Polynomials in homotopy type theory as a Kleisli category

Elies Harington[1] and Samuel Mimram[1]

[1]LIX, CNRS, École polytechnique, Institut Polytechnique de Paris, 91120 Palaiseau, France.

March 2024

Abstract

Polynomials in a category have been studied as a generalization of the traditional notion in mathematics. Their construction has recently been extended to higher groupoids, as formalized in homotopy type theory, by Finster, Mimram, Lucas and Seiller, thus resulting in a cartesian closed bicategory. We refine and extend their work in multiple directions. We begin by generalizing the construction of the free symmetric monoid monad on types in order to handle arities in an arbitrary universe. Then, we extend this monad to the (wild) category of spans of types, and thus to a comonad by self-duality. Finally, we show that the resulting Kleisli category is equivalent to the traditional category of polynomials. This thus establishes polynomials as a (homotopical) model of linear logic. In fact, we explain that it is closely related to a bicategorical model of differential linear logic introduced by Melliès.

1 Introduction

The notion of *polynomial functor* in category theory, surveyed in [11], generalizes the traditional well-known notion of polynomial in algebra. Namely, a polynomial functor is an endofunctor of sets of the form $X \mapsto \sum_{b \in B} X^{E_b}$ for some family $(E_b)_{b \in B}$ of sets which is called the *polynomial* inducing the functor. This formulation should make clear the relationship with traditional polynomials: such a functor sends an object X to a (categorical) sum of monomials, which are powers of X. Moreover, those have been generalized in various ways: typed variants can be considered, and one can make sense of polynomials in any locally cartesian closed category, as well as in type theory where they correspond to the notion of *container* [1].

The category of polynomial functors over set is cartesian, but the expected closure for the cartesian product cannot be defined as easily as expected. Firstly, the right adjoint to the cartesian product does not exist for size issues: in order to have a chance to define it, one has to restrict to polynomial functors which are *finitary*, i.e. those which are sums of a finite number of monomials of finite degree, which were also named *normal functors* by Girard [13]. If the category of finitary polynomial functors is indeed cartesian closed, it was observed early on by Girard and Taylor [13, 30] that it is not so in a satisfactory way: the category of finitary polynomial functors is actually a 2-category, but the closure fails to extend as a 2-categorical one. Moreover, the 2-categorical structure is necessary in order to directly consider polynomials (as opposed to polynomial functors): those form a bicategory (and not a category in any reasonable way). In order to handle this discrepancy, it was proposed by Finster et al. [8], following ideas from Kock and collaborators [20, 12], that this could be suitably resolved by considering finitary polynomials over groupoids instead of sets. However, the constructions cannot performed naively

(for instance, it is well-known that the category of groupoids is not cartesian closed): in order for this definition to make sense, all the constructions, such as the limits involved in the definition of the composition of polynomials, have to be taken "up to homotopy". A definition of polynomials along these lines has been performed in the context of 2-categories by Kock [20] and ∞-categories by himself, Gepner and Haugseng [12], at the cost of requiring the use of quite advanced technical tools [23].

It is advocated in [8] that a good setting to define and study polynomials is the one of homotopy type theory [31]. In this framework, types are equipped with a structure of ∞-groupoid [21, 32] and can be interpreted as spaces [18], over which one can perform constructions which are homotopy invariant by definition. In particular, the notions of finite (co)limits, which are easy to define, are actually *homotopy* (co)limits. Following these ideas, polynomials have then successfully been defined in homotopy type theory and have been shown to bear a structure of cartesian closed bicategory, when restricting to finitary ones [8] (up to actually truncating the category, as explained in section 6.1).

In this article, we further study this model and its structure on multiple aspects. First, we show that the notion of *finiteness*, which is used in order to define finitary polynomials, can actually be taken relatively to an arbitrary universe $\mathcal{V}$ (which might be taken to be the one of finite types in order to recover the traditional notion). More importantly, we show here that the cartesian closed structure of the category of finitary polynomials actually comes from what we expect to be an ∞-categorical model of linear logic. More precisely, the category of (finitary) spans can be seen as a subcategory of the category of polynomials, which formalizes the intuition that spans are "linear" polynomials, and the resulting inclusion functor has a left adjoint:

$$\text{Poly} \overset{\frown}{\underset{\smile}{\perp}} \text{Span}$$

Moreover, this adjunction is compatible with the monoidal structures of the two categories (induced by the cartesian structure in the case of polynomials). We thus obtain what is known as a *linear-non-linear adjunction* [3, 26], a notion which has been established as the standard categorical model of the multiplicative exponential fragment of linear logic. The adjunction induces a comonad on the category of spans, which can be understood as the exponential modality !, and we show here that the category Poly is actually the Kleisli category associated to the comonad: such a model is known as a *Seely category* [27, 4].

To be more precise, the above is the situation that we expect to be holding, but showing that it is formally the case in homotopy type theory is currently out of reach because there is no known definition of ∞-categories in homotopy type theory [5]. For this reason, we work here with *wild categories*, which can be understood as ∞-categories, without requiring higher coherences. Our result should however extend to ∞-categories provided that we can define those. We namely expect that the constructions we provide here are actually coherent, and moreover the constructions defined by universal properties will automatically extend to the coherent setting: for instance, our result that Span is a cartesian wild category immediately implies that it is a cartesian ∞-category, provided that we can show that Span is an ∞-category. We also explain that, by truncation, this wild category induces a (univalent) category which is a model of linear logic: this model is apparently new and of a very different nature than the traditional ones. Moreover, we explain that the wild category can be extended into a 2-coherent one, which induces a bicateogry, thus allowing us to recover the previous construction of [8]. This model of linear logic we obtain coincides with one constructed by Melliès in order to build a model of differential linear logic [25]: the observation that the Kleisli category associated to the exponential comonad is the category of polynomials is however new. Moreover, working in homotopy type theory allows us to simplify constructions and avoid invoking arguments based on model categories [15].

Plan of the paper. We begin in section 2 by recalling the general setting of homotopy type theory (section 2.1), as well as the notion of wild category (section 2.2) and classical structures for those (section 2.2.1). In section 3, we define the wild category of spans and equip it with a structure of symmetric monoidal wild category with finite products and coproducts. We then construct, in section 4, the free commutative monoid (co)monad on spans, from which we construct a structure of Seely category (theorem 35) and thus a model of linear logic (theorem 36). Finally, in section 5, we show that the Kleisli category associated to the comonad is precisely the one of polynomials (theorem 45). In section 6, we compare with related works: the bicategory of polynomials of [8] (section 6.1) and template games of [25] (section 6.2).

2 Categories in homotopy type theory

In this section, we recall the definition of wild categories in homotopy type theory. We suppose the reader already familiar with the general setting and refer to the reference book [31] for details.

2.1 Homotopy type theoretic definitions and notations

Constructions on types. Given a type A, we write $x : A$ to indicate that x is an element of type A. We write $\mathcal{U}$ for an arbitrary universe of types, so that its elements $A : \mathcal{U}$ are types. The initial and terminal types are respectively denoted $\varnothing$ and $\mathbb{1}$. Given two types A and B, we respectively write $A \times B$, $A \sqcup B$ and $A \to B$ for their product, coproduct and function types. Given $b : B$, we write $\mathrm{cst}_b : A \to B$ for the constant function whose image is b. Given a type A and a type family $B : A \to \mathcal{U}$, the corresponding dependent sum and function types are respectively written $\sum_{x:A} B(x)$ and $\prod_{x:A} B(x)$. In the former situation, the first and second projections are respectively written $\pi_1 : \Sigma AB \to A$ and $\pi_2 : \prod_{x:\Sigma AB} B(\pi_1(x))$ (and similarly for the non-dependent projections $\pi_1 : A \times B \to A$ and $\pi_2 : A \times B \to B$). Given types A, B, we write $\iota_1 : A \to A \sqcup B$ and $\iota_2 : B \to A \sqcup B$ for the canonical inclusions, and given a type C and maps $f : A \to C$, $g : B \to C$, we write $\langle f, g \rangle$ for the induced map $A \sqcup B \to C$. Given $f : A \to C, g : B \to D$, we write $f \sqcup g$ for the canonical map $A \sqcup B \to C \sqcup D$ (i.e. $f \sqcup g = \langle \iota_1 \circ f, \iota_2 \circ g \rangle$).

Paths. We write $x := y$ to indicate that x is defined to be y, and $x \equiv y$ when x is equal to y by definition. The type theory also features a propositional notion of equality: given two elements x and y of a type A, we write $x =_A y$ (or $x = y$) for the corresponding identity type, which features a distinguished proof $\mathrm{refl}_x : x = x$ of reflexivity for any element x. Given two identities $p : x = y$ and $q : y = z$, we write $p \cdot q : x = z$ for the identity obtained by transitivity. The elimination principle for paths (known as *path induction*) roughly states that in order to show that a property $P : \prod_{x:A} \prod_{p:a=x} \mathcal{U}$ holds for every possible values of x and p, it is enough to show it for $x := a$ and $p := \mathrm{refl}_a$. By using this, it is for instance easy to show that any function $f : A \to B$ induces a function $\mathrm{ap}_f : (x = y) \to (f(x) = f(y))$ defined by $\mathrm{ap}_f(\mathrm{refl}_x) := \mathrm{refl}_{f(x)}$ witnessing that equality is a congruence.

Univalence. A map $f : A \to B$ is an equivalence when it admits both a left and a right inverse. We write $A \simeq B$ for the type of equivalences between A and B. An identity is always an equivalence and thus, by path induction, there is a canonical map $(A = B) \to (A \simeq B)$. The *univalence axiom* states that this map is an equivalence.

Homotopy levels. A type A is *contractible* when it satisfies $\sum_{a:A} \prod_{x:A} a = x$, which can be thought of as the fact that A is a point up to homotopy. The type A is a *proposition* (resp. a *set*, a *groupoid*) when, for every $x, y : A$, the type $x = y$ is contractible (resp. a proposition,

a set). Given a type A, we write $\|A\|_{-1}$ for its *propositional truncation*, which is the universal way of turning it into a proposition (and the set truncation $\|A\|_0$ and groupoid truncation $\|A\|_1$ are defined similarly), with $|-|_n : A \to \|A\|_n$ as canonical quotient maps. We write $\mathrm{Set}_{\mathcal{U}}$ (resp. $\mathrm{Gpd}_{\mathcal{U}}$) for the type of sets (resp. groupoids) in the universe $\mathcal{U}$. Those constructions are detailed in [31, Chapter 7].

Fibered / indexed equivalence. Any function $f : A \to B$ induces a type family $\mathrm{fib}_f : B \to \mathcal{U}$ such that $\mathrm{fib}_f(b) := \sum_{a:A} (f(a) = b)$ is the *fiber* of f at b. Conversely, any type family $F : B \to \mathcal{U}$ induces a function which is the first projection $\pi_1 : \Sigma B F \to B$ from the *total space* $\Sigma B F$. These two operations form an equivalence between types over B and types families indexed by B [31, Section 4.8].

2.2 Wild categories

The notion of *wild category* [6] can be understood as a "non-coherent" variant of the notion of ∞-category, where we do not require the expected coherences in dimension $n > 1$. In this way, wild categories can be understood as a first step toward ∞-categories. They also constitute the right axiomatization when the objects of morphisms are sets [31, Chapter 9]. As of today, coherences can be written for $n = 2$, i.e. we can define bicategories, we expect that it might be possible to define tricategories following the traditional categorical definition [14] but not much more, and other approaches based on complete semi-segal spaces allow writing definitions which are coherent up to an arbitrary dimension. However, whether ∞-categories can be defined in full generality in homotopy type theory is still an open question, and perhaps one of the most important ones [5]. We first recall the notion of wild precategory, which is then refined into the one of wild category by further imposing a univalence property.

Definition 1. A *wild precategory* $\mathcal{C}$ consists of type families

$$\mathrm{Ob}_{\mathcal{C}} : \mathcal{U} \qquad\qquad \mathrm{Hom}_{\mathcal{C}} : \mathrm{Ob}_{\mathcal{C}} \times \mathrm{Ob}_{\mathcal{C}} \to \mathcal{U}$$

for objects and morphisms (we omit the indices in the following), together with composition and identity operations

$$- \circ - : \prod_{x,y,z:\mathrm{Ob}} \mathrm{Hom}(y,z) \times \mathrm{Hom}(x,y) \to \mathrm{Hom}(x,z) \qquad \mathrm{id} : \prod_{x:\mathrm{Ob}} \mathrm{Hom}(x,x)$$

and coherence identities witnessing for associativity

$$\alpha : \prod_{x,y,z,t:\mathrm{Ob}} \prod_{h:\mathrm{Hom}(z,t)} \prod_{g:\mathrm{Hom}(y,z)} \prod_{f:\mathrm{Hom}(x,y)} h \circ (g \circ f) = (h \circ g) \circ f$$

as well as left and right unitality

$$\lambda : \prod_{x,y:\mathrm{Ob}} \prod_{f:\mathrm{Hom}(x,y)} f \circ \mathrm{id}_x = f \qquad\qquad \rho : \prod_{x,y:\mathrm{Ob}} \prod_{f:\mathrm{Hom}(x,y)} \mathrm{id}_y \circ f = f$$

For convenience, we write $x : \mathcal{C}$ for $x : \mathrm{Ob}$ and $\mathcal{C}(x,y)$ for $\mathrm{Hom}(x,y)$.

Given a morphism $f : \mathcal{C}(x,y)$ for some objects $x, y : \mathcal{C}$, we say that f is an isomorphism when it admits both a left and a right inverse [6]. We write

$$\mathrm{isIso}(f) := \Big(\sum_{g:\mathcal{C}(y,x)} g \circ f = \mathrm{id}_x \Big) \times \Big(\sum_{g:\mathcal{C}(y,x)} f \circ g = \mathrm{id}_y \Big)$$

for the corresponding predicate. We also write $x \cong y$ for the type of isomorphisms in $\mathcal{C}(x,y)$.

Lemma 2. *Being an isomorphism is a proposition.*

Proof. Fix a morphism $f : \mathcal{C}(x,y)$ and suppose $((g,p),(h,q)) : \mathrm{isIso}(f)$. Then we have

$$g = g \circ (f \circ h) = (g \circ f) \circ h = h.$$

Hence g is both a left and right inverse to f. Thus the type $\left(\sum_{g:\mathcal{C}(y,x)} g \circ f = \mathrm{id}_x \right)$ is equivalent to $\sum_{g':\mathcal{C}(y,x)} g' = g$, which is contractible, and similarly for the second component of $\mathrm{isIso}(f)$. We proved that $\mathrm{isIso}(f) \to \mathrm{isContr}(\mathrm{isIso}(f))$, which implies that $\mathrm{isIso}(f)$ is a proposition. Note that a similar property is shown in [31, Lemma 9.1.3] for categories, but the proof relies on the fact that hom types are sets (definition 4). $\square$

Given a wild precategory $\mathcal{C}$, the identity id_x on an arbitrary object x is always an isomorphism. We can thus define a map $\prod_{x,y:\mathcal{C}} (x = y) \to (x \cong y)$ by path induction such that, for $x : \mathcal{C}$, the image of (x, x, refl_x) is id_x.

Definition 3. A wild precategory is *univalent* when the canonical map $(x = y) \to (x \cong y)$ is an equivalence of types for all $x, y : \mathcal{C}$. A *wild category* is a univalent wild precategory.

Note that since being an equivalence is a proposition, being univalent for a wild precategory is also a proposition.

The notion of (pre)category can be recovered as a specialization of the previous notion [31, Definitions 9.1.1 and 9.1.6]:

Definition 4. A *precategory* (resp. *category*) $\mathcal{C}$ is a wild precategory (resp. category) such that $\mathcal{C}(x,y)$ is a set for every objects $x, y : \mathcal{C}$.

Given a universe $\mathcal{U}$, there is a wild category $\mathrm{Type}_\mathcal{U}$ whose objects are types in $\mathcal{U}$, morphisms are maps of types, composition and identities are the usual ones, and unitality and associativity hold definitionally. Because of this, the above constructions generalize, to wild categories, traditional ones for types (for instance, the notion of univalence for wild precategories coincides with the usual one for types).

The following construction ensures that one can always formally turn a wild category into a regular one. Note that, by univalence, the type of objects in a category has to be a groupoid [31, Lemma 9.1.8].

Definition 5. Given a wild precategory $\mathcal{C}$, we write $\|\mathcal{C}\|_1$ for the precategory whose type of objects is $\|\mathrm{Ob}_\mathcal{C}\|_1$, whose type of morphisms is the map $\|\mathrm{Hom}_\mathcal{C}\|_0 : (x, y : \|\mathrm{Ob}_\mathcal{C}\|_1) \to \mathrm{Set}_\mathcal{U}$ induced by $\mathrm{Hom}_\mathcal{C}$ (using the fact that $\mathrm{Set}_\mathcal{U}$ is a groupoid [31, Theorem 7.1.11]), and composition, identities and coherence laws are induced similarly by those of $\mathcal{C}$ by truncation elimination.

Proposition 6. *The precategory $\|\mathcal{C}\|_1$ is univalent when $\mathcal{C}$ is.*

Proof. We write

$$\mathrm{pathToEquiv}_{x,y} : (x =_{\mathrm{Ob}_{\|\mathcal{C}\|_1}} y) \to (x \cong y)$$

for the canonical function associating to two equal objects of $\|\mathcal{C}\|_1$ an isomorphism between them: our aim is to show that this map is an equivalence. Since being an isomorphism is a proposition (lemma 2) and thus a groupoid [31, Theorem 7.1.7], we can suppose given $x, y : \mathcal{C}$ and we have to show $(|x|_1 = |y|_1) \simeq \|x \cong y\|_0$. We know that $|x|_1 = |y|_1$ is equivalent to $\|x = y\|_0$ [31, Theorem 7.3.12], which in turn is equivalent to $\|x \cong y\|_0$ by univalence of $\mathcal{C}$. Finally, we can show $\|x \cong y\|_0 \simeq (|x|_1 \cong |y|_1)$ by directly constructing an isomorphism between two types. $\square$

We expect that the above map is a left adjoint to the forgetful functor from wild (pre)categories to (pre)categories, we leave the proof of this fact for future work.

2.2.1 Structure in wild categories

Most of the usual categorical structures related to categories extend in the expected way to wild (pre)categories. In fact, many of those are developed in [31, Chapter 9] in the setting of homotopy type theory for categories, but immediately generalize to wild categories. Let us list a few instances of this which will be used in the following.

- The cartesian product $\mathcal{C} \times \mathcal{D}$ of wild (pre)categories $\mathcal{C}$ and $\mathcal{D}$ is the wild (pre)category with $\mathrm{Ob}_{\mathcal{C} \times \mathcal{D}} := \mathrm{Ob}_{\mathcal{C}} \times \mathrm{Ob}_{\mathcal{D}}$ and $(\mathcal{C} \times \mathcal{D})((x, x'), (y, y')) := \mathcal{C}(x, y) \times \mathcal{D}(x', y')$ with expected composition and identities.

- A *functor* $F : \mathcal{C} \to \mathcal{D}$ between wild (pre)categories consists of a function $F : \mathrm{Ob}_{\mathcal{C}} \to \mathrm{Ob}_{\mathcal{D}}$ between the respective types of objects, together with functions $F_{x,y} : \mathcal{C}(x, y) \to \mathcal{D}(Fx, Fy)$ for $x, y : \mathcal{C}$ as well as identities $F_x \mathrm{id}_x = \mathrm{id}_{Fx}$ for $x : \mathcal{C}$ and $F(g \circ f) = Fg \circ Ff$ for composable f and g. We can define identity and composite functors.

- A *natural transformation* $\alpha : F \to G$ between functors $F, G : \mathcal{C} \to \mathcal{D}$ is a family $\alpha_x : \mathcal{D}(Fx, Gx)$ of morphisms indexed by $x : \mathcal{C}$, equipped with equalities witnessing for the commutation of the expected diagrams.

- A *monad* $T : \mathcal{C} \to \mathcal{C}$ is an endofunctor equipped with natural transformations $\mu : T \circ T \Rightarrow T$ and $\eta : \mathrm{id}_{\mathcal{C}} \Rightarrow T$ together with equalities witnessing for the commutation of the expected diagrams (and comonads are defined dually).

- A *monoidal* wild precategory is a wild precategory $\mathcal{C}$ equipped with an object $\mathbb{1} : \mathcal{C}$ and a functor $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ together with expected natural isomorphisms (witnessing for associativity, and left and right unitality) and identities witnessing for the commutation of the two expected diagrams.

Given a comonad (T, δ, ε) on a wild precategory $\mathcal{C}$, we write $\mathcal{C}_T$ for the Kleisli wild precategory whose objects are the same as $\mathcal{C}$ and morphisms are $\mathcal{C}_T(x, y) = \mathcal{C}(Tx, y)$ with the usual notion of composition and identities [24, Section VI.5].

Structures specified by universal properties can of course also be translated to the setting of wild (pre)categories. For instance, an object x of a wild precategory $\mathcal{C}$ is *terminal* when for every $y : \mathcal{C}$ the type $\mathcal{C}(y, x)$ is contractible. Following the usual categorical proof, two terminal objects are always isomorphic. When $\mathcal{C}$ is univalent the terminal object is unique: the previous isomorphism implies that any two terminal objects are equal, i.e. that the type of terminal objects is a proposition. The notion of *cartesian product* of two objects $x, y : \mathcal{C}$ can be defined as a terminal object in the wild precategory of spans $x \leftarrow z \to y$ in $\mathcal{C}$. We write $\pi_1 : x \times y \to x$ and $\pi_2 : x \times y \to y$ for the two projections. Similarly, one can define the notion of *pullback* of two cofinal morphisms in a wild precategory. Finally, the dual of previous notions (initial objects, coproducts, etc.) can also be defined. Any wild category with finite products is canonically equipped with a symmetric monoidal structure, called the *cartesian symmetric monoidal structure*, with the tensor product induced by cartesian product and distinguished object being the terminal object.

Our aim in this article is to show that the wild category of spans has the structure of a wild Seely category. This notion generalizes to wild categories the notion of *Seely category*, in the same way as above. It was introduced in [27] and then refined by Bierman [4] who added a missing axiom, in order to provide a notion of categorical model of intuitionistic linear logic. A detailed presentation, as well as relationship with other categorical models of linear logic, can be found in [26].

Definition 7. A *wild Seely category* is a symmetric monoidal closed wild category $(\mathcal{C}, \otimes, \mathbb{1})$, which is moreover cartesian, with binary product and unit respectively denoted by $\&$ and $\top$, together with a comonad $(!, \delta, \varepsilon)$ on $\mathcal{C}$ called the *exponential* which is symmetric monoidal, with structural natural isomorphisms $m^2_{A,B} : !(A\&B) \to !A \otimes !B$ and $m^0_{\top} : !\top \to \mathbb{1}$ making $! : (\mathcal{C}, \otimes, \mathbb{1}) \to (\mathcal{C}, \&, \top)$ into a symmetric monoidal functor, and such that moreover the following diagram commutes:

$$
\begin{array}{ccccc}
!(A\&B) & \xrightarrow{\ \delta_{A\&B}\ } & !!(A\&B) & \xrightarrow{\ !(!\pi_1, !\pi_2)\ } & !(!A\&!B) \\
{\scriptstyle m^2_{A,B}}\downarrow & & & & \downarrow{\scriptstyle m^2_{!A,!B}} \\
!A \otimes !B & & \xrightarrow{\hspace{4cm}\delta_A \otimes \delta_B\hspace{4cm}} & & !!A \otimes !!B
\end{array}
$$

The morphisms $m^2_{A,B}$ and m^0 are often called *Seely isomorphisms*.

Any Seely category induces a linear-non-linear adjunction between itself and its Kleisli category [26].

3   Wild categories of spans

Definition 8. Given a wild category $\mathcal{C}$ with pullbacks, its wild precategory of *spans*, noted $\mathrm{Span}(\mathcal{C})$ or $\mathrm{Span}_{\mathcal{C}}$, is the wild precategory whose objects are the same as for $\mathcal{C}$ and whose morphisms are spans $a \xleftarrow{s} x \xrightarrow{t} b$, written $(s,t) : a \twoheadrightarrow b$. The identities are $a \xleftarrow{\mathrm{id}_a} a \xrightarrow{\mathrm{id}_a} a$ and composition is induced by pullback:

$$
\begin{array}{ccccccccc}
 & & & & x \times_b y & & & & \\
 & & {\scriptstyle \pi_1}\swarrow & & \vee & & \searrow{\scriptstyle \pi_2} & & \\
 & & x & & & & y & & \\
 & {\scriptstyle s}\swarrow & & \searrow{\scriptstyle t} & & {\scriptstyle u}\swarrow & & \searrow{\scriptstyle v} & \\
a & & & & b & & & & c
\end{array}
\tag{1}
$$

As explained in section 2.2.1, univalence entails that objects defined by universal properties are unique, so that defining composition by pullback in a category does not require any "choice" of pullbacks as is usually the case in set theory. Unitality, associativity (and eventual higher coherences) of $\mathrm{Span}(\mathcal{C})$ also follow from the univalence of $\mathcal{C}$ and the universal properties of iterated pullbacks. Moreover, we have that

Proposition 9. *Given a wild category $\mathcal{C}$, the wild precategory $\mathrm{Span}(\mathcal{C})$ is univalent.*

This proposition follows easily from the gollowing lemma, together with the univalence of $\mathcal{C}$.

Lemma 10. *A span $(s,t) : a \twoheadrightarrow b$ is an isomorphism in $\mathrm{Span}(\mathcal{C})$ if and only if the morphisms s and t are isomorphisms in $\mathcal{C}$.*

Proof. Suppose given a span $(s,t) : a \twoheadrightarrow b$ which admits an inverse $(u,v) : b \twoheadrightarrow a$. Since $(u,v) \circ (s,t) = \mathrm{id}_a$, with the notations of (1), we have $\pi_1 \circ s = \mathrm{id}_a$ and therefore s admits a left inverse. Similarly, s admits a right inverse and is thus invertible, and similarly t is invertible. $\square$

Proof of proposition 9. Let a, b be objects of $\mathcal{C}$. By lemma 10 and univalence of $\mathcal{C}$, we have

$$
(a \simeq_{\mathrm{Span}} b) \quad \simeq \quad \sum_{x:\mathcal{U}} (x \simeq_{\mathcal{C}} a) \times (x \simeq_{\mathcal{C}} b) \quad \simeq \quad \sum_{x:\mathcal{U}} (x = a) \times (x = b) \quad \simeq \quad (a = b)
$$

where the last step follows from classical identities and contractibility of singleton types [31, Lemma 3.11.8]. $\square$

Remark 11. As evident from the definition, the category $\mathrm{Span}(\mathcal{C})$ is self-dual: there is an involutive contravariant functor $\mathrm{Span}(\mathcal{C})^{\mathrm{op}} \to \mathrm{Span}(\mathcal{C})$ acting as the identity on morphisms and switching the two legs of morphisms.

3.1 Functoriality of the Span construction

The goal of this subsection is to address general question of the form: "given some structure on $\mathcal{C}$, does it lift to $\mathrm{Span}(\mathcal{C})$?" An ∞-categorical statement that would imply all of the results we prove in this section would be along the lines of

Conjecture 12. *The* Span *construction underlies a limit-preserving $(\infty, 2)$-functor from the $(\infty, 2)$-category of categories with pullbacks, functors preserving pullbacks, and cartesian natural transformations, to the $(\infty, 2)$-category of categories. Propositions 13 to 15 together with the fact that we have the obvious equivalence $\mathrm{Span}(\mathcal{C} \times \mathcal{D}) \simeq \mathrm{Span}(\mathcal{C}) \times \mathrm{Span}(\mathcal{D})$ form a 2-categorical approximation to this conjecture.*

However, since we cannot state this result in full generality in homotopy type theory, let alone prove it, we detail how the span construction acts on low-dimensional cells, namely functors and natural transformations.

Proposition 13. *Let $\mathcal{C}$ and $\mathcal{D}$ be wild categories with pullbacks and $F : \mathcal{C} \to \mathcal{D}$ be a functor. If F preserves pullbacks, then it lifts to a functor $\mathrm{Span}(F) : \mathrm{Span}(\mathcal{C}) \to \mathrm{Span}(\mathcal{D})$. $\mathrm{Span}(F)$ acts as F on objects, and given a span $(f, g) : a \nrightarrow b$, we have $F(f, g) := (Ff, Fg) : Fa \nrightarrow Fb$.*

Proof. We define $\mathrm{Span}(F)$ on objects by $\mathrm{Span}(F)(a) := Fa$ and on morphisms by

$$\mathrm{Span}(F)(a \xleftarrow{f} x \xrightarrow{g} b) := Fa \xleftarrow{Ff} Fx \xrightarrow{Fg} Fb$$

That $\mathrm{Span}(F)$ preserves identities immediately follows from the fact that F does, and that F preserves composition of spans is precisely the statement that F preserves pullbacks. $\qquad \square$

Before explaining how to lift natural transformations to categories of spans, we explain how to lift morphisms. Given objects $a, b : \mathcal{C}$, any morphism $f : a \to b$ canonically induces two spans

$$Lf : a \nrightarrow b := a \xleftarrow{\mathrm{id}_a} a \xrightarrow{f} b \qquad \text{and} \qquad Rf : b \nrightarrow a := b \xleftarrow{f} a \xrightarrow{\mathrm{id}_a} a$$

Moreover, we have $L(g \circ f) = Lg \circ Lf$ since the center square in the following diagram is a pullback square:

This makes L into a covariant functor $L : \mathcal{C} \to \mathrm{Span}(\mathcal{C})$. Similarly, R extends to a contravariant functor $R : \mathcal{C}^{\mathrm{op}} \to \mathrm{Span}(\mathcal{C})$. Actually, R is just L precomposed to the self-duality functor of remark 11.

Proposition 14. *Let $F, G : \mathcal{C} \to \mathcal{D}$ be pullback-preserving functors between categories with pullbacks, and $\alpha : F \Rightarrow G$ be a cartesian natural transformation. Then α extends as a natural transformation $\mathrm{Span}(\alpha) : \mathrm{Span}(F) \Rightarrow \mathrm{Span}(G)$.*

Proof. The natural transformation $\mathrm{Span}(\alpha)$ is given on objects by $\mathrm{Span}(\alpha)_a := L\alpha_a : Fa \twoheadrightarrow Ga$ for a in $\mathrm{Span}(\mathcal{C})$. Now let $a \xleftarrow{f} x \xrightarrow{g} b$ be a morphism in $\mathrm{Span}(\mathcal{C})$. The corresponding naturality square is the following square of spans:

$$
\begin{array}{ccccc}
Fa & \xleftarrow{\quad Ff \quad} & Fx & \xrightarrow{\quad Fg \quad} & Fb \\
\mathrm{id}_{Fa} \uparrow & (1) & \mathrm{id}_{Fx} & \quad & \uparrow \mathrm{id}_{Fb} \\
Fa & \xleftarrow{\ } Fa \times_{Ga} Gx \xleftarrow{\ \ h\ \ } & Fx & \dashrightarrow & Fb \\
\alpha_a \downarrow & & (2) \ \swarrow \alpha_x & (3) & \downarrow \alpha_b \\
Ga & \xleftarrow{\quad Gf \quad} & Gx & \xrightarrow{\quad Gg \quad} & Gb
\end{array}
$$

where the dotted squares are pullbacks, the middle arrow h is induced by universal property of the pullback $Fa \times_{Ga} Gx$ from the naturality square

$$
\begin{array}{ccc}
Fa & \xleftarrow{Ff} & Fx \\
\alpha_a \downarrow & & \downarrow \alpha_x \\
Ga & \xleftarrow{Gf} & Gx
\end{array}
$$

thus making the triangles (1) and (2) commute, and the square (3) commutes by naturality of α (with respect to g). This can be thought of as showing that the natural transformation $\mathrm{Span}(\alpha) : \mathrm{Span}(F) \to \mathrm{Span}(G)$ is oplax. Moreover, since α is supposed to be cartesian, the above naturality square is a pullback and thus h is an isomorphism. We can thus conclude that $\mathrm{Span}(\alpha)$ is a natural transformation. $\qquad\square$

Proposition 15. *The action on cartesian natural transformations $\alpha \mapsto \mathrm{Span}(\alpha)$ respects vertical and horizontal composition.*

Proof. First, observe that by the computation of proposition 13, the action of $\mathrm{Span}(-)$ obviously preserves the composition of functors.

First, for vertical composition of cartesian natural transformations : let $\mathcal{C}, \mathcal{D}$ be wild categories with pullbacks, $F, G, H : \mathcal{C} \to \mathcal{D}$ pullback-preserving functors, $\alpha : F \Rightarrow G$ and $\beta : G \Rightarrow H$ cartesian natural transformations. Given $A : \mathcal{C}$, we have $\mathrm{Span}(\beta \circ \alpha)_A = L(\beta_A \circ \alpha_A) = L(\beta_A) \circ L(\alpha_A) = \mathrm{Span}(\beta_A) \circ \mathrm{Span}(\alpha_A)$, so Span preserves vertical composition of cartesian natural transformations.

Now, for whiskering operations : let $\mathcal{C}, \mathcal{D}, \mathcal{E}$ be categories with pullbacks, $F, F' : \mathcal{C} \to \mathcal{D}$, $G, G' : \mathcal{D} \to \mathcal{E}$ be pullback-preserving functors, $\alpha : F \Rightarrow F'$ and $\beta : G \Rightarrow G'$ be cartesian natural transformations.

– First for left whiskering : write $G * \alpha : G \circ F \Rightarrow G \circ F'$ for the left whiskering of α by G, i.e. $(G * \alpha)_A := G(\alpha_A)$ for all $A : \mathcal{C}$. Since G preserves pullbacks, $G * \alpha$ is also a cartesian natural transformation. Moreover, for all $A : \mathcal{C}$, we have

$$
\mathrm{Span}(G * \alpha)_A = L(G(\alpha_A)) = G(L(\alpha_A)) = \mathrm{Span}(G)(\mathrm{Span}(\alpha)_A)
$$

so that Span preserves left whiskering.

– Then, for right whiskering : similarly, write $\beta * F : G \circ F \Rightarrow G' \circ F$ for the right whiskering of β by F, i.e. $(\beta * F)_A = \beta_{FA}$ for all $A : \mathcal{C}$. The naturality squares for $\beta * F$ are still naturality

9

squares for β, so they are cartesian regardless of the fact that F preserves pullbacks. Now, for all $A : \mathcal{C}$, we have

$$\text{Span}(\beta * F)_A = L(\beta_{FA}) = \text{Span}(\beta)_{FA} = (\text{Span}(\beta) * \text{Span}(F))_A$$

so that Span preserves right whiskering.

We just proved Span preserves vertical composition and left and right whiskering of cartesian natural transformations, so it also preserves horizontal composition. This concludes the proof. $\square$

Let $\mathcal{C}$ be a wild category with pullbacks and a terminal object $\mathbb{1}$. In particular, $\mathcal{C}$ admits finite products, since they are pullbacks over $\mathbb{1}$. Those finite products endow $\mathcal{C}$ with a symmetric monoidal structure.

Proposition 16. *The cartesian symmetric monoidal structure on $\mathcal{C}$ lifts to a symmetric monoidal structure on* $\text{Span}(\mathcal{C})$.

Proof. The cartesian product functor $- \times - : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ commutes with pullbacks since limits commute with limits. By proposition 13, it therefore lifts to a functor $\text{Span}(\mathcal{C} \times \mathcal{C}) \to \text{Span}(\mathcal{C})$. Moreover, there is a canonical equivalence of wild categories $\text{Span}(\mathcal{C} \times \mathcal{C}) \simeq \text{Span}(\mathcal{C}) \times \text{Span}(\mathcal{C})$. Composing $\text{Span}(- \times -)$ with this equivalence, we obtain a tensor product $- \otimes - : \text{Span}(\mathcal{C}) \times \text{Span}(\mathcal{C}) \to \text{Span}(\mathcal{C})$. The unit object of the monoidal structure is $\mathbb{1}$, seen as an object of $\text{Span}(\mathcal{C})$. The unitality, associativity, braiding, symmetry and all higher coherences for the tensor and unit are lifted from the coherences induced by the universal properties of the cartesian product in $\mathcal{C}$. $\square$

Remark 17. From the point of view of conjecture 12, we are using the following idea: a symmetric monoidal ∞-category is a commutative monoid object in the ∞-category of ∞-categories [22, Remark 2.4.2.6]. We expect that finite products on $\mathcal{C}$ should induce a structure of commutative monoid object on $\mathcal{C}$ in the ∞-category of ∞-categories with pullbacks and pullback-preserving functors. By conjecture 12, the ∞-functor Span should be cartesian, and thus preserves commutative monoid objects. Hence it should lift the cartesian symmetric monoidal structure on $\mathcal{C}$ to a symmetric monoidal structure on $\text{Span}(\mathcal{C})$. This idea tells us even more: any symmetric monoidal structure on $\mathcal{C}$ that is compatible with pullbacks should lift to a symmetric monoidal structure on $\text{Span}(\mathcal{C})$.

Proposition 18. *The symmetric monoidal wild category* $(\text{Span}(\mathcal{C}), \otimes, \mathbb{1})$ *is monoidal closed. For any objects A, B in $\mathcal{C}$, the internal hom in $\text{Span}(\mathcal{C})$ is given by $A \multimap B := A \otimes B$.*

Proof. Given objects A, B, C of $\mathcal{C}$, we have

$$
\begin{aligned}
\text{Span}_{\mathcal{C}}(A \otimes B, C) &\equiv \text{Span}_{\mathcal{C}}(A \times B, C) &&\text{(definition of } \otimes) \\
&\simeq \sum_{X \in \mathcal{C}} (X \to A \times B) \times (X \to C) &&\text{(definition of } \text{Span}_{\mathcal{C}}) \\
&\simeq \sum_{X \in \mathcal{C}} (X \to A \times B \times C) &&\text{(univ. prop. of } \times \text{ in } \mathcal{C}) \\
&\simeq \sum_{X \in \mathcal{C}} (X \to A) \times (X \to B \times C) &&\text{(univ. prop. of } \times \text{ in } \mathcal{C}) \\
&\simeq \text{Span}_{\mathcal{C}}(A, B \otimes C) &&\text{(definition of } \text{Span}_{\mathcal{C}})
\end{aligned}
$$

$\square$

Remark 19. The wild symmetric monoidal category of spans is actually compact closed, with each object being self-dual, which can be shown following the categorical proof [28]. As such it is monoidal closed with internal hom being given by $A \multimap B := A^* \otimes B \equiv A \otimes B$. Note that it also implies that it is $*$-autonomous.

3.2 Finite products in spans of types

We now focus our attention to the case $\mathcal{C} = \mathcal{U}$ and show that the category $\mathrm{Span}_\mathcal{U}$ has finite products.

Proposition 20. *The wild category* $\mathrm{Span}_\mathcal{U}$ *admits finite products, and they are computed as coproducts in* $\mathcal{C}$, *i.e. we have* $\pi_1 = R\iota_1 : A \sqcup B \twoheadrightarrow A$ *and* $\pi_2 = R\iota_2 : A \sqcup B \twoheadrightarrow B$.

Proof. We restrict our attention to binary products and the terminal object. Let A and B be types. We have

$$
\begin{aligned}
\mathrm{Span}_\mathcal{U}(X, A \sqcup B) &\simeq (X \times (A \sqcup B)) \to \mathcal{U} \qquad\qquad \text{(fibered/indexed equivalence)} \\
&\simeq ((X \times A) \sqcup (X \times B)) \to \mathcal{U} \\
&\simeq ((X \times A) \to \mathcal{U}) \times ((X \times B) \to \mathcal{U}) \\
&\simeq \mathrm{Span}_\mathcal{U}(X, A) \times \mathrm{Span}_\mathcal{U}(X, B),
\end{aligned}
$$

which proves that $A \sqcup B$ has the universal property of the cartesian product in $\mathrm{Span}_\mathcal{U}$.

For the terminal object, given a span $X \xleftarrow{s} Y \xrightarrow{t} \varnothing$, the map $t : Y \to \varnothing$ is necessarily unique and implies that $Y \simeq \varnothing$. Hence the data of such a span reduces to that of a map $\varnothing \to X$, of which there is only one. Hence $\varnothing$ is terminal in $\mathrm{Span}_\mathcal{U}$. $\qquad\square$

Remark 21. The proof of proposition 20 could be more generally performed for any category $\mathcal{C}$ which is lextensive, but our main focus will be on $\mathcal{C} := \mathcal{U}$ here.

Remark 22. By self-duality (see remark 11), proposition 20 implies that $\mathrm{Span}_\mathcal{U}$ also has coproducts.

We now have a symmetric monoidal closed category $\mathrm{Span}(\mathcal{U})$, which is furthermore cartesian (i.e. it admits finite products). To enhance it to a model of linear logic following definition 7, we need to equip it with a comonad whose underlying functor is symmetric monoidal from the cartesian structure to the monoidal structure.

4 The exponential comonad

Remember that our goal is to define a comonad on $\mathrm{Span}(\mathcal{U})$. By self duality (remark 11), this amounts to defining a monad on spans. We begin by defining a monad on $\mathrm{Type}_\mathcal{U}$, and then show that it lifts to spans.

4.1 The exponential as a monad on $\mathrm{Type}_\mathcal{U}$

In order to axiomatize the notion of "small type", we suppose given a type $\mathcal{V} : \mathcal{U}$, whose elements are codes for "small" types, together with an embedding (i.e. a fully faithful function) $\mathrm{El} : \mathcal{V} \to \mathcal{U}$ which to every code associates an actual type. In the following, for simplicity, given $A : \mathcal{V}$, we simply write $x : A$ instead of $x : \mathrm{El}\, A$ for an element x of a small type A. Because El is an embedding, the property for $A : \mathcal{U}$ of being equal to $\mathrm{El}\, A'$ for some $A' : \mathcal{V}$ is a proposition, and we say that A is $\mathcal{V}$-*small* when this is the case. We write $A \xrightarrow{\mathcal{V}} \mathcal{U}$ for the type of families

$F : A \to \mathcal{U}$ whose total space is $\mathcal{V}$-small. In the following, we always suppose that $\mathcal{V}$ is closed under dependent sums (i.e. for every $A : \mathcal{V}$ and $B : A \to \mathcal{V}$, we have $\Sigma AB : \mathcal{V}$), finite coproducts (in $\mathcal{U}$) and contains the terminal type. (Note: this axiomatization is close to the one of a *regular cardinal* in set theory).

Example 23. In a type theory with a hierarchy of cumulative universes $(\mathcal{U}_i)_{i \in \mathbb{N}}$, if $\mathcal{U} := \mathcal{U}_j$ for some j, we can take $\mathcal{V} := \mathcal{U}_i$ for any $i < j$ to be the universe of small types. In that case the map El is just the inclusion of types $\mathcal{U}_i \hookrightarrow \mathcal{U}_j$.

Example 24. Given a natural number $n : \mathbb{N}$, we write Fin_n for a type with n elements. We define the type of *finite types* as $\mathrm{Fin} := \sum_{A:\mathcal{U}} \sum_{n:\mathbb{N}} \|A = \mathrm{Fin}_n\|_{-1}$: an element of this type is a type together with the mere proof that it has a finite cardinal. We can then can take $\mathcal{V} := \mathrm{Fin}$ as universe of small types, with El being the first projection.

Definition 25. Given a type A, its *exponential* $!_{\mathcal{V}}A$ (read "bang A") relative to $\mathcal{V}$ is defined to be

$$!_{\mathcal{V}}A := \sum_{E:\mathcal{V}} (E \to A)$$

We often simply write $!A$, omitting $\mathcal{V}$ when it is obvious from the context. The exponential acts on morphisms by postcomposition: given a map $f : A \to B$, the induced map $!f : !A \to !B$ is defined by $!f(E, p) := (E, f \circ p)$. This makes $!$ into an endofunctor on $\mathrm{Type}_{\mathcal{U}}$.

Remark 26. Consider the case $\mathcal{V} \equiv \mathrm{Fin}$ of example 24. Then $!A$ is the homotopical analogue of the multiset construction. In ordinary set theory, multisets on a set A are usually defined as maps $A \to \mathbb{N}$ with finite support. In other words, a multiset on A is a finite collection of elements of A, with possible repetitions. Another way to think of a multiset X on A is simply as a finite set X whose elements are *colored* by elements of A. This intuition is implemented formally by alternatively defining a multiset on A to be the data of a finite set X together with a map $c : X \to A$. It is this latter intuition that we chose as 25 here.

Under the fibered/indexed equivalence (section 2.1), we could have also adapted the "map with finite support" point of view, defining a multiset on A to be a map $F : A \xrightarrow{\mathcal{V}} \mathcal{U}$, i.e. a map $F : A \to \mathcal{U}$ such that its total space $\sum_{x:A} F(x)$ is a $\mathcal{V}$-small type (notice how "finite support" has to be replaced with "finite total space" to make sense for an arbitrary base type A).

We say that $!A$ is a *homotopical* version of multisets in the sense that, contrary to its set-theoretic version, it remembers more symmetries. For instance, consider the set $A = \{a, b\}$, and the multiset $X = [a, a, b]$. Seen as an element of $!A$, X has two self-identifications: the identity, and the one that swaps the two elements *colored* by a. Hence the type $X =_{!A} X$ has two elements, and $!A$ is a groupoid that is not a set, even though A was.

Remark 27. Consider still $\mathcal{V} \equiv \mathrm{Fin}$. Given a set A, the set of multisets on A is the free commutative monoid on A. Similarly, given a type A seen as an ∞-groupoid, we expect that $!A$ is the free symmetric monoidal ∞-groupoid on A (and similarly we expect $!_{\mathcal{V}}A$ to be the free symmetric monoid with $\mathcal{V}$-small sums).

In order to explore the monad laws of $!$, we can first informally use the analogy with multisets. Writing $\mathrm{Mul}(A)$ for the multiset on a set A, we expect the unit $\eta_A : A \to \mathrm{Mul}(A)$ to be the map sending a to the multiset with a (once) as only element. Similarly, we expect the multiplication $\mu_A : \mathrm{Mul}(\mathrm{Mul}(A)) \to \mathrm{Mul}(A)$ to take a multiset of multisets to the multiset of its elements. This suggests defining the following natural transformations

$$\eta_A : A \to !A \qquad\qquad \mu_A : !!A \to !A$$
$$a \mapsto (\mathbb{1}, \mathrm{cst}_a) \qquad (E, p) \mapsto \left(\left(\sum_{e:E} \pi_1(p(e)) \right), ((e, e') \mapsto \pi_2(p(e))(e')) \right)$$

Note that the above definitions use the fact that $\mathcal{V}$ contains the terminal object and is closed under dependent sums.

Remark 28. By definition, we have $!!A \equiv \sum_{E:\mathcal{V}}(E \to \sum_{E':\mathcal{V}}(E' \to A))$, so an element in $!!A$ consists of a $\mathcal{V}$-small type E, and a family E' of $\mathcal{V}$-small types indexed by E, together with a map associating to each pair $(e,e') : \sum_{e:E}E'(e)$ a "color" in A. In other words, $!!A \simeq \sum_{E:\mathcal{V}}\sum_{E':E\to\mathcal{V}}(\Sigma EE' \to A)$. Under this equivalence, the multiplication map $\mu : !!A \to !A$ simply sends the triplet (E, E', p) to the total space $\Sigma EE'$ colored by p. Throughout the rest of this text, we will often abuse notation by writing $(E, E', p) : !!A$, implicitly using this equivalence.

Lemma 29. *The transformations η and μ defined above are natural.*

Proof. Let $f : A \to B$ be a map of types and consider the follow naturality squares:

$$
\begin{array}{ccc}
A & \xrightarrow{\eta_A} & !A \\
{\scriptstyle f}\downarrow & & \downarrow{\scriptstyle !f} \\
B & \xrightarrow{\eta_B} & !B
\end{array}
\qquad\qquad
\begin{array}{ccc}
!!A & \xrightarrow{\mu_A} & !A \\
{\scriptstyle !!f}\downarrow & & \downarrow{\scriptstyle !f} \\
!!B & \xrightarrow{\mu_B} & !B
\end{array}
$$

Given $a : A$, we have

$$(!f)(\eta_A(a)) = (!f)(\mathbb{1}, \mathrm{cst}_a) = (\mathbb{1}, \mathrm{cst}_{f(a)}) = \eta_B(f(a))$$

(and this equality is actually reflexivity).

Given $(E, E', p) : !!A$ (under the equivalence of remark 28), we have

$$(!f)(\mu_A(E, E', p)) = (!f)(\Sigma EE', p) = (\Sigma EE', f \circ p)$$

and

$$\mu_B((!!f)(E, E', p)) = \mu_B(E, E', f \circ p) = (\Sigma EE', f \circ p)$$

which completes the proof of naturality. $\qquad\square$

Proposition 30. *The triple $(!, \mu, \eta)$ is a monad on $\mathcal{U}$.*

Proof. Let $A : \mathcal{U}$. We need to prove the following square and triangles commute

$$
\begin{array}{ccc}
!!!A & \xrightarrow{!\mu_A} & !!A \\
{\scriptstyle \mu_{!A}}\downarrow & & \downarrow{\scriptstyle \mu_A} \\
!!A & \xrightarrow{\mu_A} & !A
\end{array}
\qquad\qquad
\begin{array}{ccccc}
!A & \xrightarrow{!\eta_A} & !!A & \xleftarrow{\eta_{!A}} & !A \\
 & {\scriptstyle \mathrm{id}_{!A}}\searrow & \downarrow{\scriptstyle \mu} & \swarrow{\scriptstyle \mathrm{id}_{!A}} & \\
 & & !A & &
\end{array}
$$

Let $(E, p) : !A$. We have

$$\mu_A(\eta_{!A}(E, p)) \simeq \mu_A(\mathbb{1}, \mathrm{cst}_E, (*, e) \mapsto p(e)) \simeq (E, p)$$

and

$$\mu_A(!\eta_A(E, p)) \simeq \mu_A(E, e \mapsto \mathbb{1}, (e, *) \mapsto p(e)) \simeq (E, p)$$

hence the triangles commute.

Now let $(E, E', E'', p) : !!!A$. Here we are using a characterization of $!!!A$ similar to that of $!!A$ in remark 28. In other words, we have $E : \mathcal{V}$, $E' : E \to \mathcal{V}$, $E'' : \Sigma EE' \to \mathcal{V}$ and $p : \Sigma E(\Sigma E'E'') \to A$. We have

$$\mu_A(!\mu_A(E, E', E'', p)) = \mu_A(E, \Sigma E'E'', p) = (\Sigma E(\Sigma E'E''), p)$$

and

$$\mu_A(\mu_{!A}(E, E', E'', p)) = \mu_A(\Sigma EE', E'', p) = \mu_A(\Sigma E(\Sigma E'E''), p)$$

hence the square commutes, which completes the proof. $\qquad\square$

13

We finally construct two isomorphisms in $\mathcal{U}$ that will become the Seely isomorphisms for $\mathrm{Span}(\mathcal{U})$ in the next section.

Proposition 31. *We have natural isomorphisms $l^2_{A,B} : {!A} \times {!B} \xrightarrow{\sim} {!}(A \sqcup B)$ for all types $A, B : \mathcal{U}$, and an isomorphism $l^0 : \mathbb{1} \xrightarrow{\sim} {!}\varnothing$. The former is defined by $l^2_{A,B}((E,p),(F,q)) := (E \sqcup F, p \sqcup q)$.*

Proof. For l^0, we have ${!}\varnothing \simeq (\varnothing \xrightarrow{\mathcal{V}} \mathcal{U}) \simeq \mathbb{1}$ by the universal property of the empty type and the fact that it is $\mathcal{V}$-small. For l^2, given types $A, B : \mathcal{U}$, we have

$$ {!A} \times {!B} \simeq (A \xrightarrow{\mathcal{V}} \mathcal{U}) \times (B \xrightarrow{\mathcal{V}} \mathcal{U}) \simeq (A \sqcup B \xrightarrow{\mathcal{V}} \mathcal{U}) \simeq {!}(A \sqcup B) $$

using remark 26 and closure of $\mathcal{V}$ under coproducts. Unfolding this equivalence, we have

$$ l^2_{A,B}((E,p)(F,q)) := (E \sqcup F, p \sqcup q). $$

Given maps $f : A \to C$ and $g : B \to D$, we need to show the following naturality square commutes.

$$
\begin{array}{ccc}
{!A} \times {!B} & \xrightarrow{l^2_{A,B}} & {!}(A \sqcup B) \\
{\scriptstyle {!f} \times {!g}}\downarrow & & \downarrow{\scriptstyle {!}(f \sqcup g)} \\
{!C} \times {!D} & \xrightarrow[l^2_{C,D}]{} & {!}(C \sqcup D)
\end{array}
$$

Let $((E,p),(F,q)) : {!A} \times {!B}$. We have

$$ ({!}(f \sqcup g))(l^2_{A,B}((E,p),(F,q))) = ({!}(f \sqcup g))(E \sqcup F, p \sqcup q) = (E \sqcup F, f \circ p \sqcup g \circ q) $$
$$ l^2_{C,D}(({!f} \times {!g})((E,p),(F,q))) = l^2_{C,D}((E, f \circ p),(F, g \circ q)) = (E \sqcup F, f \circ p \sqcup g \circ q) $$

which concludes the proof. □

4.2 Lifting the exponential monad to spans

We now show that the monad ! on $\mathcal{U}$ lifts to a monad on $\mathrm{Span}(\mathcal{U})$.

Proposition 32. *The functor ${!} : \mathcal{U} \to \mathcal{U}$ preserves pullbacks and therefore, by proposition 13, lifts to a functor $\mathrm{Span}({!}) : \mathrm{Span}_{\mathcal{U}} \to \mathrm{Span}_{\mathcal{U}}$.*

Proof. Using proposition 13, we only need to show that ${!} : \mathcal{U} \to \mathcal{U}$ preserves pullbacks. Let $A \xrightarrow{f} C \xleftarrow{g} B$ be a diagram of types. We have the following chain of equivalences:

$$
\begin{aligned}
{!A} \times_{{!C}} {!B} &\equiv \sum_{E:\mathcal{V}} \sum_{p:E\to A} \sum_{F:\mathcal{V}} \sum_{q:F\to B} ((E, f \circ p) = (F, g \circ q)) && \text{(def. of ! and pullback)} \\
&\simeq \sum_{E:\mathcal{V}} \sum_{p:E\to A} \sum_{F:\mathcal{V}} \sum_{q:F\to B} \Big(\sum_{l:E\simeq F} (f \circ p = g \circ q \circ l) \Big) && \text{(equality in Σ-types and univalence)} \\
&\simeq \sum_{E:\mathcal{V}} \sum_{F:\mathcal{V}} \sum_{l:E\simeq F} \sum_{p:E\to A} \sum_{q:F\to B} (f \circ p = g \circ q \circ l) && \text{(reordering of terms)} \\
&\simeq \sum_{E:\mathcal{V}} \sum_{p:E\to A} \sum_{q:E\to B} (f \circ p = g \circ q) && \text{(contracting (F, l) onto (E, id_E))} \\
&\simeq \sum_{E:\mathcal{V}} \sum_{p:E\to A} \sum_{q:E\to B} \Big(\prod_{e:E} (f(p(e)) = g(q(e))) \Big) && \text{(function extensionality)}
\end{aligned}
$$

14

$$\simeq \sum_{E:\mathcal{V}} (E \to A \times_C B) \qquad\qquad\qquad \text{(univ. prop. of the pullback)}$$

$$\equiv\ !(A \times_C B) \qquad\qquad\qquad\qquad\qquad \text{(def. of !)}$$

$\square$

In the following, to make notations more readable, we will still denote by ! its lifting Span(!).

Our aim is now to show that this functor inherits monad laws from the ones of the monad constructed on spans in section 4.1. Using propositions 14 and 15, we only need to show that the natural transformations η and μ are cartesian.

Proposition 33. *The natural transformation* $\eta : \mathrm{id}_{\mathcal{U}} \Rightarrow\ !$ *is cartesian.*

Proof. Let $f : A \to B$ be a map between types. Unwinding the corresponding naturality square of η, we get

$$
\begin{array}{ccc}
A & \xrightarrow{\quad f \quad} & B \\[2pt]
{\scriptstyle a \mapsto (\top,\, * \mapsto a)}\Big\downarrow & & \Big\downarrow {\scriptstyle b \mapsto (\top,\, * \mapsto b)} \\[4pt]
\sum_{E:\mathcal{V}}(E \to A) & \xrightarrow{\ (E,p)\mapsto(E,f\circ p)\ } & \sum_{E:\mathcal{V}}(E \to B)
\end{array}
$$

That square is definitionally commutative, i.e. the left-bottom and top-right compositions compute to the same term, or still in other terms, the witness of commutativity is $\mathrm{refl}_{\left(a \mapsto \left(\top,\, \mathrm{cst}_{f(a)}\right)\right)}$.

We have the following chain of equivalences:

$$!A \times_{!B} B \equiv \sum_{E:\mathcal{V}}\sum_{p:E\to A}\sum_{b:B}\big((E, f\circ p) = (\top, \mathrm{cst}_b)\big) \quad \text{(unfolding the definitions)}$$

$$\simeq \sum_{E:\mathcal{V}}\sum_{l:E\simeq\top}\sum_{p:E\to A}\sum_{b:B}(f\circ p = \mathrm{cst}_b \circ l) \quad \text{(equality in Σ-types, univalence, reordering)}$$

$$\simeq \sum_{p:\top\to A}\sum_{b:B}(f\circ p = \mathrm{cst}_b) \quad \text{(contracting (E,l) onto $(\top, \mathrm{id}_\top)$)}$$

$$\simeq \sum_{a:A}\sum_{b:B}(f(a) = b) \quad \text{(universal prop. of $\top$)}$$

$$\simeq A \quad \text{(contracting the last two terms onto $(f(a), \mathrm{refl})$)}$$

Moreover, computing this chain of equivalences from bottom to top exactly gives the connecting map $A \to\ !A \times_{!B} B$ of the pullback of the naturality square of η. So η is indeed cartesian. $\square$

Proposition 34. *The natural transformation* $\mu :\ !! \Rightarrow\ !$ *is cartesian.*

Proof.

$$!A \times_{!B}\ !!B \simeq \sum_{E:\mathcal{V}}\sum_{p:E\to A}\sum_{F:\mathcal{V}}\sum_{F':F\to\mathcal{V}}\sum_{q:\Sigma FF'\to B}(E, f\circ p) =_{!B} (\Sigma FF', q) \quad \text{(see remark 28 for $!!B$)}$$

$$\simeq \sum_{E:\mathcal{V}}\sum_{p:E\to A}\sum_{F:\mathcal{V}}\sum_{F':F\to\mathcal{V}}\sum_{q:\Sigma FF'\to B}\sum_{l:E\simeq\Sigma FF'} f\circ p = q\circ l \quad \text{(equality in Σ-types and univalence)}$$

$$\simeq \sum_{F:\mathcal{V}}\sum_{F':F\to\mathcal{V}}\sum_{p:\Sigma FF'\to A}\sum_{q:\Sigma FF'\to B} f\circ p = q \quad \text{(contracting (E,l) onto $(\Sigma FF', \mathrm{id})$)}$$

$$\simeq \sum_{F:\mathcal{V}}\sum_{F':F\to\mathcal{V}}(\Sigma FF' \to A) \quad \text{(contracting $(q,-)$ onto $(f\circ p, \mathrm{refl})$)}$$

$$\simeq\ !!A \quad \text{(remark 28)}$$

15

Like with η, one can check that the underlying map of the inverse of the equivalence thus constructed is equal to the map $!!A \to !A \times_{!B} !!B$ induced by the naturality square of μ. Hence μ is cartesian. □

Since η and μ are cartesian, they lift to natural transformations in spans, making ! into a monad on $\mathrm{Span}(\mathcal{U})$. By remark 11, they also make bang into a comonad on $\mathrm{Span}(\mathcal{U})$. We write ε and δ respectively for the counit and comultiplication of the comonad !. Unfolding the definition, we have

$$\varepsilon_A : !A \twoheadrightarrow A \qquad\qquad\qquad \delta_A : !A \twoheadrightarrow !!A$$

$$\varepsilon_A := R(\eta_A) = !A \xleftarrow{\eta_A} A \xrightarrow{\mathrm{id}_A} A \qquad\qquad \delta_A := R(\mu_A) = !A \xleftarrow{\mu_A} !!A \xrightarrow{\mathrm{id}_{!!A}} !!A$$

Similarly, l^2 being a natural isomorphism, its naturality squares are cartesian so that, under the equivalence $\mathrm{Span}(\mathcal{U} \times \mathcal{U}) \simeq \mathrm{Span}(\mathcal{U}) \times \mathrm{Span}(\mathcal{U})$ and by selfduality (remark 11), it lifts to a natural isomorphism $m^2_{A,B} := R(l^2_{A,B}) : !(A \sqcup B) \twoheadrightarrow !A \times !B$. The morphism l^0 also lifts to a morphism $m^0 := R(l^0) : !\varnothing \twoheadrightarrow \mathbb{1}$.

Theorem 35. *The symmetric monoidal closed, cartesian wild category* $(\mathrm{Span}(\mathcal{U}), \otimes, \mathbb{1}, \sqcup, \varnothing)$, *equipped with the comonad* $(!, \delta, \varepsilon)$ *and the morphisms* m^2, m^0, *is a wild Seely category.*

Proof. We need to check that $(!, m^2, m^0) : (\mathrm{Span}(\mathcal{U}), \sqcup, \varnothing) \to (\mathrm{Span}(\mathcal{U}), \otimes, \mathbb{1})$ is symmetric monoidal, and that for all $A, B : \mathcal{U}$ the following diagram commutes

$$
\begin{array}{ccccc}
!(A \sqcup B) & \xrightarrow{\delta_{A \sqcup B}} & !!(A \sqcup B) & \xrightarrow{!(!\pi_1, !\pi_2)} & !(!A \sqcup !B) \\
\Big\downarrow{m^2_{A,B}} & & & & \Big\downarrow{m^2_{!A,!B}} \\
!A \times !B & & \xrightarrow{\delta_A \times \delta_B} & & !!A \times !!B
\end{array}
$$

But this diagram is precisely the image by the functor $R : \mathcal{U}^{\mathrm{op}} \to \mathrm{Span}(\mathcal{U})$ of the diagram

$$
\begin{array}{ccccc}
!(A \sqcup B) & \xleftarrow{\mu_{A \sqcup B}} & !!(A \sqcup B) & \xleftarrow{!(!\iota_1, !\iota_2)} & !(!A \sqcup !B) \\
\Big\uparrow{l^2_{A,B}} & & & & \Big\uparrow{l^2_{!A,!B}} \\
!A \times !B & & \xleftarrow{\mu_A \times \mu_B} & & !!A \times !!B
\end{array}
$$

Let $\mathbb{E} := (E, E', p) : !!A$ and $\mathbb{F} := (F, F', q) : !!B$ (under the equivalence of remark 28). We have

$$l^2_{A,B}((\mu_A \times \mu_B)(\mathbb{E}, \mathbb{F})) = l^2_{A,B}((\Sigma EE', p), (\Sigma FF', q)) = (\Sigma EE' \sqcup \Sigma FF', p \sqcup q)$$

and

$$!!A \times !!B \xrightarrow{l^2_{!A,!B}} !(!A \sqcup !B) \xrightarrow{!\langle !\iota_1, !\iota_2 \rangle} !!(A \sqcup B) \xrightarrow{\mu_{A \sqcup B}} !(A \sqcup B)$$

$$(\mathbb{E}, \mathbb{F}) \longmapsto (E \sqcup F, (E', p) \sqcup (F', q)) \longmapsto (E \sqcup F, \langle E', F' \rangle, p \sqcup q) \longmapsto \left(\textstyle\sum_{x : E \sqcup F} \sum_{y : \langle E', F' \rangle(x)} (p \sqcup q)(x, y) \right)$$

Those two elements of $!(A \sqcup B)$ are equal by virtue of Σ-types distributing over disjoint sums.

What remains to be shown is that the following data constitutes a symmetric monoidal functor:

$$(!, m^2, m^0) : (\mathrm{Span}(\mathcal{U}), \sqcup, \varnothing) \to (\mathrm{Span}(\mathcal{U}), \times, \mathbb{1}).$$

Since the functor ! on spans is obtained as a lifting of $! : \mathcal{U} \to \mathcal{U}$, and similarly for the natural transformations m^2 and m^0, using proposition 15 we only need to show that

$$(!, l^2, l^0) : (\mathcal{U}, \sqcup, \varnothing) \to (\mathcal{U}, \times, \mathbb{1})$$

is a symmetric monoidal functor. Following [26, Section 7.3] (and writing respectively $\alpha, \lambda, \rho, \gamma$ for the obvious associativity, left-unitality, right-unitality, and symmetry isomorphisms), this amounts to showing the following four diagrams commute.

$$
\begin{array}{ccc}
(!A \times !B) \times !C & \xrightarrow{\alpha_\times} & !A \times (!B \times !C) \\
{\scriptstyle l^2_{A,B} \times \mathrm{id}_{!C}} \downarrow & & \downarrow {\scriptstyle \mathrm{id}_{!A} \times l^2_{B,C}} \\
!(A \sqcup B) \times !C & \quad (1) \quad & !A \times !(B \sqcup C) \\
{\scriptstyle l^2_{A \sqcup B, C}} \downarrow & & \downarrow {\scriptstyle l^2_{A, B \sqcup C}} \\
!((A \sqcup B) \sqcup C) & \xrightarrow[!\alpha_\sqcup]{} & !(A \sqcup (B \sqcup C))
\end{array}
\qquad
\begin{array}{ccc}
!A \times \mathbb{1} & \xrightarrow{\rho_\times} & !A \\
{\scriptstyle \mathrm{id}_{!A} \times l^0} \downarrow & (2) & \uparrow {\scriptstyle !\rho_\sqcup} \\
!A \times !\varnothing & \xrightarrow[l^2_{A, \varnothing}]{} & !(A \sqcup \varnothing)
\end{array}
$$

$$
\begin{array}{ccc}
\mathbb{1} \times !B & \xrightarrow{\lambda_\times} & !B \\
{\scriptstyle l^0 \times \mathrm{id}_{!B}} \downarrow & (3) & \uparrow {\scriptstyle !\lambda_\sqcup} \\
!\varnothing \times !B & \longrightarrow & !(\varnothing \sqcup B)
\end{array}
$$

$$
\begin{array}{ccc}
!A \times !B & \xrightarrow{\gamma_\times} & !B \times !A \\
{\scriptstyle l^2_{A,B}} \downarrow & (4) & \downarrow {\scriptstyle l^2_{B,A}} \\
!(A \sqcup B) & \xrightarrow[!\gamma_\sqcup]{} & !(B \sqcup A)
\end{array}
$$

Most verifications are straightforward:

– for diagram (1), given $(((E,p),(F,q)),(G,r)) : (!A \times !B) \times !C$, both paths in the hexagon evaluate to $(E \sqcup (F \sqcup G), p \sqcup (q \sqcup r)) : !(A \sqcup (B \sqcup C))$ up to associativity of $\sqcup$, which is an equivalence and hence an equality by univalence (see the case of (4) for more a detailed reasoning along those lines),

– diagrams (2) and (3) are trivial,

– and finally for diagram (4), given $((E,p),(F,q)) : !A \times !B$, we have

$$l^2_{B,A}(\gamma_\times((E,p),(F,q))) = l^2_{B,A}((F,q),(E,p)) = (F \sqcup E, q \sqcup p)$$

and

$$!\gamma_\sqcup(l^2_{A,B}((E,p),(F,q))) = !\gamma_\sqcup(E \sqcup F, p \sqcup q) = (E \sqcup F, \gamma_\sqcup \circ (p \sqcup q))$$

By characterization of equality in Σ-types and univalence, an equality between $(E \sqcup F, \gamma_\sqcup \circ (p \sqcup q))$ and $(F \sqcup E, q \sqcup p)$ in $!(B \sqcup A)$ consists of the data of an equivalence $f : E \sqcup F \xrightarrow{\cong} F \sqcup E$ and an equality $e : \gamma_\sqcup \circ (p \sqcup q) = (q \sqcup p) \circ f$ in $E \sqcup F \to B \sqcup A$. Choosing $f := \gamma_\sqcup$, a simple computation shows the following square commutes

$$
\begin{array}{ccc}
E \sqcup F & \xrightarrow{p \sqcup q} & A \sqcup B \\
{\scriptstyle \gamma_\sqcup} \downarrow & & \downarrow {\scriptstyle \gamma_\sqcup} \\
F \sqcup E & \xrightarrow[q \sqcup p]{} & B \sqcup A
\end{array}
$$

which concludes the proof.

$\square$

The above theorem can be interpreted as the fact that $\mathrm{Span}(\mathcal{U})$ is a "wild model" of intuitionistic linear logic. Since it is $*$-autonomous (remark 19), it also extends to a model of classical linear logic and the presence of products (proposition 20) and coproducts (remark 22) allow for modeling additives.

It is not difficult to show that the categorical structures are preserved by truncation (definition 5), and preservation of univalence was shown in proposition 6. We thus have:

Theorem 36. *The category* $\|\mathrm{Span}(\mathcal{U})\|_1$ *is a Seely category and thus a model of linear logic.*

Note that the model we obtain, which is apparently new, is of a very different nature than the usual ones for linear logic (relational model, coherence spaces, etc.). Namely, the category $\|\mathrm{Type}_{\mathcal{U}}\|_1$ of types and homotopy classes of functions embeds, via the functor induced by L, into $\|\mathrm{Span}(\mathcal{U})\|_1$. This former category can be interpreted as the homotopy category of spaces, and it is known that it is not *concrete* [10], i.e. cannot be recovered as a subcategory of Set, thus neither can be our model. We leave for future work the investigation of possible more direct descriptions of this model.

5 The exponential modality and polynomials

We show here that the wild Kleisli category associated to our comonad on $\mathrm{Span}(\mathcal{U})$ is the well-known category of polynomials (up to homotopy).

Definition 37. Let I, J be types. A polynomial from I to J is a diagram of the shape

$$I \xleftarrow{\;\;s\;\;} E \xrightarrow{\;\;p\;\;} B \xrightarrow{\;\;t\;\;} J \tag{2}$$

In other words, the type of polynomials with source I and target J is

$$\mathrm{Poly}(I, J) := \sum_{E:\mathcal{U}} \sum_{B:\mathcal{U}} (E \to I) \times (E \to B) \times (B \to J)$$

To make sense of this definition, one must understand what the data of a polynomial represents, namely, a *polynomial functor*.

5.1 The functor associated to a polynomial

In the following, given a map $f : A \to B$ and $b : B$, we write A_b for $\mathrm{fib}_f(b)$, the function being often implicit from the context. A polynomial as in definition 37 can be thought of as the data describing a (colored) polynomial functor in the following sense. Given $j : J$, B_j is the type of monomials colored by j; given $b : B_j$, the arity of the monomial b is E_b; finally, the function $s : E_b \to I$ associates, to each variable $e : E_b$ a color in I. This interpretation suggests associating to any polynomial a function, following the now classical definition of a polynomial functor [11].

Definition 38. Any polynomial $P = (E, B, s, t, p)$ as in (2) induces a map $F_P : \mathcal{U}^I \to \mathcal{U}^J$, defined by

$$F(X)(j) := \sum_{b:B_j} \prod_{e \in E_b} X(s(e))$$

called the *polynomial functor* induced by P.

Above, we abusively use the term "functor" following the traditional terminology which comes from the fact that, in category theory, a polynomial in a locally cartesian closed category $\mathcal{C}$ between objects x and y induces a functor $\mathcal{C}_{/x} \to \mathcal{C}_{/y}$. Here, we have $\mathcal{C} := \mathcal{U}$, and under the fibered/indexed equivalence the slice category $\mathcal{U}_{/X}$ is equivalent to $\mathcal{U}^X$.

Definition 39. A polynomial $P = (E, B, s, p, t)$ is said to be *linear* if the map p is an equivalence. This is precisely equivalent to asking the fibers $E_b := \mathrm{fib}_p(b)$ to be contractible for all $b : B$, in which case the products in the expression of F_P are indexed over singleton types, hence the name linear.

Remark 40. Any span $A \xleftarrow{f} X \xrightarrow{g} B$ induces a linear polynomial $(X, X, f, \mathrm{id}_X, g)$. Moreover this map is an equivalence: every linear polynomial is equivalent (and thus, by univalence, equal) to one of this form.

We can generalize definition 39 by weakening the requirements on the fibers of p, i.e. by selecting which *arities* are allowed for the products appearing the associated polynomial functor.

Definition 41. A polynomial (E, B, s, p, t) is said to be $\mathcal{V}$-ary if the fibers of the map p are $\mathcal{V}$-small. We write $\mathrm{Poly}_{\mathcal{V}}(I, J)$ for the type of $\mathcal{V}$-ary polynomials between I and J, in other words:
$$\mathrm{Poly}_{\mathcal{V}}(I, J) := \sum_{E:\mathcal{U}} \sum_{B:\mathcal{U}} (E \to I) \times (E \to_{\mathcal{V}} B) \times (B \to J)$$

(where $E \to_{\mathcal{V}} B$ denotes the type of maps $f : E \to B$ whose fibers are $\mathcal{V}$-small).

Example 42. Here are some examples of universes and the corresponding notions of polynomials:

– if $\mathcal{V}$ is the universe of contractible types, the $\mathcal{V}$-ary polynomials are the linear ones, hence the spans by remark 40,

– if $\mathcal{V}$ is the universe of propositions (i.e. subsingletons), the $\mathcal{V}$-ary polynomials could be called *affine polynomials*,

– if $\mathcal{V} \equiv \mathrm{Fin}$ is the universe of finite types, then we talk about finitary polynomial functors (those correspond to the ones of [8], which are further detailed in section 6.1).

Note that those notions make sense even for choices of $\mathcal{V}$ that do not satisfy all the axioms asked at the beginning of section 4.

Proposition 43. *There is a wild category* $\mathrm{Poly}_{\mathcal{V}}$ *whose objects are types in* $\mathcal{U}$ *and morphisms are* $\mathcal{V}$-*ary polynomials. Identities are given by identity spans seen as polynomials:* $\mathrm{Id}_A := A \xleftarrow{\mathrm{id}_A} A \xrightarrow{\mathrm{id}_A} A \xrightarrow{\mathrm{id}_A} A$. *The composition of two polynomials* $P = (I \xleftarrow{s} E \xrightarrow{p} B \xrightarrow{t} J)$ *and* $Q = (J \xleftarrow{u} F \xrightarrow{q} C \xrightarrow{v} K)$ *is given by*

$$I \xleftarrow{s \circ \pi_3} \sum_{(c,\alpha):D} \sum_{x:F_c} E_{\alpha(x)} \xrightarrow{\pi_1} D \xrightarrow{v \circ \pi_1} K$$

where $D := \sum_{c:C} \prod_{x:F_c} B_{u(x)}$.

Proof. We do not prove that composition is unital and associative since it will follow from theorem 45. The fact that $\mathrm{Poly}_{\mathcal{V}}$ is univalent was claimed in [8] without proof. Here is a sketch of proof: suppose that a polynomial $P = (I \xleftarrow{s} E \xrightarrow{p} B \xrightarrow{t} J)$ has a left and right inverse, we can prove that p has to be an isomorphism in $\mathcal{V}$, so that P is actually a span, and similarly for its two-sided inverses. Then univalence of $\mathrm{Poly}_{\mathcal{V}}$ follows from proposition 9. $\square$

5.2 Polynomials are Kleisli morphisms

Proposition 44. *Let I, J be types. There is an equivalence*

$$\text{poly-to-span} : \text{Poly}_{\mathcal{V}}(I, J) \simeq \text{Span}(!_{\mathcal{V}} I, J)$$

which maps a polynomial $I \xleftarrow{s} E \xrightarrow{p}_{\mathcal{V}} B \xrightarrow{t} J$ to $!_{\mathcal{V}} I \xleftarrow{\bar{s}} B \xrightarrow{t} J$ where $\bar{s}(b) := \big(E_b, s|_{E_b}\big)$ (where $s|_{E_b}$ is the restriction of s to E_b).

Proof. Fix $B : \mathcal{U}$.

$$\sum_{E:\mathcal{U}} (E \to I) \times (E \to_{\mathcal{V}} B) \simeq \sum_{F:B\to\mathcal{V}} (\Sigma BF \to I) \simeq B \to \sum_{F:\mathcal{V}} (F \to I) \equiv (B \to !_{\mathcal{V}} I) \qquad (3)$$

Thus, we have the following chain of equivalences:

$$\begin{aligned}
\text{Poly}_{\mathcal{V}}(I, J) &\equiv \sum_{E:\mathcal{U}} \sum_{B:\mathcal{U}} (E \to I) \times (E \to_{\mathcal{V}} B) \times (B \to J) \\
&\simeq \sum_{B:\mathcal{U}} \Big(\sum_{E:\mathcal{U}} (E \to I) \times (E \to_{\mathcal{V}} B) \Big) \times (B \to J) && \text{(reordering of terms)} \\
&\simeq \sum_{B:\mathcal{U}} (B \to !_{\mathcal{V}} I) \times (B \to J) && \text{(by (3))} \\
&\equiv \text{Span}(!_{\mathcal{V}} I, J)
\end{aligned}$$

which concludes the proof. □

Now, the interesting point is that this equivalence respects composition and identities: as we will show, there is an equivalence of categories between Poly and the Kleisli category for the comonad ! on Span. The intuition one should have here is that spans correspond to linear polynomials and ! allows for using variables many times, we thus expect a linear map which is allowed to use its arguments multiple times to be the same as a polynomial.

Theorem 45. *The Kleisli wild category $\text{Span}(\mathcal{U})_{!_{\mathcal{V}}}$ associated to the comonad $!_{\mathcal{V}}$ on $\text{Span}(\mathcal{U})$ is equivalent to the wild category $\text{Poly}_{\mathcal{V}}$ of polynomials.*

Proof. The wild categories $\text{Span}(\mathcal{U})_{!_{\mathcal{V}}}$ and $\text{Poly}_{\mathcal{V}}$ have the same type of objects, namely $\mathcal{U}$, so we can take the identity as the mapping on objects $\text{Poly}_{\mathcal{V}} \to \text{Span}(\mathcal{U})_{!_{\mathcal{V}}}$. Since it is an equivalence on objects, the functor we are constructing is in particular essentially surjective. The action on morphisms is given by the equivalence of proposition 44 (thus our functor will be fully faithful). Remains to be shown that this mapping is compatible with identities and composition.

Fix a type A. By proposition 44, the identity polynomial on A is mapped to the span $!_{\mathcal{V}} A \xleftarrow{\overline{\text{id}_A}} A \xrightarrow{\text{id}_A} A$, with $\overline{\text{id}_A}(a) = (\sum_{a':A}(a' = a), (a', p) \mapsto a')$. But the type $\sum_{a':A}(a' = a)$ is contractible with center (a, refl_a) [31, Lemma 3.11.8], so $\overline{\text{id}_A}(a) = (\mathbb{1}, \text{cst}_a) = \eta_A(a)$. So in the end poly-to-span$(\text{Id}_A) = \varepsilon_A$, which is the identity of A in $\text{Span}(\mathcal{U})_{!_{\mathcal{V}}}$.

Now for composition. Let $P = (I \xleftarrow{s} E \xrightarrow{p} B \xrightarrow{t} J)$ and $Q = (J \xleftarrow{u} F \xrightarrow{q} C \xrightarrow{v} K)$ be polynomials in $\text{Poly}_{\mathcal{V}}$. Writing $D := \sum_{c:C} \prod_{x:F_c} B_{u(x)}$ as in proposition 43, the span poly-to-span$(Q \circ P)$ is given by $!I \xleftarrow{f} D \xrightarrow{g} K$, with

$$f(c, \alpha) := \Big(\sum_{x:F_c} E_{\alpha(x)}, (x, e) \mapsto s(e) \Big) \qquad \text{and} \qquad g(c, \alpha) := v(c).$$

On the other hand, composition in $\mathrm{Span}(\mathcal{U})_{!\mathcal{V}}$ is given by the following composition of spans
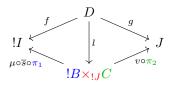
$$
\begin{array}{ccccccc}
 & & !!I & & & & !B & & & & C \\
 & \swarrow_{\mu} & & \searrow^{\mathrm{id}_{!!I}} & & \swarrow_{!\overline{s}} & & \searrow^{!t} & & \swarrow_{\overline{u}} & & \searrow^{v} \\
!I & & & & !!I & & & & !J & & & & K
\end{array}
$$

Pulling back along $\mathrm{id}_{!!I}$ does not change the morphism, so that this is the same as the composition of the spans

$$
\begin{array}{ccccccc}
 & & !B & & & & C \\
 & \swarrow_{\mu \circ !\overline{s}} & & \searrow^{!t} & & \swarrow_{\overline{u}} & & \searrow^{v} \\
!I & & & & !J & & & & K
\end{array}
$$

This composition is obtained by computing the following pullback:

$$
!B \times_{!J} C \equiv \sum_{X:\mathcal{V}} \sum_{\pi:X\to B} \sum_{c:C} (X, t \circ \pi) =_{!J} \left(F_c, u|_{F_c}\right)
$$

$$
\simeq \sum_{c:C} \sum_{X:\mathcal{V}} \sum_{k:X\simeq F_c} \sum_{\pi:X\to B} t \circ \pi = u|_{F_c} \circ k \qquad \text{(equality in } \Sigma\text{-types and univalence)}
$$

$$
\simeq \sum_{c:C} \sum_{\pi:F_c\to B} t \circ \pi = u|_{F_c} \qquad \text{(contracting } (X, k) \text{ onto } (F_c, \mathrm{id}))
$$

$$
\simeq \sum_{c:C} \sum_{\pi:F_c\to B} \prod_{x:F_c} t(\pi(x)) = u(x) \qquad \text{(function extensionality)}
$$

$$
\simeq \sum_{c:C} \prod_{x:F_c} \sum_{b:B} t(b) = u(x) \qquad \text{(swapping } \Pi \text{ and } \Sigma)
$$

$$
\equiv \sum_{c:C} \prod_{x:F_c} B_{u(x)} \qquad \text{(def. of } B_{u(x)} \equiv \mathrm{fib}_t u(x))
$$

$$
\equiv D
$$

The inverse map $l : D \to\ !B \times_{!J} C$ maps the pair (c, α) to the tuple $((F_c, \pi_1 \circ \alpha), c, (\mathrm{refl}_{F_c}, \pi_2 \circ \alpha))$. What remains to show is that the two following triangles commute.

$$
\begin{array}{ccccc}
 & & D & & \\
 & \swarrow^{f} & \downarrow^{l} & \searrow^{g} & \\
!I & & & & J \\
 & \underset{\mu \circ \overline{s} \circ \pi_1}{\longleftarrow} & !B \times_{!J} C & \underset{v \circ \pi_2}{\longrightarrow} &
\end{array}
$$

Let $(c, \alpha) : D$. We have

$$
\mu(!\overline{s}(\pi_1(l(\alpha,c)))) = \mu(!\overline{s}(F_c, \alpha)) = \mu(F_c, \overline{s} \circ \alpha) = \mu\left(F_c, x \mapsto \left(E_{\alpha(x), s|_{E_{\alpha(x)}}}\right)\right) = \left(\sum_{x:F_c} E_{\alpha(x)}, (x, e) \mapsto s(e)\right)
$$

$$
= f(c, \alpha)
$$

and $v(\pi_2(l(c,\alpha))) = v(c) = g(c, \alpha)$. This completes the proof. $\qquad\square$
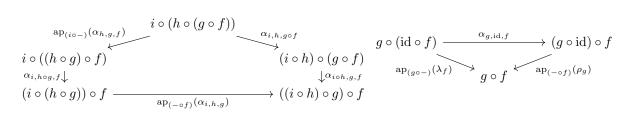
We would like to point out that the observation that polynomial functors could be described as a Kleisli category was already made by Street [29, Example 13] (although ignoring size issues).

6 Related works

6.1 Recovering the bicategory of polynomials

Let us explain here the relationship between the work in this article and the construction of the (wild) bicategory of spans performed in [8]. First, as indicated in section 2.2, the notion of wild category is "right" only when the hom-types are sets (definition 4). In the general case, one needs to add further additional axioms in order to ensure the coherence of the structure (in fact, this coherence is itself part of the structure). The "next layer" of coherences can be expressed as follows.

Definition 46. A 2-*coherence* for a wild precategory $\mathcal{C}$ consists of witnesses for the commutation of the following diagrams for any suitably composable morphisms f, g, h, i:

$$
\begin{array}{ccc}
& i \circ (h \circ (g \circ f)) & \\
\overset{\mathrm{ap}_{(i \circ -)}(\alpha_{h,g,f})}{\swarrow} & & \overset{\alpha_{i,h,g \circ f}}{\searrow} \\
i \circ ((h \circ g) \circ f) & & (i \circ h) \circ (g \circ f) \\
\alpha_{i,h \circ g,f} \downarrow & & \downarrow \alpha_{i \circ h,g,f} \\
(i \circ (h \circ g)) \circ f & \xrightarrow{\mathrm{ap}_{(- \circ f)}(\alpha_{i,h,g})} & ((i \circ h) \circ g) \circ f
\end{array}
\qquad
\begin{array}{ccc}
g \circ (\mathrm{id} \circ f) & \xrightarrow{\alpha_{g,\mathrm{id},f}} & (g \circ \mathrm{id}) \circ f \\
\mathrm{ap}_{(g \circ -)}(\lambda_f) \searrow & & \swarrow \mathrm{ap}_{(- \circ f)}(\rho_g) \\
& g \circ f &
\end{array}
$$

A *(pre)bicategory* is a wild (pre)category equipped with a 2-coherence such that $\mathcal{C}(x,y)$ is a groupoid for every objects $x, y : \mathcal{C}$.

It can be shown that the wild category $\mathrm{Span}(\mathcal{U})$ can be equipped with a 2-coherence. We expect that the monad defined in section 4 can also be shown to be coherent (this is left for future work), thus inducing a 2-coherence on the wild Kleisli category $\mathrm{Poly}(\mathcal{U})$. Moreover, this construction can be refined (in the same spirit as section 4.1) in order to replace $\mathcal{U}$ by a subuniverse of types satisfying suitable closure properties. In particular, this would allow defining $\mathrm{Poly}(\mathrm{Gpd})$, thus recovering the construction of the wild bicategory of polynomials defined in [8]. It should be noted that, contrarily to what is claimed, [8] only actually constructs a wild 2-coherent category: it is 3-truncated (and not 2-truncated as expected for a bicategory). However, this can be fixed by truncating the wild 2-coherent category into a bicategory, using a similar process as for categories (see definition 5 and proposition 6).

6.2 Towards a model of differential linear logic

In [25], Melliès defines a bicategorical model of differential linear logic based on a comonad on spans (we consider here only the particular case where the "synchronization template" category is the terminal one). When we restrict his model to groupoids (as opposed to categories), we obtain a model which is equivalent to the one of the previous section with $\mathcal{U} := \mathrm{Gpd}$ and $\mathcal{V} := \mathrm{Fin}$. We can thus conclude that the morphisms in his Kleisli category are the polynomials, this observation being new to our knowledge. This also suggests that we could extend our model into one of differential linear logic. This shall be the topic of future work.

6.3 Related models

The model we have defined is close to the one of species of structures [9] which are a categorical generalization of Joyal's combinatorial species [17]. We plan to investigate the relationship between the two in future work. In particular, the comparison should be made easier by the work of Gepner, Haugseng and Kock [12, Section 3.2] who showed that analytic and polynomial

functors coincide in the setting of ∞-categories. It would also be interesting to compare the present work with Clairambault and Forest model of thin spans of groupoids [7], which is close in spirit but technically quite different since it uses an orthogonality construction in order to avoid taking symmetries in account.

Some other authors have considered categories whose objects (as opposed to morphisms) are polynomials. Such categories have also been shown to be cartesian closed [2], to define models of linear logic [16], and to be the Kleisli category of some other category [19, Theorem 2.4.5]. Although those results seem reminiscent of the ones exposed here, it is unclear if they can be related in any way, since there is in general no connection between a bicategory and the category of endomorphisms of one of its objects.

References

[1] Thorsten Altenkirch, Neil Ghani, Peter Hancock, Conor McBride, and Peter Morris. Indexed containers. *Journal of Functional Programming*, 25, 2015. `doi:10.1017/S095679681500009X`.

[2] Thorsten Altenkirch, Paul Levy, and Sam Staton. Higher-order containers. In *Programs, Proofs, Processes: 6th Conference on Computability in Europe, CiE 2010, Ponta Delgada, Azores, Portugal, June 30–July 4, 2010. Proceedings 6*, pages 11–20. Springer, 2010. `doi:10.1007/978-3-642-13962-8_2`.

[3] P Nick Benton. A mixed linear and non-linear logic: Proofs, terms and models. In *International Workshop on Computer Science Logic*, pages 121–135. Springer, 1994. `doi:10.1007/BFb0022251`.

[4] Gavin M Bierman. What is a categorical model of intuitionistic linear logic? In *Typed Lambda Calculi and Applications: Second International Conference on Typed Lambda Calculi and Applications, TLCA'95 Edinburgh, United Kingdom, April 10–12, 1995 Proceedings 2*, pages 78–93. Springer, 1995. `doi:10.1007/BFb0014046`.

[5] Ulrik Buchholtz. Update on semisimplicial types in homotopy type theory. Workshop on Logic and higher structures,. URL: `https://www.cirm-math.fr/RepOrga/2689/Slides/s_buchholtz.pdf`.

[6] Paolo Capriotti and Nicolai Kraus. Univalent higher categories via complete semi-segal types. `doi:10.1145/3158132`.

[7] Pierre Clairambault and Simon Forest. The cartesian closed bicategory of thin spans of groupoids. In *2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13. IEEE, 2023.

[8] Eric Finster, Samuel Mimram, Maxime Lucas, and Thomas Seiller. A cartesian bicategory of polynomial functors in homotopy type theory. In *37th Conference on the Mathematical Foundations of Programming Semantics (MFPS 2021)*, 2021. `doi:10.4204/EPTCS.351.5`.

[9] Marcelo Fiore, Nicola Gambino, Martin Hyland, and Glynn Winskel. The cartesian closed bicategory of generalised species of structures. *Journal of the London Mathematical Society*, 77(1):203–220, 2008. `doi:10.1112/jlms/jdm096`.

[10] Peter Freyd. Homotopy is not concrete. In *The Steenrod Algebra and Its Applications: A Conference to Celebrate NE Steenrod's Sixtieth Birthday: Proceedings of the Conference held at the Battelle Memorial Institute, Columbus, Ohio March 30th–April 4th, 1970*, pages 25–34. Springer, 1970. `doi:10.1007/BFb0058516`.

[11] Nicola Gambino and Joachim Kock. Polynomial functors and polynomial monads. In *Mathematical proceedings of the cambridge philosophical society*, volume 154, pages 153–192. Cambridge University Press, 2013. `doi:10.1017/S0305004112000394`.

[12] David Gepner, Rune Haugseng, and Joachim Kock. ∞-operads as analytic monads. *International Mathematics Research Notices*, 2022(16):12516–12624, 2022. `doi:10.1093/imrn/rnaa332`.

[13] Jean-Yves Girard. Normal functors, power series and λ-calculus. *Annals of pure and applied logic*, 37(2):129–177, 1988. `doi:10.1016/0168-0072(88)90025-5`.

[14] Robert Gordon, Anthony John Power, and Ross Street. *Coherence for tricategories*, volume 558. American Mathematical Society, 1995. `doi:10.1090/memo/0558`.

[15] Mark Hovey. *Model categories*. Number 63. American Mathematical Society, 2007. `doi:10.1090/surv/063`.

[16] Pierre Hyvernat. A Linear Category of Polynomial Functors (extensional part). *Logical Methods in Computer Science*, 10(2), May 2014. URL: `https://lmcs.episciences.org/1185`, `doi:10.2168/LMCS-10(2:2)2014`.

[17] André Joyal. Une théorie combinatoire des séries formelles. *Advances in mathematics*, 42(1):1–82, 1981.

[18] Krzysztof Kapulkin and Peter LeFanu Lumsdaine. The simplicial model of univalent foundations (after voevodsky). *Journal of the European Mathematical Society*, 23(6):2071–2126, 2021. `doi:10.4171/JEMS/1050`.

[19] Joachim Kock. Polynomial Functors and Trees. *International Mathematics Research Notices*, 2011(3):609–673, 05 2010. `doi:10.1093/imrn/rnq068`.

[20] Joachim Kock. Data types with symmetries and polynomial functors over groupoids. *Electronic Notes in Theoretical Computer Science*, 286:351–365, 2012. `doi:10.1016/j.entcs.2013.01.001`.

[21] Peter LeFanu Lumsdaine. Weak ω-categories from intensional type theory. *Logical Methods in Computer Science*, 6, 2010. `doi:10.2168/LMCS-6(3:24)2010`.

[22] Jacob Lurie. *Higher Algebra*. URL: `https://people.math.harvard.edu/~lurie/papers/HA.pdf`.

[23] Jacob Lurie. *Higher topos theory*. Princeton University Press, 2009. `doi:10.1515/9781400830558`.

[24] Saunders Mac Lane. *Categories for the working mathematician*, volume 5. Springer Science & Business Media, 2013. `doi:10.1007/978-1-4757-4721-8`.

[25] Paul-Andre Mellies. Template games and differential linear logic. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13. IEEE. URL: `https://ieeexplore.ieee.org/document/8785830/`, `doi:10.1109/LICS.2019.8785830`.

[26] Paul-André Melliès. Categorical semantics of linear logic. *Panoramas et syntheses*, 27:15–215, 2009.

[27] Robert AG Seely. *Linear logic, ∗-autonomous categories and cofree coalgebras*, volume 92. 1989. `doi:10.1090/conm/092/1003210`.

[28] Michael Stay. Compact closed bicategories. *Theory and Applications of Categories*, 31(26):755–798, 2016. `arXiv:1301.1053`.

[29] Ross Street. Polynomials as spans. `doi:10.48550/arXiv.1903.03890`.

[30] Paul Taylor. Quantitative domains, groupoids and linear logic. In *Category Theory and Computer Science: Manchester, UK, September 5–8, 1989 Proceedings*, pages 155–181. Springer, 1989. `doi:10.1007/BFb0018351`.

[31] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics.* `https://homotopytypetheory.org/book`, Institute for Advanced Study, 2013.

[32] Benno Van Den Berg and Richard Garner. Types are weak ω-groupoids. *Proceedings of the London mathematical society*, 102(2):370–394, 2011. `doi:10.1112/plms/pdq026`.