alpha)} \triangleleft q_{(\beta)} & \xrightarrow[\mu_{(\alpha,\beta)}]{} & q_{(\alpha+\beta)} \end{array}$$

We show this by induction on α. For $\alpha = 0$ this diagram commutes because the horizontal maps are the identity and the vertical maps are $f_{(\beta)}$. Suppose that the diagram commutes for α. Then for the successor ordinal $\alpha+1$, following the definitions of $\mu_{(\alpha+1,\beta)}$ and $f_{(\alpha+1,\beta)}$ we want to show that the following diagram commutes.

$$\begin{array}{ccc} p_{(\beta)} + p \triangleleft p_{(\alpha)} \triangleleft p_{(\beta)} & \xrightarrow{(\iota_{(\beta)}, (p \triangleleft \mu_{(\alpha,\beta)}) ⨾ \lambda_{(\alpha+\beta)})} & p_{(\alpha+\beta+1)} \\ {\scriptstyle f_{(\beta)} + f \triangleleft f_{(\alpha)} \triangleleft f_{(\beta)}}\downarrow & & \downarrow{\scriptstyle f_{(\alpha+\beta+1)}} \\ q_{(\beta)} + q \triangleleft q_{(\alpha)} \triangleleft q_{(\beta)} & \xrightarrow[(\iota_{(\beta)}, (p \triangleleft \mu_{(\alpha,\beta)}) ⨾ \lambda_{(\alpha+\beta)})]{} & q_{(\alpha+\beta+1)} \end{array}$$

The diagram commutes on the first term of the coproduct because the diagram in Equation (4) commutes with the pair $\beta < \alpha + \beta$. To show that the diagram commutes on the second term, it suffices to show that the outer diagram in the following commutes.

$$\begin{array}{ccccc} p \triangleleft p_{(\alpha)} \triangleleft p_{(\beta)} & \xrightarrow{p \triangleleft \mu_{(\alpha,\beta)}} & p \triangleleft p_{(\alpha+\beta)} & \xrightarrow{\lambda_{(\alpha+\beta)}} & p_{(\alpha+\beta+1)} \\ {\scriptstyle f \triangleleft f_{(\alpha)} \triangleleft f_{(\beta)}}\downarrow & & \downarrow{\scriptstyle f \triangleleft f_{(\alpha+\beta)}} & & \downarrow{\scriptstyle f_{(\alpha+\beta+1)}} \\ q \triangleleft q_{(\alpha)} \triangleleft q_{(\beta)} & \xrightarrow[q \triangleleft \mu_{(\alpha,\beta)}]{} & q \triangleleft q_{(\alpha+\beta)} & \xrightarrow[\lambda_{(\alpha+\beta)}]{} & q_{(\alpha+\beta+1)} \end{array}$$

The left-hand square commutes by the induction hypothesis and the right-hand square commutes by definition of $f_{(\alpha+\beta+1)}$. □

B.1 Proof of Theorem 2.10

The trickiest part of proving the adjunction in Theorem 2.10 is defining the co-unit so we do it separately in the following series of Lemmas.

Lemma B.1. *Let (q, η_q, μ_q) be a $\triangleleft$-monoid. For each ordinal α, there exist a cocone of maps*

$$\epsilon_{(\alpha)} \colon q_{(\alpha)} \to q.$$

Proof. We will define the cocone of maps $\epsilon_{(\alpha)}$ inductively. First, define $\epsilon_{(0)} \colon q_{(0)} = y \to q$ to be the unit η_q.

Next, suppose that we have a cocone of maps $\epsilon_{(\alpha')} : q_{(\alpha')} \to q$ for all $\alpha' < \alpha$. If α is a limit ordinal then we define $\epsilon_{(\alpha)} \colon q_{(\alpha)} \to q$ to be the universal map induced by the cocone. If $\alpha + 1$ is a successor ordinal, then we define $\epsilon_{(\alpha+1)}$ to be the composite

$$q_{(\alpha+1)} = y + q \triangleleft q_{(\alpha)} \xrightarrow{\eta_q + q \triangleleft \epsilon_{(\alpha)}} q + q \triangleleft q \xrightarrow{(1,\mu_q)} q.$$

To show that these maps form a cocone, it suffices to show that the following diagram commutes for all $\alpha' < \alpha$.

$$\begin{array}{ccc} q_{(\alpha')} & & \\ {\scriptstyle \iota_{(\alpha')}}\downarrow & \searrow^{\epsilon_{(\alpha')}} & \\ q_{(\alpha)} & \xrightarrow[\epsilon_{(\alpha)}]{} & q \end{array}$$

If α is a limit ordinal, then this is immediate. Otherwise, α is a successor ordinal, say $\alpha = \beta + 1$. It suffices to show that the following diagram commutes.

$$\begin{array}{ccc} q_{(\beta)} & & \\ {\scriptstyle \iota_{(\beta)}}\downarrow & \searrow^{\epsilon_{(\beta)}} & \\ q_{(\beta+1)} & \xrightarrow[\epsilon_{(\beta+1)}]{} & q \end{array}$$

Consider the following three cases.

- Suppose that $\beta = 0$, then it is immediate that the desired diagram commutes as we see below.

$$\begin{array}{ccccc} q_{(0)} = y & & \xrightarrow{\eta_q} & & q \\ {\scriptstyle \iota_{(0)}}\downarrow & & & & \\ q_{(1)} = y + q \triangleleft y & \xrightarrow[\eta_q + \mu_q \circ (q \triangleleft \eta_q)]{} & q + q & \xrightarrow[(1,1)]{} & q \end{array}$$

- Next suppose that β is a successor ordinal. In particular $\beta = \gamma + 1$. To show that

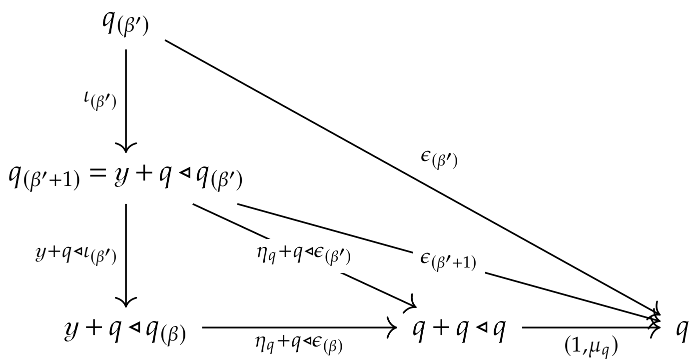
$$\begin{array}{ccc} q_{(\gamma+1)} & & \\ {\scriptstyle \iota_{(\gamma+1)}}\downarrow & \searrow^{\epsilon_{(\gamma+1)}} & \\ q_{(\gamma+2)} & \xrightarrow[\epsilon_{(\gamma+2)}]{} & q \end{array}$$

commutes, it suffices to show that

$$\begin{array}{ccc} y + q \triangleleft q_{(\gamma)} & & \\ {\scriptstyle y + q \triangleleft \iota_{(\gamma)}}\downarrow & \searrow^{\eta_q + q \triangleleft \epsilon_{(\gamma)}} & \\ y + q \triangleleft q_{(\gamma+1)} & \xrightarrow[\eta_q + q \triangleleft \epsilon_{(\gamma+1)}]{} & q + q \triangleleft q \end{array}$$

commutes. This is immediate from the induction hypothesis.

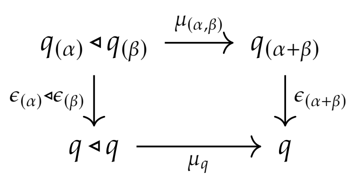
- Finally, suppose that β is a limit ordinal. Then we want to show that the outer diagram in the following commutes.



The top and bottom triangles commute by the induction hypothesis and the middle triangle commutes by definition of $\epsilon_{(\beta'+1)}$.

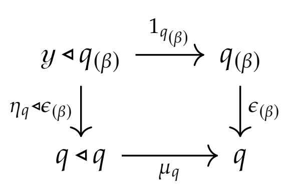
□

Lemma B.2. *For all ordinals α, β the following diagram commutes.*

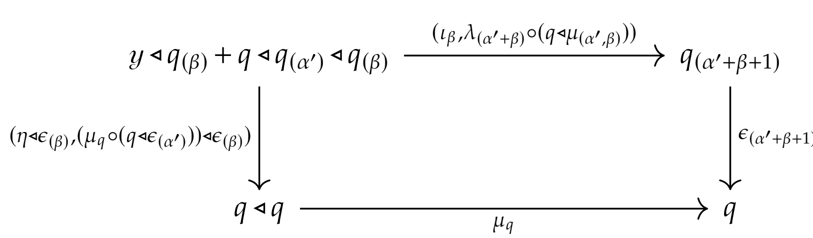


Proof. We induct on α. Suppose that for all $\alpha' < \alpha$ the diagram commutes. Consider the following cases.

- For $\alpha = 0$, we have that



  commutes by the unit law of the monoid (q, η_q, μ_q).
- Suppose that α is a successor ordinal, say $\alpha = \alpha' + 1$. We must show that the following diagram commutes.



  We will show that the diagram commutes on each component of the coproduct independently. For the first term observe that

$$\begin{array}{ccccc}
y \triangleleft q_{(\beta)} & \xrightarrow{1_{q_{(\beta)}}} & q_{(\beta)} & \xrightarrow{\iota_\beta} & q_{(\alpha'+\beta+1)} \\
{\scriptstyle \eta \triangleleft \epsilon_{(\beta)}}\downarrow & & \downarrow{\scriptstyle \epsilon_{(\beta)}} & \swarrow{\scriptstyle \epsilon_{(\alpha'+\beta+1)}} & \\
q \triangleleft q & \xrightarrow[\mu_q]{} & q & &
\end{array}$$

commutes because of the unit law of the monoid (q, η_q, μ_q) and because the maps $\epsilon_{(\beta)}$ form a cocone.

For the second term consider the following diagram.

$$\begin{array}{ccccc}
q \triangleleft q_{(\alpha')} \triangleleft q_{(\beta)} & \xrightarrow{q \triangleleft \mu_{(\alpha',\beta)}} & q \triangleleft q_{(\alpha'+\beta)} & \xrightarrow{\lambda_{(\alpha'+\beta)}} & q_{(\alpha'+\beta+1)} \\
{\scriptstyle q \triangleleft \epsilon_{(\alpha')} \triangleleft \epsilon_{(\beta)}}\downarrow & & \downarrow{\scriptstyle q \triangleleft \epsilon_{(\alpha'+\beta)}} & & \\
q \triangleleft q \triangleleft q & \xrightarrow[q \triangleleft \mu_q]{} & q \triangleleft q & & \downarrow{\scriptstyle \epsilon_{(\alpha'+\beta+1)}} \\
{\scriptstyle \mu_q \triangleleft q}\downarrow & & & \searrow{\scriptstyle \mu_q} & \\
q \triangleleft q & & \xrightarrow[\mu_q]{\qquad\qquad\qquad} & & q
\end{array}$$

The upper left square commutes by the induction hypothesis. The bottom square commutes by associativity of μ_q, and the right-most square commutes by definition of $\epsilon_{(\alpha'+\beta+1)}$. Therefore the outer square commutes as desired.

- Suppose that α is a limit ordinal. Recall that

$$q_{(\alpha)} \triangleleft q_{(\beta)} \cong \operatorname*{colim}_{\alpha' < \alpha} q_{(\alpha'+\beta)}.$$

The composite $\epsilon_{(\alpha+\beta)} \circ \mu_{(\alpha,\beta)}$ is induced by the cocone defined for $\alpha' < \alpha$ by

$$q_{(\alpha')} \triangleleft q_{(\beta)} \xrightarrow{\mu_{(\alpha',\beta)}} q_{(\alpha'+\beta)} \xrightarrow{\iota_{(\alpha'+\beta)}} q_{(\alpha+\beta)} \xrightarrow{\epsilon_{(\alpha+\beta)}} q$$

while the composite $\mu_q \circ (\epsilon_{(\alpha)} \triangleleft \epsilon_{(\beta)})$ is induced by the cocone defined for $\alpha' < \alpha$ by

$$q_{(\alpha')} \triangleleft q_{(\beta)} \xrightarrow{\epsilon_{(\alpha')} \triangleleft \epsilon_{(\beta)}} q \triangleleft q \xrightarrow{\mu_q} q.$$

For each $\alpha' < \alpha$ these composites are identical by the induction hypothesis. Therefore by uniqueness of the induced maps

$$\epsilon_{(\alpha+\beta)} \circ \mu_{(\alpha,\beta)} = \mu_q \circ (\epsilon_{(\alpha)} \triangleleft \epsilon_{(\beta)})$$

as desired.

□

Theorem 2.10. *There is an adjunction*

$$\mathbf{Poly} \underset{U}{\overset{\mathfrak{m}_-}{\underset{\longleftarrow}{\overset{\longrightarrow}{\Rightarrow}}}} \mathbf{Mod}(\mathbf{Poly}) \ .$$

Proof. For p : **Poly** define the unit of the adjunction, $\zeta_p \colon p \to \mathfrak{m}_p$, to be the composite

$$p \xrightarrow{\lambda_{(0)}} p_{(1)} \xrightarrow{\iota_{(1)}} \mathfrak{m}_p.$$

For $f \colon p \to q$ in **Poly**, consider the following diagram.

$$\begin{array}{ccccc} p & \xrightarrow{\lambda_{(0)}} & y + p = p_{(1)} & \xrightarrow{\iota_{(1)}} & \mathfrak{m}_p \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle y+f=f_{(1)}} & & \downarrow{\scriptstyle \mathfrak{m}_f} \\ q & \xrightarrow[\lambda_{(0)}]{} & y + q = q_{(1)} & \xrightarrow[\iota_{(1)}]{} & \mathfrak{m}_q \end{array}$$

The square on the left commutes by definition of $f_{(1)}$ while the square on the right commutes by the commuting diagram in Equation (4). Therefore, the outer square commutes and ζ is natural.

For the $\triangleleft$-monoid (q, η_q, μ_q), we define the counit of the adjunction $\epsilon_q \colon (\mathfrak{m}_q, \eta, \mu) \to (q, \eta_q, \mu_q)$ to be $\epsilon_{(\kappa)} \colon \mathfrak{m}_q \to q$ for κ such that q is κ-small. The map ϵ_q preserves the unit because the maps $\epsilon_{(\alpha)}$ form a cocone. That ϵ_q preserves multiplication is a direct result of Lemma B.2. Therefore ϵ_q is indeed a map of $\triangleleft$-monoids.

Next we want to show that ϵ is natural. Let $f \colon (p, \eta_p, \mu_p) \to (q, \eta_q, \mu_q)$ in **Mod**(**Poly**). It suffices to show for all α the following diagram commutes.

$$\begin{array}{ccc} p_{(\alpha)} & \xrightarrow{f_{(\alpha)}} & q_{(\alpha)} \\ {\scriptstyle \epsilon_{(\alpha)}}\downarrow & & \downarrow{\scriptstyle \epsilon_{(\alpha)}} \\ p & \xrightarrow[f]{} & q \end{array}$$

We show this by induction on α. For $\alpha = 0$, $f_{(\alpha)}$ is the identity on y and the diagram commutes because f preserves the unit.

Suppose that the diagram commutes for all $\alpha' < \alpha$. If α is a successor ordinal — say $\alpha = \alpha' + 1$ — then we want to show that the following diagram commutes.

$$\begin{array}{ccc} y + p \triangleleft p_{(\alpha')} & \xrightarrow{y + f \triangleleft f_{(\alpha')}} & y + q \triangleleft q_{(\alpha')} \\ {\scriptstyle (\eta_p, \mu_p \circ (p \triangleleft \epsilon_{(\alpha')}))}\downarrow & & \downarrow{\scriptstyle (\eta_q, \mu_q \circ (q \triangleleft \epsilon_{(\alpha')}))} \\ p & \xrightarrow{f} & q \end{array}$$

It commutes on the first term of the coproduct, again because f preserves the unit. To show that it commutes on the second term, we want to show that the outer diagram in the following commutes.

$$\begin{array}{ccc} p \triangleleft p_{(\alpha')} & \xrightarrow{f \triangleleft f_{(\alpha')}} & q \triangleleft q_{(\alpha')} \\ {\scriptstyle p \triangleleft \epsilon_{(\alpha')}}\downarrow & & \downarrow{\scriptstyle q \triangleleft \epsilon_{(\alpha')}} \\ p \triangleleft p & \xrightarrow[f \triangleleft f]{} & q \triangleleft q \\ {\scriptstyle \mu_p}\downarrow & & \downarrow{\scriptstyle \mu_q} \\ p & \xrightarrow{f} & q \end{array}$$

The top square commutes by the induction hypothesis and the bottom square commutes because f preserves multiplication.

If α is a limit ordinal, consider the cocones

$$p_{(\alpha')} \xrightarrow{f_{(\alpha')}} q_{(\alpha')} \xrightarrow{\epsilon_{(\alpha')}} q \qquad \text{and} \qquad p_{(\alpha')} \xrightarrow{\epsilon_{(\alpha')}} p \xrightarrow{f} q.$$

By uniqueness, the cocone on the left induces the map $p_{(\alpha)} \xrightarrow{f_{(\alpha)}} q_{(\alpha)} \xrightarrow{\epsilon_{(\alpha)}} q$. Also by uniqueness, the cocone on the right induces the map $p_{(\alpha)} \xrightarrow{\epsilon_{(\alpha)}} p \xrightarrow{f} q$. Furthermore, these cocones are identical by the induction hypothesis and so their induced maps are equal, as desired.

Finally, to show that the unit and counit form an adjunction we must show that for a polynomial p

$$\mathfrak{m}_p \xrightarrow{\zeta_{\mathfrak{m}_p}} \mathfrak{m}_{\mathfrak{m}_p} \xrightarrow{\epsilon_{\mathfrak{m}_p}} \mathfrak{m}_p$$

is the identity and that for a $\triangleleft$-monoid (q, η_q, μ_q)

$$q \xrightarrow{\zeta_q} \mathfrak{m}_q \xrightarrow{\epsilon_q} q$$

is the identity as well. It suffices to show the second as the first follows with $q = \mathfrak{m}_p$.

Since the diagrams below commutes, it suffices to show that $\epsilon_{(1)} \circ \lambda_{(0)}$ is the identity. This is immediate from the definition of $\epsilon_{(1)}$ and the unit law for (q, η_q, μ_q).

$$q \xrightarrow{\lambda_{(0)}} q_{(1)} \xrightarrow{\iota_{(1)}} \mathfrak{m}_q \xrightarrow{\epsilon_q} q, \qquad q_{(1)} \xrightarrow{\epsilon_{(1)}} q$$

□

C Proofs for the cofree comonad monad

Proposition 3.1. *There is a functor* $\mathfrak{c}_-\colon \mathbf{Poly} \to \mathbf{Poly}$ *such that* $\mathfrak{c}_p$ *has the structure of a* $\triangleleft$*-comonoid for each* $p : \mathbf{Poly}$,

$$\mathfrak{c}_p \to y \qquad \textit{and} \qquad \mathfrak{c}_p \to \mathfrak{c}_p \triangleleft \mathfrak{c}_p.$$

Proof. Given a polynomial p, define polynomials $p^{(i)}$ for $i \in \mathbb{N}$ by

$$p^{(0)} := y \qquad \text{and} \qquad p^{(1+i)} := y \times \left(p \triangleleft p^{(i)}\right)$$

There is a projection map $\pi^{(0)}\colon p^{(1)} \to p^{(0)}$, and if $\pi^{(i)}\colon p^{(1+i)} \to p^{(i)}$ has been defined, then we can define $\pi^{(1+i)} := y \times (p \triangleleft \pi^{(i)})$. Now define the polynomial

$$\mathfrak{c}_p := \lim \left(\cdots \xrightarrow{\pi^{(2)}} p^{(2)} \xrightarrow{\pi^{(1)}} p^{(1)} \xrightarrow{\pi^{(0)}} p^{(0)} \right) \tag{9}$$

and we note that this construction $p \mapsto \mathfrak{c}_p$ is natural in $p : \mathbf{Poly}$.

This polynomial comes equipped with a counit $\epsilon\colon \mathfrak{c}_p \to y = p^{(0)}$ given by the projection. We next construct the comultiplication $\delta\colon \mathfrak{c}_p \to \mathfrak{c}_p \triangleleft \mathfrak{c}_p$. Since $\triangleleft$ commutes with connected limits, we have

$$\mathfrak{c}_p \triangleleft \mathfrak{c}_p = \left(\lim_{i_1} p^{(i_1)}\right) \triangleleft \left(\lim_{i_2} p^{(i_2)}\right) \cong \lim_{i_1, i_2} \left(p^{(i_1)} \triangleleft p^{(i_2)}\right)$$

To obtain the comultiplication $\lim_i p^{(i)} \to \lim_{i_1,i_2}(p^{(i_1)} \triangleleft p^{(i_2)})$, it suffices to produce a natural choice of polynomial map $\varphi_{i_1,i_2} \colon p^{(i_1+i_2)} \to p^{(i_1)} \triangleleft p^{(i_2)}$ for any $i_1, i_2 : \mathbb{N}$. When $i_1 = 0$ or $i_2 = 0$, we use the unit identity for $\triangleleft$. By induction, assume given $\varphi_{i_1,1+i_2}$; we construct $\varphi_{1+i_1,1+i_2}$ as follows:

$$\begin{aligned}
p^{(1+i_1+1+i_2)} &= y \times \left(p \triangleleft p^{(i_1+1+i_2)}\right) \\
&\to y \times \left(p \triangleleft p^{(i_1)} \triangleleft p^{(1+i_2)}\right) && (10) \\
&\to \left(y \times p \triangleleft p^{(i_1)}\right) \triangleleft p^{(1+i_2)} && (11) \\
&= p^{(1+i_1)} \triangleleft p^{(1+i_2)}
\end{aligned}$$

where (10) is $\varphi_{i_1,1+i_2}$ and it remains to construct (11). Since $- \triangleleft q$ preserves products for any q, constructing (11) is equivalent to constructing two maps

$$y \times \left(p \triangleleft p^{(i_1)} \triangleleft p^{(1+i_2)}\right) \xrightarrow{\phi^{(i_1,i_2)}} p^{(1+i_2)} \qquad \text{and} \qquad y \times \left(p \triangleleft p^{(i_1)} \triangleleft p^{(1+i_2)}\right) \to p \triangleleft p^{(i_1)} \triangleleft p^{(1+i_2)}.$$

For the latter we use the second projection. The former, $\phi^{(i_1,i_2)} \colon p^{(1+i_1+1+i_2)} \to p^{(1+i_2)}$, is the more interesting one; for it we also use projections $p^{(i_1)} \to p^{(0)} = y$ and $\pi^{(i_2)} \colon p^{(i_2+1)} \to p^{(i_2)}$ to obtain:

$$y \times \left(p \triangleleft p^{(i_1)} \triangleleft p^{(1+i_2)}\right) \to y \times \left(p \triangleleft y \triangleleft p^{(i_2)}\right) \cong p^{(1+i_2)}$$

We leave the naturality of this to the reader.

It remains to check that ϵ and δ satisfy unitality and coassociativity. The base cases above imply unitality. Proving coassociativity amounts to proving that the following diagram commutes:

$$\begin{array}{ccc}
p^{(1+i_1+1+i_2+1+i_3)} & \xrightarrow{\phi^{(i_1,i_2+1+i_3)}} & p^{(1+i_2+1+i_3)} \\
{\scriptstyle \phi^{(i_1+1+i_2,i_3)}}\big\downarrow & & \big\downarrow{\scriptstyle \phi^{(i_2,i_3)}} \\
p^{(1+i_3)} & = & p^{(1+i_3)}
\end{array}$$

This can be shown by induction on i_3. □

Proposition C.1. *There is a monoidal structure on* $\mathfrak{c} \colon \mathbf{Poly} \to \mathbf{Poly}$

$$y \to \mathfrak{c}_y \qquad \textit{and} \qquad \mathfrak{c}_p \otimes \mathfrak{c}_q \to \mathfrak{c}_{p \otimes q}.$$

Proof. The polynomial $\mathfrak{c}_y \cong y^{\mathbb{N}}$ has a unique position, and this defines the first map. However, it is conceptually cleaner to realize that comonads are closed under $\otimes$ by duoidality, and hence both y and $\mathfrak{c}_p \otimes \mathfrak{c}_q$ carry comonad structures. Thus the desired maps are induced by the obvious polynomial maps $y \cong y$ and $\mathfrak{c}_p \otimes \mathfrak{c}_q \to p \otimes q$. It is straightforward to check that these are unital and associative. □

Theorem 3.2. *There is an adjunction*

$$\mathbf{Cat}^{\sharp} \underset{\mathfrak{c}_-}{\overset{U}{\underset{\longleftarrow}{\overset{\longrightarrow}{\Rightarrow}}}} \mathbf{Poly} \ .$$

Proof. We will abuse notation and denote the comonoid $(c, \epsilon, \delta) : \mathbf{Cat}^\sharp$ simply by its carrier c. We first provide the counit and unit of the desired adjunction. The counit

$$\epsilon_p \colon \mathfrak{c}_p \to p$$

is given by composing the projection map $\mathfrak{c}_p \to p^{(1)}$ from construction (9) with the projection $p^{(1)} \cong y \times p \to p$. Since $\mathfrak{c}_c$ is defined as a limit, the unit

$$\eta_c \colon c \nrightarrow \mathfrak{c}_c$$

will be given by defining maps $\eta^{(i)} \colon c \to c^{(i)}$ commuting with the projections $\pi^{(i)} \colon c^{(1+i)} \to c^{(i)}$, for each $i : \mathbb{N}$, and then showing that the resulting polynomial map η_c is indeed a cofunctor. Noting that $c^{(0)} = y$, we define

$$\eta^{(0)} := \epsilon$$

Given $\eta^{(i)} \colon c \to c^{(i)}$, we define $\eta^{(1+i)}$ as the composite

$$c \xrightarrow{(\epsilon,\delta)} y \times (c \triangleleft c) \xrightarrow{y \times (c \triangleleft \eta^{(i)})} y \times \left(c \triangleleft c^{(i)}\right) = c^{(1+i)}.$$

Clearly, we have $\eta^{(0)} = \pi^{(0)} \circ \eta^{(1)}$. It is easy to check that if $\eta^{(i)} = \pi^{(i)} \circ \eta^{(1+i)}$ then $\eta^{(1+i)} = \pi^{(1+i)} \circ \eta^{(2+i)}$. Thus we have constructed a polynomial map $\eta \colon c \to \mathfrak{c}_c$. It clearly commutes with the counit, so it suffices to show that η commutes with the comultiplication, which amounts to showing that the following diagram commutes

$$\begin{array}{ccccc}
c & \xrightarrow{\quad\delta\quad} & c \triangleleft c & \xrightarrow{(\epsilon,\delta)\triangleleft(\epsilon,\delta)} & (y \times (c \triangleleft c)) \triangleleft (y \times (c \triangleleft c)) \\
{\scriptstyle (\epsilon,\delta)}\big\downarrow & & & & \big\downarrow{\scriptstyle (y\times c\triangleleft\eta^{(i_1)})\triangleleft(y\times c\triangleleft\eta^{(i_2)})} \\
y \times (c \triangleleft c) & \xrightarrow[y\times c\triangleleft\eta^{(i_1+1+i_2)}]{} & y \times (c \triangleleft c^{(i_1+1+i_2)}) & \xrightarrow[\varphi_{1+i_1,1+i_2}]{} & (y \times (c \triangleleft c^{(i_1)})) \triangleleft (y \times (c \triangleleft c^{(i_2)}))
\end{array}$$

for all $i_1, i_2 : \mathbb{N}$, where $\varphi_{1+i_1,1+i_2}$ is the map constructed in Equation (10) and Equation (11). Commutativity follows from the counitality and coassociativity of the comonoid c.

The triangle identities are straightforward as well. Indeed, for any comonoid $c : \mathbf{Cat}^\sharp$, the composite $c \xrightarrow{U \circ \eta_c} \mathfrak{c}_c \xrightarrow{\epsilon_{Uc}} c$ is equal to the composite of $c \xrightarrow{(\epsilon,c)} c^{(1)} = y \times c$, with the projection $c^{(1)} \to c$, the result of which is the identity. Finally, for any polynomial $p : \mathbf{Poly}$, the composite $\mathfrak{c}_p \xrightarrow{\eta_{\mathfrak{c}_p}} \mathfrak{c}_{\mathfrak{c}_p} \xrightarrow{\mathfrak{c}_{\epsilon_p}} \mathfrak{c}_p$ is given by taking a limit of maps of the form

$$\mathfrak{c}_p \xrightarrow{(\epsilon,\delta)} y \times (\mathfrak{c}_p \triangleleft \mathfrak{c}_p) \xrightarrow{y\times(\mathfrak{c}_p \triangleleft \eta^{(i)})} y \times (\mathfrak{c}_p \triangleleft \mathfrak{c}_p^{(i)}) \xrightarrow{y \times (\epsilon_p \triangleleft \epsilon_p^{(i)})} y \times (p \triangleleft p^{(i)})$$

Each one is in fact the projection $\mathfrak{c}_p \to p^{(i+1)}$, so the resulting map is the identity on $\mathfrak{c}_p$, completing the proof. □

D Proofs for the module structure $\mathfrak{m}_p \otimes \mathfrak{c}_q \to \mathfrak{m}_{p \otimes q}$

Proposition 3.3. *The maps $\Xi_{p,q}$ are natural in p and q.*

Proof. To show that ψ is natural in p, it suffices to show that for a maps $p \to p'$ and $q \to q'$ in **Poly** the following diagram commutes:

$$\begin{array}{ccc} p & \longrightarrow & [\mathfrak{c}_q, \mathfrak{m}_{p\otimes q}] \\ \downarrow & & \downarrow \\ p' & \longrightarrow & [\mathfrak{c}_{q'}, \mathfrak{m}_{p'\otimes q'}] \end{array}$$

This follows immediately from the commutativity of the following diagram.

$$\begin{array}{ccccc} p \otimes \mathfrak{c}_q & \longrightarrow & p \otimes q & \longrightarrow & \mathfrak{m}_{p\otimes q} \\ \downarrow & & \downarrow & & \downarrow \\ p' \otimes \mathfrak{c}_{q'} & \longrightarrow & p' \otimes q' & \longrightarrow & \mathfrak{m}_{p'\otimes q'} \end{array}$$

Note that the square on the left commutes by naturality of the counit of the adjunction in Theorem 3.2. The square on the right commutes by naturality of unit of the adjunction in Theorem 2.10. □

Theorem 3.4. *There is a left-module over* $\mathfrak{c}_- : (\mathbf{Poly}, \otimes, y) \to (\mathbf{Poly}, \otimes, y)$ *consisting of:*

- **Poly** *as a left module category over* $(\mathbf{Poly}, \otimes, y)$*.*
- *The functor* $\mathfrak{m}_- \colon \mathbf{Poly} \to \mathbf{Poly}$*.*
- *The natural transformation* $\Xi \colon \mathfrak{m}_- \otimes \mathfrak{c}_- \Rightarrow \mathfrak{m}_{-\otimes -}$*.*

Proof. We must show that two diagrams commute. First, we will show that the following diagram commutes.

$$\begin{array}{ccc} y \otimes \mathfrak{m}_r & \xrightarrow{\cong} & \mathfrak{m}_r \\ \downarrow & & \uparrow \cong \\ \mathfrak{c}_y \otimes \mathfrak{m}_r & \xrightarrow[\Xi_{y,r}]{} & \mathfrak{m}_{y\otimes r} \end{array}$$

It suffices to show that the diagram below commutes:

$$\begin{array}{ccccc} y \otimes r & \longrightarrow & r & \longrightarrow & \mathfrak{m}_r \\ \downarrow & & \uparrow & & \uparrow \\ \mathfrak{c}_y \otimes r & \longrightarrow & y \otimes r & \longrightarrow & \mathfrak{m}_{y\otimes r} \end{array}$$

The square on the left commutes because $y \to \mathfrak{c}_y \to y$ is the identity. The square on the right commutes by the naturality of the unit of the adjunction in Theorem 2.10.

Next we must show that the following diagram commutes.

$$\begin{array}{ccc} \mathfrak{c}_p \otimes \mathfrak{c}_q \otimes \mathfrak{m}_r & \longrightarrow & \mathfrak{c}_p \otimes \mathfrak{m}_{q\otimes r} \\ \downarrow & & \downarrow \\ \mathfrak{c}_{p\otimes q} \otimes \mathfrak{m}_r & \longrightarrow & \mathfrak{m}_{p\otimes q\otimes r} \end{array}$$

It suffices to show that the following diagram commutes.

$$\begin{array}{ccccc} \mathfrak{c}_p \otimes \mathfrak{c}_q \otimes r & \longrightarrow & \mathfrak{c}_p \otimes q \otimes r & \longrightarrow & \mathfrak{c}_p \otimes \mathfrak{m}_{q\otimes r} \\ \downarrow & & \downarrow & & \downarrow \Xi_{p,q\otimes r} \\ \mathfrak{c}_{p\otimes q} \otimes r & \longrightarrow & p \otimes q \otimes r & \longrightarrow & \mathfrak{m}_{p\otimes q\otimes r} \end{array}$$

The square on the left commutes by definition of the laxator of $\mathfrak{c}_-$ while the square on the right commutes by definition of $\Xi_{p,q\otimes r}$. □